The Journal of Machine Learning Research
Print-Archive Edition

Volume 18 (papers from 2018)
Issues 150–234

Microtome Publishing
Brookline, Massachusetts
www.mtome.com

The Journal of Machine Learning Research
Print-Archive Edition
Volume 18 (papers from 2018)
Issues 150–234

The Journal of Machine Learning Research (JMLR) is an open access journal. All articles published in JMLR are freely available via electronic distribution. This Print-Archive Edition is published annually as a means of archiving the contents of the journal in perpetuity. The contents of this edition are articles published electronically in JMLR in 2018.

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

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

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

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

JMLR Editorial Board

Journal of Machine Learning Research

Volume 18, Issues 150–234 (papers from 2018)

Journal of Machine Learning Research 18 (2018) 1-30 Submitted 12/15; Revised 10/17; Published 4/18

On Binary Embedding using Circulant Matrices

Felix X. Yu[1] felixyu@google.com
Aditya Bhaskara[2] bhaskaraaditya@gmail.com
Sanjiv Kumar[1] sanjivk@google.com
Yunchao Gong[3] yunchao@cs.unc.edu
Shih-Fu Chang[4] sfchang@ee.columbia.edu
[1]Google Research, New York, NY 10011
[2]University of Utah, Salt Lake City, UT 84112
[3]Snap, Inc., Venice, CA 90291
[4]Columbia University, New York, NY 10027

Editor: Mikhail Belkin

Abstract

Binary embeddings provide efficient and powerful ways to perform operations on large scale data. However binary embedding typically requires long codes in order to preserve the discriminative power of the input space. Thus binary coding methods traditionally suffer from high computation and storage costs in such a scenario. To address this problem, we propose Circulant Binary Embedding (CBE) which generates binary codes by projecting the data with a circulant matrix. The circulant structure allows us to use Fast Fourier Transform algorithms to speed up the computation. For obtaining k-bit binary codes from d-dimensional data, our method improves the time complexity from $\mathcal{O}(dk)$ to $\mathcal{O}(d \log d)$, and the space complexity from $\mathcal{O}(dk)$ to $\mathcal{O}(d)$.

We study two settings, which differ in the way we choose the parameters of the circulant matrix. In the first, the parameters are chosen randomly and in the second, the parameters are learned using the data. For randomized CBE, we give a theoretical analysis comparing it with binary embedding using an unstructured random projection matrix. The challenge here is to show that the dependencies in the entries of the circulant matrix do not lead to a loss in performance. In the second setting, we design a novel time-frequency alternating optimization to learn data-dependent circulant projections, which alternatively minimizes the objective in original and Fourier domains. In both the settings, we show by extensive experiments that the CBE approach gives much better performance than the state-of-the-art approaches if we fix a running time, and provides much faster computation with negligible performance degradation if we fix the number of bits in the embedding.

Keywords: structured matrix, circulant matrix, dimensionality reduction, binary embedding, FFT

1. Introduction

Sketching and dimensionality reduction have become powerful and ubiquitous tools in the analysis of large high-dimensional data sets, with applications ranging from computer vision, to biology, to finance. The celebrated Johnson-Lindenstrauss lemma says that projecting high dimensional points to a random $\mathcal{O}(\log N)$-dimensional space approximately preserves all the pairwise distances between a set of N points, making it a powerful tool for nearest neighbor search, clustering, etc. This started the paradigm of designing low dimensional *sketches* (or embeddings) of high dimensional data that can be used for efficiently solving various information retrieval problems.

More recently, *binary* embeddings (or embeddings into $\{0,1\}^k$ or $\{-1,1\}^k$) have been developed for problems in which we care about preserving the *angles* between high dimensional vectors (Li et al., 2011; Gong et al., 2013; Raginsky and Lazebnik, 2009; Gong et al., 2012b; Liu et al., 2011). The main appeal of binary embeddings stems from the fact that storing them is often much more efficient than storing real valued embeddings. Furthermore, operations such as computing the Hamming distance in binary space can be performed very efficiently either using table lookup, or hardware-implemented instructions on modern computer architectures.

In this paper, we study binary embeddings of high-dimensional data. Our goal is to address one of its main challenges: even though binary embeddings are easy to manipulate, it has been observed that obtaining high accuracy results *requires* the embeddings to be rather long when the data is high dimensional (Li et al., 2011; Gong et al., 2013; Sánchez and Perronnin, 2011). Thus in applications like computer vision, biology and finance (where high dimensional data is common), the task of computing the embedding is a bottleneck. The natural algorithms have time and space complexity $\mathcal{O}(dk)$ per input point in order to produce a k-bit embedding from a d-dimensional input. Our main contribution in this work is to improve these complexities to $\mathcal{O}(d \log d)$ for time and $\mathcal{O}(d)$ for space complexity.

Our results can be viewed as binary analogs of the recent work on *fast* Johnson-Lindenstrauss transform. Starting with the work of Ailon and Chazelle (Ailon and Chazelle, 2006), there has been a lot of beautiful work on fast algorithms for dimension reduction with the goal of preserving pairwise distances between points. Various aspects, such as exploiting sparsity, and using structured matrices to reduce the space and time complexity of dimension reduction, have been explored (Ailon and Chazelle, 2006; Matoušek, 2008; Liberty et al., 2008). But the key difference in our setting is that binary embeddings are *non-linear*. This makes the analysis tricky when the projection matrices do not have independent entries. Binary embeddings are also better suited to approximate the angles between vectors (as opposed to distances). Let us see why.

The general way to compute a binary embedding of a data point $\mathbf{x} \in \mathbb{R}^d$ is to first apply a linear transformation $\mathbf{Ax}$ (for a $k \times d$ matrix $\mathbf{A}$), and then apply a *binarization* step. We consider the natural binarization of taking the sign. Thus, for a point $\mathbf{x}$, the binary embedding into $\{-1,1\}^d$ we consider is

$$h(\mathbf{x}) = \text{sign}(\mathbf{Ax}), \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{k \times d}$ as above, and $\text{sign}(\cdot)$ is a binary map which returns element-wise sign[1]. How should one pick the matrix $\mathbf{A}$? One natural choice, in light of the Johnson-Lindenstrauss lemma, is to pick it randomly, i.e., each entry is sampled from an independent Gaussian. This *data oblivious* choice is well studied (Charikar, 2002; Raginsky and Lazebnik, 2009),

1. A few methods transform the linear projection via a nonlinear map before taking the sign (Weiss et al., 2008; Raginsky and Lazebnik, 2009).

and has the nice property that for two data vectors $\mathbf{x}, \mathbf{y}$, the ℓ_1 distance between their embeddings is proportional to the angle between $\mathbf{x}$ and $\mathbf{y}$, in expectation (over the random entries in $\mathbf{A}$). This is a consequence of the fact that for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, if $\mathbf{r}$ is drawn from $\mathcal{N}(0,1)^d$,

$$\Pr[\text{sign}\,\langle \mathbf{x}, \mathbf{r}\rangle = \text{sign}\,\langle \mathbf{y}, \mathbf{r}\rangle] = \frac{\angle(\mathbf{x}, \mathbf{y})}{\pi}. \tag{2}$$

Other data oblivious methods have also been studied in the literature, by choosing different distributions for the entries of $\mathbf{A}$. While these methods do reasonably well in practice, the natural question is if adapting the matrix to the data allows us to use shorter codes (i.e., have a smaller k) while achieving a similar error. A number of such data-dependent techniques have been proposed with different optimization criteria such as reconstruction error (Kulis and Darrell, 2009), data dissimilarity (Norouzi and Fleet, 2012; Weiss et al., 2008), ranking loss (Norouzi et al., 2012), quantization error after PCA (Gong et al., 2012a), and pairwise misclassification (Wang et al., 2010). As long as data is relatively low dimensional, these methods have been shown to be quite effective for learning compact codes.

However, the $\mathcal{O}(kd)$ barrier on the space and time complexity barrier prevents them from being applied with very high-dimensional data. For instance, to generate 10K-bit binary codes for data with 1M dimensions, a huge projection matrix will be required needing tens of GB of memory.[2]

In order to overcome these computational challenges, Gong et al. (2013) proposed a *bilinear projection* based coding method. The main idea here is to reshape the input vector $\mathbf{x}$ into a matrix $\mathbf{Z}$, and apply a bilinear projection to get the binary code:

$$h(\mathbf{x}) = \text{sign}(\mathbf{R}_1^T \mathbf{Z} \mathbf{R}_2). \tag{3}$$

When the shapes of $\mathbf{Z}, \mathbf{R}_1, \mathbf{R}_2$ are chosen appropriately[3], the method has time and space complexities $\mathcal{O}(d\sqrt{k})$ and $\mathcal{O}(\sqrt{dk})$ respectively. Bilinear codes make it feasible to work with data sets of very high dimensionality and have shown good results for a variety of tasks.

1.1 Our results

In this work, we propose a novel technique, called Circulant Binary Embedding (CBE), which is even faster than the bilinear coding. The main idea is to impose a *circulant* (described in detail in Section 3) structure on the projection matrix $\mathbf{A}$ in (1). This special structure allows us to compute the product $\mathbf{Ax}$ in time $\mathcal{O}(d \log d)$ using the Fast Fourier Transform (FFT), a tool of great significance in signal processing. The space complexity is also just $\mathcal{O}(d)$, making it efficient even for very high dimensional data. Table 1 compares the time and space complexity for the various methods outlined above.

Given the efficiency of computing the CBE, two natural questions arise: how good is the obtained embedding for various information retrieval tasks? and how should we pick the parameters of the circulant $\mathbf{A}$?

In Section 4, we study the first question for *random* CBE, i.e., when the parameters of the circulant are picked randomly (independent Gaussian, followed by its shifts). Specifically,

2. In the oblivious case, one can generate the random entries of the matrix on-the-fly (with fixed seeds) without needing to store the matrix, but this increases the computational time even further.
3. Specifically, $\mathbf{Z} \in \mathbb{R}^{\sqrt{d} \times \sqrt{d}}$, $\mathbf{R}_1, \mathbf{R}_2 \in \mathbb{R}^{\sqrt{k} \times \sqrt{d}}$.

Method	Time	Space	Time (optimization)
Unstructured	$\mathcal{O}(dk)$	$\mathcal{O}(dk)$	$\mathcal{O}(Nd^2k)$
Bilinear	$\mathcal{O}(d\sqrt{k})$	$\mathcal{O}(\sqrt{dk})$	$\mathcal{O}(Nd\sqrt{k})$
Circulant ($k \leq d$)	$\mathcal{O}(d \log d)$	$\mathcal{O}(d)$	$\mathcal{O}(Nd \log d)$
Circulant ($k > d$)	$\mathcal{O}(k \log d)$	$\mathcal{O}(k)$	$\mathcal{O}(Nk \log d)$

Table 1: Comparison of the time and space complexities. d is the input dimensionality, and k is the output dimensionality (number of bits). N is the number of instances used for learning data-dependent projection matrices. See Section 3.3 for discussions on $k < d$ and $k > d$.

we analyze the *angle estimating* property of binary embeddings (Eq.(1)), which is the basis for its use in applications. Under mild assumptions, we show that using a random circulant $\mathbf{A}$ has the same qualitative guarantees as using fully random $\mathbf{A}$. These results provide some of the few theoretical guarantees we are aware of, for non-linear circulant-based embeddings. We defer the formal statements of our results to Section 4, Theorems 3 and 4. We note that in independent and very recent work, Choromanska et al. (Choromanska et al., 2016) obtain a qualitatively similar analysis of CBE, however the bounds are incomparable to ours.

In Section 5, we study the second question, i.e., learning data-dependent circulant matrices. We propose a novel and efficient algorithm, which alternatively optimizes the objective in the original and frequency domains.

Finally in Section 7, we study the empirical performance of circulant embeddings via extensive experimentation. Compared to the state-of-the-art, our methods improve the performance dramatically for a fixed computation time. If we instead fix the number of bits in the embedding, we observe that the performance degradation is negligible, while speeding up the computation many-fold (see Section 7).

2. Background and related work

The lemma of Johnson and Lindenstrauss (Johnson and Lindenstrauss, 1984) is a fundamental tool in the area of sketching and dimension reduction. The lemma states that if we have N points in d-dimensional space, projecting them to an $\mathcal{O}(\log N)$ dimensional space (independent of d!) preserves all pairwise distances. One way of doing the projection is by a random Gaussian matrix. Formally,

Lemma 1 (Johnson Lindenstrass lemma) *Let S be a set of N points in $\mathbb{R}^d$. Let $\mathbf{A} \in \mathbb{R}^{k \times d}$ be a matrix whose entries are drawn i.i.d from $\mathcal{N}(0,1)$. Then with probability at least $1 - 2N^2 e^{-(\epsilon^2 - \epsilon^3)k/4}$*

$$(1-\epsilon)\|\mathbf{x}-\mathbf{y}\|_2 \leq \frac{1}{\sqrt{k}}\|\mathbf{A}(\mathbf{x}-\mathbf{y})\|_2 \leq (1+\epsilon)\|\mathbf{x}-\mathbf{y}\|_2 \tag{4}$$

for any $\mathbf{x}, \mathbf{y} \in S$.

When $k = \mathcal{O}(\log N/\epsilon^2)$, the probability above can be made arbitrarily close to 1. The result has also been shown to be optimal even if the projection can be chosen data-dependently (Larsen and Nelson, 2016). Due to the simplicity and theoretical support, random projection based dimensionality reduction has been applied in broad applications including approximate nearest neighbor research (Indyk and Motwani, 1998), dimensionality reduction in databases (Achlioptas, 2003), and bi-Lipschitz embeddings of graphs into normed spaces (Frankl and Maehara, 1988).

However a serious concern in a few applications is the dependence of k on the accuracy ($\mathcal{O}(1/\epsilon^2)$). The space and time complexity of dimension reduction are $\mathcal{O}(kd)$, if the computation is done in the natural way. Are there faster methods when k is reasonably large? As mentioned earlier, the line of work starting with (Ailon and Chazelle, 2006) aims to improve the time and space complexity of dimension reduction. This led to work showing Johnson-Lindenstruss-type guarantees with *structured* matrices (with some randomness), including Hadamard matrices along with a sparse random Gaussian matrix (Ailon and Chazelle, 2006), sparse matrices (Matoušek, 2008), and Lean Walsh Transformations (Liberty et al., 2008). The advantage of using structured matrices is that the space and computation cost can be dramatically reduced, yet the distance preserving property remains to be competitive.

In this context, randomized circulant matrices (which are also the main tool in our work) have been studied, starting with the works (Hinrichs and Vybíral, 2011; Vybíral, 2011). The dimension reduction comprises of random sign flips followed by multiplication by a randomized circulant matrix. For d-dimensional input, reducing the dimension to k for $k < d$ has time complexity $\mathcal{O}(d \log d)$ and space complexity $\mathcal{O}(d)$, independent of k. Proving bounds similar to Lemma 1 turns out to be much more challenging because the entries of the projection matrix are now highly dependent, and thus concentration bounds are hard to prove. The first analysis (Hinrichs and Vybíral, 2011) showed that reducing to $\mathcal{O}(\log^3 N/\epsilon^2)$ dimensions (compared to $\mathcal{O}(\log N/\epsilon^2)$ in Lemma 1) preserves all pairwise distances with high probability. This was improved to $\mathcal{O}(\log^2 N/\epsilon^2)$ in (Vybíral, 2011), and furthermore to $\mathcal{O}(\log^{(1+\delta)} N/\epsilon^2)$ in (Zhang and Cheng, 2013), using matrix-valued Bernstein inequalities. These works provide the motivation for our theoretical results, however the key difference for us is the *binarization* step, which is highly non-linear. Thus we need to develop new machinery for our analysis.

Binary embeddings. Recently, structured matrices used in the context of the fast JL transform (a combination of Hadamard and sparse random Gaussian matrices) have also been studied for binary embedding (Dasgupta et al., 2011), and more recently (Yi et al., 2015). In particular, Yi et al. (2015) showed that the method can achieve ϵ distance preserving error with $\mathcal{O}(\log N/\epsilon^2)$ bits and $\mathcal{O}(d \log d)$ computational complexity, for N points ($N \ll \epsilon\sqrt{d}$). In this work, we study the application of using the circulant matrix for binary embedding. The work extends and provides theoretical justification for our previous conference paper on this topic (Yu et al., 2014).

The idea of using structured matrices to speed up linear projection has also be exploited under the settings of deep neural networks (Cheng et al., 2015; Yang et al., 2015), and kernel approximation (Yu et al., 2015; Le et al., 2013) .

3. Circulant Binary Embedding

Let us start by describing our framework and setting up the notation that we use in the rest of the paper.

3.1 The Framework

We will now describe our algorithm for generating k-bit binary codes from d-dimensional real vectors. We start by discussing the case $k = d$ and move to the general case in Section 3.3. The key player is the circulant matrix, which is defined by a real vector $\mathbf{r} = (r_0, r_1, \cdots, r_{d-1})^T$ (Gray, 2006).

$$\mathcal{C}_{\mathbf{r}} := \begin{bmatrix} r_0 & r_{d-1} & \dots & r_2 & r_1 \\ r_1 & r_0 & r_{d-1} & & r_2 \\ \vdots & r_1 & r_0 & \ddots & \vdots \\ r_{d-2} & & \ddots & \ddots & r_{d-1} \\ r_{d-1} & r_{d-2} & \dots & r_1 & r_0 \end{bmatrix}. \tag{5}$$

Let $\mathbf{D}$ be a diagonal matrix with each diagonal entry σ_i, $i = 0, \cdots, d-1$, being a Rademacher variable (± 1 with probability $1/2$):

$$\mathbf{D} = \begin{bmatrix} \sigma_0 & & & & \\ & \sigma_1 & & & \\ & & \sigma_2 & & \\ & & & \ddots & \\ & & & & \sigma_{d-1} \end{bmatrix}. \tag{6}$$

For $\mathbf{x} \in \mathbb{R}^d$, its d-bit Circulant Binary Embedding (CBE) with $\mathbf{r} \in \mathbb{R}^d$ is defined as:

$$h(\mathbf{x}) = \text{sign}(\mathcal{C}_{\mathbf{r}}\mathbf{D}\mathbf{x}), \tag{7}$$

where $\mathcal{C}_{\mathbf{r}}$ is defined as above. Note that applying $\mathbf{D}$ to $\mathbf{x}$ is equivalent to applying a random sign flip to each coordinate of $\mathbf{x}$. The necessity of such an operation is discussed in the introduction of Section 4. Since sign flipping can be carried out as a preprocessing step for each input $\mathbf{x}$, here onwards for simplicity we will drop explicit mention of $\mathbf{D}$. Hence the binary code is given as $h(\mathbf{x}) = \text{sign}(\mathcal{C}_{\mathbf{r}}\mathbf{x})$.

3.2 Computational Complexity

The main advantage of a circulant based embedding is that it can be computed quickly using the Fast Fourier Transform (FFT). The following is a folklore result, whose proof we include for completeness.

Proposition 2 *For a d-dimensional vector $\mathbf{x}$ and any $\mathbf{r} \in \Re^d$, the d-bit CBE $\text{sign}(\mathcal{C}_r(\mathbf{D}\mathbf{x}))$ can be computed using $\mathcal{O}(d)$ space and $\mathcal{O}(d \log d)$ time.*

Proof The space complexity comes only from the storage of the vector $\mathbf{r}$ and the signs $\mathbf{D}$ (which amount to $\mathcal{O}(d)$). We never need to store the full matrix $\mathcal{C}_{\mathbf{r}}$ explicitly.

The main property of a circulant matrix is that for any vector $\mathbf{y} \in \mathbf{R}^d$, we can compute $\mathcal{C}_\mathbf{r}\mathbf{y}$ in time $\mathcal{O}(d \log d)$. This is because

$$\mathcal{C}_\mathbf{r} = \mathcal{F}_d^{-1} \text{ diag}(\mathcal{F}_d \mathbf{r}) \mathcal{F}_d, \tag{8}$$

where $\mathcal{F}_d$ is the matrix corresponding to the Discrete Fourier Transform (DFT) of periodicity N, i.e., whose (i, j)th entry is given by

$$\mathcal{F}_d(i, j) = \omega^{ij}, \tag{9}$$

where ω is the Nth root of unity $e^{-2\pi\iota/N}$. The celebrated Fast Fourier Transform algorithm (Oppenheim et al., 1999) says that for any $\mathbf{z} \in \mathbf{R}^d$, we can compute $\mathcal{F}_d\mathbf{z}$ and $\mathcal{F}_d^{-1}\mathbf{z}$ in time $\mathcal{O}(d \log d)$, using $\mathcal{O}(d)$ space. This immediately implies that we can compute $\mathcal{C}_\mathbf{r}\mathbf{y}$ within the same space and time complexity bounds. ■

3.3 Generalizing to $k \neq d$

The computation above assumed that number of bits we produce (k) is equal to the input dimension. Let us now consider the general case.

When $k < d$, we still use the circulant matrix $\mathbf{R} \in \mathbb{R}^{d\times d}$ with d parameters, but the output is set to be the first k elements in (7). This is equivalent to the operation

$$\Phi(\mathbf{x}) := \text{sign}(\mathcal{C}_{\mathbf{r},k}\mathbf{D}\mathbf{x}), \tag{10}$$

where $\mathcal{C}_{\mathbf{r},k}$ the so-called *partial circulant matrix*, which is $\mathcal{C}_\mathbf{r}$ truncated to k rows. We note that CBE with $k < d$ is not computationally more efficient than that with $k = d$.

When $k > d$, using a single $\mathbf{r}$ causes repetition of bits, so we propose using $\mathcal{C}_\mathbf{r}$ for multiple $\mathbf{r}$, and concatenating their output. This gives the computational complexity $\mathcal{O}(k \log d)$, and space complexity $\mathcal{O}(k)$. Note that as the focus of this paper is on binary embedding on high-dimensional data, from here onwards, we assume $k \leq d$. The $k > d$ case is useful in other applications such as neural network (Cheng et al., 2015) and kernel approximation (Yu et al., 2015).

3.4 Choosing the Parameters r

We have presented the general framework as well as its space and computation efficiency in this section. One critical question left unanswered is how to decide the parameter $\mathbf{r}$. As mentioned in the introduction, we consider two solutions. In Section 4, we study the randomized version, where each element of $\mathbf{r}$ is independently sampled from a unit Gaussian distribution. This is inspired by the popular Locality Sensitive Hashing (simhash) approach. Section 5 introduces an optimized version, where the parameters are optimized based on training data and an distance preserving objective function.

4. Randomized CBE – A Theoretical Analysis

We now analyze the angle preserving properties of CBE when the circulant matrix used is generated from a random d-dimensional vector. Formally, we consider the partial circulant

matrix $\mathcal{C}_{\mathbf{r},k}$, for $\mathbf{r} \sim \mathcal{N}(0, 1)^d$. The embedding we consider for an $\mathbf{x} \in \mathbb{R}^d$ is given by

$$\Phi(\mathbf{x}) := \text{sign}(\mathcal{C}_{\mathbf{r},k}\mathbf{D}\mathbf{x}). \tag{11}$$

As before, $\mathbf{D}$ is a diagonal matrix of signs. Hence the embedding uses $2d$ independent 'units' of randomness.

Now, for any two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, we have that

$$\mathbb{E}\left[\frac{1}{2k}\|\Phi(\mathbf{x}) - \Phi(\mathbf{y})\|_1\right] = \frac{\angle(\mathbf{x}, \mathbf{y})}{\pi}, \tag{12}$$

implying that the random variable $(1/2k)\|\Phi(\mathbf{x}) - \Phi(\mathbf{y})\|_1$ provides an estimate for θ/π, where $\theta := \angle(\mathbf{x}, \mathbf{y})$.

We present two main results. In the first, we bound the variance of the above angle estimate for given $\mathbf{x}, \mathbf{y}$. We compare with the variance in the *fully independent* case, i.e., when we consider the embedding $\text{sign}(\mathbf{A}\mathbf{x})$, where $\mathbf{A}$ is a $k \times d$ matrix with all entries being independent (and unit normal). In this case, the variance of the estimator in Eq. (12) is equal to $\frac{1}{k}\frac{\theta}{\pi}\left(1 - \frac{\theta}{\pi}\right)$.[4]

We show that using a circulant matrix instead of A above has a similar dependence on k, as long as the vectors are *well spread*. Formally,

Theorem 3 *Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, such that $\max\{\|\mathbf{x}\|_\infty/\|\mathbf{x}\|_2, \|\mathbf{y}\|_\infty/\|\mathbf{y}\|_2\} \leq \rho$, for some parameter $\rho < 1$, and set $\theta = \angle(\mathbf{x}, \mathbf{y})$. The variance of the averaged hamming distance of k-bit code generated by randomized CBE is*

$$var\left[\frac{1}{2k}\|\Phi_x - \Phi_y\|_1\right] \leq \frac{1}{k}\frac{\theta}{\pi}\left(1 - \frac{\theta}{\pi}\right) + 32\rho. \tag{13}$$

The variance above is over the choice of $\mathbf{r}$ and the random signs $\mathbf{D}$.

Remark. For typical vectors in $\mathbb{R}^d$, we have $\|\mathbf{x}\|_\infty/\|\mathbf{x}\|_2$ to be $O(\log d/\sqrt{d})$. Further, by using the idea from Ailon and Chazelle (Ailon and Chazelle, 2006), we can pre-process the data by multiplying it with a randomly signed Hadamard matrix, and guarantee such an ℓ_∞ bound with high probability.[5] Therefore the second term becomes negligible for large d. The above result suggests that the angle preservation performance of CBE (in term of the variance) is as good as LSH for high-dimensional data.

Our second theorem gives a large-deviation bound for the angle estimate, also assuming that the vectors are well-spread. This will then enable us to obtain a dimension reduction theorem which preserves all angles up to an additive error.

Theorem 4 *Let $\mathbf{x}, \mathbf{y} \in \mathbf{R}^d$ with $\angle(\mathbf{x}, \mathbf{y}) = \theta$, and suppose $\max\{\|\mathbf{x}\|_\infty/\|\mathbf{x}\|_2, \|\mathbf{y}\|_\infty/\|\mathbf{x}\|_2\} \leq \rho$, for some parameter ρ. Now consider the k-dimensional CBE Φ_x, Φ_y of $\mathbf{x}, \mathbf{y}$ respectively, for some $k < d$. Suppose $\rho \leq \frac{\theta^2}{16k\log(k/\delta)}$. For any $\delta > 0$, we have:*

$$\Pr\left[\left|\frac{1}{2k}\|\Phi_x - \Phi_y\|_1 - \frac{\theta}{\pi}\right| > \frac{4\log(k/\delta)}{\sqrt{k}}\right] < \delta. \tag{14}$$

4. We are computing the variance of an average of i.i.d. Bernoulli random variables which take value 1 with probability $p = \theta/\pi$.
5. However, applying this pre-processing leads to *dense* vectors, which may be memory intensive for some applications. In this case, dividing the co-ordinates into blocks of size $\sim k^2$ and performing the pre-processing on the blocks separately is better for small k.

Qualitatively, the condition on ρ is similar to the one we implicitly have in Theorem 3. Unless $\rho = o(\frac{1}{k}\frac{\theta}{\pi}(1-\frac{\theta}{\pi}))$, the additive term dominates, so for the bound to be interesting, we need this condition on ρ.

We observe that Theorem 4 implies a Johnson-Lindenstrauss type theorem.

Corollary 5 *Suppose we have N vectors $\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{N-1}$ in $\mathbf{R}^d$, and define*

$$\rho_{ij} = \max\{\|\mathbf{u}_i\|_\infty/\|\mathbf{u}_i\|_2, \|\mathbf{u}_j\|_\infty/\|\mathbf{u}_j\|_2\}, \quad \theta_{ij} = \angle(\mathbf{u}_i, \mathbf{u}_j). \tag{15}$$

Let $\epsilon > 0$ be a given accuracy parameter and let $k = C\log^2 n/\epsilon^2$. Then for all i, j such that $\rho_{ij} < \frac{\theta_{ij}^2}{16k\log(2kN^2)}$, we have

$$\left|\frac{1}{2k}\|\Phi_i - \Phi_j\|_1 - \frac{\theta_{ij}}{\pi}\right| < \epsilon, \tag{16}$$

with probability at least 3/4.

Proof We can set $\delta = 1/2N^2$ in Theorem 4 and then take a union bound over all $\binom{N}{2}$ choices of pairs i, j to obtain a failure probability $\leq 1/4$. Further, for our choice of k, setting $C = 144$ and assuming N is large enough that $k < N$, we have

$$\frac{4\log(k/\epsilon)}{\sqrt{k}} < \frac{12\delta\log N}{\sqrt{C}\cdot\log N} < \epsilon. \tag{17}$$

■

In the remainder of the section, we will prove the above theorems. We start with Theorem 3, whose proof will give a basic framework for that of Theorem 4.

4.1 Variance of the angle estimator

For a vector $\mathbf{x}$ and an index i, we denote by $s_{\to i}(\mathbf{x})$ the vector *shifted* by i positions: the jth entry of $s_{\to i}$ is the $((j-i)\text{mod } d)$'th entry of $\mathbf{x}$. Further, let us define

$$F_i = \frac{1 - \text{sign}(s_{\to i}(\mathbf{r})^T\mathbf{D}\mathbf{x})\,\text{sign}(s_{\to i}(\mathbf{r})^T\mathbf{D}\mathbf{y})}{2} - \frac{\theta}{\pi}. \tag{18}$$

where $s_{\to i}(\cdot)$ is defined as the operator circularly shifting a vector by i elements[6]. By definition, we have

$$var\left[\frac{1}{2k}\|\Phi_x - \Phi_y\|_1\right] = var\left[\frac{1}{k}\sum_{i=1}^{k} F_i\right]. \tag{19}$$

6. The above comes with a slight abuse of notation, where the first column (instead of row) of the projection matrix $\mathbf{R}$ is defined as $\mathbf{r}$.

Without loss of generality, we assume $\|\mathbf{x}\|_2, \|\mathbf{y}\|_2 = 1$ (since we only care about the angle). The mean of each F_i is zero, and thus $\mathbb{E}[\frac{1}{k}\sum_{i=1}^{k} F_i] = 0$. Thus the variance is equal to

$$\begin{aligned} var\left[\frac{1}{k}\sum_{i=0}^{k-1} F_i\right] &= \mathbb{E}\left[\frac{1}{k^2}\left(\sum_{i=0}^{k-1} F_i\right)^2\right] \qquad (20)\\ &= \mathbb{E}\left[\frac{\sum_{i=0}^{k-1} F_i^2 + \sum_{i\neq j} F_iF_j}{k^2}\right] \\ &= \frac{1}{k^2}\left(k\cdot\mathbb{E}F_1^2 + \sum_{i\neq j}\mathbb{E}(F_iF_j)\right) \\ &= \frac{1}{k}\frac{\theta}{\pi}\left(1-\frac{\theta}{\pi}\right) + \frac{1}{k^2}\sum_{i\neq j}\mathbb{E}(F_iF_j). \qquad (21)\end{aligned}$$

The last step is based on variance of the fully independent case: the variance of a Bernoulli random variable which takes value 1 with probability $p = \theta/\pi$ is $\frac{\theta}{\pi}\left(1-\frac{\theta}{\pi}\right)$.

To prove the theorem, it suffices to show that $\mathbb{E}(F_iF_j) \leq 32\rho$ for all $i \neq j$. Without loss of generality, we can assume that $i = 0$, and consider $\mathbb{E}(F_0F_j)$. By definition, it is equal to

$$\mathbb{E}\left[\left(\frac{1 - \text{sign}(\mathbf{r}^T\mathbf{D}\mathbf{x})\,\text{sign}(\mathbf{r}^T\mathbf{D}\mathbf{y})}{2} - \frac{\theta}{\pi}\right)\left(\frac{1 - \text{sign}(s_{\to j}(\mathbf{r})^T\mathbf{D}\mathbf{x})\,\text{sign}(s_{\to j}(\mathbf{r})^T\mathbf{D}\mathbf{y})}{2} - \frac{\theta}{\pi}\right)\right].$$

The trick now is to observe that

$$s_{\to j}(\mathbf{r})^T\mathbf{x} = \mathbf{r}^T s_{\to(d-j)}(\mathbf{x}). \tag{22}$$

Thus setting $t = d - j$, we can write the above as

$$\mathbb{E}\left[\left(\frac{1 - \text{sign}(\mathbf{r}^T\mathbf{D}\mathbf{x})\,\text{sign}(\mathbf{r}^T\mathbf{D}\mathbf{y})}{2} - \frac{\theta}{\pi}\right)\left(\frac{1 - \text{sign}(\mathbf{r}^T s_{\to t}(\mathbf{D}\mathbf{x}))\,\text{sign}(\mathbf{r}^T s_{\to t}(\mathbf{D}\mathbf{y}))}{2} - \frac{\theta}{\pi}\right)\right].$$

The key idea is that we expect the vector $s_{\to t}(\mathbf{D}\mathbf{x})$ to be nearly orthogonal to the space containing $\mathbf{D}\mathbf{x}, \mathbf{D}\mathbf{y}$. This is because $\mathbf{D}$ is a diagonal matrix of random signs, and $\mathbf{x}$ and $\mathbf{y}$ are vectors with small ℓ_∞ norm. We show this formally in Lemma 7.

Why does this help? Suppose for a moment that $\mathbf{u} := s_{\to t}(\mathbf{D}\mathbf{x})$ and $\mathbf{v} := s_{\to t}(\mathbf{D}\mathbf{y})$ are both orthogonal to $\text{span}\{\mathbf{D}\mathbf{x}, \mathbf{D}\mathbf{y}\}$. Then for a random Gaussian $\mathbf{r}$, the random variables $\text{sign}(\mathbf{r}^T\mathbf{u})\,\text{sign}(\mathbf{r}^T\mathbf{v})$ and $\text{sign}(\mathbf{r}^T\mathbf{D}\mathbf{x})\,\text{sign}(\mathbf{r}^T\mathbf{D}\mathbf{y})$ are independent, because the former depends only on the projection of $\mathbf{r}$ onto $\text{span}\{\mathbf{u}, \mathbf{v}\}$, while the latter depends only on the projection of $\mathbf{r}$ onto $\text{span}\{\mathbf{D}\mathbf{x}, \mathbf{D}\mathbf{y}\}$. Now if these two spaces are orthogonal, the projections of a Gaussian vector onto these spaces are independent (this is a fundamental property of multidimensional Gaussians). This implies that the expectation of the product above is equal to the product of the expectations, which is zero (each expectation is zero).

The key lemma (see below) now says that even if $\mathbf{u}$ and $\mathbf{v}$ as defined above are *nearly* orthogonal to $\text{span}\{\mathbf{D}\mathbf{x}, \mathbf{D}\mathbf{y}\}$, we still get a good bound on the expectation above.

Lemma 6 *Let $\mathbf{a}, \mathbf{b}, \mathbf{u}, \mathbf{v}$ be unit vectors in $\mathbf{R}^d$ such that $\angle(\mathbf{a}, \mathbf{b}) = \angle(\mathbf{u}, \mathbf{v}) = \theta$, and let Π be the projector onto span$\{\mathbf{a}, \mathbf{b}\}$. Suppose $\max\{\|\Pi\mathbf{u}\|, \|\Pi\mathbf{v}\|\} = \delta < 1$. Then we have*

$$\mathbb{E}\left[\left(\frac{1 - \text{sign}(\mathbf{r}^T\mathbf{a})\,\text{sign}(\mathbf{r}^T\mathbf{b})}{2} - \frac{\theta}{\pi}\right)\left(\frac{1 - \text{sign}(\mathbf{r}^T\mathbf{u})\,\text{sign}(\mathbf{r}^T\mathbf{v})}{2} - \frac{\theta}{\pi}\right)\right] \leq 2\delta. \tag{23}$$

Here, the expectation is over the choice of $\mathbf{r}$.

The proof of the above lemma is moved to Appendix A.1.

We use the lemma with $\mathbf{a} = \mathbf{Dx}$ and $\mathbf{b} = \mathbf{Dy}$. To show Theorem 3, we have to prove that

$$\mathbb{E}\left[\max\{\|\Pi\mathbf{u}\|, \|\Pi\mathbf{v}\|\}\right] \leq 16\rho, \tag{24}$$

where $\Pi, \mathbf{u}, \mathbf{v}$ are defined as in the statement of Lemma 6. The expectation now is over the choice of $\mathbf{D}$. This leads us to our next lemma.

Lemma 7 *Let $\mathbf{p}, \mathbf{q} \in \mathbb{R}^d$ be vectors that satisfy $\|\mathbf{p}\|_2 = 1$ and $\|\mathbf{q}\|_\infty < \rho$ for some parameter ρ, and suppose $\mathbf{D} := diag(\sigma_0, \sigma_1, \ldots, \sigma_{d-1})$, where σ_i are random ± 1 signs. Then for any $0 < t < d$, we have*

$$\Pr[\langle \mathbf{Dp}, s_{\to t}(\mathbf{Dq})\rangle > \gamma] \leq e^{-\gamma^2/8\rho^2}. \tag{25}$$

Note that the probability is over the choice of $\mathbf{D}$.

The proof of the above lemma is moved to Appendix A.2. We remark that the lemma only assumes that $\mathbf{p}$ is a unit vector, it need not have a small ℓ_∞ norm.

We can now complete the proof of our theorem. As noted above, we need to show (24). To recall, Π is the projector onto span$\{\mathbf{Dx}, \mathbf{Dy}\}$, and we need to bound:

$$\mathbb{E}\left[\max\{\|\Pi\mathbf{u}\|, \|\Pi\mathbf{v}\|\}\right] \leq \mathbb{E}[\|\Pi\mathbf{u}\|] + \mathbb{E}[\|\Pi\mathbf{v}\|]. \tag{26}$$

Let $\mathbf{x}, \mathbf{z}$ be an orthonormal basis for span$\{\mathbf{x}, \mathbf{y}\}$; then it is easy to see that for any diagonal $\mathbf{D}$ with ± 1 entries on the diagonal, $\mathbf{Dx}, \mathbf{Dz}$ is an orthonormal basis for span$\{\mathbf{Dx}, \mathbf{Dy}\}$. Thus

$$\mathbb{E}[\|\Pi\mathbf{u}\|] \leq \mathbb{E}[|\langle \mathbf{u}, \mathbf{Dx}\rangle| + |\langle \mathbf{u}, \mathbf{Dz}\rangle|]. \tag{27}$$

Now by Lemma 7,

$$\Pr[|\langle \mathbf{u}, \mathbf{Dx}\rangle| > t\rho] \leq e^{-t^2/4}. \tag{28}$$

Integrating over t, we get $\mathbb{E}[|\langle \mathbf{u}, \mathbf{Dx}\rangle|] \leq 4\rho$. Thus we can bound the LHS of (26) by 16ρ, completing the proof of the theorem.

4.2 The Johnson-Lindenstrauss Type Result

Next, we turn to the proof of Theorem 4, where we wish to obtain a strong tail bound. At a high level, the argument consists of two steps:

- First, show that with probability $1 - \epsilon$ over the choice of $\mathbf{D}$, the k translates of $\mathbf{x}, \mathbf{y}$ satisfy certain orthogonality properties (this is in the same spirit as Lemma 7).
- Second, conditioned on orthogonality as above, with high probability over the choice of $\mathbf{r}$, we have the desired guarantee.

Next will will show the two steps respectively. Throughout this section, we denote by $X_0, X_1, \ldots, X_{k-1}$ the k shifts of $\mathbf{Dx}$, i.e., $X_i = s_{\to(i)}(\mathbf{Dx})$; define $Y_0, \ldots, Y_{k-1}$ analogously as shifts of $\mathbf{Dy}$. We will also assume that $\rho < \frac{\theta^2}{16k\log(k/\delta)}$.

The structure we require is formally the following.

Definition 8 ((γ, k)-orthogonality) *Two sequences of k unit vectors $X_0, X_1, \ldots, X_{k-1}$ and $Y_0, Y_1, \ldots, Y_{k-1}$ are said to be (γ, k)-orthogonal if there exists a decomposition (for every i)*

$$X_i = \mathbf{u}_i + \mathbf{e}_i\ ; \quad Y_i = \mathbf{v}_i + \mathbf{f}_i \tag{29}$$

satisfying the following properties:

1. *$\mathbf{u}_i$ and $\mathbf{v}_i$ are both orthogonal to span$\{\mathbf{u}_j, \mathbf{v}_j : j \neq i\}$.*
2. *$\max_i\{\|\mathbf{e}_i\|, \|\mathbf{f}_i\|\} < \gamma$.*

The lemma of the first step, as described earlier, is the following:

Lemma 9 *Let $\mathbf{x}, \mathbf{y}$ be unit vectors with $\|\mathbf{x}\|_\infty, \|\mathbf{y}\|_\infty \leq \rho$, and $\theta = \angle(\mathbf{x}, \mathbf{y})$, and let X_i, Y_i be rotations of $\mathbf{Dx}, \mathbf{Dy}$ respectively (as defined earlier). Then w.p. $1 - \delta$ over the choice of $\mathbf{D}$, the vectors $(X_i, Y_i)_{i=1}^k$ are (γ, k) orthogonal, for $\gamma = 4\sqrt{\rho}$.*

The proof of the lemma is quite technical, and is moved to Appendix A.3.

Now suppose we have that the shifts X_i, Y_i satisfy (γ, k)-orthogonality for some $\gamma > 0$. Suppose $\mathbf{u}_i, \mathbf{v}_i, \mathbf{e}_i, \mathbf{f}_i$ are as defined earlier. (γ, k)-orthogonality gives us that $\|\mathbf{e}_i\|, \|\mathbf{f}_i\| < \gamma$, which is $\ll 1$. Roughly speaking, we use this to say that *most of the time*, $\text{sign}(\langle \mathbf{r}, X_i\rangle = \langle \mathbf{r}, \mathbf{u}_i\rangle)$. Thus determining if $\text{sign}(\langle \mathbf{r}, X_i\rangle) = \text{sign}(\langle \mathbf{r}, Y_i\rangle)$ is essentially equivalent to determining if $\text{sign}(\langle \mathbf{r}, \mathbf{u}_i\rangle) = \text{sign}(\langle \mathbf{r}, \mathbf{v}_i\rangle)$. But the latter quantities, by orthogonality, are indepedent! (because the signs depend only on the projection of $\mathbf{r}$ onto the span of $\mathbf{u}_i, \mathbf{v}_i$, which is independent for different i).[7] The main lemma of the second step is the following:

Lemma 10 *Let $(X_i, Y_i)_{i=1}^k$ be a set of vectors satisfying (γ, k)-orthogonality and $\angle(X_i, Y_i) = \theta$ for all i. Then for any $\delta > 0$ and $k > \max\{1/\gamma, \log(4/\delta)\}$, we have*

$$\Pr\left[\left|\frac{1}{k}\sum_i (\text{sign}\,\langle \mathbf{r}, X_i\rangle \neq \text{sign}\,\langle \mathbf{r}, Y_i\rangle) - \frac{\theta}{\pi}\right| > \gamma \cdot (12\log(2k/\delta))\right] < 1 - \delta. \tag{30}$$

The probability here is over the choice of $\mathbf{r}$.

The proof is deferred to Appendix A.4.

We can now complete the proof of Theorem 4. It essentially follows using Lemma 9 and Lemma 10. Note that we can apply Lemma 10 because the angle between X_i and Y_i is also θ for each i (since they are shifts of $\mathbf{x}, \mathbf{y}$).

Formally, using the value of γ defined in Lemma 9, we have that the vectors X_i, Y_i are (γ, k) orthogonal with probability $1 - \delta$. Conditioned on this, the probability that the conclusion of Lemma 10 holds with probability $1 - \delta$. Thus the overall probability of success is at least $1 - 2\delta$. The theorem is thus easily proved by plugging in the value of γ from Lemma 9, together with $\rho < 1$. This completes the proof of the Theorem.

7. Again, using the property of multi-variate Gaussians that the projections onto orthogonal directions are orthogonal.

5. Optimized Binary Embedding

In the previous section, we showed the randomized CBE has LSH-like angle preserving properties, especially for high-dimensional data. One problem with the randomized CBE method is that it does not utilize the underlying data distribution while generating the matrix $\mathbf{R}$. In the next section, we propose to learn $\mathbf{R}$ in a data-dependent fashion, to minimize the distortions due to circulant projection and binarization.

The study in this section is closely related to data-dependent binary embeddings (Wang et al., 2010; Gong et al., 2012a, 2013). Different from the above works, the projection matrix has a circulant structure instead of being unstructured. The method is also related to data-dependent dimensionality reduction techniques such as PCA. Although it has been shown that the Johnson Lindenstrauss bound is tight even in the data-dependent setting (Larsen and Nelson, 2016), it is expected that optimization can improve the result of CBE: CBE differs from conventional dimensionality reduction by using the circulant structure and binarization; the data dependent bound is based on a worst case scenario of the data, whereas we study the empirical performance on real large-scale data sets.

We propose data-dependent CBE (CBE-opt), by optimizing the projection matrix with a novel time-frequency alternating optimization. We consider the following objective function in learning the d-bit CBE. The extension of learning $k < d$ bits will be shown in Section 5.2.

$$\begin{aligned} \underset{\mathbf{B},\mathbf{r}}{\text{argmin}} \quad & ||\mathbf{B} - \mathbf{X}\mathbf{R}^T||_F^2 + \lambda||\mathbf{R}\mathbf{R}^T - \mathbf{I}||_F^2 \\ \text{s.t.} \quad & \mathbf{R} = \text{circ}(\mathbf{r}), \end{aligned} \tag{31}$$

where $\mathbf{X} \in \mathbb{R}^{N\times d}$, is the data matrix containing n training points: $\mathbf{X} = [\mathbf{x}_0, \cdots, \mathbf{x}_{N-1}]^T$, and $\mathbf{B} \in \{-1, 1\}^{N\times d}$ is the corresponding binary code matrix.[8]

In the above optimization, the first term minimizes distortion due to binarization. The second term tries to make the projections (rows of $\mathbf{R}$, and hence the corresponding bits) as uncorrelated as possible. In other words, this helps to reduce the redundancy in the learned code. If $\mathbf{R}$ were to be an orthogonal matrix, the second term will vanish and the optimization would find the best rotation such that the distortion due to binarization is minimized. However, being a circulant matrix, $\mathbf{R}$, in general, will not be orthogonal[9]. A similar objective has been used in previous works including (Gong et al., 2012a, 2013) and (Wang et al., 2010).

5.1 The Time-Frequency Alternating Optimization

The above is a difficult non-convex combinatorial optimization problem. In this section we propose a novel approach to efficiently find a local solution. The idea is to alternatively optimize the objective by fixing $\mathbf{r}$, and $\mathbf{B}$, respectively. For a fixed $\mathbf{r}$, optimizing $\mathbf{B}$ can be easily performed in the input domain ("time" as opposed to "frequency"). For a fixed $\mathbf{B}$, the circulant structure of $\mathbf{R}$ makes it difficult to optimize the objective in the input domain.

8. If the data is ℓ_2 normalized, we can set $\mathbf{B} \in \{-1/\sqrt{d}, 1/\sqrt{d}\}^{N\times d}$ to make $\mathbf{B}$ and $\mathbf{X}\mathbf{R}^T$ more comparable. This does not empirically influence the performance.
9. We note that the rank of the circulant matrices can range from 1 (an all-1 matrix) to d (an identity matrix).

Hence we propose a novel method, by optimizing $\mathbf{r}$ in the frequency domain based on DFT. This leads to a very efficient procedure.

For a fixed r. The objective is independent on each element of $\mathbf{B}$. Denote B_{ij} as the element of the i-th row and j-th column of $\mathbf{B}$. It is easy to show that $\mathbf{B}$ can be updated as:

$$\begin{aligned} B_{ij} &= \begin{cases} 1 & \text{if } \mathbf{R}_j.\mathbf{x}_i \geq 0 \\ -1 & \text{if } \mathbf{R}_j.\mathbf{x}_i < 0 \end{cases}, \\ & i = 0, \cdots, N-1. \quad j = 0, \cdots, d-1. \end{aligned} \tag{32}$$

For a fixed B. Define $\tilde{\mathbf{r}}$ as the DFT of the circulant vector $\tilde{\mathbf{r}} := \mathcal{F}(\mathbf{r})$. Instead of solving $\mathbf{r}$ directly, we propose to solve $\tilde{\mathbf{r}}$, from which $\mathbf{r}$ can be recovered by IDFT.

Key to our derivation is the fact that DFT projects the signal to a set of orthogonal basis. Therefore the ℓ_2 norm can be preserved. Formally, according to Parseval's theorem, for any $\mathbf{t} \in \mathbb{C}^d$ (Oppenheim et al., 1999),

$$||\mathbf{t}||_2^2 = (1/d)||\mathcal{F}(\mathbf{t})||_2^2. \tag{33}$$

Denote $\text{diag}(\cdot)$ as the diagonal matrix formed by a vector. Denote $\Re(\cdot)$ and $\Im(\cdot)$ as the real and imaginary parts, respectively. We use $\mathbf{B}_{i\cdot}$ to denote the i-th row of $\mathbf{B}$. With complex arithmetic, the first term in (31) can be expressed in the frequency domain as:

$$\begin{aligned} &||\mathbf{B} - \mathbf{X}\mathbf{R}^T||_F^2 = \frac{1}{d}\sum_{i=0}^{N-1} ||\mathcal{F}(\mathbf{B}_{i\cdot}^T - \mathbf{R}\mathbf{x}_i)||_2^2 \\ =&\frac{1}{d}\sum_{i=0}^{N-1} ||\mathcal{F}(\mathbf{B}_{i\cdot}^T) - \tilde{\mathbf{r}} \circ \mathcal{F}(\mathbf{x}_i)||_2^2 = \frac{1}{d}\sum_{i=0}^{N-1} ||\mathcal{F}(\mathbf{B}_{i\cdot}^T) - \text{diag}(\mathcal{F}(\mathbf{x}_i))\tilde{\mathbf{r}}||_2^2 \\ =&\frac{1}{d}\sum_{i=0}^{N-1} \left(\mathcal{F}(\mathbf{B}_{i\cdot}^T) - \text{diag}(\mathcal{F}(\mathbf{x}_i))\tilde{\mathbf{r}}\right)^T \left(\mathcal{F}(\mathbf{B}_{i\cdot}^T) - \text{diag}(\mathcal{F}(\mathbf{x}_i))\tilde{\mathbf{r}}\right) \\ =&\frac{1}{d}\Big[\Re(\tilde{\mathbf{r}})^T\mathbf{M}\Re(\tilde{\mathbf{r}}) + \Im(\tilde{\mathbf{r}})^T\mathbf{M}\Im(\tilde{\mathbf{r}}) + \Re(\tilde{\mathbf{r}})^T\mathbf{h} + \Im(\tilde{\mathbf{r}})^T\mathbf{g}\Big] + ||\mathbf{B}||_F^2, \end{aligned} \tag{34}$$

where,

$$\mathbf{M} = \text{diag}\big(\sum_{i=0}^{N-1} \Re(\mathcal{F}(\mathbf{x}_i)) \circ \Re(\mathcal{F}(\mathbf{x}_i)) + \Im(\mathcal{F}(\mathbf{x}_i)) \circ \Im(\mathcal{F}(\mathbf{x}_i))\big), \tag{35}$$

$$\mathbf{h} = -2\sum_{i=0}^{N-1} \Re(\mathcal{F}(\mathbf{x}_i)) \circ \Re(\mathcal{F}(\mathbf{B}_{i\cdot}^T)) + \Im(\mathcal{F}(\mathbf{x}_i)) \circ \Im(\mathcal{F}(\mathbf{B}_{i\cdot}^T)), \tag{36}$$

$$\mathbf{g} = 2\sum_{i=0}^{N-1} \Im(\mathcal{F}(\mathbf{x}_i)) \circ \Re(\mathcal{F}(\mathbf{B}_{i\cdot}^T)) - \Re(\mathcal{F}(\mathbf{x}_i)) \circ \Im(\mathcal{F}(\mathbf{B}_{i\cdot}^T)). \tag{37}$$

The above can be derived based on the following fact. For any $\mathbf{Q} \in \mathbb{C}^{d\times d}$, $\mathbf{s}$, $\mathbf{t} \in \mathbb{C}^d$,

$$
\begin{aligned}
&||\mathbf{s}-\mathbf{Qt}||_2^2=(\mathbf{s}-\mathbf{Qt})^H(\mathbf{s}-\mathbf{Qt}) \qquad (38)\\
=&\mathbf{s}^H\mathbf{s}-\mathbf{s}^H\mathbf{Qt}-\mathbf{t}^H\mathbf{Q}^H\mathbf{s}+\mathbf{t}^H\mathbf{Q}^H\mathbf{Qt}\\
=&\Re(\mathbf{s})^T\Re(\mathbf{s})+\Im(\mathbf{s})^T\Im(\mathbf{s})-2\Re(\mathbf{t})^T(\Re(\mathbf{Q})^T\Re(\mathbf{s})+\Im(\mathbf{Q})^T\Im(\mathbf{s}))\\
&+2\Im(\mathbf{t})^T(\Im(\mathbf{Q})^T\Re(\mathbf{s})-\Re(\mathbf{Q})^T\Im(\mathbf{s}))+\Re(\mathbf{t})^T(\Re(\mathbf{Q})^T\Re(\mathbf{Q})+\Im(\mathbf{Q})^T\Im(\mathbf{Q}))\Re(\mathbf{t})\\
&+\Im(\mathbf{t})^T(\Re(\mathbf{Q})^T\Re(\mathbf{Q})+\Im(\mathbf{Q})^T\Im(\mathbf{Q}))\Im(\mathbf{t})+2\Re(\mathbf{t})^T(\Im(\mathbf{Q})^T\Re(\mathbf{Q})-\Re(\mathbf{Q})^T\Im(\mathbf{Q}))\Im(\mathbf{t}).
\end{aligned}
$$

For the second term in (31), we note that the circulant matrix can be diagonalized by DFT matrix $\mathbf{F}_d$ and its conjugate transpose $\mathbf{F}_d^H$. Formally, for $\mathbf{R}=\text{circ}(\mathbf{r})$, $\mathbf{r}\in\mathbb{R}^d$,

$$
\mathbf{R}=(1/d)\mathbf{F}_d^H\text{diag}(\mathcal{F}(\mathbf{r}))\mathbf{F}_d. \qquad (39)
$$

Let $\text{Tr}(\cdot)$ be the trace of a matrix. Therefore,

$$
\begin{aligned}
&||\mathbf{RR}^T-\mathbf{I}||_F^2=||\frac{1}{d}\mathbf{F}_d^H(\text{diag}(\tilde{\mathbf{r}})^H\text{diag}(\tilde{\mathbf{r}})-\mathbf{I})\mathbf{F}_d||_F^2 \qquad (40)\\
=&\,\text{Tr}\left[\frac{1}{d}\mathbf{F}_d^H(\text{diag}(\tilde{\mathbf{r}})^H\text{diag}(\tilde{\mathbf{r}})-\mathbf{I})^H(\text{diag}(\tilde{\mathbf{r}})^H\text{diag}(\tilde{\mathbf{r}})-\mathbf{I})\mathbf{F}_d\right]\\
=&\,\text{Tr}\left[(\text{diag}(\tilde{\mathbf{r}})^H\text{diag}(\tilde{\mathbf{r}})-\mathbf{I})^H(\text{diag}(\tilde{\mathbf{r}})^H\text{diag}(\tilde{\mathbf{r}})-\mathbf{I})\right]\\
=&||\tilde{\mathbf{r}}^H\circ\tilde{\mathbf{r}}-\mathbf{1}||_2^2=||\Re(\tilde{\mathbf{r}})^2+\Im(\tilde{\mathbf{r}})^2-\mathbf{1}||_2^2.
\end{aligned}
$$

Furthermore, as $\mathbf{r}$ is real-valued, additional constraints on $\tilde{\mathbf{r}}$ are needed. For any $u\in\mathbb{C}$, denote $\overline{u}$ as its complex conjugate. We have the following result (Oppenheim et al., 1999): For any real-valued vector $\mathbf{t}\in\mathbb{C}^d$, $\mathcal{F}(\mathbf{t})_0$ is real-valued, and

$$
\mathcal{F}(\mathbf{t})_{d-i}=\overline{\mathcal{F}(\mathbf{t})_i},\quad i=1,\cdots,\lfloor d/2\rfloor. \qquad (41)
$$

From (34) $-$ (41), the problem of optimizing $\tilde{\mathbf{r}}$ becomes

$$
\begin{aligned}
\underset{\tilde{\mathbf{r}}}{\text{argmin}}\quad & \Re(\tilde{\mathbf{r}})^T\mathbf{M}\Re(\tilde{\mathbf{r}})+\Im(\tilde{\mathbf{r}})^T\mathbf{M}\Im(\tilde{\mathbf{r}})+\Re(\tilde{\mathbf{r}})^T\mathbf{h}\\
&+\Im(\tilde{\mathbf{r}})^T\mathbf{g}+\lambda d||\Re(\tilde{\mathbf{r}})^2+\Im(\tilde{\mathbf{r}})^2-\mathbf{1}||_2^2 \qquad (42)\\
\text{s.t.}\quad & \Im(\tilde{r}_0)=0\\
& \Re(\tilde{r}_i)=\Re(\tilde{r}_{d-i}),i=1,\cdots,\lfloor d/2\rfloor\\
& \Im(\tilde{r}_i)=-\Im(\tilde{r}_{d-i}),i=1,\cdots,\lfloor d/2\rfloor.
\end{aligned}
$$

The above is non-convex. Fortunately, the objective function can be decomposed, such that we can solve two variables at a time. Denote the diagonal vector of the diagonal matrix $\mathbf{M}$ as $\mathbf{m}$. The above optimization can then be decomposed to the following sets of optimizations.

$$
\begin{aligned}
&\underset{\tilde{r}_0}{\text{argmin}}\quad m_0\tilde{r}_0^2+h_0\tilde{r}_0+\lambda d\left(\tilde{r}_0^2-1\right)^2,\ \text{s.t. }\tilde{r}_0=\overline{\tilde{r}_0}. \qquad (43)\\
&\underset{\tilde{r}_i}{\text{argmin}}\quad (m_i+m_{d-i})(\Re(\tilde{r}_i)^2+\Im(\tilde{r}_i)^2)+2\lambda d\left(\Re(\tilde{r}_i)^2+\Im(\tilde{r}_i)^2-1\right)^2\\
&\qquad\qquad +(h_i+h_{d-i})\Re(\tilde{r}_i)+(g_i-g_{d-i})\Im(\tilde{r}_i),\quad i=1,\cdots,\lfloor d/2\rfloor.
\end{aligned}
$$

In (43), we need to minimize a 4^{th} order polynomial with one variable, with the closed form solution readily available. In (44), we need to minimize a 4^{th} order polynomial with two variables. Though the closed form solution is hard to find (requiring solution of a cubic bivariate system), a local minimum can be found by gradient descent, which in practice has constant running time for such small-scale problems. The overall objective is guaranteed to be non-increasing in each step. In practice, we find that a good solution can be reached within just 5-10 iterations. Therefore in practice, the proposed time-frequency alternating optimization procedure has running time $\mathcal{O}(Nd\log d)$.

5.2 Learning with Dimensionality Reduction

In the case of learning $k<d$ bits, we need to solve the following optimization problem:

$$
\begin{aligned}
\underset{\mathbf{B},\mathbf{r}}{\text{argmin}}\quad & ||\mathbf{BP}_k-\mathbf{XP}_k^T\mathbf{R}^T||_F^2+\lambda||\mathbf{RP}_k\mathbf{P}_k^T\mathbf{R}^T-\mathbf{I}||_F^2 \qquad (44)\\
\text{s.t.}\quad & \mathbf{R}=\text{circ}(\mathbf{r}),
\end{aligned}
$$

in which $\mathbf{P}_k=\begin{bmatrix}\mathbf{I}_k & \mathbf{O}\\ \mathbf{O} & \mathbf{O}_{d-k}\end{bmatrix}$, $\mathbf{I}_k$ is a $k\times k$ identity matrix, and $\mathbf{O}_{d-k}$ is a $(d-k)\times(d-k)$ all-zero matrix.

In fact, the right multiplication of $\mathbf{P}_k$ can be understood as a "temporal cut-off", which is equivalent to a frequency domain convolution. This makes the optimization difficult, as the objective in frequency domain can no longer be decomposed. To address this issue, we propose a simple solution in which $B_{ij}=0$, $i=0,\cdots,N-1,j=k,\cdots,d-1$ in (31). Thus, the optimization procedure remains the same, and the cost is also $\mathcal{O}(Nd\log d)$. We will show in experiments that this heuristic provides good performance in practice.

6. Discussion

6.1 Limitations of the Theory for Long Codes

As was shown in earlier works (Li et al., 2011; Gong et al., 2013; Sánchez and Perronnin, 2011) and as we see in our experiments (Section 7), long codes are necessary for high-dimensional data for all binary embedding methods, either randomized or optimized.

However, when the code length is too large, our theoretical analysis is not optimal. For instance, consider our variance bound when $k>\sqrt{d}$. Here the ρ term always dominates, because for any vector, we have $\rho\geq 1/\sqrt{d}$ (at least one entry of a unit vector is at least $1/\sqrt{d}$). In numeric simulations, we see that the variance drops as $1/k$ for a larger range of k, roughly up to d. A similar behavior holds in Theorem 4, where the condition $\rho\leq\frac{\theta^2}{16k\log(k/\delta)}$ can hold only when $k<\mathcal{O}(\sqrt{d}/\log d)$. It is an interesting open question to analyze the variance and other concentration properties for larger k.

6.2 Semi-supervised Extension

In some applications, one can have access to a few labeled pairs of similar and dissimilar data points. Here we show how the CBE formulation can be extended to incorporate such

information in learning. This is achieved by adding an additional objective term $J(\mathbf{R})$.

$$\begin{aligned} \underset{\mathbf{B},\mathbf{r}}{\text{argmin}} \quad & ||\mathbf{B} - \mathbf{X}\mathbf{R}^T||_F^2 + \lambda||\mathbf{R}\mathbf{R}^T - \mathbf{I}||_F^2 + \mu J(\mathbf{R}) \\ \text{s.t.} \quad & \mathbf{R} = \text{circ}(\mathbf{r}), \end{aligned} \tag{45}$$

$$J(\mathbf{R}) = \sum_{i,j\in\mathcal{M}} ||\mathbf{R}\mathbf{x}_i - \mathbf{R}\mathbf{x}_j||_2^2 - \sum_{i,j\in\mathcal{D}} ||\mathbf{R}\mathbf{x}_i - \mathbf{R}\mathbf{x}_j||_2^2. \tag{46}$$

Here $\mathcal{M}$ and $\mathcal{D}$ are the set of "similar" and "dissimilar" instances, respectively. The intuition is to maximize the distances between the dissimilar pairs, and minimize the distances between the similar pairs. Such a term is commonly used in semi-supervised binary coding methods (Wang et al., 2010). We again use the time-frequency alternating optimization procedure of Section 5. For a fixed $\mathbf{r}$, the optimization procedure to update $\mathbf{B}$ is the same. For a fixed $\mathbf{B}$, optimizing $\mathbf{r}$ is done in frequency domain by expanding $J(\mathbf{R})$ as below, with similar techniques used in Section 5.

$$||\mathbf{R}\mathbf{x}_i - \mathbf{R}\mathbf{x}_j||_2^2 = (1/d)||\text{diag}(\mathcal{F}(\mathbf{x}_i) - \mathcal{F}(\mathbf{x}_j))\tilde{\mathbf{r}}||_2^2. \tag{47}$$

Therefore,

$$J(\mathbf{R}) = (1/d)(\Re(\tilde{\mathbf{r}})^T\mathbf{A}\Re(\tilde{\mathbf{r}}) + \Im(\tilde{\mathbf{r}})^T\mathbf{A}\Im(\tilde{\mathbf{r}})), \tag{48}$$

where $\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_2 - \mathbf{A}_3 - \mathbf{A}_4$, and

$$\mathbf{A}_1 = \sum_{(i,j)\in\mathcal{M}} \Re(\text{diag}(\mathcal{F}(\mathbf{x}_i) - \mathcal{F}(\mathbf{x}_j)))^T\Re(\text{diag}(\mathcal{F}(\mathbf{x}_i) - \mathcal{F}(\mathbf{x}_j))), \tag{49}$$

$$\mathbf{A}_2 = \sum_{(i,j)\in\mathcal{M}} \Im(\text{diag}(\mathcal{F}(\mathbf{x}_i) - \mathcal{F}(\mathbf{x}_j)))^T\Im(\text{diag}(\mathcal{F}(\mathbf{x}_i) - \mathcal{F}(\mathbf{x}_j))), \tag{50}$$

$$\mathbf{A}_3 = \sum_{(i,j)\in\mathcal{D}} \Re(\text{diag}(\mathcal{F}(\mathbf{x}_i) - \mathcal{F}(\mathbf{x}_j)))^T\Re(\text{diag}(\mathcal{F}(\mathbf{x}_i) - \mathcal{F}(\mathbf{x}_j))), \tag{51}$$

$$\mathbf{A}_4 = \sum_{(i,j)\in\mathcal{D}} \Im(\text{diag}(\mathcal{F}(\mathbf{x}_i) - \mathcal{F}(\mathbf{x}_j)))^T\Im(\text{diag}(\mathcal{F}(\mathbf{x}_i) - \mathcal{F}(\mathbf{x}_j))). \tag{52}$$

Hence, the optimization can be carried out as in Section 5, where $\mathbf{M}$ in (34) is simply replaced by $\mathbf{M}+\mu\mathbf{A}$. The semi-supervised extension improves over the non-semi-supervised version by 2% in terms of averaged AUC on the ImageNet-25600 data set.

7. Experiments

To compare the performance of the circulant binary embedding techniques, we conduct experiments on three real-world high-dimensional data sets used by the current state-of-the-art method for generating long binary codes (Gong et al., 2013). The Flickr-25600 data set contains 100K images sampled from a noisy Internet image collection. Each image is represented by a $25,600$ dimensional feature vector. To extract the feature vectors, we use

the publicly available SIFT features, which are densely extracted at three different scales. We cluster the features into 200 centers and then aggregate them into VLAD vectors (Jégou et al., 2010) of 128 200 = 25, 600 dimensions. The ImageNet-51200 contains 100k images sampled from 100 random classes from ImageNet (Deng et al., 2009), each represented by a $51,200$ dimensional VLAD vector generated by using 400 cluster centers. The third data set (ImageNet-25600) is another random subset of ImageNet containing 100K images in $25,600$ dimensional space. All the vectors are normalized to be of unit norm.

We compare the performance of the randomized (CBE-rand) and learned (CBE-opt) versions of our circulant embeddings with the current state-of-the-art for high-dimensional data, *i.e.*, bilinear embeddings. We use both the randomized (bilinear-rand) and learned (bilinear-opt) versions. Bilinear embeddings have been shown to perform similarly or better than another promising technique called Product Quantization (Jegou et al., 2011). Finally, we also compare against the binary codes produced by the baseline LSH method (Charikar, 2002), which is still applicable to 25,600 and 51,200 dimensional feature but with much longer running time and much more space. We also show an experiment with relatively low-dimensional feature (2048, with Flickr data) to compare against techniques that perform well for low-dimensional data but do not scale to high-dimensional scenario. Example techniques include ITQ (Gong et al., 2012a), SH (Weiss et al., 2008), SKLSH (Raginsky and Lazebnik, 2009), and AQBC (Gong et al., 2012b).

Following (Gong et al., 2013; Norouzi and Fleet, 2012; Gordo and Perronnin, 2011), we use 10,000 randomly sampled instances for training. We then randomly sample 500 instances, different from the training set as queries. The performance (recall@1-100) is evaluated by averaging the recalls of the query instances. The ground-truth of each query instance is defined as its 10 nearest neighbors based on ℓ_2 distance. For each data set, we conduct two sets of experiments: *fixed-time* where code generation time is fixed and *fixed-bits* where the number of bits is fixed across all techniques. We also show an experiment where the binary codes are used for classification.

The proposed CBE method is found robust to the choice of λ in (31). For example, in the retrieval experiments, the performance difference for $\lambda = 0.1$, 1, 10, is within 0.5%. Therefore, in all the experiments, we simply fix $\lambda = 1$. For the bilinear method, in order to get fast computation, the feature vector is reshaped to a near-square matrix, and the dimension of the two bilinear projection matrices are also chosen to be close to square. Parameters for other techniques are tuned to give the best results on these data sets.

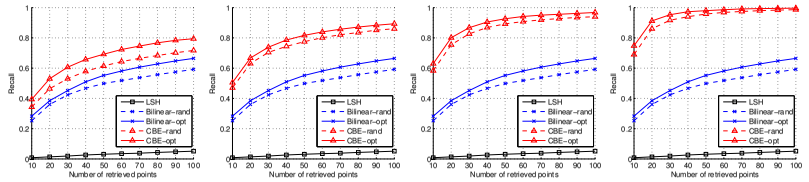
7.1 Computational Time

When generating k-bit code for d-dimensional data, the full projection, bilinear projection, and circulant projection methods have time complexity $O(kd)$, $O(\sqrt{k}d)$, and $O(d\log d)$, respectively. We compare the computational time in Table 2 on a fixed hardware. Based on our implementation, the computational time of the above three methods can be roughly characterized as $d^2 : d\sqrt{d} : 5d\log d$. Note that faster implementation of FFT algorithms will lead to better computational time for CBE by further reducing the constant factor. Due to the small storage requirement $\mathcal{O}(d)$, and the wide availability of highly optimized FFT libraries, CBE is also suitable for implementation on GPU. Our preliminary tests based on

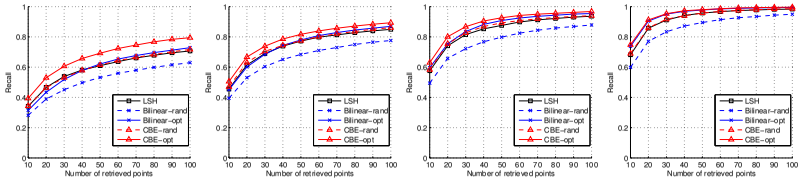
d	Full projection	Bilinear projection	Circulant projection
2^{15}	5.44×10^2	2.85	1.11
2^{17}	-	1.91×10^1	4.23
2^{20} (1M)	-	3.76×10^2	3.77×10^1
2^{24}	-	1.22×10^4	8.10×10^2
2^{27} (100M)	-	2.68×10^5	8.15×10^3

Table 2: Computational time (ms) of full projection (LSH, ITQ, SH *etc.*), bilinear projection (Bilinear), and circulant projection (CBE). The time is based on a single 2.9GHz CPU core. The error is within 10%. An empty cell indicates that the memory needed for that method is larger than the machine limit of 24GB.

GPU shows up to 20 times speedup compared with CPU. In this paper, for fair comparison, we use same CPU based implementation for all the methods.

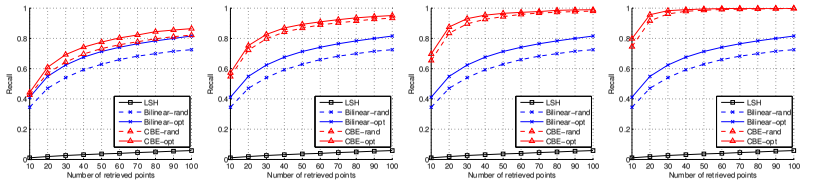


(a) #bit(CBE) = 3,200 (b) #bits(CBE) = 6,400 (c) #bits(CBE) = 12,800(d) #bits(CBE) = 25,600

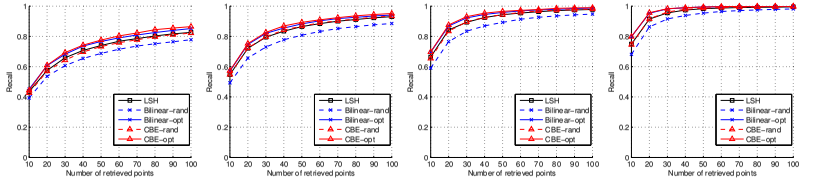


(e) # bits (all) = 3,200 (f) # bits (all) = 6,400 (g) # bits (all) = 12,800 (h) # bits (all) = 25,600

Figure 1: Recall on Flickr-25600. The standard deviation is within 1%. **First Row**: Fixed time. "# bits" is the number of bits of CBE. Other methods are using fewer bits to make their computational time identical to CBE. **Second Row**: Fixed number of bits. CBE-opt/CBE-rand are 2-3 times faster than Bilinear-opt/Bilinear-rand, and hundreds of times faster than LSH.



(a) #bits(CBE) = 3,200 (b) #bits(CBE) = 6,400 (c) #bits(CBE) = 12,800(d) #bits(CBE) = 25,600



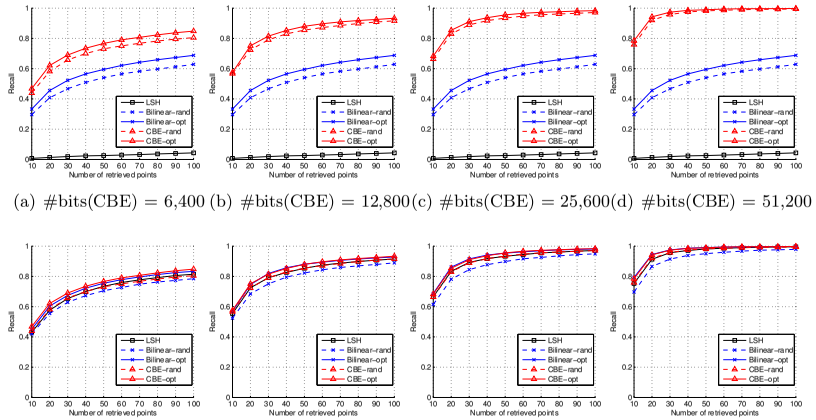
(e) # bits (all) = 3,200 (f) # bits (all) = 64,00 (g) # bits (all) = 12,800 (h) # bits (all) = 25,600

Figure 2: Recall on ImageNet-25600. The standard deviation is within 1%. **First Row**: Fixed time. "# bits" is the number of bits of CBE. Other methods are using fewer bits to make their computational time identical to CBE. **Second Row**: Fixed number of bits. CBE-opt/CBE-rand are 2-3 times faster than Bilinear-opt/Bilinear-rand, and hundreds of times faster than LSH.

7.2 Retrieval

The recalls of different methods are compared on the three data sets, shown in Figure 1 – 3. The top row in each figure shows the performance of different methods when the code generation time for all the methods is kept the same as that of CBE. For a fixed time, the proposed CBE yields much better recall than other methods. Even CBE-rand outperforms LSH and Bilinear code by a large margin. The second row compares the performance for different techniques with codes of same length. In this case, the performance of CBE-rand is almost identical to LSH even though it is hundreds of time faster. This is consistent with our analysis in Section 4. Moreover, CBE-opt/CBE-rand outperform Bilinear-opt/Bilinear-rand in addition to being 2-3 times faster.

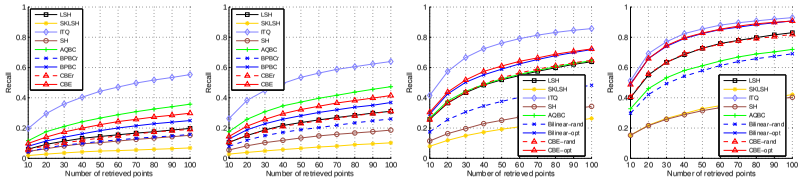
There exist several techniques that do not scale to high-dimensional case. To compare our method with those, we conduct experiments with fixed number of bits on a relatively low-dimensional data set (Flickr-2048), constructed by randomly sampling 2,048 dimensions of Flickr-25600. As shown in Figure 4, though CBE is not designed for such scenario, the CBE-opt performs better or equivalent to other techniques except ITQ which scales very poorly with d ($\mathcal{O}(d^3)$). Moreover, as the number of bits increases, the gap between ITQ and CBE becomes much smaller suggesting that the performance of ITQ is not expected to be better than CBE even if one could run ITQ on high-dimensional data.



(a) #bits(CBE) = 6,400 (b) #bits(CBE) = 12,800 (c) #bits(CBE) = 25,600 (d) #bits(CBE) = 51,200

(e) # bits (all) = 6,400 (f) # bits (all) = 12,800 (g) # bits (all) = 25,600 (h) # bits (all) = 51,200

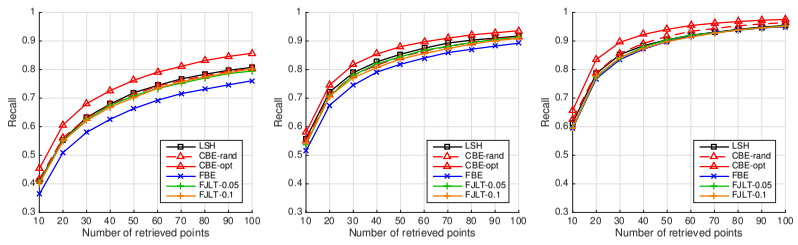
Figure 3: Recall on ImageNet-51200. The standard deviation is within 1%. **First Row**: Fixed time. "# bits" is the number of bits of CBE. Other methods are using fewer bits to make their computational time identical to CBE. **Second Row**: Fixed number of bits. CBE-opt/CBE-rand are 2-3 times faster than Bilinear-opt/Bilinear-rand, and hundreds of times faster than LSH.



(a) # bits = 256 (b) # bits = 512 (c) # bits = 1,024 (d) # bits = 2,048

Figure 4: Performance comparison on relatively low-dimensional data (Flickr-2048) with fixed number of bits. CBE gives comparable performance to the state-of-the-art even on low-dimensional data as the number of bits is increased. However, these other methods do not scale to very high-dimensional data setting which is the main focus of this work.

We also conduct additional experiments to compare CBE with the more recent Hadamard-based algorithms. The first algorithm we consider generates the binary code using the Fast



(a) # bits = 5,000 (b) # bits = 10,000 (c) # bits = 15,000

Figure 5: Recall on the Flickr-25600 data set. The methods compared are CBE-rand, CBE-opt, Fast Binary Embedding (FBE), Fast Johnson-Lindenstruss Transformation (FJLT) based methods and LSH. We follow the detailed setting of the FBE paper(Yi et al., 2015). In FJLT-p, p represents the percentage of nonzero elements in the sparse Gaussian matrix.

Johnson-Lindenstrss Transformation (FJLT). Similar to the circulant projection, FJLT has been used in dimensionality reduction (Ailon and Chazelle, 2006), deep neural networks (Yang et al., 2015), and kernel approximation (Le et al., 2013). Here, the binary code of $\mathbf{x} \in \mathbb{R}^d$ is generated as

$$h(\mathbf{x}) = \text{sign}(\mathbf{PHDx}), \tag{53}$$

where $\mathbf{P} \in \mathbb{R}^{k \times d}$ is a sparse matrix with the nonzeros entries generated iid from the standard distribution. $\mathbf{H} \in \mathbb{R}^{d \times d}$ is the Hadamard matrix, and $\mathbf{D} \in \mathbb{R}^{d \times d}$ is a diagonal matrix with random signs. Although the Hadamard transformation has computational complexity $\mathcal{O}(d \log d)$ (the multiplication with $\mathbf{H}$), this method is often slower than CBE due to the sparse Gaussian projection step (i.e., multiplication by $\mathbf{P}$).

The second method we compare with is Fast Binary Embedding (FBE). It is a theoretically sound method recently proposed in (Yi et al., 2015). FBE generates binary bits using a partial Walsh-Hadamard matrix and a set of partial Gaussian Toeplitz matrices. The method can achieve the optimal measurement complexity $\mathcal{O}(\frac{1}{\delta^2} \log N)$. Different from CBE, FBE uses multiple Toeplitz matrices each of which has more degrees of freedom than the circulant matrix ($2d$ vs. d), and their theory does not apply to the more structured CBE case. We follow the parameters settings in (Yi et al., 2015) (Flickr-25600 data set, with the number of bits 5,000, 10,000 and 15,000). Note that under the setting, FBE is at least a few times slower than CBE due to the use of multiple Toeplitz projections. Figure 5 shows the retrieval performance. Based on the experiments, in addition to being much faster than FBE and FJLT, CBE-rand provides comparable or even better performance. Another advantage of CBE is that the framework permits data-dependent optimization to further improve the performance. In all the experiments, CBE-opt achieves the highest recall by a large margin compared to other methods.

Original	LSH	Bilinear-opt	CBE-opt
25.59±0.33	23.49±0.24	24.02±0.35	24.55 ±0.30

Table 3: Multiclass classification accuracy (%) on binary coded ImageNet-25600. The binary codes of same dimensionality are 32 times more space efficient than the original features (single-float).

7.3 Classification

Besides retrieval, we also test the binary codes for classification. The advantage is to save on storage, allowing even large scale data sets to fit in memory (Li et al., 2011; Sánchez and Perronnin, 2011). We follow the asymmetric setting of (Sánchez and Perronnin, 2011) by training linear SVM on binary code sign($\mathbf{Rx}$), and testing on the original $\mathbf{Rx}$. Empirically, this has been shown to give better accuracy than the symmetric procedure. We use ImageNet-25600, with randomly sampled 100 images per category for training, 50 for validation and 50 for testing. The code dimension is set as 25,600. As shown in Table 3, CBE, which has much faster computation, does not show any performance degradation compared with LSH or bilinear codes in classification task.

8. Conclusion

We proposed a method of binary embedding for high-dimensional data. Central to our framework is to use a type of highly structured matrix, the circulant matrix, to perform the linear projection. The proposed method has time complexity $\mathcal{O}(d \log d)$ and space complexity $\mathcal{O}(d)$, while showing no performance degradation on real-world data compared with more expensive approaches ($\mathcal{O}(d^2)$ or $\mathcal{O}(d^{1.5})$). The parameters of the method can be randomly generated, where interesting theoretical analysis was carried out to show that the angle preserving quality can be as good as LSH. The parameters can also be learned based on training data with an efficient optimization algorithm.

Appendix A. Proofs of the Technical Lemmas

A.1 Proof of Lemma 6

For convenience, define $\mathbf{u}^\perp = \mathbf{u} - \Pi\mathbf{u}$, and similarly define $\mathbf{v}^\perp$. From our earlier observation about independence, we have that

$$\mathbb{E}\left[\left(\frac{1-\operatorname{sign}(\mathbf{r}^T\mathbf{a})\operatorname{sign}(\mathbf{r}^T\mathbf{b})}{2} - \frac{\theta}{\pi}\right)\left(\frac{1-\operatorname{sign}(\mathbf{r}^T\mathbf{u}^\perp)\operatorname{sign}(\mathbf{r}^T\mathbf{v}^\perp)}{2} - \frac{\theta}{\pi}\right)\right] = 0. \tag{54}$$

Because the LHS is equal to the product of the expectations, and the first term is 0. Thus the quantity we wish to bound is

$$\mathbb{E}\left[\left(\frac{1-\operatorname{sign}(\mathbf{r}^T\mathbf{a})\operatorname{sign}(\mathbf{r}^T\mathbf{b})}{2} - \frac{\theta}{\pi}\right)\left(\frac{\operatorname{sign}(\mathbf{r}^T\mathbf{u})\operatorname{sign}(\mathbf{r}^T\mathbf{v}) - \operatorname{sign}(\mathbf{r}^T\mathbf{u}^\perp)\operatorname{sign}(\mathbf{r}^T\mathbf{v}^\perp)}{2}\right)\right].$$

Now by using the fact that $\mathbb{E}[XY] \le \mathbb{E}[|X||Y|]$, together with the observation that the quantity $|(1-\operatorname{sign}(\mathbf{r}^T\mathbf{a})\operatorname{sign}(\mathbf{r}^T\mathbf{b}))/2 - \theta/\pi|$ is at most 2, we can bound the above by

$$\mathbb{E}\left[\left|\operatorname{sign}(\mathbf{r}^T\mathbf{u})\operatorname{sign}(\mathbf{r}^T\mathbf{v}) - \operatorname{sign}(\mathbf{r}^T\mathbf{u}^\perp)\operatorname{sign}(\mathbf{r}^T\mathbf{v}^\perp)\right|\right]. \tag{55}$$

This is equal to

$$2\Pr[\operatorname{sign}(\mathbf{r}^T\mathbf{u})\operatorname{sign}(\mathbf{r}^T\mathbf{v}) \neq \operatorname{sign}(\mathbf{r}^T\mathbf{u}^\perp)\operatorname{sign}(\mathbf{r}^T\mathbf{v}^\perp)], \tag{56}$$

since the term in the expectation is 2 if the product of signs is different, and 0 otherwise. To bound this, we first observe that for any two unit vectors $\mathbf{x}, \mathbf{y}$ with $\angle(\mathbf{x},\mathbf{y}) \le \epsilon$, we have $\Pr[\operatorname{sign}(\mathbf{r}^T\mathbf{x}) \neq \operatorname{sign}(\mathbf{r}^T\mathbf{y})] \le \epsilon/\pi$. We can use this to say that

$$\Pr[\operatorname{sign}(\mathbf{r}^T\mathbf{u}) \neq \operatorname{sign}(\mathbf{r}^T\mathbf{u}^\perp)] = \frac{\angle(\mathbf{u},\mathbf{u}^\perp)}{\pi}. \tag{57}$$

This angle can be bounded in our case by $(\pi/2)\cdot\delta$ by basic geometry.[10] Thus by a union bound, we have that

$$\Pr[\big(\operatorname{sign}(\mathbf{r}^T\mathbf{u}) \neq \operatorname{sign}(\mathbf{r}^T\mathbf{u}^\perp)\big) \vee \big(\operatorname{sign}(\mathbf{r}^T\mathbf{v}) \neq \operatorname{sign}(\mathbf{r}^T\mathbf{v}^\perp)\big)] \le \delta. \tag{58}$$

This completes the proof.

A.2 Proof of Lemma 7

Denoting the ith entry of $\mathbf{p}$ by p_i (so also for $\mathbf{q}$), we have that

$$S := \langle \mathbf{Dp}, s_{\to t}(\mathbf{Dq})\rangle = \sum_{i=0}^{d-1} \sigma_i\sigma_{i+t}p_iq_{i+t}. \tag{59}$$

We note that $\mathbb{E}[S] = 0$, by linearity of expectation (as $t > 0$, $\mathbb{E}[\sigma_i\sigma_{i+t}] = 0$), thus the lemma is essentially a tail bound on S. While we can appeal to standard tail bounds for quadratic forms of sub-Gaussian random variables (e.g. Hansen-Wright (Rudelson and Vershynin, 2013)), we give below a simple argument. Let us define

$$f(\sigma_0,\sigma_1,\ldots,\sigma_{d-1}) = \sum_{i=0}^{d-1} p_iq_{i+t}\sigma_i\sigma_{i+t}. \tag{60}$$

We will view f as being obtained from a martingale as follows. Define

$$Q_i := f(\sigma_0,\sigma_1,\ldots,\sigma_i,0,\ldots,0) - f(\sigma_0,\sigma_1,\ldots,\sigma_{i-1},0,\ldots,0). \tag{61}$$

In this notation, we have $S = Q_0 + Q_1 + \cdots + Q_{d-1}$.

We have the martingale property that $\mathbb{E}[Q_i|Q_0,Q_1,\ldots,Q_{i-1}] = 0$ for all i, (because σ_i is ±1 with equal probability). Further, we have the *bounded difference* property, i.e., $|Q_i| \le |p_iq_{i+t}| + |p_{i-t}q_i|$. This implies that

$$|Q_i|^2 \le 2(p_i^2q_{i+t}^2 + p_{i-t}^2q_i^2). \tag{62}$$

10. $\mathbf{u}$ is a unit vector, and $\mathbf{u}^\perp + \Pi\mathbf{u} = \mathbf{u}$, and $\|\Pi\mathbf{u}\| \le \delta$, so the angle is at most $\sin^{-1}(\delta)$.

Thus we can use Azuma's inequality to conclude that for any $\gamma > 0$,

$$\Pr[|\sum_i Q_i - \mathbb{E}[\sum_i Q_i]| > \gamma] < e^{-\frac{\gamma^2}{2\cdot\sum_i 2(p_i^2 q_{i+t}^2 + p_{i-t}^2 q_i^2)}} = e^{-\frac{\gamma^2}{8\sum_i^2 p_i^2 q_{i+t}^2}}. \quad (63)$$

We can now use the fact that $\sum_i p_i^2 q_{i+t}^2 \leq \rho^2 \sum_i p_i^2 = \rho^2$ (since $\|\mathbf{p}\|_2 = 1$ and $\|\mathbf{q}\|_\infty \leq \rho$). This establishes the lemma.

A.3 Proof of Lemma 9

First, using Lemma 7 we have, for any $i \neq j$ and $c > 0$,

$$\Pr[|\langle X_i, Y_j\rangle| > c] < e^{-c^2/8\rho^2}. \quad (64)$$

We have a similar bound for $\Pr[|\langle X_i, X_j\rangle| > c]$. Thus by setting $c = 4\rho\sqrt{\log(k/\delta)}$ (δ as in the statement of the lemma), we can take a union bound over all k^2 choices of $i \neq j$ and conclude that w.p. at least $1 - \delta$, we have

$$\max_{i\neq j}\{|\langle X_i, X_j\rangle|, |\langle X_i, Y_j\rangle|\} < 4\rho\sqrt{\log(k/\delta)}. \quad (65)$$

We now prove that whenever Eq. (65) holds, we obtain (γ, k) orthogonality for the desired γ. Let us start with a basic fact in linear algebra.

Lemma 11 *Let A be an $d \times k$ matrix with $\sigma_k(A) \geq \tau$, for some parameter τ. Then any unit vector in the column span of A can be written as $\sum_i \alpha_i A_i$, with $\sum_i \alpha_i^2 \leq 1/\tau^2$.*

Proof By the definition of σ_k, we have that for any α_i, $\|\sum_i \alpha_i A_i\|_2^2 \geq \tau^2 \left(\sum_i \alpha_i^2\right)$. Thus for any unit vector $\sum_i \alpha_i A_i$, we have $\sum_i \alpha_i^2 \leq 1/\tau^2$. ■

Now let B be the $d \times 2k$ matrix whose columns are $X_1, Y_1, X_2, Y_2, \ldots, X_k, Y_k$ in that order. Consider the entries of $B^T B$. Since X_i, Y_i are unit vectors, the diagonals are all 1. The $(2i-1, 2i)$th and $(2i, 2i-1)$th entries are exactly $\cos\theta$, because the angle between X_i, Y_i is θ. The rest of the entries are of magnitude $< \eta := 4\rho\sqrt{\log(k/\delta)}$.

Thus if we consider $M = B^T B - I$ (diagonal removed from $B^T B$), we have $-(\cos\theta + k\eta)I \preceq M \preceq (\cos\theta + k\eta)I$ (diagonal dominance). Thus we conclude that $B^T B$ has all its eigenvalues $\geq 1 - \cos\theta - k\eta$. Since $\theta \in (0, \pi/2)$, we can use the standard inequality $\cos\theta < 1 - \theta^2/2$ to conclude that the eigenvalues are $\geq \theta^2/2 - k\eta$. Now by our assumption on ρ, we have that $k\eta < \theta^2/4$. This implies that all the eigenvalues are $\geq \theta^2/4$.

Thus we have $\sigma_{2k}^2(B) \geq \theta^2/4$. We prove now that this lets us obtain a decomposition that helps us prove (γ, k)-orthogonality. A crucial observation is the following.

Lemma 12 *The projection of X_i onto span$\{X_1, Y_1, X_2, Y_2, \ldots, X_{i-1}, Y_{i-1}\}$ has length at most $\frac{2\eta\sqrt{2k}}{\theta}$.*

Proof Let $\mathcal{S}$ denote span$\{X_1, Y_1, \ldots, X_{i-1}, Y_{i-1}\}$. By definition, the squared of the length of projection is equal to $\max\{\langle y, X_i\rangle^2 \mid y \in \mathcal{S} \text{ and } \|y\|_2 = 1\}$ (this is how the projection onto a subspace can be defined).

To bound this, consider any unit vector $y \in \mathcal{S}$, and suppose we write it as $\sum_{j<i} \alpha_j X_j + \beta_j Y_j$. Let B' be the matrix that has columns X_j, Y_j, $j < i$. Then it is straightforward to see that $\sigma_{2(i-1)}(B') \geq \sigma_{2k}(B) \geq \theta/2$. Thus Claim 11 implies that $\sum_{j<i} \alpha_j^2 + \beta_j^2 \leq 4/\theta^2$. This means that

$$\langle X_i, y\rangle^2 = \big(\sum_{j<i} \alpha_j \langle X_j, X_i\rangle + \beta_j \langle Y_j, X_i\rangle\big)^2 \quad (66)$$

$$\leq \big(\sum_{j<i} \alpha_j^2 + \beta_j^2\big)\big(\sum_{j<i} \langle X_j, X_i\rangle^2 + \langle Y_j, X_i\rangle^2\big) \quad (67)$$

$$\leq \frac{4}{\theta^2} \cdot (2i-2)\eta^2. \quad (68)$$

(In the first step, we used Cauchy-Schwartz.) Taking square roots now gives the claim. ■

Now we perform the following procedure on the vectors (it is essentially Gram-Schmidt orthonormalization, with the slight twist that we deal with X_i, Y_i together):

1. Initialize: $\mathbf{u}_1 = X_1, \quad \mathbf{e}_1 = 0, \quad \mathbf{v}_1 = Y_1, \quad \mathbf{f}_1 = 0$.
2. For $i = 2, \ldots, k$, we set $\mathbf{u}_i, \mathbf{v}_i$ to be the projections of X_i, Y_i (respectively) orthogonal to span$\{X_1, Y_1, \ldots, X_{i-1}, Y_{i-1}\}$. Set $\mathbf{e}_i = X_i - \mathbf{u}_i$ and $\mathbf{f}_i = Y_i - \mathbf{v}_i$.

The important observation is that for any i, we have

$$\text{span}\{\mathbf{u}_j, \mathbf{v}_j : j < i\} = \text{span}\{\mathbf{u}_j, \mathbf{v}_j, \mathbf{e}_j, \mathbf{f}_j : j < i\} = \text{span}\{X_j, Y_j : j < i\}. \quad (69)$$

This is because by definition, $\mathbf{e}_i, \mathbf{f}_i \in \text{span}\{X_j, Y_j : j < i\}$ for all i. Thus we have that $\mathbf{u}_i$ and $\mathbf{v}_i$ satisfy the first condition in Definition 8. It just remains to analyze the lengths. Now we can use Claim 12 to conclude that

$$\|\mathbf{e}_i\|_2^2, \|\mathbf{f}_i\|_2^2 < \frac{8k\eta^2}{\theta^2} = \frac{128 \cdot k\rho^2 \log(k/\delta)}{\theta^2}. \quad (70)$$

Once again, we use the bound on ρ to conclude that this quantity is at most 16ρ. This completes the proof of Lemma 9, with $\gamma = 4\sqrt{\rho}$.

A.4 Proof of Lemma 10

We start with a simple claim about the angle between $\mathbf{u}_i$ and $\mathbf{v}_i$.

Lemma 13 *For all i, we have $\angle(\mathbf{u}_i, \mathbf{v}_i) \in (\theta - \pi\gamma, \theta + \pi\gamma)$.*

Proof The angle between X_i and $\mathbf{u}_i$ is at most $\sin^{-1}(\gamma) < (\pi/2)\gamma$. So also, the angle between Y_i and $\mathbf{v}_i$ is at most $(\pi/2)\gamma$. Thus the angle between $\mathbf{u}_i, \mathbf{v}_i$ is in the interval $(\theta - \pi\gamma, \theta + \pi\gamma)$ (by triangle inequality for the geodesic distance). ■

Let $\eta > 0$ be a parameter we will fix later (it will be a constant times $\gamma\sqrt{\log(k/\delta)}$). For all i, we define the following events:

$$E_i: \quad \min\{\langle \mathbf{r}, \mathbf{u}_i\rangle, \langle \mathbf{r}, \mathbf{v}_i\rangle\} < \eta \quad (71)$$

$$F_i: \quad \neg E_i \text{ and } \text{sign}\langle \mathbf{r}, \mathbf{u}_i\rangle \neq \text{sign}\langle \mathbf{r}, \mathbf{v}_i\rangle \quad (72)$$

The following claim now follows easily.

Lemma 14 *For any i, we have*

$$\Pr[E_i] \leq 2\eta, \tag{73}$$

$$\Pr[F_i] \in \left(\frac{\theta}{\pi} - \pi\gamma - 2\eta, \frac{\theta}{\pi}\right) \tag{74}$$

Proof The first inequality follows from the small ball probability of a univariate Gaussian (since $\langle \mathbf{r}, \mathbf{u}_i \rangle$ is a Gaussian of unit variance), and the second follows from Claim 13 and (73). ■

We will set η to be larger than $\pi\gamma$, so the RHS in (74) can be replaced with $(\theta/\pi - 3\eta, \theta/\pi)$. Furthermore, the events above for a given i depend *only* on the projection of $\mathbf{r}$ to $\text{span}\{\mathbf{u}_i, \mathbf{v}_i\}$; thus they are independent for different i. Let us abuse notation slightly and denote by E_i also the indicator random variable for the event E_i (so also F_i). Then by standard Chernoff bounds, we have for any $\tau > 0$,

$$\Pr\left[\sum_i E_i \geq 2k\eta + k\tau\right] < e^{-\frac{k\tau^2}{4\eta+\tau}}, \tag{75}$$

$$\Pr\left[\sum_i F_i \notin \left(\frac{k\theta}{\pi} - 3k\eta - k\tau, \frac{k\theta}{\pi} + k\tau\right)\right] < 2e^{-\frac{k\tau^2}{\theta+\tau}}. \tag{76}$$

Finally let H denote the event:

$$\max_i \{|\langle \mathbf{r}, \mathbf{e}_i \rangle|, |\langle \mathbf{r}, \mathbf{f}_i \rangle|\} \geq \eta. \tag{77}$$

For any i, since $\|\mathbf{e}_i\| < \gamma$, we have $\Pr[|\langle \mathbf{r}, \mathbf{e}_i \rangle| > t\gamma] \leq e^{-t^2/2}$. We can use the same bound with $\mathbf{f}_i$, and take a union bound over all i, to conclude that $\Pr[H] \leq 2k \cdot e^{-\eta^2/2\gamma^2}$.

Let us call a choice of $\mathbf{r}$ *good* if neither of the events in (75)-(76) above occur, and additionally H does not occur. Clearly, the probability of an $\mathbf{r}$ being good is at least $1 - \delta$, provided τ and η are chosen such that the RHS of the tail bounds above are all made $\leq \delta/4$.

Before setting these values, we note that for a good $\mathbf{r}$,

$$\frac{1}{k}\sum_i \mathbf{1}\{\text{sign}\,\langle \mathbf{r}, X_i \rangle \neq \text{sign}\,\langle \mathbf{r}, Y_i \rangle\} \in \left(\frac{\theta}{\pi} - 3\eta - \tau, \frac{\theta}{\pi} + 2\eta + 2\tau\right). \tag{78}$$

This is because whenever $F_i \wedge \neg H$ occurs, we have $\text{sign}\,\langle \mathbf{r}, X_i \rangle \neq \text{sign}\,\langle \mathbf{r}, Y_i \rangle$, and thus the LHS above is at least $\frac{\theta}{\pi} - 3\eta - \tau$. Also if we have $\neg H$, then the only way we can have $\text{sign}\,\langle \mathbf{r}, X_i \rangle \neq \text{sign}\,\langle \mathbf{r}, Y_i \rangle$ is if either F_i occurs, or if E_i occurs (in the latter case, it is not necessary that the signs are unequal). Thus we can upper bound the LHS by $\frac{\theta}{\pi} + 2\eta + 2\tau$.

Let us now set the values of η and τ. From the above, we need to ensure:

$$\frac{k\tau^2}{4\eta+\tau} \geq \log(4/\delta), \quad \frac{k\tau^2}{\theta+\tau} \geq \log(8/\delta), \quad \text{and} \quad \frac{\eta^2}{2\gamma^2} \geq \log(4k/\delta). \tag{79}$$

Thus we set $\eta = 2\gamma\sqrt{\log(4k/\delta)}$, and

$$\tau \geq \max\left\{\frac{2\log(8/\delta)}{k}, \sqrt{\frac{2\theta\log(8/\delta)}{k}}, \sqrt{\frac{8\eta\log(4/\delta)}{k}}\right\}. \tag{80}$$

For the above inequality to hold, it suffices to set

$$\tau \geq \frac{8\log(1/\delta)}{\sqrt{k}}. \tag{81}$$

This gives the desired bound on the deviation in the angle.

References

Dimitris Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 2003.

Nir Ailon and Bernard Chazelle. Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform. In *ACM Symposium on Theory of Computing*, 2006.

Moses S Charikar. Similarity estimation techniques from rounding algorithms. In *ACM Symposium on Theory of Computing*, 2002.

Yu Cheng, Felix Xinnan Yu, Rogerio Feris, Sanjiv Kumar, Alok Choudhary, and Shih-Fu Chang. An exploration of parameter redundancy in deep networks with circulant projections. In *IEEE International Conference on Computer Vision*, 2015.

Anna Choromanska, Krzysztof Choromanski, Mariusz Bojarski, Tony Jebara, Sanjiv Kumar, and Yann LeCun. Binary embeddings with structured hashed projections. In *International Conference on Machine Learning*, 2016.

Anirban Dasgupta, Ravi Kumar, and Tamás Sarlós. Fast locality-sensitive hashing. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2011.

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

Peter Frankl and Hiroshi Maehara. The johnson-lindenstrauss lemma and the sphericity of some graphs. *Journal of Combinatorial Theory, Series B*, 44(3):355–362, 1988.

Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (99):1, 2012a.

Yunchao Gong, Sanjiv Kumar, Vishal Verma, and Svetlana Lazebnik. Angular quantization-based binary codes for fast similarity search. In *Advances in Neural Information Processing Systems*, 2012b.

Yunchao Gong, Sanjiv Kumar, Henry A Rowley, and Svetlana Lazebnik. Learning binary codes for high-dimensional data using bilinear projections. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

Albert Gordo and Florent Perronnin. Asymmetric distances for binary embeddings. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

Robert M Gray. *Toeplitz and circulant matrices: A review*. Now Pub, 2006.

Aicke Hinrichs and Jan Vybíral. Johnson-Lindenstrauss lemma for circulant matrices. *Random Structures & Algorithms*, 39(3):391–398, 2011.

Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *ACM Symposium on Theory of Computing*, 1998.

Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33 (1):117–128, 2011.

William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary Mathematics*, 26(189-206):1, 1984.

Brian Kulis and Trevor Darrell. Learning to hash with binary reconstructive embeddings. In *Advances in Neural Information Processing Systems*, 2009.

Kasper Green Larsen and Jelani Nelson. Optimality of the Johnson-Lindenstrauss Lemma. *arXiv:1609.02094*, 2016.

Quoc Le, Tamás Sarlós, and Alex Smola. Fastfood – approximating kernel expansions in loglinear time. In *International Conference on Machine Learning*, 2013.

Ping Li, Anshumali Shrivastava, Joshua Moore, and Arnd Christian Konig. Hashing algorithms for large-scale learning. In *Advances in Neural Information Processing Systems*, 2011.

Edo Liberty, Nir Ailon, and Amit Singer. Dense fast random projections and lean walsh transforms. *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 512–522, 2008.

Wei Liu, Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. Hashing with graphs. In *International Conference on Machine Learning*, 2011.

Jiří Matoušek. On variants of the Johnson–Lindenstrauss lemma. *Random Structures & Algorithms*, 33(2):142–156, 2008.

Mohammad Norouzi and David Fleet. Minimal loss hashing for compact binary codes. In *International Conference on Machine Learning*, 2012.

Mohammad Norouzi, David Fleet, and Ruslan Salakhutdinov. Hamming distance metric learning. In *Advances in Neural Information Processing Systems*, 2012.

Alan V Oppenheim, Ronald W Schafer, John R Buck, et al. *Discrete-time signal processing*, volume 5. Prentice Hall Upper Saddle River, 1999.

Maxim Raginsky and Svetlana Lazebnik. Locality-sensitive binary codes from shift-invariant kernels. In *Advances in Neural Information Processing Systems*, 2009.

Mark Rudelson and Roman Vershynin. Hanson-wright inequality and sub-gaussian concentration. *Electron. Commun. Probab*, 18(0), 2013.

Jorge Sánchez and Florent Perronnin. High-dimensional signature compression for large-scale image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

Jan Vybíral. A variant of the Johnson–Lindenstrauss lemma for circulant matrices. *Journal of Functional Analysis*, 260(4):1096–1105, 2011.

Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. Sequential projection learning for hashing with compact codes. In *International Conference on Machine Learning*, 2010.

Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. In *Advances in Neural Information Processing Systems*, 2008.

Zichao Yang, Marcin Moczulski, Misha Denil, Nando de Freitas, Alex Smola, Le Song, and Ziyu Wang. Deep fried convnets. In *IEEE International Conference on Computer Vision*, 2015.

Xinyang Yi, Constantine Caramanis, and Eric Price. Binary embedding: Fundamental limits and fast algorithm. In *International Conference on Machine Learning*, 2015.

Felix Xinnan Yu, S. Kumar, Y. Gong, and S.-F. Chang. Circulant binary embedding. In *International Conference on Machine Learning*, 2014.

Felix Xinnan Yu, Sanjiv Kumar, Henry Rowley, and Shih-Fu Chang. Compact nonlinear maps and circulant extensions. *arXiv preprint arXiv:1503.03893*, 2015.

Hui Zhang and Lizhi Cheng. New bounds for circulant Johnson-Lindenstrauss embeddings. *arXiv preprint arXiv:1308.6339*, 2013.

Journal of Machine Learning Research 18 (2018) 1-52 Submitted 11/16; Revised 11/17; Published 4/18

Variational Fourier Features for Gaussian Processes

James Hensman[*] JAMES@PROWLER.IO
Nicolas Durrande[†] NICOLAS@PROWLER.IO
PROWLER.io
66-68 Hills Road
Cambridge, CB2 1LA, UK

Arno Solin ARNO.SOLIN@AALTO.FI
Aalto University
FI-00076 AALTO
Espoo, Finland

Editor: Manfred Opper

Abstract

This work brings together two powerful concepts in Gaussian processes: the variational approach to sparse approximation and the spectral representation of Gaussian processes. This gives rise to an approximation that inherits the benefits of the variational approach but with the representational power and computational scalability of spectral representations. The work hinges on a key result that there exist spectral features related to a finite domain of the Gaussian process which exhibit almost-independent covariances. We derive these expressions for Matérn kernels in one dimension, and generalize to more dimensions using kernels with specific structures. Under the assumption of additive Gaussian noise, our method requires only a single pass through the data set, making for very fast and accurate computation. We fit a model to 4 million training points in just a few minutes on a standard laptop. With non-conjugate likelihoods, our MCMC scheme reduces the cost of computation from $\mathcal{O}(NM^2)$ (for a sparse Gaussian process) to $\mathcal{O}(NM)$ per iteration, where N is the number of data and M is the number of features.

Keywords: Gaussian processes, Fourier features, variational inference

1. Introduction

Efficient computation in Gaussian process (GP) models is of broad interest across machine learning and statistics, with applications in the fields of spatial epidemiology (Diggle, 2013; Banerjee et al., 2014), robotics and control (Deisenroth et al., 2015), signal processing (e.g. Särkkä et al., 2013), Bayesian optimization and probabilistic numerics, (e.g. Osborne, 2010; Briol et al., 2016; Hennig et al., 2015), and many others.

Computation is challenging for several reasons. First, the computational complexity usually scales cubically with the number of data N, as one must decompose a $N \times N$ dense covariance matrix. Second, the posterior is both high-dimensional and non-Gaussian if the data likelihood is not conjugate. Third, GP models have a hierarchical structure, with the latent values of the function being conditioned on parameters of the covariance function.

In this work, we adopt a variational Bayesian framework for dealing with these issues. By this we mean that we minimize the Kullback-Leibler (KL) divergence $\mathrm{KL}[q||p]$, where q denotes an approximation to the posterior, and p is the posterior itself. Many authors have applied this framework to GP problems before: perhaps the earliest example is Csató et al. (2002). Seeger (2003) also made use of this KL criterion in the context of Gaussian processes. Particularly influentuial works include Titsias (2009), who developed the first sparse GP using the KL divergence, and Opper and Archambeau (2009), who made a case for this approximation in the non-conjugate (but not sparse) case.

The variational framework for Gaussian processes has several advantages. First, the three challenges mentioned above can be tackled by a single unified objective. Second, the accuracy of the approximation can easily be shown to increase monotonically with increasing complexity of the proposed approximating family (i.e. more inducing points is always better). Third, the framework provides an evidence lower bound (or ELBO), which can be evaluated to compare different approximations.

In this work, we combine the variational approach with Fourier features; we refer to our method as Variational Fourier Features (VFF). Whilst most sparse Gaussian process approximations rely on *inducing* or *pseudo* inputs, which lie in the same domain as the inputs to the GP, Fourier features lie in the spectral domain of the process. As inducing-point approximations build up the posterior through kernel functions, Fourier features build up the approximate posterior through sinusoids. Our approach differs from existing works which use random frequencies (Rahimi and Recht, 2008, 2009) or optimized frequencies (Lázaro-Gredilla et al., 2010), in that we use frequencies placed on a regular grid (cf. Solin and Särkkä, 2014), which is key in our derivation of a computationally convenient matrix structure.

Naive combination of Fourier features with sparse GP approaches is not feasible because the standard Fourier transform of a stochastic process does not converge, meaning that these features are associated with random variables of infinite variance. Matthews et al. (2016) have shown that valid inducing features for the variational approach require finite, deterministic projections of the process. To combat this, Figueiras-Vidal and Lázaro-Gredilla (2009) used a windowing technique, which made the integrals converge to valid inducing features (for the squared-exponential kernel), but lost perhaps the most important aspect of a Fourier approach: that the features should be independent. In this work, we combine a finite window with a Reproducing Kernel Hilbert Space (RKHS) projection to construct features that are *almost* independent, having covariance structures that are easily decomposed. This gives rise to a complexity of $\mathcal{O}(NM)$, comparing favourably with the $\mathcal{O}(NM^2)$ complexity of other variational approaches, where M denotes the number of features.

The contributions of this paper are the following:

- We introduce a novel approximation scheme for representing a GP using a Fourier decomposition. We limit our interest to the widely used Matérn family of stationary kernels, for which our proposed decomposition exhibits an almost-independent structure.

∗. This work was partially completed whilst JH was affiliated to the Centre for Health Information, Computation, and Statistcs (CHICAS), Faculty of Health and Medicine, Lancaster University, UK.

†. This work was partially completed whilst ND was affiliated to Institut Fayol–LIMOS, Mines Saint-Étienne, 158 cours Fauriel, Saint-Étienne, France.

- We combine our approximation scheme with the variational Bayesian framework for Gaussian processes, and present how the methodology generalizes to GP inference problems with general likelihoods.

Consequently, this paper aims to provide an approximate Gaussian process inference scheme which is theoretically sound, has strong representative power, is extremely fast, and—most importantly—works well in practice.

The rest of this paper is structured as follows. Section 2 reviews the relevant background related to Gaussian process inference and previous work in sparse and spectral approximations to GP models. In Section 3 the core methodology for Variational Fourier Features is presented for the one-dimensional case. In Section 4 this is extended to multidimensional cases with additive and product structures. Implementation details and computational complexity of the method are covered in Section 5. Section 6 is dedicated to a set of illustrative toy examples and a number of empirical experiments, where practical aspects of Variational Fourier Feature inference are demonstrated. Finally, we conclude the paper with a discussion in Section 7.

2. Background

In this section, we first set up the family of Gaussian process models that we will consider, which allows us to establish some notation. Next, we review some basic results regarding the spectrum of stationary Gaussian processes, and recall how Fourier features approximate the kernel. We relate sparse Gaussian process approximations and Fourier approximations by explicating them as alternative models (Quiñonero-Candela and Rasmussen, 2005). We then recap the variational approximation to Gaussian processes, including expressions for sparse approximations and approximations for non-conjugate likelihoods. The two final subsections in this section discuss decomposition of Gaussian processes: we first link decomposition and conditioning and then discuss inter-domain decomposition.

2.1 Gaussian process models

Gaussian processes are distributions over functions, defined by a mean function and a covariance function (see Williams and Rasmussen, 2006, for an introduction). In this section we consider GPs over the real line, $x \in \mathbb{R}$. Making the standard assumption that the mean function is zero, we write

$$f(x) \sim \mathcal{GP}\left(0,\, k(x, x')\right) . \tag{1}$$

The data $\mathbf{y} = [y_n]_{n=1}^N$ at locations $\mathbf{X} = [x_n]_{n=1}^N$ are conditioned on the function evaluations $\mathbf{f} = [f(x_n)]_{n=1}^N$ through some factorising likelihood

$$p(\mathbf{y} \,|\, f(x)) = p(\mathbf{y} \,|\, \mathbf{f}) = \prod_n p(y_n \,|\, f(x_n)) \,, \tag{2}$$

which we do not in general assume to be Gaussian. A defining property of Gaussian processes is that any finite number of function evaluations follow a multivariate normal distribution, so

$$p(\mathbf{f}) = \mathcal{N}(\mathbf{0},\, \mathbf{K_{ff}}) \,, \tag{3}$$

and the process conditioned on these values is given by a conditional Gaussian process:

$$f(x) \,|\, \mathbf{f} \sim \mathcal{GP}\left(\mathbf{k_f}(x)^\top \mathbf{K_{ff}^{-1}} \mathbf{f},\; k(x,x') - \mathbf{k_f}(x)^\top \mathbf{K_{ff}^{-1}} \mathbf{k_f}(x')\right) . \tag{4}$$

To compute the posterior over functions given the data $f(x) \,|\, \mathbf{y}$, it is possible to compute the posterior only for the finite set $p(\mathbf{f} \,|\, \mathbf{y})$ using standard methodologies like Markov Chain Monte Carlo (MCMC) (e.g. Murray et al., 2010; Filippone et al., 2013) or variational Bayes (e.g. Opper and Archambeau, 2009; Khan et al., 2012, 2013). Evaluating the function at a test point means averaging equation (4) over this posterior $p(\mathbf{f} \,|\, \mathbf{y})$.

In our notation, the matrix $\mathbf{K_{ff}}$ is given by evaluating the covariance function at all pairs of data points $\mathbf{K_{ff}}[i,j] = k(x_i, x_j)$. The vector valued function $\mathbf{k_f}(x)$ is given similarly, $\mathbf{k_f}(x) = [k(x_1, x), k(x_2, x), \ldots, k(x_n, x)]^\top$. We omit dependence on covariance function parameters for clarity.

2.2 Spectral representations of stationary covariances

Stationary Gaussian processes are those with a covariance function that can be written as a function of the distance between observations, $k(x,x') = k(|x - x'|) = k(r)$. Of particular interest in this work will be the Matérn family with half-integer orders, the first three of which are

$$\begin{aligned} k_{\frac{1}{2}}(r) &= \sigma^2 \exp(-r/\ell) \,, \\ k_{\frac{3}{2}}(r) &= \sigma^2 (1 + \sqrt{3} r/\ell) \exp(-\sqrt{3} r/\ell) \,, \\ k_{\frac{5}{2}}(r) &= \sigma^2 (1 + \sqrt{5} r/\ell + \tfrac{5}{3} r^2/\ell^2) \exp(-\sqrt{5} r/\ell) \,, \end{aligned} \tag{5}$$

where σ^2 and ℓ are variance and lengthscale parameters respectively.

Bochner's theorem (Akhiezer and Glazman, 1993) tells us that any continuous positive definite function, such as a covariance function, can be represented as the Fourier transform of a positive measure. If the measure has a density, it is known as the spectral density $s(\omega)$ of the covariance function. This gives rise to the Fourier duality of spectral densities and covariance functions, known as the Wiener-Khintchin theorem. It gives the following relations:

$$s(\omega) = \mathcal{F}\{k(r)\} \quad = \int_{-\infty}^{\infty} k(r) e^{-i\omega r} \,\mathrm{d}r \,, \tag{6}$$

$$k(r) = \mathcal{F}^{-1}\{s(\omega)\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} s(\omega) e^{i\omega r} \,\mathrm{d}\omega \,. \tag{7}$$

Since kernels are symmetric real functions, the spectrum of the process is also a symmetric real function. The spectra corresponding to the Matérn covariances are

$$s_{\frac{1}{2}}(\omega) = 2\sigma^2 \lambda (\lambda^2 + \omega^2)^{-1}, \qquad \lambda = \ell^{-1}, \tag{8}$$

$$s_{\frac{3}{2}}(\omega) = 4\sigma^2 \lambda^3 (\lambda^2 + \omega^2)^{-2}, \qquad \lambda = \sqrt{3}\ell^{-1}, \tag{9}$$

$$s_{\frac{5}{2}}(\omega) = \tfrac{16}{3}\sigma^2 \lambda^5 (\lambda^2 + \omega^2)^{-3}, \qquad \lambda = \sqrt{5}\ell^{-1} \,. \tag{10}$$

2.3 Model approximations

In order to overcome the cubic computational scaling of Gaussian process methods, many approximations have been proposed. Here we briefly review two types of approximations, those based on the spectrum of the process and others referred to as 'sparse' approximations.

2.3.1 Random Fourier features

Random Fourier Features (RFF) is a method for approximating kernels. The essential element of the RFF approach (Rahimi and Recht, 2008, 2009) is the realization that the Wiener-Khintchin integral (7) can be approximated by a Monte Carlo sum

$$k(r) \approx \tilde{k}(r) = \frac{\sigma^2}{M} \sum_{m=1}^{M} \cos(\omega_m r), \tag{11}$$

where the frequencies ω_m are drawn randomly from the distribution proportional to $s(\omega)$. Note that the imaginary part is zero because $s(\omega r)$ is even and $i \sin(\omega r)$ is odd.

This approximate kernel has a finite basis function expansion as

$$\boldsymbol{\phi}(x) = [\cos(\omega_1 x), \cos(\omega_2 x), \ldots, \cos(\omega_M x), \sin(\omega_1 x), \sin(\omega_2 x), \ldots, \sin(\omega_M x)]^\top, \tag{12}$$

and we can write the approximate Gaussian process as

$$f(x) \sim \mathcal{GP}\left(0, \tilde{k}(x - x')\right) \qquad \text{or} \qquad f(x) \sim \mathcal{GP}\left(0, \frac{\sigma^2}{M} \boldsymbol{\phi}(x)^\top \boldsymbol{\phi}(x')\right) \tag{13}$$

or equivalently as a parametric model, in the form

$$f(x) = \boldsymbol{\phi}(x)^\top \mathbf{w}, \tag{14}$$

$$\mathbf{w} \sim \mathcal{N}\left(\mathbf{0}, \frac{\sigma^2}{M} \mathbf{I}\right). \tag{15}$$

2.3.2 Optimized Fourier features

In order to have a representative sample of the spectrum, the RFF methodology typically requires the number of spectral sample points to be large. In Lázaro-Gredilla et al. (2010), this was addressed by the attractive choice of optimizing the spectral locations along with the hyperparameters (the 'Sparse Spectrum GP'). However, as argued by those authors, this option does not converge to the full GP and can suffer from overfitting to the training data. We confirm this empirically in a simple experiment in Section 6.

Noting that this method was prone to overfitting, Gal and Turner (2015) sought to improve the model by integrating out, rather than optimizing the frequencies. Gal and Turner derived a variational approximation that made use of a tractable integral over the frequency space. The result is an algorithm that suffers less overfitting than the Sparse Spectrum GP, yet remains flexible. We emphasize the difference to the approach in this work: Gal and Turner proposed variational inference in a sparse spectrum model that is derived from a GP model; our work aims to directly approximate the posterior of the true models using a variational representation.

2.3.3 Regular Fourier features

Another way to approximate the integral in (7) is with a regularly spaced set of frequencies ω. We refer to this method as *regular* Fourier features. While the Random Fourier Features approach could be interpreted as a Monte Carlo approximation to the Wiener-Khintchin integral, the regular Fourier features approach can be seen as a quadrature approximation of the same integral. Therefore the regular methods are not *sparse* in spectral sense, but *dense* and deterministic given the domain of interest (Solin and Särkkä, 2014).

The approximation to the original covariance function takes the form of a finite sum

$$k(r) \approx \tilde{k}(r) = \sum_m s(\omega_m) \cos(\omega_m r) \qquad \omega_m = m\Delta, \tag{16}$$

where Δ is a small number which defines the grid spacing. Similarly to Random Fourier features, the resulting approximate Gaussian process can be written as a parametric model:

$$f(x) = \boldsymbol{\phi}(x)^\top \mathbf{w}, \tag{17}$$

$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{S}) \qquad \mathbf{S} = \operatorname{diag}(s(\omega_1), \ldots, s(\omega_M), s(\omega_1), \ldots, s(\omega_M)). \tag{18}$$

Like in the Random Fourier Features approximation (15), regular Fourier features are independent (the matrix $\mathbf{S}$ is diagonal), and if the inputs are spaced on a regular grid, fast Fourier transform (FFT) methods can speed up the computations to $\mathcal{O}(N \log N)$ (see e.g. Paciorek, 2007; Fritz et al., 2009).

2.3.4 Sparse Gaussian process models

In the above we reviewed how spectral approximations result in parametric models. In their review paper, Quiñonero-Candela and Rasmussen (2005) also presented a view of sparse Gaussian processes where the result is an approximate, parametric model.

The terminology 'sparse' originally referred to methods which restricted computation to a subset of the data points (e.g. Csató and Opper, 2002; Williams and Seeger, 2001), until Snelson and Ghahramani (2005) relaxed this assumption with the idea of pseudo inputs. The pseudo-inputs (or 'inducing points') $\mathbf{Z} = [z_m]_{m=1}^M$ lie in the same domain as $\mathbf{X}$, but with $M < N$. The values of the process associated with these points are denoted $\mathbf{u} = [f(z_m)]_{m=1}^M$. The simplest form of sparse Gaussian process model using these variables is the Deterministic Training Conditional (DTC) model, written

$$f(x) = \mathbf{k}_{\mathbf{u}}(x)^\top \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{u}, \tag{19}$$

$$\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{\mathbf{uu}}). \tag{20}$$

This kind of 'projected process' approximation has also been discussed by e.g. Banerjee et al. (2008). The literature contains many parametric models that approximate Gaussian process behaviours; for example Bui and Turner (2014) included tree-structures in the approximation for extra scalability, and Moore and Russell (2015) combined local Gaussian processes with Gaussian random fields.

2.4 Variational Gaussian process approximations

We have seen how Fourier approximations and sparse Gaussian processes can be written as approximate parametric models. The variational approximation to Gaussian processes provides a more elegant, flexible and extensible solution in that the *posterior distribution* of the original model is approximated, rather than the model itself. In existing work, the variational approach has been used alongside ideas from the sparse GP literature: the remit of this work is to combine the variational methodology with Fourier based approximations. We provide a short review of the variational methodology here, for a more detailed discussion see Matthews (2016).

In variational Bayes (see e.g. Blei et al., 2016, for a contemporary review) the idea is to approximate the posterior of a model by selecting the optimal distribution from a fixed family. Optimality is usually defined through the Kullback-Leibler divergence

$$\mathrm{KL}[q(f(x)) \,\|\, p(f(x)\,|\,\mathbf{y})] = \mathbb{E}_{q(f(x))}\left[\log q(f(x)) - \log p(f(x)\,|\,\mathbf{y})\right] . \tag{21}$$

This equation is a slight abuse of notation, since it is not legitimate to write $p(f(x))$. Nonetheless, the intuition is good, and our overview has the same result as given by a more technical derivation (Matthews et al., 2016).

2.4.1 The approximating stochastic process

We are tasked with defining a family of variational distributions q. Similarly to the sparse GP models, we introduce a set of pseudo inputs (or 'inducing points') $\mathbf{Z} = [z_m]_{m=1}^M$, which lie in the same domain as $\mathbf{X}$, but with $M < N$. Note that $\mathbf{Z}$ do not form part of the model, but only the variational approximating distribution $q(f(x))$: we may choose them freely, and we assume here that $\mathbf{X}$ and $\mathbf{Z}$ do not intersect. We collect the values of the function at $\mathbf{Z}$ into a vector $\mathbf{u} = [f(z_m)]_{m=1}^M$, and write the process conditioned on these values as

$$f(x)\,|\,\mathbf{u} \sim \mathcal{GP}\left(\mathbf{k_u}(x)^\top \mathbf{K_{uu}^{-1}}\mathbf{u},\; k(x,x') - \mathbf{k_u}(x)^\top \mathbf{K_{uu}^{-1}}\mathbf{k_u}(x')\right) , \tag{22}$$

echoing the true posterior of the process, equation (4). We emphasize that we have *selected* this particular form of distribution for $q(f(x)\,|\,\mathbf{u})$, and the joint approximation for the posterior is $q(\mathbf{u})q(f(x)\,|\,\mathbf{u})$. Whilst most of the flexibility of the approximation comes from $q(\mathbf{u})$, using this form of conditional GP for $q(f(x)\,|\,\mathbf{u})$ means that the approximation cannot be written as a parametric model, and does not suffer the degenerate behaviour associated with the DTC approximation (20).

We must also choose some approximate posterior distribution $q(\mathbf{u})$, whose exact form will depend on the likelihood $p(\mathbf{y}\,|\,f(x))$ as we shall discuss momentarily.

2.4.2 The ELBO

Variational Bayes proceeds by maximizing the Evidence Lower Bound (ELBO), which indirectly *minimizes* the Kullback-Leibler objective. By expanding the true posterior using Bayes' rule, and substituting this into equation (21), we have

$$\begin{aligned}\mathrm{KL}\left[q(f(x)) \,\|\, p(f(x)\,|\,\mathbf{y})\right] &= -\mathbb{E}_{q(f(x))}\left[\log \frac{p(\mathbf{y}\,|\,f(x))\,p(f(x))}{q(f(x))}\right] + \log p(\mathbf{y}) \\ &\triangleq -\mathrm{ELBO} + \log p(\mathbf{y}) .\end{aligned} \tag{23}$$

To obtain a tractable ELBO for Gaussian processes, we factor both prior and approximate posterior processes into conditional GPs, conditioning on: the inducing input points and values $(\mathbf{Z}, \mathbf{u})$; the data pairs $(\mathbf{X}, \mathbf{f})$; the remainder of the process $f(x)$. The prior distribution on the process $p(f(x))$ can be written $p(\mathbf{u})p(\mathbf{f}\,|\,\mathbf{u})p(f\,|\,\mathbf{f},\mathbf{u})$, with

$$\begin{aligned} p(\mathbf{u}) &= \mathcal{N}(\mathbf{0},\, \mathbf{K_{uu}}) \\ p(\mathbf{f}\,|\,\mathbf{u}) &= \mathcal{N}\left(\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{u},\, \mathbf{K_{ff}} - \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{K_{fu}^\top}\right) \\ p(f(x)\,|\,\mathbf{f},\mathbf{u}) &= \mathcal{GP}\left(m^\star(x),\, k^\star(x,x')\right) . \end{aligned} \tag{24}$$

where $m^\star(x)$ and $k^\star(x,x')$ are the usual Gaussian process conditional mean and variance, conditioned on both $\mathbf{f}$ and $\mathbf{u}$. The approximate posterior process can be factored similarly:

$$\begin{aligned} q(\mathbf{f}\,|\,\mathbf{u}) &= \mathcal{N}\left(\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{u},\, \mathbf{K_{ff}} - \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{K_{fu}^\top}\right) , \\ q(f(x)\,|\,\mathbf{f},\mathbf{u}) &= \mathcal{GP}\left(m^\star(x),\, k^\star(x,x')\right) . \end{aligned} \tag{25}$$

Noting that the two processes are the same, aside for $p(\mathbf{u})$ and $q(\mathbf{u})$, substituting into equation (23) significantly simplifies the ELBO:

$$\mathrm{ELBO} = \mathbb{E}_{q(\mathbf{u})q(\mathbf{f}\,|\,\mathbf{u})}\left[\log p(\mathbf{y}\,|\,\mathbf{f})\right] - \mathbb{E}_{q(\mathbf{u})}\left[\log \frac{q(\mathbf{u})}{p(\mathbf{u})}\right] . \tag{26}$$

2.4.3 The optimal approximation

It is possible to show that the distribution $\hat{q}(\mathbf{u})$ that maximizes the ELBO is given by

$$\log \hat{q}(\mathbf{u}) = \mathbb{E}_{q(\mathbf{f}\,|\,\mathbf{u})}\left[\log p(\mathbf{y}\,|\,\mathbf{f})\right] + \log p(\mathbf{u}) + \textit{const.} \tag{27}$$

This distribution is intractable for general likelihoods, but can be well approximated using MCMC methods. Parameters of the covariance function can be incorporated into the MCMC scheme in a straightforward manner, see Hensman et al. (2015a) for details.

2.4.4 Gaussian approximations

A computationally cheaper method than sampling $\hat{q}(\mathbf{u})$ is to approximate it using a Gaussian distribution, and to optimize the ELBO with respect to the mean and covariance of the approximating distribution, in order to minimize the KL divergence (Hensman et al., 2015b). If the approximating Gaussian distribution $q(\mathbf{u})$ has mean $\mathbf{m}$ and covariance $\mathbf{\Sigma}$, then the entire approximating process is a GP, with

$$\begin{aligned} q(f(x)) &= \int q(\mathbf{u})q(f(x)\,|\,\mathbf{u})\,\mathrm{d}\mathbf{u} \\ &= \mathcal{GP}\left(\mathbf{k_u}(x)^\top \mathbf{K_{uu}^{-1}}\mathbf{m},\; k(x,x') + \mathbf{k_u}(x)^\top(\mathbf{K_{uu}^{-1}}\mathbf{\Sigma}\mathbf{K_{uu}^{-1}} - \mathbf{K_{uu}^{-1}})\mathbf{k_u}(x')\right) . \end{aligned} \tag{28}$$

The covariance function parameters can be estimated by optimization of the ELBO alongside the variational parameters, which leads to approximate maximum likelihood estimation. This may lead to some bias in the estimation of the parameters (Turner and Sahani, 2011), but the method is reported to work well in practice (Chai, 2012; Hensman et al., 2015b; Lloyd et al., 2015; Dezfouli and Bonilla, 2015).

2.4.5 Gaussian likelihood

For the special case where the data-likelihood $p(y_n \,|\, f(x_n))$ is Gaussian with noise-variance σ_n^2, the optimal distribution for q is given by

$$\hat{q}(\mathbf{u}) = \mathcal{N}\left(\hat{\mathbf{m}},\, \hat{\mathbf{\Sigma}}\right), \quad \text{with} \quad \begin{aligned} \hat{\mathbf{\Sigma}} &= [\mathbf{K}_{\mathbf{uu}}^{-1} + \sigma_n^{-2}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{uf}}\mathbf{K}_{\mathbf{uf}}^{\top}\mathbf{K}_{\mathbf{uu}}^{-1}]^{-1}\,, \\ \hat{\mathbf{m}} &= \sigma_n^{-2}\hat{\mathbf{\Sigma}}\mathbf{K}_{\mathbf{uf}}\mathbf{y}\,, \end{aligned} \tag{29}$$

and the ELBO at this optimal point is

$$\text{ELBO}\,(\hat{q}) = \log \mathcal{N}\left(\mathbf{y} \,|\, 0,\, \mathbf{K}_{\mathbf{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{uf}} + \sigma_n^2\mathbf{I}\right) - \tfrac{1}{2}\sigma_n^{-2}\text{tr}\big(\mathbf{K}_{\mathbf{ff}} - \mathbf{K}_{\mathbf{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{uf}}\big)\,. \tag{30}$$

This expression might give the misleading impression that this approximation resembles the DTC method (described by Quiñonero-Candela and Rasmussen, 2005) if one interprets the matrix $\mathbf{K}_{\mathbf{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{uf}}$ as an approximate prior. However, prediction in the variational approximation does not suffer the same degenerate behaviour as DTC (Titsias, 2009; Hensman et al., 2013).

2.4.6 Prediction

Prediction in the variational approximation to Gaussian processes differs from the view given in Quiñonero-Candela and Rasmussen (2005), and it is here that the elegance of the method is apparent. It is not necessary to make any additional approximations at predict-time, since the whole process has already been approximated. For Gaussian posteriors (and Gaussian approximate posteriors), one simply evaluates the GP (equation (28)). For non Gaussian approximations to $q(\mathbf{u})$, one must average the conditional equation (22) under $q(\mathbf{u})$.

2.5 Linking conditioning and decomposition

The sparse-variational methodology that we have presented involved factoring the Gaussian process as $p(f(x)) = p(f(x) \,|\, \mathbf{u})p(\mathbf{u})$. A related and useful concept is decomposition of the process: we will make use of such a decomposition in sebsequent sections to derive Variational Fourier Features.

Let $f(x) \sim \mathcal{GP}\big(0, k(x, x')\big)$. We can decompose f as the sum of two independent Gaussian processes:

$$g(x) \sim \mathcal{GP}\big(0,\, \mathbf{k}_{\mathbf{u}}(x)^{\top}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}}(x')\big)\,, \tag{31}$$

$$h(x) \sim \mathcal{GP}\big(0,\, k(x, x') - \mathbf{k}_{\mathbf{u}}(x)^{\top}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}}(x')\big)\,, \tag{32}$$

$$f(x) = g(x) + h(x). \tag{33}$$

It is clear from the additivity of the covariance functions that this sum recovers the correct covariance for $f(x)$. Note that the covariance of $h(x)$ is the same as for $p(f(x) \,|\, \mathbf{u})$. We say that $g(x)$ is the projection of $f(x)$ onto $\mathbf{u}$, and $h(x)$ is the orthogonal complement of $g(x)$.

Figure 1 shows the decomposition of a Gaussian process using inducing points and also with the Fourier features that we shall derive in this document.

In the variational approximation, the posterior is controlled through the variables $\mathbf{u}$, and the conditional $p(f(x) \,|\, \mathbf{u})$ remains the same in the approximate posterior as in the

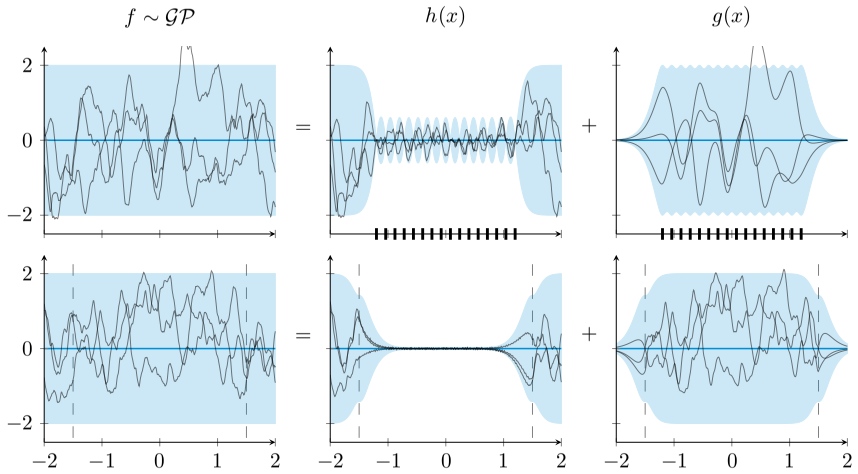


Figure 1: Decomposition of a Gaussian process with a Matérn-$\frac{3}{2}$ covariance function ($\sigma = 1, \ell = 0.2$) using inducing points (top) and Variational Fourier Features (bottom). The shaded region represents two standard deviations, and samples from the processes are shown as black lines. In the top row, the positions of the 16 inducing points are marked in black; on the bottom row 31 frequencies (hence 63 inducing variables) are used and vertical lines denote the boundary of the projection $[a, b]$. Note the variance of $h(x)$ is zero wherever there is an inducing point (top). In the bottom row, the variance is close to zero for a continuous stretch (where the data should live); in this region $h(x)$ contains only high-frequency functions that we have failed to capture with the Fourier features. The variance increases towards the boundaries: since all Fourier basis functions have the same value at a and b, they cannot fit functions with different values in a and b and this is left as a residual. Instead of adding more basis functions, such as a linear one, to account for this, we advocate for choosing an interval $[a, b]$ larger than the data.

prior. In terms of the projection, we see that the process $g(x)$ is completely controlled by the variables $\mathbf{u}$ and the process $h(x)$ is completely independent of $\mathbf{u}$. This is borne out by the independence relations of the processes, formally:

$$g(x) \,|\, \mathbf{u} = \mathcal{GP}\left(\mathbf{k_u}(x)^\top \mathbf{K_{uu}}^{-1}\mathbf{u},\, 0\right), \tag{34}$$

$$h(x) \,|\, \mathbf{u} = h(x)\,. \tag{35}$$

The assumption of the sparse variational framework is that the posterior process can be captured by the random variables $\mathbf{u}$, or equivalently by the process $g(x)$. The power of the framework lies in the fact that the ELBO can be used to evaluate whether the projection is sufficient, and that the orthogonal part $h(x)$ is not discarded but taken into account in the ELBO computation and incorporated fully at predict time. The ELBO encapsulates the degree to which the posterior can be captured by $g(x)$ (or equivalently $\mathbf{u}$).

2.6 Inter-domain approaches

The most common method for efficient approximation of Gaussian processes is based on the idea of inducing variables, denoted $u_m = f(z_m)$ in the above. A powerful idea is to generalize this, allowing a different decomposition of the process by constructing linear combinations of the process values as projections, instead of simple evaluations.

Previous work that has considered inter-domain approaches has suggested that the corresponding features have better ability to represent complicated functions (Figueiras-Vidal and Lázaro-Gredilla, 2009) than the inducing points approach, and have promise for applications in multiple-output settings (Alvarez and Lawrence, 2009).

In inter-domain Gaussian process approximations, the idea is to change

$$u_m = f(z_m) \tag{36}$$

to a projection

$$u_m = \int f(x)\,\mathrm{d}\mu_m(x)\,, \tag{37}$$

so that the variable u_m is more informative about the process, and in turn the projection onto $\mathbf{u}$ is able to capture more information about the posterior.

The utility of the variable u_m depends mostly on its covariance with the remainder of the process, which is encapsulated in the vector-valued function $\mathbf{k_u}(x)$. This vector plays a similar role in the variational method to the feature vector $\boldsymbol{\phi}(x)$ in the Fourier-based approximate models of Sections 2.3.1 and 2.3.2. The remit of this work is to construct inducing variables $\mathbf{u}$ such that $\mathbf{k_u}(x)$ contains sinusoidal functions as is $\boldsymbol{\phi}(x)$, which we will do by using Fourier projections of the process.

The central challenge is that the Fourier transform of the process does not make a valid inducing variable because its variance would be infinite. To make valid and useful inducing variables, Matthews et al. (2016) have shown that the inducing variables u_m must be "deterministic, conditioned on the whole latent function". In the following section, we construct valid inducing variables and examine their properties.

3. Variational Fourier Features

Our goal is to combine the variational sparse GP idea with Fourier features, so that the approximating process in (22) contains a mean function which is built from sinusoids, like the $\boldsymbol{\phi}(x)$ features in the Fourier-features approximate model (15).

This is more tricky than it first appears. An initial idea would be to define u_m as the Fourier transform of the process, so that $\mathbf{k_u}(x) = \mathrm{cov}(\mathbf{u}, x) \propto \boldsymbol{\phi}(x)$. The question is, what would the $\mathbf{K_{uu}}$ matrix be? One might hope for a diagonal matrix related to the spectral content of the kernel $s(\omega)$, but as we discuss in Section 3.1, the Fourier transform of a stationary GP has diverging variance, i.e. this matrix is not defined.

To proceed, we need to make two alterations to the initial idea, which we outline in Sections 3.2 and 3.3. First we must window the domain, in order to obtain variables with finite variance (Section 3.2). Second, since these variables do not have an elegant form, we switch from an L^2 integral to one under the RKHS norm (Section 3.3). Subsequently, we show that our approach has covariance structures that make for efficient computations (Section 3.4), and conclude this section by explaining the behaviour of our approximation outside the windowing box (Section 3.5).

3.1 The trouble with Fourier features

In the variational sparse GP (Section 2.4) we made use of the random variables $u_m = f(z_m)$. To obtain a Fourier-based approximation, we might consider the inter-domain variables $u_m = \int_{-\infty}^{\infty} f(t)e^{-i\omega_m t}\,\mathrm{d}t$. It is straightforward to show that such variables have zero mean and infinite variance, and so will not be useful in the variational framework; nonetheless the properties of the variables are interesting as we now demonstrate.

In order to use u_m as an inducing variable, we would require both $\mathrm{cov}(u_m, f(x))$ and $\mathrm{cov}(u_m, u_{m'})$ which make up the entries in $\mathbf{k_u}(x)$ and $\mathbf{K_{uu}}$ respectively. If f is drawn from a GP with zero mean and covariance $k(x, x')$, then formally we may write

$$\mathrm{cov}(u_m, f(x)) = \mathbb{E}\left[u_m f(x)\right] = \int_{-\infty}^{\infty} \mathbb{E}[f(t)f(x)]e^{-i\omega_m t}\,\mathrm{d}t = \int_{-\infty}^{\infty} k(t,x)e^{-i\omega_m t}\,\mathrm{d}t\,. \tag{38}$$

To solve this integral, we can first plug in the definition of the kernel function in terms of the spectrum (7), and then change the order of integration, giving

$$\mathrm{cov}(u_m, f(x)) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s(\omega)e^{i\omega(t-x)}\,\mathrm{d}\omega\, e^{-i\omega_m t}\,\mathrm{d}t = s(\omega_m)e^{-i\omega_m x}\,. \tag{39}$$

This initially appears promising: the covariance function is a sinusoid rescaled by the relevant spectrum and the elements of $\mathbf{k_u}(x)$ have the desired form. The corresponding elements of $\mathbf{K_{uu}}$ are given by the (co)variance of u_m. It is here that problems arise. Denoting the complex conjugate of u_m as $\bar{u}_m$,

$$\mathrm{cov}(u_m, u_{m'}) = \mathbb{E}\left[u_m \bar{u}_{m'}\right] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} k(t,t')e^{-i\omega_m t}\,\mathrm{d}t\, e^{i\omega_{m'}t'}\,\mathrm{d}t' = s(\omega_m)\delta(\omega_m - \omega_{m'})\,, \tag{40}$$

where $\delta(\cdot)$ is Dirac's delta. This implies that the matrix $\mathbf{K_{uu}}$ is diagonal, but with undefined entries on the diagonal. This result shows that it is simply not possible to use the Fourier transform of the whole GP as a valid inducing variable.

One approach to address this issue might be to take a power-averaged Fourier transform of the form

$$u_m = \lim_{a\to\infty} \frac{1}{a}\int_{-a/2}^{a/2} f(t)e^{-i\omega_m t}\,\mathrm{d}t\,. \tag{41}$$

Unfortunately, this still does not yield useful features, because the variables $\mathbf{u}_m$ are now independent of the function at any particular value. Writing formally:

$$\mathrm{cov}(u_m, f(x)) = \mathbb{E}[u_m f(x)] = \lim_{a\to\infty} \frac{1}{a}\int_{-a/2}^{a/2} k(x,t)e^{-i\omega_m r}\,\mathrm{d}t = 0\,. \tag{42}$$

As recognised by Figueiras-Vidal and Lázaro-Gredilla (2009), the only way to obtain valid features is to specify an input density (cf. Williams and Seeger, 2000). We proceed by considering a uniform input density on the interval $[a, b]$.

3.2 L_2 Fourier features on $[a, b]$

The reason that the Fourier transform of a GP behaves so strangely is that it must explain the behaviour over the whole real line. One might interpret the Fourier transform at a particular frequency as a sum of all the random variables in the GP (i.e. integrating over the real line) multiplied by a sinusoidal weight. The result is a Gaussian random variable, but unless the weight decays to zero at infinity, the result has infinite variance.

Our solution to this diverging integral is to window the integral. Figueiras-Vidal and Lázaro-Gredilla (2009) also use a window to ensure convergence of the integral, but their choice of a Gaussian windowing function means that the result is tractable only for Gaussian (squared exponential) covariance functions, and does not admit a diagonal form for $\mathbf{K_{uu}}$.

We will apply a square window, effectively changing the integration limits to a and b:

$$u_m = \int_a^b f(t)e^{-i\omega_m(t-a)}\,\mathrm{d}t\,. \tag{43}$$

In addition, we will assume that the frequency ω_m is harmonic on the interval $[a, b]$; that is

$$\omega_m = \frac{2\pi m}{b-a}\,. \tag{44}$$

For $x \in [a, b]$, the covariance between such inducing variables and the GP at x is

$$\mathrm{cov}(u_m, f(x)) = \mathbb{E}\left[u_m f(x)\right] = \int_a^b \mathbb{E}[f(t)f(x)]e^{-i\omega_m(t-a)}\,\mathrm{d}t = \int_a^b k(t,x)e^{-i\omega_m(t-a)}\,\mathrm{d}t\,. \tag{45}$$

These integrals are tractable for Matérn kernels. As detailed in Appendix A, we obtain the following results for Matérn-$\frac{1}{2}$ kernels:

$$\mathrm{cov}(u_m, f(x)) = s_{1/2}(\omega_m)e^{i\omega_m(x-a)} + s_{1/2}(\omega_m)\frac{1}{2\lambda}\left(\lambda[-e^{a-x} - e^{x-b}] + i\omega_m[e^{a-x} - e^{x-b}]\right)\,. \tag{46}$$

Similarly, the covariance between two inducing variables also has a closed form expression. A complete derivation is given in Appendix A for the Matérn-$\frac{1}{2}$ case. These inter-domain inducing variables have two desirable properties: they have finite variance and their associated covariance matrix $\mathbf{K_{uu}}$ can be written as a diagonal matrix plus some rank one matrices (two in the Matérn-$\frac{1}{2}$ case).

Figure 2 illustrates entries of the feature vector $\mathbf{k_u}(x) = \mathrm{cov}(u_m, f(x))$, and compares them to the RKHS features that we shall derive in the next section. We see from the Figure that the covariance function $\mathrm{cov}(u_m, f(x))$ is almost sinusoidal in x for a region sufficiently far from the boundaries a and b. The 'edge effects' depend on the involved frequency as well as the lengthscale parameter of the kernel.

We have constructed inducing variables that are valid (in the sense that they have finite variance), but the result is somewhat inelegant: the expressions for $\mathrm{cov}(u_m, f(x)$ and $\mathrm{cov}(u_m, u_{m'})$ are long for the Matérn-$\frac{1}{2}$ kernel and become trickier (but still tractable) for higher orders. We abandon them at this stage in preference of the RKHS-based inducing variables that we derive in the next section. These have preferable edge effects and an elegant expressions for the required covariances.

3.3 RKHS Fourier features on $[a, b]$

In the previous section we saw that choosing inducing variables of the form $u_i = \langle f, \cos(\omega(xa)\rangle_{L^2}$ or $u_i = \langle f, \sin(\omega(xa)\rangle_{L^2}$ resulted in features that are very distinct from cosine and sine functions. In this section, we replace the L^2 inner product by the RKHS inner product $u_i = \langle f, \cos(\omega(xa)\rangle_{\mathcal{H}}$ in order to guarantee that the features are exactly the sine and cosine functions. As we will discuss later, a direct asset of this approach is that the simple expression of the features makes the computation of the covariances between inducing variables much easier.

We start with a truncated Fourier basis defined similarly to (12)

$$\boldsymbol{\phi}(x) = [1, \cos(\omega_1(x-a)), \ldots, \cos(\omega_M(x-a)), \sin(\omega_1(x-a)), \ldots, \sin(\omega_M(x-a))]^\top\,, \tag{47}$$

where we include the constant basis function, $\phi_0(x) = 1$, accounting for $\omega_m = 0$, and define $\omega_m = \frac{2\pi m}{b-a}$ as previously.

A key RKHS result (see Berlinet and Thomas-Agnan, 2004, Theorem 11), is that if $\mathcal{F} = \mathrm{span}(\boldsymbol{\phi})$ is a subspace of a RKHS $\mathcal{H}$, then $\mathcal{F}$ has the kernel

$$k_{\mathcal{F}}(x, x') = \boldsymbol{\phi}(x)^\top \mathbf{K}_{\phi\phi}^{-1}\boldsymbol{\phi}(x')\,, \tag{48}$$

where $\mathbf{K}_{\phi\phi}[m, m'] = \langle \phi_m, \phi_{m'}\rangle_{\mathcal{H}}$ is the Gram matrix of $\boldsymbol{\phi}$ in $\mathcal{H}$. By this, we mean that for any function $f \in \mathcal{F}$, $\langle f, k_{\mathcal{F}}(\cdot, x)\rangle_{\mathcal{H}} = f(x)$. Furthermore, the coordinate of the projection of a function $h \in \mathcal{H}$ onto ϕ_m is defined as

$$\mathcal{P}_{\phi_m}(h) = \langle h, \phi_m\rangle_{\mathcal{H}}\,. \tag{49}$$

Since $\mathcal{F} \subset \mathcal{H}$, $k(x, x') - k_{\mathcal{F}}(x, x')$ is a positive definite function corresponding to the kernel of $\mathcal{F}^\perp$, and we can decompose the GP in a similar way to (33) which gives

$$g(x) \sim \mathcal{GP}(0,\, \boldsymbol{\phi}(x)^\top \mathbf{K}_{\phi\phi}^{-1}\boldsymbol{\phi}(x'))\,, \tag{50}$$

$$h(x) \sim \mathcal{GP}(0,\, k(x, x') - \boldsymbol{\phi}(x)^\top \mathbf{K}_{\phi\phi}^{-1}\boldsymbol{\phi}(x'))\,, \tag{51}$$

$$f(x) = g(x) + h(x)\,. \tag{52}$$

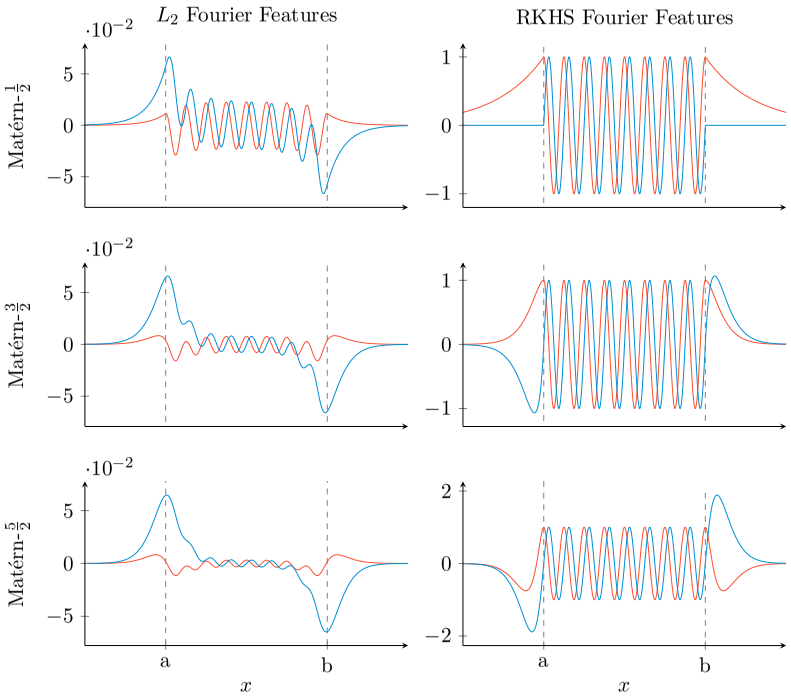


Figure 2: The covariance $\mathrm{cov}(u_m, f(x))$ as a function of x for the Matérn-$\frac{1}{2}$ (top), Matérn-$\frac{3}{2}$ (middle) and Matérn-$\frac{5}{2}$ (bottom) kernels, using L_2 Fourier features (left column) and RKHS Fourier features (right column). The red and blue lines represent the real and imaginary parts of the covariance. The boundaries of the projection $[a, b]$ are marked by vertical dashed lines. The L_2 features are wiggly with overwhelming edge effects, whereas the RKHS features are exactly sinusoids for $x \in [a, b]$. Beyond the boundaries, the covariance reverts to zero in a way which depends on the smoothness of the kernel and the lengthscale parameter. We have used $\omega_m = 16\pi/(b-a)$, and a lengthscale of $\ell = (b-a)/10$.

For this to be valid we need two things: $\mathcal{F}$ must be included in $\mathcal{H}$ and the matrix $\mathbf{K}_{\phi\phi}$ must be invertible. The first is not true in general (Matérn RKHS over $\mathbb{R}$ or the RBF and Brownian RKHS on $[0, 1]$ are counter examples) but has been proved to be true for Matérn kernels over $[a, b]$ (Durrande et al., 2016). The second is a direct conclusion of the linear independence between the elements of $\boldsymbol{\phi}$. Furthermore, Durrande et al. (2016) also detail the closed-form expressions of the inner products for the Matérn-$\frac{1}{2}$, Matérn-$\frac{3}{2}$ and Matérn-$\frac{5}{2}$ RKHSs. These expressions, which are repeated in Appendix B, are of particular interest here since they allow us to compute the entries of the $\mathbf{K}_{\phi\phi}$ matrix. Although we refer to the above reference for the constructive proof of these expressions, it is easy to check that the reproducing property is satisfied. For example, the expression given in the appendix for a Matérn-$\frac{1}{2}$ RKHS over $[a, b]$ is

$$\langle g, h\rangle_{\mathcal{H}_{1/2}} = \frac{1}{2\lambda\sigma^2}\int_a^b \big(\lambda g(t) + g'(t)\big)\big(\lambda h(t) + h'(t)\big)\,\mathrm{d}t + \frac{1}{\sigma^2} g(a)\,h(a)\,, \tag{53}$$

so we can write for any function g that is continuous and (weakly) differentiable

$$\left\langle g, k_{\frac{1}{2}}(x,\cdot)\right\rangle_{\mathcal{H}_{1/2}} = \frac{1}{2\lambda\sigma^2}\left(\int_a^x \big(\lambda g(t) + g'(t)\big)\,2\lambda\sigma^2 e^{\lambda(t-x)}\,\mathrm{d}t + \int_x^b 0\,\mathrm{d}t\right) + g(a)e^{\lambda(a-x)} \tag{54}$$

$$= \int_a^x \lambda g(t) e^{\lambda(t-x)}\,\mathrm{d}t + \Big[g(t)e^{\lambda(t-x)}\Big]_a^x - \int_a^x g(t)\lambda e^{\lambda(t-x)}\,\mathrm{d}t + g(a)e^{\lambda(a-x)} \tag{55}$$

$$= g(x) - g(a)e^{\lambda(a-x)} + g(a)e^{\lambda(a-x)} \tag{56}$$

$$= g(x)\,. \tag{57}$$

Another property of this inner product is that the Gram matrix $\mathbf{K}_{\phi\phi}$ has a particular structure which allows to reduce drastically the computational burden of computing and inverting $\mathbf{K}_{\phi\phi}$. This will be addressed in Section 3.4.

3.3.1 The corresponding random variables

To interpret these results from a Gaussian process point of view, we need to find the inducing variables $\mathbf{u}$ that would lead to the exact same decomposition.

We would like to use the RKHS inner product between the sinusoids and the GP sample path in place of the L_2 inner product (43), but since GP samples do not belong the RKHS (Driscoll, 1973), it is a priori not possible to apply $\mathcal{P}_{\phi_m}$ to f and define $u_m = \langle \phi_m, f\rangle_{\mathcal{H}}$. For example, the Matérn-$\frac{1}{2}$ inner product involves derivatives whereas the Matérn-$\frac{1}{2}$ samples are not differentiable anywhere. However, using the fact that the functions from $\boldsymbol{\phi}$ are very regular, it is possible to extend the operators $\mathcal{P}_{\phi_m} : h \mapsto \langle \phi_m, h\rangle_{\mathcal{H}}$ to square integrable functions using integration by parts. For the Matérn-$\frac{1}{2}$ kernel, integrating (53) results in

$$\mathcal{P}_{\phi_m}(h) = \frac{1}{2\lambda\sigma^2}\left(\int_a^b h(\lambda^2\phi_m - \phi_m'')\,\mathrm{d}t + h(b)\big(\lambda\phi_m(b) + \phi_m'(b)\big) + h(a)\big(\lambda\phi_m(a) - \phi_m'(a)\big)\right). \tag{58}$$

Note that the function h does not need to be differentiated, and so it is possible to apply this functional to a sample from the GP. Similar results apply for the Matérn-$\frac{3}{2}$ and Matérn-$\frac{5}{2}$ kernels. It is now possible to apply these operators to the Gaussian process in order to construct the inducing variables,

$$u_m = \mathcal{P}_{\phi_m}(f)\,. \tag{59}$$

The covariance of the inducing variables with the function values is given by

$$\mathrm{cov}(u_m, f(x)) = \mathbb{E}[u_m f(x)] = \mathbb{E}[\mathcal{P}_{\phi_m}(f) f(x)] = \mathcal{P}_{\phi_m}\left(k(x,\cdot)\right) = \phi_m(x) \tag{60}$$

which is valid for $a \leq x \leq b$. This means that $\mathbf{k_u}(x) = \boldsymbol{\phi}(x)$ when $x \in [a,b]$. Similarly the covariance of the features is given by

$$\mathrm{cov}(u_m, u_{m'}) = \mathbb{E}[u_m u_{m'}] = \mathbb{E}[\mathcal{P}_{\phi_m}(f)\mathcal{P}_{\phi_{m'}}(f)] = \mathcal{P}_{\phi_m}(\phi_{m'}) = \langle \phi_m, \phi_{m'} \rangle_{\mathcal{H}}\,, \tag{61}$$

and so we have $\mathbf{K_{uu}} = \mathbf{K}_{\phi\phi}$.

3.4 Covariance structures for RKHS Fourier features

To compute the covariance matrix $\mathbf{K_{uu}}$, we return again to the closed-form expressions for the inner product provided by Durrande et al. (2016), substituting the basis functions ϕ_m and $\phi_{m'}$ appropriately. The solutions are tractable, and we detail in the appendix the expressions of $\mathbf{K_{uu}}$ for the first three half-integer Matérn kernels. One striking property of the resulting $\mathbf{K_{uu}}$ is that they can be written as a diagonal matrix plus a few rank one matrices. For example in the Matérn-$\frac{1}{2}$ case we find that $\mathbf{K_{uu}}$ is equal to a diagonal matrix plus a rank one matrix,

$$\mathbf{K_{uu}} = \mathrm{diag}(\boldsymbol{\alpha}) + \boldsymbol{\beta}\boldsymbol{\beta}^\top\,, \tag{62}$$

with

$$\boldsymbol{\alpha} = \frac{b-a}{2}\left[2s(0)^{-1}, s(\omega_1)^{-1}, \ldots, s(\omega_M)^{-1}, s(\omega_1)^{-1}, \ldots, s(\omega_M)^{-1}\right]^\top, \tag{63}$$

$$\boldsymbol{\beta} = \left[\sigma^{-1}\ , \sigma^{-1}\ , \ldots, \sigma^{-1}\ , 0\ , \ldots, 0\ \right]^\top. \tag{64}$$

As shown in the appendix, $\mathbf{K_{uu}}$ still has a similar structure for higher order Matérn kernels: in the Matérn-$\frac{3}{2}$ case, it is the sum of a diagonal matrix and two rank one matrices and in the Matérn-$\frac{5}{2}$ case, it is the sum of a diagonal matrix and three rank one matrices.

Since $\mathbf{K_{uu}}$ has a low-rank plus diagonal form, one of the usual computational bottlenecks in the variational framework, solving $\mathbf{K_{uu}^{-1}K_{uf}}$, can be done in $\mathcal{O}(NM)$ operations, rather than the standard $\mathcal{O}(NM^2)$, by a straightforward application of the Woodbury matrix identity.

This low-rank plus diagonal structure was not published previously in Durrande et al. (2016); we believe that we are the first to make use of this result for computational efficiency and we expect it to provide similar superior computational benefits as the pure diagonal structure in Solin and Särkkä (2014).

$\phi_m(x)$, $x \in [a,b]$	Matérn-$\frac{1}{2}$	Matérn-$\frac{3}{2}$	Matérn-$\frac{5}{2}$
$\cos(\omega_m(x-a))$	$e^{-\lambda r}$	$(1+\lambda r)e^{-\lambda r}$	$(1+\lambda r+\frac{1}{2}(\lambda^2-\omega_m^2)r^2)e^{-\lambda r}$
$\sin(\omega_m(x-a))$	0	$sr\omega_m e^{-\lambda r}$	$sr\omega_m(1+\lambda r)e^{-\lambda r}$

Table 1: The covariance $\mathrm{cov}(u_m, f(x))$ for x outside the interval $[a,b]$. Here, we define r as the absolute distance to the closest edge (a or b), and s to be 1 for $x < a$ and -1 for $x > b$.

3.5 Predicting outside the interval $[a,b]$

In the preceding sections, we have defined our inducing variables and how they covary with the function; we have assumed throughout that we wish to examine a point on the latent function in the pre-defined interval $[a,b]$. This does not pose a problem at training time, since we can easily define $[a,b]$ to contain all the data. For prediction we may need to predict outside this interval. For a point on the function $f(x)$ outside the interval $[a,b]$, the covariance with the inducing variables is still well defined, though the form is slightly more complicated than the simple sinusoids inside the interval.

To obtain the expressions for the covariance beyond $[a,b]$, we apply the closed-form expression for the covariance (58), this time for $\mathrm{cov}(u_m, f(x)) = \mathrm{cov}(\mathcal{P}_{\phi_m}(f), f(x))$ with $x > b$ and $x < a$. After some fairly elementary simplifications, the results are shown in Table 1, and illustrated in Figure 2. The result is that the covariance function $\mathrm{cov}(u_m, f(x))$ returns to zero beyond the boundary, with smoothness that depends on the order of the kernel.

4. Extending the applicable kernel family

In the above we have focused on the Matérn family of kernels in one dimension. In this section we will expand the approach to higher-dimensional inputs using sums and products of kernels.

4.1 Additive kernels

A straightforward way to produce a Gaussian process prior on multiple inputs (say D) is to use a sum of independent Gaussian processes, one for each input:

$$f(\mathbf{x}) = \sum_{d=1}^{D} f_d(x_d)\,, \qquad f_d \sim \mathcal{GP}\left(0,\, k_d(x_d, x_d')\right)\,, \tag{65}$$

where x_d is the dth element in $\mathbf{x}$ and $k_d(\cdot,\cdot)$ is a kernel defined on a scalar input space. This construction leads to a Gaussian process with an additive kernel, that is we can write

$$f(\mathbf{x}) \sim \mathcal{GP}\left(0,\, \sum_{d=1}^{D} k_d(x_d, x_d')\right)\,. \tag{66}$$

Additive kernel structures have been explored by Durrande et al. (2012) and Duvenaud et al. (2011), who have shown that these kernels are well suited to high-dimensional problems. To use our Variational Fourier Features approach with an additive kernel, we assume that each function $f_d(x_d)$ has a Matérn covariance function, and decompose each of these GPs. We construct a matrix of features, with elements

$$u_{m,d} = \mathcal{P}_{\phi_m}(f_d)\,. \tag{67}$$

The result is that we have DM features. By construction, we see that features from different GPs are independent, $\mathrm{cov}(u_{m,d}, u_{m,d'}) = 0$, and that the covariance between features for the same dimension follows the construction for a single dimension. It is straightforward to exploit these independence structures for computational scalability during inference.

4.2 Separable kernels

A kernel is said to be separable if it can be written as a product of kernels with no shared inputs. This means that for any D-dimensional input $\mathbf{x}$, we can write

$$k(\mathbf{x}, \mathbf{x}') = \prod_{d=1}^{D} k_d(x_d, x_d')\,. \tag{68}$$

We can construct Variational Fourier Features for such a product kernel if each sub-kernel is in the Matérn family.

This formulation of separability has been extensively used in speeding up GP inference (see, e.g., Saatçi, 2012; Stegle et al., 2011) by writing the covariance matrix as a Kronecker product of the individual covariance matrices. These speed-ups usually require the data to be aligned with a regular or rectilinear grids (see, e.g., Solin et al., 2016), though some approaches exist to extend to observations beyond the grid (Wilson et al., 2014; Nickson et al., 2015), requiring additional approximations.

In contrast, our approach of decomposing the kernel naturally leads to a Kronecker structure, even for irregularly spaced data. Let $f(x) \sim \mathcal{GP}\left(0, \prod_d k_d(x_d, x_d')\right)$, and define a vector of features as the Kronecker product of features over each dimension,

$$\boldsymbol{\phi}(\mathbf{x}) = \bigotimes_{d=1}^{D} \left[\phi_1(x_d), \ldots, \phi_M(x_d)\right]^\top\,, \tag{69}$$

so that each of the M^D elements of $\boldsymbol{\phi}(\mathbf{x})$ is a product of one-dimensional functions $\prod_d \phi_i^{(d)}(x_d)$. We define a hyper-rectangular boundary given by $\prod_d [a_d, b_d]$, and define the inducing variables $\mathbf{u}$ using a projection similar to the one-dimensional case

$$u_m = \mathcal{P}_{\phi_m}(f)\,, \tag{70}$$

where we use the norm of the Hilbert space associated with the product kernel. The covariance between an inducing point u_m and the function is then given by

$$\mathrm{cov}(u_m, f(\mathbf{x})) = \mathbb{E}[\mathcal{P}_{\phi_m}(f) f(x)] = \mathcal{P}_{\phi_m}(k(\mathbf{x}, \cdot)) = \prod_{d=1}^{D} \phi_m^{(d)}(x_d) = \boldsymbol{\phi}_m(\mathbf{x})\,. \tag{71}$$

Extending the covariance function beyond the boundary follows similarly to the one-dimensional case above. The covariance between inducing variables is given by

$$\mathrm{cov}(u_m, u_m') = \prod_d \left\langle \phi_m^{(d)}, \phi_{m'}^{(d)} \right\rangle_{\mathcal{H}_d}\,. \tag{72}$$

This means that the covariance matrix $\mathbf{K_{uu}}$ has a Kronecker structure, $\mathbf{K_{uu}} = \bigotimes_{d=1}^{D} \mathbf{K}_{\mathbf{uu}}^{(d)}$, where each sub-matrix $\mathbf{K}_{\mathbf{uu}}^{(d)}$ has the same structure as for the one-dimensional case (62).

5. Implementation details and computational complexity

In this section we give some details of our implementation of the methods proposed, and consider the theoretical computational complexity of each. To make use of the RKHS Fourier features that we have derived, the expressions given for $\mathbf{K_{uu}}$ and $\mathbf{k_u}(x)$ can simply be plugged in to the variational framework described in Section 2.4.

5.1 Implementation

All of the methods in this paper and all of the code to replicate the experiments in Section 6 are available at `http://github.com/jameshensman/VFF`. We made use of TensorFlow (Abadi et al., 2015) and GPflow (Matthews et al., 2017) to construct model classes in Python. Some simple matrix classes in Python assisted with computing efficient solutions to low-rank matrix problems (e.g. solving $\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{m}$). Using TensorFlow's automatic differentiation made application of gradient based optimization straightforward.

5.2 The one-dimensional case

Consider first problems with only one input dimension. The sparse variational framework discussed in Section 2.4 gives three methods: if the likelihood is Gaussian, then it is possible to compute the optimal posterior in closed form (as per Titsias, 2009), if the likelihood is not Gaussian we can approximate the posterior with a Gaussian or use MCMC on the optimal distribution.

For the Gaussian case, we see that the computation of the optimal posterior (equation (29)) is dominated by the matrix multiplication $\mathbf{K_{uf}}\mathbf{K_{fu}}$, which costs $\mathcal{O}(NM^2)$ operations. In our case, since $\mathbf{K_{fu}}$ does not depend on the kernel parameters, this multiplication only needs to be computed once before optimizing (or sampling) the covariance function parameters. To compute the posterior mean and covariance, we must invert an $M \times M$ matrix, so the cost per iteration is $\mathcal{O}(M^3)$, after the initial cost of $\mathcal{O}(NM^2)$.

For the non-Gaussian case with a Gaussian approximation we must compute the marginals of $q(\mathbf{f})$, in order to evaluate the ELBO (equation (26)) for optimization. We parameterize the Gaussian covariance using a lower-triangular matrix $\mathbf{L}$, so $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^\top$, and the computational cost is dominated by the matrix multiplication $\mathbf{K_{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{L}$. Even though the matrix $\mathbf{K_{uu}}$ has computationally-convenient structure, we cannot avoid a dense matrix multiplication in this case. It may be possible that by assuming some structure for the approximate covariance, computation could be reduced, but we leave investigation of that line for future research.

We note however that the Gaussian approximation does lend itself to stochastic optimization, where the data are randomly sub-sampled (Hensman et al., 2013). In this case the computational complexity is $\mathcal{O}(\tilde{N}M^2)$, where $\tilde{N}$ is the size of the minibatch.

For the MCMC approach, evaluating the unnormalised distribution (27) and its derivative costs $\mathcal{O}(NM)$ per iteration, since again the cost is dominated by the computation of the marginals of $q(\mathbf{f}\,|\,\mathbf{u})$, which requires only the product $\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}$, and the low-rank structure of $\mathbf{K_{uu}}$ enables this to be computed efficiently.

5.3 The additive case

If we consider the kernel to be a sum of kernels over D input dimensions (Section 4.1), then we again have access to the three methods provided by the sparse GP framework. In this case, the number of inducing variables is $2MD+1$, since we have M frequencies per dimension, with sine and cosine terms.

For the method with a Gaussian likelihood, the cost is $\mathcal{O}(D^3M^3)$ per iteration, with an initial cost of $\mathcal{O}(NM^2D^2)$, following the arguments above. When the likelihood is non-Gaussian, with a Gaussian approximation to the posterior, the cost is $\mathcal{O}(NM^2D^2)$. However, if we assume that the posterior factorizes over the dimensions, then the cost can be reduced to $\mathcal{O}(NM^2D)$ (Adam et al., 2016).

For an MCMC approach, the cost is again dominated by the computation of $\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}$, which must be done once for each dimension and so the cost is $\mathcal{O}(NMD)$. We note that these operations can be effectively distributed on multiple machines (in our implementation this can be done by TensorFlow), since $\mathbf{K_{uu}}$ is a block-diagonal matrix.

5.4 The Kronecker case

If we use a separable kernel made from a product of kernels over the D dimensions as discussed in Section 4.2, then the number of inducing variables grows exponentially with the input dimension to $(2M)^D$. This gives the potential for a very detailed representation, with many basis functions, and the Kronecker structure can be exploited for efficiency in some cases. However, this approximation will be unsuitable for large D.

For the case with Gaussian noise, it is not straightforward to avoid computing the inverse of the $M^D \times M^D$ matrix, and so the cost is $\mathcal{O}(M^{3D})$. It may be possible, however, that some efficient methods for solving linear systems might be applicable (e.g. Filippone and Engler, 2015), though we leave that avenue for future work.

For the case with a Gaussian approximation to $q(\mathbf{u})$, we can exploit the Kronecker structure of the model. We note the relationship to the work of Nickson et al. (2015), which used the inducing-point approach, with the points confined to a regular lattice. In their work, the covariance structure of the approximation was constrained to also have a Kronecker structure, which appeared to work well in the cases considered. However, our experiments suggest (Section 6.3) that this approximation may not be sufficient in all cases. Our proposal is to use a sum of two Kronecker structured matrices:

$$q(\mathbf{u}) = \mathcal{N}(\mathbf{m}, \mathbf{S}), \qquad \text{where } \mathbf{S} = \bigotimes_{d=1}^{D} \mathbf{L}_d\mathbf{L}_d^\top + \bigotimes_{d=1}^{D} \mathbf{J}_d\mathbf{J}_d^\top. \tag{73}$$

Noting that we can compute the determinant of such a matrix structure in $\mathcal{O}(DM^3)$ (see e.g. Rakitsch et al., 2013), the computational cost of the method is then dominated by the cost of computing the mean and marginal variances of $q(\mathbf{f})$. For the mean, we first compute $\mathbf{K_{uu}^{-1}}\mathbf{m}$ which costs $\mathcal{O}(M^D)$ (making use of the low-rank structure of $\mathbf{K_{uu}}$), and then $\mathbf{K_{fu}}(\mathbf{K_{uu}^{-1}}\mathbf{m})$, which costs $\mathcal{O}(NM^D)$. In computing the marginal variances of $q(\mathbf{f})$, the additive structure of $\mathbf{S}$ poses no problem, and the Kronecker structures mean that $\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{S}$ can be computed in $\mathcal{O}(NM^{2D})$. Again, it is possible to make use of stochastic optimization to reduce the cost per iteration.

In the MCMC case, the cost is once again dominated by computing the marginals of $q(\mathbf{f}\,|\,\mathbf{u})$, which costs $\mathcal{O}(NM^D)$.

5.5 Centring of variables for MCMC

For an effective MCMC scheme, it is necessary to re-centre the variables $\mathbf{u}$ using a square-root of $\mathbf{K_{uu}}$ (Christensen et al., 2006; Vanhatalo and Vehtari, 2007; Murray and Adams, 2010; Filippone et al., 2013; Hensman et al., 2015a) such that

$$\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{74}$$

$$\mathbf{u} = \mathbf{K_{uu}^{1/2}}\mathbf{v}, \tag{75}$$

which gives $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{K_{uu}})$. For dense $\mathbf{K_{uu}}$ matrices, the square root $\mathbf{K_{uu}^{1/2}}$ is often chosen to be the Cholesky factor, but the structured $\mathbf{K_{uu}}$ described above has a more convenient square root, given by concatenating the two parts into

$$\mathbf{R} = \left[\mathrm{diag}(\boldsymbol{\alpha})^{\frac{1}{2}}, \boldsymbol{\beta}\right] \tag{76}$$

so that $\mathbf{R}\mathbf{R}^\top = \mathrm{diag}(\boldsymbol{\alpha}) + \boldsymbol{\beta}\boldsymbol{\beta}^\top = \mathbf{K_{uu}}$. Since $\mathbf{R}$ has an extra column (or two or three extra columns for Matérn-$\frac{3}{2}$ and Matérn-$\frac{5}{2}$ respectively) we require an extra variable(s) in the vector $\mathbf{v}$.

6. Experiments

This Section provides empirical evidence and examples of how the Variational Fourier Features method works in practice. We first cover some illustrative toy examples, which underline the accuracy of the approximation and the influence of the approximation parameters. After this we address three standard test examples for GP regression (for high- and low-dimensional inputs) and classification, where we show the method provides accurate predictions, and does so extremely fast. Finally, we examine how the approximation approaches the posterior in the MCMC case by examining a simple log Gaussian Cox process data set.

6.1 Illustrative examples

Our Variational Fourier Features (VFF) method has two tuning parameters which affect the approximation: the number of Fourier features M, and the choice of the bounds a and b. In this Section the effects related to tuning these parameters are demonstrated.

6.1.1 Regression

To perform GP regression with Variational Fourier Features, one simply takes the expressions given for standard GP approximations (equations (28) and (29)) and substitutes $\mathbf{K_{uu}}$, $\mathbf{K_{uf}}$ and $\mathbf{k_u}(x)$ appropriately. Figure 3 compares the VFF approximation with Random Fourier Features on a trivial but illuminating regression experiment.

We started by fitting a 'full' GP model (i.e. with inversion of the dense covariance matrix), maximising the likelihood with respect to the kernel's variance parameter, lengthscale parameter and the noise variance. We then used these fitted values to compare variational and Random Fourier Features.

The Random Fourier Features method has some attractive convergence properties (Rahimi and Recht, 2008), as well as being simple to implement. Here we show empirically that the VFF method approaches the true posterior much faster, in terms of the number of basis functions used.

This experiment uses a Matérn-$\frac{1}{2}$ kernel. The corresponding spectral density is proportional to a Cauchy distribution, which has heavy tails. The frequencies in Random Fourier Features are drawn from this heavy-tailed distribution. We see from Figure 3 that this leads to a poor Monte Carlo estimate of equation (7). With 20 random features, the fit of the model is poor, whilst with 20 variational features, the approximation is qualitatively better. With 100 features, the random features approach still cannot replicate the full GP, whilst the variational method is already exact since the ELBO is tight to the likelihood to within numerical precision. Even with 500 features, the RFF method cannot approximate the model well.

This demonstration is exacerbated by the choice of kernel function. For other kernels in the Matérn series, with spectra that behave more like a Gaussian, the problem is less severe. Nonetheless, this demonstrates the power of Variational Fourier Features for these models.

6.1.2 Setting a, b, and M

To use Variational Fourier Features, we must select the projection interval $[a, b]$, as well as the frequencies used. In all our experiments, we use the first M frequencies that are harmonic on the interval, that is $\omega = [\frac{2\pi m}{(b-a)}]_{m=1}^{M}$.

Figure 4 compares the effect of changing the interval size and the number of inducing frequencies, for a simple Gaussian regression problem. We drew data from a Gaussian process with a Matérn-$\frac{3}{2}$ kernel ($\sigma^2 = 1$, $\ell = 0.2$) and added Gaussian noise ($\sigma_n^2 = 0.05$). We used these known parameters for the series of regression fits shown, varying $[a, b]$ and M. The true marginal likelihood was computed to be -15.99.

The top row has the interval set too narrow, within the region of the training data (this is valid but not recommended!). Accordingly, the ELBO (marked in the upper left corner of each plot) is low, no matter how many frequencies are used. In the next row, the boundary is still too close to the data: some edge effects are visible. The low value of the ELBO reflects the low quality oif the approximation.

In the third row, the boundary $[a, b]$ is sufficiently far from the data. No edge effects are visible, but we notice extra wiggles in the approximation when insufficient frequencies

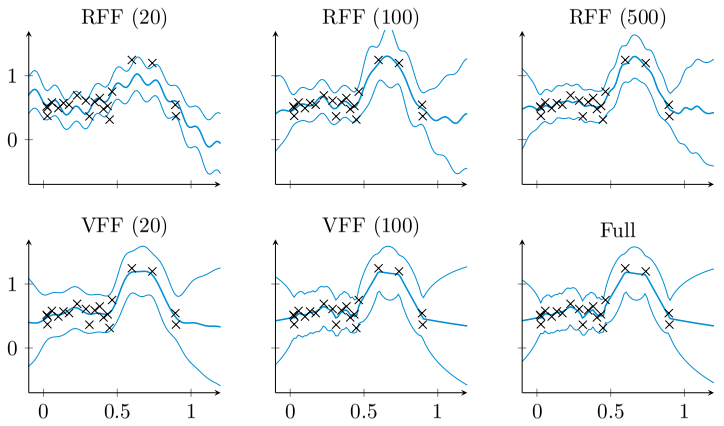


Figure 3: A comparison of our Variational Fourier Features (VFF) with Random Fourier Features (RFF). Top row: Random Fourier Features with 20, 100, and 500 frequencies. Bottom row: Variational Fourier Features with 20 and 100 frequencies, and the full-GP fit. The kernel is a Matérn-$\frac{1}{2}$, with variance, lengthscale and noise variance set by maximum likelihood according to the full GP. The bounds of the problem are set to $[-1, 2]$ for the VFF.

are used. For sufficient frequencies, and this sensible setting of the interval, the ELBO is close to the marginal likelihood.

The fourth row of Figure 4 shows the effect of choosing an interval that is too large. Because we have set the frequencies to be dependent on $(b-a)$, making this quantity large makes the frequencies too low. Sinusoidal behaviour is visible with $M = 8$ or $M = 16$, although a sensible solution is recovered once $M = 32$. Again, the ELBO reflects the quality of the approximation.

The purpose of this experiment was to illustrate how the properties of the approximation parameters (a, b, and M) affect the approximation. We have emphasized how the ELBO can be examined to ensure that two key considerations are met. First, that a and b are sufficiently far from the edges of the data. Second, that a sufficient number of frequencies are used. We emphasize again the monotonic nature of the approximation inherited from the variational framework: adding more features (increasing M) necessarily improves the approximation in the KL sense.

6.1.3 Comparison with Sparse GP for various input dimensions

The computational complexity of the VFF approach increases with the dimension of the input space. For additive kernels, we have shown that the complexity increases linearly with input dimension, and for models with a product kernel, the computation increases

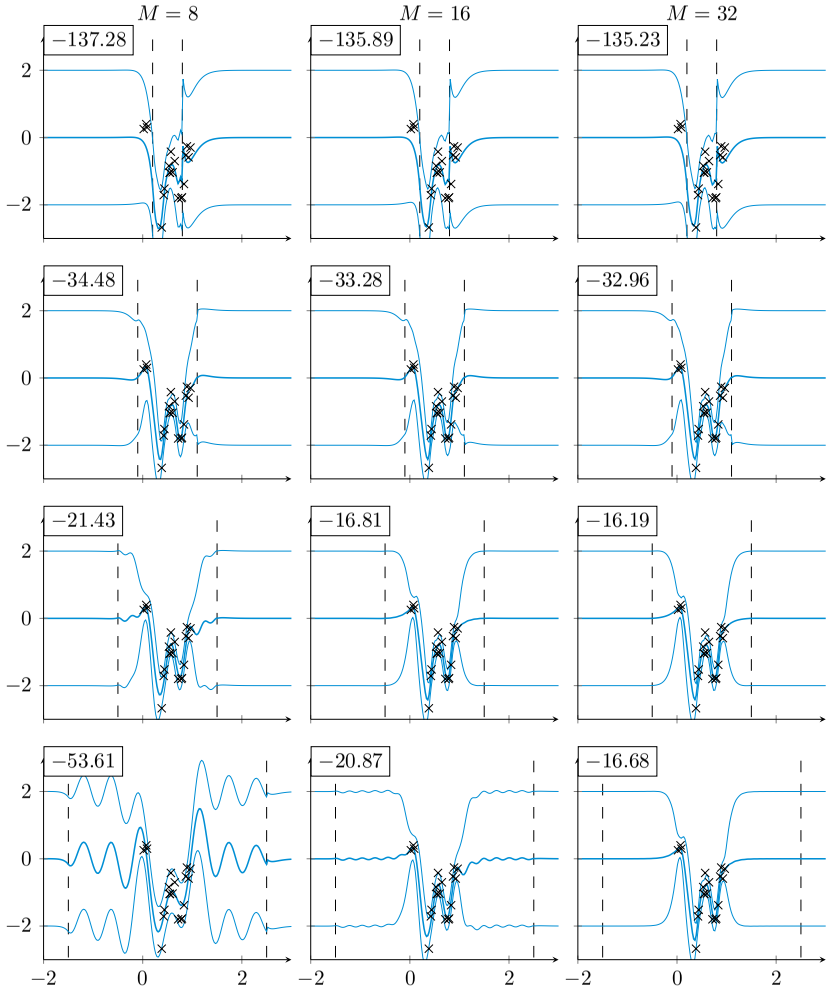


Figure 4: Demonstration of the effect of changing the interval size $[a, b]$ and the number of frequencies used. Each column has an increasing number of frequencies ($M = 8, M = 16, M = 32$), and each row has an increasing interval size, marked by vertical dashed lines. The data are drawn from a Matérn-$\frac{3}{2}$ kernel with Gaussian noise, and inference is performed with the kernel parameters set to the known values. The ELBO is shown in the top left of each plot. For reference, the true marginal likelihood in this problem is -15.99.

exponentially. In this section we explore the quality of our approximation in the product-kernel case, and compare with the sparse (inducing input) approach. We measure the time/accuracy tradeoff as a function of the input dimension and the number of inducing variables.

We consider toy data sets with input dimension $d = 1 \dots 4$, with input locations drawn uniformly over $[0, 1]^d$. The response data are given by GP samples with products of Matérn-$\frac{3}{2}$ kernels, with additive Gaussian observation noise:

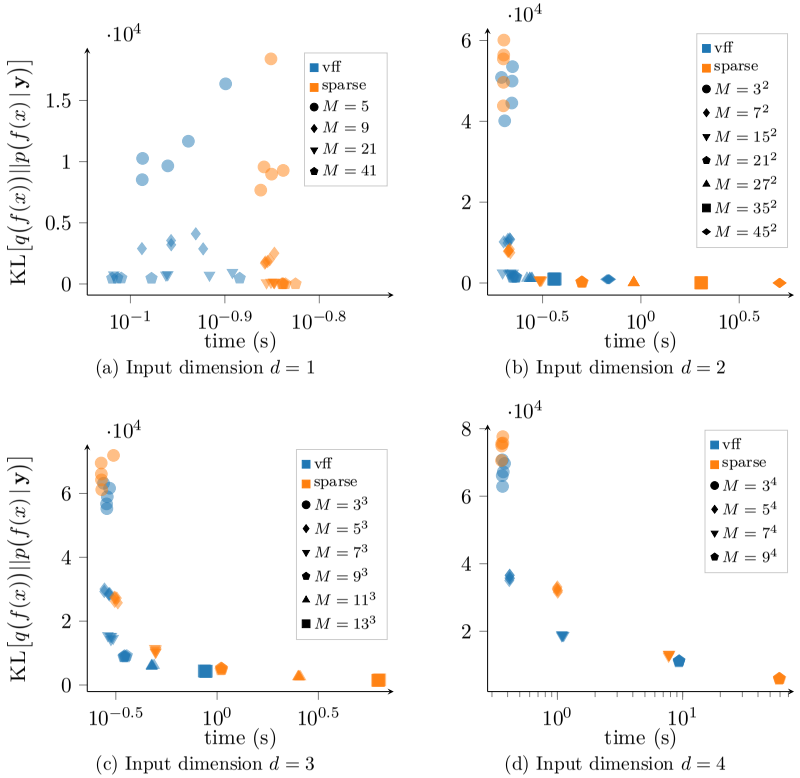
$$\begin{aligned} x_{i,j} &\sim \mathcal{U}(0,1) && \text{for } 1 \le i \le 10^4 \text{ and } 1 \le j \le d\,, \\ \mathbf{f} &\sim \mathcal{N}\Big(\mathbf{0}, \prod_{j=1}^{d} k_{3/2}(\mathbf{X}_j, \mathbf{X}_j)\Big)\,, \\ \mathbf{y} &\sim \mathcal{N}\big(\mathbf{f},\, 0.1 \times \mathbf{I}\big)\,. \end{aligned} \tag{77}$$

Given these data, we compare two approximations: one sparse GP with $M = m^d$ inducing inputs located on a regular grid with minimal and maximal values equal to 0 and 1 for each coordinate and a VFF approximation based on a Kronecker structure with $M = (m-1)/2$ frequencies for each input dimension which also leads to m^d inducing variables. For both approximations, the parameters of the covariance function were fixed to the known values used to generate the data (lengthscales $\ell = 0.2$, variance $\sigma^2 = 1$). The domain boundaries $[a, b]$ that are specific to the VFF model were fixed to $[-0.3, 1.3]$, which seemed to be a reasonable trade-off for the various dimensions after testing a few other values. We then compute the KL divergence between each approximation and the truth, by comparing the ELBO to the marginal likelihood from a 'full' GP regression model. We recorded the time required to compute the ELBO, and repeated the whole experiment five times.

The results are reported in Figure 5. In one dimension, both methods have a similar accuracy in term of KL for a given number of inducing variables but VFF appears to be faster than sparse GPs. However the low complexity of this one dimensional model does not allow us to measure the time accurately enough to distinguish the influence of the number of inducing variables on the execution time. More interesting conclusions can be drawn when the input space is greater than one: for the same number of inducing variables, VFF is significantly faster than the sparse GP (one fewer Cholesky decomposition is required) but the quality of the approximation is not as good as the sparse GP for the same number of basis functions.

For a given computational time, both methods have an equivalent accuracy but with a different number of inducing variables. The VFF basis functions appear to have less representational power, but their structure allows faster computations. Our understanding is that this smaller representational power of VFF can be explained by the fact that the inducing functions need to account for variations on $[a, b]^d$ whereas the sparse GP basis functions only need account for that on $[0, 1]^d$: a much smaller volume, especially in high dimensions. Note that the accuracy of the sparse GP can be improved by optimization of the location of the inducing inputs but that would result in computation times many times larger.

We have shown empirically that in the worst case, VFF is comparable to the sparse GP in terms of the available computate-time/accuracy tradeoff. For other models including



(a) Input dimension $d = 1$

(b) Input dimension $d = 2$

(c) Input dimension $d = 3$

(d) Input dimension $d = 4$

Figure 5: KL divergence versus wall clock time for sparse GPs and VFF with various number M of inducing inputs. The data is obtained by sampling a GP with Matérn-$\frac{3}{2}$ kernel ($\sigma^2 = 1$, $\ell = 0.2$) at 10^4 uniformly distributed input locations and $\mathcal{N}(0, 0.1)$ observation noise. The KL is computed with respect to the full GP regression model with known parameters for all models. The experiments are replicated five times with different draws of $\mathbf{X}$ and $\mathbf{y}$.

additive models, we expect much stronger performance from the VFF approximation, as the following sections demonstrate.

6.2 Additive modelling of airline delays

We demonstrate the feasibility of the variational spectral approximation in a high-dimensional GP regression example for predicting airline delays. The US flight delay prediction example (see Hensman et al., 2013, for the original example) has reached the status of a standard test data set in Gaussian process regression, partly because of its large-scale non-stationary nature and partly because of the massive size of the data set with nearly 6 million records.

This example has recently been used by Deisenroth and Ng (2015), where it was solved using distributed Gaussian processes. Samo and Roberts (2016) use this example for demonstrating the computational efficiency of string Gaussian processes. Furthermore, Adam et al. (2016) bring this problem forward as an example of a data set, where the model can be formed by the addition of multiple underlying components.

The data set consists of flight arrival and departure times for every commercial flight in the USA for the year 2008. Each record is complemented with details on the flight and the aircraft. The eight covariates $\mathbf{x}$ are the same as in Hensman et al. (2013), namely the age of the aircraft (number of years since deployment), route distance, airtime, departure time, arrival time, day of the week, day of the month, and month. We predict the delay of the aircraft at landing (in minutes), y.

Following the proposed solution by Adam et al. (2016) for this estimation problem, we use a Gaussian process regression model with a prior covariance structure given as a sum of Matérn-$\frac{3}{2}$ covariance functions for each input dimension, and assume the observations to be corrupted by independent Gaussian noise, $\varepsilon_i \sim \mathcal{N}(0, \sigma_n^2)$. The model is

$$f(\mathbf{x}) \sim \mathcal{GP}\left(0, \sum_{d=1}^{8} k(x_d, x'_d)\right), \tag{78}$$

$$y_i = f(\mathbf{x}_i) + \varepsilon_i, \tag{79}$$

for $i = 1, 2, \ldots, N$. For the Variational Fourier Feature method (VFF), we used $M = 30$ frequencies per input dimension.

We consider the entire data set of 5.93 million records. To provide comparable predictions, we follow the test setup in the paper by Samo and Roberts (2016), where results were calculated using String GPs (their method), the Bayesian committee machine (BCM, Tresp, 2000), the robust Bayesian committee machine (rBCM, Deisenroth and Ng, 2015), and stochastic variational GP inference (SVIGP, Hensman et al., 2013). The SVIGP used a squared-exponential kernel with one lengthscale per dimension. We repeated the SVIGP experiments of Samo and Roberts (2016) to complete the table for larger data sets and predictive densities.

Predictions are made for several subset sizes of the data, each selected uniformly at random: 10,000, 100,000, 1,000,000, and 5,929,413 (all data). In each case, two thirds of the data is used for training and one third for testing. For each subset size, the training is repeated ten times, and for each run, the outputs are normalized by subtracting the training sample mean from the outputs and dividing the result by the sample standard deviation. In the VFF method, the inputs were normalized to $[0, 1]$ and the domain size was set to $[-2, 3]$, i.e. the boundary is at a distance of two times the range of the data for each dimension.

N	10,000		100,000		1,000,000		5,929,413	
	MSE	NLPD	MSE	NLPD	MSE	NLPD	MSE	NLPD
VFF	0.89 ± 0.15	1.362 ± 0.091	0.82 ± 0.05	1.319 ± 0.030	0.83 ± 0.01	1.326 ± 0.008	0.827 ± 0.004	1.324 ± 0.003
Full-RBF	0.89 ± 0.16	1.349 ± 0.098	N/A	N/A	N/A	N/A	N/A	N/A
Full-additive	0.89 ± 0.16	1.362 ± 0.096	N/A	N/A	N/A	N/A	N/A	N/A
SVIGP	0.89 ± 0.16	1.354 ± 0.096	0.79 ± 0.05	1.299 ± 0.033	0.79 ± 0.01	1.301 ± 0.009	0.791 ± 0.005	1.300 ± 0.003
String GP†	1.03 ± 0.10	N/A	0.93 ± 0.03	N/A	0.93 ± 0.01	N/A	0.90 ± 0.01	N/A
rBCM†	1.06 ± 0.10	N/A	1.04 ± 0.04	N/A	N/A	N/A	N/A	N/A

Table 2: Predictive mean squared errors (MSEs) and negative log predictive densities (NLPDs) with one standard deviation on the airline arrival delays experiment. †MSE results as reported by Samo and Roberts (2016), who also list results for the BCM and rBCM methods.

Table 2 shows the predictive mean squared errors (MSEs) and the negative log predictive densities (NLPDs) with one standard deviation on the airline arrival delays experiment. In this form the MSEs over the normalized data can be interpreted as a fraction of the sample variance of airline arrival delays. Thus a MSE of 1.00 is an accuracy equivalent to using the training mean as predictor. The results for the String GP method are included for reference only and they are given as reported by Samo and Roberts (2016). They also list results for the rBCM method, which performed worse than the String GP.

Running the VFF experiment with all 5.93 million data using our Python implementation took 265 ± 6 seconds (626 ± 11 s CPU time) on a two-core MacBook Pro laptop (with all calculation done on the CPU). The results for 10,000 data were calculated in 21 ± 2 seconds (27 ± 4 s CPU time). For comparison, according to Samo and Roberts (2016), running the String GP experiment took 91.0 hours total CPU time (or 15 h of wall-clock time on an 8-core machine). Even the SVIGP method required 5.1 ± 0.1 hours of computing (27.0 ± 0.8 h CPU time) on a cluster.

For comparison, we have also included the results for the naive full GP solution for 10,000 data. These results were computed on a computing cluster and the computation time was several hours per repetition. We show results both for a squared-exponential kernel with one lengthscale per dimension, and also with an additive kernel similar to the one defined for the VFF method. As one might expect, the RBF model had a slightly lower average MSE (0.8898) than the additive model (0.89274). The difference is small: only 3 seconds of RMSE in the original units of flight-delay. The VFF result (0.8934) for the same data is close to the full GP solution (the difference being only about 0.66 s in terms of non-normalized RMSE). In terms of NLPD the results agree as well; the full RBF model is able to capture the phenomenon slightly better than the full additive model, and the VFF model approximates well the full additive model. The variability in the data is large, and including more data in the GP training gives the model more power for prediction. The SVIGP model (with a squared-exponential kernel) is the most flexible one and reaches the lowest RMSE and NLPD in the experiments.

Figure 6 shows the component-wise predictions for each of the covariates, thus summarizing the underlying structure of the delay prediction model. As one might expect, the effect of the month, day of month, and day of week are small. The only interpretable effects are seen in slightly higher delays in the summer months and around the holiday season

towards the end of the year, as well as some effects for the weekend traffic on Friday and Sunday. The age of the aircraft has a barely noticeable effect. The four remaining inputs (e–h) explain most of the delay. The effects of airtime and travel distance are strongly correlated. Strong correlation can also be seen in departure and arrival time effects, both of which show peaks during the night. The model also catches the periodic nature of the departure and arrival time effects. These results are in line with the analysis of Adam et al. (2016).

6.3 Classification with a Gaussian approximation

To perform variational classification using Variational Fourier Features, we can use a Gaussian approximation to the posterior $q(\mathbf{u})$ and optimize the ELBO with respect to the mean and variance of the approximation, as described in Section 2.4.4. To illustrate, we replicate the experiment on the banana data set from Hensman et al. (2015b). In that work, the authors showed that a variational inducing point method performed better than competing alternatives because the positions of the inducing points could be effectively optimized using the ELBO as the objective. In this work, we have selected the frequencies of the Variational Fourier Features to be placed on a regular grid: the only aspect of the Fourier features to be tuned is M, controlling the number of features used.

Since the variational framework provides a guarantee that more inducing variables must be monotonically better (Titsias, 2009), we expect that increasing the number of Variational Fourier Features will provide improved approximations. However, if we restrict the form of the Gaussian approximation by requiring a structured covariance, this may no longer be true. In practice, we have found that a freely-structured covariance matrix works well in one dimension and for additive models, but for Kronecker-structured models, optimization over an exponentially large matrix is intolerably slow. A suggestion for a related method provided by Nickson et al. (2015) is to force the Gaussian approximation to have a Kronecker-structured covariance matrix. This reduces the number of variables over which we have to optimize, but removes the guarantee that more inducing variables is always better. We find in practice that this structure does not work well: Figure 7 shows the ELBO as a function of the number of inducing variables M, for both a trained covariance and a Kronecker–structured one. We see that for the Kronecker structure, the ELBO decreases as M increases, implying that more inducing variables are actually making the approximation worse in this case.

Our suggestion is to use a sum of two Kronecker-structured matrices. The right-most plot of Figure 7 shows that the ELBO increases monotonically with M when we use this structure. We provide no guarantee that the sum of two Kronecker matrices is optimal, and suggest that future work might consider generalizing the form of this approximation.

To compare our method with the inducing-point method, we refer to Figure 8, which shows the results of fitting a Gaussian process classifier to the banana data set using increasing numbers of inducing points (IP) and increasing numbers of Fourier features (VFF). We observe the effect noted in Hensman et al. (2015b) that the inducing points move toward the decision boundary, though some slight differences from that experiment exist because of our choice of the product of Matérn-$\frac{5}{2}$ kernels, instead of the squared exponential kernel. The Figure also shows that the number of frequencies required to obtain a good approximation

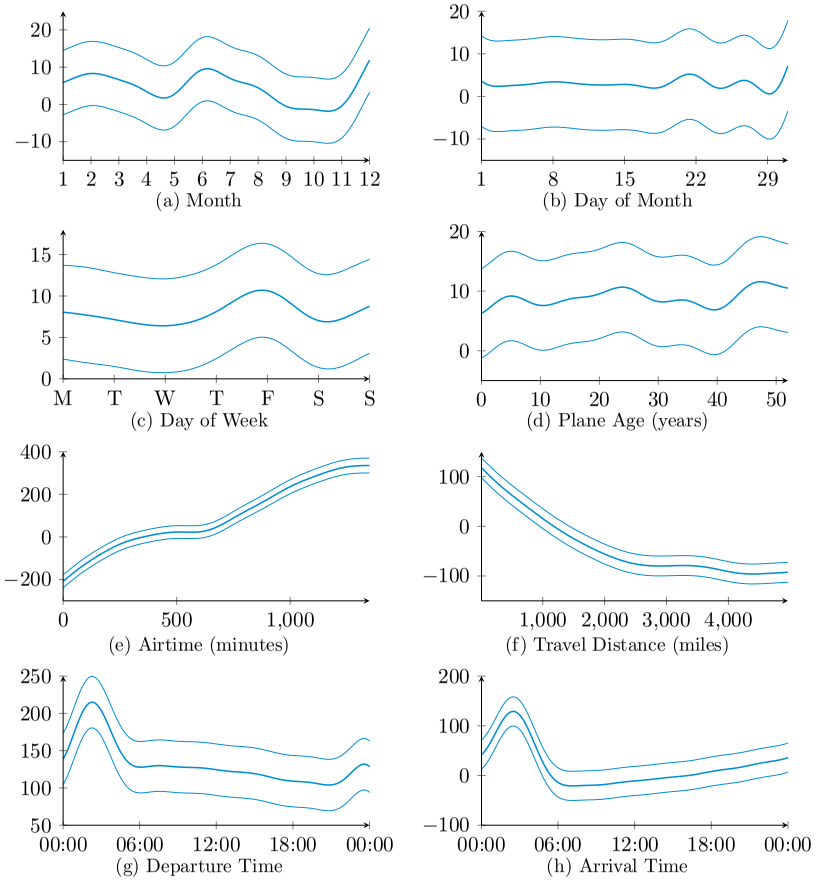


Figure 6: The posterior mean and two posterior standard deviations in the airline delay prediction experiment estimated from 5.93 million data. Each panel show the posterior over the effect from one covariate under an additive model. The vertical axis represents delay in minutes in each case.

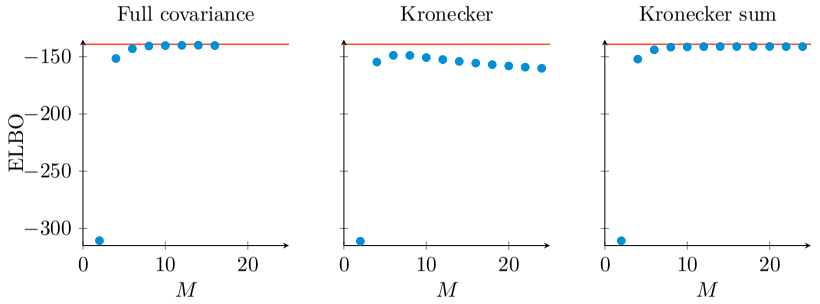


Figure 7: The increasing ELBO for the banana data set with increasing numbers of frequencies M, and a Gaussian approximation to the posterior. Left: $q(\mathbf{u})$ has an unconstrained covariance matrix. Middle: $q(\mathbf{u})$ has a Kronecker-structured covariance matrix. Right: $q(\mathbf{u})$ has a covariance matrix with a sum of two Kronecker-structured matrices. In each case the ELBO achieved using a full matrix decomposition is shown as a horizontal red line.

is reasonable, though it is not straightforward to compare directly between the VFF and IP methods. The total number of basis functions used in the Fourier features is twice the number of frequencies (M) listed, squared (since we have real and imaginary components, and take the Kronecker product). Still, the VFF method requires fewer computations per frequency than the IIP method does per inducing point, since the $\mathbf{K_{uu}}$ matrix is easily decomposed.

6.4 Solar irradiance

An alternative, but related idea to our proposed Variational Fourier Features approach is presented by Gal and Turner (2015). In that work, the authors used a Random Fourier Features model (as in equation (15)), but additionally performed a variational-Bayesian treatment of the frequencies. We repeat an experiment from that work on solar irradiance data (Lean, 2004), shown in Figure 9. In this experiment, the time series was normalized and segments of the data were removed; in Figure 9, removed data are shown in red, and vertical dashed lines denote the removed sections. The remaining data are shown in black, and we show the predicted mean (heavy blue line) and 95% confidence intervals (light blue lines). We fitted five models (or model approximations) to the remaining data, using 50 inducing points (or frequencies) except in the case of the Random Fourier Features, where we used 500 (following Gal and Turner, 2015). In each case, we used a Matérn-$\frac{5}{2}$ kernel initialized with a lengthscale of 10, and optimized the parameters with respect to the marginal likelihood (or ELBO, accordingly).

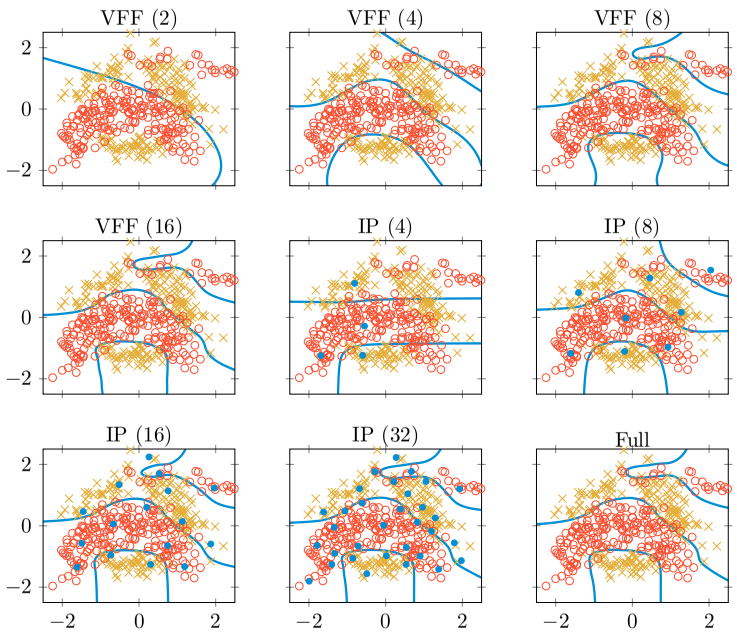


Figure 8: Classification of the banana data set with increasing number of Variational Fourier Features (VFF) and Inducing Points (IP). The two classes of data are shown as red circles and yellow crosses, and the decision boundary is shown as a solid blue line. An approximation with a full inversion of the covariance matrix is shown in the last plot. The blue dots show optimized positions of the inducing points. The Variational Fourier Features approach the true posterior more rapidly than the inducing point method.

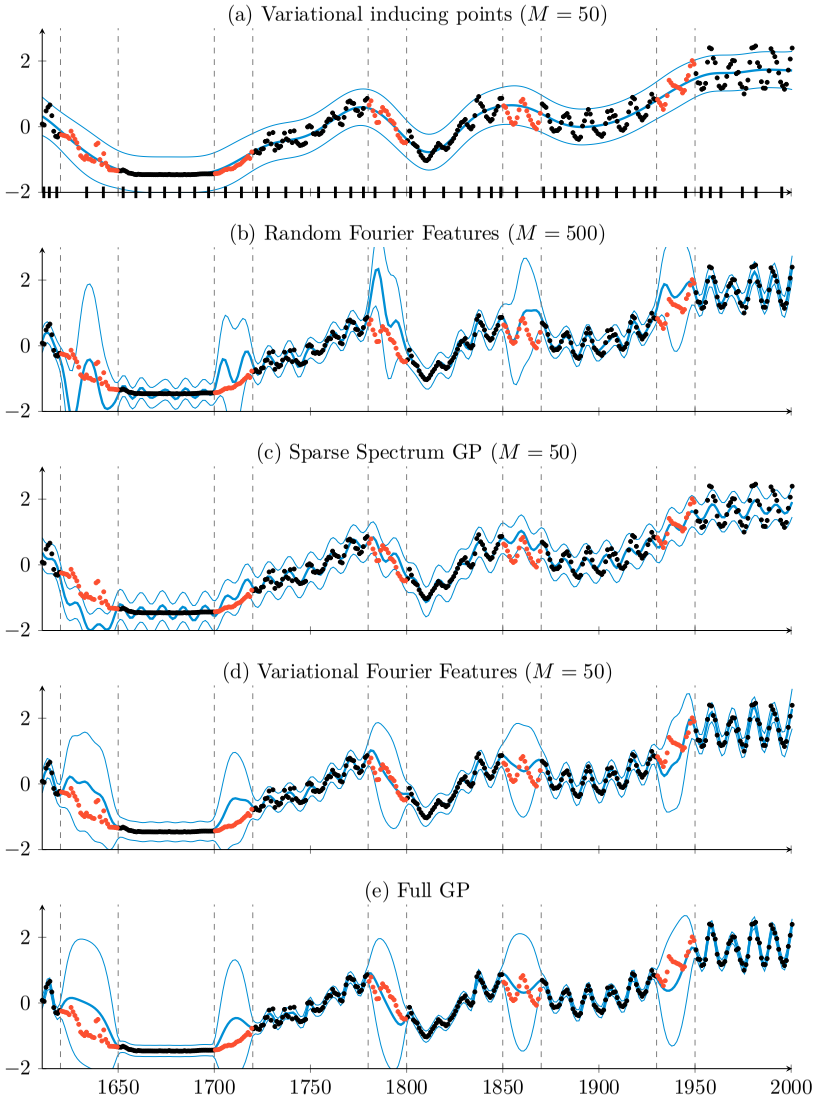


Figure 9: Replicating the experiment on solar irradiance data from Gal and Turner (2015).

Figure 9(a) shows the results of an inducing point method (still within the variational framework, based on Titsias, 2009). The positions of the inducing points z are shown as black marks on the horizontal axis. In this case, there are insufficient points to capture the complexity of the data, and the model reverts to a long lengthscale. This is an interesting case of underfitting, where the variational approximation to the function values gives rise to a bias in the estimate of the lengthscale parameter. This bias occurs because when maximising the ELBO with respect to the parameters, the KL divergence is smaller for longer lengthscales, biasing the parameter estimation in that direction. Turner and Sahani (2011) give an excellent discussion of this issue.

Figure 9(b) shows the results of the Random Fourier Features method. This method is given 500 frequencies; we concur with the findings of Gal and Turner (2015) that using only 50 frequencies gives very poor performance. Where the data are missing, the error bars provided by the model are large, similarly to the full GP (Figure 9(e)). The approximation is unable to recover the correct mean function, however, and is somewhat subject to overfitting: the model emphasises the high-frequency content in the data. In some sense, this is a desirable property as the model is more flexible than the original GP. For example, Yang et al. (2015) exploited this property to fit complex GP models. However, we advocate the variational method precisely because of the asymptotic convergence (in M) to the true posterior distribution of the original model.

Figure 9(c) shows the result of fitting the Sparse Spectrum GP approximation (Lázaro-Gredilla et al., 2010). This model is similar to the Random Fourier Features model, except that the frequencies are optimized. Here the overfitting is more severe: the model focusses entirely on the high-frequency content of the data. Although this gives good predictions in some cases (the segment around 1930 for example), the predictions are wild in others (e.g. around 1630, where none of the held-out data fall within the 95% intervals).

Figure 9(d) and (e) show the Variational Fourier Features method and the full GP method. The VFF approximation strives to approach the true posterior, and it is clear that the method is superior to the inducing points method in (a) in that the lengthscale is well estimated and the confidence intervals match the behaviour of the full GP. Some small discrepancies exist, but we note that these disappear completely at 100 inducing frequencies.

In summary, this experiment emphasises again how the variational approach attempts to approximate the posterior of the model, whereas previous approximations have *changed* the model, giving different behaviours. Within the variational methods, the Variational Fourier Features outperform the inducing point approach in this case, since our orthogonal basis functions are capable of capturing the posterior more effectively.

6.5 MCMC approximations

Above we have used the Gaussian approximation to the posterior process. Hensman et al. (2015a) propose a methodology which combines the variational approach with MCMC by sampling from the optimal approximating distribution, as we described in equation (27). The advantages of this approach are that the posterior need not be assumed Gaussian, and that the covariance parameters can also be treated in a Bayesian fashion. The VFF framework that we have described fits well into this scheme, and as we have already described,

the computational cost per iteration is linear in the number of data and in the number of required frequencies.

Here we replicate an experiment from Hensman et al. (2015a) (therein based on inducing point representations) in estimating the rate of a two-dimensional log Gaussian Cox process. In this experiment, we aim to emphasise the convergent properties of Variational Fourier Features, and show how they might be used in practice within the MCMC framework.

The data consist of a series of 127 points collected in a square area, denoting the location of pine saplings. This simple illustration was also used by Møller et al. (e.g. 1998). The model is

$$f(\mathbf{s}) \sim \mathcal{GP}\big(0, k(\mathbf{s}, \mathbf{s}')\big)\,, \tag{80}$$

$$\log \lambda(\mathbf{s}) = f(\mathbf{s}) + c\,, \tag{81}$$

$$\mathbf{y} \sim \mathcal{PP}(\lambda)\,, \tag{82}$$

where $\mathcal{PP}$ denotes an inhomogeneous Poisson process with intensity λ, $\mathbf{y}$ are spatial locations representing a draw from that Poisson process, k is a product of Matérn-$\frac{3}{2}$ covariance functions of the two-dimensional space indexed by $\mathbf{s}$, and c is a parameter to be inferred alongside the covariance function parameters. The model contains a doubly-intractable likelihood, which can be approximated by gridding the space and assuming that the intensity is constant across the bin size. The resulting approximation to the likelihood is

$$p(\mathbf{y} \mid \mathbf{f}) = \prod_{i=1}^{G} \mathcal{P}(N_i \mid \lambda(\mathbf{s}_i)\Delta)\,, \tag{83}$$

where G is the number of grid cells, $\mathbf{s}_i$ represent the centres of those cells, N_i is the number of data in the i^{th} cell, Δ is the area of a grid cell and $\mathcal{P}$ represents the Poisson distribution. To perform inference in this model, we decompose the GP with Variational Fourier features, use the rotation described in Section 5.5 to centre the $\mathbf{u}$ variables and run Hamiltonian Monte Carlo (HMC) to jointly sample the centred variables alongside the covariance function parameters and c.

To illustrate how the VFF method approaches the posterior in the MCMC case, we ran the HMC algorithm using $M = 14, 16, \ldots, 30$ frequencies. We selected the boundaries of the decomposition $[a, b]$ to be $[-1, 2]^2$, whilst the data were normalized to the unit square $[0, 1]^2$. The posterior mean intensity of the process on the unit square, along with the data, are shown in Figure 10. The figure shows that as M increases, the behaviour of the method stabilizes: there is little to distinguish $M = 30$ from $M = 28$. If M is too low, then there is a bias towards longer lengthscales, but this is alleviated as M increases. Since more inducing variables must always move the approximation toward the posterior in the KL sense (Titsias, 2009), once the optimal posterior (for each fixed M) remains unchanging with an increase in M, we must be approximating the posterior closely. A practical approach then, is to use some reasonable number for M (say, 100) and then ensure that the sampled distribution is unmoving with increased M. Since our experiments suggest that the lengthscale parameter is particularly sensitive to bias due to M being too low, we suggest examining the convergent behaviour of the lengthscale posterior. Figure 11 shows how the posterior distribution of the lengthscales converges as M increases.

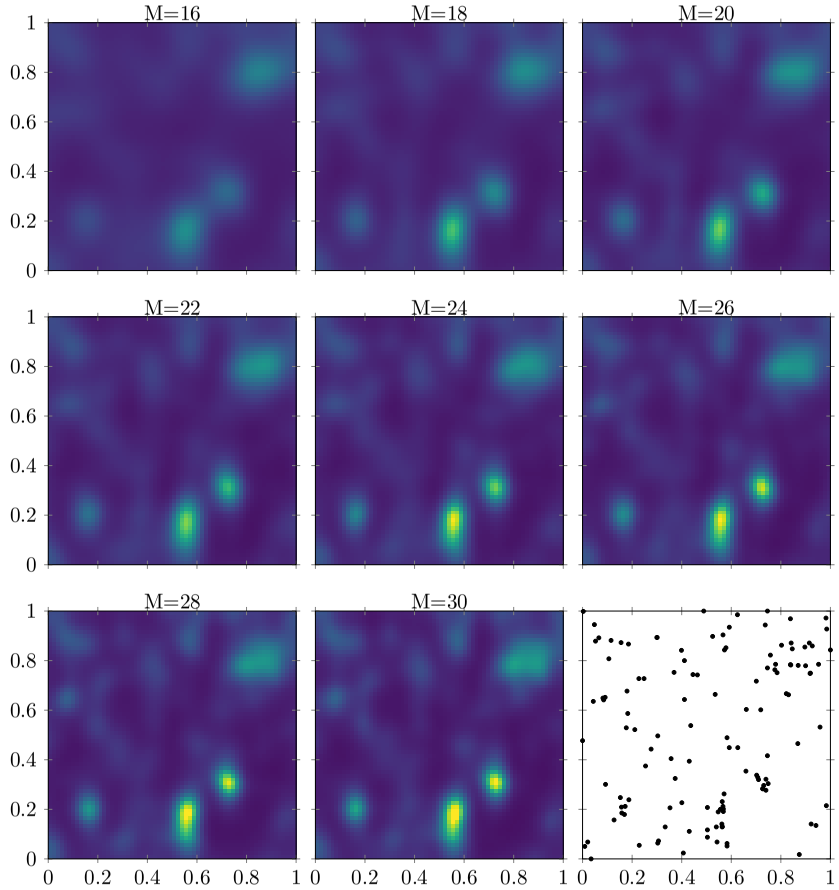


Figure 10: The posterior mean of the process intensity with increasing M (denoted above each figure). The spatial locations of the data are shown in the last pane. As M increases, the behaviour of the method stabilizes: we conclude that at $M = 30$ there is no benefit from increasing M further. If M is too low, the method is biased towards a longer lengthscale (see also Figure 11).

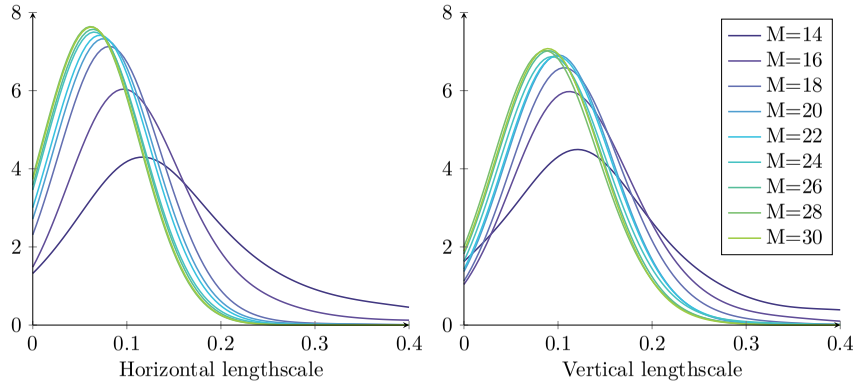


Figure 11: The posterior distribution of the two lengthscale parameters for the pines problem (see also Figure 10), estimated using kernel density smoothing on the MCMC trace. As more frequencies are used, the posterior distribution converges.

We note that the regular-lattice nature of this problem means that it can also be efficiently solved using fast Fourier transform (FFT) methods (e.g. Taylor and Diggle, 2014). However, our approach offers several advantages. First, in our method the number of frequencies used is a variational parameter: we have discussed how to select M above. Since the dimension of the vector which is subject to MCMC depends on the number of frequencies, it is desirable to decouple it from the grid density: fine grids are desired to resolve the spatial nature of the process, whilst the length of the vector $\mathbf{u}$ is desired to be short for computational reasons. Second, our approach does not require the embedding of the kernel matrix into a circulant matrix which can occasionally suffer from non-positive-definiteness. Third and most importantly, our method is not restricted to problems on a regular grid.

7. Discussion and future directions

In this paper we have presented a method which we refer to as the Variational Fourier Features (VFF) approach to Gaussian process modelling. It combines the variational approach to sparse approximations in Gaussian processes with a representation based on the spectrum of the process. This approach inherits its appealing approximative construction from the variational approach, but with the representative power and computational scalability of spectral representations.

We generalized the method to several dimensions in cases where the kernel exhibits special structure in the form of additive or separable (product) covariance functions. This choice made it possible to apply the scheme to higher-dimensional test problems in the experiments, for which we demonstrate good performance both in terms of computational speed and predictive power.

Our example application of predicting airline delays suggested that the additive structure worked well for that problem: for data subsets where the full model could be fitted, there was little performance difference between the additive and isotropic case, and indeed the VFF method was competitive with both, suggesting that the posterior was very well approximated. It seems implausible that this structure could work well for all problems, and clearly it will not work well where interesting interactions occur between covariates.

The method is capable of modelling interactions between covariates using separable kernel structures. As was noted in the text, the computational scalability in this case limits the method to low-dimensional problems, because the number of inducing frequencies scales exponentially with the input dimension. In this sense this approach is not a sparse approximation, but rather based on a 'dense' basis function projection. Like related approximation schemes (Solin and Särkkä, 2014; Wilson and Nickisch, 2015; Nickson et al., 2015), VFF scales well as the number of data increases, with the drawback of difficulty in dealing with high-dimensional problems out-of-the-box.

To further improve the computational scaling in high-dimensional problems, there are several approaches which could apply. Wilson et al. (2015) presents a projection approach (based on the ideas in Wilson and Nickisch, 2015). These ideas could be expanded to our approach as well, and we might expect the variational properties of the approximation to be useful in this case. Alternatively, replacing the dense grid spectrum with a set of sparsely well-chosen spectral points is an option that can — at least for models or data with suitable structure — provide computational remedy in high input dimensions. Future work might incorporate some ideas from the sparse spectrum method (Lázaro-Gredilla et al., 2010) where the frequencies are optimized, though we might hope that the variational formulation could prevent over-fitting.

We note that one of the key computations in our method, the multiplication $\mathbf{K_{uf}y}$, is precisely a non-uniform FFT (NUFFT, see Dutt and Rokhlin, 1993; Greengard and Lee, 2004; Bernstein et al., 2004), which can be approximated in $\mathcal{O}(N \log N)$ operations. Although we have so far avoided using this technology in our implementations, some initial experiments suggest that further speed-up is possible at the cost of a very small loss in accuracy. In the special case where the data lie on a regular grid, it is possible to compute the product $\mathbf{K_{uf}y}$ exactly in $\mathcal{O}(N \log N)$ operations using an FFT. Again we have not exploited this in our presentation so far, and future work may consider the relations between VFF and FFT methods based on the circulant embedding trick (Taylor and Diggle, 2014; Turner, 2010).

A further limitation of our current presentation is that we have only considered Matérn covariance functions with half-integer orders. A productive future direction will be to expand the number of kernels which can be decomposed using Variational Fourier Features. An interesting class of kernels may be the spectral mixture kernel (Wilson and Adams, 2013). Although that work was based on frequency-shifted versions of the RBF covariance, we anticipate that frequency-shifted Matérn kernels would work just as well, and may be more amenable to the VFF framework.

In our derivations, we have considered uniform input densities on some bounded interval $[a, b]$. Instead of considering such compact bounded subsets of $\mathbb{R}$, it might be possible to extend our methodology to more general input densities (cf. Williams and Seeger, 2000). In special cases this can lead to convenient approximations for the eigen-decomposition of the

kernel. For example, for a RBF covariance function and a Gaussian input density, the eigenfunctions can be given in closed-form in terms of Hermite polynomials (see Williams and Rasmussen, 2006, Ch. 4.3). There is a connection between harmonic Fourier approximations like ours and the covariance operator defined through the covariance function and the input density. The connection goes back to Sturm-Liouville theory and is further discussed in Solin and Särkkä (2014).

Finally, we note that VFF may be particularly well-suited to machine learning methods where Gaussian processes are embedded in some further model structure, such as the Gaussian process latent variable model (GPLVM, Lawrence, 2005) and its variational-Bayesian variant (Titsias and Lawrence, 2010), as well as deep Gaussian processes (Damianou and Lawrence, 2013). As described by Damianou et al. (2016), variational inference in these models involves propagating uncertainty by convolving the $\mathbf{K_{uf}}$ matrix with Gaussian approximations to the input distribution. Since the VFF method has precisely sinusoidal $\mathbf{K_{uf}}$ matrices, this convolution will take a simple form, and we envisage that combining the VFF framework with variational uncertainty propagation will lead to substantial improvements in inference for these models.

Acknowledgements

Part of this research was conducted during the invitation of J. Hensman at Mines Saint-Étienne. This visit was funded by the Chair in Applied Mathematics OQUAIDO, which gather partners in technological research (BRGM, CEA, IFPEN, IRSN, Safran, Storengy) and academia (Ecole Centrale de Lyon, Mines Saint-Étienne, University of Grenoble, University of Nice, University of Toulouse) on the topic of advanced methods for computer experiments. J. Hensman gratefully acknowledges a fellowship from the Medical Research Council UK. A. Solin acknowledges the Academy of Finland grant 308640. We acknowledge the computational resources provided by the Aalto Science-IT project. J. Hensman would like to thank T. Smith for insightful discussions. The authors would like to thank T. Bui, A. Vehtari, S.T. John and the anonymous reviewers who helped to improve this manuscript.

Appendix A. L_2 Projection on Fourier Features for Matérn-$\frac{1}{2}$

In this Appendix we derive expressions for Fourier features using the L_2 norm.

A.1 Covariance between inducing variable and GP

$$\operatorname{cov}(u_m, f(x)) = \mathbb{E}\left[u_m f(x)\right] = \int_a^b \mathbb{E}[f(t)f(x)]e^{-i\omega_m(t-a)}\,\mathrm{d}t = \int_a^b k(t,x)e^{-i\omega_m(t-a)}\,\mathrm{d}t \quad (84)$$

$$= e^{-i\omega_m a}\sigma^2\left(\int_a^x e^{\lambda(t-x)+i\omega_m t}\,\mathrm{d}t + \int_x^b e^{\lambda(x-t)+i\omega_m t}\,\mathrm{d}t\right) \quad (85)$$

$$= e^{-i\omega_m a}\sigma^2\left(\left[\frac{e^{\lambda(t-x)+i\omega_m t}}{i\omega_m+\lambda}\right]_a^x + \left[\frac{e^{\lambda(x-b)+i\omega_m t}}{i\omega_m-\lambda}\right]_x^b\right) \quad (86)$$

$$= e^{-i\omega_m a}\sigma^2\left(\frac{e^{i\omega_m x}-e^{\lambda(a-x)+i\omega_m a}}{i\omega_m+\lambda} + \frac{e^{\lambda(x-b)+i\omega_m b}-e^{i\omega_m x}}{i\omega_m-\lambda}\right) \quad (87)$$

$$= \sigma^2\left(\frac{e^{i\omega_m(x-a)}-e^{\lambda(a-x)}}{i\omega_m+\lambda} + \frac{e^{\lambda(x-b)}-e^{i\omega_m(x-a)}}{i\omega_m-\lambda}\right) \quad (88)$$

$$= \frac{2\sigma^2\lambda}{\lambda^2+\omega_m^2}e^{i\omega_m(x-a)} + \frac{\sigma^2}{\lambda^2+\omega_m^2}\left(\lambda[-e^{a-x}-e^{x-b}] + i\omega_m[e^{a-x}-e^{x-b}]\right) \quad (89)$$

$$= s_{1/2}(\omega_m)e^{i\omega_m(x-a)} + s_{1/2}(\omega_m)\frac{1}{2\lambda}\left(\lambda[-e^{a-x}-e^{x-b}] + i\omega_m[e^{a-x}-e^{x-b}]\right). \quad (90)$$

A.2 Covariance between inducing variables

For the L_2 Fourier features, the covariance between two features u_m and $u_{m'}$ depends on the basis functions being sines or cosines. We will thus detail the two cases considering either the real or imaginary part of $\mathbf{u}$. The following integrals, which can be computed using $\cos(\omega x) = \mathcal{R}\mathrm{e}(e^{i\omega x})$, will be of particular interest:

$$\int_x^y e^{\lambda s}\cos(\omega s)\,\mathrm{d}s = \frac{1}{\lambda^2+\omega^2}\left[e^{\lambda x}(-\lambda\cos(\omega x)-\omega\sin(\omega x)) + e^{\lambda y}(\lambda\cos(\omega y)+\omega\sin(\omega y))\right], \quad (91)$$

$$\int_x^y e^{\lambda s}\sin(\omega s)\,\mathrm{d}s = \frac{1}{\lambda^2+\omega^2}\left[e^{\lambda x}(\omega\cos(\omega x)-\lambda\sin(\omega x)) + e^{\lambda y}(-\omega\cos(\omega y)+\lambda\sin(\omega y))\right]. \quad (92)$$

case 1: $i, j \le M$ (cosine block)

$$\operatorname{cov}[u_m, u_{m'}] = \mathbb{E}\left[\int_a^b f(s)\cos(-\omega_m(s-a))\,\mathrm{d}s\int_a^b f(t)\cos(-\omega_{m'}(t-a))\,\mathrm{d}t\right] \quad (93)$$

$$= \int_a^b\int_a^b k(s,t)\cos(-\omega_m(s-a))\cos(-\omega_{m'}(t-a))\,\mathrm{d}s\,\mathrm{d}t \quad (94)$$

$$= \int_0^{a-b}\int_0^{b-a} k(s,t)\cos(\omega_m s)\cos(\omega_{m'}t)\,\mathrm{d}s\,\mathrm{d}t \quad (95)$$

$$= \frac{1}{\lambda^2+\omega_m^2}\int_0^{b-a}\left[2\lambda\cos(\omega_m t) - \lambda(e^{-\lambda t}+e^{\lambda(t+a-b)})\right]\cos(\omega_{m'}t)\,\mathrm{d}t \quad (96)$$

$$= \frac{2\lambda}{\lambda^2+\omega_m^2}\int_a^b\left[\cos(\omega_m t) - e^{-\lambda t}\right]\cos(\omega_{m'}t)\,\mathrm{d}t\,. \quad (97)$$

case 1a: $m \neq m'$

$$\operatorname{cov}[u_m, u_{m'}] = \frac{-2\lambda}{\lambda^2+\omega_m^2}\int_0^{b-a} e^{-\lambda t}\cos(\omega_{m'}t)\,\mathrm{d}t \quad (98)$$

$$= \frac{-2\lambda^2}{(\lambda^2+\omega_m^2)(\lambda^2+\omega_{m'}^2)}\left[1-e^{\lambda(a-b)}\right]. \quad (99)$$

case 1b: $m = m' \neq 0$

$$\operatorname{cov}[u_m, u_{m'}] = \frac{-2\lambda^2}{(\lambda^2+\omega_m^2)(\lambda^2+\omega_{m'}^2)}\left[1-e^{\lambda(a-b)}\right] + (b-a)\frac{\lambda}{\lambda^2+\omega_m^2}\,. \quad (100)$$

case 1c: $m = m' = 0$

$$\operatorname{cov}[u_m, u_{m'}] = \frac{-2\lambda^2}{(\lambda^2+\omega_m^2)(\lambda^2+\omega_{m'}^2)}\left[1-e^{\lambda(a-b)}\right] + 2(b-a)\frac{\lambda}{\lambda^2+\omega_m^2}\,. \quad (101)$$

case 2: m, $m' > M$ (sine block)

$$\operatorname{cov}[u_m, u_{m'}] = \int_0^{b-a}\int_0^{b-a} k(s,t)\sin(\omega_m s)\sin(\omega_{m'}t)\,\mathrm{d}s\;\,\mathrm{d}t \quad (102)$$

$$= \frac{2}{\lambda^2+\omega_m^2}\int_0^{b-a}\left[\lambda\sin(\omega_m t) + \omega_m e^{-\lambda t}\right]\sin(\omega_{m'}t)\,\mathrm{d}t\,. \quad (103)$$

case 2a: $m \neq m'$

$$\operatorname{cov}[u_m, u_{m'}] = \frac{2\omega_m}{\lambda^2+\omega_m^2}\int_0^{b-a} e^{-\lambda t}\sin(\omega_{m'}t)\,\mathrm{d}t \quad (104)$$

$$= \frac{2\omega_m\omega_{m'}}{(\lambda^2+\omega_m^2)(\lambda^2+\omega_{m'}^2)}\left[1-e^{\lambda(a-b)}\right]. \quad (105)$$

case 2b: $m = m'$

$$\operatorname{cov}[u_m, u_{m'}] = \frac{2\omega_{m'}^2}{(\lambda^2+\omega_m^2)(\lambda^2+\omega_{m'}^2)}\left[1-e^{\lambda(a-b)}\right] + (b-a)\frac{\lambda}{\lambda^2+\omega_m^2}\,. \quad (106)$$

Appendix B. Matérn inner products

The expressions of inner products for Matérn RKHS on $[a, b]$ can be found in Durrande et al. (2016). We adopt here the following notations to obtain compact expressions: for any function $h \in \mathcal{H}$, $\mathcal{I} : \ h \to h$ is the identity operator and $\mathcal{D} : \ g \to g'$ is the differentiation operator. As a consequence, $(\lambda\mathcal{I} + \mathcal{D})^2(h)$ is a shorthand for $\lambda^2 h + 2\lambda h' + h''$.

$$\text{Matérn-}\tfrac{1}{2}\text{: } \langle g, h\rangle_{\mathcal{H}_{1/2}} = \frac{1}{2\lambda\sigma^2}\int_a^b (\lambda\mathcal{I}+\mathcal{D})(g)(\lambda\mathcal{I}+\mathcal{D})(h)\,\mathrm{d}t + \frac{1}{\sigma^2}g(a)h(a)\,, \tag{107}$$

$$\begin{aligned}\text{Matérn-}\tfrac{3}{2}\text{: } \langle g, h\rangle_{\mathcal{H}_{3/2}} &= \frac{1}{4\lambda^3\sigma^2}\int_a^b (\lambda\mathcal{I}+\mathcal{D})^2(g)(\lambda\mathcal{I}+\mathcal{D})^2(h)\,\mathrm{d}t \\ &\quad + \frac{1}{\sigma^2}g(a)h(a) + \frac{1}{\lambda^2\sigma^2}g'(a)h'(a)\,,\end{aligned} \tag{108}$$

$$\begin{aligned}\text{Matérn-}\tfrac{5}{2}\text{: } \langle g, h\rangle_{\mathcal{H}_{5/2}} &= \frac{3}{16\lambda^5\sigma^2}\int_a^b (\lambda\mathcal{I}+\mathcal{D})^3(g)(\lambda\mathcal{I}+\mathcal{D})^3(h)\,\mathrm{d}t \\ &\quad + \frac{9}{8\sigma^2}g(a)h(a) + \frac{9}{8\lambda^4\sigma^2}g(a)''h''(a) \\ &\quad + \frac{3}{\lambda^2\sigma^2}\left(g'(a)h'(a) + \frac{1}{8}g''(a)h(a) + \frac{1}{8}g(a)h''(a)\right)\,.\end{aligned} \tag{109}$$

B.1 Gram matrix between Fourier features for the exponential kernel

We detail here the computations of $\mathbf{K_{uu}}[i, j] = \langle\phi_i, \phi_j\rangle_{\mathcal{H}}$ for the exponential kernel. We recall that $\phi_0 = 1$ and that $\phi_i = \cos(\omega_i(x-a))$ and $\phi_{i+M} = \sin(\omega_i(x-a))$ for $i \in \{1, \dots, M\}$. Furthermore, the frequencies ω_i are harmonic on $[a, b]$.

case 1 : $i, j \le M$ (cosine block)

$$\mathbf{K_{uu}}[i, j] = \frac{1}{2\sigma^2\lambda}\int_0^{b-a}(\lambda\cos(\omega_i s) - \omega_i\sin(\omega_i s))(\lambda\cos(\omega_j s) - \omega_j\sin(\omega_j s))\,\mathrm{d}s + \frac{1}{\sigma^2}\,. \tag{110}$$

The integral is zero for all non-diagonal terms ($i \neq j$). As a consequence, the block of $\mathbf{K_{uu}}$ associated with the cosine basis functions are $\mathrm{diag}(\alpha_{cos}) + \sigma^{-2}$, where

$$\alpha_{cos}[i] = \frac{1}{2\sigma^2\lambda}\int_0^{b-a}\lambda^2\cos(\omega_i s)^2 + \omega_i^2\sin(\omega_i s)^2\,\mathrm{d}s = \begin{cases}\dfrac{\lambda(b-a)}{2\sigma^2} & \text{if } i = 0\,,\\ \dfrac{b-a}{4\sigma^2\lambda}(\lambda^2+\omega_i^2) & \text{if } i \neq 0\,,\end{cases} \tag{111}$$

which leads to $\alpha_{cos} = \frac{1}{2}(b-a)\left[2s(0)^{-1}, s(\omega_1)^{-1}, \dots, s(\omega_M)^{-1}\right]$.

case 2 : $i, j > M$ (sine block)

$$\mathbf{K_{uu}}[i, j] = \frac{1}{2\sigma^2\lambda}\int_0^{b-a}(\lambda\sin(\omega_i s) - \omega_i\cos(\omega_i s))(\lambda\sin(\omega_j s) - \omega_j\cos(\omega_j s))\,\mathrm{d}s\,. \tag{112}$$

The $\mathbf{K_{uu}}$ block associated to the sine basis functions is exactly a diagonal matrix with:

$$\mathbf{K_{uu}}[i, i] = \frac{1}{2\sigma^2\lambda}\int_0^{b-a}(\lambda^2\sin(\omega_i s)^2 + \omega_i^2\cos(\omega_i s))\,\mathrm{d}s = \frac{b-a}{4\sigma^2\lambda}(\lambda^2+\omega_i^2)\,. \tag{113}$$

Similarly to the cosine block, we can write the sine block as $\mathrm{diag}(\alpha_{sin})$ with $\alpha_{sin} = \frac{1}{2}(b-a)\left[s(\omega_1)^{-1}, \dots, s(\omega_M)^{-1}\right]$.

case 3 : $i \le 2M + 1 < j$ (off-diagonal block) leads to $\mathbf{K}_{\phi\phi}[i, j] = 0$.

B.2 Gram matrix associated to Fourier features for the Matérn-$\frac{3}{2}$ kernel

case 1 : $i, j \le M$ (cosine block)

$$\begin{aligned}\mathbf{K}_{\phi\phi}[i, j] &= \frac{1}{4\sigma^2\lambda^3}\int_0^{b-a}(\lambda^2\cos(\omega_i s) - 2\lambda\omega_i\sin(\omega_i s) - \omega_i^2\cos(\omega_i s))\times \\ &\qquad (\lambda^2\cos(\omega_j s) - 2\lambda\omega_j\sin(\omega_j s) - \omega_j^2\cos(\omega_j s))\,\mathrm{d}s \\ &\quad + \frac{1}{\sigma^2}\cos(\omega_i a)\cos(\omega_j a) + \frac{1}{\lambda^2\sigma^2}\omega_i\omega_j\sin(\omega_i a)\sin(\omega_j a) \\ &= \begin{cases}\dfrac{1}{\sigma^2} & \text{if } i \neq j\,,\\ \dfrac{1}{4\sigma^2\lambda^3}\displaystyle\int_0^{b-a}\lambda^4\cos^2(\omega_i s) + \omega_i^4\cos^2(\omega_i s) & \\ \qquad -2\lambda^2\omega^2\cos^2(\omega_i s) + 4\lambda^2\omega_i^2\sin^2(\omega_i s)ds + \dfrac{1}{\sigma^2} & \text{if } i = j\,,\end{cases} \\ &= \begin{cases}\dfrac{1}{\sigma^2} & \text{if } i \neq j\,,\\ \dfrac{b-a}{8\sigma^2\lambda^3}(\lambda^2+\omega_i^2)^2 + \dfrac{1}{\sigma^2} & \text{if } i = j \neq 0\,,\\ \dfrac{\lambda(b-a)}{4\sigma^2} + \dfrac{1}{\sigma^2} & \text{if } i = j = 0\,.\end{cases}\end{aligned} \tag{114}$$

case 2 : $i, j > M$ (sine block)

$$\begin{aligned}\mathbf{K}_{\phi\phi}[i, j] &= \frac{1}{4\sigma^2\lambda^3}\int_0^{b-a}(\lambda^2\sin(\omega_i s) + 2\lambda\omega_i\cos(\omega_i s) - \omega_i^2\sin(\omega_i s))\times \\ &\qquad (\lambda^2\sin(\omega_j s) + 2\lambda\omega_j\cos(\omega_j s) - \omega_j^2\sin(\omega_j s))\,\mathrm{d}s \\ &\quad + \frac{1}{\sigma^2}\sin(\omega_i a)\sin(\omega_j a) + \frac{1}{\lambda^2\sigma^2}\omega_i\omega_j\cos(\omega_i a)\cos(\omega_j a) \\ &= \begin{cases}\dfrac{\omega_i\omega_j}{\lambda^2\sigma^2} & \text{if } i \neq j\,,\\ \dfrac{1}{4\sigma^2\lambda^3}\displaystyle\int_0^{b-a}\lambda^4\sin^2(\omega_i s) + \omega_i^4\sin^2(\omega_i s) & \\ \qquad -2\lambda^2\omega^2\sin^2(\omega_i s) + 4\lambda^2\omega_i^2\cos^2(\omega_i s)ds + \dfrac{\omega_i\omega_j}{\lambda^2\sigma^2} & \text{if } i = j\,,\end{cases} \\ &= \begin{cases}\dfrac{\omega_i\omega_j}{\lambda^2\sigma^2} & \text{if } i \neq j\,,\\ \dfrac{b-a}{8\sigma^2\lambda^3}(\lambda^2+\omega_i^2)^2 + \dfrac{\omega_i\omega_j}{\lambda^2\sigma^2} & \text{if } i = j\,.\end{cases}\end{aligned} \tag{115}$$

case 3: $i \leq M < j$ (off-diagonal block)

$$\begin{aligned}\mathbf{K}_{\phi\phi}[i,j] &= \frac{1}{4\sigma^2\lambda^3}\int_0^{b-a} (\lambda^2\cos(\omega_i s) - 2\lambda\omega_i\sin(\omega_i s) - \omega_i^2\cos(\omega_i s))\times \\ &\qquad (\lambda^2\sin(\omega_j s) + 2\lambda\omega_j\cos(\omega_j s) - \omega_j^2\sin(\omega_j s))\,\mathrm{d}s \\ &\quad + \frac{1}{\sigma^2}\cos(\omega_i a)\sin(\omega_j a) - \frac{1}{\lambda^2\sigma^2}\omega_i\omega_j\sin(\omega_i a)\cos(\omega_j a) \\ &= \begin{cases} 0 & \text{if } i \neq j\,, \\ \frac{1}{4\sigma^2\lambda^3}\int_0^{b-a} 2\lambda^3\omega_i\cos^2(\omega_i s) - 2\lambda\omega_i^3\cos^2(\omega_i s) & \\ \qquad -2\lambda^3\omega_i\sin^2(\omega_i s) + 2\lambda\omega_i^3\sin^2(\omega_i s)\,\mathrm{d}s & \text{if } i = j\,, \end{cases} \\ &= 0\end{aligned} \tag{116}$$

B.3 Gram matrix of Fourier features for the Matérn-$\frac{5}{2}$ kernel

$$\langle g, h\rangle_{\mathcal{H}_{5/2}} = \frac{3}{16\lambda^5\sigma^2}\int_0^{b-a} L_t(g)L_t(h)\,\mathrm{d}t + G(a)\,, \tag{117}$$

$$\begin{aligned}\text{where } L_t(g) &= \left(\lambda^3 g(t) + 3\lambda^2 g'(t) + 3\lambda g''(t) + g'''(t)\right)\,, \\ G(a) &= \frac{9}{8\sigma^2}g(a)h(a) + \frac{9}{8\lambda^4\sigma^2}g(a)''h''(a) \\ &\quad + \frac{3}{\lambda^2\sigma^2}\left(g'(a)h'(a) + \frac{1}{8}g''(a)h(a) + \frac{1}{8}g(a)h''(a)\right)\,.\end{aligned} \tag{118}$$

case 1: $i, j \leq M$ (cosine block)

$$\begin{aligned}L_x(\cos(\omega_i .)) &= \lambda^3\cos(\omega_i x) - 3\omega_i\lambda^2\sin(\omega_i x) - 3\lambda\omega_i^2\cos(\omega_i x) + \omega_i^3\sin(\omega_i x) \\ &= (\lambda^3 - 3\lambda\omega_i^2)\cos(\omega_i x) + (\omega_i^3 - 3\omega_i\lambda^2)\sin(\omega_i x) \\ G(a) &= \frac{9}{8\sigma^2} + \frac{9}{8\lambda^4\sigma^2}\omega_i^2\omega_j^2 - \frac{3}{8\lambda^2\sigma^2}(\omega_i^2 + \omega_j^2) \\ &= \frac{1}{\sigma^2} + \frac{1}{8\sigma^2}\left(\frac{3\omega_i^2}{\lambda^2} - 1\right)\left(\frac{3\omega_j^2}{\lambda^2} - 1\right)\,.\end{aligned} \tag{119}$$

$$\mathbf{K}_{\phi\phi}[i,j] = \begin{cases} G(a) & \text{if } i \neq j\,, \\ \frac{3(b-a)}{32\lambda^5\sigma^2}(\lambda^6 - 6\lambda^4\omega_i^2 + 9\lambda^2\omega_i^4 + \omega_i^6 - 6\lambda^2\omega_i^4 + 9\omega_i^2\lambda^4) + G(a) & \text{if } i = j \neq 0\,, \\ \frac{3(b-a)}{16\lambda^5\sigma^2}(\lambda^6 - 6\lambda^4\omega_i^2 + 9\lambda^2\omega_i^4) + G(a) & \text{if } i = j = 0\,, \end{cases} \tag{120}$$

which boils down to

$$\mathbf{K}_{\phi\phi}[i,j] = \begin{cases} G(a) & \text{if } i \neq j\,, \\ \frac{3(b-a)}{32\lambda^5\sigma^2}(\lambda^2 + \omega_i^2)^3 + G(a) & \text{if } i = j \neq 0\,, \\ \frac{3\lambda(b-a)}{16\sigma^2} + G(a) & \text{if } i = j = 0\,. \end{cases} \tag{121}$$

case 2: $i, j > M$ (sine block)

$$\begin{aligned}L_x(\sin(\omega_i .)) &= \lambda^3\sin(\omega_i x) + 3\omega_i\lambda^2\cos(\omega_i x) - 3\lambda\omega_i^2\sin(\omega_i x) - \omega_i^3\cos(\omega_i x) \\ &= (\lambda^3 - 3\lambda\omega_i^2)\sin(\omega_i x) - (\omega_i^3 - 3\omega_i\lambda^2)\cos(\omega_i x)\,, \\ G(a) &= \frac{3\omega_i\omega_j}{\lambda^2\sigma^2}\,,\end{aligned} \tag{122}$$

$$\mathbf{K}_{\phi\phi}[i,j] = \begin{cases} G(a) & \text{if } i \neq j\,, \\ \frac{3(b-a)}{32\lambda^5\sigma^2}(\lambda^2 + \omega_i^2)^3 + G(a) & \text{if } i = j \neq 0\,. \end{cases} \tag{123}$$

case 3: $i \leq M < j$ (off-diagonal block). As previously, calculations give $\mathbf{K}_{\phi\phi}[i,j] = 0$.

References

Martın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from `tensorflow.org`.

Vincent Adam, James Hensman, and Maneesh Sahani. Scalable transformed additive signal decomposition by non-conjugate Gaussian process inference. In IEEE International Workshop on Machine Learning for Signal Processing (MLSP), 2016.

Naum I Akhiezer and Izrail' M Glazman. Theory of Linear Operators in Hilbert Space. Dover, New York, 1993.

Mauricio Alvarez and Neil D Lawrence. Sparse convolved Gaussian processes for multi-output regression. In Advances in Neural Information Processing Systems 21, pages 57–64. Curran Associates, Inc., 2009.

Sudipto Banerjee, Alan E Gelfand, Andrew O Finley, and Huiyan Sang. Gaussian predictive process models for large spatial data sets. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 70(4):825–848, 2008.

Sudipto Banerjee, Bradley P Carlin, and Alan E Gelfand. Hierarchical Modeling and Analysis for Spatial Data. CRC Press, 2014.

Alain Berlinet and Christine Thomas-Agnan. Reproducing Kernel Hilbert Spaces in Probability and Statistics. Kluwer Academic Publishers, 2004.

Matt A Bernstein, Kevin F King, and Xiaohong Joe Zhou. Handbook of MRI Pulse Sequences. Elsevier, 2004.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. arXiv preprint arXiv:1601.00670, 2016.

François-Xavier Briol, Chris J Oates, Mark Girolami, Michael A Osborne, and Dino Sejdinovic. Probabilistic integration: A role for statisticians in numerical analysis? arXiv preprint arXiv:1512.00933, 2016.

Thang D Bui and Richard E Turner. Tree-structured Gaussian process approximations. In Advances in Neural Information Processing Systems 27, pages 2213–2221. Curran Associates, Inc., 2014.

Kian Ming A Chai. Variational multinomial logit Gaussian process. Journal of Machine Learning Research, 13:1745–1808, 2012.

Ole F Christensen, Gareth O Roberts, and Martin Sköld. Robust Markov chain Monte Carlo methods for spatial generalized linear mixed models. Journal of Computational and Graphical Statistics, 15(1):1–17, 2006.

Lehel Csató and Manfred Opper. Sparse on-line Gaussian processes. Neural Computation, 14(3):641–668, 2002.

Lehel Csató, Manfred Opper, and Ole Winther. TAP Gibbs free energy, belief propagation and sparsity. In Advances in Neural Information Processing Systems, pages 657–663, 2002.

Andreas Damianou, Michalis K Titsias, and Neil D Lawrence. Variational inference for latent variables and uncertain inputs in Gaussian processes. Journal of Machine Learning Research, 17:1–62, 2016.

Andreas C Damianou and Neil D Lawrence. Deep Gaussian processes. In International Conference on Artificial Intelligence and Statistics (AISTATS), pages 207–215, 2013.

Marc P Deisenroth and Jun Wei Ng. Distributed Gaussian processes. In International Conference on Machine Learning (ICML), pages 1481–1490, 2015.

Marc P Deisenroth, Dieter Fox, and Carl E Rasmussen. Gaussian processes for data-efficient learning in robotics and control. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(2):408–423, 2015.

Amir Dezfouli and Edwin V Bonilla. Scalable inference for Gaussian process models with black-box likelihoods. In Advances in Neural Information Processing Systems, pages 1414–1422, 2015.

Peter J Diggle. Statistical Analysis of Spatial and Spatio-Temporal Point Patterns. CRC Press, 2013.

Michael F Driscoll. The reproducing kernel Hilbert space structure of the sample paths of a Gaussian process. Probability Theory and Related Fields, 26(4):309–316, 1973.

Nicolas Durrande, David Ginsbourger, and Olivier Roustant. Additive covariance kernels for high-dimensional Gaussian process modeling. In Annales de la Faculté de Sciences de Toulouse, volume 21, pages p–481, 2012.

Nicolas Durrande, James Hensman, Magnus Rattray, and Neil D Lawrence. Detecting periodicities with Gaussian processes. PeerJ Computer Science, 2:e50, 2016.

Alok Dutt and Vladimir Rokhlin. Fast Fourier transforms for nonequispaced data. SIAM Journal on Scientific computing, 14(6):1368–1393, 1993.

David K Duvenaud, Hannes Nickisch, and Carl E Rasmussen. Additive Gaussian processes. In Advances in Neural Information Processing Systems 24, pages 226–234. Curran Associates, Inc., 2011.

Anibal Figueiras-Vidal and Miguel Lázaro-Gredilla. Inter-domain Gaussian processes for sparse inference using inducing features. In Advances in Neural Information Processing Systems 22, pages 1087–1095. Curran Associates, Inc., 2009.

Maurizio Filippone and Raphael Engler. Enabling scalable stochastic gradient-based inference for Gaussian processes by employing the Unbiased LInear System SolvEr (ULISSE). In International Conference on Machine Learning (ICML), pages 6–11, 2015.

Maurizio Filippone, Mingjun Zhong, and Mark Girolami. A comparative evaluation of stochastic-based inference methods for Gaussian process models. Machine Learning, 93 (1):93–114, 2013.

Jochen Fritz, Insa Neuweiler, and Wolfgang Nowak. Application of FFT-based algorithms for large-scale universal kriging problems. Mathematical Geosciences, 41(5):509–533, 2009.

Yarin Gal and Richard Turner. Improving the Gaussian process sparse spectrum approximation by representing uncertainty in frequency inputs. In International Conference on Machine Learning (ICML), pages 655–664, 2015.

Leslie Greengard and June-Yub Lee. Accelerating the nonuniform fast Fourier transform. SIAM Review, 46(3):443–454, 2004.

Philipp Hennig, Michael A Osborne, and Mark Girolami. Probabilistic numerics and uncertainty in computations. In Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, volume 471, page 20150142. The Royal Society, 2015.

James Hensman, Nicolò Fusi, and Neil D Lawrence. Gaussian processes for big data. In Conference on Uncertainty in Artificial Intelligence (UAI), pages 282–290. AUAI Press, 2013.

James Hensman, Alexander G de G Matthews, Maurizio Filippone, and Zoubin Ghahramani. Mcmc for variationally sparse Gaussian processes. In Advances in Neural Information Processing Systems 28, pages 1639–1647. Curran Associates, Inc., 2015a.

James Hensman, Alexander G de G Matthews, and Zoubin Ghahramani. Scalable variational Gaussian process classification. In International Conference on Artificial Intelligence and Statistics (AISTATS), pages 351–360, 2015b.

Emtiyaz Khan, Shakir Mohamed, and Kevin P Murphy. Fast Bayesian inference for non-conjugate Gaussian process regression. In Advances in Neural Information Processing Systems, pages 3140–3148, 2012.

Emtiyaz Khan, Aleksandr Aravkin, Michael Friedlander, and Matthias Seeger. Fast dual variational inference for non-conjugate latent gaussian models. In International Conference on Machine Learning, pages 951–959, 2013.

Neil D Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. Journal of Machine Learning Research, 6:1783–1816, 2005.

Miguel Lázaro-Gredilla, Joaquin Quiñonero-Candela, Carl E Rasmussen, and Aníbal R Figueiras-Vidal. Sparse spectrum Gaussian process regression. Journal of Machine Learning Research, 11:1865–1881, 2010.

Judith Lean. Solar irradiance reconstruction. Data contribution series # 2004-035, IGBP PAGES/World Data Center for Paleoclimatology NOAA/NGDC Paleoclimatology Program, Boulder, CO, USA, 2004.

Chris Lloyd, Tom Gunter, Michael Osborne, and Stephen Roberts. Variational inference for Gaussian process modulated Poisson processes. In International Conference on Machine Learning (ICML), pages 1814–1822, 2015.

Alexander G de G Matthews. Scalable Gaussian Process Inference Using Variational Methods. PhD thesis, University of Cambridge, Cambridge, UK, 2016.

Alexander G de G Matthews, James Hensman, Richard E Turner, and Zoubin Ghahramani. On sparse variational methods and the Kullback-Leibler divergence between stochastic processes. In International Conference on Artificial Intelligence and Statistics (AISTATS), pages 231–238, 2016.

Alexander G de G Matthews, Mark van der Wilk, Tom Nickson, Keisuke Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman. GPflow: A Gaussian process library using TensorFlow. Journal of Machine Learning Research, 18 (40):1–6, 2017.

Jesper Møller, Anne R Syversveen, and Rasmus P Waagepetersen. Log Gaussian cox processes. Scandinavian Journal of Statistics, 25(3):451–482, 1998.

David Moore and Stuart J Russell. Gaussian process random fields. In Advances in Neural Information Processing Systems 28, pages 3357–3365. Curran Associates, Inc., 2015.

Iain Murray and Ryan P Adams. Slice sampling covariance hyperparameters of latent Gaussian models. In Advances in Neural Information Processing Systems 23, pages 1732–1740. Curran Associates, Inc., 2010.

Iain Murray, Ryan P Adams, and David JC MacKay. Elliptical slice sampling. In International Conference on Artificial Intelligence and Statistics (AISTATS), pages 541–548, 2010.

Thomas Nickson, Tom Gunter, Chris Lloyd, Michael A Osborne, and Stephen Roberts. Blitzkriging: Kronecker-structured stochastic Gaussian processes. arXiv preprint arXiv:1510.07965, 2015.

Manfred Opper and Cédric Archambeau. The variational Gaussian approximation revisited. Neural Computation, 21(3):786–792, 2009.

Michael A Osborne. Bayesian Gaussian Processes for Sequential Prediction, Optimisation and Quadrature. PhD thesis, University of Oxford, Oxford, UK, 2010.

Christopher J Paciorek. Bayesian smoothing with Gaussian processes using Fourier basis functions in the spectralGP package. Journal of Statistical Software, 19(2):1–38, 2007.

Joaquin Quiñonero-Candela and Carl E Rasmussen. A unifying view of sparse approximate Gaussian process regression. Journal of Machine Learning Research, 6:1939–1959, 2005.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In Advances in Neural Information Processing Systems 20, pages 1177–1184. Curran Associates, Inc., 2008.

Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In Advances in Neural Information Processing Systems 21, pages 1313–1320. Curran Associates, Inc., 2009.

Barbara Rakitsch, Christoph Lippert, Karsten Borgwardt, and Oliver Stegle. It is all in the noise: Efficient multi-task Gaussian process inference with structured residuals. In Advances in Neural Information Processing Systems 26, pages 1466–1474. Curran Associates, Inc., 2013.

Yunus Saatçi. Scalable Inference for Structured Gaussian Process Models. PhD thesis, University of Cambridge, Cambridge, UK, 2012.

Yves-Laurent Kom Samo and Stephen J Roberts. String and membrane Gaussian processes. Journal of Machine Learning Research, 17:1–87, 2016.

Simo Särkkä, Arno Solin, and Jouni Hartikainen. Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing. IEEE Signal Processing Magazine, 30(4): 51–61, 2013.

Matthias Seeger. Bayesian Gaussian process models: PAC-Bayesian generalisation error bounds and sparse approximations. PhD thesis, University of Edinburgh, Edinburgh, UK, 2003.

Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In Advances in Neural Information Processing Systems 18, pages 1257–1264. MIT Press, 2005.

Arno Solin and Simo Särkkä. Hilbert space methods for reduced-rank Gaussian process regression. arXiv preprint arXiv:1401.5508, 2014.

Arno Solin, Pasi Jylänki, Jaakko Kauramäki, Tom Heskes, Marcel AJ van Gerven, and Simo Särkkä. Regularizing solutions to the MEG inverse problem using space-time separable covariance functions. arXiv preprint arXiv:1604.04931, 2016.

Oliver Stegle, Christoph Lippert, Joris M Mooij, Neil D Lawrence, and Karsten M Borgwardt. Efficient inference in matrix-variate Gaussian models with iid observation noise. In Advances in Neural Information Processing Systems 24, pages 630–638. Curran Associates, Inc., 2011.

Benjamin M Taylor and Peter J Diggle. INLA or MCMC? A tutorial and comparative evaluation for spatial prediction in log-Gaussian Cox processes. Journal of Statistical Computation and Simulation, 84(10):2266–2284, 2014.

Michalis K Titsias. Variational learning of inducing variables in sparse Gaussian processes. In International Conference on Artificial Intelligence and Statistics (AISTATS), pages 567–574, 2009.

Michalis K Titsias and Neil D Lawrence. Bayesian Gaussian process latent variable model. In International Conference on Artificial Intelligence and Statistics (AISTATS), pages 844–851, 2010.

Volker Tresp. A Bayesian committee machine. Neural Computation, 12(11):2719–2741, 2000.

Richard E Turner. Statistical Models for Natural Sounds. PhD thesis, UCL (University College London), London, UK, 2010.

Richard E Turner and Maneesh Sahani. Two problems with variational expectation maximisation for time-series models. In D. Barber, T. Cemgil, and S. Chiappa, editors, Bayesian Time Series Models, chapter 5, pages 109–130. Cambridge University Press, 2011.

Jarno Vanhatalo and Aki Vehtari. Sparse log Gaussian processes via MCMC for spatial epidemiology. In Gaussian Processes in Practice, JMLR: Workshop and Conference Proceedings, pages 73–89, 2007.

Christopher KI Williams and Carl E Rasmussen. Gaussian Processes for Machine Learning. MIT Press, 2006.

Christopher KI Williams and Matthias Seeger. The effect of the input density distribution on kernel-based classifiers. In International Conference on Machine Learning (ICML), 2000.

Christopher KI Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In Advances in Neural Information Processing Systems 13, pages 682–688. MIT Press, 2001.

Andrew G Wilson and Ryan P Adams. Gaussian process kernels for pattern discovery and extrapolation. In International Conference on Machine Learning (ICML), pages 1067–1075, 2013.

Andrew G Wilson and Hannes Nickisch. Kernel interpolation for scalable structured Gaussian processes (KISS-GP). In International Conference on Machine Learning (ICML), pages 1775–1784, 2015.

Andrew G Wilson, Elad Gilboa, John P Cunningham, and Arye Nehorai. Fast kernel learning for multidimensional pattern extrapolation. In Advances in Neural Information Processing Systems 27, pages 3626–3634. Curran Associates, Inc., 2014.

Andrew G Wilson, Christoph Dann, and Hannes Nickisch. Thoughts on massively scalable Gaussian processes. arXiv preprint arXiv:1511.01870, 2015.

Zichao Yang, Andrew G Wilson, Alex Smola, and Le Song. À la carte — Learning fast kernels. In International Conference on Artificial Intelligence and Statistics (AISTATS), pages 1098–1106, 2015.

Journal of Machine Learning Research 18 (2018) 1-6 Submitted 7/17; Revised 10/17; Published 4/18

HyperTools: a Python Toolbox for Gaining Geometric Insights into High-Dimensional Data

Andrew C. Heusser† andrew.c.heusser@dartmouth.edu
Kirsten Ziman† kirsten.k.ziman.gr@dartmouth.edu
Lucy L. W. Owen lucy.w.owen.gr@dartmouth.edu
Jeremy R. Manning jeremy.r.manning@dartmouth.edu
Department of Psychological and Brain Sciences, Dartmouth College, Hanover, NH 03755, USA
†*Denotes equal contribution*

Editor: Alexandre Gramfort

Abstract

Dimensionality reduction algorithms have played a foundational role in facilitating the deep understanding of complex high-dimensional data. One particularly useful application of dimensionality reduction techniques is in data visualization. Low-dimensional visualizations can help practitioners understand where machine learning algorithms might leverage the geometric properties of a dataset to improve performance. Another challenge is to generalize insights across datasets [e.g. data from multiple modalities describing the same system (Haxby et al., 2011), artwork or photographs of similar content in different styles (Zhu et al., 2017), etc.]. Several recently developed techniques (e.g. Haxby et al., 2011; Chen et al., 2015) use the procrustean transformation (Schönemann, 1966) to align the geometries of two or more spaces so that data with different axes may be plotted in a common space. We propose that each of these techniques (dimensionality reduction, alignment, and visualization) applied in sequence should be cast as a single conceptual *hyperplot* operation for gaining geometric insights into high-dimensional data. Our Python toolbox enables this operation in a single (highly flexible) function call.

Keywords: visualization, high-dimensional, dimensionality reduction, procrustes, time-series data

1. Introduction

A major focus of machine learning research is finding patterns in, and making predictions from, complex data that might be otherwise inscrutable to a human viewer. Paradoxically, designing effective algorithms for discovering, characterizing, and leveraging structure in many complex datasets often requires the practitioner to have some initial intuitions about the structure of the data. One commonly used approach for discovering structure in high-dimensional data is to use dimensionality reduction algorithms (e.g. Pearson, 1901; Tipping and Bishop, 1999; Jutten and Herault, 1991; Comon et al., 1991; Torgerson, 1958; van der Maaten and Hinton, 2008) to visualize high-dimensional data in two or three dimensions, thereby providing insights into geometric patterns that pervade the data (Uddenberg et al., 2016). Despite the so-called "curse of dimensionality," whereby high-dimensional data can behave differently from low-dimensional data, visualizing data via low-dimensional projections can provide a glimpse (though imperfect) into what the dataset "looks like." By leveraging different dimensionality reduction algorithms, or by viewing the data through different projections (or as an animation), one can begin to form intuitions that generalize beyond the specific imperfections of a given dimensionality reduction tool or projection. We highlight several examples of how low-dimensional projections may be used to understand the geometry of high-dimensional data in Figure 1. Complete details may be found here, and sample IPython notebooks may be found here.

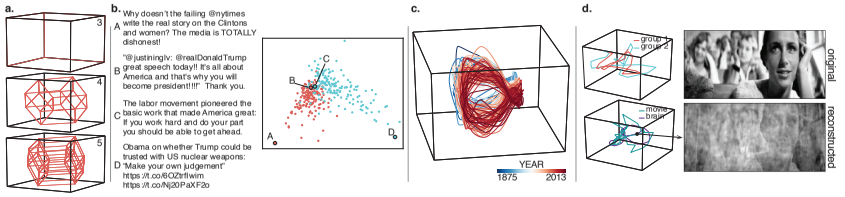


Figure 1: **Data visualization examples. a.** Hypercubes of increasing dimensionality (3, 4 and 5 dimensions). The cubes provide some insights into which aspects of high-dimensional data are preserved (or distorted) when projected onto 3 dimensions. **b.** Topic modeling (Blei et al., 2003) of political Twitter data: 2D projections of topic vectors of Hillary Clinton's (blue) and Donald Trump's (red) tweets during their 2016 presidential campaigns (link to data). The 'V' shape highlights that Trump and Clinton tweets are largely about fundamentally different topics. The highlighted examples include (A) a Trump tweet that is especially "Trump-like," (B) a Trump tweet that is Clinton-like, (C) a Clinton tweet that is Trump-like, and (D) a Clinton tweet that is especially Clinton-like. **c.** Changing temperatures across the Earth's surface from 1875—2013 (link to data). The visualization highlights the cyclical (seasonal) nature of global temperatures that occurs alongside a gradual increase in global temperatures over time. **d.** Brain/movie trajectories during movie viewing (link to data). (Top left) Group-averaged trajectories of brain activity from the ventral visual cortex, split into two randomly-selected groups of subjects (group 1: $n = 6$, group 2: $n = 5$) watching *Raiders of the Lost Ark* (Haxby et al., 2011). (Bottom left) Group-averaged trajectory of brain activity from ventral visual cortex and trajectory of the movie frames (pixel intensities over time), hyperaligned to a common space. (Right) Movie frame reconstructed from the group-averaged brain activity that was aligned to movie space. The example illustrates geometric commonalities across brains, and between the movie frames and the brain responses to those frames.

Although dimensionality reduction algorithms provide a useful means of visualizing high-dimensional data, they cannot (by themselves) solve the problem of identifying geometric similarities across different types of data. For example, as highlighted in Figure 1d, different people's brains might exhibit similar (but not identical) activity patterns while watching a movie, and those brain patterns might exhibit a similar temporal covariance structure to

the movie itself. However, if one were to project brain data and movie data onto a three dimensional space, although the overall "shapes" of those datasets might be similar, there would be no inherent reason for the datasets to align. Algorithms inspired by the procrustean transformation (Schönemann, 1966), including Hyperalignment (Haxby et al., 2011) and the Shared Response Model (Chen et al., 2015) compute the affine transformations of two or more datasets that bring the data into a common alignment. This can reveal similarities (or differences) between the underlying geometries of otherwise incompatible data. Further, because the transformation is invertible, one can map between any point in that common space (e.g. a shared movie-brain space) and the original data spaces (e.g. the movie frames; the lower right panel of Fig. 1d shows a movie frame corresponding to a brain pattern recorded during one time in the movie, and the upper panel shows the original movie frame viewed at that moment).

The `HyperTools` toolbox (current version: 0.4.2) provides a powerful set of Python functions for projecting high dimensional data onto lower-dimensional spaces, aligning data of different types, and visualizing the results in publication-quality figures and movies. A central goal of the toolbox is to cast dimensionality reduction, alignment, and visualization as a single highly flexible "hyperplot" command:

```
import hypertools as hyp                                              (1)
hyp.plot(list_of_arrays, reduce='TSNE', align='hyper', ndims=3)       (2)
```

This example function call projects the high-dimensional data onto 3 dimensions using the t-SNE algorithm, aligns the data matrices in the given list of arrays into a common space using hyperalignment, and produces a 3D plot analogous to those shown in Figure 1. These hyperplot visualizations provide an intuitive means of understanding how different observations relate to each other or how those observations change over time. These insights can guide algorithm design decisions, or help practitioners to understand which aspects of the data may be easy (or difficult) to model or measure.

2. Overview of the Toolbox

`HyperTools` is open-source, installable from GitHub or pip (`pip install hypertools`), and is distributed with the MIT License. The toolbox depends on the following open-source software packages: `Matplotlib` (Hunter, 2007) for plotting functionality, `Seaborn` (Waskom et al., 2016) for plot styling, `Scikit-learn` (Pedregosa et al., 2011) for data analysis (dimensionality reduction, clustering, etc.), and `PPCA` for inferring missing data (Tipping and Bishop, 1999). The toolbox also includes a port of the hyperalignment algorithm (Haxby et al., 2011) from the `PyMVPA` library, as well as the shared response model from the `BrainIAK` toolbox (Capota et al., 2017), as an alternative data alignment technique. At the time of writing this manuscript, the toolbox incorporates two general classes of algorithms: *dimensionality reduction* (PCA, incremental PCA, sparse PCA, kernel PCA, probabilistic PCA, t-SNE, MDS, ICA, factor analysis, truncated SVD, dictionary learning, mini-batch dictionary learning, isomap, spectral embedding, and local linear embedding) and *alignment* (hyperalignment, shared response model, and procrustean transformation). `HyperTools` provides a simple interface to these functions, with support for a variety of convenient data formats including `NumPy` arrays (van der Walt et al., 2011), `Pandas` dataframes (McKinney, 2010), or

mixed lists of arrays and dataframes. [Each to-be-analyzed (or to-be visualized) dataset must be formated as a number-of-observations (S) by number-of-features (F) array or dataframe.] `HyperTools` also adds a number of custom arguments to facilitate data visualization and manipulation of high-dimensional data. Nearly all of the `HyperTools` functions may be accessed through the `plot` function (Ex. 2). This design enables complex data analysis and plotting to be carried out in a single function call. The same function call also returns the analyzed data (and the transformations applied to it) for potential follow-up analyses. There are two general types of plots supported by the toolbox: *static plots* and *animated plots*.

2.1 Static Plots

By default, the `plot` function will perform dimensionality reduction (using incremental PCA), converting the $S \times F$ data matrix (or matrices) into an $S \times 3$ matrix (or matrices), and then create an interactive 3D line plot that can be explored by using the mouse to rotate the plot. If there are `NaNs` present in the dataset, these missing values will be automatically interpolated using PPCA (Tipping and Bishop, 1999) (preserving the dimensionality of the original data) prior to reducing the dimensionality of the data using the user-specified algorithm. The `plot` function also accepts format strings to specify line styling (following `Matplotlib` conventions):

```
hyp.plot(data)                                                        (3)
hyp.plot([data1, data2, data3], '.')                                  (4)
```

2.2 Animated Trajectory Plots

Animated 3D plots (created using the `animate` flag) are useful for visualizing high-dimensional timeseries data:

```
hyp.plot(data, animate=True).                                         (5)
```

This function call creates a 3D trajectory animated over the rows of the `data` matrix. Each frame of the animation displays a portion of the total data trajectory enclosed in a cube. In successive frames, the displayed portion of the data trajectory incrementally advances, and the camera angle rotates around the cube, providing visual access to different aspects of the data as the animation progresses.

2.3 Align

Two or more datasets may share geometric structure, but reside in different coordinate systems. The `align` function accepts a list of arrays as input and returns a hyperaligned list of arrays in a common geometric space:

```
hyp.plot([array1, array2, array3], align='hyper')                     (6)
hyperaligned_list = hyp.align([array1, array2, array3])               (7)
```

2.4 Reduce

Users can access a variety of dimensionality reduction algorithms by passing the `reduce` keyword argument to the `plot` function. We also provide API access to the `reduce`

function that underlies these transformations. At its core, the `reduce` function wraps many of `scikit-learn`'s dimensionality reduction algorithms, along with the `PPCA` library. `HyperTools` extends the functionality of these tools by providing a streamlined syntax that accepts data matrices in a larger variety of formats (e.g. `NumPy` arrays and `Pandas` dataframes, or lists of arrays and dataframes, may all be analyzed using the same syntax). The function may be used as follows:

```
hyp.plot(data, reduce='MDS')                                    (8)
reduced_data = hyp.reduce(data, reduce='TSNE', ndims=5)         (9)
```

3. Concluding Remarks

The `HyperTools` toolbox is designed to provide geometric insights into high-dimensional datasets with a single function call. We also provide a convenient syntax to access a variety of dimensionality reduction and data alignment algorithms. The above overview of the toolbox highlights its primary features; additional functionality and a detailed description of the API may be found here.

Acknowledgments

We are grateful to Luke J. Chang and Matthijs van der Meer for useful discussions, and to J. Swaroop Guntapalli for help implementing our `align` function. We also thank Ryan Arredondo, Aly Sivji and Steph Wright for their code contributions during the Mozilla Global Sprint in June 2017, and we thank Mozilla for organizing the Global Sprint. We also thank Chase Williams for contributing code. Our work was supported in part by NSF EPSCoR Award Number 1632738. The content is solely the responsibility of the authors and does not necessarily represent the official views of our supporting organizations.

References

D M Blei, A Y Ng, and M I Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993 – 1022, 2003.

M Capota, J Turek, P-H Chen, X Zhu, J R Manning, N Sundaram, B Keller, Y Wang, and Y S Shin. Brain imaging analysis kit, 2017. URL `https://doi.org/10.5281/zenodo.59780`.

P-H Chen, J Chen, Y Yeshurun, U Hasson, J Haxby, and P J Ramadge. A Reduced-Dimension fMRI Shared Response Model. In C. Cortes and N. D. Lawrence and D. D. Lee and M. Sugiyama and R. Garnett, editor, *Advances in Neural Information Processing Systems 28*, pages 460–468. Curran Associates, Inc., 2015.

P Comon, C Jutten, and J Herault. Blind separation of sources, part II: Problems statement. *Signal Processing*, 24(1):11 – 20, 1991.

J V Haxby, J S Guntupalli, A C Connolly, Y O Halchenko, B R Conroy, M I Gobbini, M Hanke, and P J Ramadge. A common, high-dimensional model of the representational space in human ventral temporal cortex. *Neuron*, 72:404–416, 2011.

J D Hunter. Matplotlib: A 2D graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.

C Jutten and J Herault. Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24(1):1–10, 1991.

W McKinney. Data structures for statistical computing in Python. In *Proceedings of the 9th Python in Science Conference*, pages 51–56, 2010.

K Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2:559–572, 1901.

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

P Schönemann. A generalized solution of the orthogonal Procrustes problem. *Psychometrika*, 31:1–10, 1966.

M E Tipping and C M Bishop. Probabilistic principal component analysis. *Journal of Royal Statistical Society, Series B*, 61(3):611–622, 1999.

W S Torgerson. *Theory and methods of scaling*. Wiley, New York, 1958.

S Uddenberg, G Newman, and B Scholl. Perceptual averaging of scientific data: Implications of ensemble representations for the perception of patterns in graphs. *Journal of Vision*, 16(12):1081, 2016.

L J P van der Maaten and G E Hinton. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

S van der Walt, S C Colbert, and G Varoquaux. The NumPy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, 13:22–30, 2011.

M Waskom, O Botvinnik, D Okane, P Hobson, David, Y Halchenko, S Lukauskas, J B Cole, J Warmenhoven, J de Ruiter, S Hoyer, J Vanderplas, S Villalba, G Kunter, E Quintero, M Martin, A Miles, K Meyer, T Augspurger, T Yarkoni, P Bachant, M Williams, C Evans, C Fitzgerald, Brian, D Wehner, G Hitz, E Ziegler, A Qalieh, and A Lee. Seaborn: v0.7.1, 2016. URL `https://doi.org/10.5281/zenodo.54844`.

J-Y Zhu, T Park, P Isola, and A A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv*, 1703.10593, 2017.

Journal of Machine Learning Research 18 (2018) 1-43 Submitted 8/17; Published 4/18

Automatic Differentiation in Machine Learning: a Survey

Atılım Güneş Baydin GUNES@ROBOTS.OX.AC.UK
Department of Engineering Science
University of Oxford
Oxford OX1 3PJ, United Kingdom

Barak A. Pearlmutter BARAK@PEARLMUTTER.NET
Department of Computer Science
National University of Ireland Maynooth
Maynooth, Co. Kildare, Ireland

Alexey Andreyevich Radul AXCH@MIT.EDU
Department of Brain and Cognitive Sciences
Massachusetts Institute of Technology
Cambridge, MA 02139, United States

Jeffrey Mark Siskind QOBI@PURDUE.EDU
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907, United States

Editor: Léon Bottou

Abstract

Derivatives, mostly in the form of gradients and Hessians, are ubiquitous in machine learning. Automatic differentiation (AD), also called algorithmic differentiation or simply "autodiff", is a family of techniques similar to but more general than backpropagation for efficiently and accurately evaluating derivatives of numeric functions expressed as computer programs. AD is a small but established field with applications in areas including computational fluid dynamics, atmospheric sciences, and engineering design optimization. Until very recently, the fields of machine learning and AD have largely been unaware of each other and, in some cases, have independently discovered each other's results. Despite its relevance, general-purpose AD has been missing from the machine learning toolbox, a situation slowly changing with its ongoing adoption under the names "dynamic computational graphs" and "differentiable programming". We survey the intersection of AD and machine learning, cover applications where AD has direct relevance, and address the main implementation techniques. By precisely defining the main differentiation techniques and their interrelationships, we aim to bring clarity to the usage of the terms "autodiff", "automatic differentiation", and "symbolic differentiation" as these are encountered more and more in machine learning settings.

Keywords: Backpropagation, Differentiable Programming

1. Introduction

Methods for the computation of derivatives in computer programs can be classified into four categories: (1) manually working out derivatives and coding them; (2) *numerical differentiation* using finite difference approximations; (3) *symbolic differentiation* using expression manipulation in computer algebra systems such as Mathematica, Maxima, and Maple; and (4) *automatic differentiation*, also called *algorithmic differentiation*, which is the subject matter of this paper.

Conventionally, many methods in machine learning have required the evaluation of derivatives and most of the traditional learning algorithms have relied on the computation of gradients and Hessians of an objective function (Sra et al., 2011). When introducing new models, machine learning researchers have spent considerable effort on the manual derivation of analytical derivatives to subsequently plug these into standard optimization procedures such as L-BFGS (Zhu et al., 1997) or stochastic gradient descent (Bottou, 1998). Manual differentiation is time consuming and prone to error. Of the other alternatives, numerical differentiation is simple to implement but can be highly inaccurate due to round-off and truncation errors (Jerrell, 1997); more importantly, it scales poorly for gradients, rendering it inappropriate for machine learning where gradients with respect to millions of parameters are commonly needed. Symbolic differentiation addresses the weaknesses of both the manual and numerical methods, but often results in complex and cryptic expressions plagued with the problem of "expression swell" (Corliss, 1988). Furthermore, manual and symbolic methods require models to be defined as closed-form expressions, ruling out or severely limiting algorithmic control flow and expressivity.

We are concerned with the powerful fourth technique, automatic differentiation (AD). AD performs a non-standard interpretation of a given computer program by replacing the domain of the variables to incorporate derivative values and redefining the semantics of the operators to propagate derivatives per the chain rule of differential calculus. Despite its widespread use in other fields, general-purpose AD has been underused by the machine learning community until very recently.[1] Following the emergence of deep learning (LeCun et al., 2015; Goodfellow et al., 2016) as the state-of-the-art in many machine learning tasks and the modern workflow based on rapid prototyping and code reuse in frameworks such as Theano (Bastien et al., 2012), Torch (Collobert et al., 2011), and TensorFlow (Abadi et al., 2016), the situation is slowly changing where projects such as autograd[2] (Maclaurin, 2016), Chainer[3] (Tokui et al., 2015), and PyTorch[4] (Paszke et al., 2017) are leading the way in bringing general-purpose AD to the mainstream.

The term "automatic" in AD can be a source of confusion, causing machine learning practitioners to put the label "automatic differentiation", or just "autodiff", on any method or tool that does not involve manual differentiation, without giving due attention to the underlying mechanism. We would like to stress that AD as a technical term refers to a specific family of techniques that compute derivatives through accumulation of values during code execution to generate numerical derivative evaluations rather than derivative

1. See, e.g., https://justindomke.wordpress.com/2009/02/17/automatic-differentiation-the-most-criminally-underused-tool-in-the-potential-machine-learning-toolbox/
2. https://github.com/HIPS/autograd
3. https://chainer.org/
4. http://pytorch.org/

expressions. This allows accurate evaluation of derivatives at machine precision with only a small constant factor of overhead and ideal asymptotic efficiency. In contrast with the effort involved in arranging code as closed-form expressions under the syntactic and semantic constraints of symbolic differentiation, AD can be applied to regular code with minimal change, allowing branching, loops, and recursion. Because of this generality, AD has been applied to computer simulations in industry and academia and found applications in fields including engineering design optimization (Forth and Evans, 2002; Casanova et al., 2002), computational fluid dynamics (Müller and Cusdin, 2005; Thomas et al., 2006; Bischof et al., 2006), physical modeling (Ekström et al., 2010), optimal control (Walther, 2007), structural mechanics (Haase et al., 2002), atmospheric sciences (Carmichael and Sandu, 1997; Charpentier and Ghemires, 2000), and computational finance (Bischof et al., 2002; Capriotti, 2011).

In machine learning, a specialized counterpart of AD known as the backpropagation algorithm has been the mainstay for training neural networks, with a colorful history of having been reinvented at various times by independent researchers (Griewank, 2012; Schmidhuber, 2015). It has been one of the most studied and used training algorithms since the day it became popular mainly through the work of Rumelhart et al. (1986). In simplest terms, backpropagation models learning as gradient descent in neural network weight space, looking for the minima of an objective function. The required gradient is obtained by the backward propagation of the sensitivity of the objective value at the output (Figure 1), utilizing the chain rule to compute partial derivatives of the objective with respect to each weight. The resulting algorithm is essentially equivalent to transforming the network evaluation function composed with the objective function under reverse mode AD, which, as we shall see, actually generalizes the backpropagation idea. Thus, a modest understanding of the mathematics underlying backpropagation provides one with sufficient background for grasping AD techniques.

In this paper we review AD from a machine learning perspective, covering its origins, applications in machine learning, and methods of implementation. Along the way, we also aim to dispel some misconceptions that we believe have impeded wider recognition of AD by the machine learning community. In Section 2 we start by explicating how AD differs from numerical and symbolic differentiation. Section 3 gives an introduction to the AD technique and its forward and reverse accumulation modes. Section 4 discusses the role of derivatives in machine learning and examines cases where AD has relevance. Section 5 covers various implementation approaches and general-purpose AD tools, followed by Section 6 where we discuss future directions.

2. What AD Is Not

Without proper introduction, one might assume that AD is either a type of numerical or symbolic differentiation. Confusion can arise because AD does in fact provide numerical values of derivatives (as opposed to derivative expressions) and it does so by using symbolic rules of differentiation (but keeping track of derivative values as opposed to the resulting expressions), giving it a two-sided nature that is partly symbolic and partly numerical (Griewank, 2003). We start by emphasizing how AD is different from, and in several aspects superior to, these two commonly encountered techniques of computing derivatives.

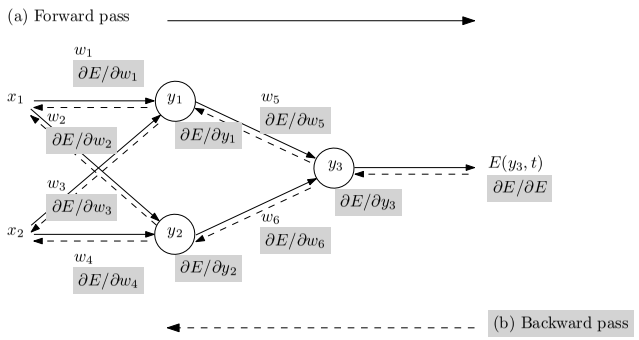


Figure 1: Overview of backpropagation. (a) Training inputs x_i are fed forward, generating corresponding activations y_i. An error E between the actual output y_3 and the target output t is computed. (b) The error adjoint is propagated backward, giving the gradient with respect to the weights $\nabla_{w_i} E = \left(\frac{\partial E}{\partial w_1}, \ldots, \frac{\partial E}{\partial w_6}\right)$, which is subsequently used in a gradient-descent procedure. The gradient with respect to inputs $\nabla_{x_i} E$ can be also computed in the same backward pass.

2.1 AD Is Not Numerical Differentiation

Numerical differentiation is the finite difference approximation of derivatives using values of the original function evaluated at some sample points (Burden and Faires, 2001) (Figure 2, lower right). In its simplest form, it is based on the limit definition of a derivative. For example, for a multivariate function $f : \mathbb{R}^n \to \mathbb{R}$, one can approximate the gradient $\nabla f = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n}\right)$ using

$$\frac{\partial f(\mathbf{x})}{\partial x_i} \approx \frac{f(\mathbf{x} + h\mathbf{e}_i) - f(\mathbf{x})}{h} , \tag{1}$$

where $\mathbf{e}_i$ is the i-th unit vector and $h > 0$ is a small step size. This has the advantage of being uncomplicated to implement, but the disadvantages of performing $O(n)$ evaluations of f for a gradient in n dimensions and requiring careful consideration in selecting the step size h.

Numerical approximations of derivatives are inherently ill-conditioned and unstable,[5] with the exception of complex variable methods that are applicable to a limited set of holomorphic functions (Fornberg, 1981). This is due to the introduction of truncation[6] and

5. Using the limit definition of the derivative for finite difference approximation commits both cardinal sins of numerical analysis: *"thou shalt not add small numbers to big numbers"*, and *"thou shalt not subtract numbers which are approximately equal"*.
6. Truncation error is the error of approximation, or inaccuracy, one gets from h not actually being zero. It is proportional to a power of h.

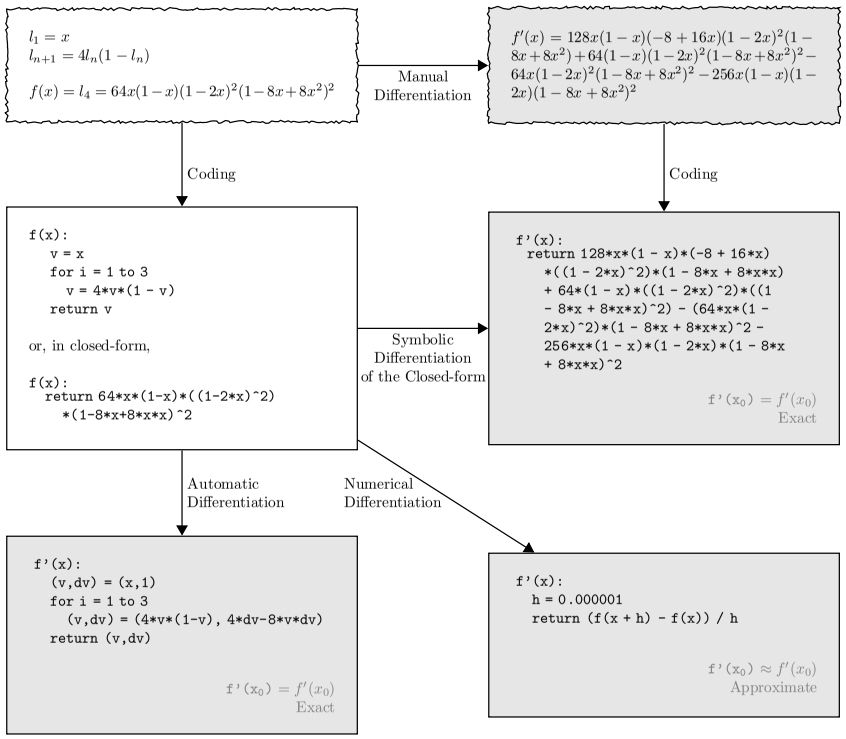


Figure 2: The range of approaches for differentiating mathematical expressions and computer code, looking at the example of a truncated logistic map (upper left). Symbolic differentiation (center right) gives exact results but requires closed-form input and suffers from expression swell; numerical differentiation (lower right) has problems of accuracy due to round-off and truncation errors; automatic differentiation (lower left) is as accurate as symbolic differentiation with only a constant factor of overhead and support for control flow.

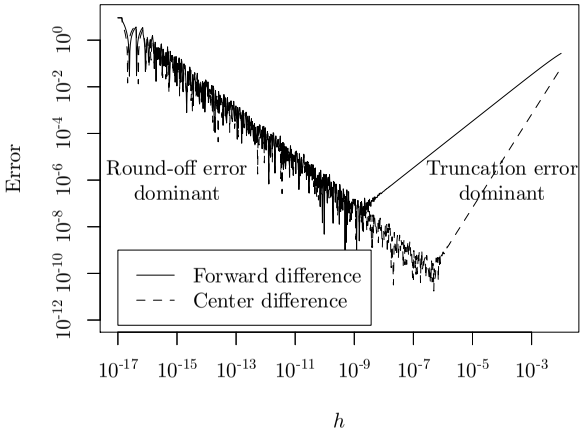


Figure 3: Error in the forward (Eq. 1) and center difference (Eq. 2) approximations as a function of step size h, for the derivative of the truncated logistic map $f(x) = 64x(1-x)(1-2x)^2(1-8x+8x^2)^2$. Plotted errors are computed using $E_{\text{forward}}(h, x_0) = \left|\frac{f(x_0+h)-f(x_0)}{h} - \frac{d}{dx}f(x)\big|_{x_0}\right|$ and $E_{\text{center}}(h, x_0) = \left|\frac{f(x_0+h)-f(x_0-h)}{2h} - \frac{d}{dx}f(x)\big|_{x_0}\right|$ at $x_0 = 0.2$.

round-off[7] errors inflicted by the limited precision of computations and the chosen value of the step size h. Truncation error tends to zero as $h \to 0$. However, as h is decreased, round-off error increases and becomes dominant (Figure 3).

Various techniques have been developed to mitigate approximation errors in numerical differentiation, such as using a center difference approximation

$$\frac{\partial f(\mathbf{x})}{\partial x_i} = \frac{f(\mathbf{x} + h\mathbf{e}_i) - f(\mathbf{x} - h\mathbf{e}_i)}{2h} + O(h^2) \,, \tag{2}$$

where the first-order errors cancel and one effectively moves the truncation error from first-order to second-order in h.[8] For the one-dimensional case, it is just as costly to compute the forward difference (Eq. 1) and the center difference (Eq. 2), requiring only two evaluations of f. However, with increasing dimensionality, a trade-off between accuracy and performance is faced, where computing a Jacobian matrix of a function $f : \mathbb{R}^n \to \mathbb{R}^m$ requires $2mn$ evaluations.

7. Round-off error is the inaccuracy one gets from valuable low-order bits of the final answer having to compete for machine-word space with high-order bits of $f(\mathbf{x}+h\mathbf{e}_i)$ and $f(\mathbf{x})$ (Eq. 1), which the computer has to store just until they cancel in the subtraction at the end. Round-off error is inversely proportional to a power of h.
8. This does not avoid either of the cardinal sins, and is still highly inaccurate due to truncation.

Other techniques for improving numerical differentiation, including higher-order finite differences, Richardson extrapolation to the limit (Brezinski and Zaglia, 1991), and differential quadrature methods using weighted sums (Bert and Malik, 1996), have increased computational complexity, do not completely eliminate approximation errors, and remain highly susceptible to floating point truncation.

The $O(n)$ complexity of numerical differentiation for a gradient in n dimensions is the main obstacle to its usefulness in machine learning, where n can be as large as millions or billions in state-of-the-art deep learning models (Shazeer et al., 2017). In contrast, approximation errors would be tolerated in a deep learning setting thanks to the well-documented error resiliency of neural network architectures (Gupta et al., 2015).

2.2 AD Is Not Symbolic Differentiation

Symbolic differentiation is the automatic manipulation of expressions for obtaining derivative expressions (Grabmeier and Kaltofen, 2003) (Figure 2, center right), carried out by applying transformations representing rules of differentiation such as

$$\begin{aligned} \frac{d}{dx}\left(f(x)+g(x)\right) &\rightsquigarrow \frac{d}{dx}f(x)+\frac{d}{dx}g(x) \\ \frac{d}{dx}\left(f(x)\,g(x)\right) &\rightsquigarrow \left(\frac{d}{dx}f(x)\right)g(x)+f(x)\left(\frac{d}{dx}g(x)\right)\ . \end{aligned} \tag{3}$$

When formulae are represented as data structures, symbolically differentiating an expression tree is a perfectly mechanistic process, considered subject to mechanical automation even at the very inception of calculus (Leibniz, 1685). This is realized in modern computer algebra systems such as Mathematica, Maxima, and Maple and machine learning frameworks such as Theano.

In optimization, symbolic derivatives can give valuable insight into the structure of the problem domain and, in some cases, produce analytical solutions of extrema (e.g., solving for $\frac{d}{dx}f(x)=0$) that can eliminate the need for derivative calculation altogether. On the other hand, symbolic derivatives do not lend themselves to efficient runtime calculation of derivative values, as they can get exponentially larger than the expression whose derivative they represent.

Consider a function $h(x)=f(x)g(x)$ and the multiplication rule in Eq. 3. Since h is a product, $h(x)$ and $\frac{d}{dx}h(x)$ have some common components, namely $f(x)$ and $g(x)$. Note also that on the right hand side, $f(x)$ and $\frac{d}{dx}f(x)$ appear separately. If we just proceeded to symbolically differentiate $f(x)$ and plugged its derivative into the appropriate place, we would have nested duplications of any computation that appears in common between $f(x)$ and $\frac{d}{dx}f(x)$. Hence, careless symbolic differentiation can easily produce exponentially large symbolic expressions which take correspondingly long to evaluate. This problem is known as *expression swell* (Table 1).

When we are concerned with the accurate numerical evaluation of derivatives and not so much with their actual symbolic form, it is in principle possible to significantly simplify computations by storing only the values of intermediate sub-expressions in memory. Moreover, for further efficiency, we can interleave as much as possible the differentiation and simplification steps. This interleaving idea forms the basis of AD and provides an account

Table 1: Iterations of the logistic map $l_{n+1}=4l_n(1-l_n)$, $l_1=x$ and the corresponding derivatives of l_n with respect to x, illustrating expression swell.

n	l_n	$\frac{d}{dx}l_n$	$\frac{d}{dx}l_n$ (Simplified form)
1	x	1	1
2	$4x(1-x)$	$4(1-x)-4x$	$4-8x$
3	$16x(1-x)(1-2x)^2$	$16(1-x)(1-2x)^2-16x(1-2x)^2-64x(1-x)(1-2x)$	$16(1-10x+24x^2-16x^3)$
4	$64x(1-x)(1-2x)^2(1-8x+8x^2)^2$	$128x(1-x)(-8+16x)(1-2x)^2(1-8x+8x^2)+64(1-x)(1-2x)^2(1-8x+8x^2)^2-64x(1-2x)^2(1-8x+8x^2)^2-256x(1-x)(1-2x)(1-8x+8x^2)^2$	$64(1-42x+504x^2-2640x^3+7040x^4-9984x^5+7168x^6-2048x^7)$

of its simplest form: *apply symbolic differentiation at the elementary operation level and keep intermediate numerical results, in lockstep with the evaluation of the main function.* This is AD in the forward accumulation mode, which we shall introduce in the following section.

3. AD and Its Main Modes

AD can be thought of as performing a non-standard interpretation of a computer program where this interpretation involves augmenting the standard computation with the calculation of various derivatives. All numerical computations are ultimately compositions of a finite set of elementary operations for which derivatives are known (Verma, 2000; Griewank and Walther, 2008), and combining the derivatives of the constituent operations through the chain rule gives the derivative of the overall composition. Usually these elementary operations include the binary arithmetic operations, the unary sign switch, and transcendental functions such as the exponential, the logarithm, and the trigonometric functions.

On the left hand side of Table 2 we see the representation of the computation $y=f(x_1,x_2)=\ln(x_1)+x_1x_2-\sin(x_2)$ as an *evaluation trace* of elementary operations—also called a Wengert list (Wengert, 1964). We adopt the three-part notation used by Griewank and Walther (2008), where a function $f:\mathbb{R}^n\to\mathbb{R}^m$ is constructed using intermediate variables v_i such that

- variables $v_{i-n}=x_i,\ i=1,\ldots,n$ are the input variables,
- variables $v_i\ i=1,\ldots,l$ are the working (intermediate) variables, and
- variables $y_{m-i}=v_{l-i},\ i=m-1,\ldots,0$ are the output variables.

Figure 4 shows the given trace of elementary operations represented as a computational graph (Bauer, 1974), useful in visualizing dependency relations between intermediate variables.

Evaluation traces form the basis of the AD techniques. An important point to note here is that AD can differentiate not only closed-form expressions in the classical sense, but also algorithms making use of control flow such as branching, loops, recursion, and procedure

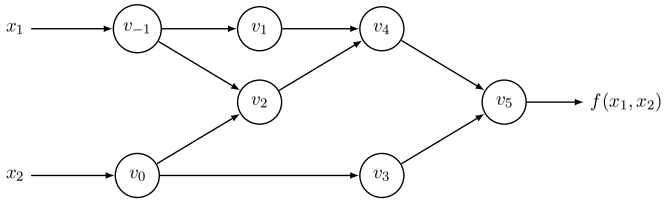


Figure 4: Computational graph of the example $f(x_1, x_2) = \ln(x_1) + x_1 x_2 - \sin(x_2)$. See the primal trace in Tables 2 or 3 for the definitions of the intermediate variables $v_{-1} \dots v_5$.

calls, giving it an important advantage over symbolic differentiation which severely limits such expressivity. This is thanks to the fact that any numeric code will eventually result in a numeric evaluation trace with particular values of the input, intermediate, and output variables, which are the only things one needs to know for computing derivatives using chain rule composition, regardless of the specific control flow path that was taken during execution. Another way of expressing this is that AD is blind with respect to any operation, including control flow statements, which do not directly alter numeric values.

3.1 Forward Mode

AD in forward accumulation mode[9] is the conceptually most simple type. Consider the evaluation trace of the function $f(x_1, x_2) = \ln(x_1) + x_1 x_2 - \sin(x_2)$ given on the left-hand side in Table 2 and in graph form in Figure 4. For computing the derivative of f with respect to x_1, we start by associating with each intermediate variable v_i a derivative

$$\dot{v}_i = \frac{\partial v_i}{\partial x_1} .$$

Applying the chain rule to each elementary operation in the forward primal trace, we generate the corresponding tangent (derivative) trace, given on the right-hand side in Table 2. Evaluating the primals v_i in lockstep with their corresponding tangents $\dot{v}_i$ gives us the required derivative in the final variable $\dot{v}_5 = \frac{\partial y}{\partial x_1}$.

This generalizes naturally to computing the Jacobian of a function $f : \mathbb{R}^n \to \mathbb{R}^m$ with n independent (input) variables x_i and m dependent (output) variables y_j. In this case, each forward pass of AD is initialized by setting only one of the variables $\dot{x}_i = 1$ and setting the rest to zero (in other words, setting $\dot{\mathbf{x}} = \mathbf{e}_i$, where $\mathbf{e}_i$ is the i-th unit vector). A run of the code with specific input values $\mathbf{x} = \mathbf{a}$ then computes

$$\dot{y}_j = \left.\frac{\partial y_j}{\partial x_i}\right|_{\mathbf{x}=\mathbf{a}} , \; j = 1, \dots, m ,$$

9. Also called *tangent linear* mode.

Table 2: Forward mode AD example, with $y = f(x_1, x_2) = \ln(x_1) + x_1 x_2 - \sin(x_2)$ evaluated at $(x_1, x_2) = (2, 5)$ and setting $\dot{x}_1 = 1$ to compute $\frac{\partial y}{\partial x_1}$. The original forward evaluation of the primals on the left is augmented by the tangent operations on the right, where each line complements the original directly to its left.

Forward Primal Trace			Forward Tangent (Derivative) Trace		
v_{-1}	$= x_1$	$= 2$	$\dot{v}_{-1}$	$= \dot{x}_1$	$= 1$
v_0	$= x_2$	$= 5$	$\dot{v}_0$	$= \dot{x}_2$	$= 0$
v_1	$= \ln v_{-1}$	$= \ln 2$	$\dot{v}_1$	$= \dot{v}_{-1}/v_{-1}$	$= 1/2$
v_2	$= v_{-1} \times v_0$	$= 2 \times 5$	$\dot{v}_2$	$= \dot{v}_{-1} \times v_0 + \dot{v}_0 \times v_{-1}$	$= 1 \times 5 + 0 \times 2$
v_3	$= \sin v_0$	$= \sin 5$	$\dot{v}_3$	$= \dot{v}_0 \times \cos v_0$	$= 0 \times \cos 5$
v_4	$= v_1 + v_2$	$= 0.693 + 10$	$\dot{v}_4$	$= \dot{v}_1 + \dot{v}_2$	$= 0.5 + 5$
v_5	$= v_4 - v_3$	$= 10.693 + 0.959$	$\dot{v}_5$	$= \dot{v}_4 - \dot{v}_3$	$= 5.5 - 0$
y	$= v_5$	$= 11.652$	$\dot{y}$	$= \dot{v}_5$	$= 5.5$

giving us one column of the Jacobian matrix

$$\mathbf{J}_f = \left.\begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}\right|_{\mathbf{x}=\mathbf{a}}$$

evaluated at point $\mathbf{a}$. Thus, the full Jacobian can be computed in n evaluations.

Furthermore, forward mode AD provides a very efficient and matrix-free way of computing Jacobian–vector products

$$\mathbf{J}_f \mathbf{r} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix} \begin{bmatrix} r_1 \\ \vdots \\ r_n \end{bmatrix} , \tag{4}$$

simply by initializing with $\dot{\mathbf{x}} = \mathbf{r}$. Thus, we can compute the Jacobian–vector product in just one forward pass. As a special case, when $f : \mathbb{R}^n \to \mathbb{R}$, we can obtain the directional derivative along a given vector $\mathbf{r}$ as a linear combination of the partial derivatives

$$\nabla f \cdot \mathbf{r}$$

by starting the AD computation with the values $\dot{\mathbf{x}} = \mathbf{r}$.

Forward mode AD is efficient and straightforward for functions $f : \mathbb{R} \to \mathbb{R}^m$, as all the derivatives $\frac{dy_i}{dx}$ can be computed with just one forward pass. Conversely, in the other extreme of $f : \mathbb{R}^n \to \mathbb{R}$, forward mode AD requires n evaluations to compute the gradient

$$\nabla f = \left(\frac{\partial y}{\partial x_1}, \dots, \frac{\partial y}{\partial x_n} \right) ,$$

which also corresponds to a $1 \times n$ Jacobian matrix that is built one column at a time with the forward mode in n evaluations.

In general, for cases $f : \mathbb{R}^n \to \mathbb{R}^m$ where $n \gg m$, a different technique is often preferred. We will describe AD in *reverse accumulation mode* in Section 3.2.

3.1.1 Dual Numbers

Mathematically, forward mode AD (represented by the left- and right-hand sides in Table 2) can be viewed as evaluating a function using dual numbers,[10] which can be defined as truncated Taylor series of the form

$$v + \dot{v}\epsilon \,,$$

where $v, \dot{v} \in \mathbb{R}$ and ϵ is a nilpotent number such that $\epsilon^2 = 0$ and $\epsilon \neq 0$. Observe, for example, that

$$\begin{aligned}(v + \dot{v}\epsilon) + (u + \dot{u}\epsilon) &= (v + u) + (\dot{v} + \dot{u})\epsilon \\ (v + \dot{v}\epsilon)(u + \dot{u}\epsilon) &= (vu) + (v\dot{u} + \dot{v}u)\epsilon \,,\end{aligned}$$

in which the coefficients of ϵ conveniently mirror symbolic differentiation rules (e.g., Eq. 3). We can utilize this by setting up a regime where

$$f(v + \dot{v}\epsilon) = f(v) + f'(v)\dot{v}\epsilon \tag{5}$$

and using dual numbers as data structures for carrying the tangent value together with the primal.[11] The chain rule works as expected on this representation: two applications of Eq. 5 give

$$\begin{aligned}f(g(v + \dot{v}\epsilon)) &= f(g(v) + g'(v)\dot{v}\epsilon) \\ &= f(g(v)) + f'(g(v))g'(v)\dot{v}\epsilon \,.\end{aligned}$$

The coefficient of ϵ on the right-hand side is exactly the derivative of the composition of f and g. This means that since we implement elementary operations to respect the invariant Eq. 5, all compositions of them will also do so. This, in turn, means that we can extract the derivative of a function by interpreting any non-dual number v as $v + 0\epsilon$ and evaluating the function in this non-standard way on an initial input with a coefficient 1 for ϵ:

$$\left.\frac{df(x)}{dx}\right|_{x=v} = \text{epsilon-coefficient}(\text{dual-version}(f)(v + 1\epsilon)) \,.$$

This also extends to arbitrary program constructs, since dual numbers, as data types, can be contained in any data structure. As long as a dual number remains in a data structure with no arithmetic operations being performed on it, it will just remain a dual number; and if it is taken out of the data structure and operated on again, then the differentiation will continue.

In practice, a function f coded in a programming language of choice would be fed into an AD tool, which would then augment it with corresponding extra code to handle the dual operations so that the function and its derivative are simultaneously computed. This can be implemented through calls to a specific library, in the form of source code transformation where a given source code will be automatically modified, or through operator overloading, making the process transparent to the user. We discuss these implementation techniques in Section 5.

10. First introduced by Clifford (1873), with important uses in linear algebra and physics.
11. Just as the complex number written $x + yi$ is represented in the computer as a pair in memory (x, y) whose two slots are reals, the dual number written $x + \dot{x}\epsilon$ is represented as the pair $(x, \dot{x})$. Such pairs are sometimes called Argand pairs (Hamilton, 1837, p107 Eqs. (157) and (158)).

3.2 Reverse Mode

AD in the reverse accumulation mode[12] corresponds to a generalized backpropagation algorithm, in that it propagates derivatives backward from a given output. This is done by complementing each intermediate variable v_i with an adjoint

$$\bar{v}_i = \frac{\partial y_j}{\partial v_i} \,,$$

which represents the sensitivity of a considered output y_j with respect to changes in v_i. In the case of backpropagation, y would be a scalar corresponding to the error E (Figure 1).

In reverse mode AD, derivatives are computed in the second phase of a two-phase process. In the first phase, the original function code is run *forward*, populating intermediate variables v_i and recording the dependencies in the computational graph through a bookkeeping procedure. In the second phase, derivatives are calculated by propagating adjoints $\bar{v}_i$ in *reverse*, from the outputs to the inputs.

Returning to the example $y = f(x_1, x_2) = \ln(x_1) + x_1 x_2 - \sin(x_2)$, in Table 3 we see the adjoint statements on the right-hand side, corresponding to each original elementary operation on the left-hand side. In simple terms, we are interested in computing the contribution $\bar{v}_i = \frac{\partial y}{\partial v_i}$ of the change in each variable v_i to the change in the output y. Taking the variable v_0 as an example, we see in Figure 4 that the only way it can affect y is through affecting v_2 and v_3, so its contribution to the change in y is given by

$$\frac{\partial y}{\partial v_0} = \frac{\partial y}{\partial v_2}\frac{\partial v_2}{\partial v_0} + \frac{\partial y}{\partial v_3}\frac{\partial v_3}{\partial v_0} \qquad \text{or} \qquad \bar{v}_0 = \bar{v}_2\frac{\partial v_2}{\partial v_0} + \bar{v}_3\frac{\partial v_3}{\partial v_0} \,.$$

In Table 3, this contribution is computed in two incremental steps

$$\bar{v}_0 = \bar{v}_3\frac{\partial v_3}{\partial v_0} \qquad \text{and} \qquad \bar{v}_0 = \bar{v}_0 + \bar{v}_2\frac{\partial v_2}{\partial v_0} \,,$$

lined up with the lines in the forward trace from which these expressions originate.

After the forward pass on the left-hand side, we run the reverse pass of the adjoints on the right-hand side, starting with $\bar{v}_5 = \bar{y} = \frac{\partial y}{\partial y} = 1$. In the end we get the derivatives $\frac{\partial y}{\partial x_1} = \bar{x}_1$ and $\frac{\partial y}{\partial x_2} = \bar{x}_2$ in just one reverse pass.

Compared with the straightforwardness of forward accumulation mode, reverse mode AD can, at first, appear somewhat "mysterious" (Dennis and Schnabel, 1996). Griewank and Walther (2008) argue that this is in part because of the common acquaintance with the chain rule as a mechanistic procedure propagating derivatives forward.

An important advantage of the reverse mode is that it is significantly less costly to evaluate (in terms of operation count) than the forward mode for functions with a large number of inputs. In the extreme case of $f : \mathbb{R}^n \to \mathbb{R}$, only one application of the reverse mode is sufficient to compute the full gradient $\nabla f = \left(\frac{\partial y}{\partial x_1}, \ldots, \frac{\partial y}{\partial x_n}\right)$, compared with the n passes of the forward mode needed for populating the same. Because machine learning practice principally involves the gradient of a scalar-valued objective with respect to a large number of parameters, this establishes the reverse mode, as opposed to the forward mode, as the mainstay technique in the form of the backpropagation algorithm.

12. Also called *adjoint* or *cotangent linear* mode.

Table 3: Reverse mode AD example, with $y = f(x_1, x_2) = \ln(x_1) + x_1 x_2 - \sin(x_2)$ evaluated at $(x_1, x_2) = (2, 5)$. After the forward evaluation of the primals on the left, the adjoint operations on the right are evaluated in reverse (cf. Figure 1). Note that both $\frac{\partial y}{\partial x_1}$ and $\frac{\partial y}{\partial x_2}$ are computed in the same reverse pass, starting from the adjoint $\bar{v}_5 = \bar{y} = \frac{\partial y}{\partial y} = 1$.

Forward Primal Trace		Reverse Adjoint (Derivative) Trace		
$v_{-1} = x_1$	$= 2$	$\bar{\boldsymbol{x}}_{\mathbf{1}} = \bar{\boldsymbol{v}}_{\mathbf{-1}}$		$\mathbf{= 5.5}$
$v_0 = x_2$	$= 5$	$\bar{\boldsymbol{x}}_{\mathbf{2}} = \bar{\boldsymbol{v}}_{\mathbf{0}}$		$\mathbf{= 1.716}$
$v_1 = \ln v_{-1}$	$= \ln 2$	$\bar{v}_{-1} = \bar{v}_{-1} + \bar{v}_1 \frac{\partial v_1}{\partial v_{-1}}$	$= \bar{v}_{-1} + \bar{v}_1 / v_{-1}$	$= 5.5$
$v_2 = v_{-1} \times v_0$	$= 2 \times 5$	$\bar{v}_0 = \bar{v}_0 + \bar{v}_2 \frac{\partial v_2}{\partial v_0}$	$= \bar{v}_0 + \bar{v}_2 \times v_{-1}$	$= 1.716$
		$\bar{v}_{-1} = \bar{v}_2 \frac{\partial v_2}{\partial v_{-1}}$	$= \bar{v}_2 \times v_0$	$= 5$
$v_3 = \sin v_0$	$= \sin 5$	$\bar{v}_0 = \bar{v}_3 \frac{\partial v_3}{\partial v_0}$	$= \bar{v}_3 \times \cos v_0$	$= -0.284$
$v_4 = v_1 + v_2$	$= 0.693 + 10$	$\bar{v}_2 = \bar{v}_4 \frac{\partial v_4}{\partial v_2}$	$= \bar{v}_4 \times 1$	$= 1$
		$\bar{v}_1 = \bar{v}_4 \frac{\partial v_4}{\partial v_1}$	$= \bar{v}_4 \times 1$	$= 1$
$v_5 = v_4 - v_3$	$= 10.693 + 0.959$	$\bar{v}_3 = \bar{v}_5 \frac{\partial v_5}{\partial v_3}$	$= \bar{v}_5 \times (-1)$	$= -1$
		$\bar{v}_4 = \bar{v}_5 \frac{\partial v_5}{\partial v_4}$	$= \bar{v}_5 \times 1$	$= 1$
$y = v_5$	$= 11.652$	$\bar{v}_5 = \bar{y}$	$= 1$	

In general, for a function $f : \mathbb{R}^n \to \mathbb{R}^m$, if we denote the operation count to evaluate the original function by $\mathrm{ops}(f)$, the time it takes to calculate the $m \times n$ Jacobian by the forward mode is $n\, c\, \mathrm{ops}(f)$, whereas the same computation can be done via reverse mode in $m\, c\, \mathrm{ops}(f)$, where c is a constant guaranteed to be $c < 6$ and typically $c \sim [2, 3]$ (Griewank and Walther, 2008). That is to say, reverse mode AD performs better when $m \ll n$.

Similar to the matrix-free computation of Jacobian–vector products with forward mode (Eq. 4), reverse mode can be used for computing the transposed Jacobian–vector product

$$\mathbf{J}_f^\intercal \mathbf{r} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_1}{\partial x_n} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix} \begin{bmatrix} r_1 \\ \vdots \\ r_m \end{bmatrix},$$

by initializing the reverse phase with $\bar{\mathbf{y}} = \mathbf{r}$.

The advantages of reverse mode AD, however, come with the cost of increased storage requirements growing (in the worst case) in proportion to the number of operations in the evaluated function. It is an active area of research to improve storage requirements in implementations by using advanced methods such as checkpointing strategies and data-flow analysis (Dauvergne and Hascoët, 2006; Siskind and Pearlmutter, 2017).

3.3 Origins of AD and Backpropagation

Ideas underlying AD date back to the 1950s (Nolan, 1953; Beda et al., 1959). Forward mode AD as a general method for evaluating partial derivatives was essentially discovered

by Wengert (1964). It was followed by a period of relatively low activity, until interest in the field was revived in the 1980s mostly through the work of Griewank (1989), also supported by improvements in modern programming languages and the feasibility of an efficient reverse mode AD.

Reverse mode AD and backpropagation have an intertwined history. The essence of the reverse mode, cast in a continuous-time formalism, is the Pontryagin maximum principle (Rozonoer, 1959; Boltyanskii et al., 1960). This method was understood in the control theory community (Bryson and Denham, 1962; Bryson and Ho, 1969) and cast in more formal terms with discrete-time variables topologically sorted in terms of dependency by Werbos (1974). Prior to Werbos, the work by Linnainmaa (1970, 1976) is often cited as the first published description of the reverse mode. Speelpenning (1980) subsequently introduced reverse mode AD as we know it, in the sense that he gave the first implementation that was actually automatic, accepting a specification of a computational process written in a general-purpose programming language and automatically performing the reverse mode transformation.

Incidentally, Hecht-Nielsen (1989) cites the work of Bryson and Ho (1969) and Werbos (1974) as the two earliest known instances of backpropagation. Within the machine learning community, the method has been reinvented several times, such as by Parker (1985), until it was eventually brought to fame by Rumelhart et al. (1986) and the Parallel Distributed Processing (PDP) group. The PDP group became aware of Parker's work only after their own discovery; similarly, Werbos' work was not appreciated until it was found by Parker (Hecht-Nielsen, 1989). This tells us an interesting story of two highly interconnected research communities that have somehow also managed to stay detached during this foundational period.

For a thorough review of the development of AD, we advise readers to refer to Rall (2006). Interested readers are highly recommended to read Griewank (2012) for an investigation of the origins of the reverse mode and Schmidhuber (2015) for the same for backpropagation.

4. AD and Machine Learning

In the following, we examine the main uses of derivatives in machine learning and report on a selection of works where general-purpose AD, as opposed to just backpropagation, has been successfully applied in a machine learning context. Areas where AD has seen use include optimization, neural networks, computer vision, natural language processing, and probabilistic inference.

4.1 Gradient-Based Optimization

Gradient-based optimization is one of the pillars of machine learning (Bottou et al., 2016). Given an objective function $f : \mathbb{R}^n \to \mathbb{R}$, classical gradient descent has the goal of finding (local) minima $\mathbf{w}^* = \arg\min_{\mathbf{w}} f(\mathbf{w})$ via updates of the form $\Delta\mathbf{w} = -\eta \nabla f$, where $\eta > 0$ is a step size. Gradient-based methods make use of the fact that f decreases steepest if one goes in the direction of the negative gradient. The convergence rate of gradient-based methods is usually improved by adaptive step-size techniques that adjust the step size η on every iteration (Duchi et al., 2011; Schaul et al., 2013; Kingma and Ba, 2015).

Table 4: Evaluation times of the Helmholtz free energy function and its gradient (Figure 5). Times are given relative to that of the original function with both (1) $n = 1$ and (2) n corresponding to each column. (For instance, reverse mode AD with $n = 43$ takes approximately twice the time to evaluate relative to the original function with $n = 43$.) Times are measured by averaging a thousand runs on a machine with Intel Core i7-4785T 2.20 GHz CPU and 16 GB RAM, using DiffSharp 0.5.7. The evaluation time for the original function with $n = 1$ is 0.0023 ms.

	n, number of variables							
	1	8	15	22	29	36	43	50
f, original								
Relative $n = 1$	1	5.12	14.51	29.11	52.58	84.00	127.33	174.44
∇f, numerical diff.								
Relative $n = 1$	1.08	35.55	176.79	499.43	1045.29	1986.70	3269.36	4995.96
Relative n in column	1.08	6.93	12.17	17.15	19.87	23.64	25.67	28.63
∇f, forward AD								
Relative $n = 1$	1.34	13.69	51.54	132.33	251.32	469.84	815.55	1342.07
Relative n in column	1.34	2.66	3.55	4.54	4.77	5.59	6.40	7.69
∇f, reverse AD								
Relative $n = 1$	1.52	11.12	31.37	67.27	113.99	174.62	254.15	342.33
Relative n in column	1.52	2.16	2.16	2.31	2.16	2.07	1.99	1.96

As we have seen, for large n, reverse mode AD provides a highly efficient method for computing gradients.[13] Figure 5 and Table 4 demonstrate how gradient computation scales differently for forward and reverse mode AD and numerical differentiation, looking at the Helmholtz free energy function that has been used in AD literature for benchmarking gradient calculations (Griewank, 1989; Griewank and Walther, 2008; Griewank et al., 2012).

Second-order methods based on Newton's method make use of both the gradient ∇f and the Hessian $\mathbf{H}_f$, working via updates of the form $\Delta\mathbf{w} = -\eta\,\mathbf{H}_f^{-1}\nabla f$ and providing significantly faster convergence (Press et al., 2007). AD provides a way of automatically computing the exact Hessian, enabling succinct and convenient general-purpose implementations.[14] Newton's method converges in fewer iterations, but this comes at the cost of having to compute $\mathbf{H}_f$ in each iteration. In large-scale problems, the Hessian is usually replaced by a numerical approximation using first-order updates from gradient evaluations, giving rise to quasi-Newton methods. A highly popular such method is the BFGS[15] algorithm, together with its limited-memory variant L-BFGS (Dennis and Schnabel, 1996). On the other hand, Hessians arising in large-scale applications are typically sparse. This spar-

13. See `http://DiffSharp.github.io/DiffSharp/examples-gradientdescent.html` for an example of a general-purpose AD-based gradient descent routine using DiffSharp.
14. See `http://DiffSharp.github.io/DiffSharp/examples-newtonsmethod.html` for an implementation of Newton's method with the full Hessian.
15. After Broyden–Fletcher–Goldfarb–Shanno, who independently discovered the method in the 1970s.

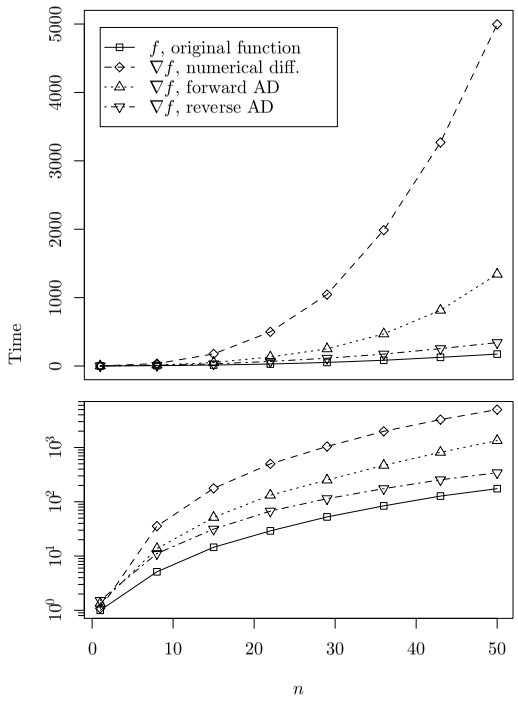


Figure 5: Evaluation time of the Helmholtz free energy function of a mixed fluid, based on the Peng-Robinson equation of state (Peng and Robinson, 1976), $f(\mathbf{x}) = RT\sum_{i=0}^{n}\log\frac{x_i}{1-\mathbf{b^T x}} - \frac{\mathbf{x^T A x}}{\sqrt{8}\mathbf{b^T x}}\log\frac{1+(1+\sqrt{2})\mathbf{b^T x}}{1+(1-\sqrt{2})\mathbf{b^T x}}$, where R is the universal gas constant, T is the absolute temperature, $\mathbf{b} \in \mathbb{R}^n$ is a vector of constants, $\mathbf{A} \in \mathbb{R}^{n\times n}$ is a symmetric matrix of constants, and $\mathbf{x} \in \mathbb{R}^n$ is the vector of independent variables describing the system. The plots show the evaluation time of f and the gradient ∇f with numerical differentiation (central difference), forward mode AD, and reverse mode AD, as a function of the number of variables n. Reported times are relative to the evaluation time of f with $n = 1$. The lower plot uses logarithmic scale for illustrating the behavior for small n. Numerical results are given in Table 4. (Code: `http://DiffSharp.github.io/DiffSharp/misc/Benchmarks-h-grad-v0.5.7.fsx`)

sity along with symmetry can be readily exploited by AD techniques such as computational graph elimination (Dixon, 1991), partial separability (Gay, 1996), and matrix coloring and compression (Gebremedhin et al., 2009).

In many cases one does not need the full Hessian but only a Hessian–vector product $\mathbf{Hv}$, which can be computed efficiently using a reverse-on-forward configuration of AD by applying the reverse mode to take the gradient of code produced by the forward mode.[16] Given the function $f : \mathbb{R}^n \to \mathbb{R}$, the evaluation point $\mathbf{x}$, and the vector $\mathbf{v}$, one can accomplish this by first computing the directional derivative $\nabla f \cdot \mathbf{v}$ through the forward mode via setting $\dot{\mathbf{x}} = \mathbf{v}$ and then applying the reverse mode on this result to get $\nabla^2 f \cdot \mathbf{v} = \mathbf{H}_f \mathbf{v}$ (Pearlmutter, 1994). This computes $\mathbf{Hv}$ with $O(n)$ complexity, even though $\mathbf{H}$ is a $n \times n$ matrix. Availability of robust AD tools may make more sophisticated optimization methods applicable to large-scale machine-learning problems. For instance, when fast stochastic Hessian–vector products are available, these can be used as the basis of stochastic Newton's methods (Agarwal et al., 2016), which have the potential to endow stochastic optimization with quadratic convergence.

Another approach for improving the rate of convergence of gradient-based methods is to use gain adaptation methods such as stochastic meta-descent (SMD) (Schraudolph, 1999), where stochastic sampling is introduced to avoid local minima and reduce the computational expense. An example using SMD with AD Hessian–vector products is given by Vishwanathan et al. (2006) on conditional random fields (CRF). Similarly, Schraudolph and Graepel (2003) use Hessian–vector products in their model combining conjugate gradient techniques with stochastic gradient descent.

4.2 Neural Networks, Deep Learning, Differentiable Programming

Training of a neural network is an optimization problem with respect to its set of weights, which can in principle be addressed by using any method ranging from evolutionary algorithms (Such et al., 2017) to gradient-based methods such as BFGS (Apostolopoulou et al., 2009) or the mainstay stochastic gradient descent (Bottou, 2010) and its many variants (Kingma and Ba, 2015; Tieleman and Hinton, 2012; Duchi et al., 2011). As we have seen, the backpropagation algorithm is only a special case of AD: by applying reverse mode AD to an objective function evaluating a network's error as a function of its weights, we can readily compute the partial derivatives needed for performing weight updates.[17]

The LUSH system (Bottou and LeCun, 2002), and its predecessor SN (Bottou and LeCun, 1988), were the first production systems that targeted efficient neural network simulation while incorporating both a general-purpose programming language and AD. Modern deep learning frameworks provide differentiation capability in one way or another, but the underlying mechanism is not always made clear and confusion abounds regarding the use of the terms "autodiff", "automatic differentiation", and "symbolic differentiation", which

16. Christianson (2012) demonstrates that the second derivative can be computed with the same arithmetic operation sequence using forward-on-reverse, reverse-on-forward, and reverse-on-reverse. The taping overheads of these methods may differ in implementation-dependent ways.
17. See `http://DiffSharp.github.io/DiffSharp/examples-neuralnetworks.html` for an implementation of backpropagation with reverse mode AD.

are sometimes even used interchangeably. In mainstream frameworks including Theano[18] (Bastien et al., 2012), TensorFlow (Abadi et al., 2016), Caffe (Jia et al., 2014), and CNTK (Seide and Agarwal, 2016) the user first constructs a model as a computational graph using a domain-specific mini language, which then gets interpreted by the framework during execution. This approach has the advantage of enabling optimizations of the computational graph structure (e.g., as in Theano), but the disadvantages of having limited and unintuitive control flow and being difficult to debug. In contrast, the lineage of recent frameworks led by autograd (Maclaurin, 2016), Chainer (Tokui et al., 2015), and PyTorch (Paszke et al., 2017) provide truly general-purpose reverse mode AD of the type we outline in Section 3, where the user directly uses the host programming language to define the model as a regular program of the forward computation. This eliminates the need for an interpreter, allows arbitrary control flow statements, and makes debugging simple and intuitive.

Simultaneously with the ongoing adoption of general-purpose AD in machine learning, we are witnessing a modeling-centric terminology emerge within the deep learning community. The terms *define-and-run* and *static computational graph* refer to Theano-like systems where a model is constructed, before execution, as a computational graph structure, which later gets executed with different inputs while remaining fixed. In contrast, the terms *define-by-run* and *dynamic computational graph* refer to the general-purpose AD capability available in newer PyTorch-like systems where a model is a regular program in the host programming language, whose execution dynamically constructs a computational graph on-the-fly that can freely change in each iteration.[19]

Differentiable programming[20] is another emerging term referring to the realization that deep learning models are essentially differentiable program templates of potential solutions to a problem. These templates are constructed as differentiable directed graphs assembled from functional blocks, and their parameters are learned using gradient-based optimization of an objective describing the task. Expressed in this paradigm, neural networks are just a class of parameterized differentiable programs composed of building blocks such as feed-forward, convolutional, and recurrent elements. We are increasingly seeing these traditional building blocks freely composed in arbitrary algorithmic structures using control flow, as well as the introduction of novel differentiable architectures such as the neural Turing machine (Graves et al., 2014), a range of controller–interface abstractions (Graves et al., 2016; Zaremba et al., 2016; Joulin and Mikolov, 2015; Sukhbaatar et al., 2015), and differentiable versions of data structures such as stacks, queues, deques (Grefenstette et al., 2015). Availability of general-purpose AD greatly simplifies the implementation of such architectures by enabling their expression as regular programs that rely on the differentiation infrastructure. Although the differentiable programming perspective on deep learning is new, we note that

18. Theano is a computational graph optimizer and compiler with GPU support and it currently handles derivatives in a highly optimized form of symbolic differentiation. The result can be interpreted as a hybrid of symbolic differentiation and reverse mode AD, but Theano does not use the general-purpose reverse accumulation as we describe in this paper. (Personal communication with the authors.)
19. Note that the terms "static" and "dynamic" here are used in the sense of having a fixed versus non-fixed computational graph topology and not in the sense of data flow architectures.
20. A term advocated by Christopher Olah (`http://colah.github.io/posts/2015-09-NN-Types-FP/`), David Dalrymple (`https://www.edge.org/response-detail/26794`), and Yann LeCun (`https://www.facebook.com/yann.lecun/posts/10155003011462143`) from a deep learning point of view. Note the difference from *differential* dynamic programming (Mayne and Jacobson, 1970) in optimal control.

programming with differentiable functions and having differentiation as a language infrastructure has been the main research subject of the AD community for many decades and realized in a wide range of systems and languages as we shall see in Section 5.

There are instances in neural network literature—albeit few—where explicit reference has been made to AD for computing error gradients, such as Eriksson et al. (1998) using AD for large-scale feed-forward networks, and the work by Yang et al. (2008), where the authors use AD to train a neural-network-based proportional-integral-derivative (PID) controller. Similarly, Rollins (2009) uses reverse mode AD in conjunction with neural networks for the problem of optimal feedback control. Another example is given for continuous time recurrent neural networks (CTRNN) by Al Seyab and Cao (2008), where the authors apply AD for the training of CTRNNs predicting dynamic behavior of nonlinear processes in real time and report significantly reduced training time compared with other methods.

4.3 Computer Vision

Since the influential work by Krizhevsky et al. (2012), computer vision has been dominated by deep learning, specifically, variations of convolutional neural networks (LeCun et al., 1998). These models are trained end-to-end, meaning that a mapping from raw input data to corresponding outputs is learned, automatically discovering the representations needed for feature detection in a process called representation learning (Bengio et al., 2013).

Besides deep learning, an interesting area where AD can be applied to computer vision problems is inverse graphics (Horn, 1977; Hinton and Ghahramani, 1997)—or analysis-by-synthesis (Yildirim et al., 2015)—where vision is seen as the inference of parameters for a generative model of a scene. Using gradient-based optimization in inverse graphics requires propagating derivatives through whole image synthesis pipelines including the renderer. Eslami et al. (2016) use numerical differentiation for this purpose. Loper and Black (2014) implement the Open Differentiable Renderer (OpenDR), which is a scene renderer that also supplies derivatives of the image pixels with respect to scene parameters, and demonstrate it in the task of fitting an articulated and deformable 3D model of the human body to image and range data from a Kinect device. Similarly, Kulkarni et al. (2015) implement a differentiable approximate renderer for the task of inference in probabilistic programs describing scenes.

Srajer et al. (2016) investigate the use of AD for three tasks in computer vision and machine learning, namely bundle adjustment (Triggs et al., 1999), Gaussian mixture model fitting, and hand tracking (Taylor et al., 2014), and provide a comprehensive benchmark of various AD tools for the computation of derivatives in these tasks.

Pock et al. (2007) make use of AD in addressing the problems of denoising, segmentation, and recovery of information from stereoscopic image pairs, and note the usefulness of AD in identifying sparsity patterns in large Jacobian and Hessian matrices. In another study, Grabner et al. (2008) use reverse mode AD for GPU-accelerated medical 2D/3D registration, a task involving the alignment of data from different sources such as X-ray images or computed tomography. The authors report a six-fold increase in speed compared with numerical differentiation using center difference (cf. our benchmark with the Helmholtz function, Figure 5 and Table 4).

Barrett and Siskind (2013) present a use of general-purpose AD for the task of video event detection using hidden Markov models (HMMs) and Dalal and Triggs (2005) object detectors, performing training on a corpus of pre-tracked video using an adaptive step size gradient descent with reverse mode AD. Initially implemented with the R6RS-AD package[21] which provides forward and reverse mode AD in Scheme, the resulting gradient code was later ported to C and highly optimized.[22]

4.4 Natural Language Processing

Natural language processing (NLP) constitutes one of the areas where rapid progress is being made by applying deep learning techniques (Goldberg, 2016), with applications in tasks including machine translation (Bahdanau et al., 2014), language modeling (Mikolov et al., 2010), dependency parsing (Chen and Manning, 2014), and question answering (Kumar et al., 2016). Besides deep learning approaches, statistical models in NLP are commonly trained using general purpose or specialized gradient-based methods and mostly remain expensive to train. Improvements in training time can be realized by using online or distributed training algorithms (Gimpel et al., 2010). An example using stochastic gradient descent for NLP is given by Finkel et al. (2008) optimizing conditional random field parsers through an objective function. Related with the work on video event detection in the previous section, Yu and Siskind (2013) report their work on sentence tracking, representing an instance of grounded language learning paired with computer vision, where the system learns word meanings from short video clips paired with descriptive sentences. The method uses HMMs to represent changes in video frames and meanings of different parts of speech. This work is implemented in C and computes the required gradients using AD through the ADOL-C tool.[23]

4.5 Probabilistic Modeling and Inference

Inference in probabilistic models can be static, such as compiling a given model to Bayesian networks and using algorithms such as belief propagation for inference; or they can be dynamic, executing a model forward many times and computing statistics on observed values to infer posterior distributions. Markov chain Monte Carlo (MCMC) (Neal, 1993) methods are often used for dynamic inference, such as the Metropolis–Hastings algorithm based on random sampling (Chib and Greenberg, 1995). Meyer et al. (2003) give an example of how AD can be used to speed up Bayesian posterior inference in MCMC, with an application in stochastic volatility. Amortized inference (Gershman and Goodman, 2014; Stuhlmüller et al., 2013) techniques based on deep learning (Le et al., 2017; Ritchie et al., 2016) work by training neural networks for performing approximate inference in generative models defined as probabilistic programs (Gordon et al., 2014).

When model parameters are continuous, the Hamiltonian—or, hybrid—Monte Carlo (HMC) algorithm provides improved convergence characteristics avoiding the slow explo-

21. `https://github.com/qobi/R6RS-AD`
22. Personal communication.
23. An implementation of the sentence tracker applied to video search using sentence-based queries can be accessed online: `http://upplysingaoflun.ecn.purdue.edu/~qobi/cccp/sentence-tracker-video-retrieval.html`

ration of random sampling, by simulating Hamiltonian dynamics through auxiliary "momentum variables" (Duane et al., 1987). The advantages of HMC come at the cost of requiring gradient evaluations of complicated probability models. AD is highly suitable here for complementing probabilistic modeling, because it relieves the user from the manual derivation of gradients for each model.[24] For instance, the probabilistic programming language Stan (Carpenter et al., 2016) implements automatic Bayesian inference based on HMC and the No-U-Turn sampler (NUTS) (Hoffman and Gelman, 2014) and uses reverse mode AD for the calculation of gradients for both HMC and NUTS (Carpenter et al., 2015). Similarly, Wingate et al. (2011) demonstrate the use of AD as a non-standard interpretation of probabilistic programs enabling efficient inference algorithms. Kucukelbir et al. (2017) present an AD-based method for deriving variational inference (VI) algorithms.

PyMC3 (Salvatier et al., 2016) allows fitting of Bayesian models using MCMC and VI, for which it uses gradients supplied by Theano. Edward (Tran et al., 2016) is a library for deep probabilistic modeling, inference, and criticism (Tran et al., 2017) that supports VI using TensorFlow. Availability of general-purpose AD in this area has enabled new libraries such as Pyro[25] and ProbTorch (Siddharth et al., 2017) for deep *universal* probabilistic programming with support for recursion and control flow, relying, in both instances, on VI using gradients supplied by PyTorch's reverse mode AD infrastructure.

When working with probabilistic models, one often needs to backpropagate derivatives through sampling operations of random variables in order to achieve stochastic optimization of model parameters. The score-function estimator, or REINFORCE (Williams, 1992), method provides a generally applicable unbiased gradient estimate, albeit with high variance. When working with continuous random variables, one can substitute a random variable by a deterministic and differentiable transformation of a simpler random variable, a method known as the "reparameterization trick" (Williams, 1992; Kingma and Welling, 2014; Rezende et al., 2014). For discrete variables, the REBAR (Tucker et al., 2017) method provides a lower-variance unbiased gradient estimator by using continuous relaxation. A generalization of REBAR called RELAX (Grathwohl et al., 2017) works by learning a free-form control variate parameterized by a neural network and is applicable in both discrete and continuous settings.

5. Implementations

It is useful to have an understanding of the different ways in which AD can be implemented. Here we cover major implementation strategies and provide a survey of existing tools.

A principal consideration in any AD implementation is the performance overhead introduced by the AD arithmetic and bookkeeping. In terms of computational complexity, AD guarantees that the amount of arithmetic goes up by no more than a small constant factor (Griewank and Walther, 2008). On the other hand, managing this arithmetic can introduce a significant overhead if done carelessly. For instance, naïvely allocating data structures for holding dual numbers will involve memory access and allocation for every arithmetic operation, which are usually more expensive than arithmetic operations on modern computers.[26] Likewise, using operator overloading may introduce method dispatches with attendant costs, which, compared to raw numerical computation of the original function, can easily amount to a slowdown of an order of magnitude.[27]

Another major issue is the risk of hitting a class of bugs called "perturbation confusion" (Siskind and Pearlmutter, 2005; Manzyuk et al., 2012). This essentially means that if two ongoing differentiations affect the same piece of code, the two formal epsilons they introduce (Section 3.1.1) need to be kept distinct. It is very easy to have bugs—particularly in performance-oriented AD implementations—that confuse these in various ways. Such situations can also arise when AD is nested, that is, derivatives are computed for functions that internally compute derivatives.

Translation of mathematics into computer code often requires attention to numeric issues. For instance, the mathematical expressions $\log(1+x)$ or $\sqrt{x^2+y^2+z^2}$ or $\tan^{-1}(y/x)$ should not be naïvely translated, but rather expressed as `log1p(x)`, `hypot(x,hypot(y,z))`, and `atan2(y,x)`. In machine learning, the most prominent example of this is probably the so-called log-sum-exp trick to improve the numerics of calculations of the form $\log \sum_i \exp x_i$. AD is not immune to such numeric considerations. For example, code calculating $E = \sum_i E_i$, processed by AD, will calculate $\nabla_w E = \sum_i \nabla_w E_i$. If the system is seeking a local minimum of E then $\nabla_w E = \sum_i \nabla_w E_i \to_t 0$, and naïvely adding a set of large numbers whose sum is near zero is numerically fraught. This is to say that AD is not immune to the perils of floating point arithmetic, and can sometimes introduce numeric issues which were not present in the primal calculation. Issues of numeric analysis are outside our present scope, but there is a robust literature on the numerics of AD (e.g., Griewank et al. (2012)) involving using subgradients to allow optimization to proceed despite non-differentiability of the objective, appropriate subgradients and approximations for functions like $|\cdot|$ and $\|\cdot\|_2$ and $\sqrt{\cdot}$ near zero, and a spate of related issues.

One should also be cautious about approximated functions and AD (Sirkes and Tziperman, 1997). In this case, if one has a procedure *approximating* an ideal function, AD always gives the derivative of the procedure that was actually programmed, which may not be a good approximation of the derivative of the ideal function that the procedure was approximating. For instance, consider e^x computed by a piecewise-rational approximation routine. Using AD on this routine would produce an approximated derivative in which each piece of the piecewise formula will get differentiated. Even if this would remain an approximation of the derivative of e^x, we know that $\frac{de^x}{dx} = e^x$ and the original approximation itself was already a better approximation for the derivative of e^x.[28] Users of AD implementations must be therefore cautious to *approximate the derivative, not differentiate the approximation.* This would require explicitly approximating a known derivative, in cases where a mathe-

24. See `http://diffsharp.github.io/DiffSharp/examples-hamiltonianmontecarlo.html` for an implementation of HMC with reverse mode AD.
25. `http://pyro.ai/`
26. The implementation of forward mode in Julia (Revels et al., 2016b) attempts to avoid this, and some current compilers can avoid this expense by unboxing dual numbers (Leroy, 1997; Jones et al., 1993b; Jones and Launchbury, 1991; Siskind and Pearlmutter, 2016). This method is also used to reduce the memory-access overhead in the implementations of forward mode in Stalingrad and the Haskell *ad* library.
27. Flow analysis (Shivers, 1991) and/or partial evaluation (Jones et al., 1993a), together with tag stripping (Appel, 1989; Peterson, 1989), can remove this method dispatch. These, together with unboxing, can often make it possible to completely eliminate the memory access, memory allocation, memory reclamation, and method dispatch overhead of dual numbers (Siskind and Pearlmutter, 2016).
28. In modern systems this is not an issue, because e^x is a primitive implemented in hardware.

matical function can only be computed approximately but has a well-defined mathematical derivative.

We note that there are similarities as well as differences between machine learning workloads and those studied in the traditional AD literature (Baydin et al., 2016b). Deep learning systems are generally compute-bound and spend a considerable amount of computation time in highly-optimized numerical kernels for matrix operations (Hadjis et al., 2015; Chetlur et al., 2014). This is a situation which is arguably amenable to operator-overloading-based AD implementations on high-level operations, as is commonly found in current machine learning frameworks. In contrast, numerical simulation workloads in traditional AD applications can be bandwidth-bound, making source code transformation and compiler optimization approaches more relevant. Another difference worth noting is that whereas high numerical precision is desirable in traditional application domains of AD such as computational fluid dynamics (Cohen and Molemaker, 2009), in deep learning lower-precision is sufficient and even desirable in improving computational efficiency, thanks to the error resiliency of neural networks (Gupta et al., 2015; Courbariaux et al., 2015).

There are instances in recent literature where implementation-related experience from the AD field has been put to use in machine learning settings. One particular area of recent interest is implicit and iterative AD techniques (Griewank and Walther, 2008), which has found use in work incorporating constrained optimization within deep learning (Amos and Kolter, 2017) and probabilistic graphical models and neural networks (Johnson et al., 2016). Another example is checkpointing strategies (Dauvergne and Hascoët, 2006; Siskind and Pearlmutter, 2017), which allow balancing of application-specific trade-offs between time and space complexities of reverse mode AD by not storing the full tape of intermediate variables in memory and reconstructing these as needed by re-running parts of the forward computation from intermediate checkpoints. This is highly relevant in deep learning workloads running on GPUs with limited memory budgets. A recent example in this area is the work by Gruslys et al. (2016), where the authors construct a checkpointing variety of the backpropagation through time (BPTT) algorithm for recurrent neural networks and demonstrate it saving up to 95% memory usage at the cost of a 33% increase in computation time in one instance.

In Table 5 we present a review of notable general-purpose AD implementations.[29] A thorough taxonomy of implementation techniques was introduced by Juedes (1991), which was later revisited by Bischof et al. (2008) and simplified into *elemental*, *operator overloading*, *compiler-based*, and *hybrid* methods. We adopt a similar classification for the following part of this section.

5.1 Elemental Libraries

These implementations form the most basic category and work by replacing mathematical operations with calls to an AD-enabled library. Methods exposed by the library are then used in function definitions, meaning that the decomposition of any function into elementary operations is done manually when writing the code.

The approach has been utilized since the early days of AD, with prototypical examples being the WCOMP and UCOMP packages of Lawson (1971), the APL package of Neidinger

29. Also see the website `http://www.autodiff.org/` for a list of tools maintained by the AD community.

Table 5: Survey of AD implementations. Tools developed primarily for machine learning are highlighted in bold.

Language	Tool	Type	Mode	Institution / Project	Reference	URL
AMPL	AMPL	INT	F, R	Bell Laboratories	Fourer et al. (2002)	`http://www.ampl.com/`
C, C++	ADIC	ST	F, R	Argonne National Laboratory	Bischof et al. (1997)	`http://www.mcs.anl.gov/research/projects/adic/`
	ADOL-C	OO	F, R	Computational Infrastructure for Operations Research	Walther and Griewank (2012)	`https://projects.coin-or.org/ADOL-C`
C++	Ceres Solver	LIB	F	Google		`http://ceres-solver.org/`
	CppAD	OO	F, R	Computational Infrastructure for Operations Research	Bell and Burke (2008)	`http://www.coin-or.org/CppAD/`
	FADBAD++	OO	F, R	Technical University of Denmark	Bendtsen and Stauning (1996)	`http://www.fadbad.com/fadbad.html`
	Mxyzptlk	OO	F	Fermi National Accelerator Laboratory	Ostiguy and Michelotti (2007)	
C#	AutoDiff	LIB	R	George Mason Univ., Dept. of Computer Science	Shtof et al. (2013)	`http://autodiff.codeplex.com/`
F#, C#	**DiffSharp**	OO	F, R	Maynooth University, Microsoft Research Cambridge	Baydin et al. (2016a)	`http://diffsharp.github.io`
Fortran	ADIFOR	ST	F, R	Argonne National Laboratory	Bischof et al. (1996)	`http://www.mcs.anl.gov/research/projects/adifor/`
	NAGWare	COM	F, R	Numerical Algorithms Group	Naumann and Riehme (2005)	`http://www.nag.co.uk/nagware/Research/ad_overview.asp`
	TAMC	ST	R	Max Planck Institute for Meteorology	Giering and Kaminski (1998)	`http://autodiff.com/tamc/`
Fortran, C	COSY	INT	F	Michigan State Univ., Biomedical and Physical Sci.	Berz et al. (1996)	`http://www.bt.pa.msu.edu/index_cosy.htm`
Haskell	Tapenade	ST	F, R	INRIA Sophia-Antipolis	Hascoët and Pascual (2013)	`http://www-sop.inria.fr/tropics/tapenade.html`
	ad	OO	F, R	Haskell package		`http://hackage.haskell.org/package/ad`
Java	ADiJaC	ST	F, R	University Politehnica of Bucharest	Slusanschi and Dumitrel (2016)	`http://adijac.cs.pub.ro`
	Deriva	LIB	R	Java & Clojure library		`https://github.com/lambder/Deriva`
Julia	JuliaDiff	OO	F, R	Julia packages	Revels et al. (2016a)	`http://www.juliadiff.org/`
Lua	**torch-autograd**	OO	R	Twitter Cortex		`https://github.com/twitter/torch-autograd`
MATLAB	ADiMat	ST	F, R	Technical University of Darmstadt, Scientific Comp.	Willkomm and Vehreschild (2013)	`http://adimat.sc.informatik.tu-darmstadt.de/`
	INTLab	OO	F	Hamburg Univ. of Technology, Inst. for Reliable Comp.	Rump (1999)	`http://www.ti3.tu-harburg.de/rump/intlab/`
	TOMLAB/MAD	OO	F	Cranfield University & Tomlab Optimization Inc.	Forth (2006)	`http://tomlab.biz/products/mad`
Python	ad	OO	R	Python package		`https://pypi.python.org/pypi/ad`
	autograd	OO	F, R	Harvard Intelligent Probabilistic Systems Group	Maclaurin (2016)	`https://github.com/HIPS/autograd`
	Chainer	OO	R	Preferred Networks	Tokui et al. (2015)	`https://chainer.org/`
	PyTorch	OO	R	PyTorch core team	Paszke et al. (2017)	`http://pytorch.org/`
	Tangent	ST	F, R	Google Brain	van Merriënboer et al. (2017)	`https://github.com/google/tangent`
Scheme	R6RS-AD	OO	F, R	Purdue Univ., School of Electrical and Computer Eng.		`https://github.com/qobi/R6RS-AD`
	Scmutils	OO	F	MIT Computer Science and Artificial Intelligence Lab.	Sussman and Wisdom (2001)	`http://groups.csail.mit.edu/mac/users/gjs/6946/refman.txt`
	Stalingrad	COM	F, R	Purdue Univ., School of Electrical and Computer Eng.	Pearlmutter and Siskind (2008)	`http://www.bcl.hamilton.ie/~qobi/stalingrad/`

F: Forward, R: Reverse; COM: Compiler, INT: Interpreter, LIB: Library, OO: Operator overloading, ST: Source transformation

(1989), and the work by Hinkins (1994). Likewise, Rich and Hill (1992) formulate their implementation of AD in MATLAB using elemental methods.

Elemental libraries still constitute the simplest strategy to implement AD for languages without operator overloading.

5.2 Compilers and Source Code Transformation

These implementations provide extensions to programming languages that automate the decomposition of algorithms into AD-enabled elementary operations. They are typically executed as preprocessors[30] to transform the input in the extended language into the original language.

Classical instances of source code transformation include the Fortran preprocessors GRESS (Horwedel et al., 1988) and PADRE2 (Kubo and Iri, 1990), which transform AD-enabled variants of Fortran into standard Fortran 77 before compiling. Similarly, the ADIFOR tool (Bischof et al., 1996), given a Fortran source code, generates an augmented code in which all specified partial derivatives are computed in addition to the original result. For procedures coded in ANSI C, the ADIC tool (Bischof et al., 1997) implements AD as a source code transformation after the specification of dependent and independent variables. A recent and popular tool also utilizing this approach is Tapenade (Pascual and Hascoët, 2008; Hascoët and Pascual, 2013), implementing forward and reverse mode AD for Fortran and C programs. Tapenade itself is implemented in Java and can be run locally or as an online service.[31]

In addition to language extensions through source code transformation, there are implementations introducing new languages with tightly integrated AD capabilities through special-purpose compilers or interpreters. Some of the earliest AD tools such as SLANG (Adamson and Winant, 1969) and PROSE (Pfeiffer, 1987) belong to this category. The NAGWare Fortran 95 compiler (Naumann and Riehme, 2005) is a more recent example, where the use of AD-related extensions triggers automatic generation of derivative code at compile time.

As an example of interpreter-based implementation, the algebraic modeling language AMPL (Fourer et al., 2002) enables objectives and constraints to be expressed in mathematical notation, from which the system deduces active variables and arranges the necessary AD computations. Other examples in this category include the FM/FAD package (Mazourik, 1991), based on the Algol-like DIFALG language, and the object-oriented COSY language (Berz et al., 1996) similar to Pascal.

The Stalingrad compiler (Pearlmutter and Siskind, 2008; Siskind and Pearlmutter, 2008a), working on the Scheme-based AD-aware VLAD language, also falls under this category. The newer DVL compiler[32] is based on Stalingrad and uses a reimplementation of portions of the VLAD language.

Motivated by machine learning applications, the Tangent library (van Merriënboer et al., 2017) implements AD using source code transformation, and accepts numeric functions written in a syntactic subset of Python and Numpy.

30. Preprocessors transform program source code before it is given as an input to a compiler.
31. http://www-tapenade.inria.fr:8080/tapenade/index.jsp
32. https://github.com/axch/dysvunctional-language

5.3 Operator Overloading

In modern programming languages with polymorphic features, operator overloading provides the most straightforward way of implementing AD, exploiting the capability of redefining elementary operation semantics.

A popular tool implemented with operator overloading in C++ is ADOL-C (Walther and Griewank, 2012). ADOL-C requires the use of AD-enabled types for variables, and records arithmetic operations on variables in tape data structures, which can subsequently be "played back" during reverse mode AD computations. The Mxyzptlk package (Michelotti, 1990) is another example for C++ capable of computing arbitrary-order partial derivatives via forward propagation. The FADBAD++ library (Bendtsen and Stauning, 1996) implements AD for C++ using templates and operator overloading. For Python, the *ad* package[33] uses operator overloading to compute first- and second-order derivatives, while the newer autograd package[34] provides forward and reverse mode AD with support for higher-order derivatives.

For functional languages, examples include R6RS-AD[35] and the AD routines within the Scmutils library[36] for Scheme, the *ad* library[37] for Haskell, and DiffSharp[38] for F# and C#.

6. Conclusions

Backpropagation and gradient-based optimization are behind virtually all recent successes in machine learning, yielding state-of-the-art results in computer vision, speech recognition and synthesis, and machine translation. We expect these techniques to remain at the core of machine learning for the foreseeable future. Research in the field involves a rapid prototyping and development cycle for testing new models and ideas, using a collection of increasingly higher-quality machine learning frameworks. These frameworks are in the process of transition from coarse-grained (module level) backpropagation towards fine-grained, general-purpose AD, allowing models to be implemented as regular programs in general-purpose programming languages with differentiation as an integral part of the infrastructure. We strongly believe that general-purpose AD is the future of gradient-based machine learning and we expect it to become an indispensable tool in the machine learning toolbox.

It is an exciting time for working at the intersection of AD and machine learning, and there are many opportunities for bringing advanced techniques and expertise from AD literature to bear on machine learning problems. Techniques that have been developed by the AD community such as tape reduction and elimination (Naumann, 2004), fixed-point iterations (Christianson, 1994), utilizing sparsity by matrix coloring (Gebremedhin et al., 2009, 2013), and reverse AD checkpointing (Dauvergne and Hascoët, 2006) are just a few examples that can find potential use in machine learning for increasing performance, improving convergence of optimization, using hardware more efficiently, and even enabling

33. http://pythonhosted.org/ad/
34. https://github.com/HIPS/autograd
35. https://github.com/NUIM-BCL/R6RS-AD
36. http://groups.csail.mit.edu/mac/users/gjs/6946/refman.txt
37. http://hackage.haskell.org/package/ad
38. http://diffsharp.github.io

new types of machine learning models to be implemented. Similarly, exciting new AD modes like direct propagation of the inverse Jacobian (Srinivasan and Todorov, 2015) have emerged from the machine learning community, but have yet to be examined and formalized by the AD community.

An important direction for future work is to make use of nested AD techniques in machine learning, allowing differentiation to be nested arbitrarily deep with referential transparency (Siskind and Pearlmutter, 2008b; Pearlmutter and Siskind, 2008). Nested AD is highly relevant in hyperparameter optimization as it can effortlessly provide exact hypergradients, that is, derivatives of a training objective with respect to the hyperparameters of an optimization routine (Maclaurin et al., 2015; Baydin et al., 2018). Potential applications include Bayesian model selection (Rasmussen and Williams, 2006) and gradient-based tuning of Hamiltonian Monte Carlo step sizes and mass matrices (Salimans et al., 2015). Besides hyperparameters, models internally using higher-order derivatives constitute a straightforward usage case for nested AD. The Riemannian manifold Langevin and Hamiltonian Monte Carlo methods (Girolami and Calderhead, 2011) use higher-order derivative information to more closely track the information geometry of the sampled distribution for faster convergence and exploration. In neural networks, it is very natural to use nested derivatives in defining objective functions that take input transformations into account, such as the Tangent Prop method (Simard et al., 1998) for imposing invariance under a set of chosen transformations.

Acknowledgments

We thank the anonymous reviewers whose comments helped improve this manuscript. This work was supported, in part, by Science Foundation Ireland grant 09/IN.1/I2637, by the Army Research Laboratory, accomplished under Cooperative Agreement Number W911NF-10-2-0060, by the National Science Foundation under Grants 1522954-IIS and 1734938-IIS, and by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DOI/IBC) contract number D17PC00341. Any opinions, findings, views, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views, official policies, or endorsements, either expressed or implied, of the sponsors. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation herein.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

D. S. Adamson and C. W. Winant. A SLANG simulation of an initially strong shock wave downstream of an infinite area change. In *Proceedings of the Conference on Applications of Continuous-System Simulation Languages*, pages 231–40, 1969.

Naman Agarwal, Brian Bullins, and Elad Hazan. Second order stochastic optimization in linear time. Technical Report arXiv:1602.03943, arXiv preprint, 2016.

R. K. Al Seyab and Y. Cao. Nonlinear system identification for predictive control using continuous time recurrent neural networks and automatic differentiation. *Journal of Process Control*, 18(6):568–581, 2008. doi: 10.1016/j.jprocont.2007.10.012.

Brandon Amos and J Zico Kolter. OptNet: Differentiable optimization as a layer in neural networks. *arXiv preprint arXiv:1703.00443*, 2017.

Marianna S. Apostolopoulou, Dimitris G. Sotiropoulos, Ioannis E. Livieris, and Panagiotis Pintelas. A memoryless BFGS neural network training algorithm. In *7th IEEE International Conference on Industrial Informatics, INDIN 2009*, pages 216–221, June 2009. doi: 10.1109/INDIN.2009.5195806.

Andrew W Appel. Runtime tags aren't necessary. *Lisp and Symbolic Computation*, 2(2): 153–162, 1989.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Daniel Paul Barrett and Jeffrey Mark Siskind. Felzenszwalb-Baum-Welch: Event detection by changing appearance. *arXiv preprint arXiv:1306.4746*, 2013.

Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian Goodfellow, Arnaud Bergeron, Nicolas Bouchard, David Warde-Farley, and Yoshua Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

Friedrich L. Bauer. Computational graphs and rounding error. *SIAM Journal on Numerical Analysis*, 11(1):87–96, 1974.

Atılım Güneş Baydin, Barak A. Pearlmutter, and Jeffrey Mark Siskind. Diffsharp: An AD library for .NET languages. In *7th International Conference on Algorithmic Differentiation, Christ Church Oxford, UK, September 12–15, 2016*, 2016a. Also arXiv:1611.03423.

Atılım Güneş Baydin, Barak A. Pearlmutter, and Jeffrey Mark Siskind. Tricks from deep learning. In *7th International Conference on Algorithmic Differentiation, Christ Church Oxford, UK, September 12–15, 2016*, 2016b. Also arXiv:1611.03777.

Atılım Güneş Baydin, Robert Cornish, David Martínez Rubio, Mark Schmidt, and Frank Wood. Online learning rate adaptation with hypergradient descent. In *Sixth International Conference on Learning Representations (ICLR), Vancouver, Canada, April 30–May 3, 2018*, 2018.

L. M. Beda, L. N. Korolev, N. V. Sukkikh, and T. S. Frolova. Programs for automatic differentiation for the machine BESM (in Russian). Technical report, Institute for Precise Mechanics and Computation Techniques, Academy of Science, Moscow, USSR, 1959.

Bradley M. Bell and James V. Burke. Algorithmic differentiation of implicit functions and optimal values. In C. H. Bischof, H. M. Bücker, P. Hovland, U. Naumann, and J. Utke, editors, *Advances in Automatic Differentiation*, volume 64 of *Lecture Notes in Computational Science and Engineering*, pages 67–77. Springer Berlin Heidelberg, 2008. doi: 10.1007/978-3-540-68942-3_7.

Claus Bendtsen and Ole Stauning. FADBAD, a flexible C++ package for automatic differentiation. Technical Report IMM-REP-1996-17, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 1996.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

Charles W. Bert and Moinuddin Malik. Differential quadrature method in computational mechanics: A review. *Applied Mechanics Reviews*, 49, 1996. doi: 10.1115/1.3101882.

Martin Berz, Kyoko Makino, Khodr Shamseddine, Georg H. Hoffstätter, and Weishi Wan. COSY INFINITY and its applications in nonlinear dynamics. In M. Berz, C. Bischof, G. Corliss, and A. Griewank, editors, *Computational Differentiation: Techniques, Applications, and Tools*, pages 363–5. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1996.

Christian Bischof, Alan Carle, George Corliss, Andreas Griewank, and Paul Hovland. ADIFOR 2.0: Automatic differentiation of Fortran 77 programs. *Computational Science Engineering, IEEE*, 3(3):18–32, 1996. doi: 10.1109/99.537089.

Christian Bischof, Lucas Roh, and Andrew Mauer-Oats. ADIC: An extensible automatic differentiation tool for ANSI-C. *Software Practice and Experience*, 27(12):1427–56, 1997.

Christian H. Bischof, H. Martin Bücker, and Bruno Lang. Automatic differentiation for computational finance. In E. J. Kontoghiorghes, B. Rustem, and S. Siokos, editors, *Computational Methods in Decision-Making, Economics and Finance*, volume 74 of *Applied Optimization*, pages 297–310. Springer US, 2002. doi: 10.1007/978-1-4757-3613-7_15.

Christian H. Bischof, H. Martin Bücker, Arno Rasch, Emil Slusanschi, and Bruno Lang. Automatic differentiation of the general-purpose computational fluid dynamics package FLUENT. *Journal of Fluids Engineering*, 129(5):652–8, 2006. doi: 10.1115/1.2720475.

Christian H. Bischof, Paul D. Hovland, and Boyana Norris. On the implementation of automatic differentiation tools. *Higher-Order and Symbolic Computation*, 21(3):311–31, 2008. doi: 10.1007/s10990-008-9034-4.

V. G. Boltyanskii, R. V. Gamkrelidze, and L. S. Pontryagin. The theory of optimal processes I: The maximum principle. *Izvest. Akad. Nauk S.S.S.R. Ser. Mat.*, 24:3–42, 1960.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

Léon Bottou and Yann LeCun. SN: A simulator for connectionist models. In *Proceedings of NeuroNimes 88*, pages 371–382, Nimes, France, 1988. URL `http://leon.bottou.org/papers/bottou-lecun-88`.

Léon Bottou and Yann LeCun. Lush reference manual, 2002. URL `http://lush.sourceforge.net/doc.html`.

Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *arXiv preprint arXiv:1606.04838*, 2016.

Léon Bottou. Online learning and stochastic approximations. *On-Line Learning in Neural Networks*, 17:9, 1998.

Claude Brezinski and M. Redivo Zaglia. *Extrapolation Methods: Theory and Practice*. North-Holland, 1991.

A. E. Bryson and W. F. Denham. A steepest ascent method for solving optimum programming problems. *Journal of Applied Mechanics*, 29(2):247, 1962. doi: 10.1115/1.3640537.

Arthur E. Bryson and Yu-Chi Ho. *Applied Optimal Control: Optimization, Estimation, and Control*. Blaisdell, Waltham, MA, 1969.

Rirchard L. Burden and J. Douglas Faires. *Numerical Analysis*. Brooks/Cole, 2001.

Luca Capriotti. Fast Greeks by algorithmic differentiation. *Journal of Computational Finance*, 14(3):3, 2011.

Gregory R. Carmichael and Adrian Sandu. Sensitivity analysis for atmospheric chemistry models via automatic differentiation. *Atmospheric Environment*, 31(3):475–89, 1997.

Bob Carpenter, Matthew D Hoffman, Marcus Brubaker, Daniel Lee, Peter Li, and Michael Betancourt. The Stan math library: Reverse-mode automatic differentiation in C++. *arXiv preprint arXiv:1509.07164*, 2015.

Bob Carpenter, Andrew Gelman, Matt Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Michael A Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software*, 20:1–37, 2016.

Daniele Casanova, Robin S. Sharp, Mark Final, Bruce Christianson, and Pat Symonds. Application of automatic diffentiation to race car performance optimisation. In George Corliss, Christèle Faure, Andreas Griewank, Lauren Hascoët, and Uwe Naumann, editors, *Automatic Differentiation of Algorithms*, pages 117–124. Springer-Verlag New York, Inc., New York, NY, USA, 2002. ISBN 0-387-95305-1.

Isabelle Charpentier and Mohammed Ghemires. Efficient adjoint derivatives: Application to the meteorological model Meso-NH. *Optimization Methods and Software*, 13(1):35–63, 2000.

Danqi Chen and Christopher Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, 2014.

Sharan Chetlur, Cliff Woolley, Philippe Vandermersch, Jonathan Cohen, John Tran, Bryan Catanzaro, and Evan Shelhamer. cuDNN: Efficient primitives for deep learning. *arXiv preprint arXiv:1410.0759*, 2014.

Siddhartha Chib and Edward Greenberg. Understanding the Metropolis-Hastings algorithm. *The American Statistician*, 49(4):327–335, 1995. doi: 10.1080/00031305.1995.10476177.

Bruce Christianson. Reverse accumulation and attractive fixed points. *Optimization Methods and Software*, 3(4):311–326, 1994.

Bruce Christianson. A Leibniz notation for automatic differentiation. In Shaun Forth, Paul Hovland, Eric Phipps, Jean Utke, and Andrea Walther, editors, *Recent Advances in Algorithmic Differentiation*, volume 87 of *Lecture Notes in Computational Science and Engineering*, pages 1–9. Springer, Berlin, 2012. ISBN 978-3-540-68935-5. doi: 10.1007/978-3-642-30023-3_1.

William K. Clifford. Preliminary sketch of bi-quaternions. *Proceedings of the London Mathematical Society*, 4:381–95, 1873.

J Cohen and M Jeroen Molemaker. A fast double precision cfd code using cuda. *Parallel Computational Fluid Dynamics: Recent Advances and Future Directions*, pages 414–429, 2009.

Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. Torch7: A Matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, number EPFL-CONF-192376, 2011.

George F. Corliss. *Application of differentiation arithmetic*, volume 19 of *Perspectives in Computing*, pages 127–48. Academic Press, Boston, 1988.

Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in Neural Information Processing Systems*, pages 3123–3131, 2015.

Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pages 886–93, Washington, DC, USA, 2005. IEEE Computer Society. doi: 10.1109/CVPR.2005.177.

Benjamin Dauvergne and Laurent Hascoët. The data-flow equations of checkpointing in reverse automatic differentiation. In V. N. Alexandrov, G. D. van Albada, P. M. A. Sloot, and J. Dongarra, editors, *Computational Science – ICCS 2006*, volume 3994 of *Lecture Notes in Computer Science*, pages 566–73, Dauvergne, 2006. Springer Berlin.

John E. Dennis and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphia, 1996.

L. C. Dixon. Use of automatic differentiation for calculating Hessians and Newton steps. In A. Griewank and G. F. Corliss, editors, *Automatic Differentiation of Algorithms: Theory, Implementation, and Application*, pages 114–125. SIAM, Philadelphia, PA, 1991.

Simon Duane, Anthony D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, 1987.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul): 2121–2159, 2011.

Ulf Ekström, Lucas Visscher, Radovan Bast, Andreas J. Thorvaldsen, and Kenneth Ruud. Arbitrary-order density functional response theory from automatic differentiation. *Journal of Chemical Theory and Computation*, 6:1971–80, 2010. doi: 10.1021/ct100117s.

Jerry Eriksson, Mårten Gulliksson, Per Lindström, and Per Åke Wedin. Regularization tools for training large feed-forward neural networks using automatic differentiation. *Optimization Methods and Software*, 10(1):49–69, 1998. doi: 10.1080/10556789808805701.

S. M. Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Koray Kavukcuoglu, and Geoffrey E. Hinton. Attend, infer, repeat: Fast scene understanding with generative models. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3225–3233. Curran Associates, Inc., 2016.

Jenny Rose Finkel, Alex Kleeman, and Christopher D. Manning. Efficient, feature-based, conditional random field parsing. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL 2008)*, pages 959–67, 2008.

Bengt Fornberg. Numerical differentiation of analytic functions. *ACM Transactions on Mathematical Software*, 7(4):512–26, 1981. doi: 10.1145/355972.355979.

Shaun A. Forth. An efficient overloaded implementation of forward mode automatic differentiation in MATLAB. *ACM Transactions on Mathematical Software*, 32(2):195–222, 2006.

Shaun A. Forth and Trevor P. Evans. Aerofoil optimisation via AD of a multigrid cell-vertex Euler flow solver. In George Corliss, Christèle Faure, Andreas Griewank, Laurent Hascoët, and Uwe Naumann, editors, *Automatic Differentiation of Algorithms: From Simulation to Optimization*, pages 153–160. Springer New York, New York, NY, 2002. ISBN 978-1-4613-0075-5. doi: 10.1007/978-1-4613-0075-5_17.

Robert Fourer, David M. Gay, and Brian W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. Duxbury Press, 2002.

David M. Gay. Automatically finding and exploiting partially separable structure in nonlinear programming problems. Technical report, Bell Laboratories, Murray Hill, NJ, 1996.

Assefaw H. Gebremedhin, Arijit Tarafdar, Alex Pothen, and Andrea Walther. Efficient computation of sparse Hessians using coloring and automatic differentiation. *INFORMS Journal on Computing*, 21(2):209–23, 2009. doi: 10.1287/ijoc.1080.0286.

Assefaw H Gebremedhin, Duc Nguyen, Md Mostofa Ali Patwary, and Alex Pothen. ColPack: Software for graph coloring and related problems in scientific computing. *ACM Transactions on Mathematical Software (TOMS)*, 40(1):1, 2013.

Samuel Gershman and Noah Goodman. Amortized inference in probabilistic reasoning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, number 36, 2014.

Ralf Giering and Thomas Kaminski. Recipes for adjoint code construction. *ACM Transactions on Mathematical Software*, 24:437–74, 1998. doi: 10.1145/293686.293695.

Kevin Gimpel, Dipanjan Das, and Noah A. Smith. Distributed asynchronous online learning for natural language processing. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, CoNLL '10, pages 213–222, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

Mark Girolami and Be Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

Yoav Goldberg. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420, 2016.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

Andrew D Gordon, Thomas A Henzinger, Aditya V Nori, and Sriram K Rajamani. Probabilistic programming. In *Proceedings of the on Future of Software Engineering*, pages 167–181. ACM, 2014.

Johannes Grabmeier and Erich Kaltofen. *Computer Algebra Handbook: Foundations, Applications, Systems*. Springer, 2003.

Markus Grabner, Thomas Pock, Tobias Gross, and Bernhard Kainz. Automatic differentiation for GPU-accelerated 2D/3D registration. In C. H. Bischof, H. M. Bücker, P. Hovland, U. Naumann, and J. Utke, editors, *Advances in Automatic Differentiation*, volume 64 of *Lecture Notes in Computational Science and Engineering*, pages 259–269. Springer Berlin Heidelberg, 2008. doi: 10.1007/978-3-540-68942-3_23.

Will Grathwohl, Dami Choi, Yuhuai Wu, Geoff Roeder, and David Duvenaud. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. *arXiv preprint arXiv:1711.00123*, 2017.

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural Turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476, 2016.

Edward Grefenstette, Karl Moritz Hermann, Mustafa Suleyman, and Phil Blunsom. Learning to transduce with unbounded memory. In *Advances in Neural Information Processing Systems*, pages 1828–1836, 2015.

Andreas Griewank. On automatic differentiation. In M. Iri and K. Tanabe, editors, *Mathematical Programming: Recent Developments and Applications*, pages 83–108. Kluwer Academic Publishers, 1989.

Andreas Griewank. A mathematical view of automatic differentiation. *Acta Numerica*, 12: 321–98, 2003. doi: 10.1017/S0962492902000132.

Andreas Griewank. Who invented the reverse mode of differentiation? *Documenta Mathematica*, Extra Volume ISMP:389–400, 2012.

Andreas Griewank and Andrea Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Society for Industrial and Applied Mathematics, Philadelphia, 2008. doi: 10.1137/1.9780898717761.

Andreas Griewank, Kshitij Kulshreshtha, and Andrea Walther. On the numerical stability of algorithmic differentiation. *Computing*, 94(2-4):125–149, 2012.

Audrunas Gruslys, Rémi Munos, Ivo Danihelka, Marc Lanctot, and Alex Graves. Memory-efficient backpropagation through time. In *Advances in Neural Information Processing Systems*, pages 4125–4133, 2016.

Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1737–1746, 2015.

Gundolf Haase, Ulrich Langer, Ewald Lindner, and Wolfram Mühlhuber. Optimal sizing of industrial structural mechanics problems using AD. In *Automatic Differentiation of Algorithms*, pages 181–188. Springer, 2002.

Stefan Hadjis, Firas Abuzaid, Ce Zhang, and Christopher Ré. Caffe con troll: Shallow ideas to speed up deep learning. In *Proceedings of the Fourth Workshop on Data analytics in the Cloud*, page 2. ACM, 2015.

William Rowan Hamilton. Theory of conjugate functions, or algebraic couples; with a preliminary and elementary essay on algebra as the science of pure time. *Transactions of the Royal Irish Academy*, 17:293–422, 1837.

Laurent Hascoët and Valérie Pascual. The Tapenade automatic differentiation tool: principles, model, and specification. *ACM Transactions on Mathematical Software*, 39(3), 2013. doi: 10.1145/2450153.2450158.

Robert Hecht-Nielsen. Theory of the backpropagation neural network. In *International Joint Conference on Neural Networks, IJCNN 1989*, pages 593–605. IEEE, 1989.

Ruth L. Hinkins. Parallel computation of automatic differentiation applied to magnetic field calculations. Technical report, Lawrence Berkeley Lab., CA, 1994.

Geoffrey E. Hinton and Zoubin Ghahramani. Generative models for discovering sparse distributed representations. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 352(1358):1177–1190, 1997.

Matthew D. Hoffman and Andrew Gelman. The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15: 1351–1381, 2014.

Berthold K. P. Horn. Understanding image intensities. *Artificial Intelligence*, 8:201–231, 1977.

Jim E. Horwedel, Brian A. Worley, E. M. Oblow, and F. G. Pin. GRESS version 1.0 user's manual. Technical Memorandum ORNL/TM 10835, Martin Marietta Energy Systems, Inc., Oak Ridge National Laboratory, Oak Ridge, 1988.

Max E. Jerrell. Automatic differentiation and interval arithmetic for estimation of disequilibrium models. *Computational Economics*, 10(3):295–316, 1997.

Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

Matthew Johnson, David K Duvenaud, Alex Wiltschko, Ryan P Adams, and Sandeep R Datta. Composing graphical models with neural networks for structured representations and fast inference. In *Advances in Neural Information Processing Systems*, pages 2946–2954, 2016.

Neil D Jones, Carsten K Gomard, and Peter Sestoft. *Partial evaluation and automatic program generation*. Peter Sestoft, 1993a.

Simon L Peyton Jones and John Launchbury. Unboxed values as first class citizens in a non-strict functional language. In *Conference on Functional Programming Languages and Computer Architecture*, pages 636–666. Springer, 1991.

SL Peyton Jones, Cordy Hall, Kevin Hammond, Will Partain, and Philip Wadler. The Glasgow Haskell compiler: a technical overview. In *Proc. UK Joint Framework for Information Technology (JFIT) Technical Conference*, volume 93, 1993b.

Armand Joulin and Tomas Mikolov. Inferring algorithmic patterns with stack-augmented recurrent nets. In *Advances in Neural Information Processing Systems*, pages 190–198, 2015.

David W. Juedes. A taxonomy of automatic differentiation tools. In A. Griewank and G. F. Corliss, editors, *Automatic Differentiation of Algorithms: Theory, Implementation, and Application*, pages 315–29. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1991.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *The International Conference on Learning Representations (ICLR), San Diego*, 2015.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

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

K. Kubo and M. Iri. PADRE2, version 1—user's manual. Research Memorandum RMI 90-01, Department of Mathematical Engineering and Information Physics, University of Tokyo, Tokyo, 1990.

Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M. Blei. Automatic differentiation variational inference. *Journal of Machine Learning Research*, 18(14):1–45, 2017.

Tejas D. Kulkarni, Pushmeet Kohli, Joshua B. Tenenbaum, and Vikash Mansinghka. Picture: A probabilistic programming language for scene perception. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1378–1387, New York, New York, USA, 20–22 Jun 2016. PMLR.

C. L. Lawson. Computing derivatives using W-arithmetic and U-arithmetic. Internal Computing Memorandum CM-286, Jet Propulsion Laboratory, Pasadena, CA, 1971.

Tuan Anh Le, Atılım Güneş Baydin, and Frank Wood. Inference compilation and universal probabilistic programming. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54 of *Proceedings of Machine Learning Research*, pages 1338–1348, Fort Lauderdale, FL, USA, 2017. PMLR.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

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

G. W. Leibniz. *Machina arithmetica in qua non additio tantum et subtractio sed et multiplicatio nullo, diviso vero paene nullo animi labore peragantur*. Hannover, 1685.

Xavier Leroy. The effectiveness of type-based unboxing. In *TIC 1997: Workshop Types in Compilation*, 1997.

Seppo Linnainmaa. The representation of the cumulative rounding error of an algorithm as a taylor expansion of the local rounding errors. Master's thesis, University of Helsinki, 1970.

Seppo Linnainmaa. Taylor expansion of the accumulated rounding error. *BIT Numerical Mathematics*, 16(2):146–160, 1976.

Matthew M. Loper and Michael J. Black. OpenDR: An approximate differentiable renderer. In *European Conference on Computer Vision*, pages 154–169. Springer, 2014.

Dougal Maclaurin. *Modeling, Inference and Optimization with Composable Differentiable Procedures*. PhD thesis, School of Engineering and Applied Sciences, Harvard University, 2016.

Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *International Conference on Machine Learning*, pages 2113–2122, 2015.

Oleksandr Manzyuk, Barak A. Pearlmutter, Alexey Andreyevich Radul, David R Rush, and Jeffrey Mark Siskind. Confusion of tagged perturbations in forward automatic differentiation of higher-order functions. *arXiv preprint arXiv:1211.4892*, 2012.

David Q. Mayne and David H. Jacobson. *Differential Dynamic Programming*. American Elsevier Pub. Co., New York, 1970.

Vladimir Mazourik. Integration of automatic differentiation into a numerical library for PC's. In A. Griewank and G. F. Corliss, editors, *Automatic Differentiation of Algorithms: Theory, Implementation, and Application*, pages 315–29. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1991.

Renate Meyer, David A. Fournier, and Andreas Berg. Stochastic volatility: Bayesian computation using automatic differentiation and the extended Kalman filter. *Econometrics Journal*, 6(2):408–420, 2003. doi: 10.1111/1368-423X.t01-1-00116.

L. Michelotti. MXYZPTLK: A practical, user-friendly C++ implementation of differential algebra: User's guide. Technical Memorandum FN-535, Fermi National Accelerator Laboratory, Batavia, IL, 1990.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

J. D. Müller and P. Cusdin. On the performance of discrete adjoint CFD codes using automatic differentiation. *International Journal for Numerical Methods in Fluids*, 47 (8-9):939–945, 2005. ISSN 1097-0363. doi: 10.1002/fld.885.

Uwe Naumann. Optimal accumulation of Jacobian matrices by elimination methods on the dual computational graph. *Mathematical Programming*, 99(3):399–421, 2004.

Uwe Naumann and Jan Riehme. Computing adjoints with the NAGWare Fortran 95 compiler. In H. M. Bücker, G. Corliss, P. Hovland, U. Naumann, and B. Norris, editors, *Automatic Differentiation: Applications, Theory, and Implementations*, Lecture Notes in Computational Science and Engineering, pages 159–69. Springer, 2005.

Radford M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto, 1993.

Richard D. Neidinger. Automatic differentiation and APL. *College Mathematics Journal*, 20(3):238–51, 1989. doi: 10.2307/2686776.

John F. Nolan. Analytical differentiation on a digital computer. Master's thesis, Massachusetts Institute of Technology, 1953.

J. F. Ostiguy and L. Michelotti. Mxyzptlk: An efficient, native C++ differentiation engine. In *Particle Accelerator Conference (PAC 2007)*, pages 3489–91. IEEE, 2007. doi: 10.1109/PAC.2007.4440468.

David B. Parker. Learning-logic: Casting the cortex of the human brain in silicon. Technical Report TR-47, Center for Computational Research in Economics and Management Science, MIT, 1985.

Valérie Pascual and Laurent Hascoët. TAPENADE for C. In *Advances in Automatic Differentiation*, Lecture Notes in Computational Science and Engineering, pages 199–210. Springer, 2008. doi: 10.1007/978-3-540-68942-3_18.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS 2017 Autodiff Workshop: The Future of Gradient-based Machine Learning Software and Techniques, Long Beach, CA, US, December 9, 2017*, 2017.

Barak A. Pearlmutter. Fast exact multiplication by the Hessian. *Neural Computation*, 6: 147–60, 1994. doi: 10.1162/neco.1994.6.1.147.

Barak A. Pearlmutter and Jeffrey Mark Siskind. Reverse-mode AD in a functional framework: Lambda the ultimate backpropagator. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 30(2):1–36, March 2008. doi: 10.1145/1330017.1330018.

Ding-Yu Peng and Donald B. Robinson. A new two-constant equation of state. *Industrial and Engineering Chemistry Fundamentals*, 15(1):59–64, 1976. doi: 10.1021/i160057a011.

John Peterson. Untagged data in tagged environments: Choosing optimal representations at compile time. In *Proceedings of the Fourth International Conference on Functional Programming Languages and Computer Architecture*, pages 89–99. ACM, 1989.

F. W. Pfeiffer. Automatic differentiation in PROSE. *SIGNUM Newsletter*, 22(1):2–8, 1987. doi: 10.1145/24680.24681.

Thomas Pock, Michael Pock, and Horst Bischof. Algorithmic differentiation: Application to variational problems in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1180–1193, 2007. doi: 10.1109/TPAMI.2007.1044.

William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 2007.

Louise B. Rall. Perspectives on automatic differentiation: Past, present, and future? In M. Bücker, G. Corliss, U. Naumann, P. Hovland, and B. Norris, editors, *Automatic Differentiation: Applications, Theory, and Implementations*, volume 50 of *Lecture Notes in Computational Science and Engineering*, pages 1–14. Springer Berlin Heidelberg, 2006.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

J. Revels, M. Lubin, and T. Papamarkou. Forward-mode automatic differentiation in Julia. *arXiv:1607.07892 [cs.MS]*, 2016a. URL `https://arxiv.org/abs/1607.07892`.

Jarrett Revels, Miles Lubin, and Theodore Papamarkou. Forward-mode automatic differentiation in Julia. *arXiv preprint arXiv:1607.07892*, 2016b.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286, 2014.

Lawrence C. Rich and David R. Hill. Automatic differentiation in MATLAB. *Applied Numerical Mathematics*, 9:33–43, 1992.

Daniel Ritchie, Paul Horsfall, and Noah D Goodman. Deep amortized inference for probabilistic programs. *arXiv preprint arXiv:1610.05735*, 2016.

Elizabeth Rollins. Optimization of neural network feedback control systems using automatic differentiation. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 2009.

L. I. Rozonoer. L. S. Pontryagin's maximum principle in the theory of optimum systems—Part II. *Automat. i Telemekh.*, 20:1441–1458, 1959.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533, 1986.

Siegfried M. Rump. INTLAB—INTerval LABoratory. In *Developments in Reliable Computing*, pages 77–104. Kluwer Academic Publishers, Dordrecht, 1999. doi: 10.1007/978-94-017-1247-7_7.

Tim Salimans, Diederik Kingma, and Max Welling. Markov chain Monte Carlo and variational inference: Bridging the gap. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1218–1226, 2015.

John Salvatier, Thomas V Wiecki, and Christopher Fonnesbeck. Probabilistic programming in Python using PyMC3. *PeerJ Computer Science*, 2:e55, 2016.

Tom Schaul, Sixin Zhang, and Yann LeCun. No more pesky learning rates. In *International Conference on Machine Learning*, pages 343–351, 2013.

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

Nicol N. Schraudolph. Local gain adaptation in stochastic gradient descent. In *Proceedings of the International Conference on Artificial Neural Networks*, pages 569–74, Edinburgh, Scotland, 1999. IEE London. doi: 10.1049/cp:19991170.

Nicol N. Schraudolph and Thore Graepel. Combining conjugate direction methods with stochastic approximation of gradients. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, 2003.

Frank Seide and Amit Agarwal. CNTK: Microsoft's open-source deep-learning toolkit. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 2135–2135, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2945397.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations 2017*, 2017.

Olin Shivers. *Control-flow analysis of higher-order languages*. PhD thesis, Carnegie Mellon University, 1991.

Alex Shtof, Alexander Agathos, Yotam Gingold, Ariel Shamir, and Daniel Cohen-Or. Geosemantic snapping for sketch-based modeling. *Computer Graphics Forum*, 32(2):245–53, 2013. doi: 10.1111/cgf.12044.

N. Siddharth, Brooks Paige, Jan-Willem van de Meent, Alban Desmaison, Noah D. Goodman, Pushmeet Kohli, Frank Wood, and Philip Torr. Learning disentangled representations with semi-supervised deep generative models. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5927–5937. Curran Associates, Inc., 2017.

Patrice Simard, Yann LeCun, John Denker, and Bernard Victorri. Transformation invariance in pattern recognition, tangent distance and tangent propagation. In G. Orr and K. Muller, editors, *Neural Networks: Tricks of the Trade*. Springer, 1998.

Z. Sirkes and E. Tziperman. Finite difference of adjoint or adjoint of finite difference? *Monthly Weather Review*, 125(12):3373–8, 1997. doi: 10.1175/1520-0493(1997)125⟨3373:FDOAOA⟩2.0.CO;2.

Jeffrey Mark Siskind and Barak A. Pearlmutter. Perturbation confusion and referential transparency: Correct functional implementation of forward-mode AD. In Andrew Butterfield, editor, *Implementation and Application of Functional Languages—17th International Workshop, IFL'05*, pages 1–9, Dublin, Ireland, 2005. Trinity College Dublin Computer Science Department Technical Report TCD-CS-2005-60.

Jeffrey Mark Siskind and Barak A. Pearlmutter. Using polyvariant union-free flow analysis to compile a higher-order functional-programming language with a first-class derivative operator to efficient Fortran-like code. Technical Report TR-ECE-08-01, School of Electrical and Computer Engineering, Purdue University, 2008a.

Jeffrey Mark Siskind and Barak A. Pearlmutter. Nesting forward-mode AD in a functional framework. *Higher-Order and Symbolic Computation*, 21(4):361–376, 2008b.

Jeffrey Mark Siskind and Barak A. Pearlmutter. Efficient implementation of a higher-order language with built-in AD. In *7th International Conference on Algorithmic Differentiation, Christ Church Oxford, UK, September 12–15, 2016*, 2016. Also arXiv:1611.03416.

Jeffrey Mark Siskind and Barak A. Pearlmutter. Divide-and-conquer checkpointing for arbitrary programs with no user annotation. In *NIPS 2017 Autodiff Workshop: The Future of Gradient-based Machine Learning Software and Techniques, Long Beach, CA, US, December 9, 2017*, 2017. Also arXiv:1708.06799.

Emil I. Slusanschi and Vlad Dumitrel. ADiJaC—Automatic differentiation of Java classfiles. *ACM Transaction on Mathematical Software*, 43(2):9:1–9:33, September 2016. ISSN 0098-3500. doi: 10.1145/2904901.

Bert Speelpenning. *Compiling Fast Partial Derivatives of Functions Given by Algorithms*. PhD thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 1980.

Suvrit Sra, Sebastian Nowozin, and Stephen J. Wright. *Optimization for Machine Learning*. MIT Press, 2011.

Filip Srajer, Zuzana Kukelova, and Andrew Fitzgibbon. A benchmark of selected algorithmic differentiation tools on some problems in machine learning and computer vision. In *AD2016: The 7th International Conference on Algorithmic Differentiation, Monday 12th–Thursday 15th September 2016, Christ Church Oxford, UK: Programme and Abstracts*, pages 181–184. Society for Industrial and Applied Mathematics (SIAM), 2016.

Akshay Srinivasan and Emanuel Todorov. Graphical Newton. Technical Report arXiv:1508.00952, arXiv preprint, 2015.

Andreas Stuhlmüller, Jacob Taylor, and Noah Goodman. Learning stochastic inverses. In *Advances in Neural Information Processing Systems*, pages 3048–3056, 2013.

Felipe Petroski Such, Vashisht Madhavan, Edoardo Conti, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. *arXiv preprint arXiv:1712.06567*, 2017.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *Advances in Neural Information Processing Systems*, pages 2440–2448, 2015.

Gerald J. Sussman and Jack Wisdom. *Structure and Interpretation of Classical Mechanics*. MIT Press, 2001. doi: 10.1063/1.1457268.

Jonathan Taylor, Richard Stebbing, Varun Ramakrishna, Cem Keskin, Jamie Shotton, Shahram Izadi, Aaron Hertzmann, and Andrew Fitzgibbon. User-specific hand modeling from monocular depth sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 644–651, 2014.

Jeffrey P. Thomas, Earl H. Dowell, and Kenneth C. Hall. Using automatic differentiation to create a nonlinear reduced order model of a computational fluid dynamic solver. *AIAA Paper*, 7115:2006, 2006.

T. Tieleman and G. Hinton. Lecture 6.5—RMSProp: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4 (2), 2012.

Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.

Dustin Tran, Alp Kucukelbir, Adji B. Dieng, Maja Rudolph, Dawen Liang, and David M. Blei. Edward: A library for probabilistic modeling, inference, and criticism. *arXiv preprint arXiv:1610.09787*, 2016.

Dustin Tran, Matthew D. Hoffman, Rif A. Saurous, Eugene Brevdo, Kevin Murphy, and David M. Blei. Deep probabilistic programming. In *International Conference on Learning Representations*, 2017.

Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment—a modern synthesis. In *International Workshop on Vision Algorithms*, pages 298–372. Springer, 1999.

George Tucker, Andriy Mnih, Chris J. Maddison, John Lawson, and Jascha Sohl-Dickstein. REBAR: Low-variance, unbiased gradient estimates for discrete latent variable models. In *Advances in Neural Information Processing Systems*, pages 2624–2633, 2017.

Bart van Merriënboer, Alexander B. Wiltschko, and Dan Moldovan. Tangent: Automatic differentiation using source code transformation in Python. *arXiv preprint arXiv:1711.02712*, 2017.

Arun Verma. An introduction to automatic differentiation. *Current Science*, 78(7):804–7, 2000.

S. V. N. Vishwanathan, Nicol N. Schraudolph, Mark W. Schmidt, and Kevin P. Murphy. Accelerated training of conditional random fields with stochastic gradient methods. In

Proceedings of the 23rd International Conference on Machine Learning (ICML '06), pages 969–76, 2006. doi: 10.1145/1143844.1143966.

Andrea Walther. Automatic differentiation of explicit Runge-Kutta methods for optimal control. *Computational Optimization and Applications*, 36(1):83–108, 2007. doi: 10.1007/s10589-006-0397-3.

Andrea Walther and Andreas Griewank. Getting started with ADOL-C. In U. Naumann and O. Schenk, editors, *Combinatorial Scientific Computing*, chapter 7, pages 181–202. Chapman-Hall CRC Computational Science, 2012. doi: 10.1201/b11644-8.

Robert E. Wengert. A simple automatic derivative evaluation program. *Communications of the ACM*, 7:463–4, 1964.

Paul J. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, 1974.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, 1992.

J. Willkomm and A. Vehreschild. The ADiMat handbook, 2013. URL `http://adimat.sc.informatik.tu-darmstadt.de/doc/`.

David Wingate, Noah Goodman, Andreas Stuhlmüller, and Jeffrey Mark Siskind. Nonstandard interpretations of probabilistic programs for efficient inference. *Advances in Neural Information Processing Systems*, 23, 2011.

Weiwei Yang, Yong Zhao, Li Yan, and Xiaoqian Chen. Application of PID controller based on BP neural network using automatic differentiation method. In F. Sun, J. Zhang, Y. Tan, J. Cao, and W. Yu, editors, *Advances in Neural Networks—ISNN 2008*, volume 5264 of *Lecture Notes in Computer Science*, pages 702–711. Springer Berlin Heidelberg, 2008. doi: 10.1007/978-3-540-87734-9_80.

Ilker Yildirim, Tejas D. Kulkarni, Winrich A. Freiwald, and Joshua B. Tenenbaum. Efficient and robust analysis-by-synthesis in vision: A computational framework, behavioral tests, and modeling neuronal representations. In *Annual Conference of the Cognitive Science Society*, 2015.

Haonan Yu and Jeffrey Mark Siskind. Grounded language learning from video described with sentences. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 53–63, Sofia, Bulgaria, 2013. Association for Computational Linguistics.

Wojciech Zaremba, Tomas Mikolov, Armand Joulin, and Rob Fergus. Learning simple algorithms from examples. In *International Conference on Machine Learning*, pages 421–429, 2016.

Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):550–60, 1997. doi: 10.1145/279232.279236.

Journal of Machine Learning Research 18 (2018) 1-28 Submitted 4/15; Revised 8/15; Published 4/18

Normal Bandits of Unknown Means and Variances

Wesley Cowan CWCOWAN@MATH.RUTGERS.EDU
Department of Mathematics
Rutgers University
110 Frelinghuysen Rd., Piscataway, NJ 08854, USA

Junya Honda HONDA@IT.K.U-TOKYO.AC.JP
Department of Complexity Science and Engineering
Graduate School of Frontier Sciences, The University of Tokyo
5-1-5 Kashiwanoha, Kashiwa-shi, Chiba 277-8561, Japan

Michael N. Katehakis MNK@RUTGERS.EDU
Department of Management Science and Information Systems
Rutgers University
100 Rockafeller Rd., Piscataway, NJ 08854, USA

Editor: Csaba Szepesvári

Abstract

Consider the problem of sampling sequentially from a finite number of $N \geqslant 2$ populations, specified by random variables X_k^i, $i = 1,\ldots,N$, and $k = 1,2,\ldots$; where X_k^i denotes the outcome from population i the k^{th} time it is sampled. It is assumed that for each fixed i, $\{X_k^i\}_{k\geqslant 1}$ is a sequence of i.i.d. normal random variables, with unknown mean μ_i and unknown variance σ_i^2. The objective is to have a policy π for deciding from which of the N populations to sample from at any time $t = 1,2,\ldots$ so as to maximize the expected sum of outcomes of n total samples or equivalently to minimize the regret due to lack on information of the parameters μ_i and σ_i^2. In this paper, we present a simple inflated sample mean (ISM) index policy that is asymptotically optimal in the sense of Theorem 4 below. This resolves a standing open problem from Burnetas and Katehakis (1996b). Additionally, finite horizon regret bounds are given.

Keywords: Inflated Sample Means, UCB policies, Multi-armed Bandits, Sequential Allocation

1. Introduction and Summary

Consider the problem of a controller sampling sequentially from a finite number of $N \geqslant 2$ populations or 'bandits', where the measurements from population i are specified by a sequence of i.i.d. random variables $\{X_k^i\}_{k\geqslant 1}$, taken to be normal with finite mean μ_i and finite variance σ_i^2. The means $\{\mu_i\}$ and variances $\{\sigma_i^2\}$ are taken to be unknown to the controller. It is convenient to define the maximum mean, $\mu^* = \max_i\{\mu_i\}$, and the bandit discrepancies $\{\Delta_i\}$ where $\Delta_i = \mu^* - \mu_i \geqslant 0$. It is additionally convenient to define σ_*^2 as the minimal variance of any bandit that achieves μ^*, that is $\sigma_*^2 = \min_{i:\mu_i=\mu^*}\sigma_i^2$.

In this paper, given k samples from population i we will take the estimators: $\bar{X}_k^i = \sum_{t=1}^k X_t^i/k$ and $S_i^2(k) = \sum_{t=1}^k \left(X_t^i - \bar{X}_k^i\right)^2/k$ for μ_i and σ_i^2 respectively. Note that the use of the biased estimator for the variance, with the $1/k$ factor in place of $1/(k-1)$, is largely for aesthetic purposes - the results presented here adapt to the use of the unbiased estimator as well.

For any adaptive, non-anticipatory policy π, $\pi(t) = i$ indicates that the controller samples bandit i at time t. Define $T_\pi^i(n) = \sum_{t=1}^n \mathbb{1}\{\pi(t) = i\}$, denoting the number of times bandit i has been sampled during the periods $t = 1,\ldots,n$ under policy π; we take, as a convenience, $T_\pi^i(0) = 0$ for all i,π. The *value* of a policy π is the expected sum of the first n outcomes under π, which we define to be the function $V_\pi(n)$:

$$V_\pi(n) = \mathbb{E}\left[\sum_{i=1}^N \sum_{k=1}^{T_\pi^i(n)} X_k^i\right] = \sum_{i=1}^N \mu_i \mathbb{E}\left[T_\pi^i(n)\right], \tag{1}$$

where for simplicity the dependence of $V_\pi(n)$ on the true, unknown, values of the parameters $\underline{\mu} = (\mu_1,\ldots,\mu_N)$ and $\underline{\sigma}^2 = (\sigma_1^2,\ldots,\sigma_N^2)$, is supressed. The *pseudo-regret*, or simply *regret*, of a policy is taken to be the expected loss due to ignorance of the parameters $\underline{\mu}$ and $\underline{\sigma}^2$ by the controller. Had the controller complete information, she would at every round activate some bandit i^* such that $\mu_{i^*} = \mu^* = \max_i\{\mu_i\}$. For a given policy π, we define the expected regret of that policy at time n as

$$R_\pi(n) = n\mu^* - V_\pi(n) = \sum_{i=1}^n \Delta_i \mathbb{E}\left[T_\pi^i(n)\right]. \tag{2}$$

It follows from Eqs. (1) and (2) that maximization of $V_\pi(n)$ with respect to π is equivalent to minimization of $R_\pi(n)$. This type of loss due to ignorance of the means (regret) was first introduced in the context of an $N = 2$ problem by Robbins (1952) as the 'loss per trial' $L_\pi(n)/n = \mu^* - \sum_{i=1}^N \sum_{k=1}^{T_\pi^i(n)} X_k^i/n$ (for which $R_\pi(n) = \mathbb{E}[L_\pi(n)]$). Robbins constructed a modified (along two sparse sequences) 'play the winner' policy, π_R, such that for all choices of bandit parameters, $L_{\pi_R}(n) = o(n)$ (a.s.) and $R_{\pi_R}(n) = o(n)$, using for his derivation only the assumption of the Strong Law of Large Numbers. Following Burnetas and Katehakis (1996b) when $n \to \infty$, if π is such that $R_\pi(n) = o(n)$ for all choices of bandit parameters, we say policy π is ***uniformly convergent*** (UC) (since then $\lim_{n\to\infty} V_\pi(n)/n = \mu^*$). However, if under a policy π, $R_\pi(n)$ grew at a slower pace, such as $R_\pi(n) = o(n^{1/2})$, or better $R_\pi(n) = o(n^{1/100})$ etc., then the controller would be assured that π is making an effective trade-off between exploration and exploitation. It turns out that it is possible to construct ***'uniformly fast convergent'*** (UFC) policies, also known as *consistent* or *strongly consistent*, defined

as the policies π for which:

$$R_\pi(n) = o(n^\alpha), \text{ for all } \alpha > 0 \text{ for all } (\underline{\mu}, \underline{\sigma}^2).$$

For clarification, it is worth noting here that while a naive policy such as 'always activate bandit 1' will have 0 regret for some choices of $(\underline{\mu}, \underline{\sigma}^2)$ (in particular, those for which $\mu_1 = \mu^*$), such a policy will have linear regret for any other choice of $(\underline{\mu}, \underline{\sigma}^2)$ and hence cannot be a UFC policy.

The existence of UFC policies in the case considered here is well established, e.g., Auer et al. (2002) (Fig. 4 therein) presented the following UFC policy π_{ACF}:

Policy π_{ACF} (UCB1-NORMAL). At each $n = 1, 2, \ldots$:

i) Sample from any bandit i for which $T^i_{\pi_{\text{ACF}}}(n) < \lceil 8 \ln n \rceil$.

ii) If $T^i_{\pi_{\text{ACF}}}(n) > \lceil 8 \ln n \rceil$, for all $i = 1, \ldots, N$, sample from bandit $\pi_{\text{ACF}}(n+1)$ with

$$\pi_{\text{ACF}}(n+1) = \arg\max_i \left\{ \bar{X}^i_{T^i_\pi(n)} + 4 \cdot S_i(T^i_\pi(n)) \sqrt{\frac{\ln n}{T^i_\pi(n)}} \right\}. \quad (3)$$

(Taking, in this case, $S_i^2(k)$ as the unbiased estimator.)

Additionally, Auer et al. (2002) (in Theorem 4 therein) gave the following bound:

$$R_{\pi_{\text{ACF}}}(n) \leqslant M_{\text{ACF}}(\underline{\mu}, \underline{\sigma}^2) \ln n + C_{\text{ACF}}(\underline{\mu}), \text{ for all } n \text{ and all } (\underline{\mu}, \underline{\sigma}^2), \quad (4)$$

with

$$M_{\text{ACF}}(\underline{\mu}, \underline{\sigma}^2) = 256 \sum_{i:\mu_i \neq \mu^*} \frac{\sigma_i^2}{\Delta_i} + 8 \sum_{i=1}^N \Delta_i, \quad (5)$$

$$C_{\text{ACF}}(\underline{\mu}) = (1 + \frac{\pi^2}{2}) \sum_{i=1}^N \Delta_i. \quad (6)$$

Ineq. (4) readily implies that $R_{\pi_{\text{ACF}}}(n) \leqslant M_{\text{ACF}}(\underline{\mu}, \underline{\sigma}^2) \ln n + o(\ln n)$. Thus, since $\ln n = o(n^\alpha)$ for all $\alpha > 0$ and $R_{\pi_{\text{ACF}}}(n) \geqslant 0$, it follows that π_{ACF} is uniformly fast convergent.

Given that UFC policies exist, the question immediately follows: just how fast can they be? The primary motivation of this paper is the following general result, from Burnetas and Katehakis (1996b), which leveraged the UFC property to establish an asymptotic lower bound on the growth of regret for any such policy π, as well as determining a constant associated with that growth, i.e, that for any UFC policy π, the following holds:

$$\liminf_{n\to\infty} \frac{R_\pi(n)}{\ln n} \geqslant \mathbb{M}_{\text{BK}}(\underline{\mu}, \underline{\sigma}^2), \text{ for all } (\underline{\mu}, \underline{\sigma}^2), \quad (7)$$

where the bound itself $\mathbb{M}_{\text{BK}}(\underline{\mu}, \underline{\sigma}^2)$ is determined by the specific distributions of the populations, in this case

$$\mathbb{M}_{\text{BK}}(\underline{\mu}, \underline{\sigma}^2) = \sum_{i:\mu_i \neq \mu^*} \frac{2\Delta_i}{\ln\left(1 + \frac{\Delta_i^2}{\sigma_i^2}\right)}. \quad (8)$$

For comparison, depending on the specifics of the bandit distributions, there can be considerable distance between the logarithmic term of the upper bound of Eq. (4) and the lower bound implied by Ineq. (7).

The derivation of Ineq. (7) implies that in order to guarantee that a policy is uniformly fast convergent, sub-optimal populations have to be sampled at least a logarithmic number of times. The above bound is a special case of a more general result derived in Burnetas and Katehakis (1996b) (part 1 of Theorem 1 therein) for distributions with multi-parameters $\underline{\theta}$ being unknown:

$$\mathbb{M}_{\text{BK}}(\underline{\theta}) = \sum_{i:\mu_i \neq \mu^*} \frac{\Delta_i}{\mathbb{K}(\underline{\theta}^i, \mu^*)}, \quad (9)$$

where

$$\mathbb{K}(\underline{\theta}, \mu^*) = \inf_{\underline{\theta}'} \{ \mathbb{I}(f_{\underline{\theta}}; f_{\underline{\theta}'}) : \mu(\underline{\theta}') > \mu^* \}, \quad (10)$$

taking $\mathbb{I}(f; g)$ to represent the Kullback-Leibler divergence between densities f and g. For the case of normal distributions with unknown means and variances, the derivation of Eq. (8) from Eq. (9) is given as Proposition 7 in the Appendix.

Previously, Lai and Robbins (1985) had obtained a lower bound for distributions with one-parameter (such as in the current problem of Normal populations with unknown mean but known variance), policies that achieved the lower bound were called ***asymptotically efficient*** or ***asymptotically optimal***.

Ineq. (7) motivates the definition of a uniformly fast convergent policy π as having a ***uniformly maximal convergence rate*** (UM) or simply being *asymptotically optimal*, within the class of uniformly fast convergent policies, if $\lim_{n\to\infty} R_\pi(n)/\ln n = \mathbb{M}_{\text{BK}}(\underline{\mu}, \underline{\sigma}^2)$, since then $V_\pi(n) = n\mu^* - \mathbb{M}_{\text{BK}}(\underline{\mu}, \underline{\sigma}^2) \ln n + o(\ln n)$.

Burnetas and Katehakis (1996b) proposed the following index policy π_{BK} as one that could achieve this lower bound:

Policy π_{BK} (ISM-NORMAL^0)

i) For $n = 1, 2, \ldots, 2N$ sample each bandit twice, and

ii) for $n \geqslant 2N$, sample from bandit $\pi_{\text{BK}}(n+1)$ with

$$\pi_{\text{BK}}(n+1) = \arg\max_i \left\{ \bar{X}^i_{T^i_\pi(n)} + S_i(T^i_\pi(n)) \sqrt{n^{\frac{2}{T^i_\pi(n)}} - 1} \right\}. \quad (11)$$

Burnetas and Katehakis (1996b) were not able to establish the asymptotic optimality of the π_{BK} policy because they were not able to establish a sufficient condition (*Condition* A3 therein), which we express here as the following equivalent conjecture (the referenced open question in the Abstract).

Conjecture 1 *For each i, for every $\varepsilon > 0$, and for $k \to \infty$, the following is true:*

$$\mathbb{P}\left(\bar{X}^i_j + S_i(j)\sqrt{k^{2/j}-1} < \mu_i - \varepsilon \text{ for some } 2 \leqslant j \leqslant k\right) = o(1/k). \tag{12}$$

We show that the above conjecture is *false*, cf. Proposition 9 in the Appendix. In addition, it will follow as a result of Theorem 2 that $R_{\pi_{\text{BK}}} \geqslant O(\sqrt{n})$, i.e., π_{BK} fails to be asymptotically optimal.

One of the central results of this paper is to establish that with a small change (though with large effect), the policy π_{BK} may be modified to one that is provably asymptotically optimal. We introduce in this paper the policy π_{CHK} defined as follows

Policy π_{CHK} (ISM-NORMAL^2)

i) For $n = 1, 2, \ldots, 3N$ sample each bandit three times, and

ii) for $n \geqslant 3N$, sample from bandit $\pi_{\text{CHK}}(n+1)$ with

$$\pi_{\text{CHK}}(n+1) = \arg\max_i \left\{ \bar{X}^i_{T^i_\pi(n)} + S_i(T^i_\pi(n))\sqrt{n^{\frac{2}{T^i_\pi(n)-2}}-1} \right\}. \tag{13}$$

Note that the policy π_{CHK} is only a slight modification of π_{BK}, introducing a -2 in the power of n under the radical. This change is seemingly asymptotically negligible, as in practice, $T^i_{\pi_{\text{BK}}}(n) \to \infty$ (a.s.) with n. It will be shown that not only is π_{CHK} asymptotically optimal (Theorem 5 below), but also:

Theorem 2 *Consider a policy $\pi_{(a,b)}$, with $a > b$, that initially samples each bandit a times, then successively activates bandits according to the maximal index* $\arg\max_i u_i(n, T^i_{\pi_{(a,b)}}(n))$ *where*

$$u_i(n,k) = \bar{X}^i_k + S_i(k)\sqrt{n^{\frac{2}{k-b}}-1}. \tag{14}$$

Then, if the optimal bandit is unique, for $b < 1$,

$$R_{\pi_{(a,b)}} \geqslant O(n^{\frac{1-b}{a-b}}). \tag{15}$$

The proof is given in the Appendix.

Remark 1. 1) We note that the indices of policy π_{CHK} are a significant modification of those of the optimal allocation policy $\pi_{\underline{\sigma}^2}$ for the case of normal bandits with *known* variances,

cf. Burnetas and Katehakis (1996b) and Katehakis and Robbins (1995), which are:

$$\pi_{\underline{\sigma}^2}(n+1) = \arg\max_i \left\{ \bar{X}^i_{T^i_\pi(n)} + \sigma_i\sqrt{\frac{2\ln n}{T^i_\pi(n)}} \right\}$$

the difference being replacing the term $\sigma_i\sqrt{\frac{2\ln n}{T^i_\pi(n)}}$ in $\pi_{\underline{\sigma}^2}$ by $S_i(T^i_\pi(n))\sqrt{n^{\frac{2}{T^i_\pi(n)-2}}-1}$ in π_{CHK}. However, the upper confidence bounds used in policy π_{ACF} are a minor modification of the optimal policy $\pi_{\underline{\sigma}^2}$ the difference being replacing the term $\sigma_i\sqrt{\frac{2\ln n}{T^i_\pi(n)}}$ in $\pi_{\underline{\sigma}^2}$ by $S_i(T^i_\pi(n))\sqrt{\frac{16\ln n}{T^i_\pi(n)}}$ in π_{ACF}.

2) The π_{BK} and $\pi_{\underline{\sigma}^2}$ policies can be seen as connected in the following way, however, observing that $2\ln n/T^i_\pi(n)$ is a first-order approximation of $n^{2/T^i_\pi(n)} - 1 = e^{2\ln n/T^i_\pi(n)} - 1$.

Following Robbins (1952), and additionally Gittins (1979), Lai and Robbins (1985) and Weber (1992) there is a large literature on versions of this problem, cf. Burnetas and Katehakis (2003), Burnetas and Katehakis (1997b) and references therein. For recent work in this area we refer to Audibert et al. (2009), Auer and Ortner (2010), Gittins et al. (2011), Bubeck and Slivkins (2012), Cappé et al. (2013), Kaufmann (2015), Li et al. (2014), Cowan and Katehakis (2015b), Cowan and Katehakis (2015c), and references therein. For more general dynamic programming extensions we refer to Burnetas and Katehakis (1997a), Butenko et al. (2003), Tewari and Bartlett (2008), Audibert et al. (2009), Littman (2012), Abbasi et al. (2013),Feinberg et al. (2014) and references therein. Other related work in this area includes: Burnetas and Katehakis (1993), Burnetas and Katehakis (1996a), Lagoudakis and Parr (2003), Bartlett and Tewari (2009), Tekin and Liu (2012), Jouini et al. (2009), Dayanik et al. (2013), Filippi et al. (2010), Osband and Van Roy (2014), Denardo et al. (2013).

To our knowledge, outside the work in Lai and Robbins (1985), Burnetas and Katehakis (1996b) and Burnetas and Katehakis (1997a), asymptotically optimal policies have only been developed in in Honda and Takemura (2011), and in Honda and Takemura (2010) for the problem of finite known support where optimal policies, cyclic and randomized, that are simpler to implement than those consider in Burnetas and Katehakis (1996b) were constructed. Recently in Cowan and Katehakis (2015a), an asymptotically optimal policy for uniform bandits of unknown support was constructed. The question of whether asymptotically optimal policies exist in the case discussed herein of normal bandits with unknown means and unknown variances was recently resolved in the positive by Honda and Takemura (2013) who demonstrated that a form of Thompson sampling with certain priors on $(\underline{\mu}, \underline{\sigma}^2)$ achieves the asymptotic lower bound $\mathbb{M}_{\text{BK}}(\underline{\mu}, \underline{\sigma}^2)$.

The structure of the rest of the paper is as follows. In Section 2, Theorem 4 establishes a finite horizon bound on the regret of π_{CHK}. From this bound, it follows that π_{CHK} is asymptotically optimal (Theorem 5), and we provide a bound on the remainder term (Theorem 6).

Additionally, in Section 3, the Thompson sampling policy of Honda and Takemura (2013) and π_{CHK} are compared and discussed, as both achieve asymptotic optimality.

2. The Optimality Theorem and Finite Time Bounds

The main results of this paper, that Conjecture 1 is false (cf. Proposition 9 in the Appendix), the asymptotic optimality, and the bounds on the behavior of π_{CHK}, all depend on the following probability bounds; we note that tighter bounds seem possible, but these are sufficient for the proof of the main Theorem.

Proposition 3 *Let Z, U be independent random variables, $Z \sim N(0,1)$ a standard normal, and $U \sim \chi^2_d$ a chi-squared distribution with d degrees of freedom, where $d \geqslant 2$.*

For $\delta > 0$, $p > 0$, the following holds for all $k \geqslant 1$:

$$\frac{1}{2}\mathbb{P}\left(\frac{1}{4}Z^2 \geqslant U \geqslant \delta^2\right) k^{-d/p} \leqslant \mathbb{P}\left(\delta + \sqrt{U}\sqrt{k^{2/p}-1} < Z\right) \leqslant \frac{e^{-(1+\delta^2)/2} p}{2\delta^2\sqrt{d}} \frac{k^{(1-d)/p}}{\ln k}. \quad (16)$$

The proof is given in the Appendix. The bounds provided by this proposition are hardly intuitive, and it is not clear that they are of any particular interest in their own right. However, the order results for the dependence on k, p, and d allow the analysis of this paper to go through.

Theorem 4 *For policy π_{CHK} as defined above, under any choice of bandit parameters, the following bounds hold for all $n \geqslant 3N$ and all $\varepsilon \in (0,1)$:*

$$R_{\pi_{CHK}}(n) \leqslant \sum_{i:\mu_i \neq \mu^*} \left(\frac{2\ln n}{\ln\left(1 + \frac{\Delta_i^2}{\sigma_i^2}\frac{(1-\varepsilon)^2}{(1+\varepsilon)}\right)} + \sqrt{\frac{\pi}{2e}}\frac{8\sigma_*^3}{\Delta_i^3\varepsilon^3}\ln\ln n + \frac{8}{\varepsilon^2} + \frac{8\sigma_i^2}{\Delta_i^2\varepsilon^2} + 3 \right) \Delta_i. \quad (17)$$

Before giving the proof of this bound, we present two results, the first demonstrating the asymptotic optimality of π_{CHK}, the second giving an ε-free version of the above bound, which gives a bound on the sub-logarithmic remainder term. It is worth noting the following: the bounds of Theorem 4 can actually be improved, through the use of a modified version of Proposition 3, to eliminate the $\ln\ln n$ dependence, so the only dependence on n is through the initial $\ln n$ term. The cost of this, however, is a dependence on a larger power of $1/\varepsilon$. The particular form of the bound given in Eq. (17) was chosen to simplify the following two results, cf. Remark 5 in the proof of Proposition 3.

Theorem 5 *For a policy π_{CHK} as defined above, π_{CHK} is asymptotically optimal in the sense that for any choice of bandit parameters,*

$$\lim_{n\to\infty} \frac{R_{\pi_{CHK}}(n)}{\ln n} = \mathbb{M}_{BK}(\underline{\mu}, \underline{\sigma}^2). \quad (18)$$

Proof For any ε such that $0 < \varepsilon < 1$, we have from Theorem 4 that the following holds:

$$\limsup_{n\to\infty} \frac{R_{\pi_{\text{CHK}}}(n)}{\ln n} \leqslant \sum_{i:\mu_i\neq\mu^*} \frac{2\Delta_i}{\ln\left(1 + \frac{\Delta_i^2}{\sigma_i^2}\frac{(1-\varepsilon)^2}{(1+\varepsilon)}\right)}. \quad (19)$$

Taking the infimum over all such ε,

$$\limsup_{n\to\infty} \frac{R_{\pi_{\text{CHK}}}(n)}{\ln n} \leqslant \sum_{i:\mu_i\neq\mu^*} \frac{2\Delta_i}{\ln\left(1 + \frac{\Delta_i^2}{\sigma_i^2}\right)} = \mathbb{M}_{\text{BK}}(\underline{\mu}, \underline{\sigma}^2), \quad (20)$$

and observing the lower bound of Ineq. (7) completes the result. ■

Having established the primary growth order of the regret under policy π_{CHK}, in the following theorem we give a bound on the growth of the remainder term. The utility of this bound depends to some extent on the specific bandit parameters, but we view this particular form of the remainder term as an artifact of the analysis given here, rather than an inherent property of the policy itself. Alternative analyses might yield tighter bounds, we simply establish a convenient bound on the growth order of the remainder.

Theorem 6 *For a policy π_{CHK} as defined above, $R_{\pi_{CHK}}(n) \leqslant \mathbb{M}_{BK}(\underline{\mu}, \underline{\sigma}^2)\ln n + O((\ln n)^{3/4}\ln\ln n)$, and more concretely*

$$\begin{aligned} R_{\pi_{CHK}}(n) \leqslant\ & M^0_{CHK}(\underline{\mu}, \underline{\sigma}^2)\ln n + M^1_{CHK}(\underline{\mu}, \underline{\sigma}^2)(\ln n)^{3/4}\ln\ln n \\ & + M^2_{CHK}(\underline{\mu}, \underline{\sigma}^2)(\ln n)^{3/4} \\ & + M^3_{CHK}(\underline{\mu}, \underline{\sigma}^2)(\ln n)^{1/2} \\ & + M^4_{CHK}(\underline{\mu}, \underline{\sigma}^2), \end{aligned} \quad (21)$$

where

$$\begin{aligned} M^0_{CHK}(\underline{\mu}, \underline{\sigma}^2) &= \mathbb{M}_{BK}(\underline{\mu}, \underline{\sigma}^2) \\ M^1_{CHK}(\underline{\mu}, \underline{\sigma}^2) &= 64\sqrt{\frac{\pi}{2e}} \sum_{i:\mu_i\neq\mu^*} \left(\frac{\sigma_*^3}{\Delta_i^2}\right) \\ M^2_{CHK}(\underline{\mu}, \underline{\sigma}^2) &= 10 \sum_{i:\mu_i\neq\mu^*} \left(\frac{\Delta_i^3}{\left(\sigma_i^2 + \Delta_i^2\right)\left(\ln\left(1 + \frac{\Delta_i^2}{\sigma_i^2}\right)\right)^2} \right) \\ M^3_{CHK}(\underline{\mu}, \underline{\sigma}^2) &= 32 \sum_{i:\mu_i\neq\mu^*} \left(\Delta_i + \frac{\sigma_i^2}{\Delta_i}\right) \\ M^4_{CHK}(\underline{\mu}, \underline{\sigma}^2) &= 3 \sum_{i:\mu_i\neq\mu^*} \Delta_i. \end{aligned} \quad (22)$$

While the above bound admittedly has a more complex form than such a bound as in Eq. (4), it demonstrates the asymptotic optimality of the dominating term, and bounds the sub-logarithmic remainder term.

Proof The bound follows directly from Theorem 4, taking $\varepsilon = \frac{1}{2}(\ln n)^{-1/4}$ for $n \geqslant 3$, and observing the following bound, that for ε such that $0 < \varepsilon < 1/2$,

$$\frac{1}{\ln\left(1+\frac{\Delta_i^2}{\sigma_i^2}\frac{(1-\varepsilon)^2}{1+\varepsilon}\right)} \leqslant \frac{1}{\ln\left(1+\frac{\Delta_i^2}{\sigma_i^2}\right)} + \frac{10\Delta_i^2}{\left(\sigma_i^2+\Delta_i^2\right)\left(\ln\left(1+\frac{\Delta_i^2}{\sigma_i^2}\right)\right)^2}\varepsilon. \tag{23}$$

This inequality is proven separately as Proposition 10 in the Appendix. ■

We make no claim that the results of Theorems 4, 6 are the best achievable for this policy π_{CHK}. At several points in the proofs, choices of convenience were made in the bounding of terms, and different techniques may yield tighter bounds still. But they are sufficient to demonstrate the asymptotic optimality of π_{CHK}, and give useful bounds on the growth of $R_{\pi_{\text{CHK}}}(n)$.

Proof [of Theorem 3] In this proof, we take $\pi = \pi_{\text{CHK}}$ as defined above. For notational convenience, we define the index function

$$u_i(k,j) = \bar{X}_j^i + S_i(j)\sqrt{k^{\frac{2}{j-2}} - 1}. \tag{24}$$

The structure of this proof will be to bound the expected value of $T_\pi^i(n)$ for all sub-optimal bandits i, and use this to bound the regret $R_\pi(n)$. The basic techniques follow those in Katehakis and Robbins (1995) for the known variance case, modified accordingly here for the unknown variance case and assisted by the probability bound of Proposition 3. For any i such that $\mu_i \neq \mu^*$, we define the following quantities: Let $1 > \varepsilon > 0$ and define $\tilde{\varepsilon} = \Delta_i\varepsilon/2$. For $n \geqslant 3N$,

$$\begin{aligned}
n_1^i(n,\varepsilon) &= \sum_{t=3N}^{n} \mathbb{1}\{\pi(t+1) = i, u_i(t, T_\pi^i(t)) \geqslant \mu^* - \tilde{\varepsilon}, \bar{X}^i_{T_\pi^i(t)} \leqslant \mu_i + \tilde{\varepsilon}, S_i^2(T_\pi^i(t)) \leqslant \sigma_i^2(1+\varepsilon)\} \\
n_2^i(n,\varepsilon) &= \sum_{t=3N}^{n} \mathbb{1}\{\pi(t+1) = i, u_i(t, T_\pi^i(t)) \geqslant \mu^* - \tilde{\varepsilon}, \bar{X}^i_{T_\pi^i(t)} \leqslant \mu_i + \tilde{\varepsilon}, S_i^2(T_\pi^i(t)) > \sigma_i^2(1+\varepsilon)\} \\
n_3^i(n,\varepsilon) &= \sum_{t=3N}^{n} \mathbb{1}\{\pi(t+1) = i, u_i(t, T_\pi^i(t)) \geqslant \mu^* - \tilde{\varepsilon}, \bar{X}^i_{T_\pi^i(t)} > \mu_i + \tilde{\varepsilon}\} \\
n_4^i(n,\varepsilon) &= \sum_{t=3N}^{n} \mathbb{1}\{\pi(t+1) = i, u_i(t, T_\pi^i(t)) < \mu^* - \tilde{\varepsilon}\}.
\end{aligned} \tag{25}$$

Hence, we have the following relationship for $n \geqslant 3N$, that

$$T_\pi^i(n+1) = 3 + \sum_{t=3N}^{n} \mathbb{1}\{\pi(t+1) = i\} = 3 + n_1^i(n,\varepsilon) + n_2^i(n,\varepsilon) + n_3^i(n,\varepsilon) + n_4^i(n,\varepsilon). \tag{26}$$

The proof proceeds by bounding, in expectation, each of the four terms.

Observe that, by the structure of the index function u_i,

$$\begin{aligned}
n_1^i(n,\varepsilon) &\leqslant \sum_{t=3N}^{n} \mathbb{1}\left\{\pi(t+1) = i, (\mu_i + \tilde{\varepsilon}) + \sigma_i\sqrt{1+\varepsilon}\sqrt{t^{\frac{2}{T_\pi^i(t)-2}} - 1} \geqslant \mu^* - \tilde{\varepsilon}\right\} \\
&= \sum_{t=3N}^{n} \mathbb{1}\left\{\pi(t+1) = i, T_\pi^i(t) \leqslant \frac{2\ln t}{\ln\left(1+\frac{1}{\sigma_i^2}\frac{(\Delta_i - 2\tilde{\varepsilon})^2}{(1+\varepsilon)}\right)} + 2\right\} \\
&= \sum_{t=3N}^{n} \mathbb{1}\left\{\pi(t+1) = i, T_\pi^i(t) \leqslant \frac{2\ln t}{\ln\left(1+\frac{\Delta_i^2}{\sigma_i^2}\frac{(1-\varepsilon)^2}{(1+\varepsilon)}\right)} + 2\right\} \\
&\leqslant \sum_{t=3N}^{n} \mathbb{1}\left\{\pi(t+1) = i, T_\pi^i(t) \leqslant \frac{2\ln n}{\ln\left(1+\frac{\Delta_i^2}{\sigma_i^2}\frac{(1-\varepsilon)^2}{(1+\varepsilon)}\right)} + 2\right\} \\
&\leqslant \frac{2\ln n}{\ln\left(1+\frac{\Delta_i^2}{\sigma_i^2}\frac{(1-\varepsilon)^2}{(1+\varepsilon)}\right)} + 2 + 1 - 3 \\
&= \frac{2\ln n}{\ln\left(1+\frac{\Delta_i^2}{\sigma_i^2}\frac{(1-\varepsilon)^2}{(1+\varepsilon)}\right)}.
\end{aligned} \tag{27}$$

The last inequality follows, observing that $T_\pi^i(n)$ may be expressed as the sum of $\pi(t) = i$ indicators, and seeing that the additional condition bounds the number of non-zero terms in this sum. The additional $+1$ term accounts for the potential $\pi(n+1) = i$ term beyond the condition on $T_\pi^i(t)$, and the -3 accounts for the initial three activations of bandit i, which are not counted within the bounds of the sum. Note, this bound holds surely, over all outcomes.

For the second term,

$$\begin{aligned}
n_2^i(n,\varepsilon) &\leqslant \sum_{t=3N}^{n} \mathbb{1}\{\pi(t+1) = i, S_i^2(T_\pi^i(t)) > \sigma_i^2(1+\varepsilon)\} \\
&= \sum_{t=3N}^{n} \sum_{k=3}^{t} \mathbb{1}\{\pi(t+1) = i, S_i^2(k) > \sigma_i^2(1+\varepsilon), T_\pi^i(t) = k\} \\
&= \sum_{t=3N}^{n} \sum_{k=3}^{t} \mathbb{1}\{\pi(t+1) = i, T_\pi^i(t) = k\}\mathbb{1}\{S_i^2(k) > \sigma_i^2(1+\varepsilon)\} \\
&\leqslant \sum_{k=3}^{n} \mathbb{1}\{S_i^2(k) > \sigma_i^2(1+\varepsilon)\} \sum_{t=k}^{n} \mathbb{1}\{\pi(t+1) = i, T_\pi^i(t) = k\} \\
&\leqslant \sum_{k=3}^{n} \mathbb{1}\{S_i^2(k) > \sigma_i^2(1+\varepsilon)\}.
\end{aligned} \tag{28}$$

The last inequality follows as, for fixed k, $\{\pi(t+1)=i, T_\pi^i(t)=k\}$ may be true for at most one value of t. Recall that $kS_i^2(k)/\sigma_i^2$ has the distribution of a χ^2_{k-1} random variable. Letting $U_k \sim \chi^2_k$, from the above we have

$$\begin{aligned}
\mathbb{E}\left[n_2^i(n,\varepsilon)\right] &\leqslant \sum_{k=3}^{n} \mathbb{P}\left(S_i^2(k) > \sigma_i^2(1+\varepsilon)\right) \\
&= \sum_{k=3}^{\infty} \mathbb{P}\left(U_{k-1}/k > (1+\varepsilon)\right) \\
&\leqslant \sum_{k=3}^{\infty} \mathbb{P}\left(U_{k-1}/(k-1) > (1+\varepsilon)\right) \\
&\leqslant \sum_{k=1}^{\infty} \mathbb{P}\left(U_k > k(1+\varepsilon)\right) \\
&\leqslant \frac{1}{\sqrt{\frac{e^\varepsilon}{1+\varepsilon}-1}} \leqslant \frac{8}{\varepsilon^2} < \infty.
\end{aligned} \tag{29}$$

The penultimate step is a Chernoff bound on the terms, $\mathbb{P}\left(U_k > k(1+\varepsilon)\right) \leqslant (e^{-\varepsilon}(1+\varepsilon))^{k/2}$. As this bound is not common, it is verified in the Appendix as Proposition 8.

To bound the third term, a similar rearrangement to Eq. (28) (using the sample mean instead of the sample variance) yields:

$$n_3^i(n,\varepsilon) \leqslant \sum_{t=3N}^{n} \mathbb{1}\{\pi(t+1)=i, \bar{X}^i_{T_\pi^i(t)} > \mu_i + \tilde{\varepsilon}\} \leqslant \sum_{k=3}^{n} \mathbb{1}\{\bar{X}_k^i > \mu_i + \tilde{\varepsilon}\}. \tag{30}$$

Recalling that $\bar{X}_k^i - \mu_i \sim Z\sigma_i/\sqrt{k}$ for Z a standard normal,

$$\mathbb{E}\left[n_3^i(n,\varepsilon)\right] \leqslant \sum_{k=3}^{n} \mathbb{P}\left(\bar{X}_k^i > \mu_i + \tilde{\varepsilon}\right) \leqslant \sum_{k=1}^{\infty} \mathbb{P}\left(Z\sigma_i/\sqrt{k} > \tilde{\varepsilon}\right) \leqslant \frac{1}{e^{\frac{\tilde{\varepsilon}^2}{2\sigma_i^2}}-1} \leqslant \frac{2\sigma_i^2}{\tilde{\varepsilon}^2} < \infty. \tag{31}$$

The penultimate step is the standard Chernoff bound on the terms, $\mathbb{P}\left(Z > \delta\sqrt{k}\right) \leqslant e^{-k\delta^2/2}$.

To bound the n_4^i term, observe that in the event $\pi(t+1)=i$, from the structure of the policy it must be true that $u_i(t, T_\pi^i(t)) = \max_j u_j(t, T_\pi^j(t))$. Thus, if i^* is some bandit such that $\mu_{i^*} = \mu^*$, $u_{i^*}(t, T_\pi^{i^*}(t)) \leqslant u_i(t, T_\pi^i(t))$. In particular, we take i^* to be a bandit that not only achieves the maximal mean μ^*, but also the minimal variance among optimal bandits, $\sigma_{i^*}^2 = \sigma_*^2$. We have the following bound,

$$\begin{aligned}
n_4^i(n,\varepsilon) &\leqslant \sum_{t=3N}^{n} \mathbb{1}\{\pi(t+1)=i, u_{i^*}(t, T_\pi^{i^*}(t)) < \mu^* - \tilde{\varepsilon}\} \\
&\leqslant \sum_{t=3N}^{n} \mathbb{1}\{u_{i^*}(t, T_\pi^{i^*}(t)) < \mu^* - \tilde{\varepsilon}\} \\
&\leqslant \sum_{t=3N}^{n} \mathbb{1}\{u_{i^*}(t,s) < \mu^* - \tilde{\varepsilon} \text{ for some } 3 \leqslant s \leqslant t\}.
\end{aligned} \tag{32}$$

The last step follows as for t in this range, $3 \leqslant T_\pi^{i^*}(t) \leqslant t$. Hence

$$\mathbb{E}\left[n_4^i(n,\varepsilon)\right] \leqslant \sum_{t=3N}^{n} \mathbb{P}\left(u_{i^*}(t,s) < \mu^* - \tilde{\varepsilon} \text{ for some } 3 \leqslant s \leqslant t\right). \tag{33}$$

As an aside, this is essentially the point at which the conjectured Eq. (12) would have come into play for the proof of the optimality of π_{BK}, bounding the growth of the corresponding term for that policy. We will essentially prove a successful version of that conjecture here. Define the events $A^*_{s,t,\tilde{\varepsilon}} = \{u_{i^*}(t,s) < \mu^* - \tilde{\varepsilon}\}$. Observing that $\sqrt{s}(\bar{X}_s^{i^*} - \mu^*)/\sigma_* \sim Z$ and $S^2_{i^*}(s) \sim \sigma_*^2 U_{s-1}/s$ where Z has a standard normal distribution and $U_{s-1} \sim \chi^2_{s-1}$, Z and U_{s-1} independent,

$$\begin{aligned}
\mathbb{P}\left(A^*_{s,t,\tilde{\varepsilon}}\right) &= \mathbb{P}\left(\bar{X}_s^{i^*} + S_{i^*}(s)\sqrt{t^{\frac{2}{s-2}}-1} < \mu^* - \tilde{\varepsilon}\right) \\
&= \mathbb{P}\left(\mu^* + Z\frac{\sigma_*}{\sqrt{s}} + \sigma_*\frac{\sqrt{U_{s-1}}}{\sqrt{s}}\sqrt{t^{\frac{2}{s-2}}-1} < \mu^* - \tilde{\varepsilon}\right) \\
&= \mathbb{P}\left(Z + \sqrt{U_{s-1}}\sqrt{t^{\frac{2}{s-2}}-1} < -\frac{\tilde{\varepsilon}}{\sigma_*}\sqrt{s}\right).
\end{aligned} \tag{34}$$

Observing the symmetry of the standard normal distribution, the above may be rewritten as

$$\begin{aligned}
\mathbb{P}\left(A^*_{s,t,\tilde{\varepsilon}}\right) &= \mathbb{P}\left(\frac{\tilde{\varepsilon}}{\sigma_*}\sqrt{s} + \sqrt{U_{s-1}}\sqrt{t^{\frac{2}{s-2}}-1} < Z\right) \\
&\leqslant \frac{e^{-(\tilde{\varepsilon}/\sigma_*)^2 s/2}(s-2)}{2(\tilde{\varepsilon}/\sigma_*)^2 s\sqrt{e(s-1)}}\left(\frac{t^{-1}}{\ln t}\right) \\
&\leqslant \frac{e^{-(\tilde{\varepsilon}/\sigma_*)^2 s/2}}{2(\tilde{\varepsilon}/\sigma_*)^2}\frac{1}{\sqrt{es}}\left(\frac{t^{-1}}{\ln t}\right) \\
&= \left(\frac{1}{2(\tilde{\varepsilon}/\sigma_*)^2\sqrt{e}}\right)\frac{e^{-(\tilde{\varepsilon}/\sigma_*)^2 s/2}}{\sqrt{s}}\left(\frac{t^{-1}}{\ln t}\right),
\end{aligned} \tag{35}$$

where the first inequality follows as an application of Proposition 3, and the second since $s \geqslant 3$.

Applying a union bound to Eq. (33),

$$\begin{aligned}
\mathbb{E}\left[n_4^i(n,\varepsilon)\right] &\leqslant \sum_{t=3N}^{n}\sum_{s=3}^{t}\mathbb{P}\left(A^*_{s,t,\tilde{\varepsilon}}\right) \\
&\leqslant \sum_{t=3N}^{n}\sum_{s=3}^{t}\left(\frac{1}{2(\tilde{\varepsilon}/\sigma_*)^2\sqrt{e}}\right)\frac{e^{-(\tilde{\varepsilon}/\sigma_*)^2 s/2}}{\sqrt{s}}\left(\frac{t^{-1}}{\ln t}\right) \\
&\leqslant \left(\frac{1}{2(\tilde{\varepsilon}/\sigma_*)^2\sqrt{e}}\right)\int_{s=0}^{\infty}\frac{e^{-(\tilde{\varepsilon}/\sigma_*)^2 s/2}}{\sqrt{s}}ds\int_{t=e}^{n}\left(\frac{t^{-1}}{\ln t}\right)dt \qquad (36)\\
&= \left(\frac{1}{2(\tilde{\varepsilon}/\sigma_*)^2\sqrt{e}}\right)\frac{\sqrt{2\pi}}{(\tilde{\varepsilon}/\sigma_*)}\ln\ln n \\
&= \sqrt{\frac{\pi}{2e}}\frac{\sigma_*^3}{\tilde{\varepsilon}^3}\ln\ln n.
\end{aligned}$$

The bounds follow, removing the dependence of the s-sum on t by extending it to ∞, and bounding the sums by integrals of the (decreasing) summands by slightly extending the range of each.

From the above results, and observing that $T_\pi^i(n) \leqslant T_\pi^i(n+1)$, it follows from Eq. (26) that for any ε such that $0 < \varepsilon < 1$,

$$\begin{aligned}
\mathbb{E}\left[T_\pi^i(n)\right] &\leqslant 3 + \frac{2\ln n}{\ln\left(1+\frac{\Delta_i^2}{\sigma_i^2}\frac{(1-\varepsilon)^2}{(1+\varepsilon)}\right)} + \frac{8}{\varepsilon^2} + \frac{2\sigma_i^2}{\tilde{\varepsilon}^2} + \sqrt{\frac{\pi}{2e}}\frac{\sigma_*^3}{\tilde{\varepsilon}^3}\ln\ln n \\
&= 3 + \frac{2\ln n}{\ln\left(1+\frac{\Delta_i^2}{\sigma_i^2}\frac{(1-\varepsilon)^2}{(1+\varepsilon)}\right)} + \frac{8}{\varepsilon^2} + \frac{8\sigma_i^2}{\Delta_i^2\varepsilon^2} + \sqrt{\frac{\pi}{2e}}\frac{8\sigma_*^3}{\Delta_i^3\varepsilon^3}\ln\ln n. \qquad (37)
\end{aligned}$$

The result then follows from the definition of regret in Eq. (2). ■

Remark 2. It is interesting to note in the above proof the effect of the -2 in the exponent on t in Eq. (35) and Eq. (36), as this is effectively what differentiates the asymptotically optimal π_{CHK} from the sub-optimal π_{BK}. With the -2, the application of Proposition 3 yields a $t^{-1}/\ln t$ bound, while without the -2, the resulting t-term is $t^{-1+2/s}/\ln t$.

Remark 3. Numerical Regret Comparison: Figure 1 shows the results of a small simulation study done on a set of six populations with means and variances given in Table 1. It provides plots of the regrets when implementing policies π_{CHK} (the index policy of Eq. (13)), π_{ACF} (the index policy of Eq. (3)) , and π_G a 'greedy' policy that always activates the bandit with the current highest average. Each policy was implemented over a horizon of 100,000 activations, each replicated 10,000 times to produce a good estimate of the average regret $R_\pi(n)$ over the times indicated. The left plot is on the time scale of the first $10,000$ activations, and the right is on the full time scale of $100,000$ activations.

μ_i	8	8	7.9	7	-1	0
σ_i^2	1	1.4	0.5	3	1	4

Table 1

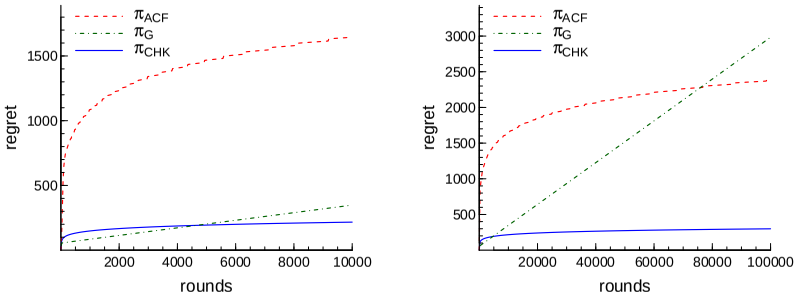


Figure 1: Numerical Regret Comparison of π_{ACF}, π_{CHK}, and π_{G}; Left: $[0,10,000]$ range, Right: $[0,100,000]$ range.

Remark 4. *Bounds and Limits:* Figure 2 shows first (left) a comparison of the theoretical bounds on the regret, $B_{\pi_{\text{ACF}}}(n)$ and $B_{\pi_{\text{CHK}}}(n)$ representing the theoretical regret bounds of the RHS of Eq. (4) and Eq. (17) respectively, taking $\varepsilon = (\ln n)^{1/4}$ in the latter case, for the means and variances indicated in Table 1. Additionally, Figure 2 (right) shows the convergence of $R_{\pi_{\text{CHK}}}(n)/\ln n$ to the theoretical lower bound $\mathbb{M}_{\text{BK}}(\underline{\mu},\underline{\sigma^2})$. It is worth noting that the convergence to the asymptotic limit from below is an artifact of the specific bandit parameters chosen in this case. Alternative parameters can be found that result in convergence from above, for instance parameter choices that force the initial activation period to accumulate regret above this limit.

3. A Comparison of π_{CHK} and Thompson Sampling

Honda and Takemura (2013) considered the following policy:

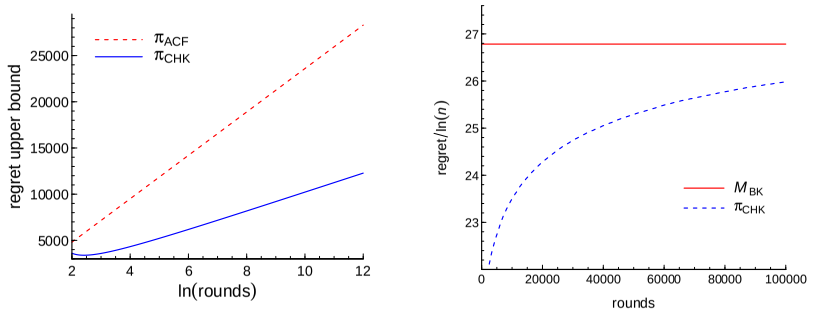


Figure 2: Left: Plots of $B_{\pi_{\text{ACF}}}(n)$ and $B_{\pi_{\text{CHK}}}(n)$. Right: Convergence of $R_{\pi_{\text{CHK}}}(n)/\ln(n)$ to $\mathbb{M}_{\text{BK}}(\underline{\mu},\underline{\sigma}^2)$.

Policy π_{TS} (TS-NORMAL$^{\alpha}$)

i) Initially, sample each bandit $\tilde{n} \geqslant \max(2, 3-\lfloor 2\alpha \rfloor)$ times.

ii) For $n \geqslant \tilde{n}$: For each i generate a random sample

$$U_n^i \sim \bar{X}^i_{T^i_\pi(n)} + S_i(T^i_\pi(n)) \frac{T_{i,n}(T^i_\pi(n)+2\alpha-1)}{\sqrt{T^i_\pi(n)+2\alpha-1}},$$

with $T_{i,n}(d)$ a t-distribution with degree d, i.e., the posterior distribution for μ_i, given $\left(\bar{X}^i_{T^i_\pi(n)}, S_i^2(T^i_\pi(n))\right)$, and a prior for $(\mu_i,\sigma_i^2) \propto (\sigma_i^2)^{-1-\alpha}$.

iii) Then, take

$$\pi_{\text{TS}}(n+1) = \arg\max_i U_n^i. \tag{38}$$

It was proven in Honda and Takemura (2013) that for $\alpha < 0$, the above Thompson sampling algorithm is asymptotically optimal, i.e., $\lim_{n\to\infty} R_{\pi_{\text{TS}}}(n)/\ln n = \mathbb{M}_{\text{BK}}(\underline{\mu},\underline{\sigma}^2)$, and further that $R_{\pi_{\text{TS}}}(n) = \mathbb{M}_{\text{BK}}(\underline{\mu},\underline{\sigma}^2)\ln n + O((\ln n)^{4/5})$.

Policies π_{TS} and π_{CHK} differ decidedly in structure. One key difference, π_{TS} is an inherently randomized policy, while decisions under π_{CHK} are completely determined given the bandit results at a given time. Given that both π_{TS} and π_{CHK} are asymptotically optimal, it is interesting to compare the performances of these two algorithms over finite time horizons,

and observe any practical differences between them. To that end, two small simulation studies were done for different sets of bandit parameters $(\underline{\mu},\underline{\sigma}^2)$. In each case, the uniform prior $\alpha = -1$ was used. The simulations were carried out on a 10,000 round time horizon, and replicated sufficiently many times to get good estimates for the expected regret over the times indicated.

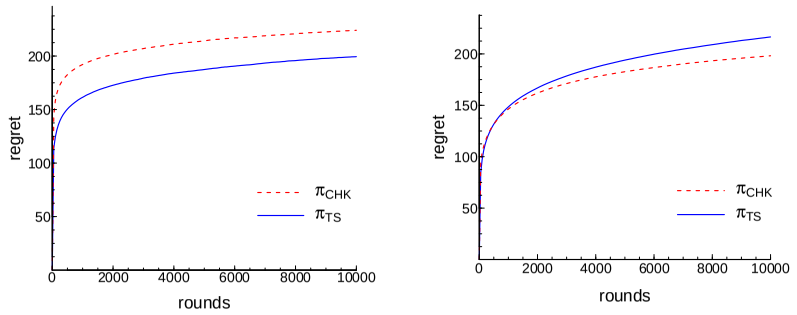


Figure 3: Numerical Regret Comparison of $\pi_{\text{CHK}}$and π_{TS} for the parameters, of Table 1, left and Table 2, right.

μ_i	10	9	8	7	-1	0
σ_i^2	8	1	1	0.5	1	4

Table 2

We observe from the above, and from general sampling of bandit parameters, that π_{TS} and π_{CHK} generally produce comparable expected regret. A general exploration of random parameters suggests that, on average, π_{TS} is slightly superior to π_{CHK} in cases where all bandits have roughly equal variances, while π_{CHK} has an edge when the optimal bandits have large variance relative to the other bandits, and the size of the bandit discrepancies. We additionally plot the variance in sample regret associated with the previous simulations (Fig. 4). Additional numerical experiments, not pictured here, indicate that the superior policy in each case may exhibit a slightly heavier tail distribution towards larger regret. In general, the question of which policy is superior seems largely context specific.

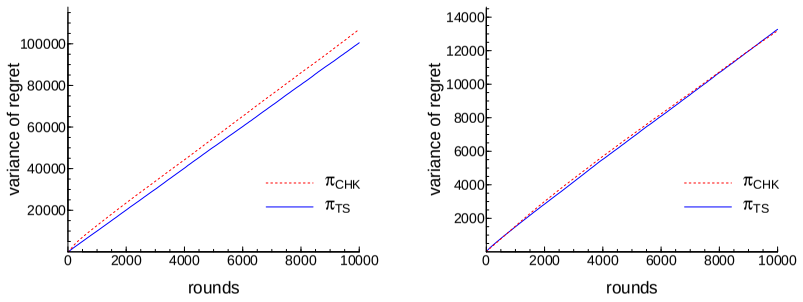


Figure 4: Numerical comparison of variance of sample regret for $\pi_{\text{CHK}}$and π_{TS} for different parameters, of Table 1, left and Table 2, right.

References

Yasin Abbasi, Peter L. Bartlett, Varun Kanade, Yevgeny Seldin, and Csaba Szepesvári. Online learning in Markov decision processes with adversarially chosen transition probability distributions. In *Advances in Neural Information Processing Systems*, pages 2508–2516, 2013.

Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. Exploration - exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410 (19):1876 – 1902, 2009.

Peter Auer and Ronald Ortner. Ucb revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1-2):55 – 65, 2010.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235 – 256, 2002.

Peter L Bartlett and Ambuj Tewari. Regal: A regularization based algorithm for reinforcement learning in weakly communicating mdps. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 35 – 42. AUAI Press, 2009.

Sébastien Bubeck and Aleksandrs Slivkins. The best of both worlds: Stochastic and adversarial bandits. arXiv preprint arXiv:1202.4473, 2012.

Apostolos N. Burnetas and Michael N. Katehakis. On sequencing two types of tasks on a single processor under incomplete information. *Probability in the Engineering and Informational Sciences*, 7(1):85 – 119, 1993.

Apostolos N. Burnetas and Michael N. Katehakis. On large deviations properties of sequential allocation problems. *Stochastic Analysis and Applications*, 14(1):23 – 31, 1996a.

Apostolos N. Burnetas and Michael N. Katehakis. Optimal adaptive policies for sequential allocation problems. *Advances in Applied Mathematics*, 17(2):122 – 142, 1996b.

Apostolos N. Burnetas and Michael N. Katehakis. Optimal adaptive policies for Markov decision processes. *Mathematics of Operations Research*, 22(1):222 – 255, 1997a.

Apostolos N. Burnetas and Michael N. Katehakis. On the finite horizon one-armed bandit problem. *Stochastic Analysis and Applications*, 16(1):845 – 859, 1997b.

Apostolos N. Burnetas and Michael N. Katehakis. Asymptotic Bayes analysis for the finite-horizon one-armed-bandit problem. *Probability in the Engineering and Informational Sciences*, 17(01):53 – 82, 2003.

Sergiy Butenko, Panos M Pardalos, and Robert Murphey. *Cooperative Control: Models, Applications, and Algorithms*. Kluwer Academic Publishers, 2003.

Olivier Cappé, Aurélien Garivier, Odalric-Ambrym Maillard, Rémi Munos, and Gilles Stoltz. Kullback - Leibler upper confidence bounds for optimal sequential allocation. *The Annals of Statistics*, 41(3):1516 – 1541, 2013.

Wesley Cowan and Michael N. Katehakis. An asymptotically optimal UCB policy for uniform bandits of unknown support. *arXiv preprint arXiv:1505.01918*, 2015a.

Wesley Cowan and Michael N. Katehakis. Asymptotic behavior of minimal-exploration allocation policies: Almost sure, arbitrarily slow growing regret. *arXiv preprint arXiv:1505.02865*, 2015b.

Wesley Cowan and Michael N. Katehakis. Multi-armed bandits under general depreciation and commitment. *Probability in the Engineering and Informational Sciences*, 29(01):51 – 76, 2015c.

Savas Dayanik, Warren B Powell, and Kazutoshi Yamazaki. Asymptotically optimal Bayesian sequential change detection and identification rules. *Annals of Operations Research*, 208(1):337 – 370, 2013.

Eric V Denardo, Eugene A Feinberg, and Uriel G Rothblum. The multi-armed bandit, with constraints. In M.N. Katehakis, S.M. Ross, and J. Yang, editors, *Cyrus Derman Memorial Volume I: Optimization under Uncertainty: Costs, Risks and Revenues*. Annals of Operations Research, Springer, New York, 2013.

Eugene A Feinberg, Pavlo O Kasyanov, and Michael Z Zgurovsky. Convergence of value iterations for total-cost mdps and pomdps with general state and action sets. In *Adaptive Dynamic Programming and Reinforcement Learning (ADPRL), 2014 IEEE Symposium on*, pages 1 – 8. IEEE, 2014.

Sarah Filippi, Olivier Cappé, and Aurélien Garivier. Optimism in reinforcement learning based on Kullback Leibler divergence. In *48th Annual Allerton Conference on Communication, Control, and Computing*, 2010.

John C. Gittins. Bandit processes and dynamic allocation indices (with discussion). *J. Roy. Stat. Soc. Ser. B*, 41:335–340, 1979.

John C. Gittins, Kevin Glazebrook, and Richard R. Weber. *Multi-armed Bandit Allocation Indices*. John Wiley & Sons, West Sussex, U.K., 2011.

Junya Honda and Akimichi Takemura. An asymptotically optimal bandit algorithm for bounded support models. In *COLT*, pages 67 – 79. Citeseer, 2010.

Junya Honda and Akimichi Takemura. An asymptotically optimal policy for finite support models in the multiarmed bandit problem. *Machine Learning*, 85(3):361 – 391, 2011.

Junya Honda and Akimichi Takemura. Optimality of Thompson sampling for Gaussian bandits depends on priors. *arXiv preprint arXiv:1311.1894*, 2013.

Wassim Jouini, Damien Ernst, Christophe Moy, and Jacques Palicot. Multi-armed bandit based policies for cognitive radio's decision making issues. In *3rd international conference on Signals, Circuits and Systems (SCS)*, 2009.

Michael N. Katehakis and Herbert Robbins. Sequential choice from several populations. *Proceedings of the National Academy of Sciences of the United States of America*, 92 (19):8584, 1995.

Emilie Kaufmann. Analyse de stratégies Bayésiennes et fréquentistes pour l'allocation séquentielle de ressources. *Doctorat*, ParisTech., Jul. 31 2015.

Michail G Lagoudakis and Ronald Parr. Least-squares policy iteration. *The Journal of Machine Learning Research*, 4:1107 – 1149, 2003.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4 – 2, 1985.

Lihong Li, Remi Munos, and Csaba Szepesvári. On minimax optimal offline policy evaluation. arXiv preprint arXiv:1409.3653, 2014.

Michael L Littman. Inducing partially observable Markov decision processes. In *ICGI*, pages 145 – 148, 2012.

Ian Osband and Benjamin Van Roy. Near-optimal reinforcement learning in factored mdps. In *Advances in Neural Information Processing Systems*, pages 604 – 612, 2014.

Herbert Robbins. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Monthly*, 58:527–536, 1952.

Cem Tekin and Mingyan Liu. Approximately optimal adaptive learning in opportunistic spectrum access. In *INFOCOM, 2012 Proceedings IEEE*, pages 1548 – 1556. IEEE, 2012.

Ambuj Tewari and Peter L Bartlett. Optimistic linear programming gives logarithmic regret for irreducible mdps. In *Advances in Neural Information Processing Systems*, pages 1505 – 1512, 2008.

Richard R Weber. On the Gittins index for multiarmed bandits. *The Annals of Applied Probability*, 2(4):1024 – 1033, 1992.

Acknowledgement: We are grateful for the comments of three reviewers, especially one remark by David Pal which allowed us to drop an unnecessary additional constant term in Eq. (27).

We acknowledge support for this work from the National Science Foundation (NSF grant CMMI-14-50743) and the Japan Society for the Promotion of Science (JSPS Grant-in-Aid for Scientific Research No. 26106506).

Appendix A. Additional Proofs

Proof [of Proposition 3] Let $P = \mathbb{P}\left(\delta + \sqrt{U}\sqrt{k^{2/p}-1} < Z\right)$. Note immediately, $P \geqslant \mathbb{P}\left(\delta + \sqrt{U}k^{1/p} < Z\right)$. Further,

$$\begin{aligned} P &\geqslant \mathbb{P}\left(\delta + \sqrt{U}k^{1/p} < Z \text{ and } \sqrt{U}k^{1/p} \geqslant \delta\right) \\ &\geqslant \mathbb{P}\left(2\sqrt{U}k^{1/p} < Z \text{ and } \sqrt{U}k^{1/p} \geqslant \delta\right) \\ &= \int_{\frac{\delta^2}{k^{2/p}}}^{\infty} \int_{2\sqrt{u}k^{1/p}}^{\infty} \frac{e^{-z^2/2}}{\sqrt{2\pi}} f_d(u) dz du. \end{aligned} \tag{39}$$

Where $f_d(u)$ is taken to be the density of a χ^2_d-random variable. Letting $\tilde{u} = k^{2/p}u$,

$$\begin{aligned} P &\geqslant \frac{1}{k^{2/p}} \int_{\delta^2}^{\infty} \int_{2\sqrt{\tilde{u}}}^{\infty} \frac{e^{-z^2/2}}{\sqrt{2\pi}} f_d\left(\frac{\tilde{u}}{k^{2/p}}\right) dz d\tilde{u} \\ &= \frac{1}{k^{2/p}} \int_{\delta^2}^{\infty} \int_{2\sqrt{\tilde{u}}}^{\infty} \frac{e^{-z^2/2}}{\sqrt{2\pi}} \frac{1}{2^{d/2}\Gamma(d/2)} \left(\frac{\tilde{u}}{k^{2/p}}\right)^{d/2-1} e^{-\frac{\tilde{u}}{2k^{2/p}}} dz d\tilde{u} \\ &= \left(\frac{1}{k^{2/p}}\right)^{d/2} \int_{\delta^2}^{\infty} \int_{2\sqrt{\tilde{u}}}^{\infty} \frac{e^{-z^2/2}}{\sqrt{2\pi}} \frac{1}{2^{d/2}\Gamma(d/2)} \tilde{u}^{d/2-1} e^{-\frac{\tilde{u}}{2k^{2/p}}} dz d\tilde{u}. \end{aligned} \tag{40}$$

Observing that $k^{2/p} \geqslant 1$,

$$\begin{aligned} P &\geqslant \left(\frac{1}{k^{2/p}}\right)^{d/2} \int_{\delta^2}^{\infty} \int_{2\sqrt{\tilde{u}}}^{\infty} \frac{e^{-z^2/2}}{\sqrt{2\pi}} \frac{1}{2^{d/2}\Gamma(d/2)} \tilde{u}^{d/2-1} e^{-\frac{\tilde{u}}{2}} dz d\tilde{u} \\ &= k^{-d/p} \mathbb{P}\left(2\sqrt{U} \leqslant Z \text{ and } U \geqslant \delta^2\right) \\ &= \frac{1}{2} k^{-d/p} \mathbb{P}\left(4U \leqslant Z^2 \text{ and } U \geqslant \delta^2\right) = \frac{1}{2} k^{-d/p} \mathbb{P}\left(\frac{1}{4}Z^2 \geqslant U \geqslant \delta^2\right). \end{aligned} \tag{41}$$

The exchange from integral to probability is simply the interpretation of the integrand as the joint pdf of U and Z.

For the upper bound, we utilize the classic normal tail bound, $\mathbb{P}(x < Z) \leqslant e^{-x^2/2}/(x\sqrt{2\pi})$.

$$P \leqslant \mathbb{E}\left[\frac{e^{-\left(\delta+\sqrt{U}\sqrt{k^{2/p}-1}\right)^2/2}}{(\delta+\sqrt{U}\sqrt{k^{2/p}-1})\sqrt{2\pi}}\right] \leqslant \frac{e^{-\delta^2/2}}{\delta\sqrt{2\pi}} \mathbb{E}\left[e^{-\delta\sqrt{U}\sqrt{k^{2/p}-1}-\frac{1}{2}U\left(k^{2/p}-1\right)}\right]. \tag{42}$$

Observing the bound that for positive x, $e^{-x} \leqslant 1/x$, and recalling that $d \geqslant 2$,

$$\begin{aligned} P &\leqslant \frac{e^{-\delta^2/2}}{\delta\sqrt{2\pi}} \mathbb{E}\left[\frac{e^{-\frac{1}{2}U\left(k^{2/p}-1\right)}}{\delta\sqrt{U}\sqrt{k^{2/p}-1}}\right] \\ &= \frac{e^{-\delta^2/2}}{\delta^2\sqrt{2\pi}\sqrt{k^{2/p}-1}} \mathbb{E}\left[U^{-\frac{1}{2}} e^{-\frac{1}{2}U\left(k^{2/p}-1\right)}\right] \\ &= \frac{e^{-\delta^2/2}}{\delta^2\sqrt{2\pi}\sqrt{k^{2/p}-1}} \left(\frac{k^{(1-d)/p}\Gamma\left(\frac{d}{2}-\frac{1}{2}\right)}{\sqrt{2}\Gamma\left(\frac{d}{2}\right)}\right). \end{aligned} \tag{43}$$

Here we utilize the following bounds: $e^x - 1 \geqslant (e/2)x^2$, which is easy to prove, and $\Gamma(d/2-1/2)/\Gamma(d/2) \leqslant \sqrt{2\pi/d}$, which may be proved on integer $d \geqslant 2$ by induction. This yields:

$$P \leqslant \frac{e^{-(1+\delta^2)/2} p}{2\delta^2 \ln k} \frac{k^{(1-d)/p}}{\sqrt{d}}. \tag{44}$$

This completes the proof.

Remark 5. Room for Improvement: The choice of the $e^x - 1 \geqslant (e/2)x^2$ bound above was in fact arbitrary - other bounds, such as involving alternative powers of x, could be used. This would influence how the resulting bound on P is utilized, for instance in the proof of Theorem 4. The use of $e^{-x} \leqslant 1/x$ in Eq. (43) should be considered similarly. ■

Proposition 7 *In the case of normal distributions with unknown means and variances,*

$$\mathbb{M}_{BK}(\underline{\mu}, \underline{\sigma}^2) = \sum_{i:\mu_i \neq \mu^*} \frac{2\Delta_i}{\ln\left(1+\frac{\Delta_i^2}{\sigma_i^2}\right)}. \tag{45}$$

Proof From Eq. (9) and Eq. (10), it suffices to show that in the case of normal distributions with unknown means and variances, for any sub-optimal bandit i,

$$\mathbb{K}((\mu_i, \sigma_i^2), \mu^*) = \inf_{(\tilde{\mu}, \tilde{\sigma}^2)} \{\mathbb{I}(f_{(\mu_i,\sigma_i^2)}; f_{(\tilde{\mu},\tilde{\sigma}^2)}) : \tilde{\mu} > \mu^*\} = \frac{1}{2}\ln\left(1+\frac{\Delta_i^2}{\sigma_i^2}\right), \tag{46}$$

where again $\mathbb{I}(f;g)$ is the Kullback-Leibler divergence between densities f and g. Taking the densities here as normal, we have

$$\begin{aligned} \mathbb{I}(f_{(\mu_i,\sigma_i^2)}; f_{(\tilde{\mu},\tilde{\sigma}^2)}) &= \int_{-\infty}^{\infty} \ln\left(\frac{f_{(\mu_i,\sigma_i^2)}(x)}{f_{(\tilde{\mu},\tilde{\sigma}^2)}(x)}\right) f_{(\mu_i,\sigma_i^2)}(x) dx \\ &= \int_{-\infty}^{\infty} \left(-\frac{(x-\mu_i)^2}{2\sigma_i^2} + \frac{(x-\tilde{\mu})^2}{2\tilde{\sigma}^2} + \ln\left(\frac{\tilde{\sigma}}{\sigma_i}\right)\right) \frac{1}{\sigma_i\sqrt{2\pi}} e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}} dx \\ &= \frac{(\tilde{\mu}-\mu_i)^2 + (\sigma_i^2 - \tilde{\sigma}^2)}{2\tilde{\sigma}^2} + \ln\left(\frac{\tilde{\sigma}}{\sigma_i}\right). \end{aligned} \tag{47}$$

Restricting to $\tilde{\mu} > \mu^*$ and $\tilde{\sigma}^2 > 0$, the infimum is realized (since $\mu^* > \mu_i$) taking $\tilde{\mu} = \mu^*$ and $\tilde{\sigma}^2 = (\mu^* - \mu_i)^2 + \sigma_i^2$, yielding

$$\mathbb{K}((\mu_i, \sigma_i^2), \mu^*) = \frac{1}{2}\ln\left(1+\frac{(\mu^*-\mu_i)^2}{\sigma_i^2}\right) = \frac{1}{2}\ln\left(1+\frac{\Delta_i^2}{\sigma_i^2}\right). \tag{48}$$

■

Proposition 8 *For a χ^2_k random variable U_k, and $\varepsilon > 0$,*

$$\mathbb{P}(U_k > k(1+\varepsilon)) \leqslant (e^{-\varepsilon}(1+\varepsilon))^{k/2}. \tag{49}$$

Proof Let $r > 0$, and let Z be a standard normal random variable. We have that

$$\mathbb{P}(U_k > k(1+\varepsilon)) = \mathbb{P}(e^{rU_k} > e^{rk(1+\varepsilon)}) \leqslant \frac{\mathbb{E}\left[e^{rU_k}\right]}{e^{rk(1+\varepsilon)}} = \frac{\mathbb{E}\left[e^{rZ^2}\right]^k}{e^{rk(1+\varepsilon)}}, \tag{50}$$

the last step following from viewing the U_k as the sum of k independent squared standard normals. Hence,

$$\mathbb{P}(U_k > k(1+\varepsilon)) \leqslant \left(\frac{\mathbb{E}\left[e^{rZ^2}\right]}{e^{r(1+\varepsilon)}}\right)^k = \left(\frac{1}{e^{r(1+\varepsilon)}\sqrt{1-2r}}\right)^k, \tag{51}$$

if $0 < r < 1/2$. Taking $r = (1/2)(\varepsilon/(1+\varepsilon))$ completes the result. ■

Proposition 9 *Conjecture 1 is* **false** *and for each i, for* $\varepsilon > 0$*,*

$$\frac{\mathbb{P}\left(\bar{X}_j^i + S_i(j)\sqrt{k^{2/j}-1} < \mu_i - \varepsilon \text{ for some } 2 \leqslant j \leqslant k\right)}{1/k} \to \infty \text{ as } k \to \infty. \tag{52}$$

.

Proof Define the events $A^i_{j,k,\varepsilon} = \{\bar{X}_j^i + S_i(j)\sqrt{k^{2/j}-1} < \mu_i - \varepsilon\}$. As the samples are taken to be normally distributed with mean μ_i and variance σ_i^2, we have that $\bar{X}_j^i - \mu_i \sim Z\sigma_i/\sqrt{j}$ and $S_i^2(j) \sim \sigma_i^2 U/j$, where Z is a standard normal, $U \sim \chi^2_{j-1}$, and Z, U independent. Hence,

$$\mathbb{P}(A^i_{j,k,\varepsilon}) = \mathbb{P}\left(Z\frac{\sigma_i}{\sqrt{j}} + \sqrt{U\frac{\sigma_i^2}{j}}\sqrt{k^{2/j}-1} < -\varepsilon\right) = \mathbb{P}\left(\frac{\varepsilon}{\sigma_i}\sqrt{j} + \sqrt{U}\sqrt{k^{2/j}-1} < Z\right). \tag{53}$$

The last step is simply a re-arrangement, and an observation on the symmetry of the distribution of Z. For $j \geqslant 3$, we may apply Proposition 3 here for $d = j-1$, $p = j$, to yield

$$\mathbb{P}(A^i_{j,k,\varepsilon}) \geqslant \frac{1}{2}\frac{k^{1/j}}{k}\mathbb{P}\left(\frac{1}{4}Z^2 \geqslant U \geqslant \frac{\varepsilon^2}{\sigma_i^2}j\right). \tag{54}$$

For a fixed $j_0 \geqslant 3$, for $k \geqslant j_0$ we have

$$\mathbb{P}\left(A^i_{j,k,\varepsilon} \text{ for some } 2 \leqslant j \leqslant k\right) \geqslant \mathbb{P}(A^i_{j_0,k,\varepsilon}) \geqslant O(1/k)\,k^{1/j_0}. \tag{55}$$

The proposition follows immediately. ■

Proposition 10 *For* $G > 0$*,* $0 \leqslant \varepsilon < 1/2$*, the following holds:*

$$\frac{1}{\ln\left(1+G\frac{(1-\varepsilon)^2}{1+\varepsilon}\right)} \leqslant \frac{1}{\ln(1+G)} + \frac{10G}{(1+G)\left(\ln(1+G)\right)^2}\varepsilon. \tag{56}$$

Proof For any $G > 0$, the function $1/\ln\left(1+G\frac{(1-\varepsilon)^2}{1+\varepsilon}\right)$ is positive, increasing, and convex on $\varepsilon \in [0,1)$ (Proposition 11). For a given $G > 0$, noting that the above inequality holds (as equality) at $\varepsilon = 0$, due to the convexity it suffices to show that the inequality is satisfied at $\varepsilon = 1/2$, or

$$\frac{1}{\ln\left(1+\frac{G}{6}\right)} \leqslant \frac{5G}{(1+G)\left(\ln(1+G)\right)^2} + \frac{1}{\ln(1+G)}. \tag{57}$$

Equivalently, we consider the inequality

$$0 \leqslant \frac{5G}{(1+G)} + \ln(1+G) - \frac{\left(\ln(1+G)\right)^2}{\ln\left(1+\frac{G}{6}\right)}. \tag{58}$$

Define the function $F(G)$ to be the RHS of Ineq. (58). Note that as $G \to 0$, $F(G) \to 0$, and in simplified form we have (for $G > 0$ and the limit as $G \to 0$),

$$F'(G) = \frac{\left((1+G)\ln(1+G) - (6+G)\ln\left(1+\frac{G}{6}\right)\right)^2}{(1+G)^2(6+G)\ln\left(1+\frac{G}{6}\right)^2} \geqslant 0. \tag{59}$$

It follows that $F(G) \geqslant 0$, and hence the desired inequality holds at $\varepsilon = 1/2$. This completes the proof. ■

Proposition 11 *The function* $H_G(\varepsilon) = 1/\ln\left(1+G\frac{(1-\varepsilon)^2}{1+\varepsilon}\right)$ *is positive, increasing, and convex in* $\varepsilon \in [0,1)$*, for any constant* $G > 0$*.*

Proof That $H_G(\varepsilon)$ is positive and increasing in ε, follows immediately from inspection of H_G and H'_G, given the hypotheses on G, and ε.

To demonstrate convexity, by inspection of the terms of $H''_G(\varepsilon)$, it suffices to show that for all relevant G, and ε, the following inequality holds.

$$2G(1-\varepsilon)^2(3+\varepsilon)^2 + \left(-8(1+\varepsilon) + G(1-\varepsilon)^2(1+\varepsilon(6+\varepsilon))\right)\ln\left(1+G\frac{(1-\varepsilon)^2}{1+\varepsilon}\right) \geqslant 0. \tag{60}$$

Defining $C = G(1-\varepsilon)^2/(1+\varepsilon)$, it is sufficient to show that for all $C > 0$ and $\varepsilon \in [0,1)$ (eliminating a factor of $(1+\varepsilon)$ from the above),

$$2C(3+\varepsilon)^2 + (-8 + C(1+\varepsilon(6+\varepsilon)))\ln(1+C) \geqslant 0. \tag{61}$$

Defining $J_C(\varepsilon)$ as the LHS of the above, note that $J_C'(\varepsilon) = 2C(3+\varepsilon)(2+\ln(1+C)) > 0$. It suffices then to show $J_C(0) \geqslant 0$, or $18C + (C-8)\ln(1+C) \geqslant 0$. Note this holds at $C = 0$, and $d/dC[J_C(0)] = (10+19C)/(1+C) + \ln(1+C) > 0$ for $C \geqslant 0$. Hence, $J_C(\varepsilon) \geqslant 0$, and $H_G''(\varepsilon) \geqslant 0$. ■

Proof [of Theorem 2] In the interests of comparing π_{BK} and π_{CHK}, consider a general policy π depending on $a > b$ that initially samples each bandit a times, then for times greater than aN, samples according to the maximal index

$$u_i(n,k) = \bar{X}_k^i + S_i(k)\sqrt{n^{\frac{2}{k-b}} - 1}.$$

Note, π_{BK} corresponds to the choices $a = 2, b = 0$, and π_{CHK} corresponds to the choices $a = 3, b = 2$.

Let i^* be the optimal bandit, and let j be such that $\mu^* = \mu_{i^*} > \mu_j = \max_{k:\mu_k \neq \mu^*}$. Let $\tilde{\varepsilon} = 2\sigma_j$.

First, for $n > aN$, we have the following bound:

$$\sum_{t=aN}^{n} \mathbb{1}\{\pi(t+1) \neq i^*\} \geqslant \mathbb{1}\left\{\bigcap_{k=1}^{\infty}\{\bar{X}_k^j \geqslant \mu_j - \tilde{\varepsilon}\}\right\} \sum_{m=1}^{n-aN+1} \mathbb{1}\left\{\bigcap_{t=aN}^{aN+m-1}\left\{u_{i^*}(t,a) < \mu_j - \tilde{\varepsilon}\right\}\right\}. \tag{62}$$

The above inequality can be seen in the following way: In attempting to bound the sub optimal activations of π beyond time $t = aN$ from below, we may restrict ourselves to the event that the sample mean for $j \neq i^*$ is *never* below $\mu_j - \tilde{\varepsilon}$ (and hence, the index for j is never below $\mu_j - \tilde{\varepsilon}$) and count only the initial consecutive non-activations of i^* beyond time $t = aN$. The number of these initial consecutive non-activations, restricted in this way, is bound from below by the number of times the index for i^* is consecutively below $\mu_j - \tilde{\varepsilon}$, counted by the righthand sum.

Noting that $u_{i^*}(t,a)$ is an increasing function of t, we have that

$$\begin{aligned}
&\sum_{m=1}^{n-aN+1} \mathbb{1}\left\{\bigcap_{t=aN}^{aN+m-1}\left\{u_{i^*}(t,a) < \mu_j - \tilde{\varepsilon}\right\}\right\} \\
&= \sum_{m=1}^{n-aN+1} \mathbb{1}\left\{u_{i^*}(aN+m-1,a) < \mu_j - \tilde{\varepsilon}\right\} \\
&= \sum_{m=1}^{n-aN+1} \mathbb{1}\left\{\bar{X}_a^{i^*} + S_{i^*}(a)\sqrt{(aN+m-1)^{\frac{2}{a-b}} - 1} < \mu_j - \tilde{\varepsilon}\right\} \\
&= \mathbb{1}\left\{\bar{X}_a^{i^*} < \mu_j - \tilde{\varepsilon}\right\} \sum_{m=1}^{n-aN+1} \mathbb{1}\left\{m < \left(\left(\frac{(\mu_j - \tilde{\varepsilon}) - \bar{X}_a^{i^*}}{S_{i^*}(a)}\right)^2 + 1\right)^{\frac{a-b}{2}} + 1 - aN\right\} \qquad (63) \\
&\geqslant \mathbb{1}\left\{\bar{X}_a^{i^*} < \mu_j - \tilde{\varepsilon}\right\} \min\left\{n - aN + 1, \left(\left(\frac{(\mu_j - \tilde{\varepsilon}) - \bar{X}_a^{i^*}}{S_{i^*}(a)}\right)^2 + 1\right)^{\frac{a-b}{2}} - aN\right\} \\
&\geqslant \mathbb{1}\left\{\bar{X}_a^{i^*} < \mu_j - \tilde{\varepsilon}\right\} \min\left\{n, \left(\left(\frac{(\mu_j - \tilde{\varepsilon}) - \bar{X}_a^{i^*}}{S_{i^*}(a)}\right)^2 + 1\right)^{\frac{a-b}{2}}\right\} - aN.
\end{aligned}$$

From the above, we have that

$$\begin{aligned}
&\sum_{t=aN}^{n} \mathbb{1}\{\pi(t+1) \neq i^*\} \\
&\geqslant \mathbb{1}\left\{\bigcap_{k=1}^{\infty}\{\bar{X}_k^j \geqslant \mu_j - \tilde{\varepsilon}\}\right\} \mathbb{1}\left\{\bar{X}_a^{i^*} < \mu_j - \tilde{\varepsilon}\right\} \min\left\{n, \left(\left(\frac{(\mu_j - \tilde{\varepsilon}) - \bar{X}_a^{i^*}}{S_{i^*}(a)}\right)^2 + 1\right)^{\frac{a-b}{2}}\right\} - aN.
\end{aligned} \tag{64}$$

To compute the relevant expectations, note that (recycling the bound from Eq. (31)),

$$\mathbb{P}\left(\bigcap_{k=1}^{\infty}\{\bar{X}_k^j \geqslant \mu_j - \tilde{\varepsilon}\}\right) = 1 - \mathbb{P}\left(\bigcup_{k=1}^{\infty}\{\bar{X}_k^j < \mu_j - \tilde{\varepsilon}\}\right) \geqslant 1 - \sum_{k=1}^{\infty}\mathbb{P}\left(\bar{X}_k^j < \mu_j - \tilde{\varepsilon}\right) \geqslant \frac{1}{2}. \tag{65}$$

Hence,

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=aN}^{n}\mathbb{1}\{\pi(t+1)\neq i^*\}\right]+aN\\
&\geqslant\frac{1}{2}\mathbb{E}\left[\mathbb{1}\left\{\bar{X}_a^{i^*}<\mu_j-\tilde{\varepsilon}\right\}\min\left\{n,\left(\left(\frac{(\mu_j-\tilde{\varepsilon})-\bar{X}_a^{i^*}}{S_{i^*}(a)}\right)^2+1\right)^{\frac{a-b}{2}}\right\}\right]\\
&=\frac{1}{2}\mathbb{E}\left[\mathbb{1}\left\{\Delta_j+\tilde{\varepsilon}+\sigma_{i^*}Z/\sqrt{a}<0\right\}\min\left\{n,\left(\left(\frac{\Delta_j+\tilde{\varepsilon}+\sigma_{i^*}Z/\sqrt{a}}{\sigma_{i^*}\sqrt{U}/\sqrt{a}}\right)^2+1\right)^{\frac{a-b}{2}}\right\}\right]\\
&=\frac{1}{2}\mathbb{E}\left[\mathbb{1}\left\{\tilde{\Delta}+Z<0\right\}\min\left\{n,\left(\left(\frac{\tilde{\Delta}+Z}{\sqrt{U}}\right)^2+1\right)^{\frac{a-b}{2}}\right\}\right],
\end{aligned}\tag{66}$$

recalling that $\bar{X}_a^{i^*}\sim\mu^*+\sigma_{i^*}Z/\sqrt{a}$ and $S_{i^*}(a)\sim\sigma_{i^*}^2U/a$ where Z,U are independent, Z a standard normal and U a χ^2_{a-1} random variable, and taking $\tilde{\Delta}=\sqrt{a}(\Delta_j+\tilde{\varepsilon})/\sigma_{i^*}>0$. Taking $d=a-1$,

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=aN}^{n}\mathbb{1}\{\pi(t+1)\neq i^*\}\right]+aN\\
&\geqslant O(1)\int_0^\infty\int_{-\infty}^{-\tilde{\Delta}}\min\left\{n,\left(\left(\frac{\tilde{\Delta}+z}{\sqrt{u}}\right)^2+1\right)^{\frac{a-b}{2}}\right\}e^{-z^2/2}u^{\frac{d}{2}-1}e^{-u/2}dzdu.
\end{aligned}\tag{67}$$

Taking the transformation $(z,u)=(-\tilde{\Delta}-\cos(\theta)\sqrt{r},r\sin(\theta)^2)$, for $r\in[0,\infty)$, $\theta\in[0,\pi/2]$, we have $dzdu=2\sin(\theta)\sqrt{r}drd\theta$, and

$$\begin{aligned}
&\int_0^\infty\int_{-\infty}^{-\tilde{\Delta}}\min\left\{n,\left(\left(\frac{\tilde{\Delta}+z}{\sqrt{u}}\right)^2+1\right)^{\frac{a-b}{2}}\right\}e^{-z^2/2-u/2}u^{\frac{d}{2}-1}dzdu\\
&=2\int_0^{\pi/2}\int_0^\infty\min\left\{n,\csc(\theta)^{a-b}\right\}e^{-\frac{r}{2}-\tilde{\Delta}\cos(\theta)\sqrt{r}-\frac{\tilde{\Delta}^2}{2}}r^{\frac{d-1}{2}}\sin(\theta)^{d-1}drd\theta\\
&\geqslant 2\int_0^{\pi/2}\int_0^\infty\min\left\{n,\csc(\theta)^{a-b}\right\}e^{-\frac{r}{2}-\tilde{\Delta}\sqrt{r}-\frac{\tilde{\Delta}^2}{2}}r^{\frac{d-1}{2}}\sin(\theta)^{d-1}drd\theta\\
&=2\left(\int_0^\infty e^{-\frac{1}{2}(\tilde{\Delta}+\sqrt{r})^2}r^{\frac{d-1}{2}}dr\right)\left(\int_0^{\pi/2}\min\left\{n,\csc(\theta)^{a-b}\right\}\sin(\theta)^{a-2}d\theta\right)\\
&\geqslant 2\left(\int_0^\infty e^{-\frac{1}{2}(\tilde{\Delta}+\sqrt{r})^2}r^{\frac{d-1}{2}}dr\right)\left(\int_{\arcsin\left(n^{-\frac{1}{a-b}}\right)}^{\pi/2}\sin(\theta)^{b-2}d\theta\right).
\end{aligned}\tag{68}$$

From the above, for $b\geqslant 2$, the above integral converges to a constant as $n\to\infty$, and in that sense the bound is uninformative, giving an $O(1)$ lower bound. For $b<2$, taking the bounds that $\theta\geqslant\sin(\theta)$ on the indicated range, and $\arcsin(x)\leqslant\pi/2x$ for $x\in[0,1]$, we have

$$\mathbb{E}\left[\sum_{t=aN}^{n}\mathbb{1}\{\pi(t+1)\neq i^*\}\right]+aN\geqslant O(1)\int_{\frac{\pi}{2}n^{-\frac{1}{a-b}}}^{\pi/2}\theta^{b-2}d\theta=O(1)\int_{n^{-\frac{1}{a-b}}}^1\tau^{b-2}d\tau.\tag{69}$$

Noting that $R_\pi(n)\geqslant\Delta_j\mathbb{E}[\sum_{t=aN}^n\mathbb{1}\{\pi(t+1)\neq i^*\}]$, we may therefore summarize as

$$R_\pi(n)\geqslant\begin{cases}O(1) & \text{if } b>1,\\ O(\ln n) & \text{if } b=1,\\ O\left(\frac{n^{\frac{1-b}{a-b}}-1}{1-b}\right) & \text{if } b<1.\end{cases}\tag{70}$$

While the above bound is uninformative in the case of $\pi=\pi_{\mathrm{CHK}}$ (with $a=3,b=2$), it follows that $\pi=\pi_{\mathrm{BK}}$ (with $a=2,b=0$) suffers from at least $O(\sqrt{n})$ regret. ■

Journal of Machine Learning Research 18 (2018) 1-33 Submitted 5/15; Revised 1/17; Published 4/18

Cost-Sensitive Learning with Noisy Labels

Nagarajan Natarajan nagarajn@microsoft.com*
Microsoft Research,
Bangalore 560001, INDIA

Inderjit S. Dhillon inderjit@cs.utexas.edu
Dept. of Computer Science
University of Texas at Austin
Austin, TX 78701

Pradeep Ravikumar pradeepr@cs.cmu.edu
Machine Learning Dept.
Carnegie Mellon University
Pittsburgh, PA 15213

Ambuj Tewari tewaria@umich.edu
Dept. of Statistics, and
Dept. of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109

Editor: Guy Lebanon

Abstract

We study binary classification in the presence of *class-conditional* random noise, where the learner gets to see labels that are flipped independently with some probability, and where the flip probability depends on the class. Our goal is to devise learning algorithms that are efficient and statistically consistent with respect to commonly used utility measures. In particular, we look at a family of measures motivated by their application in domains where cost-sensitive learning is necessary (for example, when there is class imbalance). In contrast to most of the existing literature on consistent classification that are limited to the classical 0-1 loss, our analysis includes more general utility measures such as the AM measure (arithmetic mean of True Positive Rate and True Negative Rate). For this problem of cost-sensitive learning under class-conditional random noise, we develop two approaches that are based on suitably modifying surrogate losses. First, we provide a simple unbiased estimator of any loss, and obtain performance bounds for empirical utility maximization in the presence of i.i.d. data with noisy labels. If the loss function satisfies a simple symmetry condition, we show that using unbiased estimator leads to an efficient algorithm for empirical maximization. Second, by leveraging a reduction of risk minimization under noisy labels to classification with weighted 0-1 loss, we suggest the use of a simple weighted surrogate loss, for which we are able to obtain strong utility bounds. This approach implies that methods already used in practice, such as biased SVM and weighted logistic regression, are provably noise-tolerant. For two practically important measures in our family, we show that the proposed methods are competitive with respect to recently proposed methods for dealing with label noise in several benchmark data sets.

Keywords: class-conditional label noise, statistical consistency, cost-sensitive learning

1. Introduction

Learning from noisy training data is a problem of theoretical as well as practical interest in machine learning. In many applications such as learning to classify images, it is often the case that the labels are noisy. Even human labelers are susceptible to errors in labeling; for instance, certain image categories may be hard to discern. Designing learning algorithms that help maximize a desired performance measure in such noisy settings, and understanding their statistical consistency properties are the objectives of our current work.

One of the earliest known attempts at learning in the presence of label noise was by Bylander (1994) that concerned learnability of linear threshold functions (LTFs) in the Probably Approximately Correct (PAC) model. In particular, he showed that if the noise rate is uniform and if there is a sufficient margin under the clean distribution, then it is possible to PAC-learn LTFs. He also extended the result to a more realistic noise model called monotonic noise (Bylander, 1998), where the noise rate is allowed to vary per example, but is assumed to be a monotonic function of the distance of the example from the true hyperplane. Blum and Mitchell (1998), and later Cohen (1997) improved the PAC-learnability results of Bylander (1994) showing that linear threshold functions are efficiently learnable without the margin requirement in the uniform label noise model. A Bayesian approach to the problem of noisy labels is taken by Graepel and Herbrich (2000) and Lawrence and Schölkopf (2001). Cesa-Bianchi et al. (2011) focus on online learning algorithms where only unbiased estimates of the gradient of the loss are needed to provide guarantees for learning with noisy data. However, they consider a much harder noise model where instances *as well as* labels are noisy. Because of the harder noise model, they necessarily require multiple noisy copies per clean example and the unbiased estimation schemes also become fairly complicated, particularly for non-smooth classification losses such as the hinge loss.

In order to more clearly understand the impact of label noise, it is useful to consider a more natural and simpler formalism for label noise, where a random noise process corrupts the labels (Biggio et al., 2011), which otherwise arise from some "clean" distribution. There has been a long line of work in the theoretical machine learning community on such formalisms. Soon after the introduction of the noise-free PAC model, Angluin and Laird (1988) proposed the *random classification noise* (RCN) model where each label is flipped independently with some probability $\rho \in [0, 1/2)$. It is known (Aslam and Decatur, 1996; Cesa-Bianchi et al., 1999) that finiteness of the VC dimension characterizes learnability in the RCN model. Similarly, in the online mistake bound model, the parameter that characterizes learnability without noise — the Littlestone dimension — continues to characterize learnability even in the presence of random label noise (Ben-David et al., 2009). These results are for the so-called 0-1 loss: if the true label is $y \in \{-1, +1\}$ and the prediction is t, the 0-1 loss defined as $\ell_{0\text{-}1}(y, t) = 1_{\{yt \leq 0\}}$ (where $1_{\{P\}}$ denotes the indicator function that takes value 1 if the predicate P is true or 0 otherwise), is a non-convex function of the prediction t. On the other hand, learning with convex losses has been addressed only under limiting assumptions like separability or uniform noise rates (Manwani and Sastry,

∗. The work in this manuscript was done when the author was a graduate student at the University of Texas at Austin.

2013). A great deal of practical work has also been done on the problem; see, for instance, the survey article by Nettleton et al. (2010).

In this paper, we consider the *class-conditional* random label noise (abbreviated CCN) setting. Here, the data consists of iid samples drawn from a noisy version D_ρ of an underlying "clean" distribution D, and where the noise rates depend on the class label. To the best of our knowledge, general results in this setting have not been obtained before. We note that developing guarantees in the presence of CCN label noise also has implications in varied partially-supervised settings such as learning from only positive and unlabeled data (Elkan and Noto, 2008), which can be cast under this setting. For the theoretical results presented in this work, we assume that the true noise rates (that characterize D_ρ) are known. In practice, one may use the domain knowledge to provide an estimate for noise rates (see Section 6.3), or use a plug-in estimator for noise rates such as the one prescribed by Scott (2015).

A key facet of the classification problem is the evaluation metric which captures discrepancy between the predicted label and the true label, and which we would want to minimize (or correspondingly, an evaluation utility measure which we would want to maximize). While classification accuracy is a popular utility measure, many other performance measures have also been considered in practice. One important family of measures constitutes cost-sensitive learning, and is motivated by applications and domains where misclassification cost could depend on the category of the example. For example, in disease diagnosis, false positives and false negatives often have very different associated impacts. Most, if not all, of the existing theoretical work on classification focuses on obtaining consistent learning algorithms for the 0-1 loss or its surrogates. In this paper, we consider a general class of utility measures that can be expressed as a linear combination of the entries of the "confusion matrix," namely, true positives, true negatives, false positives and false negatives.

Towards this problem of learning classifiers with respect to general utility measures and class conditional label noise, we develop two methods for suitably modifying *any given surrogate loss function* ℓ, and show that minimizing the sample average of the modified proxy loss function $\tilde{\ell}$ leads to provable utility bounds where the utility is calculated on the clean distribution.

In our first approach, the modified or proxy loss is an unbiased estimate of the loss function associated with the utility measure of interest. The idea of using unbiased estimators is well-known in stochastic optimization (Nemirovski et al., 2009). Nonetheless, we bring out some important aspects of using unbiased estimators of loss functions for empirical utility maximization under CCN. In particular, we give a simple symmetry condition on the loss (enjoyed, for instance, by the Huber, logistic, and squared losses) to ensure that the proxy loss is also convex. Hinge loss does not satisfy the symmetry condition, and thus leads to a non-convex problem. We nonetheless provide a convex surrogate, leveraging the fact that the non-convex hinge problem is "close" to a convex problem (Theorem 12). This is strikingly different from the online learning setting (examined in Section 4) that requires only the expected loss to be convex.

Our second approach is based on the fundamental observation that the minimizer of the risk (i.e. probability of misclassification) under the noisy distribution differs from that of the clean distribution *only* in where it thresholds $\eta(x) = P(Y = 1|x)$ to decide the label. In order to correct for the threshold, we then propose a simple weighted loss function, where

the weights are label-dependent, as the proxy loss function. Our analysis builds on the notion of consistency of weighted loss functions studied by Scott (2012). This approach leads to a remarkable result that appropriately weighted losses like biased SVMs studied by Liu et al. (2003) are robust to CCN.

The key contributions of the paper are summarized below:

1. We develop methods for learning, that are provably consistent, (a) in the presence of asymmetric label noise, and (b) with respect to general cost-sensitive utility measures beyond the classical 0-1 loss.

2. To the best of our knowledge, we are the first to provide guarantees for cost-sensitive learning under random label noise in the general setting of convex surrogates, without any assumptions on the true distribution.

3. As one consequence of our results, we resolve an elusive theoretical gap in the understanding of practical methods like biased SVM and weighted logistic regression: as we show, they are provably noise-tolerant (Theorem 18). We obtain the result as a consequence of being able to linearly relate the risk w.r.t. a weighted 0-1 loss under the noisy distribution to that w.r.t. the 0-1 loss under the clean distribution (Theorem 16).

4. Our proxy losses are easy to compute: the proposed approaches yield efficient algorithms.

5. Experiments on benchmark data sets show that the methods are robust even at high noise rates, for maximizing different performance measures from our family.

In a preliminary version of the paper (Natarajan et al., 2013), we provided guarantees for risk minimization (using the 0-1 loss) in the presence of class-conditional label noise. In this paper, we provide a more general and detailed treatment of the theory, by characterizing the optimal classifiers under more general performance measures used in practice (in Sections 4 and 5). We extend our earlier approach (Natarajan et al., 2013) to cost-sensitive learning (Section 3.2). Our results in this paper also serve to generalize some of the known consistency results for performance measures such as the AM measure, even in the noise-free setting.

We now expand on the organization of the paper. We begin by discussing some closely related work in Section 2. We introduce and set the problem up formally in Section 3. The class-conditional noise model is specified by two parameters ρ_{+1} and ρ_{-1} which correspond to the rates at which positive and negative labels are flipped (independently) respectively. To build the theory, we assume that the rates are known to the learner. We do not make any assumptions on the underlying distribution. We introduce the family of measures $\mathcal{U}$ that constitute cost-sensitive learning in Section 3. It is well-known that the classical 0-1 loss is optimized by thresholding $P(Y = 1|x)$ at $1/2$. Cost-sensitive measures are of particular interest to our work in this paper because their optimal decision function exhibits a simple form — thresholding the conditional probability $P(Y = 1|x)$ at a certain value (stated in Lemma 2). Study of consistent learning algorithms, for many performance measures other than classification accuracy, is limited even in the noise-free case. Menon et al. (2013)

showed consistency of certain empirical estimation algorithms for the AM measure (defined in Proposition 1). An important result that connects the excess risk of a decision function (in terms of the 0-1 loss), $R(f) - \min_f R(f)$, and its "utility deficit", $\max_f \mathcal{U}(f) - \mathcal{U}(f)$, is established in Lemma 3. As a consequence of this result, we are able to use the surrogates for 0-1 loss for empirical estimation, in order to maximize cost-sensitive measures.

We describe our first approach of using unbiased surrogate loss functions in Section 4. Here, the idea is to construct an unbiased estimator of a given loss function (a surrogate of the 0-1 loss). The unbiased estimator involves the noise rates ρ_{+1} and ρ_{-1}. For optimizing a given utility measure $\mathcal{U}$, we propose an empirical risk minimization procedure based on the unbiased surrogate thus obtained. We establish utility deficit bounds for the resulting empirical estimator in Theorem 9. Here, we also look at the online learning setting, where examples arrive sequentially (with noisy labels), and obtain similar consistency guarantees. Our second approach is detailed in Section 5. The key observation is that the optimal decision function for utility $\mathcal{U}$ in our family with respect to the noisy distribution is simply given by thresholding $P(Y = 1|x)$ with respect to the *clean* distribution, at a certain value that depends only on the distribution and the measure $\mathcal{U}$ itself. This enables us to use a weighted surrogate of the 0-1 loss, where the weights depend on the measure $\mathcal{U}$ and noise rates ρ_{+1} and ρ_{-1}. We provide rigorous guarantees for consistency of the resulting empirical estimator in Theorem 18.

We present detailed experimental results that support our theory in Section 6. We perform experiments on synthetic and benchmark data sets, on both the proposed approaches. We compare to state-of-the-art algorithms for learning with noisy data on different data sets and different noise settings. We use two performance measures in experiments: classification accuracy and the AM measure, as representatives of the family of measures considered in the paper.

2. Related Work

Stempfel and Ralaivola (2009) propose minimizing an unbiased proxy for the case of the hinge loss. However the hinge loss leads to a non-convex problem. Therefore, they propose heuristic minimization approaches for which no theoretical guarantees are provided. We address the issue in Section 4.1. As Adaboost is very sensitive to label noise, random label noise has also been considered in the context of boosting. Freund (2009) proposes a boosting algorithm based on a non-convex potential that is empirically seen to be robust against random label noise. Long and Servedio (2010) prove that any method based on a convex potential is inherently ill-suited to random label noise. Biggio et al. (2011) consider robust SVM formulation in the presence of random and adversarial label noise. However, they do not provide any theoretical justification. Practitioners have developed several noise tolerant versions of the perceptron algorithm, although many are heuristic and are not known to be provably robust. This includes the passive-aggressive family of algorithms (Crammer et al., 2006), confidence weighted learning (Dredze et al., 2008), AROW (Crammer et al., 2009) and the NHERD algorithm (Crammer and Lee, 2010). The survey article by Khardon and Wachman (2007) provides an overview of some of this literature. To the best of our knowledge, there are no known mistake-bounded perceptron algorithms under asymmetric label noise.

Manwani and Sastry (2013) consider whether empirical risk minimization of the loss itself on the noisy data is a good idea when the goal is to obtain small risk under the clean distribution. But the answer is affirmative only for 0-1 and squared losses. Therefore, if empirical risk minimization over noisy samples has to work, we necessarily have to change the loss used to calculate the empirical risk. More recently, Ghosh et al. (2014) prove that a loss function ℓ satisfying the symmetry condition $\ell(f(\mathbf{x}), 1) + \ell(f(\mathbf{x}), -1) = C, \forall \mathbf{x}, \forall f$ for some constant C are noise-tolerant, under the assumption that the classes are separable under the clean distribution (here, ℓ is said to be *noise-tolerant* if $E_{(X,Y)\sim D}\big[\ell_{0\text{-}1}(f^*_\ell(X), Y)\big] = E_{(X,Y)\sim D}\big[\ell_{0\text{-}1}(\tilde{f}^*_\ell(X), Y)\big]$, where f^*_ℓ and $\tilde{f}^*_\ell$ denote the minimizers of ℓ-risk under clean and noisy distribution respectively). Furthermore, they show that by choosing a sufficiently large value of a parameter in the loss functions such as sigmoid loss, ramp loss and probit loss, the losses can be made tolerant to non-uniform label noise (i.e. noise rate is allowed to depend on the example) as well. Unfortunately, the aforementioned loss functions are all non-convex, and convex losses used in practice do not satisfy the sufficiency conditions. It remains an open question if the symmetry condition is indeed necessary for noise tolerance, at least under the separability assumption.

van Rooyen and Williamson (2015) extend the idea behind the method of unbiased estimators to more general learning settings beyond binary classification. For example, they consider semi-supervised learning, classification with more than two classes, and learning with partial labels (a partial label is a *set* of labels containing the true label).

Scott et al. (2013) also study the problem of learning classifiers under the class-conditional noise model. However, they approach the problem from a different set of assumptions — the noise rates are *not* known, and the true distribution satisfies a certain "mutual irreducibility" property. They model the observed noisy instances as arising from "contaminated" mixtures of positive and negative classes and show that the mixture proportions can be consistently estimated by *maximal denoising* of the noisy distributions. Blanchard and Scott (2014) establish similar results for the multi-class classification problem. Scott (2015) provides a consistent estimator with convergence rates for the cost parameter α in our weighted surrogate loss (Equation 2). In this paper, however, we select α by cross-validation and also examine the sensitivity of selecting α (in Section 6.3).

3. Preliminaries

Let D be the underlying true distribution generating $(X, Y) \in \mathcal{X} \times \{\pm 1\}$ pairs from which n iid samples $(X_1, Y_1), \ldots, (X_n, Y_n)$ are drawn. Let $\eta(X) = P(Y = 1|X)$ under D.

3.1 Class-conditional Noise

After injecting random classification noise (independently for each i) into these samples, corrupted samples $(X_1, \tilde{Y}_1), \ldots, (X_n, \tilde{Y}_n)$ are obtained. The class-conditional random noise model (CCN, for short) is given by:

$$P(\tilde{Y} = -1|Y = +1) = \rho_{+1},$$
$$P(\tilde{Y} = +1|Y = -1) = \rho_{-1}, \text{ and}$$
$$\rho_{+1} + \rho_{-1} < 1.$$

The corrupted samples are what the learning algorithm sees. We will assume that the noise rates ρ_{+1} and ρ_{-1} are known to the learner. Let the distribution of $(X, \tilde{Y})$ be D_ρ. Noisy labels are denoted by $\tilde{y}$. Let $\tilde{\eta}(X) = P(\tilde{Y} = 1|X)$ under D_ρ.

3.2 Cost-sensitive Classification

Let $f : \mathcal{X} \to \mathbb{R}$ denote a real-valued decision function. The goal in classification is to learn f from a training sample, such that some cost or loss measure is minimized. The most common measure is the probability of misclassification, also called the **risk**, which is simply the expected 0-1 loss defined as

$$R(f) := R_D(f) := \mathbb{E}_{(X,Y)\sim D}\left[1_{\{\text{sign}(f(X))\neq Y\}}\right].$$

Minimizing the 0-1 loss on a training sample, over some class of decision functions, is often intractable. In practice, it is common to minimize a *surrogate* loss function that is chosen for its computational advantages such as convexity. Minimizing risk (or equivalently, maximizing the accuracy) of a classifier is however not always appropriate, and in fact practitioners have devised many alternative performance metrics to address specific needs of a target domain. Class imbalance is an important scenario where accuracy of classifier is not a good metric to optimize: a trivial classifier that assigns all the examples to the majority class will have a high accuracy. However, little is known about optimal classification or consistent algorithms for binary classification w.r.t. general performance measures, even when the observations are noise-free. An important family of performance measures that is preferred in many scenarios including heavy class imbalance and asymmetry in real-world costs associated with specific classes constitutes *cost-sensitive* learning. Cost-sensitive performance measures are given by a weighted combination of the four fundamental population quantities associated with the "confusion matrix" - true positives, false positives (also known as type-I error), false negatives (also known as type-II error) and true negatives as defined below:

$$TP(f;D) = E_{(X,Y)\sim D}[1_{\{f(X)=1,Y=1\}}], \quad TN(f;D) = E_{(X,Y)\sim D}[1_{\{f(X)=-1,Y=-1\}}]$$

$$FP(f;D) = E_{(X,Y)\sim D}[1_{\{f(X)=1,Y=-1\}}], \quad FN(f;D) = E_{(X,Y)\sim D}[1_{\{f(X)=-1,Y=1\}}].$$

We consider the family of measures $\mathcal{U}$ defined by:

$$\mathcal{U}(f;D) = a_{11}TP(f;D) + a_{10}FP(f;D) + a_{01}FN(f;D) + a_{00}TN(f;D), \tag{1}$$

given constants $a_0, a_{11}, a_{10}, a_{01}, a_{00}$ (that could depend on D). In the remainder of the paper, $\mathcal{U}$ refers to a measure in this family, unless specified otherwise. We will use the terms *performance* measure and *utility* measure interchangeably in this paper.

Next, we state two important, commonly-used measures in this family.

Proposition 1 *1. The Accuracy measure $\mathcal{U}_{Acc}(f;D)$ belongs in the family (1) with $a_{11} = a_{00} = 1$ and $a_{10} = a_{01} = 0$.*

2. The AM (Arithmetic Mean of TPR and TNR) measure defined as

$$\mathcal{U}_{AM}(f;D) := \frac{1}{2}\big(TPR(f;D) + TNR(f;D)\big),$$

where $TPR(f;D) = P(f(X) = 1|Y = 1)$ is the true positive rate and $TNR(f;D) = P(f(X) = -1|Y = -1)$ is the true negative rate, belongs in the family (1) with constants $a_{10} = a_{01} = 0$, $a_{11} = \frac{1}{2(1-\pi)}$ and $a_{00} = \frac{1}{2\pi}$, where $\pi = P(Y = 1)$ under D.

3.2.1 Cost-sensitive Classification without Label Noise

Before we present our approaches for cost-sensitive classification under class-conditional label noise, it will be useful to consider the setting without such label noise, and setup appropriate notation. Given a utility measure $\mathcal{U}$ and training data, our goal is to learn a decision function f that maximizes $\mathcal{U}$ with respect to the clean distribution. The optimal decision function (called Bayes optimal) that maximizes $\mathcal{U}$ over all real-valued decision functions is denoted as $f^\star(x) := \arg\max_f \mathcal{U}(f;D)$. We denote by $\mathcal{U}^*$ the optimal utility value, i.e. $\mathcal{U}^* = \mathcal{U}(f^\star)$. It is not always possible to characterize the Bayes optimal of arbitrary performance measures. Cost-sensitive measures are particularly interesting because their Bayes optimal exhibits a simple form, and as a consequence, consistent algorithms are readily obtained in practice in the noise-free case. Bayes optimal classifier for the family (1) is characterized in the following Lemma. Recall that $\eta(x) = P(Y = 1|x)$ under D.

Lemma 2 *The Bayes optimal of any measure $\mathcal{U}$ in family (1) is given by*

$$\arg\max_f \mathcal{U}(f;D) = \text{sign}(\eta(x) - \delta_D^*),$$

where the threshold is defined as

$$\delta_D^* = \frac{a_{00} - a_{10}}{a_{00} - a_{10} - a_{01} + a_{11}}.$$

The proof is simple and can be found elsewhere (Elkan, 2001). It is well-known that accuracy $\mathcal{U}_{Acc}(f;D)$ is maximized by $\text{sign}(\eta(x) - 1/2)$ which is also readily obtained by applying the above lemma. For the AM measure $\mathcal{U}_{AM}(f;D)$, one easily verifies that the threshold is $\pi = P(Y = 1)$.

For any general measure $\mathcal{U}$, we are interested in controlling the *deficit utility* which is $\mathcal{U}^* - \mathcal{U}(f;D)$. The following simple lemma relates the deficit utility for the family (1) to that of a certain weighted 0-1 risk.

Lemma 3 *Define α-weighted risk under distribution D as:*

$$R_\alpha(f) := R_{\alpha,D}(f) := E_{(X,Y)\sim D}\Big[(1-\alpha)1_{\{Y=1\}}1_{\{f(X)\leq 0\}} + \alpha 1_{\{Y=-1\}}1_{\{f(X)>0\}}\Big].$$

For any measure $\mathcal{U}$ in the family (1):

$$R_{\delta_D^*}(f) - R^*_{\delta_D^*} = \frac{1}{(a_{11} + a_{00}) - (a_{10} + a_{01})}(\mathcal{U}^* - \mathcal{U}(f;D)),$$

where $R^*_{\delta_D^*} = \min_f R_{\delta_D^*}(f)$.

Proof Let $c_1 = (a_{11}+a_{00})-(a_{10}-a_{01})$ and $c_2 = a_{00}-a_{10}$. From Lemma 2, we know $\delta_D^* = \frac{c_2}{c_1}$. Note that $1 > c_1 > 0$ for otherwise maximizing $\mathcal{U}$ would not make sense (See Remark 4), and therefore $0 \leq \delta_D^* \leq 1$. For any f, let θ denote the classifier $\theta(x) = \text{sign}(f(x))$. We can rewrite $\mathcal{U}(\theta)$ as $\mathcal{U}(\theta) = c_1[(1-\delta_D^*)TP + \delta_D^* TN] + \tilde{A}$, where $\tilde{A}$ is a constant. We have:

$$\begin{aligned}
R_{\delta_D^*}(\theta) &= E_{(X,Y)\sim D}\left[\left((1-\delta_D^*)1_{\{Y=1\}} + \delta_D^* 1_{\{Y=0\}}\right).1_{\{\theta(X)\neq Y\}}\right] \\
&= (1-\delta_D^*)P(Y=1,\theta(X)=-1) + \delta_D^* P(Y=-1,\theta(X)=1) \\
&= (1-\delta_D^*)FN + \delta_D^* FP \\
&= (1-\delta_D^*)(\pi - TP) + \delta_D^*(1-\pi-TN) \\
&= (1-\delta_D^*)\pi + \delta_D^*(1-\pi) - \left((1-\delta_D^*)TP + \delta_D^* TN\right) \\
&= (1-\delta_D^*)\pi + \delta_D^*(1-\pi) + \frac{\tilde{A}}{c_1} - \frac{1}{c_1}\mathcal{U}(\theta).
\end{aligned}$$

Observing that $(1-\delta_D^*)\pi + \delta_D^*(1-\pi) + \frac{\tilde{A}}{c_1}$ is a constant independent of θ, the proof is complete. ∎

Remark 4 *Note that we can assume $(a_{11}+a_{00}) - (a_{10}+a_{01}) > 0$, otherwise maximizing $\mathcal{U}$ would not make sense. If indeed, $(a_{11}+a_{00}) - (a_{10}+a_{01}) < 0$, then Lemma 3 still holds but with $\mathcal{U}^*$ interpreted as $\mathcal{U}^* = \min_f \mathcal{U}(f;D)$.*

Of course, minimizing the α-weighted risk on a training sample is not tractable. Scott (2012) extends the notion of the classification calibration defined by Bartlett et al. (2006) for the (unweighted) 0-1 loss. The following result of Scott (2012) tells us that by using a similarly weighted surrogate loss function ℓ_α, one can control the excess α-weighted risk. Define ℓ_α-risk, $R_{\ell_\alpha,D}(f) = E_{(X,Y)\sim D}[\ell_\alpha(f(X),Y)]$, and $R^*_{\ell_\alpha} = \min_f R_{\ell_\alpha,D}(f)$.

Lemma 5 (α-classification calibration (Scott, 2012)) *Given a loss function $\ell(t,y)$, and $\alpha \in (0,1)$, define the α-weighted loss:*

$$\ell_\alpha(t,y) = \left((1-\alpha)1_{\{y=1\}} + \alpha 1_{\{y=-1\}}\right)\ell(t,y)\ . \tag{2}$$

ℓ_α is α-classification calibrated (or α-CC) iff there exists a convex, non-decreasing and invertible *transformation $\psi_{\ell,\alpha}$, with $\psi_{\ell,\alpha}(0) = 0$, such that*

$$\psi_{\ell,\alpha}(R_\alpha(f) - R^*_\alpha) \leq R_{\ell_\alpha,D}(f) - R^*_{\ell_\alpha}\ .$$

In other words, consistency with respect to ℓ_α-risk implies consistency with respect to α-weighted (0-1) risk for α-CC losses. Also, for any ℓ that is classification-calibrated (Bartlett et al., 2006) (such as logistic, hinge and squared losses), the corresponding ℓ_α is α-CC.

If we choose $\alpha = \delta_D^*$, Lemmas 3 and 5 together guarantee the consistency of using a weighted surrogate loss function in practice, when we obtain samples from the clean distribution D. However, if the labels are noisy, the outlined procedure is no longer consistent. One necessarily has to change the loss function ℓ or rather ℓ_α to be able to tolerate the noise, as described in the next two sections.

Remark 6 *Most commonly used loss functions such as hinge and logistic losses are (even) margin losses, i.e. $\ell(t,y) = \phi(ty)$, for some $\phi : \mathbb{R} \to [0,\infty)$. We could also consider an* uneven margin *loss function of the form:*

$$\ell(t,y) = 1_{\{y=1\}}\phi(t) + 1_{\{y=-1\}}\beta\phi(-\gamma t),$$

for $\beta,\gamma > 0$. Scott (2012) showed that for convex ϕ, the above defined uneven margin loss ℓ is classification-calibrated, and in turn, the corresponding ℓ_α is α-CC, when $\beta = \frac{1}{\gamma}$. Such uneven margin losses have been used in practice mostly as heuristics as pointed out in (Scott, 2012). Thus, in principle, we could use uneven margin losses, and all the results in this manuscript will hold just the same.

Notation. We use letters with the 'tilde' accent to denote noisy versions of quantities or variables, e.g. $\tilde{\ell}$ is the loss function to be used on the noisy data, and $\tilde{y}$ denotes a noisy label. We use $\mathcal{F} : \mathcal{X} \to \mathbb{R}$ to denote a fixed class of real-valued decision functions. If f is not quantified in a minimization, then it is implicit that the minimization is over all measurable functions. Instances are denoted by $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$. Though most of our results apply to a general function class $\mathcal{F}$, we instantiate $\mathcal{F}$ to be the set of hyperplanes of bounded L_2 norm, $\mathcal{W} = \{\mathbf{w} \in \mathbb{R}^d : \|\mathbf{w}\|_2 \leq W_2\}$ for certain specific results.

4. Approach of Unbiased Surrogates

The method of unbiased surrogates uses the noise rates to construct an unbiased estimator $\tilde{\ell}(t,\tilde{y})$ for the loss $\ell(t,y)$. The following key lemma tells us how to construct unbiased estimator of the loss from noisy labels.

Lemma 7 *Let $\ell(t,y)$ be any bounded loss function. Then, if we define,*

$$\tilde{\ell}(t,y) := \frac{(1-\rho_{-y})\,\ell(t,y) - \rho_y\,\ell(t,-y)}{1-\rho_{+1}-\rho_{-1}}$$

we have, for any t,y, $\mathbb{E}_{\tilde{y}}\left[\tilde{\ell}(t,\tilde{y})\right] = \ell(t,y)$. In particular, for any given $\alpha \in (0,1)$, $\mathbb{E}_{\tilde{y}}\left[\tilde{\ell}_\alpha(t,\tilde{y})\right] = \ell_\alpha(t,y)$, where $\ell_\alpha(t,y)$ is defined as in (2).

Proof One could directly compute and see that $\tilde{\ell}$ is unbiased. But to give a little more insight into what motivates the definition of $\tilde{\ell}$, consider the conditions that unbiasedness imposes on it. We should have, for every t,

$$\mathbb{E}_{\tilde{y}\underset{y}{\sim}\rho}\left[\tilde{\ell}(t,\tilde{y})\right] = \ell(t,y)\ .$$

Considering the cases $y=+1$ and $y=-1$ separately, gives the equations

$$\begin{aligned}
(1-\rho_{+1})\tilde{\ell}(t,+1) + \rho_{+1}\tilde{\ell}(t,-1) &= \ell(t,+1)\ , \\
(1-\rho_{-1})\tilde{\ell}(t,-1) + \rho_{-1}\tilde{\ell}(t,+1) &= \ell(t,-1)\ .
\end{aligned}$$

Solving these two equations for $\tilde{\ell}(t,+1)$ and $\tilde{\ell}(t,-1)$ gives

$$\tilde{\ell}(t,+1) = \frac{(1-\rho_{-1})\ell(t,+1) - \rho_{+1}\ell(t,-1)}{1-\rho_{+1}-\rho_{-1}} ,$$
$$\tilde{\ell}(t,-1) = \frac{(1-\rho_{+1})\ell(t,-1) - \rho_{-1}\ell(t,+1)}{1-\rho_{+1}-\rho_{-1}} .$$

The second part of the lemma follows by observing that ℓ_α is bounded too. ■

In Section 3, we saw that in the noise-free case one can bound the deficit utility $\mathcal{U}^* - \mathcal{U}(f;D)$ by using a weighted surrogate loss approach with $\alpha = \delta_D^*$. In the presence of noisy labels, we can try to learn a good predictor that optimizes the measure $\mathcal{U}$ of the form (1) by minimizing the sample average

$$\hat{f} \leftarrow \operatorname*{argmin}_{f\in\mathcal{F}} \widehat{R}_{\tilde{\ell}_\alpha}(f) := \sum_{i=1}^{n} \tilde{\ell}_\alpha(f(X_i),\tilde{Y}_i) . \tag{3}$$

where $\alpha = \delta_D^*$ as before. By unbiasedness of $\tilde{\ell}_\alpha$ (Lemma 7), we know that, for any fixed $f \in \mathcal{F}$, the above sample average converges to $R_{\ell_\alpha,D}(f)$ even though the former is computed using noisy labels whereas the latter depends on the true labels. The following result gives a performance guarantee for this procedure in terms of the Rademacher complexity of the function class $\mathcal{F}$. The main idea in the proof is to use the contraction principle for Rademacher complexity to get rid of the dependence on the proxy loss $\tilde{\ell}_\alpha$. The price to pay for this is L_ρ, the Lipschitz constant of $\tilde{\ell}_\alpha$.

Lemma 8 *Let $\ell(t,y)$ be L-Lipschitz in t (for every y). Then, for any $\alpha \in (0,1)$, with probability at least $1-\delta$,*

$$\max_{f\in\mathcal{F}} |\widehat{R}_{\tilde{\ell}_\alpha}(f) - R_{\tilde{\ell}_\alpha,D_\rho}(f)| \le 2L_\rho\mathfrak{R}(\mathcal{F}) + \sqrt{\frac{\log(1/\delta)}{2n}}$$

where $\mathfrak{R}(\mathcal{F}) := \mathbb{E}_{X_i,\epsilon_i}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\epsilon_i f(X_i)\right]$ is the Rademacher complexity of the function class $\mathcal{F}$ and $L_\rho \le 2L/(1-\rho_{+1}-\rho_{-1})$ is the Lipschitz constant of $\tilde{\ell}_\alpha$. Note that ϵ_i's are iid Rademacher (symmetric Bernoulli) random variables.

Proof By the basic Rademacher bound on the maximal deviation between risks and empirical risks over $f \in \mathcal{F}$, we get

$$\max_{f\in\mathcal{F}} |\widehat{R}_{\tilde{\ell}_\alpha}(f) - R_{\tilde{\ell}_\alpha,D_\rho}(f)| \le 2\cdot\mathfrak{R}(\tilde{\ell}_\alpha\circ\mathcal{F}) + \sqrt{\frac{\log(1/\delta)}{2n}}$$

where

$$\mathfrak{R}(\tilde{\ell}_\alpha\circ\mathcal{F}) := \mathbb{E}_{X_i,\tilde{Y}_i,\epsilon_i}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}\epsilon_i\tilde{\ell}_\alpha(f(X_i),\tilde{Y}_i)\right]$$

If ℓ is L-Lipschitz then for any $\alpha \in (0,1)$, $\tilde{\ell}_\alpha$ is L_ρ Lipschitz for $L_\rho = (1+|\rho_{+1}-\rho_{-1}|)L/(1-\rho_{+1}-\rho_{-1}) \le 2L/(1-\rho_{+1}-\rho_{-1})$ and hence by the Lipschitz composition property of Rademacher averages, we have

$$\mathfrak{R}(\tilde{\ell}_\alpha\circ\mathcal{F}) \le L_\rho\cdot\mathfrak{R}(\mathcal{F}) .$$

■

The above lemma immediately leads to a performance bound for $\hat{f}$ with respect to the clean distribution D. Our first main result is stated in the theorem below. The proof relies on using the α-CC property of the modified surrogate loss function.

Theorem 9 *For any $\alpha \in (0,1)$, with probability at least $1-\delta$,*

$$R_{\ell_\alpha,D}(\hat{f}) \le \min_{f\in\mathcal{F}} R_{\ell_\alpha,D}(f) + 4L_\rho\mathfrak{R}(\mathcal{F}) + 2\sqrt{\frac{\log(1/\delta)}{2n}} .$$

Furthermore, if ℓ_α is α-CC, then for the choice $\alpha = \delta_D^*$, there exists a nondecreasing function $\zeta_{\ell,\alpha}$ with $\zeta_{\ell,\alpha}(0) = 0$ such that,

$$\mathcal{U}^* - \mathcal{U}(\hat{f};D) \le \zeta_{\ell,\alpha}\left(\min_{f\in\mathcal{F}} R_{\ell_\alpha,D}(f) - \min_f R_{\ell_\alpha,D}(f) + 4L_\rho\mathfrak{R}(\mathcal{F}) + 2\sqrt{\frac{\log(1/\delta)}{2n}}\right) .$$

Proof Let $f^\star$ be the minimizer of $R_{\ell_\alpha,D}(\cdot)$ over $\mathcal{F}$. We have

$$\begin{aligned}
&R_{\ell_\alpha,D}(\hat{f}) - R_{\ell_\alpha,D}(f^\star) \\
&= R_{\tilde{\ell}_\alpha,D_\rho}(\hat{f}) - R_{\tilde{\ell}_\alpha,D_\rho}(f^\star) \\
&= \widehat{R}_{\tilde{\ell}_\alpha}(\hat{f}) - \widehat{R}_{\tilde{\ell}_\alpha}(f^\star) + (R_{\tilde{\ell}_\alpha,D_\rho}(\hat{f}) - \widehat{R}_{\tilde{\ell}_\alpha}(\hat{f})) \\
&\quad + (\widehat{R}_{\tilde{\ell}_\alpha}(f^\star) - R_{\tilde{\ell}_\alpha,D_\rho}(f^\star)) \\
&\le 0 + 2\max_{f\in\mathcal{F}} |\widehat{R}_{\tilde{\ell}_\alpha}(f) - R_{\tilde{\ell}_\alpha,D_\rho}(f)| .
\end{aligned}$$

We can now apply Lemma 8 to control the last quantity above, and thus obtain the first statement of the theorem. Now, if ℓ_α is α-CC, then for $\alpha = \delta_D^*$, we know from Lemma 5 there exists a convex, invertible, nondecreasing transformation ψ_ℓ with $\psi_{\ell,\alpha}(0) = 0$ such that,

$$\psi_{\ell,\alpha}(R_\alpha(f) - R_\alpha^*) \le R_{\ell_\alpha,D}(f) - \min_f R_{\ell_\alpha,D}(f)$$

Subtracting $\min_f R_{\ell_\alpha,D}(f)$ off either sides of the first inequality in the theorem statement, and realizing that $\psi_{\ell,\alpha}^{-1}$ is nondecreasing as well, with $\psi_{\ell,\alpha}^{-1}(0) = 0$, we get:

$$R_\alpha(\hat{f}) - R_\alpha^* \le \psi_\ell^{-1}\left(\min_{f\in\mathcal{F}} R_{\ell,D}(f) - \min_f R_{\ell,D}(f) + 4L_\rho\mathfrak{R}(\mathcal{F}) + 2\sqrt{\frac{\log(1/\delta)}{2n}}\right) .$$

Finally we can use Lemma 3 to bound $\mathcal{U}^* - \mathcal{U}(\hat{f};D)$, by setting $\zeta_{\ell,\alpha} = (a_{11}+a_{00}-a_{01}-a_{10})\psi_{\ell,\alpha}^{-1}$. ■

The term on the right hand side involves both approximation error (that is small if $\mathcal{F}$ is large) and estimation error (that is small if $\mathcal{F}$ is small). However, by appropriately increasing the richness of the class $\mathcal{F}$ with sample size, we can ensure that the utility of $\hat{f}$ approaches the optimal utility under the true distribution. This is despite the fact that the method of unbiased estimators computes the empirical minimizer $\hat{f}$ on a sample from the noisy distribution. Getting the optimal empirical minimizer $\hat{f}$ is efficient if $\tilde{\ell}_\alpha$, or rather $\tilde{\ell}$, is convex. Next, we address the issue of convexity of $\tilde{\ell}$.

4.1 Convex losses and their estimators

Note that the loss $\tilde{\ell}$ may not be convex even if we start with a convex ℓ. An example is provided by the familiar hinge loss $\ell_{\text{hin}}(t,y) = [1-yt]_+$. Stempfel and Ralaivola (2009) showed that $\tilde{\ell}_{\text{hin}}$ is not convex in general (of course, when $\rho_{+1} = \rho_{-1} = 0$, it is convex). Below we provide a simple condition to ensure convexity of $\tilde{\ell}$.

Lemma 10 *Suppose $\ell(t,y)$ is convex and twice differentiable almost everywhere in t (for every y) and also satisfies the symmetry property*

$$\forall t \in \mathbb{R},\ \ell''(t,y) = \ell''(t,-y)\ .$$

Then $\tilde{\ell}(t,y)$ is also convex in t.

Proof Let us compute $\tilde{\ell}''(t,y)$ (recall that differentiation is w.r.t. t) and show that it is non-negative under the symmetry condition $\ell''(t,y) = \ell''(t,-y)$. We have

$$\begin{aligned}
\tilde{\ell}''(t,y) &= \frac{(1-\rho_{-y})\ell''(t,y) - \rho_y \ell''(t,-y)}{1-\rho_{+1}-\rho_{-1}} \\
&= \frac{(1-\rho_{-y})\ell''(t,y) - \rho_y \ell''(t,y)}{1-\rho_{+1}-\rho_{-1}} \\
&= \frac{(1-\rho_{-y}-\rho_y)\ell''(t,y)}{1-\rho_{+1}-\rho_{-1}} \\
&= \ell''(t,y) \geq 0\ ,
\end{aligned}$$

since ℓ is convex in t. ■

Examples satisfying the conditions of the lemma above are the squared loss $\ell_{\text{sq}}(t,y) = (t-y)^2$, the logistic loss $\ell_{\log}(t,y) = \log(1+\exp(-ty))$ and the Huber loss:

$$\ell_{\text{Hub}}(t,y) = \begin{cases} -4yt & \text{if } yt < -1 \\ (t-y)^2 & \text{if } -1 \leq yt \leq 1 \\ 0 & \text{if } yt > 1 \end{cases}$$

Consider the case where $\tilde{\ell}$ turns out to be non-convex when ℓ is convex, as in $\tilde{\ell}_{\text{hin}}$. In the online learning setting (where the adversary chooses a sequence of examples, and the prediction of a learner at round i is based on the history of $i-1$ examples with independently

flipped labels) which we will discuss shortly, we would use a stochastic mirror descent type algorithm (Nemirovski et al., 2009) to arrive at risk bounds similar to Theorem 9. Then, we only need the expected loss to be convex and therefore ℓ_{hin} does not present a problem. At first blush, it may appear that we do not have much hope of obtaining $\hat{f}$ in the iid setting efficiently. However, Lemma 8 provides a clue.

We will now focus on the function class $\mathcal{W}$ of hyperplanes. Even though $\widehat{R}_{\tilde{\ell}}(\mathbf{w})$ is non-convex, it is uniformly close to $R_{\tilde{\ell},D_\rho}(\mathbf{w})$. Since $R_{\tilde{\ell},D_\rho}(\mathbf{w}) = R_{\ell,D}(\mathbf{w})$, this shows that $\widehat{R}_{\tilde{\ell}}(\mathbf{w})$ is uniformly close to a convex function over $\mathbf{w} \in \mathcal{W}$. The following result shows that we can therefore approximately minimize $F(\mathbf{w}) = \widehat{R}_{\tilde{\ell}}(\mathbf{w})$ by minimizing the biconjugate $F^{\star\star}$. Recall that the (Fenchel) biconjugate $F^{\star\star}$ is the largest convex function that minorizes F.

Lemma 11 *Let $F : \mathcal{W} \to \mathbb{R}$ be a non-convex function defined on function class $\mathcal{W}$ such it is ε-close to a convex function $G : \mathcal{W} \to \mathbb{R}$:*

$$\forall \mathbf{w} \in \mathcal{W},\ |F(\mathbf{w}) - G(\mathbf{w})| \leq \varepsilon$$

Then any minimizer of $F^{\star\star}$ is a 2ε-approximate (global) minimizer of F.

Proof Since $F \geq G - \varepsilon$ and $F^{\star\star}$ is the largest convex function that minorizes F, we must have $F^{\star\star} \geq G - \varepsilon$. This means that $F^{\star\star} + 2\varepsilon \geq G + \varepsilon \geq F$. Thus, F is sandwiched between $F^{\star\star} + 2\varepsilon$ and $F^{\star\star}$. The lemma follows directly from this. ■

Now, the following theorem establishes bounds for the case when $\tilde{\ell}$ is non-convex, via the solution obtained by minimizing the convex function $F^{\star\star}$.

Theorem 12 *Let ℓ be a loss, such as the hinge loss, for which $\tilde{\ell}$ is non-convex. Let $\mathcal{W} = \{\mathbf{w} : \|\mathbf{w}_2\| \leq W_2\}$, let $\|X_i\|_2 \leq X_2$ almost surely, and let $\hat{\mathbf{w}}_{\text{approx}}$ be any (exact) minimizer of the convex problem*

$$\min_{\mathbf{w}\in\mathcal{W}}\ F^{\star\star}(\mathbf{w})\ ,$$

where $F^{\star\star}(\mathbf{w})$ is the (Fenchel) biconjugate of the function $F(\mathbf{w}) = \widehat{R}_{\tilde{\ell}_\alpha}(\mathbf{w})$, where $\alpha = \delta_D^*$. Then, with probability at least $1-\delta$, $\hat{\mathbf{w}}_{\text{approx}}$ is a 2ε-minimizer of $\widehat{R}_{\tilde{\ell}_\alpha}(\cdot)$ where

$$\varepsilon = \frac{2L_\rho X_2 W_2}{\sqrt{n}} + \sqrt{\frac{\log(1/\delta)}{2n}}\ .$$

Therefore, with probability at least $1-\delta$,

$$R_{\ell_\alpha,D}(\hat{\mathbf{w}}_{\text{approx}}) \leq \min_{\mathbf{w}\in\mathcal{W}} R_{\ell_\alpha,D}(\mathbf{w}) + 4\varepsilon\ .$$

Proof The first part of the theorem follows by combining Lemma 8 and Lemma 11, using the fact that if $\|\mathbf{w}\|_2 \leq W_2$ for any $\mathbf{w}$ and $\|X_i\|_2 \leq X_2$ then, $\mathfrak{R}(\mathcal{W}) \leq W_2 X_2/\sqrt{n}$. Note that Theorem 9 is true also for 2ε-minimizers of the empirical risk $\widehat{R}_{\tilde{\ell}_\alpha}$ provided we add 2ε to the right hand side. ■

Numerical or symbolic computation of the biconjugate of a multidimensional function is difficult, in general, but can be done in special cases. It will be interesting to see if techniques from Computational Convex Analysis (Lucet, 2010) can be used to efficiently compute the biconjugate above.

4.2 Online learning setting

Consider the setting where an adversary chooses a sequence $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ of examples. At time i, the learner has to make a prediction based on $(\mathbf{x}_1, \tilde{y}_1), \ldots, (\mathbf{x}_{i-1}, \tilde{y}_{i-1})$ and $\mathbf{x}_i$. But the learner's cumulative loss as well as that of the best fixed predictor in hindsight are both computed using the true labels y_i. Note that if $\ell(t, y)$ is convex in t (for every y), and for a given $\alpha \in (0, 1)$ we choose $\lambda_1 \in \partial \ell_\alpha(t, y)$ and $\lambda_2 \in \partial \ell_\alpha(t, -y)$, (where $\partial \ell_\alpha$ is the subdifferential w.r.t. t) we have

$$\mathbb{E}_{\tilde{y}} \left[g(t, \tilde{y}) \right] \in \partial \ell_\alpha(t, y) \tag{4}$$

where

$$g(t, y) = \frac{(1 - \rho_{-y})\lambda_1 - \rho_y \lambda_2}{1 - \rho_{+1} - \rho_{-1}} \tag{5}$$

We show that Algorithm 1 indeed satisfies low regret (in expectation) on the original sequence chosen by the adversary even though it only receives noisy versions of the labels. We fix the function class to be the set $\mathcal{W}$ of bounded-norm hyperplanes.

Algorithm 1: Online learning using unbiased gradients

Choose learning rate $\gamma > 0$
$\mathcal{W} = \{\mathbf{w} : \|\mathbf{w}\|_2 \leq W_2\}$
$\Pi_{\mathcal{W}}(\cdot)$ = Euclidean projection onto $\mathcal{W}$
Initialize $\mathbf{w}_0 \leftarrow \mathbf{0}$
for $i = 1$ to n **do**
  Receive $\mathbf{x}_i \in \mathbb{R}^d$
  Predict $\langle \mathbf{w}_{i-1}, \mathbf{x}_i \rangle$
  Receive noisy label $\tilde{y}_i$
  Update $\mathbf{w}_i \leftarrow \Pi_{\mathcal{W}} \left(\mathbf{w}_{i-1} - \gamma g(\langle \mathbf{w}_{i-1}, \mathbf{x}_i \rangle, \tilde{y}_i) \mathbf{x}_i \right)$ where $g(\cdot, \cdot)$ is defined in (5)
end for

Theorem 13 *Let $\ell(t, y)$ be convex and L-Lipschitz in t (for every y). Fix an arbitrary sequence $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, and $\alpha \in (0, 1)$. If Algorithm 1 is run on noisy data set $(\mathbf{x}_1, \tilde{y}_1), \ldots, (\mathbf{x}_n, \tilde{y}_n)$ with learning rate $\gamma = W_2/(X_2 L_\rho \sqrt{n})$ where $\tilde{y}_i$ is noisy version of y_i with noise rates ρ_{+1}, ρ_{-1}, then we have*

$$\mathbb{E}_{\tilde{y}_{1:n}} \left[\sum_{i=1}^{n} \ell_\alpha(\langle \mathbf{w}_{i-1}, \mathbf{x}_i \rangle, y_i) \right] - \min_{\|\mathbf{w}\|_2 \leq W_2} \sum_{i=1}^{n} \ell_\alpha(\langle \mathbf{w}, \mathbf{x}_i \rangle, y_i) \leq L_\rho X_2 W_2 \sqrt{n} \ ,$$

where $L_\rho := (1 + |\rho_{+1} - \rho_{-1}|)L/(1 - \rho_{+1} - \rho_{-1})$ and it is assumed that $\|\mathbf{x}_i\| \leq X_2$ for all $i \in [n]$.

Proof Let us use the abbreviation g_i for $g(\langle \mathbf{w}_{i-1}, \mathbf{x}_i \rangle, \tilde{y}_i)\mathbf{x}_i$ so that the update in Algorithm 1 becomes $\mathbf{w}_i \leftarrow \Pi_{\mathcal{W}} \left(\mathbf{w}_{i-1} - \gamma g_i \right)$. It is well known (Zinkevich, 2003) that, for any $\mathbf{w}$,

$$\sum_{i=1}^{n} \langle g_i, \mathbf{w}_{i-1} - \mathbf{w} \rangle \leq \frac{\gamma}{2} \sum_{i=1}^{n} \|g_i\|^2 + \frac{\|\mathbf{w}\|^2}{2\gamma} \ . \tag{6}$$

Since ℓ is L-Lipschitz, the λ_1, λ_2 appearing in the definition (5) of $g(\cdot, \cdot)$ satisfy $|\lambda_1|, |\lambda_2| \leq L$. This implies $|g(t, y)| \leq (1 + |\rho_{+1} - \rho_{-1}|)L/(1 - \rho_{+1} - \rho_{-1}) = L_\rho$ and hence $\|g_i\| \leq L_\rho X_2$. Thus, we have, for any $\mathbf{w}$ with $\|\mathbf{w}\| \leq W_2$, $\sum_{i=1}^{n} \langle g_i, \mathbf{w}_{i-1} - \mathbf{w} \rangle \leq \frac{\gamma L_\rho^2 X_2^2 n}{2} + \frac{W_2^2}{2\gamma}$. Choosing $\gamma = (W_2/L_\rho X_2)\frac{1}{\sqrt{n}}$, we get $\sum_{i=1}^{n} \langle g_i, \mathbf{w}_{i-1} - \mathbf{w} \rangle \leq L_\rho X_2 W_2 \sqrt{n}$. Note that $\mathbf{w}_{i-1}$ only depends on $\tilde{y}_{1:i-1}$. Hence

$$\mathbb{E}_{\tilde{y}_i} \left[\langle g_i, \mathbf{w}_{i-1} - \mathbf{w} \rangle \mid \tilde{y}_{1:i-1} \right] = \left\langle \mathbb{E}_{\tilde{y}_i} \left[g_i \mid \tilde{y}_{1:i-1} \right], \mathbf{w}_{i-1} - \mathbf{w} \right\rangle \geq \ell_\alpha(\langle \mathbf{w}_{i-1}, \mathbf{x}_i \rangle, y_i) - \ell_\alpha(\langle \mathbf{w}, \mathbf{x}_i \rangle, y_i)$$

because $\mathbb{E}_{\tilde{y}_i} \left[g_i \mid \tilde{y}_{1:i-1} \right] \in \partial_{\mathbf{w}=\mathbf{w}_{i-1}} \ell_\alpha(\langle \mathbf{w}, \mathbf{x}_i \rangle, y_i)$ by (4) and the chain rule for differentiation, and $\ell_\alpha(\langle \mathbf{w}, \mathbf{x}_i \rangle, y_i)$ is convex in $\mathbf{w}$. Thus, for any $\mathbf{w}$ with $\|\mathbf{w}\|_2 \leq W_2$,

$$\mathbb{E}_{\tilde{y}_{1:n}} \left[\sum_{i=1}^{n} \ell_\alpha(\langle \mathbf{w}_{i-1}, \mathbf{x}_i \rangle, y_i) \right] - \sum_{i=1}^{n} \ell_\alpha(\langle \mathbf{w}, \mathbf{x}_i \rangle, y_i) \leq L_\rho X_2 W_2 \sqrt{n}.$$

Since the above inequality is true for any $\mathbf{w}$ with $\|\mathbf{w}\|_2 \leq 1$, the statement of the theorem follows. ■

5. Approach of Surrogates for Weighted 0-1 Loss

The second approach is based on directly obtaining weighted surrogates for $\mathcal{U}$. We develop the method of "label-dependent" costs from two key observations. First, the Bayes classifier under noisy distribution, denoted by $\tilde{f}^*$, simply uses a threshold that, in general, is different from that under clean distribution. Second, $\tilde{f}^*$ is the minimizer of a certain weighted 0-1 loss under the noisy distribution. The framework we develop here generalizes known results for the uniform noise rate setting $\rho_{+1} = \rho_{-1}$ and offers a more fundamental insight into the problem.

From Lemma 2, we know that the optimal Bayes classifier $\tilde{f}^*$ under D_ρ thresholds $\tilde{\eta}(X) = P(\tilde{Y} = 1|X)$ at a certain $\tilde{\delta}^*$. Now, noting that:

$$\tilde{\eta}(x) = (1 - \rho_{+1})\eta(x) + \rho_{-1}(1 - \eta(x)) = (1 - \rho_{+1} - \rho_{-1})\eta(x) + \rho_{-1},$$

we see that $\tilde{f}^*$ can be written as:

$$\tilde{f}^*(x) = \text{sign}\left(\eta(x) - \frac{\tilde{\delta}^* - \rho_{-1}}{1 - \rho_{+1} - \rho_{-1}} \right). \tag{7}$$

Not surprisingly, this optimal threshold simplifies for cost-sensitive performance measures. In particular, as shown in the following corollary, the optimal threshold for the AM measure does *not* change under the noisy distribution.

Corollary 14 *1. For the Accuracy measure:*

$$\arg\max_f \mathcal{U}_{Acc}(f; D_\rho) = \arg\min_f R_{D_\rho}(f) = \text{sign}\left(\eta(x) - \frac{1/2 - \rho_{-1}}{1 - \rho_{+1} - \rho_{-1}} \right).$$

2. For the AM Measure:

$$\arg\max_f \mathcal{U}_{AM}(f;D) = \arg\max_f \mathcal{U}_{AM}(f;D_\rho) = \text{sign}(\eta(x) - \pi).$$

Proof

1. For the 0-1 loss, $\delta^*_D = \tilde{\delta}^* = 1/2$ and the result is immediate.
2. We know $\delta^*_D = \pi$ from Lemma 2. Also, $\tilde{\delta}^* = P(\tilde{Y} = 1) = (1 - \rho_{-1} - \rho_{+1})\pi + \rho_{-1}$. Substituting in (7), we observe that the threshold remains π.

■

Interestingly, this *noisy* Bayes classifier can also be obtained as the minimizer of a weighted 0-1 loss; which as we will show, allows us to "correct" for the threshold under the noisy distribution. Let us first introduce the notion of label-dependent costs for binary classification. We can write the 0-1 loss as a label-dependent loss as follows:

$$1_{\{\text{sign}(f(X))\neq Y\}} = 1_{\{Y=1\}}1_{\{f(X)\leq 0\}} + 1_{\{Y=-1\}}1_{\{f(X)>0\}}$$

Clearly, the classical 0-1 loss is *unweighted*. Consider the α-weighted 0-1 loss (which is a special case of the weighted loss (2)):

$$U_\alpha(t,y) = (1-\alpha)1_{\{y=1\}}1_{\{t\leq 0\}} + \alpha 1_{\{y=-1\}}1_{\{t>0\}},$$

where $\alpha \in (0,1)$. In fact we see that minimization w.r.t. the 0-1 loss is equivalent to that w.r.t. $U_{1/2}(f(X), Y)$. It is not a coincidence that Bayes optimal f^* has a threshold $1/2$. The following lemma (Scott, 2012) shows that in fact for any α-weighted 0-1 loss, the minimizer thresholds $\eta(x)$ at α.

Lemma 15 (α-weighted Bayes optimal (Scott, 2012)) *For $\alpha \in (0,1)$,*

$$f^*_\alpha := \arg\min_f R_\alpha(f) = \text{sign}(\eta(x) - \alpha)\ .$$

At this juncture, we are interested in the following question: For a given δ, does there exist an $\alpha \in (0,1)$ such that the minimizer of U_α-risk under noisy distribution D_ρ has the same sign as that of the Bayes optimal f^*_δ? We now present our second main result in the following theorem that makes a stronger statement — the U_α-risk under noisy distribution D_ρ is linearly related to U_δ-risk under the clean distribution D. The corollary of the theorem answers the question in the affirmative.

Theorem 16 *For any given $\delta \in (0,1)$, for the choices,*

$$\alpha^* = \rho_{-1} + (1 - \rho_{+1} - \rho_{-1})\delta \text{ and } A_\rho = 1 - \rho_{+1} - \rho_{-1},$$

there exists a constant B_X that is independent of f such that, for all functions f,

$$R_{\alpha^*,D_\rho}(f) = A_\rho R_{\delta,D}(f) + B_X.$$

Proof For simplicity, let us think of f as $\{\pm 1\}$-valued. We have,

$$C_{\delta,D}(f) = \mathbb{E}_Y\left[(1-\delta)1_{\{Y=1\}}1_{\{f(X)\neq 1\}} + \delta 1_{\{Y=-1\}}1_{\{f(X)\neq -1\}}\right]$$

and

$$C_{\alpha,D_\rho}(f) = \mathbb{E}_{\tilde{Y}}\left[(1-\alpha)1_{\{\tilde{Y}=1\}}1_{\{f(X)\neq 1\}} + \alpha 1_{\{\tilde{Y}=-1\}}1_{\{f(X)\neq -1\}}\right].$$

Note that $R_{\delta,D}(f) = \mathbb{E}_X\left[C_{\delta,D}(f)\right]$, and $R_{\alpha,D_\rho}(f) = \mathbb{E}_X\left[C_{\alpha,D_\rho}(f)\right]$. Also note that $C_{\delta,D}(f) = (1-\delta)\eta(X)$ if $f(X) = -1$, and $C_{\delta,D}(f) = \delta(1-\eta(X))$ otherwise.
Similarly, $C_{\alpha,D_\rho}(f) = (1-\alpha)\tilde{\eta}(X)$ if $f(X) = -1$ and $C_{\alpha,D_\rho}(f) = \alpha(1-\tilde{\eta}(X))$ otherwise.
We want to find A and B such that the following equations hold simultaneously:

$$(1-\alpha)\tilde{\eta}(X) = A(1-\delta)\eta(X) + B$$
$$\alpha(1-\tilde{\eta}(X)) = A\delta(1-\eta(X)) + B$$

Using the relation between $\eta(X)$ and $\tilde{\eta}(X)$ and solving for A we get,

$$A = \frac{(1-\rho_{+1}-\rho_{-1})\eta(X) + \rho_{-1} - \alpha}{\eta(X) - \delta}.$$

Choosing $\alpha = \alpha^* = \rho_{-1} + (1-\rho_{+1}-\rho_{-1})\delta$, and simplifying, we get a constant A that depends only on the noise rates:

$$A = A_\rho = 1 - \rho_{+1} - \rho_{-1}.$$

Consequently,

$$B = \rho_{-1}(1-\alpha^*) + (\delta - \alpha^*)(1-\rho_{+1}-\rho_{-1})\eta(X).$$

Taking expectation with respect to X, we conclude:

$$R_{\alpha^*,D_\rho}(f) = A_\rho R_{\delta,D}(f) + B_X,$$

where $B_X = \mathbb{E}_X[B]$. ■

Corollary 17 *Let $\alpha^* = \rho_{-1} + (1-\rho_{+1}-\rho_{-1})\delta^*_D$. The $\alpha^\star$-weighted Bayes optimal classifier under noisy distribution coincides with that of $\mathcal{U}$ measure under clean distribution:*

$$\operatorname*{argmin}_f R_{\alpha^*,D_\rho}(f) = \operatorname*{argmin}_f R_{\delta^*_D,D}(f) = \operatorname*{argmin}_f \mathcal{U}(f;D).$$

We are now ready to state our next main result — a certain weighted ERM is consistent: i.e. the "true" performance of the empirical minimizer w.r.t. the noisy distribution converges to the optimal performance $\mathcal{U}^*$ at a steady rate. The resulting bound has a striking resemblance to that of our first result in Theorem 9. The proof technique is similar to that of Theorem 9, and crucially relies on using the relationship established in Theorem 16.

Theorem 18 *Given a convex loss function $\ell : \mathbb{R} \to [0, \infty)$ with Lipschitz constant L such that it is classification-calibrated (i.e. $\ell'(0) < 0$), consider the empirical risk minimization problem with noisy labels:*

$$\hat{f}_\alpha = \operatorname*{argmin}_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell_\alpha(f(X_i), \tilde{Y}_i). \tag{8}$$

where ℓ_α is defined as in (2). Then, for the choice of α^* in Corollary 17, there exists a nondecreasing function $\zeta_{\ell_{\alpha^\star}}$ with $\zeta_{\ell_{\alpha^\star}}(0) = 0$, such that the following bound holds with probability at least $1 - \delta$:

$$\mathcal{U}^* - \mathcal{U}(\hat{f}_{\alpha^*}; D) \leq \frac{1}{A_\rho} \zeta_{\ell_{\alpha^\star}} \left(\min_{f \in \mathcal{F}} R_{\alpha^*, D_\rho}(f) - \min_f R_{\alpha^*, D_\rho}(f) + 4L\mathfrak{R}(\mathcal{F}) + 2\sqrt{\frac{\log(1/\delta)}{2n}} \right),$$

where $A_\rho = 1 - \rho_{+1} - \rho_{-1}$.

Proof From Corollary 4.1 of Scott (2012), we can infer that ℓ_α is α-CC for given $\alpha \in (0, 1)$, as ℓ is convex, classification-calibrated and $\ell'(0) < 0$. Then, from Theorem 3.1 of Scott (2012), there exists an *invertible*, *non-decreasing* convex transformation ψ_{ℓ_α} with $\psi_{\ell_\alpha}(0) = 0$ such that, for any f and any distribution D,

$$\psi_{\ell_\alpha}(R_{\alpha,D}(f) - \min_f R_{\alpha,D}(f)) \leq R_{\ell_\alpha,D}(f) - \min_f R_{\ell_\alpha,D}(f).$$

Fix distribution to be D_ρ, and let $f = \hat{f}_\alpha$. The RHS of the above inequality can then be controlled similarly as in the proof of Theorem 9. It is easy to see that the Lipschitz constant of ℓ_α is same as that of ℓ, denoted L. With probability at least $1 - \delta$:

$$R_{\ell_\alpha,D_\rho}(\hat{f}_\alpha) - \min_{f \in \mathcal{F}} R_{\ell_\alpha,D_\rho}(f) \leq 4L\mathfrak{R}(\mathcal{F}) + 2\sqrt{\frac{\log(1/\delta)}{2n}}.$$

Now consider $R_{\alpha,D_\rho}(f) - \min_f R_{\alpha,D_\rho}(f)$. Using the linear relationship between R_{α,D_ρ} and $R_{\delta^*_D,D}$ at α^* (Theorem 16), we get $R_{\alpha^*,D_\rho}(f) - \min_f R_{\alpha^*,D_\rho}(f) = A_\rho(R_{\delta^*_D,D}(f) - R^*_{\delta^*_D})$. B_X vanishes because it is constant for the distribution D_ρ. Note that $\psi^{-1}_{\ell_{\alpha^*}}$ is nondecreasing as well and $\psi^{-1}_{\ell_{\alpha^*}}(0) = 0$. Subtracting $\min_f R_{\alpha^*,D_\rho}(f)$ from both sides of the second inequality above, we get: With probability at least $1 - \delta$,

$$R_{\delta^*_D,D}(\hat{f}_{\alpha^*}) - R^*_{\delta^*_D} \leq A_\rho^{-1} \psi^{-1}_{\ell_{\alpha^\star}} \left(\min_{f \in \mathcal{F}} R_{\alpha^*,D_\rho}(f) - \min_f R_{\alpha^*,D_\rho}(f) + 4L\mathfrak{R}(\mathcal{F}) + 2\sqrt{\frac{\log(1/\delta)}{2n}} \right).$$

Invoking Lemma 3 and setting $\zeta_{\ell_{\alpha^\star}} = (a_{11} + a_{00} - a_{10} - a_{01})\psi^{-1}_{\ell_{\alpha^\star}}$, the proof is complete. ■

6. Experiments

In our first set of experiments, we demonstrate the robustness of the proposed algorithms to increasing rates of label noise on synthetic and real-world data sets. In our second set of experiments, we also conduct a comparison of the performance of our two proposed methods with state-of-the-art methods for dealing with random label noise. In our experiments, we use the two utility measures listed in Proposition 1, i.e. $\mathcal{U}_{Acc}$ and $\mathcal{U}_{AM}$; note that the utility measures are computed with respect to the clean distribution. For given noise rates ρ_{+1} and ρ_{-1}, labels are flipped accordingly. To account for randomness in the flips to simulate a given noise rate, we repeat each experiment 3 times, with independent corruptions of the data set for same setting of ρ_{+1} and ρ_{-1}, and present the mean accuracy over the trials. Specifically, we divide each data set randomly into three training and test sets, and compute average utility over 3 train-test splits. We use cross-validation to tune parameters specific to the algorithms. Note that we perform cross-validation on a separate validation set with *noisy* labels. In our final set of experiments, we address a practical question of specifying true noise rates to the algorithms, and study how misspecification of noise rates affects the performance of the algorithms.

Proposed methods. For evaluation, we choose the following representative algorithms based on each of the two proposed methods: For the method in Section 4, we use unbiased estimator of the logistic loss. Here, the resulting ERM, i.e. (3) with $\ell_{\log}$, is solved using a gradient descent procedure. We refer to this as $\tilde{\ell}_{\log}$ for ease in the remainder of the section. For the method in Section 5 we use the widely-used C-SVM (Liu et al., 2003; Mordelet and Vert, 2014) method as well as weighted logistic regression, wherein we apply different costs on positive and negative examples in the respective loss functions. We use the `libsvm` library to solve the resulting ERM problems, i.e. (8) with ℓ_{hin} or $\ell_{\log}$ respectively. In all the cases, we tune the parameters α, ρ_{+1} and ρ_{-1} by cross-validation (on noisy validation set).

6.1 Synthetic data

First, we use the synthetic 2D linearly separable data set shown in Figure 1(a). We observe from experiments that our methods achieve over 90% accuracy even when $\rho_{+1} = \rho_{-1} = 0.4$. Figure 1 shows the performance of $\tilde{\ell}_{\log}$ on the data set for different noise rates. Next, we use a 2D UCI benchmark non-separable data set ('banana'). The data set and classification results using C-SVM (which corresponds to vanilla SVM for uniform noise rates, $\alpha^* = 1/2$) are shown in Figure 2. The results for higher noise rates are impressive as observed from Figures 2(d) and 2(e). The 'banana' data set has been used in previous research on classification with noisy labels. In particular, the Random Projection classifier (Stempfel and Ralaivola, 2007) that learns a kernel perceptron in the presence of noisy labels achieves about 84% accuracy at $\rho_{+1} = \rho_{-1} = 0.3$ as observed from our experiments (as well as shown by Stempfel and Ralaivola, 2007), and the random hyperplane sampling method (Stempfel et al., 2007) gets about the same accuracy at $(\rho_{+1}, \rho_{-1}) = (0.2, 0.4)$ (as reported by Stempfel et al., 2007). Contrast these with C-SVM that achieves about 90% accuracy at $\rho_{+1} = \rho_{-1} = 0.2$ and over 88% accuracy at $\rho_{+1} = \rho_{-1} = 0.4$.

6.2 Comparison with state-of-the-art methods on UCI benchmark data

We next compare our methods with three state-of-the-art methods for dealing with random classification noise: Random Projection (RP) classifier (Stempfel and Ralaivola, 2007), NHERD (Crammer and Lee, 2010) (*project* and *exact* variants, which were shown to be the

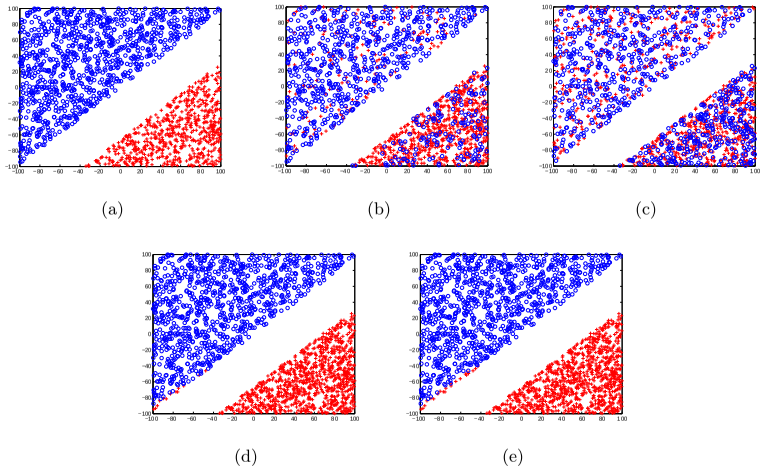


Figure 1: Classification of linearly separable synthetic data set using $\tilde{\ell}_{\log}$. The noise-free data is shown in the leftmost panel. Plots (b) and (c) show training data corrupted with noise rates ($\rho_{+1} = \rho_{-1} = \rho$) 0.2 and 0.4 respectively. Plots (d) and (e) show the corresponding classification results. The algorithm achieves 98.5% accuracy even at 0.4 noise rate per class. (Best viewed in color).

best performing variants from among the NHERD family of methods proposed by Crammer et al. 2006, 2009; Dredze et al. 2008), and perceptron algorithm with margin (PAM) which was shown to be robust to label noise by Khardon and Wachman (2007). We use seven standard UCI classification data sets listed in Table 1; here, data sets 1 through 6 are preprocessed and made available by Gunnar Rätsch.[1]

Using linear kernel. Results for the accuracy measure, for different settings of noise rates, using linear kernel in the compared methods, are shown in Table 2. C-SVM is competitive in 5 out of 7 data sets (Breast cancer, Thyroid, German, Image and Spambase), while relatively poorer in the other two. Note that in many cases, especially when $\rho_{+1} = \rho_{-1}$, the standard SVM (i.e. where positive and negative examples are weighted equally) as well as C-SVM (where α parameter that controls the relative weighting is tuned) yield the same accuracy, indicating that the cross-validation effectively selects equal weights; recall that the theory indeed suggests when the noise rates are equal, the optimal choice of weights are equal, i.e. $\alpha = 1/2$ (see Section 5). The corresponding results for the AM measure, for different settings of noise rates, are shown in Table 5. We find that C-SVM is competitive in three data sets, and NHERD is competitive in most of the data sets. Also observe that

1. `http://theoval.cmp.uea.ac.uk/matlab`

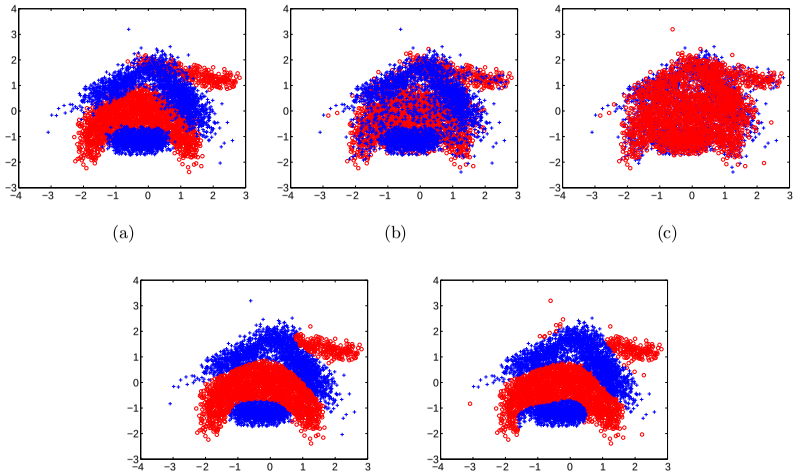


Figure 2: Classification of 'banana' data set using C-SVM. The noise-free data is shown in (a). Plots (b) and (c) show training data corrupted with noise rates ($\rho_{+1} = \rho_{-1} = \rho$) 0.2 and 0.4 respectively. Note that for $\rho_{+1} = \rho_{-1}$, $\alpha^* = 1/2$ (i.e. C-SVM reduces to regular SVM). Plots (d) and (e) show the corresponding classification results (Accuracies are 90.6% and 88.5% respectively). Even when 40% of the labels are corrupted ($\rho_{+1} = \rho_{-1} = 0.4$), the algorithm recovers the class structures as observed from plot (e). Note that the accuracy of the method at $\rho = 0$ is 90.8%.

at high noise rates AM is a more reliable measure of performance — in case of the four data sets Breast cancer, Diabetes, Thyroid and German which have class imbalance, the classifier optimized for the accuracy measure (Table 2) tends to bias its predictions towards the majority class (suggested by accuracy values matching the class imbalance ratio) but the achieved AM values are low.

We present the results for logistic loss based methods, using linear kernel, in Tables 3 and 6. As in the case of SVM based methods, we find, when $\rho_{+1} = \rho_{-1}$, the standard logistic regression (i.e. where positive and negative examples are weighted equally) as well as weighted logistic regression in the third column (where α parameter that controls the relative weighting is tuned) yield the same accuracy, indicating that the cross-validation effectively selects equal weights. In terms of accuracy, we find that $\tilde{\ell}_{\log}$ (in the second column) is competitive in all the data sets, wherease in terms of AM measure (see Table 6), (weighted) logistic regression performs the best more often.

Data set	Dim	Num. Positives	Num. Negatives
Breast cancer	9	77	186
Diabetes	8	268	500
Thyroid	5	65	150
German	20	300	700
Heart	13	120	150
Image	18	1188	898
Spambase	57	1813	2788

Table 1: UCI data sets used in experiments.

Data set	Noise rates	SVM	C-SVM	PAM	NHERD	RP
Breast cancer	$\rho_{+1} = \rho_{-1} = 0.2$	**70.25**	**70.25**	68.42	64.90	38.95
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	**71.53**	**71.53**	69.97	65.68	66.93
	$\rho_{+1} = \rho_{-1} = 0.4$	69.49	**70.77**	44.25	56.50	55.95
Diabetes	$\rho_{+1} = \rho_{-1} = 0.2$	**75.35**	**75.35**	62.76	73.18	71.09
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	**75.52**	**75.52**	57.64	**74.74**	73.26
	$\rho_{+1} = \rho_{-1} = 0.4$	68.84	68.84	51.52	**71.09**	68.58
Thyroid	$\rho_{+1} = \rho_{-1} = 0.2$	**81.94**	**81.94**	63.58	78.49	78.89
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	**86.63**	**86.63**	45.48	**87.78**	78.89
	$\rho_{+1} = \rho_{-1} = 0.4$	76.63	76.63	70.98	**85.95**	70.69
German	$\rho_{+1} = \rho_{-1} = 0.2$	**72.87**	**72.87**	55.47	67.80	67.72
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	**69.46**	**69.46**	50.02	67.80	63.93
	$\rho_{+1} = \rho_{-1} = 0.4$	**62.20**	56.60	41.33	54.80	56.27
Heart	$\rho_{+1} = \rho_{-1} = 0.2$	**82.96**	**82.96**	73.09	**82.96**	76.05
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	77.53	77.53	71.60	**81.48**	78.77
	$\rho_{+1} = \rho_{-1} = 0.4$	**78.27**	72.35	61.23	52.59	72.59
Image	$\rho_{+1} = \rho_{-1} = 0.2$	**79.55**	**79.55**	70.66	77.76	**80.51**
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	74.05	74.05	68.94	**79.39**	**81.02**
	$\rho_{+1} = \rho_{-1} = 0.4$	**73.73**	**73.73**	63.66	69.61	70.79
Spambase	$\rho_{+1} = \rho_{-1} = 0.2$	87.03	87.03	40.04	**88.67**	62.22
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	**88.61**	**88.61**	39.46	76.80	63.33
	$\rho_{+1} = \rho_{-1} = 0.4$	**81.28**	**81.28**	39.17	**82.03**	66.00

Table 2: $\mathcal{U}_{Acc}$ measure of classification (linear) algorithms on UCI benchmark data sets. Entries within 1% from the best in each row are in bold. All the methods use *linear* kernel. *All method-specific parameters are estimated through cross-validation.* We show the best performing NHERD variant ('project' and 'exact') in each case.

Using Gaussian kernel. For kernelized algorithms, we set the Gaussian kernel width parameter γ to $1/d$ where d is the dimensionality of data (the default parameter setting in `libsvm`). The results comparing SVM based methods are presented in Tables 4 and 7 for accuracy and AM measure respectively. We see a similar trend in performances as in the case of linear kernel. Note that the NHERD method is not kernelizable, so the results are omitted.

Overall, the experimental results support the theoretical guarantees; we observe that the proposed methods are competitive and are able to tolerate moderate to high amounts of label noise in the data.

Data set	Noise rates	Logistic Regression	Approach 1: (3) with $\ell_{\log}$	Approach 2: (8) with $\ell_{\log}$
Breast cancer	$\rho_{+1} = \rho_{-1} = 0.2$	65.86	**70.12**	66.40
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	64.11	**70.07**	**69.46**
	$\rho_{+1} = \rho_{-1} = 0.4$	59.51	**67.79**	56.43
Diabetes	$\rho_{+1} = \rho_{-1} = 0.2$	73.52	**76.04**	73.52
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	71.96	**75.52**	72.48
	$\rho_{+1} = \rho_{-1} = 0.4$	**67.62**	65.89	**66.75**
Thyroid	$\rho_{+1} = \rho_{-1} = 0.2$	82.54	**87.80**	82.54
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	**82.26**	80.34	**82.23**
	$\rho_{+1} = \rho_{-1} = 0.4$	77.28	**83.10**	76.36
German	$\rho_{+1} = \rho_{-1} = 0.2$	66.33	**71.80**	66.33
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	66.93	**71.40**	68.33
	$\rho_{+1} = \rho_{-1} = 0.4$	55.87	**67.19**	55.41
Heart	$\rho_{+1} = \rho_{-1} = 0.2$	81.23	**82.96**	81.23
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	81.73	**84.44**	81.73
	$\rho_{+1} = \rho_{-1} = 0.4$	**73.58**	57.04	**73.58**
Image	$\rho_{+1} = \rho_{-1} = 0.2$	**82.90**	**82.45**	**82.90**
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	**82.07**	**82.55**	**82.07**
	$\rho_{+1} = \rho_{-1} = 0.4$	**76.25**	63.47	**76.25**
Spambase	$\rho_{+1} = \rho_{-1} = 0.2$	87.72	**89.80**	87.72
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	72.37	**89.28**	72.37
	$\rho_{+1} = \rho_{-1} = 0.4$	79.88	**80.22**	79.88

Table 3: $\mathcal{U}_{Acc}$ measure of logistic loss based classification algorithms on UCI benchmark data sets. Entries within 1% from the best in each row are in bold. All the methods use *linear* kernel. *All method-specific parameters are estimated through cross-validation.*

Data set	Noise rates	SVM	C-SVM	PAM	RP
Breast cancer	$\rho_{+1} = \rho_{-1} = 0.2$	**70.77**	**70.77**	67.45	65.91
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	71.02	71.02	**73.31**	70.01
	$\rho_{+1} = \rho_{-1} = 0.4$	62.64	62.64	60.56	**63.92**
Diabetes	$\rho_{+1} = \rho_{-1} = 0.2$	**74.91**	73.35	**74.65**	72.40
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	**73.87**	73.09	72.66	68.14
	$\rho_{+1} = \rho_{-1} = 0.4$	55.30	52.86	63.45	**65.19**
Thyroid	$\rho_{+1} = \rho_{-1} = 0.2$	**92.23**	**92.23**	91.92	83.53
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	**84.09**	**84.09**	**85.33**	80.06
	$\rho_{+1} = \rho_{-1} = 0.4$	73.86	73.86	**82.56**	84.43
German	$\rho_{+1} = \rho_{-1} = 0.2$	**74.20**	**74.20**	73.80	72.14
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	70.40	70.40	70.67	**72.60**
	$\rho_{+1} = \rho_{-1} = 0.4$	**61.45**	**61.45**	59.73	59.52
Heart	$\rho_{+1} = \rho_{-1} = 0.2$	66.17	70.86	**78.27**	**78.27**
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	**77.04**	**77.04**	73.83	68.64
	$\rho_{+1} = \rho_{-1} = 0.4$	60.00	60.74	67.41	68.89
Image	$\rho_{+1} = \rho_{-1} = 0.2$	**94.09**	**94.09**	92.36	80.50
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	**91.50**	**91.50**	86.26	73.86
	$\rho_{+1} = \rho_{-1} = 0.4$	**81.11**	**81.11**	80.38	75.29
Spambase	$\rho_{+1} = \rho_{-1} = 0.2$	**78.41**	**78.41**	77.30	59.94
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	**75.16**	**75.16**	**75.46**	57.29
	$\rho_{+1} = \rho_{-1} = 0.4$	62.72	**65.20**	63.55	55.01

Table 4: $\mathcal{U}_{Acc}$ measure of classification (kernelized) algorithms on UCI benchmark data sets. Entries within 1% from the best in each row are in bold. All the methods use *Gaussian* kernel with width $\gamma = 1/d$ (where d is the number of dimensions). *All method-specific parameters are estimated through cross-validation.* NHERD algorithm is excluded as it is not kernelizable.

Data set	Noise rates	SVM	C-SVM	PAM	NHERD	RP
Breast cancer	$\rho_{+1} = \rho_{-1} = 0.2$	50.82	50.82	62.94	**66.14**	37.58
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	57.48	57.48	59.94	**64.28**	62.69
	$\rho_{+1} = \rho_{-1} = 0.4$	52.59	50.83	**56.52**	**56.21**	**56.02**
Diabetes	$\rho_{+1} = \rho_{-1} = 0.2$	70.85	70.85	69.90	**74.48**	72.17
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	**75.26**	**75.26**	66.19	**76.66**	73.80
	$\rho_{+1} = \rho_{-1} = 0.4$	63.15	63.15	60.16	**71.88**	69.00
Thyroid	$\rho_{+1} = \rho_{-1} = 0.2$	72.16	72.16	67.25	**77.67**	74.16
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	79.69	79.69	55.05	**83.99**	74.05
	$\rho_{+1} = \rho_{-1} = 0.4$	64.23	64.23	55.45	**82.97**	66.62
German	$\rho_{+1} = \rho_{-1} = 0.2$	62.15	62.15	64.68	**70.64**	67.72
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	69.59	69.59	63.09	**70.45**	65.23
	$\rho_{+1} = \rho_{-1} = 0.4$	54.13	53.51	54.62	54.70	**56.00**
Heart	$\rho_{+1} = \rho_{-1} = 0.2$	**81.82**	**81.82**	75.05	**82.97**	75.99
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	77.98	77.98	73.10	**82.19**	78.46
	$\rho_{+1} = \rho_{-1} = 0.4$	**76.42**	67.81	66.16	52.07	72.85
Image	$\rho_{+1} = \rho_{-1} = 0.2$	76.43	76.43	67.29	76.75	**79.23**
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	75.70	75.70	67.13	**80.21**	76.87
	$\rho_{+1} = \rho_{-1} = 0.4$	**69.68**	**69.68**	58.03	**70.64**	**70.68**
Spambase	$\rho_{+1} = \rho_{-1} = 0.2$	85.88	85.88	50.00	**88.62**	61.19
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	**88.27**	**88.27**	52.07	80.06	67.67
	$\rho_{+1} = \rho_{-1} = 0.4$	78.81	78.81	50.00	**81.51**	63.26

Table 5: $\mathcal{U}_{AM}$ measure of classification (linear) algorithms on UCI benchmark data sets. Entries within 1% from the best in each row are in bold. All the methods use *linear* kernel. *All method-specific parameters are estimated through cross-validation.* We show the best performing NHERD variant ('project' and 'exact') in each case.

Data set	Noise rates	Logistic Regression	Approach 1: (3) with $\ell_{\log}$	Approach 2: (8) with $\ell_{\log}$
Breast cancer	$\rho_{+1} = \rho_{-1} = 0.2$	**65.95**	59.58	**65.20**
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	61.61	56.28	**65.75**
	$\rho_{+1} = \rho_{-1} = 0.4$	**57.11**	51.50	54.50
Diabetes	$\rho_{+1} = \rho_{-1} = 0.2$	**74.70**	63.37	**74.70**
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	**73.38**	63.13	**73.74**
	$\rho_{+1} = \rho_{-1} = 0.4$	**68.18**	56.07	**67.19**
Thyroid	$\rho_{+1} = \rho_{-1} = 0.2$	78.70	**82.42**	78.70
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	**79.28**	68.04	**79.38**
	$\rho_{+1} = \rho_{-1} = 0.4$	**73.46**	53.19	**72.41**
German	$\rho_{+1} = \rho_{-1} = 0.2$	**69.24**	67.47	**69.24**
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	**69.59**	53.87	**70.44**
	$\rho_{+1} = \rho_{-1} = 0.4$	**57.07**	51.50	**56.65**
Heart	$\rho_{+1} = \rho_{-1} = 0.2$	**81.48**	**80.92**	**81.48**
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	81.73	**83.37**	81.73
	$\rho_{+1} = \rho_{-1} = 0.4$	**74.13**	51.59	**74.13**
Image	$\rho_{+1} = \rho_{-1} = 0.2$	**81.79**	80.23	**81.79**
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	**81.12**	**81.18**	**81.12**
	$\rho_{+1} = \rho_{-1} = 0.4$	**75.87**	56.60	**75.87**
Spambase	$\rho_{+1} = \rho_{-1} = 0.2$	**88.38**	**89.05**	**88.38**
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	76.78	**88.27**	76.78
	$\rho_{+1} = \rho_{-1} = 0.4$	**80.97**	75.80	**80.97**

Table 6: $\mathcal{U}_{AM}$ measure of logistic loss based classification algorithms on UCI benchmark data sets. Entries within 1% from the best in each row are in bold. All the methods use *linear* kernel. *All method-specific parameters are estimated through cross-validation.*

Data set	Noise rates	SVM	C-SVM	PAM	RP
Breast cancer	$\rho_{+1} = \rho_{-1} = 0.2$	56.28	56.28	**58.02**	54.83
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	56.96	56.96	57.08	**58.15**
	$\rho_{+1} = \rho_{-1} = 0.4$	50.84	50.84	49.84	**52.89**
Diabetes	$\rho_{+1} = \rho_{-1} = 0.2$	**70.00**	66.68	**70.07**	68.20
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	**73.98**	71.29	**72.90**	69.32
	$\rho_{+1} = \rho_{-1} = 0.4$	56.75	52.04	58.48	**61.41**
Thyroid	$\rho_{+1} = \rho_{-1} = 0.2$	**88.87**	**88.87**	**89.77**	84.72
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	74.97	74.97	**78.26**	68.13
	$\rho_{+1} = \rho_{-1} = 0.4$	66.70	66.70	74.53	**80.59**
German	$\rho_{+1} = \rho_{-1} = 0.2$	**63.51**	**63.51**	**64.46**	**64.41**
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	**67.79**	**67.79**	**67.71**	65.86
	$\rho_{+1} = \rho_{-1} = 0.4$	52.60	52.60	53.27	**54.61**
Heart	$\rho_{+1} = \rho_{-1} = 0.2$	63.98	68.86	**77.41**	**77.65**
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	**77.61**	**77.61**	74.99	69.59
	$\rho_{+1} = \rho_{-1} = 0.4$	57.80	56.44	**66.71**	65.85
Image	$\rho_{+1} = \rho_{-1} = 0.2$	**93.51**	**93.51**	91.45	80.47
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	**92.00**	**92.00**	87.40	75.42
	$\rho_{+1} = \rho_{-1} = 0.4$	**78.90**	**78.90**	77.78	75.58
Spambase	$\rho_{+1} = \rho_{-1} = 0.2$	**77.11**	**77.11**	75.82	56.44
	$\rho_{+1} = 0.3, \rho_{-1} = 0.1$	**77.47**	**77.47**	**77.75**	57.42
	$\rho_{+1} = \rho_{-1} = 0.4$	55.52	**59.57**	**60.84**	53.27

Table 7: $\mathcal{U}_{AM}$ measure of classification (kernelized) algorithms on UCI benchmark data sets. Entries within 1% from the best in each row are in bold. All the methods use *Gaussian* kernel with width $\gamma = 1/d$ (where d is the number of dimensions). *All method-specific parameters are estimated through cross-validation.* NHERD algorithm is excluded as it is not kernelizable.

6.3 Knowledge of noise rates

The proposed algorithms require the knowledge of noise rates ρ_{+1} and ρ_{-1}. However, in practice, we do not know the true value of noise rates, and therefore we resort to cross-validating the values in our experiments. In some cases (and domains), we may be able to approximately specify noise rates. This motivates our study presented in Figure 3. True noise rates $\rho_{+1} = \rho_{-1} = \rho$ are misspecified as $(\rho_{+1} \pm \epsilon, \rho_{-1} \pm \epsilon)$ for $\epsilon \in \{0.1, 0.2, 0.3, 0.4\}$. The ratio between the average accuracy for a given ϵ and the accuracy at $\epsilon = 0$, i.e. when true noise rates are specified, is a measure of sensitivity of the algorithms to ϵ-misspecification of noise rates. We would want the ratio to be close to 1 for a given ϵ, which would suggest that the method is fairly robust with respect to the ϵ-misspecification. The results in Figure 3 show that the proposed methods are robust to ϵ-misspecification of noise rates, which in turn suggests that our methods can find better use in applications where labels can be noisy *and* noise rates are approximately known, without resorting to ad-hoc cross-validation procedures on the noisy data. We emphasize here that in case the true noise rates are known, our methods can benefit from that knowledge as observed from our experiments, whereas the competitive methods *cannot* as they do not involve noise rates.

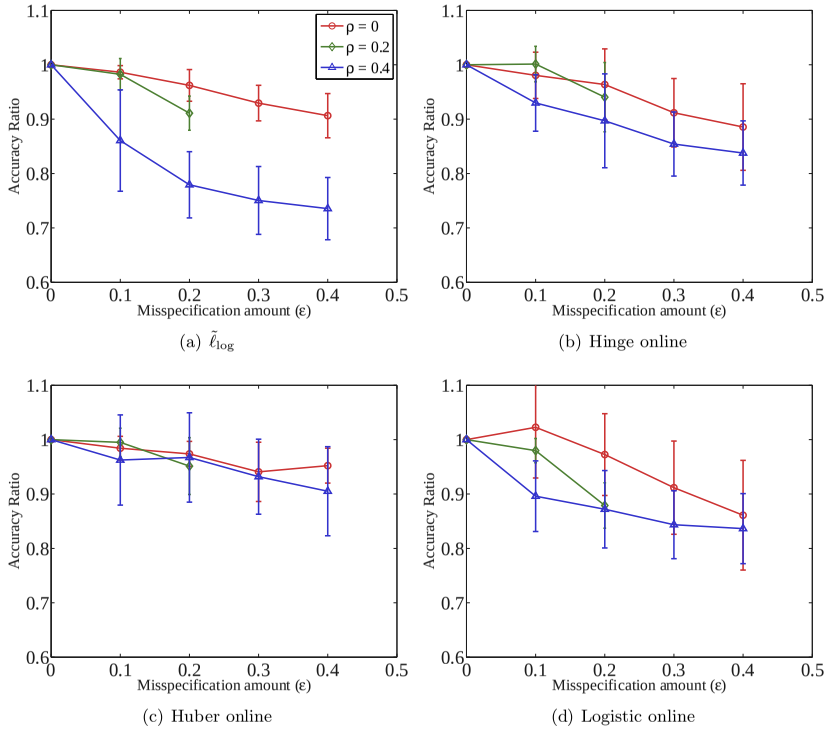
7. Conclusions and Future Work

We addressed learning in the presence of asymmetric random label noise with respect to general cost-sensitive utilities. We have obtained general theoretical results as well as efficient algorithms for this setting using the methods of unbiased estimators and weighted loss functions. The proposed algorithms are easy to implement and the classification performance is encouraging even at high noise rates and in particular is competitive with state-of-the-art methods on benchmark data. Our developments provide a new family of methods that can be applied to the positive-unlabeled learning problem (Elkan and Noto, 2008), but the implications of our methods for this setting should be carefully analyzed. We could consider harder noise models such as label noise depending on the example, and nastier variants of label noise where labels to flip are chosen adversarially.

Our analysis in this paper covers cost-sensitive classification losses, but there are other measures used in practice such as F_β, that are not covered by our family. Consistent learning for such general performance measures is beginning to be understood in the noise-free setting (Koyejo et al., 2014; Narasimhan et al., 2014). It would be interesting to see if we can extend some of the ideas in this paper to more general utility measures. It will also be of interest to extend our methods to deal with label noise in more general learning problems such as classification with a reject option, multiclass classification, learning with partial labels, learning to rank, and multilabel classification. Some of these extensions have begun to occur (van Rooyen and Williamson, 2015) but others are yet to be explored.

Acknowledgments

We gratefully acknowledge the support of NSF under grants CCF-1320746 and CCF-1117055; P.R. acknowledges the support of ARO via W911NF-12-1-0390 and NSF via IIS-1149803, IIS-1320894. A.T. acknowledges the support of NSF via CCF-1422157.



(a) $\tilde{\ell}_{\log}$ (b) Hinge online

(c) Huber online (d) Logistic online

Figure 3: Study of sensitivity of batch ($\tilde{\ell}_{\log}$) and online (Hinge, Huber and Logistic) methods (Algorithm 1) to specification of noise rates ρ_{+1} and ρ_{-1}. True noise rates $\rho_{+1} = \rho_{-1} = \rho$ are misspecified as $(\rho_{+1} \pm \epsilon, \rho_{-1} \pm \epsilon)$ for $\epsilon \in \{0.1, 0.2, 0.3, 0.4\}$. The ratio between the average accuracy for a given ϵ and the accuracy at $\epsilon = 0$, i.e. when true noise rates are specified, is plotted for different values of noise rates ρ. The ratio is computed for each of the 6 UCI data sets in Table 1 and the mean and the standard deviation of the ratios are shown. Ratio being equal to 1 for a given ϵ means that the performance of the algorithm, on average, is unaltered by misspecification of noise rates up to ϵ. As expected, the ratio decreases, i.e. the algorithms perform worse as ϵ increases. Most of the ratios being close to 1 suggests that the proposed methods are fairly robust with respect to ϵ-misspecification of noise rates.

References

D. Angluin and P. Laird. Learning from noisy examples. *Mach. Learn.*, 2(4):343–370, 1988.

Javed A. Aslam and Scott E. Decatur. On the sample complexity of noise-tolerant learning. *Inf. Process. Lett.*, 57(4):189–195, 1996.

Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. Agnostic online learning. In *Conference on Learning Theory (COLT)*, 2009.

Battista Biggio, Blaine Nelson, and Pavel Laskov. Support vector machines under adversarial label noise. *J. Mach. Learn. Res. (JMLR)*, 20:97–112, 2011.

Gilles Blanchard and Clayton Scott. Decontamination of mutually contaminated models. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1–9, 2014.

A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Computational Learning Theory (COLT)*, pages 92–100. ACM, 1998.

Tom Bylander. Learning linear threshold functions in the presence of classification noise. In *Conference on Learning Theory (COLT)*, pages 340–347, NY, USA, 1994. ACM.

Tom Bylander. Learning noisy linear threshold functions. *Technical Report*, 1998.

Nicolò Cesa-Bianchi, Eli Dichterman, Paul Fischer, Eli Shamir, and Hans Ulrich Simon. Sample-efficient strategies for learning in the presence of noise. *J. ACM*, 46(5):684–719, 1999.

Nicolò Cesa-Bianchi, Shai Shalev-Shwartz, and Ohad Shamir. Online learning of noisy data. *IEEE Transactions on Information Theory*, 57(12):7907–7931, 2011.

Edith Cohen. Learning noisy perceptrons by a perceptron in polynomial time. In *Foundations of Computer Science*, pages 514–523. IEEE, 1997.

K. Crammer and D. Lee. Learning via Gaussian Herding. In *Neural Information Processing Systems (NIPS)*, pages 451–459, 2010.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *J. Mach. Learn. Res. (JMLR)*, 7:551–585, 2006.

Koby Crammer, Alex Kulesza, and Mark Dredze. Adaptive regularization of weight vectors. In *Neural Information Processing Systems (NIPS)*, pages 414–422, 2009.

Mark Dredze, Koby Crammer, and Fernando Pereira. Confidence-weighted linear classification. In *International Conference on Machine Learning (ICML)*, pages 264–271, 2008.

C. Elkan and K. Noto. Learning classifiers from only positive and unlabeled data. In *Intl. Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 213–220, 2008.

Charles Elkan. The foundations of cost-sensitive learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, IJCAI'01, pages 973–978, 2001.

Yoav Freund. A more robust boosting algorithm, 2009. preprint `arXiv:0905.2138 [stat.ML]` available at `http://arxiv.org/abs/0905.2138`.

Aritra Ghosh, Naresh Manwani, and P. S. Sastry. Making risk minimization tolerant to label noise. *CoRR*, abs/1403.3610, 2014. URL `http://arxiv.org/abs/1403.3610`.

T. Graepel and R. Herbrich. The kernel Gibbs sampler. In *Neural Information Processing Systems (NIPS)*, pages 514–520, 2000.

Roni Khardon and Gabriel Wachman. Noise tolerant variants of the perceptron algorithm. *J. Mach. Learn. Res. (JMLR)*, 8:227–248, 2007.

Oluwasanmi O Koyejo, Nagarajan Natarajan, Pradeep K Ravikumar, and Inderjit S Dhillon. Consistent binary classification with generalized performance metrics. In *Neural Information Processing Systems (NIPS)*, pages 2744–2752, 2014.

Neil D. Lawrence and Bernhard Schölkopf. Estimating a kernel Fisher discriminant in the presence of label noise. In *International Conference on Machine Learning (ICML)*, pages 306–313, 2001.

Bing Liu, Yang Dai, Xiaoli Li, Wee Sun Lee, and Philip S Yu. Building text classifiers using positive and unlabeled examples. In *ICDM 2003.*, pages 179–186. IEEE, 2003.

Philip M. Long and Rocco A. Servedio. Random classification noise defeats all convex potential boosters. *Mach. Learn.*, 78(3):287–304, 2010.

Yves Lucet. What shape is your conjugate? a survey of computational convex analysis and its applications. *SIAM Rev.*, 52(3):505–542, August 2010. ISSN 0036-1445.

Naresh Manwani and P. S. Sastry. Noise tolerance under risk minimization. *IEEE Trans. Syst. Man and Cybern. Part B*, 2013. URL: `http://arxiv.org/abs/1109.5231`.

Aditya Menon, Harikrishna Narasimhan, Shivani Agarwal, and Sanjay Chawla. On the statistical consistency of algorithms for binary classification under class imbalance. In *International Conference on Machine Learning (ICML)*, pages 603–611, 2013.

Fantine Mordelet and J-P Vert. A bagging SVM to learn from positive and unlabeled examples. *Pattern Recognition Letters*, 37:201–209, 2014.

Harikrishna Narasimhan, Rohit Vaish, and Shivani Agarwal. On the statistical consistency of plug-in classifiers for non-decomposable performance measures. In *Neural Information Processing Systems (NIPS)*, pages 1493–1501, 2014.

Nagarajan Natarajan, Inderjit Dhillon, Pradeep Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Neural Information Processing Systems (NIPS)*, pages 1196–1204, 2013.

A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM J. on Opt.*, 19(4):1574–1609, 2009.

David F. Nettleton, A. Orriols-Puig, and A. Fornells. A study of the effect of different types of noise on the precision of supervised learning techniques. *Artif. Intell. Rev.*, 33 (4):275–306, 2010.

C. Scott. A rate of convergence for mixture proportion estimation, with application to learning from noisy labels. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 38 of *JMLR Workshop and Conference Proceedings*, pages 838–846, 2015.

Clayton Scott. Calibrated asymmetric surrogate losses. *Electronic J. of Stat.*, 6:958–992, 2012.

Clayton Scott, Gilles Blanchard, and Gregory Handy. Classification with asymmetric label noise: Consistency and maximal denoising. *Conference on Learning Theory (COLT)*, 30: 489–511, 2013.

G. Stempfel and L. Ralaivola. Learning kernel perceptrons on noisy data using random projections. In *Algorithmic Learning Theory (ALT)*, pages 328–342. Springer, 2007.

G. Stempfel, L. Ralaivola, and F. Denis. Learning from noisy data using hyperplane sampling and sample averages. 2007.

Guillaume Stempfel and Liva Ralaivola. Learning SVMs from sloppily labeled data. In *Artificial Neural Networks*, pages 884–893. Springer-Verlag, 2009.

Brendan van Rooyen and Robert C Williamson. Learning in the presence of corruption, 2015. arXiv preprint arXiv:1504.00091.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning (ICML)*, pages 928–936, 2003.

Journal of Machine Learning Research 18 (2018) 1-42 Submitted 5/15; Revised 9/16; Published 4/18

Provably Correct Algorithms for Matrix Column Subset Selection with Selectively Sampled Data

Yining Wang YININGWA@CS.CMU.EDU
Aarti Singh AARTI@CS.CMU.EDU
Machine Learning Department, School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213, USA

Editor: Sujay Sanghavi

Abstract

We consider the problem of matrix column subset selection, which selects a subset of columns from an input matrix such that the input can be well approximated by the span of the selected columns. Column subset selection has been applied to numerous real-world data applications such as population genetics summarization, electronic circuits testing and recommendation systems. In many applications the complete data matrix is unavailable and one needs to select representative columns by inspecting only a small portion of the input matrix. In this paper we propose the first provably correct column subset selection algorithms for partially observed data matrices. Our proposed algorithms exhibit different merits and limitations in terms of statistical accuracy, computational efficiency, sample complexity and sampling schemes, which provides a nice exploration of the tradeoff between these desired properties for column subset selection. The proposed methods employ the idea of feedback driven sampling and are inspired by several sampling schemes previously introduced for low-rank matrix approximation tasks (Drineas et al., 2008; Frieze et al., 2004; Deshpande and Vempala, 2006; Krishnamurthy and Singh, 2014). Our analysis shows that, under the assumption that the input data matrix has incoherent rows but possibly coherent columns, all algorithms provably converge to the best low-rank approximation of the original data as number of selected columns increases. Furthermore, two of the proposed algorithms enjoy a relative error bound, which is preferred for column subset selection and matrix approximation purposes. We also demonstrate through both theoretical and empirical analysis the power of feedback driven sampling compared to uniform random sampling on input matrices with highly correlated columns.

Keywords: Column subset selection, active learning, leverage scores

1. Introduction

Given a matrix $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$, the *column subset selection* problem aims to find s exact columns in $\mathbf{M}$ that capture as much of $\mathbf{M}$ as possible. More specifically, we want to select s columns of $\mathbf{M}$ to form a column sub-matrix $\mathbf{C} \in \mathbb{R}^{n_1 \times s}$ to minimize the norm of the following residue

$$\min_{\mathbf{X} \in \mathbb{R}^{s \times n_2}} \|\mathbf{M} - \mathbf{C}\mathbf{X}\|_\xi = \|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_\xi, \tag{1}$$

where $\mathbf{C}^\dagger$ is the Moore-Penrose pseudoinverse of $\mathbf{C}$ and $\xi = 2$ or F denotes the spectral or Frobenius norm. In this paper we mainly focus on the Frobenius norm, as was the case in previous theoretical analysis for sampling based column subset selection algorithms (Drineas et al., 2008; Frieze et al., 2004; Deshpande and Vempala, 2006; Deshpande et al., 2006). To evaluate the performance of column subset selection, one compares the residue norm defined in Eq. (1) with $\|\mathbf{M} - \mathbf{M}_k\|_\xi$, where $\mathbf{M}_k$ is the best rank-k approximation of $\mathbf{M}$. Usually the number of selected columns s is larger than or equal to the target rank k. Two forms of error guarantee are common: additive error guarantee in Eq. (2) and relative error guarantee in Eq. (3), with $0 < \epsilon < 1$ and $c > 1$ (ideally $c = 1 + \epsilon$).

$$\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_\xi \leq \|\mathbf{M} - \mathbf{M}_k\|_\xi + \epsilon\|\mathbf{M}\|_F; \tag{2}$$

$$\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_\xi \leq c\|\mathbf{M} - \mathbf{M}_k\|_\xi. \tag{3}$$

In general, relative error bound is much more appreciated because $\|\mathbf{M}\|_\xi$ is usually large in practice, while $\|\mathbf{M} - \mathbf{M}_k\|_2$ is expected to be small when the goal is low-rank approximation. In addition, when $\mathbf{M}$ is an exact low-rank matrix Eq. (3) implies perfect reconstruction, while the error in Eq. (2) remains non-zero. The column subset selection problem can be considered as a form of *unsupervised feature selection* or *prototype selection*, which arises frequently in the analysis of large data sets. For example, column subset selection has been applied to various tasks such as summarizing population genetics, testing electronic circuits, recommendation systems, etc. Interested readers should refer to (Boutsidis et al., 2009; Balzano et al., 2010a) for further motivations.

Many methods have been proposed for the column subset selection problem (Chan, 1987; Gu and Eisenstat, 1996; Frieze et al., 2004; Deshpande et al., 2006; Drineas et al., 2008; Boutsidis et al., 2014). An excellent summarization of these methods and their theoretical guarantee is available in Table 1 in (Boutsidis et al., 2009). Most of these methods can be roughly categorized into two classes. One class of algorithms are based on *rank-revealing QR* (RRQR) decomposition (Chan, 1987; Gu and Eisenstat, 1996) and it has been shown in (Boutsidis et al., 2009) that RRQR is nearly optimal in terms of residue norm under the $s = k$ setting, that is, exact k columns are selected to reconstruct an input matrix. On the other hand, sampling based methods (Frieze et al., 2004; Deshpande et al., 2006; Drineas et al., 2008) try to select columns by sampling from certain distributions over all columns of an input matrix. Extension of sampling based methods to general low-rank matrix approximation problems is also investigated (Cohen et al., 2015; Bhojanapalli et al., 2015). These algorithms are much faster than RRQR and achieves comparable performance if the sampling distribution is carefully selected and slight over-sampling (i.e., $s > k$) is allowed (Deshpande et al., 2006; Drineas et al., 2008). In (Boutsidis et al., 2009) sampling based and RRQR based algorithms are unified to arrive at an efficient column subset selection method that uses exactly $s = k$ columns and is nearly optimal.

Although the column subset selection problem with access to the full input matrix has been extensively studied, often in practice it is hard or even impossible to obtain the complete data. For example, for the genetic variation detection problem it could be expensive and time-consuming to obtain full DNA sequences of an entire population. Several heuristic algorithms have been proposed recently for column subset selection with missing data, including the Block OMP algorithm (Balzano et al., 2010a) and the group Lasso formulation explored in (Bien et al., 2010). Nevertheless, no theoretical guarantee or error bounds have been derived for these methods. The presence of missing data poses new challenges for column subset selection, as many well-established algorithms seem incapable

of handling missing data in an elegant way. Below we identify a few key challenges that prevent application of previous theoretical results on column subset selection under the missing data setting:

- **Coherent matrix design**: most previous results on the completion or recovery of low rank matrices with incomplete data assume the underlying data matrix is *incoherent* (Recht, 2011; Candes and Plan, 2010; Keshavan et al., 2010), which intuitively assumes all rows and columns in the data matrix are weakly correlated. [1] On the other hand, previous algorithms on column subset selection and matrix CUR decomposition spent most efforts on dealing with coherent matrices (Deshpande et al., 2006; Drineas et al., 2008; Boutsidis et al., 2009; Boutsidis and Woodruff, 2014). In fact, one can show that under standard incoherence assumptions of matrix completion algorithms a high-quality column subset can be obtained by sampling each column uniformly at random, which trivializes the problem (Xu et al., 2015). Such gap in problem assumptions renders column subset selection on incomplete coherent matrices particularly difficult. In this paper, we explore the possibility of a weaker incoherence assumption that bridges the gap. We present and discuss detailed assumptions considered in this paper in Sec. 1.1.

- **Limitation of existing sampling schemes**: previous matrix completion methods usually assume the observed data are sampled uniformly at random. However, in (Krishnamurthy and Singh, 2014) it is proved that uniform sampling (in fact any sampling scheme with apriori fixed sampling distribution) is not sufficient to complete a coherent matrix. Though in (Chen et al., 2013) a provably correct sampling scheme was proposed for any matrix based on statistical leverage scores, which is also the key ingredient of many previous column subset selection and matrix CUR decomposition algorithms (Drineas et al., 2008; Boutsidis et al., 2009; Boutsidis and Woodruff, 2014), it is very difficult to approximate the leverage scores of an incomplete coherent matrix. Common perturbation results on singular vector space (e.g., Wedin's theorem) fail because closeness between two subspaces does not imply closeness in their leverage scores since the latter are defined in an infinity norm manner (see Section 2.1 for details).

- **Limitation of zero filling**: A straightforward algorithm for missing data column subset selection is to first fill all unobserved entries with zero and then properly scale the observed ones so that the completed matrix is consistent with the underlying data matrix in expectation (Achlioptas and McSherry, 2007; Achlioptas et al., 2013). Column subset selection algorithms designed for fully observed data could be applied afterwards on the zero-filled matrix. However, the zero filling procedure can change the underlying subspace of a matrix drastically (Balzano et al., 2010b) and usually leads to additive error bounds as in Eq. (2). To achieve stronger relative error bounds we need an algorithm that goes beyond the zero filling idea.

In this paper, we propose three column subset selection algorithms based on the idea of *active sampling* of the input matrix. In our algorithms, observed matrix entries are

1. The precise definition of incoherence is given in Section 1.3.

chosen sequentially and in a feedback-driven manner. We motivate this sampling setting from both practical and theoretical perspectives. In applications where each entry of a data matrix $\mathbf{M}$ represents results from an expensive or time-consuming experiment, it makes sense to carefully select which entry to query (experiment), possibly in a feedback-driven manner, so as to reduce experimental cost. For example, if $\mathbf{M}$ has drugs as its columns and targets (proteins) as its rows, it makes sense to cautiously select drug-target pairs for sequential experimental study in order to find important drugs/targets with typical drug-target interactions. From a theoretical perspective, we show in Section 7.1 that no passive sampling scheme is capable of achieving relative-error column subset selection with high probability, even if the column space of $\mathbf{M}$ is incoherent. Such results suggest that active/adaptive sampling is to some extent unavoidable, unless both row and column spaces of $\mathbf{M}$ are incoherent.

We also remark that the algorithms we consider make very few measurements of the input matrix, which differs from previous feedback-driven re-sampling methods in the theoretical computer science literature (e.g., (Wang and Zhang, 2013)) that requires access to the entire input matrix. Active sampling has been shown to outperform all passive schemes in several settings (cf. (Haupt et al., 2011; Kolar et al., 2011)), and furthermore it works for completion of matrices with incoherent rows/columns under which passive learning provably fails (Krishnamurthy and Singh, 2013, 2014). To the best of our knowledge, the algorithms proposed in this paper are the first column subset selection algorithms for coherent matrices that enjoy theoretical error guarantee with missing data, whether passive or active. Furthermore, two of our proposed methods achieve relative error bounds.

1.1 Assumptions

Completing/approximating partially observed low-rank matrices using a subset of columns requires certain assumptions on the input data matrix $\mathbf{M}$ (Candes and Plan, 2010; Chen et al., 2013; Recht, 2011; Xu et al., 2015). To see this, consider the extreme-case example where the input data matrix $\mathbf{M}$ consists of *exactly* one non-zero element (i.e., $\mathbf{M}_{ij} = 1\{i = i^*, j = j^*\}$ for some $i^* \in [n_1]$ and $j^* \in [n_2]$). In this case, the relative approximation quality $c = \|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_\xi / \|\mathbf{M} - \mathbf{M}_1\|_\xi$ in Eq. (3) would be infinity if column j^* is not selected in $\mathbf{C}$. In addition, it is clearly impossible to correctly identify j^* using $o(n_1 n_2)$ observations even with active sampling strategies. Therefore, additional assumptions on $\mathbf{M}$ are required to provably approximate a partially observed matrix using column subsets.

In this work we consider the assumption that the top-k *column space* of the input matrix $\mathbf{M}$ is incoherent (detailed mathematical definition given in Sec. 2.1), while placing no incoherence or spikiness assumptions on the actual *columns*, rows or the row space of $\mathbf{M}$. In addition to the necessity of incoherence assumptions for incomplete matrix approximation problems discussed above, we further motivate the "one-sided" incoherence assumption from two perspectives:

- Column subset selection with incomplete observation remains a non-trivial problem even if the column space is assumed to be incoherent. Due to the possible heterogeneity of the columns, naive methods such as column subsets sampled uniformly at random are in general bad approximations of the original data matrix $\mathbf{M}$. Existing column

subset selection algorithms for fully-observed matrices also need to be majorly revised to accommodate missing matrix components.

- Compared to existing work on approximating low-rank incomplete matrices, our assumptions (one-sided incoherence) are arguably weaker. Xu et al. (2015) analyzed matrix CUR approximation of partially observed matrices, but assumed that both column and row spaces are incoherent; Krishnamurthy and Singh (2014) derived an adaptive sampling procedure to complete a low-rank matrix with only one-sided incoherence assumptions, but only achieved additive error bounds for noisy low-rank matrices.

- Finally, the one-sided incoherence assumption is reasonable in a number of practical scenarios. For example, in the application of drug-target interaction prediction, the one-sided incoherence assumption allows for highly specialized or diverse drugs while assuming some predictability between target protein responses.

1.2 Our contributions

The main contribution of this paper is three provably correct algorithms for column subset selection, which are inspired by existing work on column subset selection for fully-observed matrices, but only inspect a small portion of the input matrix. The sampling schemes for the proposed algorithms and their main merits and drawbacks are summarized below:

1. **Norm sampling**: The algorithm is simple and works for any input matrix with incoherent column subspace. However, it only achieves an additive error bound as in Eq. (2). It is also inferior than the other two proposed methods in terms of residue error on both synthetic and real-world data sets.

2. **Iterative norm sampling**: The iterative norm sampling algorithm enjoys relative error guarantees as in Eq. (3) at the expense of being much more complicated and computationally expensive. In addition, its correctness is only proved for low-rank matrices with incoherent column space corrupted with i.i.d. Gaussian noise.

3. **Approximate leverage score sampling**: The algorithm enjoys relative error guarantee for general (high-rank) input matrices with incoherent column space. However, it requires more over-sampling and its error bound is worse than the one for iterative norm sampling on noisy low-rank matrices. Moreover, to actually reconstruct the data matrix [2] the approximate leverage score sampling scheme requires sampling a subset of both entire rows and columns, while both norm based algorithms only require sampling of some entire columns.

In summary, our proposed algorithms offer a rich, provably correct toolset for column subset selection with missing data. Furthermore, a comprehensive understanding of the design tradeoffs among statistical accuracy, computational efficiency, sample complexity, and sampling scheme, etc. is achieved by analyzing different aspects of the proposed methods. Our analysis could provide further insights into other matrix completion/approximation tasks on partially observed data.

2. See Section 1.3 for the distinction between selection and reconstruction.

We also perform comprehensive experimental study of column subset selection with missing data using the proposed algorithms as well as modifications of heuristic algorithms proposed recently (Balzano et al., 2010a; Bien et al., 2010) on synthetic matrices and two real-world applications: tagging Single Nucleotide Polymorphisms (tSNP) selection and column based image compression. Our empirical study verifies most of our theoretical results and reveals a few interesting observations that are previously unknown. For instance, though leverage score sampling is widely considered as the state-of-the-art for matrix CUR approximation and column subset selection, our experimental results show that under certain low-noise regimes (meaning that the input matrix is very close to low rank) iterative norm sampling is more preferred and achieves smaller error. These observations open new questions and suggest the need for new analysis in related fields, even for the fully observed case.

1.3 Notations

For any matrix $\mathbf{M}$ we use $\mathbf{M}^{(i)}$ to denote the i-th column of $\mathbf{M}$. Similarly, $\mathbf{M}_{(i)}$ denotes the i-th row of $\mathbf{M}$. All norms $\|\cdot\|$ are ℓ_2 norms or the matrix spectral norm unless otherwise specified.

We assume the input matrix is of size $n_1 \times n_2$, $n = \max(n_1, n_2)$. We further assume that $n_1 \leq n_2$. We use $\boldsymbol{x}_i = \mathbf{M}^{(i)} \in \mathbb{R}^{n_1}$ to denote the i-th column of $\mathbf{M}$. Furthermore, for any column vector $\boldsymbol{x}_i \in \mathbb{R}^{n_1}$ and index subset $\Omega \subseteq [n_1]$, define the subsampled vector $\boldsymbol{x}_{i,\Omega}$ and the scaled subsampled vector $\mathcal{R}_\Omega(\boldsymbol{x}_i)$ as

$$\boldsymbol{x}_{i,\Omega} = \mathbf{1}_\Omega \circ \boldsymbol{x}_i, \quad \mathcal{R}_\Omega(\boldsymbol{x}_i) = \frac{n_1}{|\Omega|}\mathbf{1}_\Omega \circ \boldsymbol{x}_i, \tag{4}$$

where $\mathbf{1}_\Omega \in \{0,1\}^{n_1}$ is the indicator vector of Ω and $\circ$ is the Hadamard product (entrywise product). We also generalize the definition in Eq. (4) to matrices by applying the same operator on each column.

We use $\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_\xi$ to denote the *selection error* and $\|\mathbf{M} - \mathbf{C}\mathbf{X}\|_\xi$ to denote the *reconstruction error*. The difference between the two types of error is that for selection error an algorithm is only required to output indices of the selected columns while for reconstruction error an algorithm needs to output both the selected columns $\mathbf{C}$ and the coefficient matrix $\mathbf{X}$ so that $\mathbf{C}\mathbf{X}$ is close to $\mathbf{M}$. We remark that the reconstruction error always upper bounds the selection error due to Eq. (1). On the other hand, there is no simple procedure to compute $\mathbf{C}^\dagger\mathbf{M}$ when $\mathbf{M}$ is not fully observed.

1.4 Outline of the paper

The paper is organized as follows: in Section 2 we provide background knowledge and review several concepts that are important to our analysis. We then present main results of the paper, the three proposed algorithms and their theoretical guarantees in Section 3. Proofs for main results given in Section 3 are sketched in Section 4 and some technical lemmas and complete proof details are deferred to the appendix. In Section 5 we briefly describe previously proposed heuristic based algorithm for column subset selection with missing data and their implementation details. Experimental results are presented in Section 6 and we discuss several aspects including the limitation of passive sampling and time complexity of proposed algorithms in Section 7.

2. Preliminaries

This section provides necessary background knowledge for the analysis in this paper. We first review the concept of *coherence*, which plays an important row in sampling based matrix algorithms. We then summarize three matrix sampling schemes proposed in previous literature.

2.1 Subspace and vector incoherence

Incoherence plays a crucial role in various matrix completion and approximation tasks (Recht, 2011; Krishnamurthy and Singh, 2014; Candes and Plan, 2010; Keshavan et al., 2010). For any matrix $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$ of rank k, singular value decomposition yields $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{n_1 \times k}$ and $\mathbf{V} \in \mathbb{R}^{n_2 \times k}$ have orthonormal columns. Let $\mathcal{U} = \text{span}(\mathbf{U})$ and $\mathcal{V} = \text{span}(\mathbf{V})$ be the column and row space of $\mathbf{M}$. The *column space coherence* is defined as

$$\mu(\mathcal{U}) := \frac{n_1}{k} \max_{i=1}^{n_1} \|\mathbf{U}^\top \boldsymbol{e}_i\|_2^2 = \frac{n_1}{k} \max_{i=1}^{n_1} \|\mathbf{U}_{(i)}\|_2^2. \tag{5}$$

Note that $\mu(\mathcal{U})$ is always between 1 and n_1/k. Similarly, the *row space coherence* is defined as

$$\mu(\mathcal{V}) := \frac{n_2}{k} \max_{i=1}^{n_2} \|\mathbf{V}^\top \boldsymbol{e}_i\|_2^2 = \frac{n_2}{k} \max_{i=1}^{n_2} \|\mathbf{V}_{(i)}\|_2^2. \tag{6}$$

In this paper we also make use of incoherence level of vectors, which previously appeared in (Balzano et al., 2010b; Krishnamurthy and Singh, 2013, 2014). For a column vector $\boldsymbol{x} \in \mathbb{R}^{n_1}$, its incoherence is defined as

$$\mu(\boldsymbol{x}) := \frac{n_1 \|\boldsymbol{x}\|_\infty^2}{\|\boldsymbol{x}\|_2^2}. \tag{7}$$

It is an easy observation that if $\boldsymbol{x}$ lies in the subspace $\mathcal{U}$ then $\mu(\boldsymbol{x}) \leq k\mu(\mathcal{U})$. In this paper we adopt incoherence assumptions on the column space $\mathcal{U}$, which subsequently yields incoherent row vectors $\boldsymbol{x}_i$. No incoherence assumption on the row space $\mathcal{V}$ or row vectors $\mathbf{M}_{(i)}$ is made.

2.2 Matrix sampling schemes

Norm sampling: Norm sampling for column subset selection was proposed in (Frieze et al., 2004) and has found applications in a number of matrix computation tasks, e.g., approximate matrix multiplication (Drineas et al., 2006a) and low-rank or compressed matrix approximation (Drineas et al., 2006c,b). The idea is to sample each column with probability proportional to its squared ℓ_2 norm, i.e., $\Pr[i \in C] \propto \|\mathbf{M}^{(i)}\|_2^2$ for $i \in \{1, 2, \cdots, n_2\}$. These types of algorithms usually come with an additive error bound on their approximation performance.

Volume sampling: For volume sampling (Deshpande et al., 2006), a subset of columns C is picked with probability proportional to the volume of the simplex spanned by columns in C. That is, $\Pr[C] \propto \text{vol}(\Delta(C))$ where $\Delta(C)$ is the simplex spanned by $\{\mathbf{M}^{(C(1))}, \cdots, \mathbf{M}^{(C(k))}\}$. Computationally efficient volume sampling algorithms exist (Deshpande and Rademacher, 2010; Anari et al., 2016). These methods are based on the computation of characteristic polynomials of the projected data matrix (Deshpande and Rademacher, 2010) or an

Table 1: Summary of theoretical guarantees of proposed algorithms. s denotes the number of selected columns and m denotes the expected number of observed matrix entries. Dependency on failure probability δ and other poly-logarithmic dependency is omitted. $\mathcal{U}$ represents the column space of $\mathbf{A}$.

	Error type	Error bound	s	m	Assumptions
Norm	$\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_F$	$\|\mathbf{M} - \mathbf{M}_k\|_F + \epsilon\|\mathbf{M}\|_F$	$\Omega(k/\epsilon^2)$	$\widetilde{\Omega}(\mu_1 n)$	$\max_{i=1}^{n_2} \mu(\mathbf{M}_{(i)}) \leq \mu_1$
	$\|\mathbf{M} - \mathbf{C}\mathbf{X}\|_F$	$\|\mathbf{M} - \mathbf{M}_k\|_F + 2\epsilon\|\mathbf{M}\|_F$	$\Omega(k/\epsilon^2)$	$\widetilde{\Omega}(k\mu_1 n/\epsilon^4)$	same as above
Iter. norm	$\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_F$	$\sqrt{2.5^k(k+1)!}\|\mathbf{M} - \mathbf{M}_k\|_F$	k	$\widetilde{\Omega}(k^2\mu_0 n)$	$\mathbf{M} = \mathbf{A} + \mathbf{R}$; $\mu(\mathcal{U}) \leq \mu_0$
	$\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_F$	$\sqrt{1+3\epsilon}\|\mathbf{M} - \mathbf{M}_k\|_F$	$\Theta(k^2 \log k + k/\epsilon)$	$\widetilde{\Omega}\left(\frac{k\mu_0 n}{\epsilon}\left(k + \frac{1}{\epsilon}\right)\right)$	same as above
	$\|\mathbf{M} - \mathbf{C}\mathbf{X}\|_F$	$\sqrt{1+3\epsilon}\|\mathbf{M} - \mathbf{M}_k\|_F$	$\Theta(k^2 \log k + k/\epsilon)$	$\widetilde{\Omega}\left(\frac{k\mu_0 n}{\epsilon}\left(k + \frac{1}{\epsilon}\right)\right)$	same as above
Lev. score	$\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_F$	$3(1+\epsilon)\|\mathbf{M} - \mathbf{M}_k\|_F$	$\Omega(k^2/\epsilon^2)$	$\Omega(k^2\mu_0 n/\epsilon^2)$	$\mu(\mathcal{U}) \leq \mu_0$

MCMC sampling procedure (Anari et al., 2016). Under the partially observed setting, both approaches are difficult to apply. For the characteristic polynomials approach, one has to estimate the characteristic polynomial and essentially the least singular value of the target matrix $\mathbf{M}$ up to relative error bounds. This is not possible unless the matrix is very well-conditioned, which violates the setting that $\mathbf{M}$ is approximately low-rank. For the MCMC sampling procedure, it was shown in (Anari et al., 2016) that $O(kn_2)$ iterations are needed for the sampling Markov chain to mix. As each sampling iteration requires observing one entire column, performing $O(kn_2)$ iterations essentially requires observing $O(kn_2)$ columns, i.e., the entire matrix $\mathbf{M}$. On the other hand, an iterative norm sampling procedure is known to perform *approximate volume sampling* and therefore enjoys multiplicative approximation bounds for column subset selection (Deshpande and Vempala, 2006). In this paper we generalize the iterative norm sampling scheme to the partially observed setting and demonstrate similar multiplicative approximation error guarantees.

Leverage score sampling: The leverage score sampling scheme was introduced in (Drineas et al., 2008) to get relative error bounds for CUR matrix approximation and has later been applied to coherent matrix completion (Chen et al., 2013). For each row $i \in \{1, \cdots, n_1\}$ and column $j \in \{1, \cdots, n_2\}$ define $\mu_i := \frac{n_1}{k}\|\mathbf{U}^\top \boldsymbol{e}_i\|_2^2$ and $\nu_j := \frac{n_2}{k}\|\mathbf{V}^\top \boldsymbol{e}_j\|_2^2$ to be their *unnormalized leverage scores*, where $\mathbf{U} \in \mathbb{R}^{n_1 \times k}$ and $\mathbf{V} \in \mathbb{R}^{n_2 \times k}$ are the top-k left and right singular vectors of an input matrix $\mathbf{M}$. It was shown in (Drineas et al., 2008) that if rows and columns are sampled with probability proportional to their leverage scores then a relative error guarantee is possible for matrix CUR approximation and column subset selection.

3. Column subset selection via active sampling

In this section we propose three column subset selection algorithms that only observe a small portion of an input matrix. All algorithms employ the idea of active sampling to handle matrices with coherent rows. While Algorithm 1 achieves an additive reconstruction error guarantee for any matrix, Algorithm 2 achieves a relative-error reconstruction guarantee when the input matrix has certain structure. Finally, Algorithm 3 achieves a relative-error selection error bound for any general input matrix at the expense of slower error rate

Algorithm 1 Active norm sampling for column subset selection with missing data

1: **Input**: size of column subset s, expected number of samples per column m_1 and m_2.
2: **Norm estimation**: For each column i, sample each index in $\Omega_{1,i} \subseteq [n_1]$ i.i.d. from Bernoulli(m_1/n_1). observe $\boldsymbol{x}_{i,\Omega_{1,i}}$ and compute $\hat{c}_i = \frac{n_1}{m_1}\|\boldsymbol{x}_{i,\Omega_{1,i}}\|_2^2$. Define $\hat{f} = \sum_i \hat{c}_i$.
3: **Column subset selection**: Set $\mathbf{C} = \mathbf{0} \in \mathbb{R}^{n_1 \times s}$.
- For $t \in [s]$: sample $i_t \in [n_2]$ such that $\Pr[i_t = j] = \hat{c}_j/\hat{f}$. Observe $\mathbf{M}^{(i_t)}$ in full and set $\mathbf{C}^{(t)} = \mathbf{M}^{(i_t)}$.

4: **Matrix approximation**: Set $\widehat{\mathbf{M}} = \mathbf{0} \in \mathbb{R}^{n_1 \times n_2}$.
- For each column $\boldsymbol{x}_i$, sample each index in $\Omega_{2,i} \subseteq [n_1]$ i.i.d. from Bernoulli($m_{2,i}/n_1$), where $m_{2,i} = m_2 n_2 \hat{c}_i/\hat{f}$; observe $\boldsymbol{x}_{i,\Omega_{2,i}}$.
- Update: $\widehat{\mathbf{M}} = \widehat{\mathbf{M}} + (\mathcal{R}_{\Omega_{2,i}}(\boldsymbol{x}_i))\boldsymbol{e}_i^\top$.

5: **Output**: selected columns $\mathbf{C}$ and coefficient matrix $\mathbf{X} = \mathbf{C}^\dagger\widehat{\mathbf{M}}$.

and more sampled columns. Table 1 summarizes the main theoretical guarantees for the proposed algorithms.

3.1 ℓ_2 norm sampling

We first present an active norm sampling algorithm (Algorithm 1) for column subset selection under the missing data setting. The algorithm is inspired by the norm sampling work for column subset selection by Frieze et al. (Frieze et al., 2004) and the low-rank matrix approximation work by Krishnamurthy and Singh (Krishnamurthy and Singh, 2014).

The first step of Algorithm 1 is to estimate the ℓ_2 norm for each column by uniform subsampling. Afterwards, s columns of $\mathbf{M}$ are selected independently with probability proportional to their ℓ_2 norms. Finally, the algorithm constructs a sparse approximation of the input matrix by sampling each matrix entry with probability proportional to the square of the corresponding column's norm and then a $\mathbf{CX}$ approximation is obtained.

When the input matrix $\mathbf{M}$ has incoherent columns, the selection error as well as $\mathbf{CX}$ reconstruction error can be bounded as in Theorem 1.

Theorem 1 *Suppose* $\max_{i=1}^{n_2}\mu(\boldsymbol{x}_i) \le \mu_1$ *for some positive constant* μ_1. *Let* $\mathbf{C}$ *and* $\mathbf{X}$ *be the output of Algorithm 1. Denote* $\mathbf{M}_k$ *as the best rank-k approximation of* $\mathbf{M}$. *Fix* $\delta = \delta_1 + \delta_2 + \delta_3 > 0$. *With probability at least* $1 - \delta$, *we have*

$$\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_F \le \|\mathbf{M} - \mathbf{M}_k\|_F + \epsilon\|\mathbf{M}\|_F \tag{8}$$

provided that $s = \Omega(k\epsilon^{-2}/\delta_2)$, $m_1 = \Omega(\mu_1 \log(n/\delta_1))$. *Furthermore, if* $m_2 = \Omega(\mu_1 s \log^2(n/\delta_3)/(\delta_2\epsilon^2))$ *then with probability* $\ge 1 - \delta$ *we have the following bound on reconstruction error:*

$$\|\mathbf{M} - \mathbf{C}\mathbf{X}\|_F \le \|\mathbf{M} - \mathbf{M}_k\|_F + 2\epsilon\|\mathbf{M}\|_F. \tag{9}$$

Algorithm 2 Active iterative norm sampling for column subset selection for data corrupted by Gaussian noise

1: **Input**: target rank $k < \min(n_1, n_2)$, error tolerance parameter ϵ, δ and expected number of samples per column m.
2: **Entrywise sampling**: For each column i, sample each index in an index set $\Omega_i \subseteq [n_1]$ i.i.d. from Bernoulli(m/n_1). Observe $\boldsymbol{x}_{i,\Omega_i}$.
3: **Approximate volume sampling**: Set $C = \mathcal{U} = \emptyset$. Let $\mathbf{U}$ be an orthonormal basis of $\mathcal{U}$.
4: **for** $t = 1, 2, \cdots, k$ **do**
5: For $i \in \{1, \cdots, n_2\}$, compute $\hat{c}_i^{(t)} = \frac{n_1}{m}\|\boldsymbol{x}_{i,\Omega_i} - \mathbf{U}_{\Omega_i}(\mathbf{U}_{\Omega_i}^\top\mathbf{U}_{\Omega_i})^{-1}\mathbf{U}_{\Omega_i}^\top\boldsymbol{x}_{i,\Omega_i}\|_2^2$.
6: Set $\hat{f}^{(t)} = \sum_{i=1}^{n_2}\hat{c}_i^{(t)}$.
7: Select a column i_t at random, with probability $\Pr[i_t = j] = \hat{p}_j^{(t)} = \hat{c}_j^{(t)}/\hat{f}^{(t)}$.
8: Observe $\mathbf{M}^{(i_t)}$ in full and update: $C \leftarrow C \cup \{i_t\}$, $\mathcal{U} \leftarrow \text{span}(\mathcal{U}, \{\mathbf{M}^{(i_t)}\})$.
9: **end for**
10: **Active norm sampling**: set $T = (k+1)\log(k+1)$ and $s_1 = s_2 = \cdots = s_{T-1} = 5k$, $s_T = 10k/\epsilon\delta$; $S = \emptyset$, $\mathcal{S} = \emptyset$. Suppose $\mathbf{U}$ is an orthonormal basis of $\text{span}(\mathcal{U}, \mathcal{S})$.
11: **for** $t = 1, 2, \cdots, T$ **do**
12: For $i \in \{1, \cdots, n_2\}$, compute $\hat{c}_i^{(t)} = \frac{n_1}{m}\|\boldsymbol{x}_{i,\Omega_i} - \mathbf{U}_{\Omega_i}(\mathbf{U}_{\Omega_i}^\top\mathbf{U}_{\Omega_i})^{-1}\mathbf{U}_{\Omega_i}^\top\boldsymbol{x}_{i,\Omega_i}\|_2^2$.
13: Set $\hat{f}^{(t)} = \sum_{i=1}^{n_2}\hat{c}_i^{(t)}$.
14: Select s_t columns $S_t = (i_1, \cdots, i_{s_t})$ independently at random, with probability $\Pr[j \in S_t] = \hat{q}_j^{(t)} = \hat{c}_j^{(t)}/\hat{f}^{(t)}$.
15: Observe $\mathbf{M}^{(S_t)}$ in full and update: $S \leftarrow S \cup S_t$, $\mathcal{S} \leftarrow \text{span}(\mathcal{S}, \{\mathbf{M}^{(S_t)}\})$.
16: **end for**
17: **Matrix approximation**: $\widehat{\mathbf{M}} = \sum_{i=1}^{n_2}\mathbf{U}(\mathbf{U}_{\Omega_i}^\top\mathbf{U}_{\Omega_i})^{-1}\mathbf{U}_{\Omega_i}\boldsymbol{x}_{i,\Omega_i}\boldsymbol{e}_i^\top$, where $\mathbf{U} \in \mathbb{R}^{n_1 \times (s+k)}$ is an orthonormal basis of $\text{span}(\mathcal{U}_0, \mathcal{U}_1)$.
18: **Output**: selected column subsets $\mathbf{C} = (\mathbf{M}^{(C(1))}, \cdots, \mathbf{M}^{(C(k))}) \in \mathbb{R}^{n_1 \times k}$, $\mathbf{S} = (\mathbf{M}^{(C)}, \mathbf{M}^{(S_1)}, \cdots, \mathbf{M}^{(S_T)}) \in \mathbb{R}^{n_1 \times s}$ where $s = k + s_1 + \cdots + s_T$ and $\mathbf{X} = \mathbf{S}\mathbf{S}^\dagger\widehat{\mathbf{M}}$.

As a remark, Theorem 1 shows that one can achieve ϵ additive reconstruction error using Algorithm 1 with expected sample complexity (omitting dependency on δ)

$$\Omega\left(\mu_1 n_2 \log(n) + \frac{kn_1}{\epsilon^2} + \frac{k\mu_1 n_2 \log^2(n)}{\epsilon^4}\right) = \Omega(k\mu_1\epsilon^{-4} n \log^2 n).$$

3.2 Iterative norm sampling

In this section we present Algorithm 2, another active sampling algorithm based on the idea of iterative norm sampling and approximate volume sampling introduced in (Deshpande and Vempala, 2006). Though Algorithm 2 is more complicated than Algorithm 1, it achieves a relative error bound on inputs that are noisy perturbation of some underlying low-rank matrix.

Algorithm 2 employs the idea of *iterative norm sampling*. That is, after selecting l columns from $\mathbf{M}$, the next column (or next several columns depending on the error type) is sampled according to column norms of a *projected* matrix $\mathcal{P}_{\mathcal{C}^\perp}(\mathbf{M})$, where $\mathcal{C}$ is the subspace

spanned by currently selected columns. It can be shown that iterative norm sampling serves as an approximation of *volume sampling*, a sampling scheme that is known to have relative error guarantees (Deshpande et al., 2006; Deshpande and Vempala, 2006).

Theorem 2 shows that when the input matrix $\mathbf{M}$ is the sum of an exact low rank matrix $\mathbf{A}$ and a stochastic noise matrix $\mathbf{R}$, then by selecting exact k columns from $\mathbf{M}$ using iterative norm sampling one can upper bound the selection error $\|\mathbf{M} - \mathbf{C}\mathbf{C}^{\dagger}\mathbf{M}\|_F$ by the best rank-k approximation error $\|\mathbf{M} - \mathbf{M}_k\|_F$ within a multiplicative factor that does not depend on the matrix size n. Such relative error guarantee is much stronger than the additive error bound provided in Theorem 1 as when $\mathbf{M}$ is exactly low rank the error is eliminated with high probability. In fact, when the input matrix $\mathbf{M}$ is exactly low rank the first phase of the proposed algorithm (Line 1 to Line 9 in Algorithm 2) resembles the adaptive sampling algorithm proposed in (Krishnamurthy and Singh, 2013, 2014) for matrix and tensor completion in the sense that at each iteration all columns falling exactly onto the span of already selected columns will have zero norm after projection and hence will never be sampled again. However, we are unable to generalize our algorithm to general full-rank inputs because it is difficult to bound the incoherence level of projected columns (and hence the projection accuracy itself later on) without a stochastic noise model. We present a new algorithm with slightly worse error bounds in Section 3.3 which can handle general high-rank inputs.

Though Eq. (10) is a relative error bound, the multiplicative factor scales exponentially with the intrinsic rank k, which is not completely satisfactory. As a remedy, we show that by slightly over-sampling the columns ($\Theta(k^2 \log k + k/\epsilon\delta)$ instead of k columns) the selection error as well as the $\mathbf{CX}$ reconstruction error could be upper bounded by $\|\mathbf{M} - \mathbf{M}_k\|_F$ within only a $(1 + 3\epsilon)$ factor, which implies that the error bounds are nearly optimal when the number of selected columns s is sufficiently large, for example, $s = \Omega(k^2 \log k + k/\epsilon\delta)$.

Theorem 2 *Fix $\delta > 0$. Suppose $\mathbf{M} = \mathbf{A} + \mathbf{R}$, where $\mathbf{A}$ is a rank-k deterministic matrix with incoherent column space (i.e., $\mu(\mathcal{U}(\mathbf{A})) \le \mu_0$) and $\mathbf{R}$ is a random matrix with i.i.d. zero-mean Gaussian distributed entries. Suppose $k = O(n_1/\log(n_2/\delta))$. Let $\mathbf{C}, \mathbf{S}$ and $\mathbf{X}$ be the output of Algorithm 2. Then the following holds:*

1. *If $m = \Omega(k^2 \mu_0 \log^2(n/\delta))$ then with probability $\ge 1 - \delta$*

$$\|\mathbf{M} - \mathbf{C}\mathbf{C}^{\dagger}\mathbf{M}\|_F^2 \le \frac{2.5^k (k+1)!}{\delta} \|\mathbf{R}\|_F^2. \tag{10}$$

   The column subset size is k and the corresponding sample complexity is $\Omega(k^2 \mu_0 n \log^2(n/\delta))$.

2. *If $m = \Omega(\epsilon^{-1} s \mu_0 \log^2(n/\delta))$ with $s = \Theta(k^2 \log k + k/\epsilon\delta)$, then with probability $\ge 1 - \delta$*

$$\|\mathbf{M} - \mathbf{S}\mathbf{S}^{\dagger}\mathbf{M}\|_F^2 \le \|\mathbf{M} - \mathbf{S}\mathbf{X}\|_F^2 \le (1 + 3\epsilon)\|\mathbf{R}\|_F^2. \tag{11}$$

   The column subset size is $\Theta(k^2 \log k + k/\epsilon\delta)$ and the sample complexity is (omitting dependence on δ)

$$\Omega\left(\frac{k^2 \mu_0 n \log k \log^2(n)}{\epsilon} + \frac{k \mu_0 n \log^2(n)}{\epsilon^2}\right).$$

Algorithm 3 Approximate leverage score sampling for column subset selection on general input matrices

1: **Input**: target rank k, size of column subset s, expected number of row samples m.
2: **Leverage score estimation**: Set $\mathcal{S} = \emptyset$.
- For each row i, with probability m/n_1 observe the row $\mathbf{M}_{(i)}$ in full and update $\mathcal{S} \leftarrow \text{span}(\mathcal{S}, \{\mathbf{M}_{(i)}\})$.
- Compute the first k right singular vectors of $\mathcal{S}$ (denoted by $\mathbf{S}_k \in \mathcal{R}^{n_2 \times k}$) and estimate the unnormalized row space leverage scores as $\tilde{l}_j = \|\mathbf{S}_k{}^{\top} \boldsymbol{e}_j\|_2^2$, $j \in \{1, 2, \cdots, n_1\}$.

3: **Column subset selection**: Set $C = \emptyset$.
- For $t \in \{1, 2, \cdots, s\}$ select a column $i_t \in [n_2]$ with probability $Pr[i_t = j] = \hat{p}_j = \tilde{l}_j/k$; update $C \leftarrow C \cup \{i_t\}$.

4: **Output**: the selected column indices $C \subseteq \{1, 2, \cdots, n_2\}$ and actual columns $\mathbf{C} = (\mathbf{M}^{(C(1))}, \cdots, \mathbf{M}^{(C(s))})$.

3.3 Approximate leverage score sampling

The third sampling-based column subset selection algorithm for partially observed matrices is presented in Algorithm 3. The proposed algorithm was based on the leverage score sampling scheme for matrix CUR approximation introduced in (Drineas et al., 2008). To compute the sampling distribution (i.e., leverage scores) from partially observed data, the algorithm subsamples a small number of rows from the input matrix and uses leverage scores of the row space of the subsampled matrix to form the sampling distribution. Note that we do not attempt to approximate leverage scores of the original input matrix directly; instead, we compute leverage scores of another matrix that is a good approximation of the original data. Such technique was also explored in (Drineas et al., 2012) to approximate statistical leverages in a fully observed setting. Afterwards, column sampling distribution is constructed using the estimated leverage scores and a subset of columns are selected according to the constructed sampling distribution.

We bound the selection error $\|\mathbf{M} - \mathbf{C}\mathbf{C}^{\dagger}\mathbf{M}\|_F$ of the approximate leverage score sampling algorithm in Theorem 3. Note that unlike Theorem 1 and 2, only selection error bound is provided since for deterministic full-rank input matrices it is challenging to approximately compute the projection of $\mathbf{M}$ onto span($\mathbf{C}$) because the projected vector may no longer be incoherent (this is in fact the reason why Theorem 2 holds only for low-rank matrices perturbed by Gaussian noise, and we believe similar conclusion should also hold for Algorithm 3 is the stronger assumption of Gaussian noise perturbation is made). It remains an open problem to approximately compute $\mathbf{C}^{\dagger}\mathbf{M}$ given $\mathbf{C}$ with provable guarantee for general matrix $\mathbf{M}$ without observing it in full. Eq. (3) shows that Algorithm 3 enjoys a relative error bound on the selection error. In fact, when the input matrix $\mathbf{M}$ is exactly low rank then Algorithm 3 is akin to the two-step matrix completion method proposed in (Chen et al., 2013) for column incoherent inputs.

Although Theorem 3 shows that Algorithm 3 generalizes the relative selection error bound in Theorem 2 to general input matrices, it also reveals several drawbacks of the

approximate leverage score sampling algorithm compared to the iterative norm sampling method. First, Algorithm 3 always needs to over-sample columns (at the level of $\Theta(k^2/\epsilon^2)$, which is even more than Algorithm 2 for a $(1+\epsilon)$ reconstruction error bound); in contrast, the iterative norm sampling algorithm only requires exact k selected columns to guarantee a relative error bound. In addition, Eq. (12) shows that the selection error bound is suboptimal even if s is sufficiently large because of the $(3+3\epsilon)$ multiplicative term.

Theorem 3 *Suppose* $\mathbf{M}$ *is an input matrix with incoherent top-k column space (i.e.,* $\mu(\mathcal{U}_k(\mathbf{M})) \leq \mu_0$*) and* C *is the column indices output by Algorithm 3. If* $m = \Omega(\epsilon^{-2}\mu_0 k^2 \log(1/\delta))$ *and* $s = \Omega(\epsilon^{-2}k^2\log(1/\delta))$ *then with probability* $\geq 1-\delta$ *the following holds:*

$$\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_F \leq 3(1+\epsilon)\|\mathbf{M}-\mathbf{M}_k\|_F, \tag{12}$$

where $\mathbf{C} = [\mathbf{M}^{(C(1))}, \cdots, \mathbf{M}^{(C(s))}] \in \mathbb{R}^{n_1 \times s}$ *are the selected columns and* $\mathbf{M}_k$ *is the best rank-k approximation of* $\mathbf{M}$*.*

4. Proofs

In this section we provide proof sketches of the main results (Theorem 1, 2 and 3). Some technical lemmas and complete proof details are deferred to Appendix A and B.

4.1 Proof sketch of Theorem 1

The proof of Theorem 1 can be divided into two steps. First, in Lemma 1 we show that (approximate) column sampling yields an additive error bound for column subset selection. Its proof is very similar to the one presented in (Frieze et al., 2004) and we defer it to Appendix A. Second, we cite a lemma from (Krishnamurthy and Singh, 2014) to show that with high probability the first pass in Algorithm 1 gives accurate estimates of column norms of the input matrix $\mathbf{M}$.

Lemma 1 *Provided that* $(1-\alpha)\|\boldsymbol{x}_i\|_2^2 \leq \hat{c}_i \leq (1+\alpha)\|\boldsymbol{x}_i\|_2^2$ *for* $i = 1, 2, \cdots, n_2$*, with probability* $\geq 1-\delta$ *we have*

$$\|\mathbf{M} - \mathcal{P}_C(\mathbf{M})\|_F \leq \|\mathbf{M}-\mathbf{M}_k\|_F + \sqrt{\frac{(1+\alpha)k}{(1-\alpha)\delta s}}\|\mathbf{M}\|_F, \tag{13}$$

where $\mathbf{M}_k$ *is the best rank-k approximation of* $\mathbf{M}$*.*

Lemma 2 ((Krishnamurthy and Singh, 2014), Lemma 10) *Fix* $\alpha, \delta \in (0,1)$*. Assume* $\mu(\boldsymbol{x}_i) \leq \mu_0$ *holds for* $i = 1, 2, \cdots, n_2$*. For some fixed* $i \in \{1, \cdots, n_2\}$ *with probability* $\geq 1-2\delta$ *we have*

$$(1-\alpha)\|\boldsymbol{x}_i\|_2^2 \leq \hat{c}_i \leq (1+\alpha)\|\boldsymbol{x}_i\|_2^2 \tag{14}$$

with $\alpha = \sqrt{\frac{2\mu_0}{m_1}\log(1/\delta)} + \frac{2\mu_0}{3m_1}\log(1/\delta)$*. Furthermore, if* $m_1 = \Omega(\mu_0 \log(n_2/\delta))$ *with carefully chosen constants then Eq. (14) holds uniformly for all columns with* $\alpha = 0.5$*.*

Combining Lemma 1 and Lemma 2 and setting $s = \Omega(k\epsilon^{-2}/\delta)$ for some target accuracy threshold ϵ we have that with probability $1-3\delta$ the selection error bound Eq. (8) holds.

In order to bound the reconstruction error $\|\mathbf{M} - \mathbf{C}\mathbf{X}\|_F^2$, we cite another lemma from (Krishnamurthy and Singh, 2014) that analyzes the performance of the second pass of Algorithm 1. At a higher level, Lemma 3 is a consequence of matrix Bernstein inequality (Tropp, 2012) which asserts that the *spectral norm* of a matrix can be preserved by a sum of properly scaled randomly sampled sub-matrices.

Lemma 3 ((Krishnamurthy and Singh, 2014), Lemma 9) *Provided that* $(1-\alpha)\|\boldsymbol{x}_i\|_2^2 \leq \hat{c}_i \leq (1+\alpha)\|\boldsymbol{x}_i\|_2^2$ *for* $i = 1, 2, \cdots, n_2$*, with probability* $\geq 1-\delta$ *we have*

$$\|\mathbf{M} - \widehat{\mathbf{M}}\|_2 \leq \|\mathbf{M}\|_F \sqrt{\frac{1+\alpha}{1-\alpha}} \left(\frac{4}{3}\sqrt{\frac{n_1\mu_0}{m_2 n_2}} \log\left(\frac{n_1+n_2}{\delta}\right) + \sqrt{\frac{4}{m_2}\max\left(\frac{n_1}{n_2}, \mu_0\right)\log\left(\frac{n_1+n_2}{\delta}\right)} \right). \tag{15}$$

The complete proof of Theorem 1 is deferred to Appendix A.

4.2 Proof sketch of Theorem 2

In this section we give proof sketch of Eq. (10) and Eq. (11) separately.

4.2.1 Proof sketch of $\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_F$ error bound

We take three steps to prove the $\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_F$ error bound in Theorem 2. At the first step, we show that when the input matrix has a low rank plus noise structure then with high probability *for all small subsets of columns* the spanned subspace has incoherent column space (assuming the low-rank matrix has incoherent column space) and furthermore, the projection of the other columns onto the orthogonal complement of the spanned subspace are incoherent, too. Given the incoherence condition we can easily prove a norm estimation result similar to Lemma 2, which is the second step. Finally, we note that the approximate iterative norm sampling procedure is an approximation of *volume sampling*, a column sampling scheme that is known to yield a relative error bound.

STEP 1: We first prove that when the input matrix $\mathbf{M}$ is a noisy low-rank matrix with incoherent column space, with high probability *a fixed* column subset also has incoherent column space. This is intuitive because the Gaussian perturbation matrix is highly incoherent with overwhelming probability. A more rigorous statement is shown in Lemma 4.

Lemma 4 *Suppose* $\mathbf{A}$ *has incoherent column space, i.e.,* $\mu(\mathcal{U}(\mathbf{A})) \leq \mu_0$*. Fix* $C \subseteq [n_2]$ *to be any subset of column indices that has* s *elements and* $\delta > 0$*. Let* $\mathbf{C} = [\mathbf{M}^{(C(1))}, \cdots, \mathbf{M}^{(C(s))}] \in \mathbb{R}^{n_1 \times s}$ *be the compressed matrix and* $\mathcal{U}(C) = \mathrm{span}(\mathbf{C})$ *denote the subspace spanned by the selected columns. Suppose* $\max(s,k) \leq n_1/4 - k$ *and* $\log(4n_2/\delta) \leq n_1/64$*. Then with probability* $\geq 1-\delta$ *over the random drawn of* $\mathbf{R}$ *we have*

$$\mu(\mathcal{U}(C)) = \frac{n_1}{s}\max_{1\leq i\leq n_1}\|\mathcal{P}_{\mathcal{U}(C)}\boldsymbol{e}_i\|_2^2 = O\left(\frac{k\mu_0 + s + \sqrt{s\log(n_1/\delta)} + \log(n_1/\delta)}{s}\right); \tag{16}$$

furthermore, with probability $\geq 1-\delta$ *the following holds:*

$$\mu(\mathcal{P}_{\mathcal{U}(C)^\perp}(\mathbf{M}^{(i)})) = O(k\mu_0 + \log(n_1 n_2/\delta)), \quad \forall i \notin C. \tag{17}$$

At a higher level, Lemma 4 is a consequence of the Gaussian white noise being highly incoherent, and the fact that the randomness imposed on each column of the input matrix is independent. The complete proof can be found in Appendix B.

Given Lemma 4, Corollary 1 holds by taking a uniform bound over all $\sum_{j=1}^{s}\binom{n_2}{j} = O(s(n_2)^s)$ column subsets that contain no more than s elements. The $2s\log(4n_2/\delta) \leq n_1/64$ condition is only used to ensure that the desired failure probability δ is not exponentially small. Typically, in practice the intrinsic dimension k and/or the target column subset size s is much smaller than the ambient dimension n_1.

Corollary 1 *Fix* $\delta > 0$ *and* $s \geq k$. *Suppose* $s \leq n_1/8$ *and* $2s\log(4n_2/\delta) \leq n_1/64$. *With probability* $\geq 1-\delta$ *the following holds: for any subset* $C \subseteq [n_2]$ *with at most* s *elements, the spanned subspace* $\mathcal{U}(C)$ *satisfies*

$$\mu(\mathcal{U}(C)) \leq O((k+s)|C|^{-1}\mu_0 \log(n/\delta)); \tag{18}$$

furthermore,

$$\mu(\mathcal{P}_{\mathcal{U}(C)^\perp}(\mathbf{M}^{(i)})) = O((k+s)\mu_0 \log(n/\delta)), \quad \forall i \notin C. \tag{19}$$

STEP 2: In this step, we prove that the norm estimation scheme in Algorithm 2 works when the incoherence conditions in Eq. (18) and (19) are satisfied. More specifically, we have the following lemma bounding the norm estimation error:

Lemma 5 *Fix* $i \in \{1, \cdots, n_2\}$, $t \in \{1, \cdots, k\}$ *and* $\delta, \delta' > 0$. *Suppose Eq. (18) and (19) hold with probability* $\geq 1-\delta$. *Let* $\mathcal{S}_t$ *be the subspace spanned by selected columns at the* t*-th round and let* $\hat{c}_i^{(t)}$ *denote the estimated squared norm of the* i*th column. If* m *satisfies*

$$m = \Omega(k\mu_0 \log(n/\delta)\log(k/\delta')), \tag{20}$$

then with probability $\geq 1-\delta-4\delta'$ *we have*

$$\frac{1}{2}\|[\mathbf{E}_t]_{(i)}\|_2^2 \leq \hat{c}_i^{(t)} \leq \frac{5}{4}\|[\mathbf{E}_t]_{(i)}\|_2^2. \tag{21}$$

Here $\mathbf{E}_t = \mathcal{P}_{\mathcal{S}_t^\perp}(\mathbf{M})$ *denotes the projected matrix at the* t*-th round.*

Lemma 5 is similar with previous results on subspace detection (Balzano et al., 2010b) and matrix approximation (Krishnamurthy and Singh, 2014). The intuition behind Lemma 5 is that one can accurately estimate the ℓ_2 norm of a vector by uniform subsampling entries of the vector, provided that the vector itself is incoherent. The proof of Lemma 5 is deferred to Appendix B.

Similar to the first step, by taking a union bound over all possible subsets of picked columns and $n_2 - k$ unpicked columns we can prove a stronger version of Lemma 5, as shown in Corollary 2.

Corollary 2 *Fix* $\delta, \delta' > 0$. *Suppose Eq. (18) and (19) hold with probability* $\geq 1-\delta$. *If*

$$m \geq \Omega(k^2\mu_0 \log(n/\delta)\log(n/\delta')) \tag{22}$$

then with probability $\geq 1-\delta-4\delta'$ *the following property holds for any selected column subset by Algorithm 2:*

$$\frac{2}{5}\frac{\|[\mathbf{E}_t]_{(i)}\|_2^2}{\|\mathbf{E}_t\|_F^2} \leq \hat{p}_i^{(t)} \leq \frac{5}{2}\frac{\|[\mathbf{E}_t]_{(i)}\|_2^2}{\|\mathbf{E}_t\|_F^2}, \forall i \in [n_2], t \in [k], \tag{23}$$

where $\hat{p}_i^{(t)} = \hat{c}_i^{(t)}/\hat{f}^{(t)}$ *is the sampling probability of the* i*th column at round* t.

STEP 3: To begin with, we define *volume sampling* distributions:

Definition 1 (volume sampling, (Deshpande et al., 2006)) *A distribution* p *over column subsets of size* k *is a volume sampling distribution if*

$$p(C) = \frac{\mathrm{vol}(\Delta(C))^2}{\sum_{T:|T|=k}\mathrm{vol}(\Delta(T))^2}, \quad \forall |C| = k. \tag{24}$$

Volume sampling has been shown to achieve a relative error bound for column subset selection, which is made precise by Theorem 4 cited from (Deshpande and Vempala, 2006; Deshpande et al., 2006).

Theorem 4 ((Deshpande and Vempala, 2006), Theorem 4) *Fix a matrix* $\mathbf{M}$ *and let* $\mathbf{M}_k$ *denote the best rank-k approximation of* $\mathbf{M}$. *If the sampling distribution* p *is a volume sampling distribution defined in Eq. (24) then*

$$\mathbb{E}_C\left[\|\mathbf{M} - \mathcal{P}_{\mathcal{V}(C)}(\mathbf{M})\|_F^2\right] \leq (k+1)\|\mathbf{M} - \mathbf{M}_k\|_F^2; \tag{25}$$

furthermore, applying Markov's inequality one can show that with probability $\geq 1-\delta$

$$\|\mathbf{M} - \mathcal{P}_{\mathcal{V}(C)}(\mathbf{M})\|_F^2 \leq \frac{k+1}{\delta}\|\mathbf{M} - \mathbf{M}_k\|_F^2. \tag{26}$$

In general, exact volume sampling is difficult to employ under partial observation settings, as we explained in Sec. 2.2. However, in (Deshpande and Vempala, 2006) it was shown that iterative norm sampling serves as an approximate of volume sampling and achieves a relative error bound as well. In Lemma 6 we present an extension of this result. Namely, *approximate* iterative column norm sampling is an approximate of volume sampling, too. Its proof is very similar to the one presented in (Deshpande and Vempala, 2006) and we defer it to Appendix B.

Lemma 6 *Let* p *be the volume sampling distribution defined in Eq. (24). Suppose the sampling distribution of a* k*-round sampling strategy* $\hat{p}$ *satisfies Eq. (23). Then we have*

$$\hat{p}_C \leq 2.5^k k! p_C, \quad \forall |C| = k. \tag{27}$$

We can now prove the error bound for selection error $\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_F$ of Algorithm 2 by combining Corollary 1, 2, Lemma 6 and Theorem 4, with failure probability δ, δ' set at $O(1/k)$ to facilitate a union bound argument across all iterations. In particular, Corollary 1 and 2 guarantees that Algorithm 2 estimates column norms accurately with high probability; then one can apply Lemma 6 to show that the sampling distribution employed in the algorithm is actually an approximate volume sampling distribution, which is known to achieve relative error bounds (by Theorem 4).

4.2.2 Proof sketch of $\|\mathbf{M} - \mathbf{S}\mathbf{X}\|_F$ error bound

We first present a theorem, which is a generalization of Theorem 2.1 in (Deshpande et al., 2006).

Theorem 5 ((Deshpande et al., 2006), Theorem 2.1) *Suppose $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$ is the input matrix and $\mathcal{U} \subseteq \mathbb{R}^{n_1}$ is an arbitrary vector space. Let $\mathbf{S} \in \mathbb{R}^{n_1 \times s}$ be a random sample of s columns in $\mathbf{M}$ from a distribution q such that*

$$\frac{(1-\alpha)\|\mathbf{E}^{(i)}\|_2^2}{(1+\alpha)\|\mathbf{E}\|_F^2} \leq q_i \leq \frac{(1+\alpha)\|\mathbf{E}^{(i)}\|_2^2}{(1-\alpha)\|\mathbf{E}\|_F^2}, \quad \forall i \in \{1, 2, \cdots, n_2\}, \tag{28}$$

where $\mathbf{E} = \mathcal{P}_{\mathcal{U}^\perp}(\mathbf{M})$ is the projection of $\mathbf{M}$ onto the orthogonal complement of $\mathcal{U}$. Then

$$\mathbb{E}_S\left[\|\mathbf{M} - \mathcal{P}_{\mathrm{span}(\mathcal{U},\mathbf{S}),k}(\mathbf{M})\|_F^2\right] \leq \|\mathbf{M} - \mathbf{M}_k\|_F^2 + \frac{(1+\alpha)k}{(1-\alpha)s}\|\mathbf{E}\|_F^2, \tag{29}$$

where $\mathbf{M}_k$ denotes the best rank-k approximation of $\mathbf{M}$.

Intuitively speaking, Theorem 5 states that relative estimation of residues $\mathcal{P}_{\mathcal{U}^\perp}(\mathbf{M})$ would yield relative estimation of the data matrix $\mathbf{M}$ itself.

In the remainder of the proof we assume $s = \Omega(k^2\log(k) + k/\epsilon\delta)$ is the number of columns selected in $\mathbf{S}$ in Algorithm 2. Corollary 1 asserts that with high probability $\mu(\mathcal{U}(\mathbf{S})) = O(s|C|^{-1}\mu_0\log(n/\delta))$ and $\mu(\mathcal{P}_{\mathcal{U}(\mathbf{S})^\perp}(\mathbf{M}^{(i)})) = O(s\mu_0\log(n/\delta))$ for any subset S with $|S| \leq s$. Subsequently, we can apply Lemma 5 and a union bound over n_2 columns and T rounds to obtain the following proposition:

Proposition 1 *Fix $\delta, \delta' > 0$. If $m = \Omega(s\mu_0\log(n/\delta)\log(nT/\delta'))$ then with probability $\geq 1 - \delta - \delta'$*

$$\frac{2\|\mathbf{E}_t^{(i)}\|_2^2}{5\|\mathbf{E}_t\|_F^2} \leq \hat{q}_i^{(t)} \leq \frac{5\|\mathbf{E}_t^{(i)}\|_2^2}{2\|\mathbf{E}_t\|_F^2}, \quad \forall i \in \{1, 2, \cdots, n_2\}, t \in \{1, 2, \cdots, T\}. \tag{30}$$

Here $\mathbf{E}_t = \mathbf{M} - \mathcal{P}_{\mathrm{span}(\mathcal{U}\cup\mathcal{S}_1\cup\cdots\cup\mathcal{S}_{t-1})}(\mathbf{M})$ is the residue at round t of the active norm sampling procedure.

Note that we do not need to take a union bound over all $\binom{n_2}{s}$ column subsets because this time we do not require the sampling distribution of Algorithm 2 to be close uniformly to the true active norm sampling procedure.

Consequently, combining Theorem 5 and Proposition 1 we obtain Lemma 7. Its proof is deferred to Appendix B.

Lemma 7 *Fix $\delta, \delta' > 0$. If $m = \Omega(s\mu_0\log^2(n/\delta))$ and $s_1 = \cdots = s_{T-1} = 5k$, $s_T = 10k/\epsilon\delta'$ then with probability $\geq 1 - 2\delta - \delta''$*

$$\|\mathbf{M} - \mathcal{P}_{\mathcal{U}\cup\mathcal{S}_1\cup\cdots\cup\mathcal{S}_T}(\mathbf{M})\|_F^2 \leq (1+\epsilon/2)\|\mathbf{M} - \mathbf{M}_k\|_F^2 + \frac{\epsilon/2}{2^T}\|\mathbf{M} - \mathcal{P}_{\mathcal{U}}(\mathbf{M})\|_F^2. \tag{31}$$

Applying Theorem 4, Lemma 6 and note that $2^{(k+1)\log(k+1)} = (k+1)^{(k+1)} \geq (k+1)!$, we immediately have Corollary 3.

Corollary 3 *Fix $\delta > 0$. Suppose $T = (k+1)\log(k+1)$ and $m, s_1, \cdots, s_T$ be set as in Lemma 7. Then with probability $\geq 1 - 4\delta$ one has*

$$\|\mathbf{M} - \mathbf{S}\mathbf{S}^\dagger\mathbf{M}\|_F^2 = \|\mathbf{M} - \mathcal{P}_{\mathcal{U}\cup\mathcal{S}_1\cup\cdots\cup\mathcal{S}_T}(\mathbf{M})\|_F^2 \leq (1+\epsilon)\|\mathbf{M} - \mathbf{M}_k\|_F^2 \leq (1+\epsilon)\|\mathbf{R}\|_F^2. \tag{32}$$

To reconstruct the coefficient matrix $\mathbf{X}$ and to further bound the reconstruction error $\|\mathbf{M} - \mathbf{S}\mathbf{X}\|_F$, we apply the $\mathbf{U}(\mathbf{U}_\Omega^\top\mathbf{U}_\Omega)^{-1}\mathbf{U}_\Omega$ operator on every column to build a low-rank approximation $\widehat{\mathbf{M}}$. It was shown in (Krishnamurthy and Singh, 2013; Balzano et al., 2010b) that this operator recovers all components in the underlying subspace $\mathcal{U}$ with high probability, and hence achieves a relative error bound for low-rank matrix approximation. More specifically, we have Lemma 8, which is proved in Appendix B.

Lemma 8 *Fix $\delta, \delta'' > 0$ and $\epsilon > 0$. Let $\mathbf{S} \in \mathbb{R}^{n_1 \times s}$ and $\mathbf{X} \in \mathbb{R}^{s \times n_2}$ be the output of Algorithm 2. Suppose Corollary 3 holds with probability $\geq 1 - \delta$. If m satisfies*

$$m = \Omega(\epsilon^{-1}s\mu_0\log(n/\delta)\log(n/\delta'')), \tag{33}$$

then with probability $\geq 1 - \delta - \delta''$ we have

$$\|\mathbf{M} - \widehat{\mathbf{M}}\|_F^2 \leq (1+\epsilon)\|\mathbf{M} - \mathbf{S}\mathbf{S}^\dagger\mathbf{M}\|_F^2. \tag{34}$$

Note that all columns of $\widehat{\mathbf{M}}$ are in the subspace $\mathcal{U}(S)$. Therefore, $\mathbf{S}\mathbf{X} = \mathbf{S}\mathbf{S}^\dagger\widehat{\mathbf{M}} = \widehat{\mathbf{M}}$. The proof of Eq. (11) is then completed by noting that $(1+\epsilon)^2 \leq 1 + 3\epsilon$ whenever $\epsilon \leq 1$.

4.3 Proof of Theorem 3

Before presenting the proof, we first present a theorem cited from (Drineas et al., 2008). In general, Theorem 6 claims that if columns are selected with probability proportional to their row-space leverage scores then the resulting column subset is a relative-error approximation of the original input matrix.

Theorem 6 ((Drineas et al., 2008), Theorem 3) *Let $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$ be the input matrix and k be a rank parameter. Suppose a subset of columns $C = \{i_1, i_2, \cdots, i_s\} \subseteq [n_2]$ is selected such that*

$$\Pr[i_t = j] = p_j \geq \frac{\beta\|\mathbf{V}_k^\top\boldsymbol{e}_j\|_2^2}{k}, \quad \forall t \in \{1, \cdots, s\}, j \in \{1, \cdots, n_2\}. \tag{35}$$

Here $\mathbf{V}_k \in \mathbb{R}^{n_2 \times k}$ *is the top-*k *right singular vectors of* $\mathbf{M}$. *If* $s = \Omega(\beta^{-1}\epsilon^{-2}k^2\log(1/\delta))$ *then with probability* $\geq 1-\delta$ *one has*

$$\|\mathbf{M} - \mathbf{C}\mathbf{C}^{\dagger}\mathbf{M}\|_F \leq (1+\epsilon)\|\mathbf{M} - \mathbf{M}_k\|_F. \tag{36}$$

In the sequel we use $\mathcal{Q}_S(\mathbf{M})$ to denote the matrix formed by projecting each row of $\mathbf{M}$ to a row subspace $\mathcal{S}$ and $\mathcal{P}_C(\mathbf{M})$ to denote the matrix formed by projecting each column of $\mathbf{M}$ to a column subspace $\mathcal{C}$. Since $\mathbf{M}$ has incoherent column space, the uniform sampling distribution $p_j = 1/n_1$ satisfies Eq. (35) with $\beta = 1/\mu_0$. Consequently, by Theorem 6 the computed row space $\mathcal{S}$ satisfies

$$\|\mathbf{M} - \mathcal{Q}_S(\mathbf{M})\|_F \leq (1+\epsilon)\|\mathbf{M} - \mathbf{M}_k\|_F \tag{37}$$

with high probability when $m = \Omega(k^2/\beta\epsilon^2) = \Omega(\mu_0 k^2/\epsilon^2)$.

Next, note that though we do not know $\mathcal{Q}_S(\mathbf{M})$, we know its row space $\mathcal{S}$. Subsequently, we can compute the exact leverage scores of $\mathcal{Q}_S(\mathbf{M})$, i.e., $\|\mathbf{S}_k^{\top}\boldsymbol{e}_j\|_2^2$ for $j = 1, 2, \cdots, n_2$. With the computed leverage scores, performing leverage score sampling on $\mathcal{Q}_S(\mathbf{M})$ as in Algorithm 3 and applying Theorem 6 we obtain

$$\|\mathcal{Q}_S(\mathbf{M}) - \mathcal{P}_C(\mathcal{Q}_S(\mathbf{M}))\|_F \leq (1+\epsilon)\|\mathcal{Q}_S(\mathbf{M}) - [\mathcal{Q}_S(\mathbf{M})]_k\|_F, \tag{38}$$

where $[\mathcal{Q}_S(\mathbf{M})]_k$ denotes the best rank-k approximation of $\mathcal{Q}_S(\mathbf{M})$. Note that

$$\|\mathcal{Q}_S(\mathbf{M}) - [\mathcal{Q}_S(\mathbf{M})]_k\|_F \leq \|\mathcal{Q}_S(\mathbf{M}) - \mathcal{Q}_S(\mathbf{M}_k)\|_F = \|\mathcal{Q}_S(\mathbf{M} - \mathbf{M}_k)\|_F \leq \|\mathbf{M} - \mathbf{M}_k\|_F \tag{39}$$

because $\mathcal{Q}_S(\mathbf{M}_k)$ has rank at most k. Consequently, the selection error $\|\mathbf{M} - \mathcal{P}_C(\mathbf{M})\|_F$ can be bounded as follows:

$$\begin{aligned}
\|\mathbf{M} - \mathcal{P}_C(\mathbf{M})\|_F &\leq \|\mathbf{M} - \mathcal{Q}_S(\mathbf{M})\|_F + \|\mathcal{Q}_S(\mathbf{M}) - \mathcal{P}_C(\mathcal{Q}_S(\mathbf{M}))\|_F + \|\mathcal{P}_C(\mathcal{Q}_S(\mathbf{M})) - \mathcal{P}_C(\mathbf{M})\|_F \\
&\leq \|\mathbf{M} - \mathcal{Q}_S(\mathbf{M})\|_F + \|\mathcal{Q}_S(\mathbf{M}) - \mathcal{P}_C(\mathcal{Q}_S(\mathbf{M}))\|_F + \|\mathcal{Q}_S(\mathbf{M}) - \mathbf{M}\|_F \\
&\leq 3(1+\epsilon)\|\mathbf{M} - \mathbf{M}_k\|_F.
\end{aligned}$$

5. Related work on column subset selection with missing data

In this section we review two previously proposed algorithms for column subset selection with missing data. Both algorithms are heuristic based and no theoretical analysis is available. We also remark that both methods employ the passive sampling scheme as observation models. In fact, they work for any subset of observed matrix entries.

5.1 Block orthogonal matching pursuit (Block OMP)

A block OMP algorithm was proposed in (Balzano et al., 2010a) for column subset selection with missing data. Let $\mathbf{W} \in \{0,1\}^{n_1 \times n_2}$ denote the "mask" of observed entries; that is,

$$\mathbf{W}_{ij} = \begin{cases} 1, & \text{if } \mathbf{M}_{ij} \text{ is observed;} \\ 0, & \text{if } \mathbf{M}_{ij} \text{ is not observed.} \end{cases}$$

Algorithm 4 A block OMP algorithm for column subset selection with missing data

Input: size of column subset s, observation mask $\mathbf{W} \in \{0,1\}^{n_1 \times n_2}$.
Initialization: Set $C = \emptyset$, $\mathcal{C} = \emptyset$, $\mathbf{Y} = \mathbf{W} \circ \mathbf{M}$, $\mathbf{Y}^{(1)} = \mathbf{Y}$.
for $t = 1, 2, \cdots, s$ **do**
  Compute $\mathbf{D} = \mathbf{Y}^{\top}(\mathbf{W} \circ \mathbf{Y}^{(t)})$. Let $\{\boldsymbol{d}_i\}_{i=1}^{n_2}$ be rows of $\mathbf{D}$.
  Column selection: $i_t = \text{argmax}_{1 \leq i \leq n_2}\|\boldsymbol{d}_i\|_2$; update: $C \leftarrow C \cup \{i_t\}$, $\mathcal{C} \leftarrow \text{span}(\mathcal{C}, \mathbf{Y}_{(i_t)})$.
  Back projection: $\mathbf{Y}^{(t+1)} = \mathbf{Y}^{(t)} - \mathcal{P}_{\mathcal{C}}(\mathbf{Y}^{(t)})$.
end for
Output: the selected column indices $C \subseteq \{1, 2, \cdots, n_2\}$.

We also use $\circ$ to denote the Hadamard product (entrywise product) between two matrices of the same dimension.

The pseudocode is presented in Algorithm 4. Note that Algorithm 4 has very similar framework compared with the iterative norm sampling algorithm: both methods select columns in an iterative manner and after each column is selected, the contribution of selected columns is removed from the input matrix by projecting onto the complement of the subspace spanned by selected columns. Nevertheless, there are some major differences. First, in iterative norm sampling we select a column according to their residue norms while in block OMP we base such selection on inner products between the original input matrix and the residue one. In addition, due to the passive sampling nature Algorithm 4 uses the zero-filled data matrix to approximate subspace spanned by selected columns. In contrast, iterative norm sampling computes this subspace exactly by active sampling.

5.2 Group Lasso

The group Lasso formulation was originally proposed in (Bien et al., 2010) as a convex optimization alternative for matrix column subset selection and CUR decomposition for fully-observed matrices. It was briefly remarked in (Balzano et al., 2010a) that group Lasso could be extended to the case when only partial observations are available. In this paper we make such extension precise by proposing the following convex optimization problem:

$$\min_{\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}} \|\mathbf{W} \circ \mathbf{M} - (\mathbf{W} \circ \mathbf{M})\mathbf{X}\|_F^2 + \lambda\|\mathbf{X}\|_{1,2}, \quad s.t.\ \text{diag}(\mathbf{X}) = \mathbf{0}. \tag{40}$$

Here in Eq. (40) $\mathbf{W} \in \{0,1\}^{n_1 \times n_2}$ denotes the mask for observed matrix entries and $\circ$ denotes the Hadamard (entrywise) matrix product. $\|\mathbf{X}\|_{1,2} = \sum_{i=1}^{n_2}\|\mathbf{X}_{(i)}\|_2$ denotes the 1,2-norm of matrix $\mathbf{X}$, which is the sum of ℓ_2 norm of all rows in $\mathbf{X}$. The nonzero rows in the optimal solution $\mathbf{X}$ correspond to the selected columns.

Eq. (40) could be solved using standard convex optimization methods, e.g., proximal gradient descent (Mosci et al., 2010). However, to make Eq. (40) a working column subset selection algorithm one needs to carefully choose the regularization parameter λ so that the resulting optimal solution $\mathbf{X}$ has no more than s nonzero columns. Such selection could be time-consuming and inexact. As a workaround, we implement the solution path algorithm for group Lasso problems in (Yang and Zou, 2014).

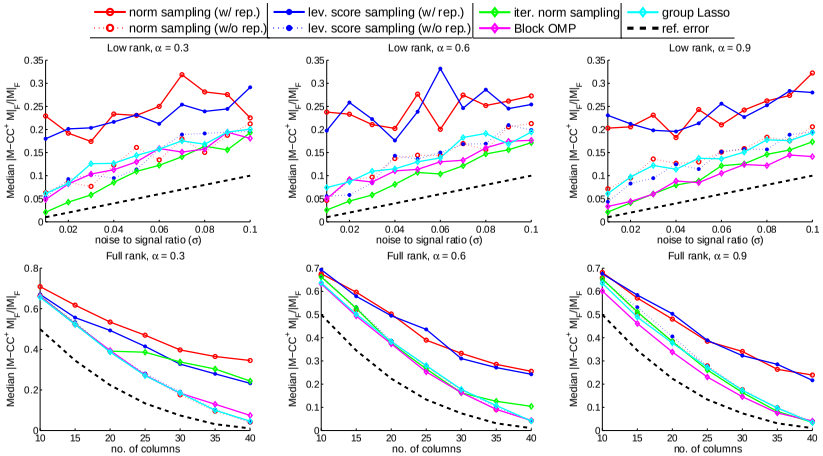


Figure 1: Selection error on Gaussian random matrices. Top row: low-rank plus noise inputs, $s = k = 15$; bottom row: full-rank inputs. The black dashed lines denote noise-to-signal ratio σ in the first row and $\|\mathbf{M} - \mathbf{M}_k\|_F$ in the second row. α indicates the observation rate (i.e., the number of observed entries divided by $n_1 n_2$, the total number of matrix entries). All algorithms are run for 8 times on each data set and the median error is reported. We report the median instead of the mean because the performance of norm and leverage score sampling is quite variable.

5.3 Discussion on theoretical assumptions of block OMP and group Lasso

We discuss theoretical assumptions required for block OMP and group Lasso approaches. It should be noted that for the particular matrix column subset selection problem, neither Balzano et al. (2010a) or Bien et al. (2010) provides rigorous theoretical guarantee of approximation error of the selected column subsets. However, it is informative to compare to typical assumptions that are used to analyze block OMP and group Lasso for regression problems in the existing literature (Yuan and Lin, 2006; Lounici et al., 2011). In most cases, certain "restricted eigenvalue" conditions on the design matrix $\mathbf{X}$, which roughly corresponds to a "weak correlation" condition among columns of a data matrix. This explains the worse performance of both methods on data sets that have highly correlated columns (e.g., many repeated columns), as we shown in later sections on experimental results.

6. Experiments

In this section we report experimental results on both synthetic and real-world data sets for our proposed column subset selection algorithms as well as other competitive methods. All algorithms are implemented in Matlab. To make fair comparisons, all input matrices $\mathbf{M}$ are normalized so that $\|\mathbf{M}\|_F^2 = 1$.

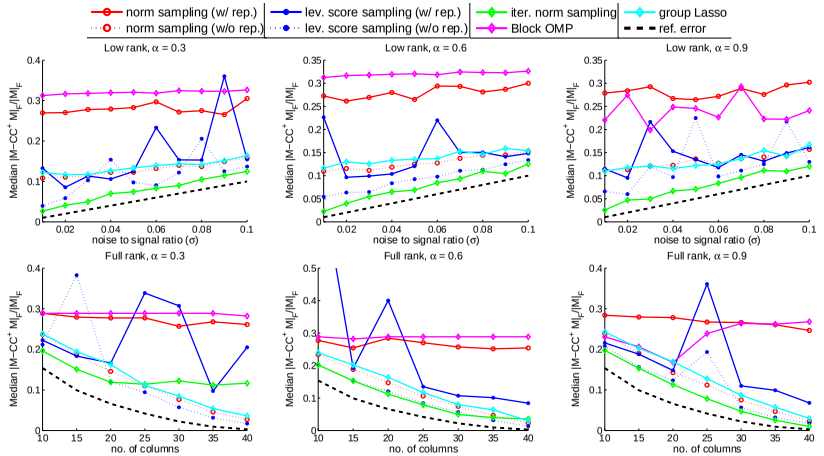


Figure 2: Selection error on matrices with coherent columns. Top row: low-rank plus noise inputs, $s = k = 15$; bottom row: full-rank inputs. α indicates the observation rate. The black dashed lines denote noise-to-signal ratio σ in the first row and $\|\mathbf{M} - \mathbf{M}_k\|_F$ in the second row. All algorithms are run for 8 times on each data set and the median error is reported.

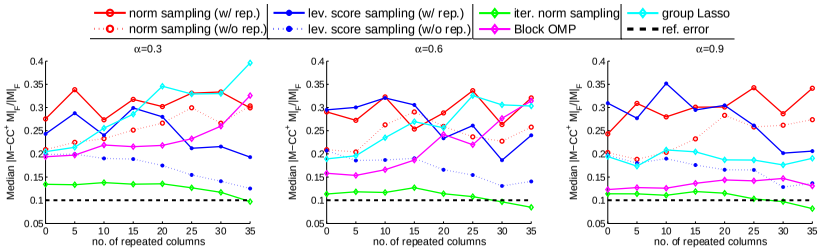


Figure 3: Selection error on matrices with varying number of repeated columns. Both s and k are set to 15 and the noise-to-signal ratio σ is set to 0.1. α indicates the observation rate. All algorithms are run for 8 times on each data set and the median error is reported.

6.1 Synthetic data sets

We first test the proposed algorithms on synthetic data sets. The input matrix has dimension $n_1 = n_2 = n = 50$. To generate the synthetic data, we consider two different settings listed below:

1. **Random Gaussian matrices**: for random Gaussian matrices each entry $\mathbf{M}_{ij}$ are i.i.d. sampled from a normal distribution $\mathcal{N}(0, 1)$. For low rank matrices, we first

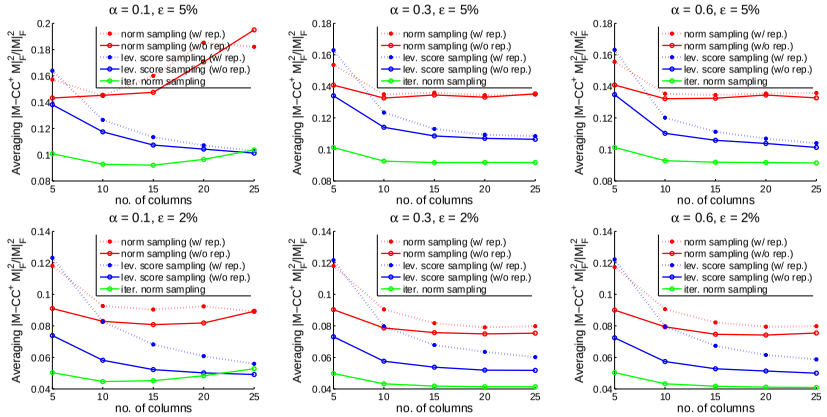


Figure 4: Selection error or sampling based algorithm on Hapmap phase2 data set. α indicates the observation rate. Top row: top-k PCA captures 95% variance within each SNP window; bottom row: top-k PCA captures 98% variance within each SNP window.

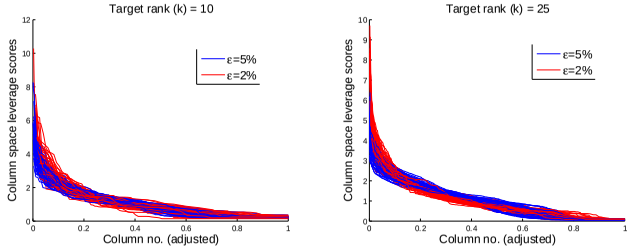


Figure 5: Sorted column space leverage scores for different ε and k settings. For each setting 50 windows are picked at random and their leverage scores are plotted. Each plotted line is properly scaled along the X axis so that they have the same length even though actual window sizes vary.

generate a random Gaussian matrix $\mathbf{B} \in \mathbb{R}^{n\times k}$ where k is the intrinsic rank and then form the data matrix $\mathbf{M}$ as $\mathbf{M} = \mathbf{B}\mathbf{B}^\top$. I.i.d. Gaussian noise $\mathbf{R}$ with $\mathbf{R}_{ij} \sim \mathcal{N}(0, \sigma^2)$ is then appended to the synthesized low-rank matrix. We remark that data matrices generated in this manner have both incoherent column and row space with high probability.

2. **Matrices with coherent columns**: we took a simple procedure to generate matrices with coherent columns in order to highlight the power of proposed algorithms and baseline methods. After generating a random Gaussian matrix $\mathbf{M} = \mathbf{B}\mathbf{B}^\top$, we pick

(a) The 512×512 8-bit gray scale Lena test image before compression.

(b) Norm sampling (without replacement). Selection error $\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_F/\|\mathbf{M}\|_F = 0.106$.

(c) Iterative norm sampling. Selection error $\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_F/\|\mathbf{M}\|_F = 0.088$.

(d) Approx. leverage score sampling (without replacement). $\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_F/\|\mathbf{M}\|_F = 0.103$.

Figure 6: Column-based image compression results on the Lena standard test image. Left: actively sampled image pixels; middle: the selected columns; right: the reconstructed images. Number of selected columns is set to 50 and the pixel subsampling rate α is set to 0.3.

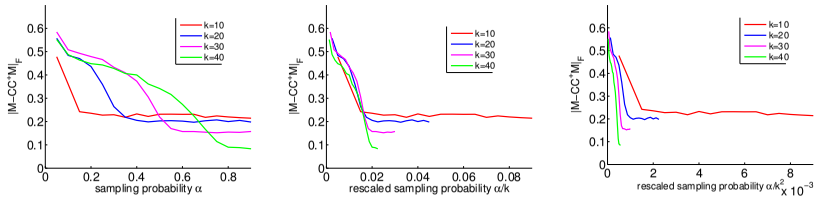


Figure 7: Selection error $\|\mathbf{M} - \mathbf{C}\mathbf{C}^{\dagger}\mathbf{M}\|_F$ for the iterative norm sampling algorithm as a function of α (left), α/k (middle) and α/k^2 (right). Error curves plotted under 4 different rank (k) settings.

a column $\boldsymbol{x}$ from $\mathbf{M}$ uniformly at random. We then take $\tilde{\boldsymbol{x}} = 10\boldsymbol{x}$ and repeat the column for 5 times. As a result, the newly formed data matrix will have 5 identical columns with significantly higher norms compared to the other columns.

In Figure 1 we report the selection error $\|\mathbf{M} - \mathbf{C}\mathbf{C}^{\dagger}\mathbf{M}\|_F$ of proposed and baseline algorithms on random Gaussian matrices and in Figure 2 we report the same results on matrices with coherent columns. Results on both low-rank plus noise and high-rank inputs are reported. For low-rank matrices, both the intrinsic rank k and the number of selected columns s are set to 15. Each algorithm is run for 8 times on the same input and the median selection error is reported. For norm sampling and approximate leverage score sampling, we implement two variants: in the *sampling with replacement* scheme the algorithm samples each column from a sampling distribution (based on either norm or leverage score estimation) with replacement; while in the *sampling without replacement* scheme a column is never sampled twice. Note that all theoretical results in Section 3 are proved for sampling with replacement algorithms.

From Figure 1 we observe that all algorithms perform similarly, with the exception of two sampling with replacement algorithms and iterative norm sampling when both rank and missing rate are high. [3] For the latter case, we conjecture that the degradation of performance is due to inaccurate norm estimation of column residues; in fact, the iterative norm sampling only provably works when the input matrix has a low-rank plus noise structure (see Theorem 2). On the other hand, when either the target rank or the missing rate is not too high iterative norm sampling works just as good; it is particularly competitive when the true rank of the input matrix is low (see the top row of Figure 1).

When the input matrix has coherent columns, as shown in Figure 2, it becomes easier to observe performance gaps among different algorithms. The block OMP algorithm completely fails in such cases and the selection error for group Lasso also increases considerably. This is due to the fact that both algorithms observe matrix entries by sampling uniformly at random and hence could be poorly informed when the underlying matrix is highly coherent. On the other hand, both leverage score sampling and iterative norm sampling are more robust to column coherence. The coherence among columns also makes the separation between norm sampling and volume sampling clearer in Figure 2. In particular, there is a

3. We discuss on the poor performance of with replacement algorithms in Section 7.5.

Table 2: Averaging SNP window sizes for different ε values and number of selected columns per window.

	5 columns	10 columns	15 columns	20 columns	25 columns
$\varepsilon = 95\%$	63.4	248.9	516.3	891.0	1405.7
$\varepsilon = 98\%$	18.8	62.1	123.4	203.8	309.7

significant gap between the two sampling with replacement curves and the norm sampling algorithm degrades to its worst-case additive error bound (see Theorem 1). The gap between the sampling without replacement curves is smaller since the coherent column is only repeated for 5 times in the design and so an algorithm can not be "too wrong" if it samples columns without replacement.

To further investigate how the proposed and baseline algorithms adapt to different levels of coherence, we report in Figure 3 the selection error on noisy low-rank matrices with varying number of repeated columns. Matrices with more repeated columns have higher coherence level. We can see that there is a clear separation of two groups of algorithms: the first group includes norm sampling, block OMP and group Lasso, whose error increases as the matrix becomes more coherent. Also, design matrix assumptions (e.g., restricted isometry) are violated for group Lasso. This suggests that these algorithms only have additive error bounds, or adapt poorly to column coherence of the underlying data matrix. On the other hand, the selection error of volume sampling and iterative norm sampling remains stable or slightly decreases. This is consistent with our theoretical results that both volume sampling and iterative norm sampling enjoy relative error bounds.

6.2 Application to tagging Single Nucleotide Polymorphisms (tSNPs) selection

We apply our proposed methods on real-world genetic data sets. We consider the tagging Single Nucleotide Polymorphisms (tSNP) selection task as described in (Ke and Cardon, 2003; Paschou et al., 2007). The task aims at selecting a small set of SNPs in human genes such that the selected SNPs (called tagging SNPs) capture the genetic information within a specific genome region. More specifically, given an $n_1 \times n_2$ matrix with each row corresponding to the genome expression for an individual, we want to select k columns (typically $k \ll n_2$) corresponding to k tagging SNPs that best capture the entire SNP matrix across different individuals. Matrix column subset selection methods have been successfully applied to the tSNP selection problem (Paschou et al., 2007).

In this section we demonstrate that our proposed algorithms could achieve the same objective while allowing many missing entries in the raw data matrix. We also compare the selection error of the proposed methods under different missing rate and number of tSNP settings. We did not apply Block OMP and group Lasso because the former cannot handle coherent data matrices and the latter does not scale well. The data set we used is the HapMap Phase 2 data set (international HapMap consortium, 2003). For demonstration purposes, we use gene data for the first chromosome of a joint east Asian population consisting of Han Chinese in Beijing (CHB) and Japanese in Tokyo (JPT). The data matrix

consists of 89 rows (individuals) and 311,854 columns (SNPs). Each matrix entry has two letters $b_1 b_2$ describing a specific gene expression for an individual.

We follow the same step as described in (Javed et al., 2011) to preprocess the data. We first convert the raw data matrix into a numerical matrix $\mathbf{M}$ with +1/0/-1 entries as follows: let B_1 and B_2 be the bases that appear for the jth SNP. Fix an individual i with its gene expression $b_1 b_2$. If $b_1 b_2 = B_1 B_1$ then $\mathbf{M}_{ij}$ is set to -1; else if $b_1 b_2 = B_2 B_2$ then $\mathbf{M}_{ij}$ is set 1; otherwise $\mathbf{M}_{ij}$ is set to 0. We further split the SNPs into multiple consecutive "windows" so that within each window w the SVD reconstruction error $\|\mathbf{M}^{(w)} - \mathbf{M}_k^{(w)}\|_F^2 / \|\mathbf{M}^{(w)}\|_F^2$ is no larger than ε with ε set to 5% and 2%. We refer the readers to Figure 1 in (Javed et al., 2011) for details of the preprocessing steps. Averaging window length (i.e., number of SNPs within each window) are shown in Table 2 for different k and ε settings. After preprocessing, column subset selection algorithms are performed for each SNP window and the selection error is averaged across all windows, as reported in Figure 4. The number of selected columns per window (k) ranges from 5 to 25 and the sampling budget α ranges from 10% to 60%.

In Figure 4 we observe that iterative norm sampling and approximate leverage score sampling outperform norm sampling by a large margin. This is because the truncated data matrix within each window is very close to an exact low-rank matrix and hence relative error algorithms achieve much better performance than additive error ones. In addition, approximate leverage score sampling significantly outperforms norm sampling under both the with replacement and without replacement schemes. This shows that the heterogeneity of human SNPs cannot be captured merely by their norms because the norm is simply the proportion of heterozygous within a population and provides little information about its importance across the entire chromosome. The spikiness of leverage score distribution is empirically verified in Figure 5. Finally, we remark that sampling without replacement is much better than sampling with replacement and should always be preferred in practice. We discuss this aspect in Section 7.5.

6.3 Application to column-based image compression

In this section we show how active sampling can be applied to column-based image compression without observing entire images. Given an image, we first actively subsample a small number of pixels from the original image. We then select a subset of columns based on the observed pixels and reconstruct the entire image by projecting each column to the space spanned by the selected column subsets.

In Figure 6 we depicted the final compressed image as well as intermediate steps (e.g., subsampled pixels and selected columns) on the 512×512 8-bit gray scale Lena standard test image. We also report the mean and standard deviation of selection error across 10 runs under different settings of target column subset sizes in Table 3.

Table 3 shows that the iterative norm sampling algorithm consistently outperforms norm sampling and so is the leverage score sampling method when the target column subset size is large, which implies small oracle error $\|\mathbf{M} - \mathbf{M}_k\|_F^2$. To get an intuitive sense of why this is the case, we refer the readers to the selected columns for each of the sampling algorithm as shown in Figure 6 (the middle column). It can be seen that the norm sampling algorithm (Figure 6b) oversamples columns in relatively easy regions (e.g., the white bar on the left

Table 3: Relative selection error $\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_F / \|\mathbf{M}\|_F$ on the standard Lena test image (512×512) for norm sampling (Norm), iterative norm sampling (Iter. norm) and approximate leverage score sampling (Lev. score). Results also compared to a uniform sampling baseline (Uniform) and the truncated SVD lower bound (SVD). The percentage of observed entries α is set to $\alpha = 30\%$. Number of columns used for reconstruction varies from 25 to 100.

	Uniform	Norm	Iter. norm	Lev. score	SVD
25 columns	$.151 \pm .009$	$.147 \pm .004$	$\mathbf{.136 \pm .004}$	$.148 \pm .007$	.092
50 columns	$.104 \pm .004$	$.103 \pm .003$	$\mathbf{.092 \pm .001}$	$.105 \pm .003$	.059
100 columns	$.064 \pm .002$	$.065 \pm .001$	$\mathbf{.053 \pm .001}$	$.061 \pm .002$	.032

side and the smooth part of the face) because these regions have large pixel values (i.e., they are whiter than the other pixels) and hence have larger column norms. In contrast, the iterative norm sampling algorithm (Figure 6c) focuses most sampled columns on the tassel and hair parts which are complicated and cannot be well approximated by other columns. This shows that the iterative norm sampling method has the power to adapt to highly heterogeneous columns and produce better approximations. Finally, we remark that though both leverage score sampling and iterative norm sampling have relative error guarantees, in practice the iterative norm sampling performs much better than leverage score sampling for matrices whose rank is not very high.

7. Discussion

We discuss on several aspects of the proposed algorithms and their analysis.

7.1 Limitation of passive sampling

In most cases the observed entries of a partially observable matrix are sampled according to some sampling schemes. We say a sampling scheme is *passive* when the sampling distribution (i.e., probability of observing a particular matrix entry) is fixed a priori and does not depend on the data matrix. On the other hand, an *active* sampling scheme adapts its sampling distribution according to previous observations and requests unknown data points in a feedback driven way. We mainly focus on active sampling methods in this paper (both Algorithm 1 and 2 perform active sampling). However, Algorithm 3 only requires passive sampling because the sampling distribution of rows is the uniform distribution and is fixed a priori.

Passive sampling is known to work poorly for coherent matrices (Krishnamurthy and Singh, 2014; Chen et al., 2013). In this section, we make the following three remarks on the power of passive sampling for column subset selection:

Remark 1 The $\|\mathbf{M} - \mathbf{C}\mathbf{X}\|_\xi$ reconstruction error bound for column subset selection is hard for passive sampling. In particular, it can be shown that no passive sampling algorithm achieves relative reconstruction error bound with high probability unless it observes $\Omega(n_1 n_2)$

entries of an $n_1 \times n_2$ matrix $\mathbf{M}$. This holds true even if $\mathbf{M}$ is assumed to be exact low rank and has incoherent column space.

This remark can be formalized by noting that when $\mathbf{M}$ is exact low rank then relative reconstruction error implies exact recovery of $\mathbf{M}$, or in other words, matrix completion. Here we cite the hardness result in (Krishnamurthy and Singh, 2014) for completing coherent matrix by passive sampling. Similar results could also be obtained by applying Theorem 6 in (Chen et al., 2013).

Theorem 7 (Theorem 2, (Krishnamurthy and Singh, 2014)) *Let $\mathcal{X}$ denote all $n_1 \times n_2$ matrices whose rank is no more than k and column space has incoherence μ_0 as defined in Eq. (5). Fix $m < n_1 n_2$ and let $\mathcal{Q}$ denote all passive sampling distributions over m samples of $n_1 n_2$ matrix entries. Let $\mathcal{F} = \{f : \mathbb{R}^m \to \mathcal{X}\}$ be the collection of (possibly random) matrix completion algorithms. We then have*

$$R^*_{\text{mc}} := \inf_{f \in \mathcal{F}} \inf_{q \in \mathcal{Q}} \sup_{\mathbf{X} \in \mathcal{X}} \Pr_{\Omega \sim q; f} [f(\Omega, \mathbf{X}_\Omega) \neq \mathbf{X}] \geq \frac{1}{2} - \left\lceil \frac{m}{(1 - \frac{k-1}{k\mu_0})n_1} \right\rceil \frac{1}{2(n-k)}, \tag{41}$$

where $n = \max(n_1, n_2)$. As a remark, when μ_0 is a constant then $R^*_{\text{mc}} = \Omega(1)$ whenever $m = o(n_1(n_2 - k))$.

Remark 2 For the $\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger \mathbf{M}\|_\xi$ selection error (with only column indices C output by an CSS algorithm), it is possible for a passive sampling algorithm to achieve a relative error bound with high probability. In fact, Algorithm 3 and Theorem 3 precisely accomplish this. In addition, when the input matrix is exact low rank, Theorem 3 implies that there exists a passive sampling algorithm that outputs a small subset of columns which span the entire column subspace of a row-coherent matrix with high probability. This result shows column subset selection is easier than matrix completion when only indices of the selected column subset are required. It does not violate Theorem 7, however, because knowing which columns span the column space of an input matrix does not imply we can complete the matrix without further samples.

Remark 3 Although Remark 2 and Theorem 3 shows that it is possible to achieve relative $\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger \mathbf{M}\|_F$ error bound for row coherent matrices via passive sampling, we show in this section that passive sampling is insufficient under a slightly weaker notion of column incoherence. In particular, instead of assuming $\mu(\mathcal{U}) \leq \mu_0$ on the column space as in Eq. (5), we assume $\mu(\boldsymbol{x}_i) \leq \mu_1$ where μ_1 is independent of k for every column $\boldsymbol{x}_i$ as in Eq. (7). Note that if $\text{rank}(\mathcal{U}) = k$ and $\boldsymbol{x}_i \in \mathcal{U}$ then $\mu(\boldsymbol{x}_i) \leq k\mu(\mathcal{U})$. So for exact low rank matrices the vector-based incoherence assumption in Eq. (7) is weaker than the subspace-based incoherence assumption in Eq. (5). We then have the following theorem, which is proved in Appendix C.

Theorem 8 *Let $\mathcal{X}'$ denote all $n_1 \times n_2$ matrices whose rank is no more than k and incoherence $\mu_1 \geq 1 + \frac{1}{n_1 - 1}$ as defined in Eq. (7) for each column. Fix $m < n_1 n_2$ and let $\mathcal{Q}$ denote all passive sampling distributions over m samples of $n_1 n_2$ matrix entries. Let $\mathcal{F}' = \{f : \mathbb{R}^m \to [n_2]^k\}$ be the collection of (possibly random) column subset selection algorithms. We then have*

$$R^*_{\text{css}} := \inf_{f \in \mathcal{F}'} \inf_{q \in \mathcal{Q}} \sup_{\mathbf{X} \in \mathcal{X}'} \Pr_{\Omega \sim q; f} [\mathbf{X} \neq \mathbf{X}_C \mathbf{X}_C^\dagger \mathbf{X}] \geq \frac{1}{2} - \frac{m}{2n_1(n_2 - k)}, \tag{42}$$

where $C = f(\mathbf{X}, \mathbf{X}_\Omega)$ is the output column subset of f. As a remark, the failure probability R^*_{css} satisfies $R^*_{\text{css}} = \Omega(1)$ whenever $m = o(n_1(n_2 - k))$.

Theorem 8 combined with Theorem 7 shows a separation of hardness between column subset selection and matrix completion. It also formalizes the intuitive limited power of passive sampling over coherent matrices.

7.2 Time complexity

In this section we report the theoretical time complexity of our proposed algorithms as well as the optimization based methods for comparison in Table 4. We assume the input matrix $\mathbf{M}$ is square $n \times n$ and we are using s columns to approximate the top-k component of $\mathbf{M}$. Let $\alpha = m/n^2$ be the percentage of observed data. $\texttt{svd}(a, b, c)$ denotes the time for computing the top-c truncated SVD of an $a \times b$ matrix.

Suppose the observation ratio α is a constant and the `svd` operation takes quadratic time. Then the time complexity for all algorithms can be sorted as

$$\begin{aligned}\textsc{Norm}; O(n^2) < \textsc{Lev. score}; O(kn^2) < \textsc{Iter. norm}, \textsc{Block OMP}; O(sn^3) \\ < \textsc{gLasso}, O(T(n^3 + s^2 n^2)).\end{aligned} \tag{43}$$

Perhaps not surprisingly, in Section 6.2 and 6.3 on real-world data sets we show the reverse holds for selection error for the first three algorithms in Eq. (43).

7.3 Sample complexity, column subset size and selection error

We remark on the connection of sample complexity (i.e., number of observed matrix entries), size of column subsets and reconstruction error for column subset selection. For column subset selection when the target column subset size is fixed the sample complexity acts more like a threshold: if not enough number of matrix entries are observed then the algorithm fails since the column norms are not accurately estimated, but when a sufficient number of observations are available the reconstruction error does not differ much. Such phase transition was also observed in other matrix completion/approximation tasks as well, for example, in (Krishnamurthy and Singh, 2014). In fact, the guarantee in Eq. (8), for example, is exactly the same as in (Frieze et al., 2004) under the fully observed setting, i.e., $m_1 = n_1$.

The bottom three plots in Figure 2 are an excellent illustration of this phenomenon. When $\alpha = 0.3$ the selection error of Algorithm 2 is very high, which means the algorithm

Table 4: Time complexity of proposed and baseline algorithms. k denotes the intrinsic rank and s denotes the number of selected columns. Dependency on failure probability δ and other poly-logarithmic dependency is omitted.

Algorithm	Norm	Iter. norm*	Lev. score	Block OMP*	gLasso†
Time Complexity	$O(\alpha n^2)$	$O(\alpha^2 s n^3)$	$O(\text{svd}(\alpha n, n, k))$	$O(\alpha^2 s n^3)$	$O(T(n^3 + s^2 n^2))$

*Assume $\alpha n > s$ and $\alpha^2 n > 1$.

†Using solution path implementation; T is the desired number of λ values.

does not have enough samples. However, for $\alpha = 0.6$ and $\alpha = 0.9$ the performance of Algorithm 2 is very similar.

7.4 Sample complexity of the iterative norm sampling algorithm

We try to verify the sample complexity dependence on the intrinsic matrix rank k for the iterative norm sampling algorithm (Algorithm 2). To do this, we run Algorithm 2 under various settings of intrinsic dimension k and the sampling probability α (which is basically proportional to the expected number of per-column samples m). We then plot the selection error $\|\mathbf{M} - \mathbf{C}\mathbf{C}^{\dagger}\mathbf{M}\|_F$ against α, α/k and α/k^2 in Figure 7.

Theorem 2 states that the dependence of m on k should be $m = \widetilde{O}(k^2)$ ignoring logarithmic factors. However, in Figure 7 one can observe that when the selection error is plotted against α/k the different curves coincide. This suggests that the actual dependence of m on k should be close to linear instead of quadratic. It is an interesting question whether we can get rid of the use of union bounds over all n_2-choose-k column subsets in the proof of Theorem 2 in order to get a near linear dependence over k. Note that the curves converge to different values for different k settings because selection error decreases when more columns are used to reconstruct the input matrix.

7.5 Sampling with and without replacement

In the experiments we observe that for norm sampling (Algorithm 1) and approximate leverage score sampling (Algorithm 3) the two column sampling schemes, i.e., sampling with and without replacement, makes a big difference in practice (e.g., see Figure 1, 2, and 4). In fact, sampling without replacement always outperforms sampling with replacement because under the latter scheme there is a positive probability of sampling the same column more than once. Though we analyzed both algorithm under the sampling with replacement scheme, in practice sampling without replacement should always be used since it makes no sense to select a column more than once. Finally, we remark that for iterative norm sampling (Algorithm 2) a column will never be picked more than once since the (estimated) projected norm of an already selected column is zero with probability 1.

Acknowledgments

We would like to thank Akshay Krishnamurthy for helpful discussion on the proof of Theorem 2 and James Duyck for a solution path implementation of group Lasso for column subset selection. This work is supported in part by grants NSF-1252412 and AFOSR-FA9550-14-1-0285.

A. Analysis of the active norm sampling algorithm

Proof [Proof of Lemma 1] This lemma is a direct corollary of Theorem 2 from (Frieze et al., 2004). First, let $P_i = \hat{c}_i/\hat{f}$ be the probability of selecting the i-th column of $\mathbf{M}$.

By assumption, we have $P_i \geq \frac{1-\alpha}{1+\alpha}\|\boldsymbol{x}_i\|_2^2/\|\mathbf{M}\|_F^2$. Applying Theorem 2 [4] from (Frieze et al., 2004) we have that with probability at least $1-\delta$, there exists an orthonormal set of vectors $\boldsymbol{y}^{(1)}, \cdots, \boldsymbol{y}^{(k)} \in \mathbb{R}^{n_1}$ in span($\mathbf{C}$) such that

$$\left\| \mathbf{M} - \left(\sum_{j=1}^{k} \boldsymbol{y}^{(j)} \boldsymbol{y}^{(j)^\top} \right) \mathbf{M} \right\|_F^2 \leq \|\mathbf{M} - \mathbf{M}_k\|_F^2 + \frac{(1+\alpha)k}{(1-\alpha)\delta s}\|\mathbf{M}\|_F^2. \tag{44}$$

Finally, to complete the proof, note that every column of $\left(\sum_{j=1}^{k} \boldsymbol{y}^{(j)} \boldsymbol{y}^{(j)^\top}\right) \mathbf{M}$ can be represented as a linear combination of columns in $\mathbf{C}$; furthermore,

$$\|\mathbf{M} - \mathcal{P}_C(\mathbf{M})\|_F = \min_{\mathbf{X} \in \mathbb{R}^{k \times n_2}} \|\mathbf{M} - \mathbf{C}\mathbf{X}\|_F \leq \left\| \mathbf{M} - \left(\sum_{j=1}^{k} \boldsymbol{y}^{(j)} \boldsymbol{y}^{(j)^\top} \right) \mathbf{M} \right\|_F. \tag{45}$$

■

Proof [Proof of Theorem 1] First, set $m_1 = \Omega(\mu_0 \log(n_2/\delta_1))$ we have that with probability $\geq 1 - \delta_1$ the inequality

$$(1-\alpha)\|\boldsymbol{x}_i\|_2^2 \leq \hat{c}_i \leq (1+\alpha)\|\boldsymbol{x}_i\|_2^2$$

holds with $\alpha = 0.5$ for every column i, using Lemma 2. Next, putting $s \geq 6k/\delta_2\epsilon^2$ and applying Lemma 1 we get

$$\|\mathbf{M} - \mathcal{P}_C(\mathbf{M})\|_F \leq \|\mathbf{M} - \mathbf{M}_k\|_F + \epsilon\|\mathbf{M}\|_F \tag{46}$$

with probability at least $1 - \delta_2$. Finally, note that when $\alpha \leq 1/2$ and $n_1 \leq n_2$ the bound in Lemma 3 is dominated by

$$\|\mathbf{M} - \widehat{\mathbf{M}}\|_2 \leq \|\mathbf{M}\|_F \cdot O\left(\sqrt{\frac{\mu_0}{m_2}} \log\left(\frac{n_1 + n_2}{\delta} \right) \right). \tag{47}$$

Consequently, for any $\epsilon' > 0$ if $m_2 = \Omega((\epsilon')^{-2}\mu_0 \log^2((n_1+n_2)/\delta_3)$ we have with probability $\geq 1 - \delta_3$

$$\|\mathbf{M} - \widehat{\mathbf{M}}\|_2 \leq \epsilon'\|\mathbf{M}\|_F. \tag{48}$$

The proof is then completed by taking $\epsilon' = \epsilon/\sqrt{s}$:

$$\begin{aligned}
\|\mathbf{M} - \mathbf{C}\mathbf{X}\|_F &= \|\mathbf{M} - \mathcal{P}_C(\widehat{\mathbf{M}})\|_F \\
&\leq \|\mathbf{M} - \mathcal{P}_C(\mathbf{M})\|_F + \|\mathcal{P}_C(\mathbf{M} - \widehat{\mathbf{M}})\|_F \\
&\leq \|\mathbf{M} - \mathbf{M}_k\|_F + \epsilon\|\mathbf{M}\|_F + \sqrt{s}\|\mathcal{P}_C(\mathbf{M} - \widehat{\mathbf{M}})\|_2 \\
&\leq \|\mathbf{M} - \mathbf{M}_k\|_F + \epsilon\|\mathbf{M}\|_F + \sqrt{s} \cdot \epsilon'\|\mathbf{M}\|_F \\
&\leq \|\mathbf{M} - \mathbf{M}_k\|_F + 2\epsilon\|\mathbf{M}\|_F.
\end{aligned}$$

■

4. The original theorem concerns random samples of rows; it is essentially the same for random samples of columns.

B. Analysis of the iterative norm sampling algorithm

Proof [Proof of Lemma 4]

We first prove Eq. (16). Observe that $\dim(\mathcal{U}(C)) \leq s$. Let $\mathbf{R}_C = (\mathbf{R}^{(C(1))}, \cdots, \mathbf{R}^{(C(s))}) \in \mathbb{R}^{n_1 \times s}$ denote the selected s columns in the noise matrix $\mathbf{R}$ and let $\mathcal{R}(C) = \text{span}(\mathbf{R}_C)$ denote the span of selected columns in $\mathbf{R}$. By definition, $\mathcal{U}(C) \subseteq \mathcal{U} \cup \mathcal{R}(C)$, where $\mathcal{U} = \text{span}(\mathbf{A})$ denotes the subspace spanned by columns in the deterministic matrix $\mathbf{A}$. Consequently, we have the following bound on $\|\mathcal{P}_{\mathcal{U}(C)}\boldsymbol{e}_i\|$ (assuming each entry in $\mathbf{R}$ follows a zero-mean Gaussian distribution with σ^2 variance):

$$\begin{aligned}
\|\mathcal{P}_{\mathcal{U}(C)}\boldsymbol{e}_i\|_2^2 &\leq \|\mathcal{P}_{\mathcal{U}}\boldsymbol{e}_i\|_2^2 + \|\mathcal{P}_{\mathcal{U}^\perp \cap \mathcal{R}(C)}\boldsymbol{e}_i\|_2^2 \\
&\leq \|\mathcal{P}_{\mathcal{U}}\boldsymbol{e}_i\|_2^2 + \|\mathcal{P}_{\mathcal{R}(C)}\boldsymbol{e}_i\|_2^2 \\
&\leq \frac{k\mu_0}{n_1} + \|\mathbf{R}_C\|_2^2 \|(\mathbf{R}_C^\top \mathbf{R}_C)^{-1}\|_2^2 \|\mathbf{R}_C^\top \boldsymbol{e}_i\|_2^2 \\
&\leq \frac{k\mu_0}{n_1} + \frac{(\sqrt{n_1}+\sqrt{s}+\epsilon)^2\sigma^2}{(\sqrt{n_1}-\sqrt{s}-\epsilon)^4\sigma^4} \cdot \sigma^2(s + 2\sqrt{s\log(2/\delta)} + 2\log(2/\delta)).
\end{aligned}$$

For the last inequality we apply Lemma 14 to bound the largest and smallest singular values of $\mathbf{R}_C$ and Lemma 12 to bound $\|\mathbf{R}_C^\top \boldsymbol{e}_i\|_2^2$, because $\mathbf{R}_C^\top \boldsymbol{e}_i$ follow i.i.d. Gaussian distributions with covariance $\sigma^2 \mathbf{I}_{s\times s}$. If ϵ is set as $\epsilon = \sqrt{2\log(4/\delta)}$ then the last inequality holds with probability at least $1-\delta$. Furthermore, when $s \leq n_1/2$ and δ is not exponentially small (e.g., $\sqrt{2\log(4/\delta)} \leq \frac{\sqrt{n_1}}{4}$), the fraction $\frac{(\sqrt{n_1}+\sqrt{s}+\epsilon)^2}{(\sqrt{n_1}-\sqrt{s}-\epsilon)^4}$ is approximately $O(1/n_1)$. As a result, with probability $1 - n_1\delta$ the following holds:

$$\begin{aligned}
&\mu(\mathcal{U}(C)) = \frac{n_1}{s} \max_{1\leq i\leq n_1} \|\mathcal{P}_{\mathcal{U}(C)}\boldsymbol{e}_i\|_2^2 \\
&\leq \frac{n_1}{s}\left(\frac{k\mu_0}{n_1} + O\left(\frac{s + \sqrt{s\log(1/\delta)} + \log(1/\delta)}{n_1}\right)\right) = O\left(\frac{k\mu_0 + s + \sqrt{s\log(1/\delta)} + \log(1/\delta)}{s}\right).
\end{aligned} \tag{49}$$

Finally, putting $\delta' = n_1/\delta$ we prove Eq. (16).

Next we try to prove Eq. (17). Let $\boldsymbol{x}$ be the i-th column of $\mathbf{M}$ and write $\boldsymbol{x} = \boldsymbol{a} + \boldsymbol{r}$, where $\boldsymbol{a} = \mathcal{P}_{\mathcal{U}}(\boldsymbol{x})$ and $\boldsymbol{r} = \mathcal{P}_{\mathcal{U}^\perp}(\boldsymbol{x})$. Since the deterministic component of $\boldsymbol{x}$ lives in $\mathcal{U}$ and the random component of $\boldsymbol{x}$ is a vector with each entry sampled from i.i.d. zero-mean Gaussian distributions, we know that $\boldsymbol{r}$ is also a zero-mean random Gaussian vector with i.i.d. sampled entries. Note that $\mathcal{U}(C)$ does not depend on the randomness over $\{\mathbf{M}^{(i)} : i \notin C\}$. Therefore, in the following analysis we will assume $\mathcal{U}(C)$ to be a fixed subspace $\widetilde{\mathcal{U}}$ with dimension at most s.

The projected vector $\boldsymbol{x}' = \mathcal{P}_{\widetilde{\mathcal{U}}^\perp}\boldsymbol{x}$ can be written as $\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{a}} + \tilde{\boldsymbol{r}}$, where $\tilde{\boldsymbol{a}} = \mathcal{P}_{\widetilde{\mathcal{U}}^\perp}\boldsymbol{a}$ and $\tilde{\boldsymbol{r}} = \mathcal{P}_{\widetilde{\mathcal{U}}^\perp}\boldsymbol{r}$. By definition, $\tilde{\boldsymbol{a}}$ lives in the subspace $\mathcal{U} \cap \widetilde{\mathcal{U}}^\perp$. So it satisfies the incoherence assumption

$$\mu(\tilde{\boldsymbol{a}}) = \frac{n_1\|\tilde{\boldsymbol{a}}\|_\infty^2}{\|\tilde{\boldsymbol{a}}\|_2^2} \leq k\mu(\mathcal{U}) \leq k\mu_0. \tag{50}$$

On the other hand, because $\tilde{\boldsymbol{r}}$ is an orthogonal projection of some random Gaussian variable, $\tilde{\boldsymbol{r}}$ is still a Gaussian random vector, which lives in $\mathcal{U}^\perp \cap \widetilde{\mathcal{U}}^\perp$ with rank at least $n_1 - k - s$.

Subsequently, we have

$$\begin{aligned}
\mu(\tilde{\boldsymbol{x}}) &= n_1\frac{\|\tilde{\boldsymbol{x}}\|_\infty^2}{\|\tilde{\boldsymbol{x}}\|_2^2} \leq 3n_1\frac{\|\tilde{\boldsymbol{a}}\|_\infty^2 + \|\tilde{\boldsymbol{r}}\|_\infty^2}{\|\tilde{\boldsymbol{a}}\|_2^2 + \|\tilde{\boldsymbol{r}}\|_2^2} \\
&\leq 3n_1\frac{\|\tilde{\boldsymbol{a}}\|_\infty^2}{\|\tilde{\boldsymbol{a}}\|_2^2} + 3n_1\frac{\|\tilde{\boldsymbol{r}}\|_\infty^2}{\|\tilde{\boldsymbol{r}}\|_2^2} \\
&\leq 3k\mu_0 + \frac{6\sigma^2 n_1\log(2n_1n_2/\delta)}{\sigma^2(n_1-k-s) - 2\sigma^2\sqrt{(n_1-k-s)\log(n_2/\delta)}}.
\end{aligned}$$

For the second inequality we use the fact that $\frac{\sum_i a_i}{\sum_i b_i} \leq \sum_i \frac{a_i}{b_i}$ whenever $a_i, b_i \geq 0$. For the last inequality we use Lemma 13 on the enumerator and Lemma 12 on the denominator. Finally, note that when $\max(s,k) \leq n_1/4$ and $\log(n_2/\delta) \leq n_1/64$ the denominator can be lower bounded by $\sigma^2 n_1/4$; subsequently, we can bound $\mu(\tilde{\boldsymbol{x}})$ as

$$\mu(\tilde{\boldsymbol{x}}) \leq 3k\mu_0 + \frac{24\sigma^2 n_1\log(2n_1n_2/\delta)}{\sigma^2 n_1} \leq 3k\mu_0 + 24\log(2n_1n_2/\delta). \tag{51}$$

Taking a union bound over all $n_2 - s$ columns yields the result. ■

To prove the norm estimation consistency result in Lemma 5 we first cite a seminal theorem from (Krishnamurthy and Singh, 2014) which provides a tight error bound on a subsampled projected vector in terms of the norm of the true projected vector.

Theorem 9 *Let $\mathcal{U}$ be a k-dimensional subspace of $\mathbb{R}^n$ and $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{v}$, where $\boldsymbol{x} \in \mathcal{U}$ and $\boldsymbol{v} \in \mathcal{U}^\perp$. Fix $\delta' > 0$, $m \geq \max\{\frac{8}{3}k\mu(\mathcal{U})\log\left(\frac{2k}{\delta'}\right), 4\mu(\boldsymbol{v})\log(1/\delta')\}$ and let Ω be an index set with entries sampled uniformly with replacement with probability m/n. Then with probability at least $1 - 4\delta'$:*

$$\frac{m(1-\alpha) - k\mu(\mathcal{U})\frac{\beta}{1-\gamma}}{n}\|\boldsymbol{v}\|_2^2 \leq \|\boldsymbol{y}_\Omega - \mathcal{P}_{U_\Omega}\boldsymbol{y}_\Omega\|_2^2 \leq (1+\alpha)\frac{m}{n}\|\boldsymbol{v}\|_2^2, \tag{52}$$

where $\alpha = \sqrt{2\frac{\mu(\boldsymbol{v})}{m}\log(1/\delta')} + 2\frac{\mu(\boldsymbol{v})}{3m}\log(1/\delta')$, $\beta = (1+2\sqrt{\log(1/\delta')})^2$ and $\gamma = \sqrt{\frac{8k\mu(\mathcal{U})}{3m}\log(2k/\delta')}$.

We are now ready to prove Lemma 5.

Proof [Proof of Lemma 5] By Algorithm 2, we know that $\dim(\mathcal{S}_t) = t$ with probability 1. Let $\boldsymbol{y} = \mathbf{M}^{(i)}$ denote the i-th column of $\mathbf{M}$ and let $\boldsymbol{v} = \mathcal{P}_{\mathcal{S}_t}\boldsymbol{y}$ be the projected vector. We can apply Theorem 9 to bound the estimation error between $\|\boldsymbol{v}\|$ and $\|\boldsymbol{y}_\Omega - \mathcal{P}_{\mathcal{S}_t(\Omega)}\boldsymbol{y}_\Omega\|$.

First, when m is set as in Eq. (20) it is clear that the conditions $m \geq \frac{8}{3}t\mu(\mathcal{U})\log\left(\frac{2t}{\delta'}\right) = \Omega(k\mu_0\log(n/\delta)\log(k/\delta'))$ and $m \geq 4\mu(\boldsymbol{v})\log(1/\delta') = \Omega(k\mu_0\log(n/\delta)\log(1/\delta'))$ are satisfied. We next turn to the analysis of α, β and γ. More specifically, we want $\alpha = O(1)$, $\gamma = O(1)$ and $\frac{t\mu(\mathcal{U})}{m}\beta = O(1)$.

For α, $\alpha = O(1)$ implies $m = \Omega(\mu(\boldsymbol{v})\log(1/\delta')) = \Omega(k\mu_0\log(n/\delta)\log(1/\delta'))$. Therefore, by carefully selecting constants in $\Omega(\cdot)$ we can make $\alpha \leq 1/4$.

For γ, $\gamma = O(1)$ implies $m = \Omega(t\mu(\mathcal{U})\log(t/\delta')) = \Omega(k\mu_0\log(n/\delta)\log(k/\delta'))$. By carefully selecting constants in $\Omega(\cdot)$ we can make $\gamma \leq 0.2$.

For β, $\frac{t\mu(\mathcal{U})}{m}\beta = O(1)$ implies $m = O(t\mu(\mathcal{U})\beta) = O(k\mu_0\log(n/\delta)\log(1/\delta'))$. By carefully selecting constants we can have $\beta \leq 0.2$. Finally, combining bounds on α, β and γ we prove the desired result. ■

Before proving Lemma 6, we first cite a lemma from (Deshpande et al., 2006) that connects the volume of a simplex to the permutation sum of singular values.

Lemma 9 ((Deshpande et al., 2006)) *Fix $\mathbf{A} \in \mathbb{R}^{m\times n}$ with $m \leq n$. Suppose $\sigma_1, \cdots, \sigma_m$ are singular values of $\mathbf{A}$. Then*

$$\sum_{S\subseteq[n],|S|=k} \mathrm{vol}(\Delta(S))^2 = \frac{1}{(k!)^2} \sum_{1\leq i_1<i_2<\cdots<i_k\leq m} \sigma_{i_1}^2\sigma_{i_2}^2\cdots\sigma_{i_k}^2. \tag{53}$$

Now we are ready to prove Lemma 6.

Proof [Proof of Lemma 6] Let $\mathbf{M}_k$ denote the best rank-k approximation of $\mathbf{M}$ and assume the singular values of $\mathbf{M}$ are $\{\sigma_i\}_{i=1}^{n_1}$. Let $C = \{i_1, \cdots, i_k\}$ be the selected columns. Let $\tau \in \Pi_k$, where Π_k denotes all permutations with k elements. By $\mathcal{H}_{\tau,t}$ we denote the linear subspace spanned by $\{\mathbf{M}^{(\tau(i_1))}, \cdots, \mathbf{M}^{(\tau(i_t))}\}$ and let $d(\mathbf{M}^{(i)}, \mathcal{H}_{\tau,t})$ denote the distance between column $\mathbf{M}^{(i)}$ and subspace $\mathcal{H}_{\tau,t}$. We then have

$$\begin{aligned}
\hat{p}_C &\leq \sum_{\tau\in\Pi_k} \left(\frac{5}{2}\right)^k \frac{\|\mathbf{M}^{(\tau(i_1))}\|_2^2}{\|\mathbf{M}\|_F^2} \frac{d(\mathbf{M}^{(\tau(i_2))}, \mathcal{H}_{\tau,1})^2}{\sum_{i=1}^{n_2} d(\mathbf{M}^{(i)}, \mathcal{H}_{\tau,1})^2} \cdots \frac{d(\mathbf{M}^{(\tau(i_k))}, \mathcal{H}_{\tau,k-1})^2}{\sum_{i=1}^{n_2} d(\mathbf{M}^{(i)}, \mathcal{H}_{\tau,k-1})^2} \\
&\leq 2.5^k \cdot \frac{\sum_{\tau\in\Pi_k} \|\mathbf{M}^{(\tau(i_1))}\|^2 d(\mathbf{M}^{(\tau(i_2))}, \mathcal{H}_{\tau,1})^2 \cdots d(\mathbf{M}^{(\tau(i_k))}, \mathcal{H}_{\tau,k-1})^2}{\|\mathbf{M}\|_F^2 \|\mathbf{M}-\mathbf{M}_1\|_F^2 \cdots \|\mathbf{M}-\mathbf{M}_{k-1}\|_F^2} \\
&= 2.5^k \cdot \frac{\sum_{\tau\in\Pi_k} (k!)^2 \mathrm{vol}(\Delta(C))^2}{\|\mathbf{M}\|_F^2 \|\mathbf{M}-\mathbf{M}_1\|_F^2 \cdots \|\mathbf{M}-\mathbf{M}_{k-1}\|_F^2} \\
&= 2.5^k \cdot \frac{(k!)^3 \mathrm{vol}(\Delta(C))^2}{\sum_{i=1}^{n_1}\sigma_i^2 \sum_{i=2}^{n_1}\sigma_i^2 \cdots \sum_{i=k}^{n_1}\sigma_i^2} \\
&\leq 2.5^k \cdot \frac{(k!)^3 \mathrm{vol}(\Delta(C))^2}{\sum_{1\leq i_1<i_2<\cdots<i_k\leq n_1} \sigma_{i_1}^2\sigma_{i_2}^2\cdots\sigma_{i_k}^2} \\
&= 2.5^k \cdot \frac{k! \mathrm{vol}(\Delta(C))^2}{\sum_{T:|T|=k} \mathrm{vol}(\Delta(T))^2} = 2.5^k k! p_C.
\end{aligned}$$

For the first inequality we apply Eq. (23) and for the second to last inequality we apply Lemma 9. ■

Lemma 7 can be proved by applying Theorem 5 for T rounds, given the norm estimation accuracy bound in Proposition 1.

Proof [Proof of Lemma 7] First note that

$$\|\mathbf{M} - \mathcal{P}_{\mathcal{U}\cup\mathcal{S}_1\cup\cdots\cup\mathcal{S}_T}(\mathbf{M})\|_F^2 \leq \|\mathbf{M} - \mathcal{P}_{\mathcal{U}\cup\mathcal{S}_1\cup\cdots\cup\mathcal{S}_T,k}(\mathbf{M})\|_F^2.$$

Applying Theorem 5 with $\frac{1+\alpha}{1-\alpha} = \frac{5}{2}$, we have

$$\begin{aligned}
&\mathbb{E}\left[\|\mathbf{M} - \mathcal{P}_{\mathcal{U}\cup\mathcal{S}_1\cup\cdots\cup\mathcal{S}_T}(\mathbf{M})\|_F^2\right] \\
\leq\ & \|\mathbf{M}-\mathbf{M}_k\|_F^2 + \frac{5k}{2s_T}\mathbb{E}\left[\|\mathbf{M} - \mathcal{P}_{\mathcal{U}\cup\mathcal{S}_1\cup\cdots\mathcal{S}_{T-1}}(\mathbf{M})\|_F\right]^2 \\
\leq\ & \|\mathbf{M}-\mathbf{M}_k\|_F^2 + \frac{5k}{2s_T}\left(\|\mathbf{M}-\mathbf{M}_k\|_F^2 + \frac{5k}{2s_{T-1}}\mathbb{E}\left[\|\mathbf{M} - \mathcal{P}_{\mathcal{U}\cup\mathcal{S}_1\cup\cdots\mathcal{S}_{T-2}}(\mathbf{M})\|_F^2\right]\right) \\
\leq\ & \cdots \\
\leq\ & \left(1 + \frac{5}{2}\frac{k}{s_T} + \left(\frac{5}{2}\right)^2 \frac{k^2}{s_T s_{T-1}} + \cdots + \left(\frac{5}{2}\right)^{T-1} \frac{k^{T-1}}{s_{T-1}\cdots s_1}\right)\|\mathbf{M}-\mathbf{M}_k\|_F^2 \\
&+ \left(\frac{5}{2}\right)^T \frac{k^T}{s_T s_{T-1}\cdots s_1}\|\mathbf{M} - \mathcal{P}_{\mathcal{U}}(\mathbf{M})\|_F^2 \\
\leq\ & \left(1 + \frac{\epsilon}{4\delta} + \frac{\epsilon}{20\delta} + \cdots\right)\|\mathbf{M}-\mathbf{M}_k\|_F^2 + \frac{\epsilon/2}{2^T\delta}\|\mathbf{E}\|_F^2 \\
\leq\ & \left(1 + \frac{\epsilon}{2\delta}\right)\|\mathbf{M}-\mathbf{M}_k\|_F^2 + \frac{\epsilon/2}{2^T\delta}\|\mathbf{E}\|_F^2.
\end{aligned}$$

Finally applying Markov's inequality we complete the proof. ■

To prove the reconstruction error bound in Lemma 8 we need the following two technical lemmas, cited from (Krishnamurthy and Singh, 2013; Balzano et al., 2010b).

Lemma 10 ((Krishnamurthy and Singh, 2013)) *Suppose $\mathcal{U} \subseteq \mathbb{R}^n$ has dimension k and $\mathbf{U} \in \mathbb{R}^{n\times k}$ is the orthogonal matrix associated with $\mathcal{U}$. Let $\Omega \subseteq [n]$ be a subset of indices each sampled from i.i.d. Bernoulli distributions with probability m/n_1. Then for some vector $\boldsymbol{y} \in \mathbb{R}^n$, with probability at least $1-\delta$:*

$$\|\mathbf{U}_\Omega^\top \boldsymbol{y}_\Omega\|_2^2 \leq \beta\frac{m}{n_1}\frac{k\mu(\mathcal{U})}{n_1}\|\boldsymbol{y}\|_2^2, \tag{54}$$

where β is defined in Theorem 9.

Lemma 11 ((Balzano et al., 2010b)) *With the same notation in Lemma 10 and Theorem 9. With probability $\geq 1-\delta$ one has*

$$\|(\mathbf{U}_\Omega^\top\mathbf{U}_\Omega)^{-1}\| \leq \frac{n_1}{(1-\gamma)m}, \tag{55}$$

provided that $\gamma < 1$.

Now we can prove Lemma 8.

Proof [Proof of Lemma 8] Let $\mathcal{U} = \mathcal{U}(S)$ and $\mathbf{U} \in \mathbb{R}^{n_1\times s}$ be the orthogonal matrix associated with $\mathcal{U}$. Fix a column i and let $\boldsymbol{x} = \mathbf{M}^{(i)} = \boldsymbol{a} + \boldsymbol{r}$, where $\boldsymbol{a} \in \mathcal{U}$ and $\boldsymbol{r} \in \mathcal{U}^\perp$. What we want is to bound $\|\boldsymbol{x} - \mathbf{U}(\mathbf{U}_\Omega^\top\mathbf{U}_\Omega)^{-1}\mathbf{U}_\Omega^\top\boldsymbol{x}_\Omega\|_2^2$ in terms of $\|\boldsymbol{r}\|_2^2$.

Write $\boldsymbol{a} = \mathbf{U}\tilde{\boldsymbol{a}}$. By Lemma 11, if m satisfies the condition given in the Lemma then with probability over $1-\delta-\delta''$ we know $(\mathbf{U}_\Omega^\top\mathbf{U}_\Omega)$ is invertible and furthermore, $\|(\mathbf{U}_\Omega^\top\mathbf{U}_\Omega)^{-1}\|_2 \leq 2n_1/m$. Consequently,

$$\mathbf{U}(\mathbf{U}_\Omega^\top\mathbf{U}_\Omega)^{-1}\mathbf{U}_\Omega^\top\boldsymbol{a}_\Omega = \mathbf{U}(\mathbf{U}_\Omega^\top\mathbf{U}_\Omega)^{-1}\mathbf{U}_\Omega^\top\mathbf{U}_\Omega\tilde{\boldsymbol{a}} = \mathbf{U}\tilde{\boldsymbol{a}} = \boldsymbol{a}. \tag{56}$$

That is, the subsampled projector preserves components of $\boldsymbol{x}$ in subspace $\mathcal{U}$.

Now let's consider the noise term $\boldsymbol{r}$. By Corollary 1 with probability $\geq 1-\delta$ we can bound the incoherence level of $\boldsymbol{y}$ as $\mu(\boldsymbol{y}) = O(s\mu_0 \log(n/\delta))$. The incoherence of subspace $\mathcal{U}$ can also be bounded as $\mu(\mathcal{U}) = O(\mu_0 \log(n/\delta))$. Subsequently, given $m = \Omega(\epsilon^{-1} s\mu_0 \log(n/\delta)\log(n/\delta''))$ we have (with probability $\geq 1-\delta-2\delta''$)

$$\begin{aligned} & \|\boldsymbol{x} - \mathbf{U}(\mathbf{U}_\Omega^\top \mathbf{U}_\Omega)^{-1}\mathbf{U}_\Omega^\top(\boldsymbol{a}+\boldsymbol{r})|_2^2 \\ = \; & \|\boldsymbol{a}+\boldsymbol{r} - \mathbf{U}(\mathbf{U}_\Omega^\top \mathbf{U}_\Omega)^{-1}\mathbf{U}_\Omega^\top(\boldsymbol{a}+\boldsymbol{r})\|_2^2 \\ = \; & \|\boldsymbol{r} - \mathbf{U}(\mathbf{U}_\Omega^\top \mathbf{U}_\Omega)^{-1}\mathbf{U}_\Omega^\top\boldsymbol{r}\|_2^2 \\ \leq \; & \|\boldsymbol{r}\|_2^2 + \|(\mathbf{U}_\Omega^\top \mathbf{U}_\Omega)^{-1}\|_2^2 \|\mathbf{U}_\Omega^\top \boldsymbol{r}\|_2^2 \\ \leq \; & (1+O(\epsilon))\|\boldsymbol{r}\|_2^2. \end{aligned}$$

For the second to last inequality we use the fact that $\boldsymbol{r} \in \mathcal{U}^\perp$. By carefully selecting constants in Eq. (22) we can make

$$\|\boldsymbol{x} - \mathbf{U}(\mathbf{U}_\Omega^\top \mathbf{U}_\Omega)^{-1}\mathbf{U}_\Omega^\top \boldsymbol{x}\|_2^2 \leq (1+\epsilon)\|\mathcal{P}_{\mathcal{U}^\perp}\boldsymbol{x}\|_2^2. \tag{57}$$

Summing over all n_2 columns yields the desired result. ■

C. Proof of lower bound for passive sampling

Proof [Proof of Theorem 8] Let $\widetilde{\mathcal{X}} = \{\mathbf{X}_1, \cdots, \mathbf{X}_T\} \subseteq \mathcal{X}'$ be a finite subset of $\mathcal{X}'$ which we specify later. Let π be any prior distribution over $\widetilde{\mathcal{X}}$. We then have the following chain of inequalities:

$$\begin{aligned} R^*_{\text{css}} &= \inf_{f\in\mathcal{F}'} \inf_{q\in\mathcal{Q}} \sup_{\mathbf{X}\in\mathcal{X}'} \Pr_{\Omega\sim q;f}[\mathbf{X} \neq \mathbf{X}_C\mathbf{X}_C^\dagger\mathbf{X}] \\ &\geq \inf_{f\in\mathcal{F}'} \inf_{q\in\mathcal{Q}} \Pr_{\Omega\sim q;\mathbf{X}\sim\pi;f}[\mathbf{X} \neq \mathbf{X}_C\mathbf{X}_C^\dagger\mathbf{X}] \qquad (58) \\ &\geq \inf_{f\in\mathcal{F}'} \min_{|\Omega|=m} \Pr_{\mathbf{X}\sim\pi;f}[\mathbf{X} \neq \mathbf{X}_C\mathbf{X}_C^\dagger\mathbf{X}]. \qquad (59) \end{aligned}$$

Here Eq. (58) uses the fact that the maximum dominates any expectation over the same set and for Eq. (59) we apply Yao's principle, which asserts that the worst-case performance of a randomized algorithm is better (i.e., lower bounded) by the averaging performance of a deterministic algorithm. Hence, when the input matrix $\mathbf{X}$ is randomized by a prior π it suffices to consider only deterministic sampling schemes, which corresponds to a subset of matrix entries Ω fixed a priori, with size $|\Omega| = m$.

We next construct the subset $\widetilde{\mathcal{X}}$ and let π be the uniform distribution over $\widetilde{\mathcal{X}}$. Let $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{k-2} \in \mathbb{R}^{n_1}$ be an arbitrary set of linear independent column vectors with $[\boldsymbol{x}_i]_1 = 0$ for all $i = 1, 2, \cdots, k-2$ and $\mu(\boldsymbol{x}_1), \cdots, \mu(\boldsymbol{x}_{k-2}) = 1 + \frac{1}{n_1-1}$. This can be done by setting all nonzero entries in $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{k-2}$ to ± 1. In addition, we define $\boldsymbol{y} := (1, 1, \cdots, 1)$ and $\boldsymbol{e}_j = (0, \cdots, 0, 1, 0, \cdots, 0)$ with the only nonzero entry at the jth position. Next, define $\widetilde{\mathbf{X}} = \{\mathbf{X}^{i,j}\}_{i=k-1,j=1}^{n_2,n_1}$ with

$$\mathbf{X}^{i,j}_{(\ell)} = \begin{cases} \boldsymbol{x}_\ell & \text{if } \ell \leq k-2, \\ \boldsymbol{y} - 2\boldsymbol{e}_j & \text{if } \ell = i, \\ \boldsymbol{y} & \text{otherwise.} \end{cases} \tag{60}$$

It follows by definition that $\text{rank}(\mathbf{X}^{i,j}) = k$ and $\mu(\mathbf{X}^{i,j}_{(\ell)}) \leq \mu_1 = 1 + \frac{1}{n_1-1}$ for all i, j and ℓ. Furthermore, for fixed i and j one necessary condition for $\mathbf{X} = \mathbf{X}_C\mathbf{X}_C^\dagger\mathbf{X}$ is $\{1, 2, \cdots, k-2, i\} \subseteq C$. Therefore, if for distinct i_1, i_2, i_3, i_4 and some j_1, j_2, j_3, j_4 one has $\mathbf{X}^{i_1,j_1}_\Omega = \cdots = \mathbf{X}^{i_4,j_4}_\Omega$ then the best a column subset selection algorithm f could do is random guessing and hence $\Pr[\mathbf{X} \neq \mathbf{X}_C\mathbf{X}_C^\dagger\mathbf{X}] \geq 1/2$. Consequently, for fixed Ω one has

$$\inf_{f\in\mathcal{F}'} \Pr_{\mathbf{X}\sim\pi;f}[\mathbf{X} \neq \mathbf{X}_C\mathbf{X}_C^\dagger\mathbf{X}] \geq \frac{1}{2} - \frac{1}{2}\left|\left\{\mathbf{X}^{i,j} : \mathbf{X}^{i',j'}_\Omega \neq \mathbf{X}^{i,j}_\Omega, \forall i' \neq i, j' \in [n_1]\right\}\right|. \tag{61}$$

The final step is to bound the size of the set $E = \{\mathbf{X}^{i,j} : \mathbf{X}^{i',j'}_\Omega \neq \mathbf{X}^{i,j}_\Omega, \forall i' \neq i, j' \in [n_1]\}$. Note that if $\mathbf{X}_\Omega$ is $+1$ on all entries (i,j) with $i > k-2$ then $\mathbf{X} \notin E$ because for every $\mathbf{X}' \in \widetilde{\mathcal{X}}$, $\mathbf{X}'_\Omega = \mathbf{X}_\Omega$. Consequently,

$$|E| \leq \frac{|\Omega|}{n_1(n_2-k+2)} \leq \frac{m}{n_1(n_2-k)}. \tag{62}$$

Plugging Eq. (62) into Eq. (61) we complete the proof of Theorem 8. ■

D. Some concentration inequalities

Lemma 12 ((Laurent and Massart, 2000)) *Let $X \sim \chi^2_d$. Then with probability $\geq 1-2\delta$ the following holds:*

$$-2\sqrt{d\log(1/\delta)} \leq X - d \leq 2\sqrt{d\log(1/\delta)} + 2\log(1/\delta). \tag{63}$$

Lemma 13 *Let $X_1, \cdots, X_n \sim \mathcal{N}(0, \sigma^2)$. Then with probability $\geq 1-\delta$ the following holds:*

$$\max_i |X_i| \leq \sigma\sqrt{2\log(2n/\delta)}. \tag{64}$$

Lemma 14 ((Vershynin, 2010)) *Let $\mathbf{X}$ be an $n \times t$ random matrix with i.i.d. standard Gaussian random entries. If $t < n$ then for every $\epsilon \geq 0$ with probability $\geq 1 - 2\exp(-\epsilon^2/2)$ the following holds:*

$$\sqrt{n} - \sqrt{t} - \epsilon \leq \sigma_{\min}(\mathbf{X}) \leq \sigma_{\max}(\mathbf{X}) \leq \sqrt{n} + \sqrt{t} + \epsilon. \tag{65}$$

Lemma 15 (Noncommutative Bernstein Inequality, (Gross et al., 2010; Recht, 2011)) *Let $\mathbf{X}_1, \cdots, \mathbf{X}_m$ be independent zero-mean square $n \times n$ random matrices. Suppose $\rho_k^2 = \max(\|\mathbb{E}[\mathbf{X}_k\mathbf{X}_k^\top]\|_2, \|\mathbb{E}[\mathbf{X}_k^\top\mathbf{X}_k]\|_2)$ and $\|\mathbf{X}_k\|_2 \leq M$ with probability 1 for all k. Then for any $t > 0$,*

$$\Pr\left[\left\|\sum_{k=1}^m \mathbf{X}_k\right\|_2 > t\right] \leq 2n\exp\left(-\frac{t^2/2}{\sum_{k=1}^m \rho_k^2 + Mt/3}\right). \tag{66}$$

References

Dimitris Achlioptas and Frank McSherry. Fast computation of low-rank matrix approximations. *Journal of the ACM*, 2007.

Dimitris Achlioptas, Zohar Karnin, and Edo Liberty. Near-optimal entrywise sampling for data matrices. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2013.

Nima Anari, Shayan Oveis Gharan, and Alireza Rezaei. Monte carlo markov chain algorithms for sampling strongly rayleigh distributions and determinantal point processes, 2016.

Laura Balzano, Robert Nowak, and Waheed Bajwa. Column subset selection with missing data. In *Proceedings of NIPS Workshop on Low-Rank Methods for Large-Scale Machine Learning*, 2010a.

Laura Balzano, Benjamin Recht, and Robert Nowak. High-dimensional matched subspace detection when data are missing. In *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, 2010b.

Srinadh Bhojanapalli, Prateek Jain, and Sujay Sanghavi. Tighter low-rank approximation via sampling the leveraged element. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2015.

Jacob Bien, Ya Xu, and Michael Mahoney. CUR from a sparse optimization viewpoint. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2010.

Christos Boutsidis and David P Woodruff. Optimal CUR matrix decompositions. In *Proceedings of Annual ACM Symposium on the Theory of Computing (STOC)*, 2014.

Christos Boutsidis, Michael Mahoney, and Petros Drineas. An improved approximation algorithm for the column subset selection problem. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2009.

Christos Boutsidis, Petros Drineas, and Malik Magdon-Ismail. Near-optimal column-based matrix reconstruction. *SIAM Journal on Computing*, 43(2):687–717, 2014.

Emmanuel J Candes and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

Tony F Chan. Rank revealing QR factorizations. *Linear Algebra and Its Applications*, 88: 67–82, 1987.

Yudong Chen, Srinadh Bhojanapalli, Sujay Sanghavi, and Rachel Ward. Completing any low-rank matrix, provably. *arXiv preprint arXiv:1306.2979*, 2013.

Michael B Cohen, Yin Tat Lee, Cameron Musco, Christopher Musco, Richard Peng, and Aaron Sidford. Uniform sampling for matrix approximation. In *Proceedings of Annual Innovations in Theoretical Computer Science (ITCS)*, 2015.

Amit Deshpande and Luis Rademacher. Efficient volume sampling for row/column subset selection. In *Proceedings of Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2010.

Amit Deshpande and Santosh Vempala. Adaptive sampling and fast low-rank matrix approximation. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 292–303. 2006.

Amit Deshpande, Luis Rademacher, Santosh Vempala, and Grant Wang. Matrix approximation and projective clustering via volume sampling. *Theory of Computing*, 2:225–247, 2006.

Petros Drineas, Ravi Kannan, and Michael Mahoney. Fast monte carlo algorithms for matrices I: Approximating matrix multiplication. *SIAM Journal on Computing*, 36(1): 132–157, 2006a.

Petros Drineas, Ravi Kannan, and Michael W Mahoney. Fast monte carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition. *SIAM Journal on Computing*, 36(1):184–206, 2006b.

Petros Drineas, Ravi Kannan, and Michael W Mahoney. Fast monte carlo algorithms for matrices II: Computing a low-rank approximation to a matrix. *SIAM Journal on Computing*, 36(1):158–183, 2006c.

Petros Drineas, Michael W Mahoney, and S Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2):844–881, 2008.

Petros Drineas, Malik Magdon-Ismail, Michael W Mahoney, and David P Woodruff. Fast approximation of matrix coherence and statistical leverage. *The Journal of Machine Learning Research*, 13(1):3475–3506, 2012.

Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast monte-carlo algorithms for finding low-rank approximations. *Journal of the ACM*, 51(6):1025–1041, 2004.

David Gross, Yi-Kai Liu, Steven T Flammia, Stephen Becker, and Jens Eisert. Quantum state tomography via compressed sensing. *Physical Review Letters*, 105(15):150401, 2010.

Ming Gu and Stanley C Eisenstat. Efficient algorithms for computing a strong rank-revealing QR factorization. *SIAM Journal on Scientific Computing*, 17(4):848–869, 1996.

Jarvis Haupt, Rui M Castro, and Robert Nowak. Distilled sensing: Adaptive sampling for sparse detection and estimation. *IEEE Transactions on Information Theory*, 57(9): 6222–6235, 2011.

The international HapMap consortium. The international HapMap project. *Nature*, 437: 789–796, 2003.

Asif Javed, Petros Drineas, Michael Mahoney, and Peristera Paschou. Efficient genomewide selection of PCA-correlated tSNPs for genotype imputation. *Annals of Human Genetics*, 75(6):707–722, 2011.

Xiayi Ke and Lon Cardon. Efficient selective screening of haplotype tag SNPs. *Bioinformatics*, 19(2):287–288, 2003.

Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, 2010.

Mladen Kolar, Sivaraman Balakrishnan, Alessandro Rinaldo, and Aarti Singh. Minimax localization of structural information in large noisy matrices. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2011.

Akshay Krishnamurthy and Aarti Singh. Low-rank matrix and tensor completion via adaptive sampling. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2013.

Akshay Krishnamurthy and Aarti Singh. On the power of adaptivity in matrix completion and approximation. *arXiv preprint arXiv:1407.3619*, 2014.

Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *The Annals of Statistics*, 28(5):1302–1338, 2000.

Karim Lounici, Massimiliano Pontil, Sara Van De Geer, and Alexandre B Tsybakov. Oracle inequalities and optimal inference under group sparsity. *The Annals of Statistics*, pages 2164–2204, 2011.

Sofia Mosci, Lorenzo Rosasco, Matteo Santoro, Alessandro Verri, and Silvia Villa. Solving structured sparsity regularization with proximal methods. In *Proceedings of European Conference on Machine Learning (ECML)*, 2010.

Peristera Paschou, Michael Mahoney, Asif Javed, Judith Kidd, Andrew Pakstis, Sheng Gu, Kenneth Kidd, and Petros Drineas. Intra- and interpopulation genotype reconstruction from tagging SNPs. *Genome Research*, 17(1):96–107, 2007.

Benjamin Recht. A simpler approach to matrix completion. *The Journal of Machine Learning Research*, 12:3413–3430, 2011.

Joel Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.

Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

Shusen Wang and Zhihua Zhang. Improving CUR matrix decomposition and the nyström approximation via adaptive sampling. *The Journal of Machine Learning Research*, 14(1): 2729–2769, 2013.

Miao Xu, Rong Jin, and Zhi-Hua Zhou. CUR algorithm for partially observed matrices. In *Proceedings of International Conference on Machine Learning (ICML)*, 2015.

Yi Yang and Hui Zou. A fast unified algorithm for solving group-lasso penalize learning problems. *Statistics and Computing*, pages 1–13, 2014.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

Journal of Machine Learning Research 18 (2018) 1-55 Submitted 7/15; Revised 1/18; Published 4/18

A Study of the Classification of Low-Dimensional Data with Supervised Manifold Learning

Elif Vural* velif@metu.edu.tr
Department of Electrical and Electronics Engineering
Middle East Technical University
Ankara, 06800, Turkey

Christine Guillemot christine.guillemot@inria.fr
Centre de Recherche INRIA Bretagne Atlantique
Campus Universitaire de Beaulieu
35042 Rennes, France

Editor: Gert Lanckriet

Abstract

Supervised manifold learning methods learn data representations by preserving the geometric structure of data while enhancing the separation between data samples from different classes. In this work, we propose a theoretical study of supervised manifold learning for classification. We consider nonlinear dimensionality reduction algorithms that yield linearly separable embeddings of training data and present generalization bounds for this type of algorithms. A necessary condition for satisfactory generalization performance is that the embedding allow the construction of a sufficiently regular interpolation function in relation with the separation margin of the embedding. We show that for supervised embeddings satisfying this condition, the classification error decays at an exponential rate with the number of training samples. Finally, we examine the separability of supervised nonlinear embeddings that aim to preserve the low-dimensional geometric structure of data based on graph representations. The proposed analysis is supported by experiments on several real data sets.

Keywords: Manifold learning, dimensionality reduction, classification, out-of-sample extensions, RBF interpolation

1. Introduction

In many data analysis problems, data samples have an intrinsically low-dimensional structure although they reside in a high-dimensional ambient space. The learning of low-dimensional structures in collections of data has been a well studied topic of the last two decades (Tenenbaum et al., 2000), (Roweis and Saul, 2000), (Belkin and Niyogi, 2003), (He and Niyogi, 2004), (Donoho and Grimes, 2003), (Zhang and Zha, 2005). Following these works, many classification methods have been proposed in the recent years to apply such manifold learning techniques to learn classifiers that are adapted to the geometric structure of low-dimensional data (Hua et al., 2012), (Yang et al., 2011), (Zhang et al., 2012), (Sugiyama, 2007), (Raducanu and Dornaika, 2012). The common approach in such works is to learn a data representation that enhances the between-class separation while preserving the intrinsic low-dimensional structure of data. While many efforts have focused on the practical aspects of learning such supervised embeddings for training data, the generalization performance of these methods as supervised classification algorithms has not been investigated much yet. In this work, we aim to study nonlinear supervised dimensionality reduction methods and present performance bounds based on the properties of the embedding and the interpolation function used for generalizing the embedding.

Several supervised manifold learning methods extend the Laplacian eigenmaps algorithm (Belkin and Niyogi, 2003), or its linear variant LPP (He and Niyogi, 2004) to the classification problem. The algorithms proposed by Hua et al. (2012), Yang et al. (2011), Zhang et al. (2012) provide a supervised extension of the LPP algorithm and learn a linear projection that preserves the proximity of neighboring samples from the same class, while increasing the distance between nearby samples from different classes. The method by Sugiyama (2007) proposes an adaptation of the Fisher metric for linear manifold learning, which is in fact shown to be equivalent to the above methods by Yang et al. (2011), Zhang et al. (2012). In (Li et al., 2013), (Cui and Fan, 2012), (Wang and Chen, 2009), some other similar Fisher-based linear manifold learning methods are proposed. In (Raducanu and Dornaika, 2012) a method relying on a similar formulation as in (Hua et al., 2012), (Yang et al., 2011), (Zhang et al., 2012) is presented, which, however, learns a nonlinear embedding. The main advantage of linear dimensionality reduction methods over nonlinear ones is that the generalization of the learnt embedding to novel (initially unavailable) samples is straightforward. However, nonlinear manifold learning algorithms are more flexible as the possible data representations they can learn belong to a wider family of functions, e.g., one can always find a nonlinear embedding to make training samples from different classes linearly separable. On the other hand, when a nonlinear embedding is used, one must also determine a suitable interpolation function to generalize the embedding to new samples, and the choice of the interpolator is critical for the classification performance.

The common effort in all of these supervised dimensionality reduction methods is to learn an embedding that increases the separation between different classes, while preserving the geometric structure of data. It is interesting to note that supervised manifold learning methods achieve separability by reducing the dimension of data, while kernel methods in traditional classifiers achieve this by increasing the dimension of data. Meanwhile, making training data linearly separable in supervised manifold learning does not mean much only by itself. Assuming that the data are sampled from a continuous distribution (hence two samples coincide with 0 probability), it is almost always possible to separate a discrete set of samples from different classes with a nonlinear embedding, e.g., even with a simple embedding such as the one mapping each sample to a vector encoding its class label. What actually matters is how the embedding generalizes to test data, i.e., where the test samples will be mapped to in the low-dimensional domain of embedding and how well the performance will be. The generalization for test data is straightforward for kernel methods, it is determined by the underlying main algorithm. However, in nonlinear supervised manifold learning, this question has rather been overlooked so far. In this work we aim to fill this gap and look into the generalization capabilities of supervised manifold learning algorithms. We study the conditions that must be satisfied by the embedding of the training samples and the interpolation function for satisfactory generalization of the classifier. We then ex-

∗. Most part of the work was performed while the first author was at INRIA.

amine the rates of convergence of supervised manifold learning algorithms that satisfy these conditions.

In Section 2, we consider arbitrary supervised manifold learning algorithms that compute a linearly separable embedding of training samples. We study the generalization capability of such algorithms for two types of out-of-sample interpolation functions. We first consider arbitrary interpolation functions that are Lipschitz-continuous on the support of each class, and then focus on out-of-sample extensions with radial basis function (RBF) kernels, which is a popular family of interpolation functions. For both types of interpolators, we derive conditions that must be satisfied by the embedding of the training samples and the regularity of the interpolation function that generalizes the embedding to test samples, when a nearest neighbor or linear classifier is used in the low-dimensional domain of embedding. These conditions enforce the Lipschitz constant of the interpolator to be sufficiently small, in comparison with the separation margin between training samples from different classes in the low-dimensional domain of embedding. The practical value of these results resides in their implications about what must really be taken into account when designing a supervised dimensionality reduction algorithm: Achieving a good separation margin does not suffice by itself; the geometric structure must also be preserved so as to ensure that a sufficiently regular interpolator can be found to generalize the embedding to the whole ambient space. We then particularly consider Gaussian RBF kernels and show the existence of an optimal value for the kernel scale by studying the condition in our main result that links the separation with the Lipschitz constant of the kernel.

Our results in Section 2 also provide bounds on the rate of convergence of the classification error of supervised embeddings. We show that the misclassification error probability decays at an exponential rate with the number of samples, provided that the interpolation function is sufficiently regular with respect to the separation margin of the embedding. These convergence rates are higher than those reported in previous results on RBF networks (Niyogi and Girosi, 1996), (Lin et al., 2014), (Hernández-Aguirre et al., 2002), and regularized least-squares regression algorithms (Caponnetto and De Vito, 2007), (Steinwart et al., 2009). The essential difference between our results and such previous works is that those assume a general setting and do not focus on a particular data model, whereas our results are rather relevant to settings where the support of each class admits some certain structure, so as to allow the existence of an interpolator that is sufficiently regular on the support of each class. Moreover, in contrast with these previous works, our bounds are independent of the ambient space dimension and vary only with the intrinsic dimensions of the class supports as they characterize the error in terms of the covering numbers of the supports.

The results in Section 2 assume an embedding that makes training samples from different classes linearly separable. Even if most nonlinear dimensionality reduction methods are observed to yield separable embeddings in practice, we aim to verify this theoretically in Section 3. In particular, we focus on the nonlinear version of the supervised Laplacian eigenmaps embeddings (Raducanu and Dornaika, 2012), (Hua et al., 2012), (Yang et al., 2011), (Zhang et al., 2012). Supervised Laplacian eigenmaps methods embed the data with the eigenvectors of the linear combination of two graph Laplacian matrices that encode the links between neighboring samples from the same class and different classes. In such a data representation, the coordinates of neighboring data samples change slowly within the same

class and rapidly across different classes. We study the conditions for the linear separability of these embeddings and characterize their separation margin in terms of some graph and algorithm parameters.

In Section 4, we evaluate our results with experiments on several real data sets. We study the implications of the condition derived in Section 2 on the separability margin - interpolator regularity tradeoff. The experimental comparison of several supervised dimensionality reduction algorithms shows that this compromise between the separation and interpolator regularity can indeed be related to the practical classification performance of a supervised manifold learning algorithm. This suggests that, one can possibly improve the accuracy of supervised dimensionality reduction algorithms by considering more carefully the generalization capability of the embedding during the learning. We then study the variation of the classification performance with parameters such as the sample size, the RBF kernel scale, and the dimension of the embedding, in view of the generalization bounds presented in Section 2. Finally, we conclude in Section 5.

2. Performance Bounds for Supervised Manifold Learning Methods

2.1 Notation and Problem Formulation

Consider a setting with M data classes where the samples of each class $m \in \{1, \dots, M\}$ are drawn from a probability measure ν_m in a Hilbert space H such that ν_m has a bounded support $\mathcal{M}_m \subset H$. Let $\mathcal{X} = \{x_i\}_{i=1}^N \subset H$ be a set of N training samples such that each x_i is drawn from one of the probability measures ν_m, and the samples drawn from each ν_m are independent and identically distributed. We denote the class label of x_i by $C_i \in \{1, 2, \dots, M\}$.

Let $Y = \{y_i\}_{i=1}^N \subset \mathbb{R}^d$ be a d-dimensional embedding of $\mathcal{X}$, where each y_i corresponds to x_i. We consider supervised embeddings such that Y is linearly separable. Linear separability is defined as follows:

Definition 1 *The data representation Y is linearly separable with a margin of $\gamma > 0$, if for any two classes $k, l \in \{1, 2, \dots, M\}$, there exists a separating hyperplane defined by $\omega_{kl} \in \mathbb{R}^d$, $\|\omega_{kl}\| = 1$ and $b_{kl} \in \mathbb{R}$ such that*

$$\begin{aligned} \omega_{kl}^T y_i + b_{kl} &\geq \gamma/2 \qquad \text{if } C_i = k \\ \omega_{kl}^T y_i + b_{kl} &\leq -\gamma/2 \qquad \text{if } C_i = l. \end{aligned} \tag{1}$$

The above definition of separability implies the following. For any given class m, there exists a set of hyperplanes $\{\omega_{mk}\}_{k \neq m} \subset \mathbb{R}^d$, $\|\omega_{mk}\| = 1$, and a set of real numbers $\{b_{mk}\}_{k \neq m} \subset \mathbb{R}$ that separate class m from other classes, such that for all y_i of class $C_i = m$

$$\omega_{mk}^T y_i + b_{mk} > \gamma/2, \quad \forall k \neq m \tag{2}$$

and for all y_i of class $C_i \neq m$, there exists a k such that

$$\omega_{mk}^T y_i + b_{mk} < -\gamma/2. \tag{3}$$

These hyperplanes are obtained by setting $\omega_{km} = -\omega_{mk}$, $b_{km} = -b_{mk}$.

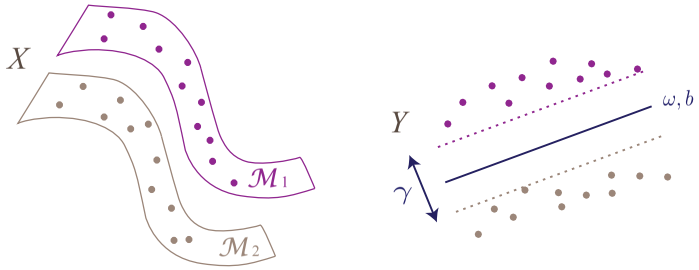


Figure 1: Illustration of a linearly separable embedding. Data in X are sampled from two different classes with supports $\mathcal{M}_1$, $\mathcal{M}_2$. The samples X are mapped to the coordinates Y with a low-dimensional embedding, where the two classes become linearly separable with margin γ with the hyperplane given by ω, b.

Figure 1 shows an illustration of a linearly separable embedding of data samples from two classes. Manifold learning methods typically compute a low-dimensional embedding Y of training data $\mathcal{X}$ in a pointwise manner, i.e., the coordinates y_i are computed only for the initially available training samples x_i. However, in a classification problem, in order to estimate the class label of a new data sample x of unknown class, x needs to be mapped to the low-dimensional domain of embedding as well. The construction of a function $f : H \to \mathbb{R}^d$ that generalizes the learnt embedding to the whole space is known as the out-of-sample generalization problem. Smooth functions are commonly used for out-of-sample interpolation, e.g. as in (Qiao et al., 2013), (Peherstorfer et al., 2011).

Now let x be a test sample drawn from the probability measure ν_m, hence, the true class label of x is m. In our study, we consider two basic classification schemes in the domain of embedding:

Linear classifier. The embeddings of the training samples are used to compute the separating hyperplanes, i.e., the classifier parameters $\{\omega_{mk}\}$ and $\{b_{mk}\}$. Then, mapping x to the low-dimensional domain as $f(x) \in \mathbb{R}^d$, the class label of x is estimated as $\hat{C}(x) = l$ if there exists $l \in \{1, \ldots, M\}$ such that

$$\omega_{lk}^T f(x) + b_{lk} > 0, \quad \forall k \in \{1, \ldots, M\} \setminus \{l\}. \tag{4}$$

Note that the existence of such an l is not guaranteed in general for any x, but for a given x there cannot be more than one l satisfying the above condition. Then x is classified correctly if the estimated class label agrees with the true class label, i.e., $\hat{C}(x) = l = m$.

Nearest neighbor classification. The test sample x is assigned the class label of the closest training point in the domain of embedding, i.e., $\hat{C}(x) = C_{i'}$, where

$$i' = \arg \min_{i=1,\ldots,N} \|y_i - f(x)\|.$$

In the rest of this section, we study the generalization performance of supervised dimensionality reduction methods. We first consider in Section 2.2 interpolation functions that

vary regularly on each class support and we search for a lower bound on the probability of correctly classifying a new data sample in terms of the regularity of f, the separation of the embedding, and the sampling density. Then in Section 2.3, we study the classification performance for a particular type of interpolation functions, namely RBF interpolators, which is one of the most popular ones (Peherstorfer et al., 2011), (Chin and Suter, 2008). We focus particularly on Gaussian RBF interpolators in Section 2.4 and derive some results regarding the existence of an optimal kernel scale parameter. Lastly, we discuss our results in comparison with previous literature in Section 2.5.

In the results in Sections 2.2-2.4, we keep a generic formulation and simply treat the supports $\{\mathcal{M}_m\}$ as arbitrary bounded subsets of H, each of which represents a different data class. Nevertheless, from the perspective of manifold learning, our results are of interest especially when the data is assumed to have an underlying low-dimensional structure. In Section 2.5, we study the implications of our results for the setting where $\mathcal{M}_m$ are low-dimensional manifolds. We then examine how the proposed bounds vary in relation to the intrinsic dimensions of $\{\mathcal{M}_m\}$.

2.2 Out-of-Sample Interpolation with Regular Functions

Let $f : H \to \mathbb{R}^d$ be an out-of-sample interpolation function such that $f(x_i) = y_i$ for each training sample x_i, $i = 1, \ldots, N$. Assume that f is Lipschitz continuous with constant $L > 0$ when restricted to any one of the supports $\mathcal{M}_m$; i.e., for any $m \in \{1, \ldots, M\}$ and any $u, v \in \mathcal{M}_m$

$$\|f(u) - f(v)\| \leq L \|u - v\|,$$

where $\|\cdot\|$ denotes above the ℓ_2-norm if the argument is in $\mathbb{R}^d$, and the norm induced from the inner product in H if the argument is in H.

We will find a relation between the classification accuracy and the number of training samples via the covering number of the supports $\mathcal{M}_m$. Let $B_\epsilon(x) \subset H$ denote an open ball of radius ϵ around x

$$B_\epsilon(x) = \{u \in H : \|x - u\| < \epsilon\}.$$

The covering number $\mathcal{N}(\epsilon, A)$ of a set $A \subset H$ is defined as the smallest number of open balls B_ϵ of radius ϵ whose union contains A (Kulkarni and Posner, 1995)

$$\mathcal{N}(\epsilon, A) = \inf\{k : \exists\, u_1, \ldots, u_k \in H \text{ s.t. } A \subset \bigcup_{i=1}^{k} B_\epsilon(u_i)\}.$$

We assume that the supports $\mathcal{M}_m$ are totally bounded, i.e., $\mathcal{M}_m$ has a finite covering number $\mathcal{N}(\epsilon, \mathcal{M}_m)$ for any $\epsilon > 0$.

We state below a lower bound for the probability of correctly classifying a sample x drawn from ν_m, in terms of the number of training samples drawn from ν_m, the separation of the embedding and the regularity of f.

Theorem 2 *For some ϵ with $0 < \epsilon \leq \gamma/(2L)$, let the training set $\mathcal{X}$ contain at least N_m samples drawn i.i.d. according to a probability measure ν_m such that*

$$N_m \geq \mathcal{N}(\epsilon/2, \mathcal{M}_m).$$

Let Y be an embedding of the training samples $\mathcal{X}$ that is linearly separable with margin larger than γ, and let f be an interpolation function that is Lipschitz continuous with constant L on the support $\mathcal{M}_m$. Then the probability of correctly classifying a test sample x drawn from ν_m independently from the training samples with the linear classifier (4) *is lower bounded as*

$$P\left(\hat{C}(x) = m\right) \geq 1 - \frac{\mathcal{N}(\epsilon/2, \mathcal{M}_m)}{2N_m}.$$

The proof of the theorem is given in Appendix A.1. Theorem 2 establishes a link between the classification performance and the separation of the embedding of the training samples. In particular, due to the condition $\epsilon \leq \gamma/(2L)$, the increase in the separation γ allows a larger value for ϵ, provided that the interpolator regularity is not affected much. This reduces the covering number $\mathcal{N}(\epsilon/2, \mathcal{M}_m)$ in return and increases the probability of correct classification. Similarly, from the condition $\epsilon \leq \gamma/(2L)$, one can also observe that at a given separation γ, a smaller Lipschitz constant L for the interpolation function allows the parameter ϵ to take a larger value. This reduces the covering number $\mathcal{N}(\epsilon/2, \mathcal{M}_m)$ and therefore increases the correct classification probability. Thus, choosing a more regular interpolator at a given separation helps improve the classification performance. If the ϵ parameter is fixed, the Lipschitz constant of the interpolator is allowed to increase only proportionally to the separation margin. The condition that the interpolator must be sufficiently regular in comparison with the separation suggests that increasing the separation too much at the cost of impairing the interpolator regularity may degrade the classifier performance. In the case that the supports $\mathcal{M}_m$ are low-dimensional manifolds, the covering number $\mathcal{N}(\epsilon/2, \mathcal{M}_m)$ increases at a geometric rate with the intrinsic dimension D of the manifold, since a D-dimensional manifold is locally homeomorphic to $\mathbb{R}^D$. Therefore, from the condition on the number of samples, N_m should increase at a geometric rate with D.

In Theorem 2 the probability of misclassification decreases with the number N_m of training samples at a rate of $O(N_m^{-1})$. In the rest of this section, we show that it is in fact possible to obtain an exponential convergence rate with linear and NN-classifiers under certain assumptions. We first present the following lemma.

Lemma 3 *Let $\mathcal{X} = \{x_i\}_{i=1}^N \subset H$ be a set of training samples such that each x_i is drawn i.i.d. from one of the probability measures $\{\nu_m\}_{m=1}^M$. Let x be a test sample randomly drawn according to the probability measure ν_m of class m. Let*

$$A = \{x_i \in \mathcal{X} : x_i \in B_\delta(x), x_i \sim \nu_m\} \tag{5}$$

be the set of samples in $\mathcal{X}$ that are in a δ-neighborhood of x and also drawn from the measure ν_m. Assume that A contains $|A| = Q$ samples. Then

$$P\left(\|f(x) - \frac{1}{Q}\sum_{x_j \in A} f(x_j)\| \leq L\delta + \sqrt{d}\epsilon\right) \geq 1 - 2d\exp\left(-\frac{Q\,\epsilon^2}{2L^2\delta^2}\right). \tag{6}$$

Lemma 3 is proved in Appendix A.2. The inequality in (6) shows that as the number Q of training samples falling in a neighborhood of a test point x increases, the probability of the deviation of $f(x)$ from its average within the neighborhood decreases. The parameter ϵ captures the relation between the amount and the probability of deviation.

When studying the classification accuracy in the main result below, we will use the following generalized definition of the linear separation.

Definition 4 *Let Y be a linearly separable embedding with margin γ such that each pair (k, l) of classes are separated with the hyperplanes given by ω_{kl}, b_{kl} as defined in Definition 1. We say that the linear classifier given by $\{\omega_{kl}\}$, $\{b_{kl}\}$ has a Q-mean separability margin of $\gamma_Q > 0$ if any choice of Q samples $\{y_{k,i}\}_{i=1}^Q \subset Y$ from class k and Q samples $\{y_{l,i}\}_{i=1}^Q \subset Y$ from class l, $l \neq k$, satisfies*

$$\begin{aligned} \omega_{kl}^T\left(\frac{1}{Q}\sum_{i=1}^Q y_{k,i}\right) + b_{kl} &\geq \gamma_Q/2 \\ \omega_{kl}^T\left(\frac{1}{Q}\sum_{i=1}^Q y_{l,i}\right) + b_{kl} &\leq -\gamma_Q/2. \end{aligned} \tag{7}$$

The above definition of separability is more flexible than the one in Definition 1. Clearly, an embedding that is linearly separable with margin γ has a Q-mean separability margin of $\gamma_Q \geq \gamma$ for any Q. As in the previous section, we consider that the test sample x is classified with the linear classifier (4) in the low-dimensional domain, defined with respect to the set of hyperplanes given by $\{\omega_{mk}\}$ and $\{b_{mk}\}$ as in (2) and (3).

In the following result, we show that an exponential convergence rate can be obtained with linear classifiers in supervised manifold learning. We define beforehand a parameter depending on δ, which gives the smallest possible measure of the δ-neighborhood $B_\delta(x)$ of a point x in support $\mathcal{M}_m$.

$$\eta_{m,\delta} := \inf_{x \in \mathcal{M}_m} \nu_m(B_\delta(x)).$$

Theorem 5 *Let $\mathcal{X} = \{x_i\}_{i=1}^N \subset H$ be a set of training samples such that each x_i is drawn i.i.d. from one of the probability measures $\{\nu_m\}_{m=1}^M$. Let Y be an embedding of $\mathcal{X}$ in $\mathbb{R}^d$ that is linearly separable with a Q-mean separability margin larger than γ_Q. For a given $\epsilon > 0$ and $\delta > 0$, let f be a Lipschitz-continuous interpolator such that*

$$L\delta + \sqrt{d}\epsilon \leq \frac{\gamma_Q}{2}. \tag{8}$$

Consider a test sample x randomly drawn according to the probability measure ν_m of class m. If $\mathcal{X}$ contains at least N_m training samples drawn i.i.d. from ν_m such that

$$N_m > \frac{Q}{\eta_{m,\delta}},$$

then the probability of correctly classifying x with the linear classifier given in (4) *is lower bounded as*

$$P\left(\hat{C}(x) = m\right) \geq 1 - \exp\left(-\frac{2\,(N_m\,\eta_{m,\delta} - Q)^2}{N_m}\right) - 2d\exp\left(-\frac{Q\,\epsilon^2}{2L^2\delta^2}\right). \tag{9}$$

Theorem 5 is proved in Appendix A.3. The theorem shows how the classification accuracy is influenced by the separation of the classes in the embedding, the smoothness of the out-of-sample interpolant, and the number of training samples drawn from the density of each class. The condition in (8) points to the tradeoff between the separation and the regularity of the interpolation function. As the Lipschitz constant L of the interpolation function f increases, f becomes less "regular", and a higher separation γ_Q is needed to meet the condition. This is coherent with the expectation that, when f becomes irregular, the classifier becomes more sensitive to the perturbations of the data, e.g., due to noise. The requirement of a higher separation is then for ensuring a larger margin in the linear classifier, which compensates for the irregularity of f. From (8), it is also observed that the separation should increase with the dimension d as well, and also with ϵ, whose increase improves the confidence of the bound (9). Note that the condition in (8) implies also the following: When computing an embedding, it is not advisable to increase the separation of training data unconditionally. In particular, increasing the separation too much may violate the preservation of the geometry and yield an irregular interpolator. Hence, when designing a supervised dimensionality reduction algorithm, one must pay attention to the regularity of the resulting interpolator as much as the enhancement of the separation margin.

Next, we discuss the roles of the parameters Q and δ. The term $\exp(-Q\,\epsilon^2/(2L^2\delta^2))$ in the correct classification probability bound (9) shows that, for fixed δ, the confidence increases with the value of Q. Meanwhile, due to the numerator of the term $\exp(-2\,(N_m\,\eta_{m,\delta}-Q)^2/N_m)$, for a high confidence, the number of samples N_m should also be relatively big with respect to Q to have a high overall confidence. Similarly, at fixed Q, δ should be made smaller to increase the confidence due to the term $\exp(-(Q\,\epsilon^2)/(2L^2\delta^2))$, which then reduces the parameter $\eta_{m,\delta}$ and eventually requires the number of samples N_m to take a sufficiently large value in order to make the term $\exp(-2\,(N_m\,\eta_{m,\delta}-Q)^2/N_m)$ small and have a high confidence. Therefore, these two parameters Q and δ behave in a similar way, and determine the relation between the number of samples and the correct classification probability, i.e., they indicate how large N_m should be in order to have a certain confidence of correct classification.

Theorem 5 studies the setting where the class labels are estimated with a linear classifier in the domain of embedding. We also provide another result below that analyses the performance when a nearest-neighbor classifier is used in the domain of embedding.

Theorem 6 *Let $\mathcal{X}=\{x_i\}_{i=1}^N \subset H$ be a set of training samples such that each x_i is drawn i.i.d. from one of the probability measures $\{\nu_m\}_{m=1}^M$. Let Y be an embedding of $\mathcal{X}$ in $\mathbb{R}^d$ such that*

$$\begin{aligned}&\|y_i-y_j\|<D_\delta,\ \text{if } \|x_i-x_j\|\le\delta \text{ and } C_i=C_j\\&\|y_i-y_j\|>\gamma,\quad \text{if } C_i\neq C_j,\end{aligned}$$

hence, nearby samples from the same class are mapped to nearby points, and samples from different classes are separated by a distance of at least γ in the embedding.

For given $\epsilon>0$ and $\delta>0$, let f be a Lipschitz-continuous interpolation function such that

$$L\delta+\sqrt{d}\epsilon+D_{2\delta}\le\frac{\gamma}{2}. \tag{10}$$

Consider a test sample x randomly drawn according to the probability measure ν_m of class m. If $\mathcal{X}$ contains at least N_m training samples drawn i.i.d. from ν_m such that

$$N_m>\frac{Q}{\eta_{m,\delta}},$$

then the probability of correctly classifying x with nearest-neighbor classification in $\mathbb{R}^d$ is lower bounded as

$$P\left(\hat{C}(x)=m\right)\ge 1-\exp\left(-\frac{2\,(N_m\,\eta_{m,\delta}-Q)^2}{N_m}\right)-2d\exp\left(-\frac{Q\,\epsilon^2}{2L^2\delta^2}\right). \tag{11}$$

Theorem 6 is proved in Appendix A.4. Theorem 6 is quite similar to Theorem 5 and can be interpreted similarly. Unlike in the previous result, the separability condition of the embedding is based on the pairwise distances of samples from different classes here. The condition (10) suggests that the result is useful when the parameter $D_{2\delta}$ is sufficiently small, which requires the embedding to map nearby samples from the same class in the ambient space to nearby points.

In this section, we have characterized the regularity of the interpolation functions via their rates of variation when restricted to the supports $\mathcal{M}_m$. While the results of this section are generic in the sense that they are valid for any interpolation function with the described regularity properties, we have not examined the construction of such functions. In a practical classification problem where one uses a particular type of interpolation functions, one would also be interested in the adaptation of these results to obtain performance guarantees for the particular type of function used. Hence, in the following section we focus on a popular family of smooth functions; radial basis function (RBF) interpolators, and study the classification performance of this particular type of interpolators.

2.3 Out-of-Sample Interpolation with RBF Interpolators

Here we consider an RBF interpolation function $f: H\to\mathbb{R}^d$ of the form

$$f(x)=[f^1(x)\,f^2(x)\,\ldots\,f^d(x)],$$

such that each component f^k of f is given by

$$f^k(x)=\sum_{i=1}^N c_i^k\,\phi(\|x-x_i\|),$$

where $\phi:\mathbb{R}\to\mathbb{R}^+$ is a kernel function, $c_i^k\in\mathbb{R}$ are coefficients, and x_i are kernel centers. In interpolation with RBF functions, it is common to choose the set of kernel centers as the set of available data samples. Hence, we assume that the set of kernel centers $\{x_i\}_{i=1}^N$ is selected to be the same as the set of training samples $\mathcal{X}$. We consider a setting where the coefficients c_i^k are set such that $f(x_i)=y_i$, i.e., f maps each training point in $\mathcal{X}$ to its embedding previously computed with supervised manifold learning.

We consider the RBF kernel ϕ to be a Lipschitz continuous function with constant $L_\phi>0$, hence, for any $u,v\in\mathbb{R}$

$$|\phi(u)-\phi(v)|\le L_\phi\,|u-v|.$$

Also, let $\mathcal{C}$ be an upper bound on the coefficient magnitudes such that for all $k = 1, \ldots, d$

$$\sum_{i=1}^{N} |c_i^k| \leq \mathcal{C}.$$

In the following, we analyze the classification accuracy and extend the results in Section 2.2 to the case of RBF interpolators. We first give the following result, which probabilistically bounds how much the value of the interpolator f at a point x randomly drawn from ν_m may deviate from the average interpolator value of the training points of the same class within a neighborhood of x.

Lemma 7 *Let $\mathcal{X} = \{x_i\}_{i=1}^N \subset H$ be a set of training samples such that each x_i is drawn i.i.d. from one of the probability measures $\{\nu_m\}_{m=1}^M$. Let x be a test sample randomly drawn according to the probability measure ν_m of class m. Let*

$$A = \{x_i \in \mathcal{X} : x_i \in B_\delta(x), x_i \sim \nu_m\} \tag{12}$$

be the set of samples in $\mathcal{X}$ that are in a δ-neighborhood of x and also drawn from the measure ν_m. Assume that A contains $|A| = Q$ samples. Then

$$P\left(\|f(x) - \frac{1}{Q}\sum_{x_j \in A} f(x_j)\| \leq \sqrt{d}\mathcal{C}(L_\phi \delta + \epsilon)\right) \geq 1 - 2N \exp\left(-\frac{(Q-1)\,\epsilon^2}{2L_\phi^2 \delta^2}\right). \tag{13}$$

The proof of Lemma 7 is given in Appendix A.5. The lemma states a result similar to the one in Lemma 3; however, is specialized to the case where f is an RBF interpolator.

We are now ready to present the following main result.

Theorem 8 *Let $\mathcal{X} = \{x_i\}_{i=1}^N \subset H$ be a set of training samples such that each x_i is drawn i.i.d. from one of the probability measures $\{\nu_m\}_{m=1}^M$. Let Y be an embedding of $\mathcal{X}$ in $\mathbb{R}^d$ that is linearly separable with a Q-mean separability margin larger than γ_Q. For a given $\epsilon > 0$ and $\delta > 0$, let f be an RBF interpolator such that*

$$\sqrt{d}\,\mathcal{C}\,(L_\phi \delta + \epsilon) \leq \frac{\gamma_Q}{2}. \tag{14}$$

Consider a test sample x randomly drawn according to the probability measure ν_m of class m. If $\mathcal{X}$ contains at least N_m training samples drawn i.i.d. from ν_m such that

$$N_m > \frac{Q}{\eta_{m,\delta}},$$

then the probability of correctly classifying x with the linear classifier given in (4) is lower bounded as

$$P\left(\hat{C}(x) = m\right) \geq 1 - \exp\left(-\frac{2\,(N_m\,\eta_{m,\delta} - Q)^2}{N_m}\right) - 2N \exp\left(-\frac{(Q-1)\,\epsilon^2}{2L_\phi^2 \delta^2}\right). \tag{15}$$

The theorem is proved in Appendix A.6. The theorem bounds the classification accuracy in terms of the smoothness of the RBF interpolation function and the number of samples. The condition in (14) characterizes the compromise between the separation and the regularity of the interpolator, which depends on the Lipschitz constant of the RBF kernels and the coefficient magnitude. As the Lipschitz constant L_ϕ and the coefficient magnitude parameter $\mathcal{C}$ increase (i.e., f becomes less "regular"), a higher separation γ_Q is required to provide a performance guarantee. When the separation margin of the embedding and the interpolator satisfy the condition in (14), the misclassification probability decays exponentially as the number of training samples increases, similarly to the results in Section 2.2.

Theorem 8 studies the misclassification probability when the class labels in the low-dimensional domain are estimated with a linear classifier. We also present below a bound on the misclassification probability when the nearest-neighbor classifier is used in the low-dimensional domain.

Theorem 9 *Let $\mathcal{X} = \{x_i\}_{i=1}^N \subset H$ be a set of training samples such that each x_i is drawn i.i.d. from one of the probability measures $\{\nu_m\}_{m=1}^M$. Let Y be an embedding of $\mathcal{X}$ in $\mathbb{R}^d$ such that*

$$\begin{aligned} &\|y_i - y_j\| < D_\delta, \ \text{if } \|x_i - x_j\| \leq \delta \text{ and } C_i = C_j \\ &\|y_i - y_j\| > \gamma, \quad \text{if } C_i \neq C_j. \end{aligned}$$

For given $\epsilon > 0$ and $\delta > 0$, let f be an RBF interpolator such that

$$\sqrt{d}\,\mathcal{C}\,(L_\phi \delta + \epsilon) + D_{2\delta} \leq \frac{\gamma}{2}. \tag{16}$$

Consider a test sample x randomly drawn according to the probability measure ν_m of class m. If $\mathcal{X}$ contains at least N_m training samples drawn i.i.d. from ν_m such that

$$N_m > \frac{Q}{\eta_{m,\delta}},$$

then the probability of correctly classifying x with nearest-neighbor classification in $\mathbb{R}^d$ is lower bounded as

$$P\left(\hat{C}(x) = m\right) \geq 1 - \exp\left(-\frac{2\,(N_m\,\eta_{m,\delta} - Q)^2}{N_m}\right) - 2N \exp\left(-\frac{(Q-1)\,\epsilon^2}{2L_\phi^2 \delta^2}\right). \tag{17}$$

Theorem 9 is proved in Appendix A.7. While it provides the exact convergence rate as in Theorem 8, the necessary condition in (16) includes also the parameter $D_{2\delta}$. Hence, if the embedding maps nearby samples from the same class to nearby points, and a compromise is achieved between the separation and the interpolator regularity, the misclassification probability can be upper bounded.

2.4 Optimizing the Scale of Gaussian RBF Kernels

In data interpolation with RBFs, it is known that the accuracy of interpolation is quite sensitive to the choice of the shape parameter for several kernels including the Gaussian kernel (Baxter, 1992). The relation between the shape parameter and the performance of interpolation has been an important problem of interest (Piret, 2007). In this section, we focus on the Gaussian RBF kernel, which is a popular choice for RBF interpolation due to its smoothness and good spatial localization properties. We study the choice of the scale parameter of the kernel within the context of classification.

We consider the RBF kernel given by

$$\phi(r) = e^{-\frac{r^2}{\sigma^2}},$$

where σ is the scale parameter of the Gaussian function. We focus on the condition (14) in Theorem 8

$$\sqrt{d}\,\mathcal{C}\,(L_\phi \delta + \epsilon) \leq \gamma_Q/2,$$

(or equivalently the condition (16) if the nearest neighbor classifier is used), which relates the interpolation function properties with the separation. In particular, for a given separation margin, this condition is satisfied more easily when the term on the left hand side of the inequality is smaller. Thus, in the following, we derive an expression for the left hand side of the above inequality by deriving the Lipschitz constant L_ϕ and the coefficient bound $\mathcal{C}$ in terms of the scale parameter σ of the Gaussian kernel. We then study the scale parameter that minimizes $\sqrt{d}\,\mathcal{C}\,(L_\phi \delta + \epsilon)$.

Writing the condition $f(x_i) = y_i$ in a matrix form for each dimension $k = 1, \ldots, d$, we have

$$\Phi c^k = y^k, \tag{18}$$

where $\Phi \in \mathbb{R}^{N\times N}$ is a matrix whose (i,j)-th entry is given by $\Phi_{ij} = \phi(\|x_i - x_j\|)$, $c^k \in \mathbb{R}^{N\times 1}$ is the coefficient vector whose i-th entry is c_i^k, and $y^k \in \mathbb{R}^{N\times 1}$ is the data coordinate vector giving the k-th dimensions of the embeddings of all samples, i.e., $y_i^k = Y_{ik}$. Assuming that the embedding is computed with the usual scale constraint $Y^TY = I$, we have $\|y^k\| = 1$. The norm of the coefficient vector can then be bounded as

$$\|c^k\| \leq \|\Phi^{-1}\|\|y^k\| = \|\Phi^{-1}\|. \tag{19}$$

In the rest of this section, we assume that the data $\mathcal{X}$ are sampled from the Euclidean space, i.e., $H = \mathbb{R}^n$. We first use a result by Narcowich et al. (1994) in order to bound the norm $\|\Phi^{-1}\|$ of the inverse matrix. From (Narcowich et al., 1994, Theorem 4.1) we get[1]

$$\|\Phi^{-1}\| \leq \beta\, \sigma^{-n} e^{\alpha\sigma^2}, \tag{20}$$

where $\alpha > 0$ and $\beta > 0$ are constants depending on the dimension n and the minimum distance between the training points $\mathcal{X}$ (separation radius) (Narcowich et al., 1994). As the

1. The result stated in (Narcowich et al., 1994, Theorem 4.1) is adapted to our study by taking the measure as $\beta(\rho) = \delta(\rho - \rho_0)$ so that the RBF kernel defined in (Narcowich et al., 1994, (1.1)) corresponds to a Gaussian function as $F(r) = \exp(-\rho_0\, r^2)$. The scale of the Gaussian kernel is then given by $\sigma = {\rho_0}^{-1/2}$.

ℓ_1-norm of the coefficient vector can be bounded as $\|c^k\|_1 \leq \sqrt{N}\|c^k\|$, from (19) one can set the parameter $\mathcal{C}$ that upper bounds the coefficients magnitudes as

$$\mathcal{C} = a\sigma^{-n} e^{\alpha\sigma^2},$$

where $a = \beta\sqrt{N}$.

Next, we derive a Lipschitz constant for the Gaussian kernel $\phi(r)$ in terms of σ. Setting the second derivative of ϕ to zero

$$\frac{d^2\phi}{dr^2} = e^{-\frac{r^2}{\sigma^2}} \left(\frac{4r^2}{\sigma^4} - \frac{2}{\sigma^2}\right) = 0,$$

we get that the maximum value of $|d\phi/dr|$ is attained at $r = \sigma/\sqrt{2}$. Evaluating $|d\phi/dr|$ at this value, we obtain

$$L_\phi = \sqrt{2} e^{-\frac{1}{2}} \sigma^{-1}.$$

Now rewriting the condition (14) of the theorem, we have

$$\sqrt{d}\,\mathcal{C}\,(L_\phi \delta + \epsilon) = a_1 \sigma^{-n-1} e^{\alpha\sigma^2} + a_2 \sigma^{-n} e^{\alpha\sigma^2} \leq \gamma_Q/2,$$

where $a_1 = \sqrt{2d}\, a\, e^{-1/2}\delta$ and $a_2 = \sqrt{d}\, a\, \epsilon$. We thus determine the Gaussian scale parameter σ that minimizes

$$F(\sigma) = a_1 \sigma^{-n-1} e^{\alpha\sigma^2} + a_2 \sigma^{-n} e^{\alpha\sigma^2}.$$

First, notice that as $\sigma \to 0$ and $\sigma \to \infty$, the function $F(\sigma) \to \infty$. Therefore, it has at least one minimum. Setting

$$\frac{dF}{d\sigma} = e^{\alpha\sigma^2} \sigma^{-n-2} \big(2\alpha a_2 \sigma^3 + 2\alpha a_1 \sigma^2 - a_2 n\sigma - a_1(n+1)\big) = 0,$$

we need to solve

$$2\alpha a_2 \sigma^3 + 2\alpha a_1 \sigma^2 - a_2 n\sigma - a_1(n+1) = 0. \tag{21}$$

The leading and the second-degree coefficients are positive, while the first-degree and the constant coefficients are negative in the above cubic polynomial. Then, the sum of the roots is negative and the product of the roots is positive. Therefore, there is one and only one positive root σ_{opt}, which is the unique minimizer of $F(\sigma)$.

The existence of an optimal scale parameter $0 < \sigma_{opt} < \infty$ for the RBF kernel can be intuitively explained as follows. When σ takes too small values, the support of the RBF function concentrated around the training points does not sufficiently cover the whole class supports $\mathcal{M}_m$. This manifests itself in (14) with the increase in the term L_ϕ, which indicates that the interpolation function is not sufficiently regular. This weakens the guarantee that a test sample will be interpolated sufficiently close to its neighboring training samples from the same class and mapped to the correct side of the hyperplane in the linear classifier. On the other hand, when σ increases too much, the stability of the linear system (18) is impaired and the coefficients c increase too much. This results in an overfitting of the interpolator and, therefore, decreases the classification performance. Hence, the analysis in this section provides a theoretical justification of the common knowledge that σ should be set to a sufficiently large value while avoiding overfitting.

Remark: It is also interesting to observe how the optimal scale parameter changes with the number of samples N. In the study (Narcowich et al., 1994), the constants α and β in (20) are shown to vary with the separation radius q at rates $\alpha = O(q^{-2})$ and $\beta = O(q^n)$, where the separation radius q is proportional to the smallest distance between two distinct training samples. Then a reasonable assumption is that the separation radius q should typically decrease at rate $O(N^{-1/n})$ as N increases. Using this relation, we get that α and β should vary at rates $\alpha = O(N^{2/n})$ and $\beta = O(N^{-1})$ with N. It follows that $a = \beta\sqrt{N} = O(N^{-1/2})$, and the parameters a_1, a_2 of the cubic polynomial in (21) also vary with N at rates $a_1 = O(N^{-1/2})$, $a_2 = O(N^{-1/2})$. The equation (21) in σ can then be rearranged as

$$b_3\sigma^3 + b_2\sigma^2 - b_1\sigma - b_0 = 0,$$

such that the constants vary with N at rates $b_3 = O(N^{2/n})$, $b_2 = O(N^{2/n})$, $b_1 = O(1)$, $b_0 = O(1)$. We can then inspect how the roots of this equation change with N as N increases. Since b_3 and b_2 dominate the other coefficients for large N, three real roots will exist if N is sufficiently large, two of which are negative and one is positive. The sum of the pairwise products of the roots is negative and it decays with N at rate $O(N^{-2/n})$, and the product of the roots also decays with N. Then at least two of the roots must decay with N. Meanwhile, the sum of the three roots is $O(1)$ and negative. This shows that one of the negative roots is $O(1)$, i.e., does not decay with N. From the product of three roots, we then observe that the product of the two decaying roots is $O(N^{-2/n})$. However, their sum also decays at the same rate (from the sum of the pairwise products), which is possible if their dominant terms have the same rate and cancel each other. We conclude that both of the decaying roots vary at rate $O(N^{-1/n})$, one of which is the positive root and the optimal value σ_{opt} of the scale parameter.

This analysis shows that the scale parameter of the Gaussian kernel should be adapted to the number of training samples, and a smaller kernel scale must be preferred for a larger number of training samples. In fact, the relation $\sigma_{opt} = O(N^{-1/n})$ is quite intuitive, as the average or typical distance between two samples will also decrease at rate $O(N^{-1/n})$ as the number of samples N increases in an n-dimensional space. Then the above result simply suggests that the kernel scale should be chosen as proportional to the average distance between the training samples.

2.5 Discussion of the Results in Relation with Previous Results

In Theorems 8 and 9, we have presented a result that characterizes the performance of classification with RBF interpolation functions. In particular, we have considered a setting where an RBF interpolator is fitted to each dimension of a low-dimensional embedding where different classes are separable. Our study has several links with RBF networks or least-squares regression algorithms. In this section, we interpret our findings in relation with previously established results.

Several previous works study the performance of learning by considering a probability measure ρ defined on $X \times Y$, where X and Y are two sets. The "label" set Y is often taken as an interval $[-L, L]$. Given a set of data pairs $\{(x_j, y_j)\}_{j=1}^N$ sampled from the distribution

ρ, the RBF network estimates a function $\hat{f}$ of the form

$$\hat{f}(x) = \sum_{i=1}^{R} c_i \, \phi\left(\frac{\|x - t_i\|}{\sigma_i}\right). \tag{22}$$

The number of RBF terms R may be different from the number of samples N in general. The function $\hat{f}$ minimizes the empirical error

$$\hat{f} = \arg\min_f \sum_{j=1}^{N} \left(f(x_j) - y_j\right)^2.$$

The function $\hat{f}$ estimated from a finite collection of data samples is often compared to the regression function (Cucker and Smale, 2002)

$$f_o(x) = \int_Y y \, d\rho(y|x),$$

where $d\rho(y|x)$ is the conditional probability measure on Y. The regression function f_o minimizes the expected risk as

$$f_o = \arg\min_f \int_{X\times Y} \left(f(x) - y\right)^2 d\rho.$$

As the probability measure ρ is not known in practice, the estimate $\hat{f}$ of f_o is obtained from data samples. Several previous works have characterized the performance of learning by studying the approximation error (Niyogi and Girosi, 1996), (Lin et al., 2014)

$$\mathbb{E}[(f_o - \hat{f})^2] = \int_X (f_o(x) - \hat{f}(x))^2 d\rho_X(x), \tag{23}$$

where ρ_X is the marginal probability measure on X. This definition of the approximation error can be adapted to our setting as follows. In our problem the distribution of each class is assumed to have a bounded support, which is a special case of modeling the data with an overall probability distribution ρ. If the supports $\mathcal{M}_m$ are assumed to be nonintersecting, the regression function f_o is given by

$$f_o(x) = \sum_{m=1}^{M} m \, I_m(x),$$

which corresponds to the class labels $m = 1, \ldots, M$, where I_m is the indicator function of the support $\mathcal{M}_m$. It is then easy to show that the approximation error $\mathbb{E}[(f_o - \hat{f})^2]$ can be bounded as a constant times the probability of misclassification $P(\hat{C}(x) \neq m)$. Hence, we can compare our misclassification probability bounds in Section 2.3 with the approximation error in other works.

The study in (Niyogi and Girosi, 1996) assumes that the regression function is an element of the Bessel potential space of a sufficiently high order and that the sum of the coefficients

$|c_i|$ is bounded. It is then shown that for data sampled from $\mathbb{R}^n$, with probability greater than $1-\delta$ the approximation error in (23) can be bounded as

$$\mathbb{E}[(f_o - \hat{f})^2] \leq O\left(\frac{1}{R}\right) + O\left(\sqrt{\frac{Rn\log(RN) - \log(\delta)}{N}}\right), \tag{24}$$

where R is the number of RBF terms.

The analysis by Lin et al. (2014) considers families of RBF kernels that include the Gaussian function. Supposing that the regression function f_o is of Sobolev class W_2^r, and that the number of RBF terms is given by $R = N^{\frac{n}{n+2r}}$ in terms of the number of samples N, the approximation error is bounded as

$$\mathbb{E}[(f_o - \hat{f})^2] \leq O(N^{-\frac{2r}{n+2r}}\log^2(N)). \tag{25}$$

Next, we overview the study by Hernández-Aguirre et al. (2002), which studies the performance of RBFs in a Probably Approximately Correct (PAC)-learning framework. For $X \subset \mathbb{R}^n$, a family $\mathcal{F}$ of measurable functions from X to $[0,1]$ is considered and the problem of approximating a target function f_0 known only through examples with a function in $\hat{f} \in \mathcal{F}$ is studied. The authors use a previous result from (Vidyasagar, 1997) that relates the accuracy of empirical risk minimization to the covering number of $\mathcal{F}$ and the number of samples. Combining this result with the bounds on covering number estimates of Lipschitz continuous functions (Kolmogorov and Tihomirov, 1961), the following result is obtained for PAC function learning with RBF neural networks with Gaussian kernel. Let the coefficients be bounded as $|c_i| \leq A$, a common scale parameter be chosen as $\sigma_i = \sigma$, and $\mathbb{E}[|f_0 - \hat{f}|]$ be computed under a uniform probability measure ρ. Then if the number of samples satisfies

$$N \geq \frac{8}{\varepsilon^2}\log\left(\frac{\sqrt{2}RnA}{e^{-1/2}\sigma\zeta}\right), \tag{26}$$

an approximation of the target function is obtained with accuracy parameter ε and confidence parameter ζ:

$$P(\mathbb{E}[|f_0 - \hat{f}|] > \varepsilon) \leq \zeta. \tag{27}$$

In the above expression, the expectation is over the test samples, whereas the probability is over the training samples; i.e., over all possible distributions of training samples, the probability of having the average approximation error larger than ε is bounded. Note that, our results in Theorems 8 and 9, when translated into the above PAC-learning framework, correspond to a confidence parameter of $\zeta = 0$. This is because the misclassification probability bound of a test sample is valid for any choice of the training samples, provided that the condition (14) (or the condition (16)) holds. Thus, in our result the probability running over the training samples in (27) has no counterpart. When we take $\zeta = 0$, the above result does not provide a useful bound since $N \to \infty$ as $\zeta \to 0$. By contrast, our result is valid only if the conditions (14), (16) on the interpolation function holds. It is easy to show that, assuming nonintersecting class supports $\mathcal{M}_m$, the expression $\mathbb{E}[|f_0 - \hat{f}|]$ is given by a constant times the probability of misclassification. The accuracy parameter ε can then be seen as the counterpart of the misclassification probability upper bound given on the right hand sides of (15) and (17) (the expression subtracted from 1). At fixed N, the dependence

of the accuracy on the kernel scale parameter is monotonic in the bound (26); ε decreases as σ increases. Therefore, this bound does not guide the selection of the scale parameter of the RBF kernel, while the discussion in Section 2.4 (confirmed by the experimental results in Section 4.2) suggests the existence of an optimal scale.

Finally, we mention some results on the learning performance of regularized least squares regression algorithms. In (Caponnetto and De Vito, 2007) optimal rates are derived for the regularized least squares method in a Reproducing Kernel Hilbert Space (RKHS) in the minimax sense. It is shown that, under some hypotheses concerning the data probability measure and the complexity of the family of learnt functions, the maximum error (yielded by the worst distribution) obtained with the regularized least squares method converges at a rate of $O(1/N)$. Next, the work in (Steinwart et al., 2009) shows that, in regularized least squares regression over a RKHS, if the eigenvalues of the kernel integral operator decay sufficiently fast, and if the ℓ_∞-norms of regression functions can be bounded, the error of the classifier converges at a rate of up to $O(1/N)$ with high probability. Steinwart et al. also examine the learning performance in relation with the exponent of the function norm in the regularization term and show that the learning rate is not affected by the choice of the exponent of the function norm.

We now overview the three bounds given in (24), (25), and (26) in terms of the dependence of the error on the number of samples. The results in (24) and (25) provide a useful bound only in the case where the number of samples N is larger than the number of RBF terms R, contrary to our study where we treat the case $R = N$. If it is assumed that N is sufficiently larger than R, the result in (24) predicts a rate of decay of only $O(\sqrt{\log(N)/N})$ in the misclassification probability. The bound in (25) improves with the Sobolev regularity of the regression function; however, the dependence of the error on the number of samples is of a similar nature to the one in (24). Considering ε as a misclassification error parameter in the bound in (26), the error decreases at a rate of $O(N^{-1/2})$ as the number of samples increases. The analysis in (Caponnetto and De Vito, 2007) and (Steinwart et al., 2009) also provide the similar rates of convergence of $O(N^{-1})$. Meanwhile, our results in Theorems 8 and 9 predict an exponential decay in the misclassification probability as the number of samples N increases (under the reasonable assumption that $N_m = O(N)$ for each class m). The reason why we arrive at a more optimistic bound is the specialization of the analysis to the considered particular setting, where the support of each class is assumed to be restricted to a totally bounded region in the ambient space, as well as the assumed relations between the separation margin of the embedding and the regularity of the interpolation function.

Another difference between these previous results and ours is the dependence on the dimension. The results in (24), (25), and (26) predict an increase in the error at the respective rates of $O(\sqrt{n})$, $O(e^{-1/n})$, and $O(\sqrt{\log n})$ with the ambient space dimension n. While these results assume that the data $\mathcal{X} \subset \mathbb{R}^n$ is in an Euclidean space of dimension n, our study assumes the data $\mathcal{X}$ to be in a generic Hilbert space H. The results in Theorems 5-8 involve the dimension d of the low-dimensional space of embedding and does not explicitly depend on the dimension of the ambient Hilbert space H (which could be infinite-dimensional). However, especially in the context of manifold learning, it is interesting to analyze the dependence of our bound on the intrinsic dimension of the class supports $\mathcal{M}_m$.

In order to put the expressions (15), (17) in a more convenient form, let us reduce one parameter by setting $Q = N_m \eta_{m,\delta}/2$. Then the misclassification probability is of

$$O\left(\exp(-N_m \eta_{m,\delta}^2) + N \exp\left(-\frac{N_m\, \eta_{m,\delta}\, \epsilon^2}{L_\phi^2\, \delta^2}\right)\right).$$

We can relate the dependence of this expression on the intrinsic dimension as follows. Since the supports $\mathcal{M}_m$ are assumed to be totally bounded, one can define a parameter Θ that represents the "diameter" of $\mathcal{M}_m$, i.e., the largest distance between any two points on $\mathcal{M}_m$. Then the measure $\eta_{m,\delta}$ of the minimum ball of radius δ in $\mathcal{M}_m$ is of $O((\delta/\Theta)^D)$, where D is the intrinsic dimension of $\mathcal{M}_m$. Replacing this in the above expression gives the probability of misclassification as

$$O\left(\exp\left(-\frac{N_m\, \delta^{2D}}{\Theta^{2D}}\right) + N \exp\left(-\frac{N_m\, \delta^{D-2}\, \epsilon^2}{L_\phi^2\, \Theta^D}\right)\right).$$

This shows that in order to retain the correct classification guarantee, as the intrinsic dimension D grows, the number of samples N_m should increase at a geometric rate with D. In supervised manifold learning problems, data sets usually have a low intrinsic dimension, therefore, this geometric rate of increase can often be tolerated. Meanwhile the dimension of the ambient space is typically high, so that performance bounds independent of the ambient space dimension are of particular interest. Note that generalization bounds in terms of the intrinsic dimension have been proposed in some previous works as well (Bickel and Li, 2007), (Kpotufe, 2011), for the local linear regression and the K-NN regression problems.

3. Separability of Supervised Nonlinear Embeddings

In the results in Section 2, we have presented generalization bounds for classifiers based on linearly separable embeddings. One may wonder if the separability assumption is easy to satisfy when computing structure-preserving nonlinear embeddings of data. In this section, we try to answer this question by focusing on a particular family of supervised dimensionality reduction algorithms, i.e., supervised Laplacian eigenmaps embeddings, and analyze the conditions of separability. We first discuss the supervised Laplacian eigenmaps embeddings in Section 3.1 and then present results in Section 3.2 about the linearly separability of these embeddings.

3.1 Supervised Laplacian Eigenmaps Embeddings

Let $\mathcal{X} = \{x_i\}_{i=1}^N \subset H$ be a set of training samples, where each x_i belongs to one of M classes. Most manifold learning algorithms rely on a graph representation of data. This graph can be a complete graph in some works, in which case an edge exists between each pair of samples. Meanwhile, in some manifold learning algorithms, in order to better capture the intrinsic geometric structure of data, each data sample is connected only to its nearest neighbors in the graph. In this case, an edge exists only between neighboring data samples.

In our analysis, we consider a weighted data graph G each vertex of which represents a point x_i. We write $x_i \sim x_j$, or simply $i \sim j$ if the graph contains an edge between the

data samples x_i, x_j. We denote the edge weight as $w_{ij} > 0$. The weights w_{ij} are usually determined as a positive and monotonically decreasing function of the distance between x_i and x_j in H, where the Gaussian function is a common choice. Nevertheless, we maintain a generic formulation here without making any assumption on the neighborhood or weight selection strategies.

Now let G_w and G_b represent two subgraphs of G, which contain the edges of G that are respectively within the same class and between different classes. Hence, G_w contains an edge $i \sim_w j$ between samples x_i and x_j, if $i \sim j$ and $C_i = C_j$. Similarly, G_b contains an edge $i \sim_b j$ if $i \sim j$ and $C_i \neq C_j$. We assume that all vertices of G are contained in both G_w and G_b; and that G_w has exactly M connected components such that the training samples in each class form a connected component[2]. We also assume that G_w and G_b do not contain any isolated vertices; i.e., each data sample x_i has at least one neighbor in both graphs.

The $N \times N$ weight matrices W_w and W_b of G_w and G_b have entries as follows.

$$W_w(i,j) = \begin{cases} w_{ij} & \text{if } i \sim j \text{ and } C_i = C_j \\ 0 & \text{otherwise} \end{cases}$$

$$W_b(i,j) = \begin{cases} w_{ij} & \text{if } i \sim j \text{ and } C_i \neq C_j \\ 0 & \text{otherwise} \end{cases}$$

Let $d_w(i)$ and $d_b(i)$ denote the degrees of x_i in G_w and G_b

$$d_w(i) = \sum_{j \sim_w i} w_{ij}, \qquad d_b(i) = \sum_{j \sim_b i} w_{ij},$$

and D_w, D_b denote the $N \times N$ diagonal degree matrices given by $D_w(i,i) = d_w(i)$, $D_b(i,i) = d_b(i)$. The normalized graph Laplacian matrices L_w and L_b of G_w and G_b are then defined as

$$L_w := D_w^{-1/2}(D_w - W_w)D_w^{-1/2}, \qquad L_b := D_b^{-1/2}(D_b - W_b)D_b^{-1/2}.$$

Supervised extensions of the Laplacian eigenmaps and LPP algorithms seek a d-dimensional embedding of the data set $\mathcal{X}$, such that each x_i is represented by a vector $y_i \in \mathbb{R}^{d\times 1}$. Denoting the new data matrix as $Y = [y_1\, y_2\, \ldots\, y_N]^T \in \mathbb{R}^{N\times d}$, the coordinates of data samples are computed by solving the problem

$$\text{"Minimize } \mathrm{tr}(Y^T L_w Y) \text{ while maximizing } \mathrm{tr}(Y^T L_b Y)." \tag{28}$$

The reason behind this formulation can be explained as follows. For a graph Laplacian matrix $L = D^{-1/2}(D - W)D^{-1/2}$, where D and W are respectively the degree and the weight matrices, defining the coordinates $Z = D^{-1/2}Y$ normalized with the vertex degrees, we have

$$\mathrm{tr}(Y^T L\, Y) = \mathrm{tr}(Z^T(D - W)Z) = \sum_{i\sim j} \|z_i - z_j\|^2 w_{ij}, \tag{29}$$

2. The straightforward application of common graph construction strategies, like connecting each training sample to its K-nearest neighbors or to its neighbors within a given distance, may result in several disconnected components in a single class in the graph if there is much diversity in that class. However, this difficulty can be easily overcome by introducing extra edges to bridge between graph components that are originally disconnected.

where z_i is the i-th row of Z giving the normalized coordinates of the embedding of the data sample x_i. Hence, the problem in (28) seeks a representation Y that maps nearby samples in the same class to nearby points, while mapping nearby samples from different classes to distant points. In fact, when the samples x_i are assumed to come from a manifold $\mathcal{M}$, the term $y^T L y$ is the discrete equivalent of

$$\int_{\mathcal{M}} \|\nabla f(x)\|^2 dx,$$

where $f : \mathcal{M} \to \mathbb{R}$ is a continuous function on the manifold that extends the one-dimensional coordinates y to the whole manifold. Hence, the term $\mathrm{tr}(Y^T L Y)$ captures the rate of change of the learnt coordinate vectors Y over the underlying manifold. Then, in a setting where the samples of different classes come from M different manifolds $\{\mathcal{M}_m\}_{m=1}^M$, the formulation in (28) looks for a function that has a slow variation on each manifold $\mathcal{M}_m$, while having a fast variation "between" different manifolds.

The supervised learning problem in (28) has so far been studied by several authors with slight variations in their problem formulations. Raducanu and Dornaika (2012) minimize a weighted difference of the within-class and between-class similarity terms in (28) in order to learn a nonlinear embedding. Meanwhile, linear dimensionality reduction methods pose the manifold learning problem as the learning of a linear projection matrix $P \in \mathbb{R}^{d \times n}$; therefore, solve the problem in (28) under the constraint $y_i = P\, x_i$, where $x_i \in \mathbb{R}^{n \times 1}$ and $d < n$. Hua et al. (2012) formulate the problem as the minimization of the difference of the within-class and the between-class similarity terms in (28) as well. Thus, their algorithm can be seen as the linear version of the method by Raducanu and Dornaika (2012). Sugiyama (2007) proposes an adaptation of the Fisher discriminant analysis algorithm to preserve the local structures of data. Data sample pairs are weighted with respect to their affinities in the construction of the within-class and the between-class scatter matrices in Fisher discriminant analysis. Then the trace of the ratio of the between-class and the within-class scatter matrices is maximized to learn a linear embedding. Meanwhile, the within-class and the between-class local scatter matrices are closely related to the two terms in (28) as shown by Yang et al. (2011). The terms $Y^T L_w Y$ and $Y^T L_b Y$, when evaluated under the constraint $y_i = P\, x_i$, become equal to the locally weighted within-class and between-class scatter matrices of the projected data. Cui and Fan (2012) and Wang and Chen (2009) propose to maximize the ratio of the between-class and the within-class local scatters in the learning. Yang et al. (2011) optimize the same objective function, while they construct the between-class graph only on the centers of mass of the classes. Zhang et al. (2012) similarly optimize a Fisher metric to maximize the ratio of the between- and within-class scatters; however, the total scatter is also taken into account in the objective function in order to preserve the overall manifold structure.

All of the above methods use similar formulations of the supervised manifold learning problem and give comparable results. In our study, we base our analysis on the following formal problem definition

$$\min_Y \mathrm{tr}(Y^T L_w Y) - \mu\, \mathrm{tr}(Y^T L_b Y) \text{ subject to } Y^T Y = I, \tag{30}$$

which minimizes the difference of the within-class and the between-class similarity terms as in works such as (Raducanu and Dornaika, 2012) and (Hua et al., 2012). Here I is the

$d \times d$ identity matrix and $\mu > 0$ is a parameter adjusting the weights of the two terms. The condition $Y^T Y = I$ is a commonly used constraint to remove the scale ambiguity of the coordinates. The solution of the problem (30) is given by the first d eigenvectors of the matrix

$$L_w - \mu L_b$$

corresponding to its smallest eigenvalues.

Our purpose in this section is then to theoretically study the linear separability of the learnt coordinates of training data, with respect to the definition of linear separability given in (1). In the following, we determine some conditions on the graph properties and the weight parameter μ that ensure the linear separability. We derive lower bounds on the margin γ and study its dependence on the model parameters. Let us give beforehand the following definitions about the graphs G_w and G_b.

Definition 10 *The volume of the subgraph of G_w that corresponds to the connected component containing samples from class k is*

$$V_k := \sum_{i:\, C_i = k} d_w(i).$$

We define the maximal within-class volume as

$$V_{max} := \max_{k=1,..,M} V_k.$$

The volume of the component of G_b containing the edges between the samples of classes k and l is [3]

$$V_{kl}^b := \sum_{\substack{i \sim_b j \\ C_i = k, C_j = l}} 2\, w_{ij}.$$

We then define the maximal pairwise between-class volume as

$$V_{max}^b := \max_{k \neq l} V_{kl}^b.$$

In a connected graph, the distance between two vertices x_i and x_j is the number of edges in a shortest path joining x_i and x_j. The diameter of the graph is then given by the maximum distance between any two vertices in the graph (Chung, 1996). We define the diameter of the connected component of G_w corresponding to class k as follows.

Definition 11 *For any two vertices x_i and x_j such that $C_i = C_j = k$, consider a within-class shortest path joining x_i and x_j, which contains samples only from class k. Then the diameter D_k of the connected component of G_w corresponding to class k is the maximum number of edges in the within-class shortest path joining any two vertices x_i and x_j from class k.*

Definition 12 *The minimum edge weight within class k is defined as*

$$w_{min,k} := \min_{\substack{i \sim_w j \\ C_i = C_j = k}} w_{ij}.$$

3. In order to keep the analogy with the definition of V_k, a 2 factor is introduced in this expression as each edge is counted only once in the sum.

3.2 Separability Bounds for Two Classes

We now present a lower bound for the linear separability of the embedding obtained by solving (30) in a setting with two classes $C_i \in \{1,2\}$. We first show that an embedding of dimension $d = 1$ is sufficient to achieve linear separability for the case of two classes. We then derive a lower bound on the separation in terms of the graph parameters and the algorithm parameter μ.

Consider a one-dimensional embedding $Y = y = [y_1\, y_2\, \ldots\, y_N]^T \in \mathbb{R}^{N\times 1}$, where $y_i \in \mathbb{R}$ is the coordinate of the data sample x_i in the one-dimensional space. The coordinate vector y is given by the eigenvector of $L_w - \mu L_b$ corresponding to its smallest eigenvalue. We begin with presenting the following result, which states that the samples from the two classes are always mapped to different halves (nonnegative or nonpositive) of the real line.

Lemma 13 *The learnt embedding y of dimension $d = 1$ satisfies*

$$\begin{aligned} y_i &\le 0 \qquad \text{if } C_i = 1 \text{ (or respectively } C_i\text{=2)} \\ y_i &\ge 0 \qquad \text{if } C_i = 2 \text{ (or respectively } C_i\text{=1)} \end{aligned}$$

for any $\mu > 0$ and for any choice of the graph parameters.

Lemma 13 is proved in Appendix B.1. The lemma states that in one-dimensional embeddings of two classes, samples from different classes always have coordinates with different signs. Therefore, the hyperplane given by $\omega = 1$, $b = 0$ separates the data as $\omega^T y_i \le 0$ for $C_i = 1$ and $\omega^T y_i \ge 0$ for $C_i = 2$ (since the embedding is one dimensional, the vector ω is a scalar in this case). However, this does not guarantee that the data is separable with a positive margin $\gamma > 0$. In the following result, we show that a positive margin exists and give a lower bound on it. In the rest of this section, we assume without loss of generality that classes 1 and 2 are respectively mapped to the negative and positive halves of the real axis.

Theorem 14 *Defining the normalized data coordinates $z = D_w^{-1/2} y$, let*

$$z_{1,max} := \max_{i:\, C_i=1} z_i \qquad z_{2,min} := \min_{i:\, C_i=2} z_i$$

denote the maximum and minimum coordinates that classes 1 and 2 are respectively mapped to with a one-dimensional embedding learnt with supervised Laplacian eigenmaps. We also define the parameters

$$\overline{w}_{min} = \min_{k\in\{1,2\}} \frac{w_{min,k}}{D_k}\,, \qquad \beta_i = \frac{d_w(i)}{d_b(i)}\,, \qquad \beta_{max} = \max_i \beta_i\,,$$

where D_k is the diameter of the graph corresponding to class k as defined in Definition 11. Then, if the weight parameter is chosen such that $0 < \mu < \overline{w}_{min}/(\beta_{max} V^b_{max})$, any supervised Laplacian embedding of dimension $d \ge 1$ is linearly separable with a positive margin lower bounded as below:

$$z_{2,min} - z_{1,max} \ge \frac{1}{\sqrt{V_{max}}} \left(1 - \sqrt{\frac{\mu \beta_{max} V^b_{max}}{\overline{w}_{min}}}\right). \tag{31}$$

The proof of Theorem 14 is given in Appendix B.2. The proof is based on a variational characterization of the eigenvector of $L_w - \mu L_b$ corresponding to its smallest eigenvalue, whose elements are then bounded in terms of the parameters of the graph such as the diameters and volumes of its connected components.

Theorem 14 states that an embedding learnt with the supervised Laplacian eigenmaps method makes two classes linearly separable if the weight parameter μ is chosen sufficiently small. In particular, the theorem shows that, for any $0 < \delta < {V_{max}}^{-1/2}$, a choice of the weight parameter μ satisfying

$$0 < \mu \le \frac{\overline{w}_{min}}{\beta_{max}\, V^b_{max}} \left(1 - \sqrt{V_{max}}\, \delta\right)^2$$

guarantees a separation of $z_{2,min} - z_{1,max} \ge \delta$ between classes 1 and 2 at $d = 1$. Here, we use the symbol δ to denote the separation in the normalized coordinates z. In practice, either one of the normalized eigenvectors z or the original eigenvectors y can be used for embedding the data. If the original eigenvectors y are used, due to the relation $y = D_w^{1/2} z$, we can lower bound the separation as $y_{2,min} - y_{1,max} \ge \sqrt{d_{w,min}}(z_{2,min} - z_{1,max})$ where $d_{w,min} = \min_i d_w(i)$. Thus, for any embedding of dimension $d \ge 1$, there exists a hyperplane that results in a linear separation with a margin γ of at least

$$\gamma \ge \sqrt{\frac{d_{w,min}}{V_{max}}} \left(1 - \sqrt{\frac{\mu \beta_{max} V^b_{max}}{\overline{w}_{min}}}\right).$$

Next, we comment on the dependence of the separation on μ. The inequality in (31) shows that the lower bound on the separation $z_{2,min} - z_{1,max}$ has a variation of $O(1 - \sqrt{\mu})$ with the weight parameter μ. The fact that the separation decreases with the increase in μ seems counterintuitive at first; this parameter weights the between-class dissimilarity in the objective function. This can be explained as follows. When μ is high, the algorithm tries to increase the distance between neighboring samples from different classes as much as possible by moving them away from the origin (remember that different classes are mapped to the positive and the negative sides of the real line). However, since the normalized coordinate vector z has to respect the equality $z^T D_w z = 1$, the total squared norm of the coordinates cannot be arbitrarily large. Due to this constraint, setting μ to a high value causes the algorithm to map non-neighboring samples from different classes to nearby coordinates close to the origin. This occurs since the increase in μ reduces the impact of the first term $y^T L_w y$ in the overall objective and results in an embedding with a weaker link between the samples of the same class. This causes a polarization of the data and eventually reduces the separation. Hence, the μ parameter should be carefully chosen and should not take too large values.

Theorem 14 characterizes the separation at $d = 1$ in terms of the distance between the supports of the two classes. Meanwhile, it is also interesting to determine the individual distances of the supports of the two classes to the origin. In the following corollary, we present a lower bound on the distance between the coordinates of any sample and the origin.

Corollary 15 *The distance between the supports of the first and the second classes and the origin in a one-dimensional embedding is lower bounded in terms of the separation between the two classes as*

$$\min\{|z_{1,max}|,\ |z_{2,min}|\} \geq \frac{1}{2}\frac{\beta_{min}}{\beta_{max}}(z_{2,min} - z_{1,max})$$

where

$$\beta_{min} = \min_i \beta_i, \qquad \beta_{max} = \max_i \beta_i.$$

Corollary 15 is proved in Appendix B.3. The proof is based on a Lagrangian formulation of the embedding as a constrained optimization problem, which then allows us to establish a link between the separation and the individual distances of class supports to the origin. The corollary states a lower bound on the portion of the overall separation lying in the negative or the positive sides of the real line. In particular, if the vertex degrees are equal for all samples in G_w and G_b (which is the case, for instance, if all vertices have the same number of neighbors and a constant weight of $w_{ij} = 1$ is assigned to the edges), since $\beta_{min} = \beta_{max}$, the portions of the overall separation in the positive and negative sides of the real line will be equal.

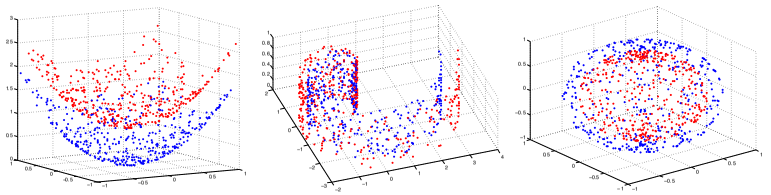
We have examined the linear separability of supervised Laplacian embeddings for the case of two classes in this section. An extension of these results to the case of multiple classes under some assumptions is available in the accompanying technical report (Vural and Guillemot, 2016b).

4. Experimental Results

In this section, we present results on synthetical and real data sets. We compare several supervised manifold learning methods and study their performances in relation with our theoretical results.

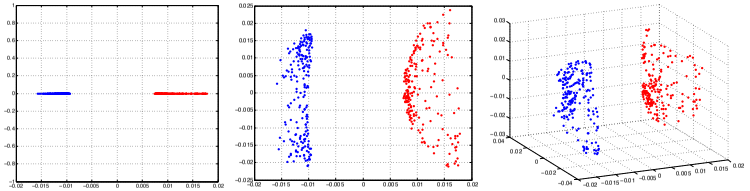
4.1 Separability of Embeddings with Supervised Manifold Learning

We first present results on synthetical data in order to study the embeddings obtained with supervised dimensionality reduction. We test the supervised Laplacian eigenmaps algorithm in a setting with two classes. We generate samples from two nonintersecting and linearly nonseparable surfaces in $\mathbb{R}^3$ that represent two different classes. We experiment on three different types of surfaces; namely, quadratic surfaces, Swiss rolls and spheres. The data sampled from these surfaces are shown in Figure 2. We choose $N = 200$ samples from each class. We construct the graph G_w by connecting each sample to its K-nearest neighbors from the same class, where K is chosen between 20 and 30. The graph G_b is constructed similarly, where each sample is connected to its $K/5$ nearest neighbors from the other class. The graph weights are determined as a Gaussian function of the distance between the samples. The embeddings are then computed by minimizing the objective function in (30). The one-dimensional, two-dimensional, and three-dimensional embeddings obtained for the quadratic surface are shown in Figure 3, where the weight parameter is taken as $\mu = 0.57$



(a) Quadratic surfaces (b) Swissrolls (c) Spheres

Figure 2: Data sampled from two-dimensional synthetical surfaces. Red and blue colors represent two different classes.

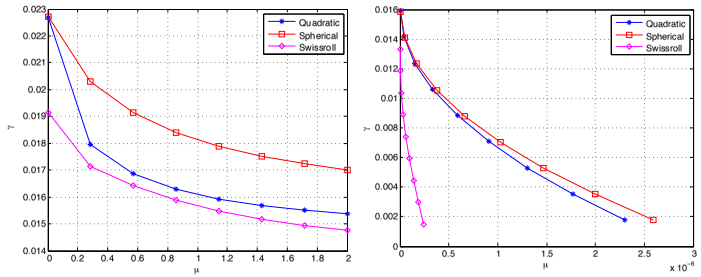


(a) 1-D embedding (b) 2-D embedding (c) 3-D embedding

Figure 3: Supervised Laplacian embeddings of data sampled from quadratic surfaces.

(to have a visually clear embedding for the purpose of illustration). Similar results are obtained on the Swiss roll and the spherical surface. One can observe that the data samples that were initially linearly nonseparable become linearly separable when embedded with the supervised Laplacian eigenmaps algorithm. The two classes are mapped to different (positive or negative) sides of the real line in Figure 3(a) as predicted by Lemma 13. The separation in the 2-D and 3-D embeddings in Figure 3 is close to the separation obtained with the 1-D embedding.

We then compute and plot the separation obtained at different values of μ. Figure 4(a) shows the experimental value of the separation $\gamma = z_{2,min} - z_{1,max}$ obtained with the 1-D embedding for the three types of surfaces. Figure 4(b) shows the theoretical upper bound for μ in Theorem 14 that guarantees a separation of at least γ. Both the experimental value and the theoretical bound for the separation γ decrease with the increase in the parameter μ. This is in agreement with (31), which predicts a decrease of $O(1-\sqrt{\mu})$ in the separation with respect to μ. The theoretical bound for the separation is seen to decrease at a relatively faster rate with μ for the Swiss roll data set. This is due to the particular structure of this data set with a nonuniform sampling density where the sampling is sparser away from the spiral center. The parameter $\overline{w}_{min}$ then takes a small value, which consequently leads to a



(a) Experimental value of the separation γ (b) Theoretical upper bound for μ that guarantees a separation of at least γ

Figure 4: Variation of the separation γ between the two classes with the parameter μ for the synthetic data sets

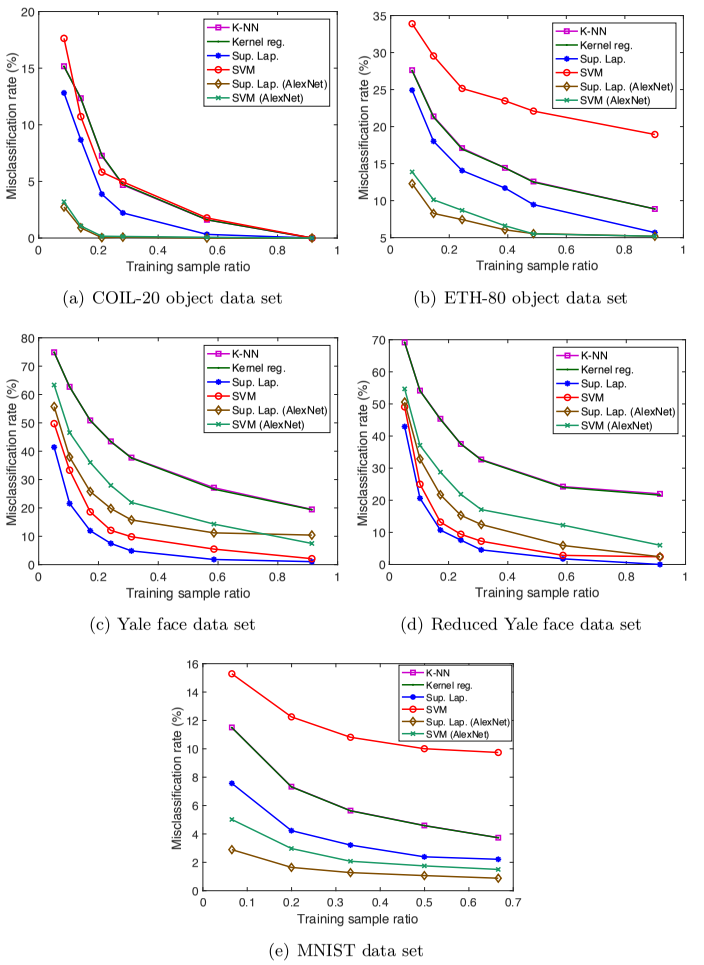
fast rate of decrease for the separation due to (31). Comparing Figures 4(a) and 4(b), one observes that the theoretical bounds for the separation are numerically more pessimistic than their experimental values, which is a result of the fact that our results are obtained with a worst-case analysis. Nevertheless, the theoretical bounds capture well the actual variation of the separation margin with μ.

4.2 Classification Performance of Supervised Manifold Learning Algorithms

We now study the overall performance of classification obtained in a setting with supervised manifold learning, where the out-of-sample generalization is achieved with smooth RBF interpolators. We evaluate the theoretical results of Section 2 on several real data sets: the COIL-20 object database (Nene et al., 1996), the Yale face database (Georghiades et al., 2001), the ETH-80 object database (Leibe and Schiele, 2003), and the MNIST handwritten digit database (LeCun et al., 1998). The COIL-20, Yale face, ETH-80, and MNIST databases contain a total of 1420, 2204, 3280, and 70046 images from 20, 38, 8, and 10 image classes respectively. The images in the COIL-20, Yale and ETH-80 data sets are converted to greyscale, normalized, and downsampled to a resolution of respectively 32×32, 20×17, and 20×20 pixels.

4.2.1 Comparison of Supervised Manifold Learning to Baseline Classifiers

We first compare the performance of supervised manifold learning with some reference classification methods. The performances of SVM, K-NN, kernel regression, and the supervised Laplacian eigenmaps methods are evaluated and compared. Figure 5 reports the results obtained on the COIL-20 data set, the ETH-80 data set, the Yale data set, a subset of the Yale data set consisting of its first 10 classes (reduced Yale data set), and the MNIST data set. The SVM, K-NN, and kernel regression algorithms are applied in the original



(a) COIL-20 object data set (b) ETH-80 object data set

(c) Yale face data set (d) Reduced Yale face data set

(e) MNIST data set

Figure 5: Comparison of the performance of several supervised classification methods

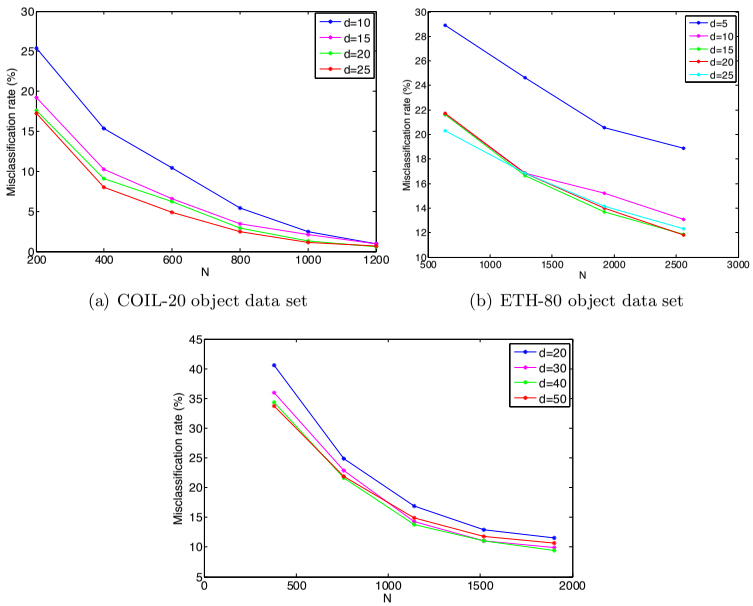
domain and their hyperparameters are optimized with cross-validation. In the supervised Laplacian eigenmaps method, the embedding of the training images into a low-dimensional space is computed. Then, an out-of-sample interpolator with Gaussian RBFs is constructed that maps the training samples to their embedded coordinates as described in Section 2.3.

Test samples are mapped to the low-dimensional domain via the RBF interpolator and the class labels of test samples are estimated via nearest-neighbor classification in the low-dimensional domain. The supervised Laplacian eigenmaps and the SVM methods are also tested over an alternative representation of the image data sets based on deep learning. The images are provided as input to the pretrained AlexNet convolutional neural network proposed in (Krizhevsky et al., 2012), and the activation values at the second fully connected layer are used as the feature representations of the images. The feature representations of training and test images are then provided to the supervised Laplacian eigenmaps and the SVM methods. The plots in Figure 5 show the variation of the misclassification rate of test samples in percentage with the ratio of the number of training samples in the whole data set. The results are the average of 5 repetitions of the experiment with different random choices for the training and test samples.

The results in Figure 5 show that the best results are obtained with the supervised Laplacian eigenmaps algorithm in general. The performances of the algorithms improve with the number of training images as expected. In the COIL-20 and ETH-80 object data sets, the supervised Laplacian eigenmaps and the SVM algorithms yield significantly smaller error when applied to the feature representations of the images obtained with deep learning. Meanwhile, in the Yale face data set these two methods perform better on raw image intensity maps. This can be explained with the fact that the AlexNet model may be more successful in extracting useful features for object images rather than face images as it is trained on many common object and animal classes. It is interesting to compare Figures 5(c) and 5(d). While the performances of the supervised Laplacian eigenmaps and the SVM methods are closer in the reduced version of the Yale database with 10 classes, the performance gap between the supervised Laplacian eigenmaps method and the other methods is larger for the full data set with 38 classes. This can be explained with the fact that the linear separability of different classes degrades as the number of classes increases, thus causing a degradation in the performance of the classifiers in comparison. Meanwhile, the performance of the supervised Laplacian eigenmaps method is not much affected by the increase in the number of classes. The K-NN and kernel regression classifiers are seen to give almost the same performance in the plots in Figure 5. The number of neighbors is set as $K = 1$ for the K-NN algorithm in these experiments, where it has been observed to attain its best performance; and the scale parameter of the kernel regression algorithm is optimized to get the best accuracy, which has turned out to take relatively small values. Hence the performances of these two classifiers practically correspond to that of the nearest-neighbor classifier in the original domain.

4.2.2 Variation of the Error with Algorithm Parameters and Sample Size

We first study the evolution of the classification error with the number of training samples. Figures 6(a)- 6(c) show the variation of the misclassification rate of test samples with respect to the total number of training samples N for the COIL-20, ETH-80 and Yale data sets. Each curve in the figure shows the errors obtained at a different value of the dimension d of the embedding. The decrease in the misclassification rate with the number of training samples is in agreement with the results in Section 2 as expected.



(a) COIL-20 object data set

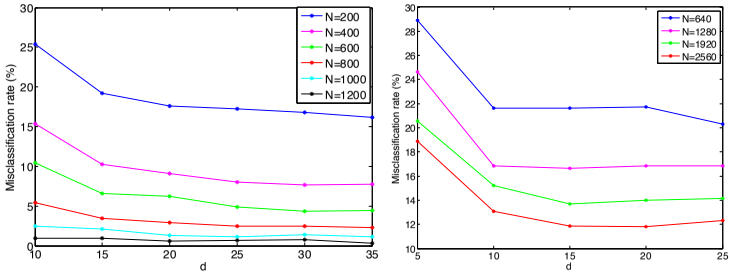
(b) ETH-80 object data set

(c) Yale face data set

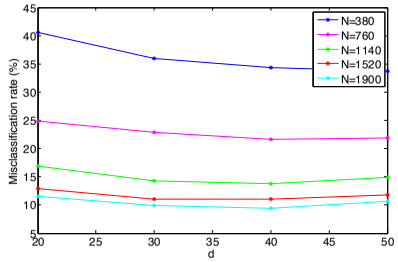
Figure 6: Variation of the misclassification rate with the number of training samples

The results of Figure 6 are replotted in Figure 7, where the variation of the misclassification rate is shown with respect to the dimension d of the embedding at different N values. It is observed that there may exist an optimal value of the dimension that minimizes the misclassification rate. This can be interpreted in light of the conditions (14) and (16) in Theorems 8 and 9, which impose a lower bound on the separability margin γ_Q in terms of the dimension d of the embedding. In the supervised Laplacian eigenmaps algorithm, the first few dimensions are critical and effective for separating different classes. The decrease in the error with the increase in the dimension for small values of d can be explained with the fact that the separation increases with d at small d, thereby satisfying the conditions (14), (16). Meanwhile, the error may stagnate or increase if the dimension d increases beyond a certain value, as the separation does not necessarily increase at the same rate.

We then examine the variation of the misclassification rate with the separation. We obtain embeddings at different separation values γ by changing the parameter μ of the supervised Laplacian eigenmaps algorithm. Figure 8 shows the variation of the misclassification rate with the separation γ. Each curve is obtained at a different value of the scale parameter σ of the RBF kernels. It is seen that the misclassification rate decreases

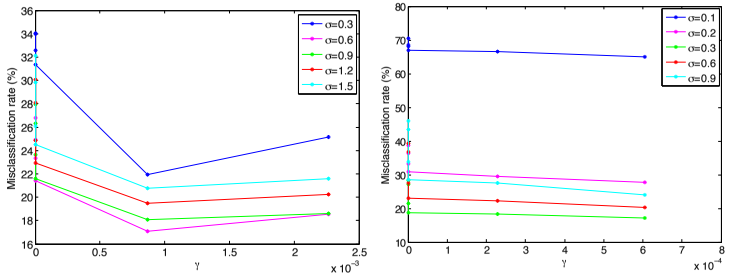


(a) COIL-20 object data set (b) ETH-80 object data set

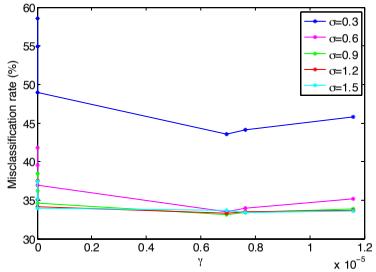


(c) Yale face data set

Figure 7: Variation of the misclassification rate with the dimension of the embedding



(a) COIL-20 object data set (b) ETH-80 object data set

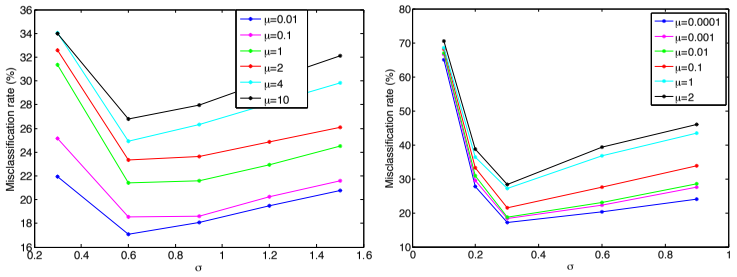


(c) Yale face data set

Figure 8: Variation of the misclassification rate with the separation

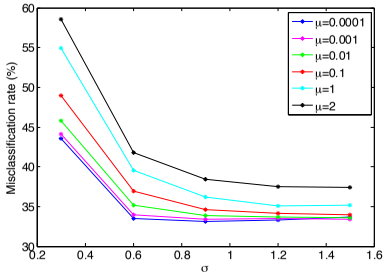
in general with the separation for small γ values. This is in agreement with our results, as the conditions (14), (16) require the separation to be higher than a threshold. On the other hand, the possible increase in the error at relatively large values of the separation is due to the following. These parts of the plots are obtained at very small μ values, which typically result in a deformed embedding with a degenerate geometry. The deformation of structure at too small values of μ may cause the interpolation function to be irregular and hence result in an increase in the error. The tradeoff between the separation and the interpolation function regularity is further studied in Section 4.2.3.

Finally, Figure 9 shows the relation between the misclassification error and the scale parameter σ of the Gaussian RBF kernels. Each curve is obtained at a different value of the μ parameter. The optimum value of the scale parameter minimizing the misclassification error can be observed in most experiments. These results confirm the findings of Section 2.4, suggesting that there exists a unique value of σ that minimizes the left hand side of the conditions (14), (16), which probabilistically guarantee the correct classification of data.



(a) COIL-20 object data set

(b) ETH-80 object data set



(c) Yale face data set

Figure 9: Variation of the misclassification rate with the scale parameter

4.2.3 Performance Analysis of Several Supervised Manifold Learning Algorithms

Next, we compare several supervised manifold learning methods. We aim to interpret the performance differences of different types of embeddings in light of our theoretical results in Section 2.3. First, remember from Theorem 8 that the condition

$$\sqrt{d}\,\mathcal{C}\,(L_\phi \delta + \epsilon) \leq \gamma/2 \tag{32}$$

needs to be satisfied (or, equivalently the condition (16) from Theorem 9) in order for the generalization bounds to hold. This preliminary condition basically states that a compromise must be achieved between the regularity of the interpolation function, captured via the terms $\mathcal{C}$ and L_ϕ, and the separation γ of the embedding of training samples, in order to bound the misclassification error. In other words, increasing the separation too much in the embedding of training samples does not necessarily lead to good classification performance if the interpolation function has poor regularity.

Hence, when comparing different embeddings in the experiments of this section, we define a condition parameter given by

$$\frac{\sqrt{d}\mathcal{C}L_\phi}{\gamma},$$

which represents the ratio of the left and right hand sides of (32) (by fixing the probability parameters δ and ϵ). Setting the Lipschitz constant of the Gaussian RBF kernel as $L_\phi = \sqrt{2}e^{-\frac{1}{2}}\sigma^{-1}$ (see Section 2.4 for details), we can equivalently define the condition parameter as

$$\kappa = \frac{\sqrt{d}\mathcal{C}}{\sigma\gamma} \tag{33}$$

and study this condition parameter for the supervised dimensionality methods in comparison. Note that a smaller condition parameter means that the necessary conditions of Theorems 8 and 9 are more likely to be satisfied, hence hinting at the expectation of a better classification accuracy.

We compare the following supervised embeddings:

- Supervised Laplacian eigenmaps embedding obtained by solving (30):

$$\min_Y \operatorname{tr}(Y^T L_w Y) - \mu \operatorname{tr}(Y^T L_b Y) \text{ subject to } Y^T Y = I.$$

- Fisher embedding[4], obtained by solving

$$\max_Y \frac{\operatorname{tr}(Y^T L_b Y)}{\operatorname{tr}(Y^T L_w Y)}. \tag{34}$$

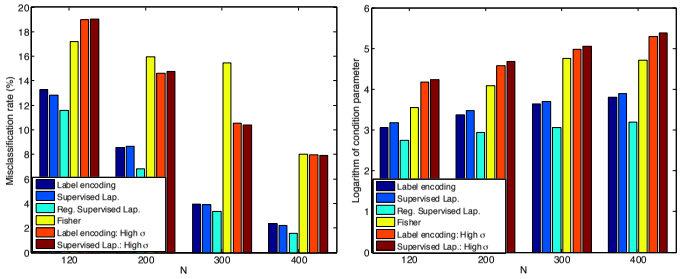
- Label encoding, which maps each data sample to its label vector of the form

$$[0\ 0 \ldots 1 \ldots 0],$$

where the only nonzero entry corresponds to its class.

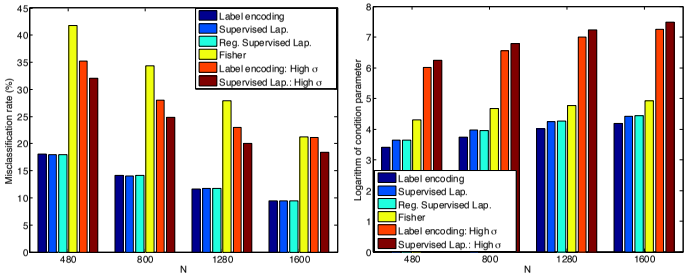
The label encoding method is included in the experiments to provide a reference, which can also be regarded as a degenerate supervised manifold learning algorithm that provides maximal separation between data samples from different classes. In all of the above methods the training samples are embedded into the low-dimensional domain, and test samples are mapped via Gaussian RBF interpolators and assigned labels via nearest neighbor classification in the low-dimensional domain. The scale parameter σ of the RBF kernel is set to a reference value in each data set within the typical range $[0.5, 1]$ where the best accuracy is attained. We have fixed the weight parameter as $\mu = 0.01$ in all setups, and set the dimension of the embedding as equal to the number of classes. In order to study the properties of the interpolation function in relation with the condition parameter in (33), we also test the supervised Laplacian eigenmaps and the label encoding methods under RBF

4. We use a nonlinear version of the formulation in (Wang and Chen, 2009) by removing the constraint that the embedding be given by a linear projection of the data.



(a) (b)

Figure 10: Misclassification rates and the condition parameters of the embeddings for the COIL-20 object data set

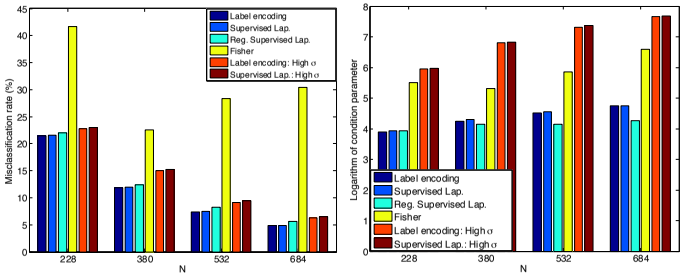


(a) (b)

Figure 11: Misclassification rates and the condition parameters of the embeddings for the ETH-80 object data set

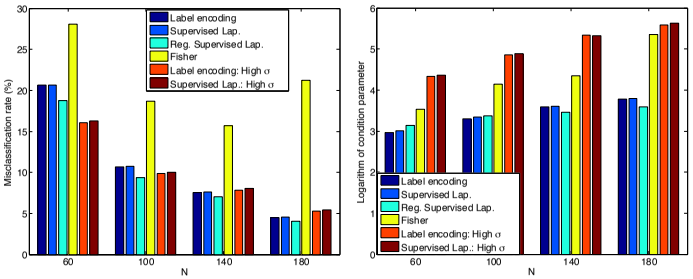
interpolators with high scale parameters, which are chosen as a few times the reference σ value giving the best results. Finally, we also include in the comparisons a regularized version of the supervised Laplacian eigenmaps embedding by controlling the magnitude of the interpolation function.

The results obtained on the COIL-20, ETH-80, Yale and reduced Yale data sets are reported respectively in Figures 10-13. In each figure, panel (a) shows the misclassification rates of the embeddings and panel (b) shows the condition parameters of the embeddings at different total number of training samples (N). The logarithm of the condition parameter is plotted for ease of visualization.



(a) (b)

Figure 12: Misclassification rates and the condition parameters of the embeddings for the Yale face data set



(a) (b)

Figure 13: Misclassification rates and the condition parameters of the embeddings for the reduced Yale face data set

The plots in Figures 10-13 show that the label encoding, supervised Laplacian eigenmaps, and the regularized supervised Laplacian eigenmaps embeddings yield better classification accuracy than the other three methods (supervised Fisher, and the embeddings with high scale parameters) in all experiments, with the only exception of the cases $N = 60$ and $N = 100$ for the reduced Yale data set. Meanwhile, examining the condition parameters of the embeddings, we observe that label encoding, supervised Laplacian eigenmaps, and the regularized supervised Laplacian eigenmaps embeddings always have a smaller condition parameter than the other three methods. This observation confirms the intuition provided by the necessary conditions of Theorems 8 and 9: A compromise between the separation and the interpolator regularity is required for good classification accuracy. The increase in the condition parameter as N increases is since the coefficient bound $\mathcal{C}$ involves a summation over all training samples. The reason why the embeddings with high σ parameters yield better classification accuracy than the other ones in the cases $N = 60$ and $N = 100$ for the reduced Yale data set is that a larger RBF scale helps better cover up the ambient space when the number of training samples is particularly low.

In the COIL-20 and the reduced Yale data sets, the best classification accuracy is obtained with the regularized supervised Laplacian eigenmaps method, while this is also the method having the smallest condition number, except for the smallest two values of N in the reduced Yale data set. In the ETH-80 and Full Yale data sets, the classification accuracy of label encoding attains that of the supervised Laplacian eigenmaps method. The condition parameter of the label encoding embedding is relatively small in these two data sets; in fact, in ETH-80 the label encoding embedding has the smallest condition number among all methods. This may be useful for explaining why this simple classification method has quite favorable performance in this data set. Likewise, if we leave aside the versions of the methods with high-scale interpolators, the Fisher embedding has the highest misclassification rate compared to label encoding, the supervised Laplacian, and the regularized supervised Laplacian embeddings, while it also has the highest condition parameter among these methods. [5]

To conclude, the results in this section suggest that the experimental findings are in agreement with the main results in Section 2.3, justifying the pertinence of the conditions (14) and (16) to classification accuracy, hence suggesting that a balance must be sought between the separability margin of the embedding and the regularity of the interpolation function in supervised manifold learning.

5. Conclusions

Most of the current supervised manifold learning algorithms focus on learning representations of training data, while the generalization properties of these representations have not been understood well yet. In this work, we have proposed a theoretical analysis of the performance of supervised manifold learning methods. We have presented generalization bounds for nonlinear supervised manifold learning algorithms and explored how the classification accuracy relates to several setup parameters such as the linear separation

5. The formulation in (34) has been observed to give highly polarized embeddings in (Vural and Guillemot, 2016a), where the samples of only few classes stretch out along each dimension and all the other classes are mapped close to zero.

margin of the embedding, the regularity of the interpolation function, the number of training samples, and the intrinsic dimensions of the class supports (manifolds). Our results suggest that embeddings of training data with good generalization capacities must allow the construction of sufficiently regular interpolation functions that extend the mapping to new data. We have then examined whether the assumption of linear separability is easy to satisfy for structure-preserving supervised embedding algorithms. We have taken the supervised Laplacian eigenmaps algorithms as reference, and showed that these methods can yield linearly separable embeddings. Providing insight about the generalization capabilities of supervised dimensionality reduction algorithms, our findings can be helpful in the classification of low-dimensional data sets.

Acknowledgments

We would like to thank Pascal Frossard and Alhussein Fawzi for the helpful discussions that contributed to this study.

Appendix A. Proof of the Results in Section 2

A.1 Proof of Theorem 2

Proof Given x, let $x_i \in \mathcal{X}$ be the nearest neighbor of x in $\mathcal{X}$ that is sampled from ν_m

$$i = \arg\min_j \|x - x_j\| \text{ s.t. } x_j \sim \nu_m.$$

Due to the separation hypothesis,

$$\omega_{mk}^T y_i + b_{mk} > \gamma/2, \quad \forall k = 1, \ldots, M-1.$$

We have

$$\begin{aligned} \omega_{mk}^T f(x) + b_{mk} &= \omega_{mk}^T f(x_i) + b_{mk} + \omega_{mk}^T (f(x) - f(x_i)) \\ &\geq \omega_{mk}^T y_i + b_{mk} - \left|\omega_{mk}^T (f(x) - f(x_i))\right| \\ &> \gamma/2 - \|f(x) - f(x_i)\| \ \geq \ \gamma/2 - L\|x - x_i\|. \end{aligned}$$

Then if the condition $L\|x - x_i\| \leq \gamma/2$ is satisfied, from the above inequality we have $\omega_{mk}^T f(x) + b_{mk} > 0$ for all $k = 1, \ldots, M-1$. This gives $\hat{C}(x) = m$ and thus ensures that x is classified correctly.

In the sequel, we lower bound the probability that the distance $\|x - x_i\|$ between x and its nearest neighbor from the same class is smaller than $\gamma/2$. We employ the following result by Kulkarni and Posner (1995). It is demonstrated in the proof of Theorem 1 in (Kulkarni and Posner, 1995) that, if $\mathcal{X}$ contains at least N_m samples drawn i.i.d. from ν_m such that $N_m \geq \mathcal{N}(\epsilon/2, \mathcal{M}_m)$ for some $\epsilon > 0$, then the probability of $\|x - x_i\|$ being larger than ϵ can be upper bounded in terms of the covering number of $\mathcal{M}_m$ as

$$P(\|x - x_i\| > \epsilon) \leq \frac{\mathcal{N}(\epsilon/2, \mathcal{M}_m)}{2N_m}.$$

Therefore, for any ϵ such that $\epsilon \leq \gamma/(2L)$ and $N_m \geq \mathcal{N}(\epsilon/2, \mathcal{M}_m)$, with probability at least $1 - \mathcal{N}(\epsilon/2, \mathcal{M}_m)/(2N_m)$, we have

$$\|x - x_i\| \leq \epsilon \leq \gamma/(2L),$$

thus, the class label of x is correctly estimated as $\hat{C}(x) = m$ due to the above discussion. ■

A.2 Proof of Lemma 3

Proof We first bound the deviation of $f(x)$ from the sample average of f in the neighborhood of x as

$$\left\| f(x) - \frac{1}{Q} \sum_{x_j \in A} f(x_j) \right\| \leq \|f(x) - m_f\| + \left\| \frac{1}{Q} \sum_{x_j \in A} f(x_j) - m_f \right\|, \tag{35}$$

where m_f is the conditional expectation of $f(u)$, given $u \in B_\delta(x)$

$$m_f = \mathbb{E}_u\big[f(u) \,|\, u \in B_\delta(x)\big] = \frac{1}{\nu_m\big(B_\delta(x)\big)} \int_{B_\delta(x)} f(u)\; d\nu_m(u).$$

The first term in (35) can be bounded as

$$\begin{aligned} \|f(x) - m_f\| &= \left\| \frac{1}{\nu_m\big(B_\delta(x)\big)} \int_{B_\delta(x)} \big(f(x) - f(u)\big)\, d\nu_m(u) \right\| \\ &\leq \frac{1}{\nu_m\big(B_\delta(x)\big)} \int_{B_\delta(x)} \|f(x) - f(u)\|\, d\nu_m(u) \leq \frac{1}{\nu_m\big(B_\delta(x)\big)} \int_{B_\delta(x)} L\|x - u\|\, d\nu_m(u) \\ &\leq \frac{1}{\nu_m\big(B_\delta(x)\big)} \int_{B_\delta(x)} L\delta\, d\nu_m(u) = L\delta, \end{aligned} \tag{36}$$

where the second inequality follows from the fact that f is Lipschitz continuous on the support $\mathcal{M}_m$, where the measure ν_m is nonzero.

The second term in (35) is given by

$$\left\| \frac{1}{Q} \sum_{x_j \in A} f(x_j) - m_f \right\| = \left(\sum_{k=1}^{d} \left| \frac{1}{Q} \sum_{x_j \in A} f^k(x_j) - m_f^k \right|^2 \right)^{1/2}, \tag{37}$$

where m_f^k denotes the k-th component of m_f, for $k = 1, \ldots, d$. Consider the random variables $f^k(x_j)$. Defining

$$f^k_{\min} = \inf_{u \in B_\delta(x)} f^k(u), \qquad f^k_{\max} = \sup_{u \in B_\delta(x)} f^k(u),$$

it follows that $f^k_{\max} - f^k_{\min} \leq 2L\delta$ due to the Lipschitz continuity of f. Then from Hoeffding's inequality, we have

$$P\left(\left| \frac{1}{Q} \sum_{x_j \in A} f^k(x_j) - m_f^k \right| \geq \epsilon \right) \leq 2\exp\left(-\frac{2Q\epsilon^2}{(f^k_{\max} - f^k_{\min})^2} \right) \leq 2\exp\left(-\frac{Q\epsilon^2}{2L^2\delta^2} \right).$$

From the union bound, we get that with probability at least $1 - 2d\exp\left(-\frac{Q\epsilon^2}{2L^2\delta^2}\right)$, for all k

$$\left| \frac{1}{Q} \sum_{x_j \in A} f^k(x_j) - m_f^k \right| \leq \epsilon,$$

which yields from (37)

$$\left\| \frac{1}{Q} \sum_{x_j \in A} f(x_j) - m_f \right\| \leq \sqrt{d}\epsilon.$$

Combining this result with the bound in (36), we conclude that with probability at least $1 - 2d\exp\left(-\frac{Q\epsilon^2}{2L^2\delta^2}\right)$

$$\left\| f(x) - \frac{1}{Q} \sum_{x_j \in A} f(x_j) \right\| \leq L\delta + \sqrt{d}\epsilon.$$

■

A.3 Proof of Theorem 5

Proof Given the test sample x and a training sample x_i drawn i.i.d. with respect to ν_m, the probability that x_i lies within a δ-neighborhood of x is given by

$$P(x_i \in B_\delta(x)) = \nu_m(B_\delta(x)) \geq \eta_{m,\delta}.$$

Then, among the N_m samples drawn with respect to ν_m, the probability that $B_\delta(x)$ contains at least Q samples is given by

$$\begin{aligned} P(|A| \geq Q) &= \sum_{q=Q}^{N_m} \binom{N_m}{q} \Big(\nu_m(B_\delta(x))\Big)^q \Big(1 - \nu_m(B_\delta(x))\Big)^{N_m - q} \\ &\geq \sum_{q=Q}^{N_m} \binom{N_m}{q} (\eta_{m,\delta})^q \left(1 - \eta_{m,\delta}\right)^{N_m - q}, \end{aligned}$$

where the set A is defined as in (5). The last expression above is the probability of having at least Q successes out of N_m realizations of a Bernoulli random variable with probability parameter $\eta_{m,\delta}$. This probability can be lower bounded using a tail bound for binomial distributions. We thus have

$$P(|A| \geq Q) \geq 1 - \exp\left(\frac{-2\left(N_m\, \eta_{m,\delta} - Q\right)^2}{N_m} \right),$$

which simply follows from interpreting $|A|$ as the sum of of N_m i.i.d. observations of a Bernoulli distributed random variable and then applying Hoeffding's inequality as shown by Herbrich (1999), under the hypothesis that $N_m > Q/\eta_{m,\delta}$.

Assuming that $B_\delta(x)$ contains at least Q samples, Lemma 3 states that with probability at least

$$1 - 2d\exp\left(-\frac{|A|\epsilon^2}{2L^2\delta^2}\right) \geq 1 - 2d\exp\left(-\frac{Q\epsilon^2}{2L^2\delta^2}\right)$$

the deviation between $f(x)$ and the sample average of its neighbors is bounded as

$$\left\| f(x) - \frac{1}{|A|}\sum_{x_j\in A} f(x_j) \right\| \leq L\delta + \sqrt{d}\epsilon.$$

Hence, with probability at least

$$\left(1 - \exp\left(\frac{-2\left(N_m\,\eta_{m,\delta} - Q\right)^2}{N_m}\right)\right)\left(1 - 2d\exp\left(-\frac{Q\epsilon^2}{2L^2\delta^2}\right)\right)$$
$$\geq 1 - \exp\left(\frac{-2\left(N_m\,\eta_{m,\delta} - Q\right)^2}{N_m}\right) - 2d\exp\left(-\frac{Q\epsilon^2}{2L^2\delta^2}\right)$$

we have

$$\left\| f(x) - \frac{1}{|A|}\sum_{x_j\in A} f(x_j) \right\| \leq L\delta + \sqrt{d}\epsilon. \tag{38}$$

The class label of a test sample x drawn from ν_m is correctly estimated with respect to the classifier (4) if

$$\omega_{mk}^T f(x) + b_{mk} > 0, \quad \forall k = 1,\ldots,M-1,\ k \neq m.$$

If the condition in (38) is satisfied, for all $k \neq m$, we have

$$\begin{aligned}
\omega_{mk}^T f(x) + b_{mk} &= \omega_{mk}^T \frac{1}{|A|}\sum_{x_j\in A} f(x_j) + b_{mk} + \omega_{mk}^T\left(f(x) - \frac{1}{|A|}\sum_{x_j\in A} f(x_j)\right) \\
&\geq \omega_{mk}^T \frac{1}{|A|}\sum_{x_j\in A} f(x_j) + b_{mk} - \left\| f(x) - \frac{1}{|A|}\sum_{x_j\in A} f(x_j)\right\| \\
&> \gamma_Q/2 - \left\| f(x) - \frac{1}{|A|}\sum_{x_j\in A} f(x_j)\right\| \geq \gamma_Q/2 - L\delta - \sqrt{d}\epsilon \geq 0.
\end{aligned}$$

Here, we obtain the second inequality from the hypothesis that the embedding is Q-mean separable with margin larger than γ_Q, which implies that the embedding is also R-mean separable with margin larger than γ_Q, for $R > Q$. Then the last inequality is due to the condition (8) on the interpolation function in the theorem. We thus get that with probability at least

$$1 - \exp\left(\frac{-2\left(N_m\,\eta_{m,\delta} - Q\right)^2}{N_m}\right) - 2d\exp\left(-\frac{Q\epsilon^2}{2L^2\delta^2}\right),$$

$\omega_{mk}^T f(x) + b_{mk} > 0$ for all $k \neq m$, hence, the sample x is correctly classified. This concludes the proof of the theorem. ■

A.4 Proof of Theorem 6

Proof Remember from the proof of Theorem 5 that with probability at least

$$1 - \exp\left(\frac{-2\left(N_m\,\eta_{m,\delta} - Q\right)^2}{N_m}\right) - 2d\exp\left(-\frac{Q\epsilon^2}{2L^2\delta^2}\right)$$

the δ-neighborhood $B_\delta(x)$ of a test sample x from class m contains at least Q samples from class m, and

$$\left\| f(x) - \frac{1}{|A|}\sum_{x_j\in A} f(x_j) \right\| \leq L\delta + \sqrt{d}\epsilon, \tag{39}$$

where A is the set of training samples in $B_\delta(x)$ from class m.

Let $x_i, x_j \in A$ be two training samples from class m in $B_\delta(x)$. As $\|x_i - x_j\| \leq 2\delta$, by the hypothesis on the embedding, we have $\|y_i - y_j\| = \|f(x_i) - f(x_j)\| \leq D_{2\delta}$, which gives

$$\|f(x_i) - \frac{1}{|A|}\sum_{x_j\in A} f(x_j)\| = \left\| \frac{1}{|A|}\sum_{x_j\in A}\left(f(x_i) - f(x_j)\right)\right\| \leq \frac{1}{|A|}\sum_{x_j\in A}\|f(x_i) - f(x_j)\| \leq D_{2\delta}.$$

Then, for any $x_i \in B_\delta(x)$,

$$\begin{aligned}
\|f(x) - f(x_i)\| &= \|f(x) - \frac{1}{|A|}\sum_{x_j\in A} f(x_j) + \frac{1}{|A|}\sum_{x_j\in A} f(x_j) - f(x_i)\| \\
&\leq \|f(x) - \frac{1}{|A|}\sum_{x_j\in A} f(x_j)\| + D_{2\delta}.
\end{aligned}$$

Combining this with (39), we get that with probability at least

$$1 - \exp\left(\frac{-2\left(N_m\,\eta_{m,\delta} - Q\right)^2}{N_m}\right) - 2d\exp\left(-\frac{Q\epsilon^2}{2L^2\delta^2}\right)$$

$B_\delta(x)$ will contain at least Q samples x_i from class m such that

$$\|f(x) - f(x_i)\| \leq L\delta + \sqrt{d}\epsilon + D_{2\delta}. \tag{40}$$

Now, assuming (40), let x_i' be a training sample from another class (other than m). We have

$$\|f(x) - f(x_i')\| \geq \|f(x_i) - f(x_i')\| - \|f(x) - f(x_i)\| > \gamma - (L\delta + \sqrt{d}\epsilon + D_{2\delta}),$$

which follows from (40) and the hypothesis on the embedding that $\|f(x_i) - f(x_i')\| > \gamma$.

It follows from the condition (10) that $\gamma \geq 2L\delta + 2\sqrt{d}\epsilon + 2D_{2\delta}$. Using this in the above equation, we get

$$\|f(x) - f(x_i')\| > L\delta + \sqrt{d}\epsilon + D_{2\delta}.$$

This means that the distance of $f(x)$ to the embedding of any other sample from another class is more than $L\delta + \sqrt{d}\epsilon + D_{2\delta}$, while there are samples from its own class within a distance of $L\delta + \sqrt{d}\epsilon + D_{2\delta}$ to $f(x)$. Therefore, x is classified correctly with nearest-neighbor classification in the low-dimensional domain of embedding. ■

A.5 Proof of Lemma 7

Proof The deviation of each component $f^k(x)$ of the interpolator from the sample average in the neighborhood of x is given by

$$\left| f^k(x) - \frac{1}{Q}\sum_{x_j\in A} f^k(x_j) \right| = \left| \sum_{i=1}^{N} c_i^k \left(\phi(\|x-x_i\|) - \frac{1}{Q}\sum_{x_j\in A}\phi(\|x_j-x_i\|) \right) \right|. \tag{41}$$

We thus proceed by studying the term

$$\phi(\|x-x_i\|) - \frac{1}{Q}\sum_{x_j\in A}\phi(\|x_j-x_i\|), \tag{42}$$

which will then be used in the above expression to arrive at the stated result.

Now let $x_i \in \mathcal{X}$ be any training sample. In order to study the term in (42), we first look at

$$\left| \phi(\|x-x_i\|) - \mathbb{E}_u\big[\phi(\|u-x_i\|)\,|\,u\in B_\delta(x)\big] \right|,$$

where $\mathbb{E}_u\big[\phi(\|u-x_i\|)\,|\,u\in B_\delta(x)\big]$ denotes the conditional expectation of $\phi(\|u-x_i\|)$ over u, given $u\in B_\delta(x)$. The conditional expectation is given by

$$\mathbb{E}_u\big[\phi(\|u-x_i\|)\,|\,u\in B_\delta(x)\big] = \frac{1}{\nu_m\big(B_\delta(x)\big)}\int_{B_\delta(x)} \phi(\|u-x_i\|)\, d\nu_m(u).$$

We have

$$\begin{aligned}
&\left| \phi(\|x-x_i\|) - \mathbb{E}_u\big[\phi(\|u-x_i\|)\,|\,u\in B_\delta(x)\big] \right| \\
&\qquad = \frac{1}{\nu_m\big(B_\delta(x)\big)} \left| \int_{B_\delta(x)} \big(\phi(\|x-x_i\|) - \phi(\|u-x_i\|)\big)\, d\nu_m(u) \right| \\
&\qquad \le \frac{1}{\nu_m\big(B_\delta(x)\big)} \int_{B_\delta(x)} \big|\phi(\|x-x_i\|) - \phi(\|u-x_i\|)\big|\, d\nu_m(u).
\end{aligned}$$

The term in the integral is bounded as

$$\big|\phi(\|x-x_i\|) - \phi(\|u-x_i\|)\big| \le L_\phi \big|\|x-x_i\| - \|u-x_i\|\big| \le L_\phi \|x-u\|.$$

Using this in the above term, we get

$$\begin{aligned}
&\left| \phi(\|x-x_i\|) - \mathbb{E}_u\big[\phi(\|u-x_i\|)\,|\,u\in B_\delta(x)\big] \right| \\
&\qquad \le \frac{L_\phi}{\nu_m\big(B_\delta(x)\big)} \int_{B_\delta(x)} \|x-u\|\, d\nu_m(u) = L_\phi\, \mathbb{E}_u\big[\|u-x\|\,|\,u\in B_\delta(x)\big] \\
&\qquad \le L_\phi\, \delta.
\end{aligned} \tag{43}$$

We now analyze the term in (42) for a given x_i for two different cases, i.e., for $x_i \notin B_\delta(x)$ and $x_i \in B_\delta(x)$. We first look at the case $x_i \notin B_\delta(x)$. For $x_j \in B_\delta(x)$, let

$$\zeta_j := \phi(\|x_j - x_i\|).$$

The observations ζ_j are i.i.d. (since x_j are i.i.d.) with mean $m_\zeta = \mathbb{E}_u\big[\phi(\|u-x_i\|)\,|\,u\in B_\delta(x)\big]$ and take values in the interval $\zeta_{\min} \le \zeta_j \le \zeta_{\max}$, where

$$\zeta_{\min} := \inf_{u\in B_\delta(x)} \phi(\|u-x_i\|), \qquad \zeta_{\max} := \sup_{u\in B_\delta(x)} \phi(\|u-x_i\|).$$

Since for any $u_1, u_2 \in B_\delta(x)$, $\|u_1-u_2\| \le 2\delta$, it follows from the Lipschitz continuity of ϕ that $\zeta_{\max} - \zeta_{\min} \le 2L_\phi\delta$. Using this together with the Hoeffding's inequality, we get

$$P\left(\left| \frac{1}{Q}\sum_{x_j\in A} \zeta_j - m_\zeta \right| \ge \epsilon \right) \le 2\exp\left(-\frac{2\,Q\,\epsilon^2}{(\zeta_{\max}-\zeta_{\min})^2} \right) \le 2\exp\left(-\frac{Q\,\epsilon^2}{2L_\phi^2\delta^2} \right). \tag{44}$$

We have

$$\left| \phi(\|x-x_i\|) - \frac{1}{Q}\sum_{x_j\in A}\phi(\|x_j-x_i\|) \right| \le \big|\phi(\|x-x_i\|) - m_\zeta\big| + \left| m_\zeta - \frac{1}{Q}\sum_{x_j\in A}\phi(\|x_j-x_i\|) \right|.$$

Using (43) and (44) in the above equation, it holds with probability at least

$$1 - 2\exp\left(-\frac{Q\,\epsilon^2}{2L_\phi^2\delta^2} \right)$$

that

$$\left| \phi(\|x-x_i\|) - \frac{1}{Q}\sum_{x_j\in A}\phi(\|x_j-x_i\|) \right| \le L_\phi\delta + \epsilon.$$

Next, we study the case $x_i \in B_\delta(x)$. For any fixed $x_i \in B_\delta(x)$, hence $x_i \in A$, we have

$$\begin{aligned}
&\left| \phi(\|x-x_i\|) - \frac{1}{Q}\sum_{x_j\in A}\phi(\|x_j-x_i\|) \right| \\
&= \left| \frac{1}{Q}\phi(\|x-x_i\|) + \frac{Q-1}{Q}\phi(\|x-x_i\|) - \frac{1}{Q}\phi(\|x_i-x_i\|) - \frac{1}{Q}\sum_{x_j\in A\setminus\{x_i\}}\phi(\|x_j-x_i\|) \right| \\
&\le \frac{1}{Q}\left|\phi(\|x-x_i\|) - \phi(\|x_i-x_i\|)\right| + \frac{Q-1}{Q}\left| \phi(\|x-x_i\|) - \frac{1}{Q-1}\sum_{x_j\in A\setminus\{x_i\}}\phi(\|x_j-x_i\|) \right|.
\end{aligned}$$

The first term above is bounded as

$$\frac{1}{Q}\left|\phi(\|x-x_i\|) - \phi(\|x_i-x_i\|)\right| \le \frac{L_\phi\delta}{Q}.$$

Next, similarly to the analysis of the case $x_i \ne B_\delta(x)$, we get that for $x_i \in B_\delta(x)$ with probability at least

$$1 - 2\exp\left(-\frac{(Q-1)\,\epsilon^2}{2L_\phi^2\delta^2} \right)$$

it holds that

$$\left| \phi(\|x-x_i\|) - \frac{1}{Q-1}\sum_{x_j\in A\setminus\{x_i\}}\phi(\|x_j-x_i\|) \right| \le L_\phi\delta + \epsilon,$$

hence

$$\left|\phi(\|x-x_i\|)-\frac{1}{Q}\sum_{x_j\in A}\phi(\|x_j-x_i\|)\right|\leq\frac{L_\phi\delta}{Q}+\frac{Q-1}{Q}(L_\phi\delta+\epsilon)\leq L_\phi\delta+\epsilon.$$

Combining the analyses of the cases $x_i\neq B_\delta(x)$ and $x_i\in B_\delta(x)$, we conclude that for any given $x_i\in\mathcal{X}$,

$$P\left(\left|\phi(\|x-x_i\|)-\frac{1}{Q}\sum_{x_j\in A}\phi(\|x_j-x_i\|)\right|\leq L_\phi\delta+\epsilon\right)\geq 1-2\exp\left(-\frac{(Q-1)\,\epsilon^2}{2L_\phi^2\delta^2}\right).$$

Therefore, applying the union bound on all N samples x_i in $\mathcal{X}$, we get that with probability at least

$$1-2N\exp\left(-\frac{(Q-1)\,\epsilon^2}{2L_\phi^2\delta^2}\right)$$

it holds that

$$\left|\phi(\|x-x_i\|)-\frac{1}{Q}\sum_{x_j\in A}\phi(\|x_j-x_i\|)\right|\leq L_\phi\delta+\epsilon \tag{45}$$

for all $x_i\in\mathcal{X}$.

We can now use this in (41) to bound the deviation of $f^k(x)$ from the empirical mean of f^k in the neighbourhood of x. Assuming that the condition (45) holds for all $x_i\in\mathcal{X}$, we obtain

$$\begin{aligned}\left|f^k(x)-\frac{1}{Q}\sum_{x_j\in A}f^k(x_j)\right|&=\left|\sum_{i=1}^N c_i^k\left(\phi(\|x-x_i\|)-\frac{1}{Q}\sum_{x_j\in A}\phi(\|x_j-x_i\|)\right)\right|\\&\leq(L_\phi\delta+\epsilon)\sum_{i=1}^N|c_i^k|\leq\mathcal{C}(L_\phi\delta+\epsilon),\end{aligned}$$

which gives

$$\|f(x)-\frac{1}{Q}\sum_{x_j\in A}f(x_j)\|=\left(\sum_{k=1}^d\left(f^k(x)-\frac{1}{Q}\sum_{x_j\in A}f^k(x_j)\right)^2\right)^{1/2}\leq\sqrt{d}\mathcal{C}(L_\phi\delta+\epsilon).$$

We thus get

$$P\left(\|f(x)-\frac{1}{Q}\sum_{x_j\in A}f(x_j)\|\leq\sqrt{d}\mathcal{C}(L_\phi\delta+\epsilon)\right)\geq 1-2N\exp\left(-\frac{(Q-1)\,\epsilon^2}{2L_\phi^2\delta^2}\right),$$

which completes the proof. ■

A.6 Proof of Theorem 8

Proof

Remember from the proof of Theorem 5 that

$$P(|A|\geq Q)\geq 1-\exp\left(\frac{-2\,(N_m\,\eta_{m,\delta}-Q)^2}{N_m}\right).$$

Lemma 7 states that, if $B_\delta(x)$ contains at least Q samples from class m, i.e., $|A|\geq Q$, then

$$\begin{aligned}P\left(\|f(x)-\frac{1}{|A|}\sum_{x_j\in A}f(x_j)\|\leq\sqrt{d}\mathcal{C}(L_\phi\delta+\epsilon)\right)&\geq 1-2N\exp\left(-\frac{(|A|-1)\,\epsilon^2}{2L_\phi^2\delta^2}\right)\\&\geq 1-2N\exp\left(-\frac{(Q-1)\,\epsilon^2}{2L_\phi^2\delta^2}\right).\end{aligned}$$

Hence, combining these two results (multiplying both probabilities), we get that with probability at least

$$\begin{aligned}&\left(1-\exp\left(\frac{-2\,(N_m\,\eta_{m,\delta}-Q)^2}{N_m}\right)\right)\left(1-2N\exp\left(-\frac{(Q-1)\,\epsilon^2}{2L_\phi^2\delta^2}\right)\right)\\&\geq 1-\exp\left(\frac{-2\,(N_m\,\eta_{m,\delta}-Q)^2}{N_m}\right)-2N\exp\left(-\frac{(Q-1)\,\epsilon^2}{2L_\phi^2\delta^2}\right)\end{aligned}$$

it holds that

$$\|f(x)-\frac{1}{|A|}\sum_{x_j\in A}f(x_j)\|\leq\sqrt{d}\mathcal{C}(L_\phi\delta+\epsilon). \tag{46}$$

A test sample x drawn from ν_m is classified correctly with the linear classifier if

$$\omega_{mk}^T\,f(x)+b_{mk}>0,\quad\forall k=1,\ldots,M-1,\,k\neq m.$$

If the condition in (46) is satisfied, for all $k\neq m$, we have

$$\begin{aligned}\omega_{mk}^T\,f(x)+b_{mk}&=\omega_{mk}^T\frac{1}{|A|}\sum_{x_j\in A}f(x_j)+b_{mk}+\omega_{mk}^T\left(f(x)-\frac{1}{|A|}\sum_{x_j\in A}f(x_j)\right)\\&\geq\omega_{mk}^T\frac{1}{|A|}\sum_{x_j\in A}f(x_j)+b_{mk}-\|f(x)-\frac{1}{|A|}\sum_{x_j\in A}f(x_j)\|\\&>\gamma_Q/2-\|f(x)-\frac{1}{|A|}\sum_{x_j\in A}f(x_j)\|\geq\gamma_Q/2-\sqrt{d}\mathcal{C}(L_\phi\delta+\epsilon)\geq 0.\end{aligned}$$

We thus conclude that with probability at least

$$1-\exp\left(\frac{-2\,(N_m\,\eta_{m,\delta}-Q)^2}{N_m}\right)-2N\exp\left(-\frac{(Q-1)\,\epsilon^2}{2L_\phi^2\delta^2}\right),$$

$\omega_{mk}^T\,f(x)+b_{mk}>0$ for all $k\neq m$, hence, the class label of x is estimated correctly. ■

A.7 Proof of Theorem 9

Proof First, recall from the proof of Theorem 8 that, with probability at least

$$1-\exp\left(\frac{-2\left(N_m\,\eta_{m,\delta}-Q\right)^2}{N_m}\right)-2N\exp\left(-\frac{(Q-1)\,\epsilon^2}{2L_\phi^2\delta^2}\right)$$

the δ-neighborhood $B_\delta(x)$ of a test sample x from class m contains at least Q samples from class m, and

$$\|f(x)-\frac{1}{|A|}\sum_{x_j\in A}f(x_j)\|\le\sqrt{d}\mathcal{C}(L_\phi\delta+\epsilon), \tag{47}$$

where A is the set of training samples in $B_\delta(x)$ from class m.

Then it is easy to show that (as in the proof of Theorem 6), with probability at least

$$1-\exp\left(\frac{-2\left(N_m\,\eta_{m,\delta}-Q\right)^2}{N_m}\right)-2N\exp\left(-\frac{(Q-1)\,\epsilon^2}{2L_\phi^2\delta^2}\right),$$

$B_\delta(x)$ will contain at least Q samples x_i from class m such that

$$\|f(x)-f(x_i)\|\le\sqrt{d}\mathcal{C}(L_\phi\delta+\epsilon)+D_{2\delta}. \tag{48}$$

Hence, for a training sample x_i' from another class (other than m), we have

$$\|f(x)-f(x_i')\|\ge\|f(x_i)-f(x_i')\|-\|f(x)-f(x_i)\|>\gamma-(\sqrt{d}\mathcal{C}(L_\phi\delta+\epsilon)+D_{2\delta}),$$

which follows from (48) and the hypothesis on the embedding that $\|f(x_i)-f(x_i')\|>\gamma$.

Due to the condition (16), we have $\gamma\ge 2\sqrt{d}\,\mathcal{C}\,(L_\phi\delta+\epsilon)+2D_{2\delta}$. Using this above equation, we obtain

$$\|f(x)-f(x_i')\|>\sqrt{d}\mathcal{C}(L_\phi\delta+\epsilon)+D_{2\delta}.$$

Therefore, the distance of $f(x)$ to the embedding of the samples from other classes is more than $\sqrt{d}\mathcal{C}(L_\phi\delta+\epsilon)+D_{2\delta}$, while there are samples from its own class within a distance of $\sqrt{d}\mathcal{C}(L_\phi\delta+\epsilon)+D_{2\delta}$ to $f(x)$. We thus conclude that the class label of x is estimated correctly with nearest-neighbor classification in the low-dimensional domain of embedding. ■

Appendix B. Proof of the Results in Section 3

B.1 Proof of Lemma 13

Proof The coordinate vector y is the eigenvector of the matrix $L_w-\mu L_b$ corresponding to its minimum eigenvalue. Hence,

$$y=\arg\min_{\substack{\xi\\ \|\xi\|=1}}\xi^T(L_w-\mu L_b)\xi.$$

Equivalently, defining the degree-normalized coordinates $z=D_w^{-1/2}y$, and thus replacing the above ξ by $D_w^{1/2}\xi$, we have

$$\begin{aligned} z&=\arg\min_{\substack{\xi\\ \xi^TD_w\xi=1}}N(\xi)\\ N(\xi)&=\xi^TD_w^{1/2}(L_w-\mu L_b)D_w^{1/2}\xi\\ &=\xi^T(D_w-W_w)\xi-\mu\,\xi^T(D_wD_b^{-1})^{1/2}\,(D_b-W_b)\,(D_b^{-1}D_w)^{1/2}\xi. \end{aligned} \tag{49}$$

Then, denoting $\beta_i=d_w(i)/d_b(i)$, the term $N(\xi)$ can be rearranged as

$$\begin{aligned} N(\xi)&=\sum_i\xi_i\Big(d_w(i)\,\xi_i-\sum_{j\sim_w i}\xi_j\,w_{ij}\Big)-\mu\sum_i\xi_i\Big(d_w(i)\,\xi_i-\sum_{j\sim_b i}\xi_j\,w_{ij}\,\sqrt{\beta_i\beta_j}\Big)\\ &=\sum_i\xi_i\sum_{j\sim_w i}(\xi_i-\xi_j)w_{ij}-\mu\sum_i\xi_i\sum_{j\sim_b i}(\beta_i\xi_i-\sqrt{\beta_i\beta_j}\,\xi_j)w_{ij}\\ &=\sum_i\sum_{j\sim_w i}(\xi_i^2-\xi_i\xi_j)w_{ij}-\mu\sum_i\sum_{j\sim_b i}(\beta_i\xi_i^2-\sqrt{\beta_i\beta_j}\xi_i\xi_j)w_{ij}, \end{aligned}$$

which gives [6]

$$N(\xi)=\sum_{i\sim_w j}(\xi_i-\xi_j)^2\,w_{ij}-\mu\sum_{i\sim_b j}\left(\sqrt{\beta_i}\xi_i-\sqrt{\beta_j}\xi_j\right)^2\,w_{ij} \tag{50}$$

by grouping the neighboring (i,j) pairs in the inner sums. Now, for any $\xi\in\mathbb{R}^{N\times 1}$ such that $\xi^TD_w\xi=1$, we define ξ^* as follows

$$\xi_i^*=\begin{cases}-|\xi_i| & \text{if } C_i=1\\ \ \ |\xi_i| & \text{if } C_i=2.\end{cases} \tag{51}$$

Clearly, ξ^* also satisfies $(\xi^*)^TD_w\xi^*=1$. From (50), it can be easily checked that $N(\xi^*)\le N(\xi)$ for any ξ, Then, a minimizer z of the problem (49) has to be of the separable form defined in (51); otherwise z^* would yield a smaller value for the function N, which would contradict the fact that z is a minimizer. Note that the equality $N(z^*)=N(z)$ holds only if $z=z^*$ or $z=-z^*$, thus when z is separable. Therefore, the embedding z satisfies the condition

$$z_i\le 0\ \text{ if } C_i=1,\qquad\qquad z_i\ge 0\ \text{ if } C_i=2,$$

or the equivalent condition

$$z_i\le 0\ \text{ if } C_i=2,\qquad\qquad z_i\ge 0\ \text{ if } C_i=1.$$

Finally, since $y_i=\sqrt{d_w(i)}\,z_i$, the same property also holds for the embedding y. ■

6. In our notation, the terms $i\sim_w j$ and $i\sim_b j$ in the summation indices as in (50) refer to edges rather than neighboring (i,j)-pairs; i.e., each pair is counted only once in the summation.

B.2 Proof of Theorem 14

Proof From (49) and (50), we have

$$z = \arg \min_{\substack{\xi \\ \xi^T D_w \xi = 1}} \sum_{i \sim_w j} (\xi_i - \xi_j)^2 \, w_{ij} - \mu \sum_{i \sim_b j} \left(\sqrt{\beta_i} \xi_i - \sqrt{\beta_j} \xi_j \right)^2 \, w_{ij}. \tag{52}$$

Thus, at the optimal solution z the objective function takes the value

$$N(z) = \sum_{i \sim_w j} (z_i - z_j)^2 \, w_{ij} - \mu \sum_{i \sim_b j} \left(\sqrt{\beta_i} z_i - \sqrt{\beta_j} z_j \right)^2 \, w_{ij}. \tag{53}$$

In the following, we derive a lower bound for the first sum and an upper bound for the second sum in (53). We begin with the first sum. Let $i_{1,min}$, $i_{1,max}$, $i_{2,min}$ and $i_{2,max}$ denote the indices of the data samples in class 1 and class 2 that are respectively mapped to the extremal coordinates $z_{1,min}$, $z_{1,max}$, $z_{2,min}$, $z_{2,max}$, where

$$z_{k,min} = \min_{i:\, C_i = k} z_i \,, \qquad\qquad z_{k,max} = \max_{i:\, C_i = k} z_i \,.$$

Let $P_1 = \{(x_{k_{i-1}}, x_{k_i})\}_{i=1}^{L_1}$ be a shortest path of length L_1 joining $x_{i_{1,min}}$ and $x_{i_{1,max}}$ and $P_2 = \{(x_{n_{i-1}}, x_{n_i})\}_{i=1}^{L_2}$ be a shortest path of length L_2 joining $x_{i_{2,min}}$ and $x_{i_{2,max}}$. We have

$$\begin{aligned} \sum_{i \sim_w j} (z_i - z_j)^2 \, w_{ij} &\geq \sum_{i=1}^{L_1} (z_{k_i} - z_{k_{i-1}})^2 \, w_{k_{i-1} k_i} + \sum_{i=1}^{L_2} (z_{n_i} - z_{n_{i-1}})^2 \, w_{n_{i-1} n_i} \\ &\geq w_{min,1} \sum_{i=1}^{L_1} (z_{k_i} - z_{k_{i-1}})^2 + w_{min,2} \sum_{i=1}^{L_2} (z_{n_i} - z_{n_{i-1}})^2 \,, \end{aligned} \tag{54}$$

where the first inequality simply follows from the fact that the set of edges making up $P_1 \cup P_2$ are contained in the set of all edges in G_w. For a sequence $\{a_i\}_{i=0}^{L}$, the following inequality holds

$$\begin{aligned} (a_L - a_0)^2 &= \sum_{i=1}^{L} (a_i - a_{i-1})^2 + \sum_{\substack{i,j=1 \\ i \neq j}}^{L} (a_i - a_{i-1})(a_j - a_{j-1}) \\ &\leq \sum_{i=1}^{L} (a_i - a_{i-1})^2 + \frac{1}{2} \sum_{\substack{i,j=1 \\ i \neq j}}^{L} \left((a_i - a_{i-1})^2 + (a_j - a_{j-1})^2 \right) = L \sum_{i=1}^{L} (a_i - a_{i-1})^2. \end{aligned}$$

Hence,

$$\sum_{i=1}^{L} (a_i - a_{i-1})^2 \geq \frac{1}{L} (a_L - a_0)^2.$$

Using this inequality in (54), we get

$$\sum_{i \sim_w j} (z_i - z_j)^2 \, w_{ij} \geq \frac{w_{min,1}}{L_1} (z_{1,max} - z_{1,min})^2 + \frac{w_{min,2}}{L_2} (z_{2,max} - z_{2,min})^2.$$

Since the path lengths L_1 and L_2 are upper bounded by the diameters D_1 and D_2, we finally obtain the lower bound

$$\sum_{i \sim_w j} (z_i - z_j)^2 \, w_{ij} \geq \frac{w_{min,1}}{D_1} (z_{1,max} - z_{1,min})^2 + \frac{w_{min,2}}{D_2} (z_{2,max} - z_{2,min})^2. \tag{55}$$

Next, we find an upper bound for the second sum in (53). Using Lemma 13, we obtain the following inequality

$$\begin{aligned} \sum_{i \sim_b j} \left(\sqrt{\beta_i} z_i - \sqrt{\beta_j} z_j \right)^2 \, w_{ij} &\leq \sum_{i \sim_b j} (z_{2,max} - z_{1,min})^2 \, \beta_{max} \, w_{ij} \\ &= \frac{1}{2} (z_{2,max} - z_{1,min})^2 \, \beta_{max} V^b_{max}. \end{aligned} \tag{56}$$

Now, since the solution z in (52) minimizes the objective function $N(\xi)$, we have

$$N(z) = \lambda_{\min}(L_w - \mu L_b),$$

where $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ respectively denote the minimum and the maximum eigenvalues of a matrix. For two Hermitian matrices A and B, the inequality $\lambda_{\min}(A+B) \leq \lambda_{\min}(A) + \lambda_{\max}(B)$ holds. As L_w and L_b are graph Laplacian matrices, we have $\lambda_{\min}(L_w) = \lambda_{\min}(L_b) = 0$ and thus

$$N(z) = \lambda_{\min}(L_w - \mu L_b) \leq \lambda_{\min}(L_w) + \lambda_{\max}(-\mu L_b) = \lambda_{\min}(L_w) - \mu \lambda_{\min}(L_b) = 0.$$

Using in (53) the above inequality and the lower and upper bounds in (55) and (56), we obtain

$$\begin{aligned} 0 \geq N(z) &= \sum_{i \sim_w j} (z_i - z_j)^2 \, w_{ij} - \mu \sum_{i \sim_b j} \left(\sqrt{\beta_i} z_i - \sqrt{\beta_j} z_j \right)^2 \, w_{ij} \\ &\geq \frac{w_{min,1}}{D_1} (z_{1,max} - z_{1,min})^2 + \frac{w_{min,2}}{D_2} (z_{2,max} - z_{2,min})^2 \\ &\quad - \frac{1}{2} \mu (z_{2,max} - z_{1,min})^2 \, \beta_{max} V^b_{max}. \end{aligned}$$

Hence

$$\frac{w_{min,1}}{D_1} (z_{1,max} - z_{1,min})^2 + \frac{w_{min,2}}{D_2} (z_{2,max} - z_{2,min})^2 \leq \frac{1}{2} \mu (z_{2,max} - z_{1,min})^2 \, \beta_{max} V^b_{max}. \tag{57}$$

The RHS of the above inequality is related to the overall support $z_{2,max} - z_{1,min}$ of the data, whereas the terms on the LHS are related to the individual supports $z_{1,max} - z_{1,min}$ and $z_{2,max} - z_{2,min}$ of the two classes in the learnt embedding. Meanwhile, the separation $z_{2,min} - z_{1,max}$ between the two classes is given by the gap between the overall support and the sums of the individual supports. In order to use the above inequality in view of this observation, we first derive a lower bound on the RHS term. Since $z^T D_w z = 1$, we have

$$\begin{aligned} 1 = \sum_i z_i^2 \, d_w(i) &= \sum_{i:\, C_i = 1} z_i^2 \, d_w(i) + \sum_{i:\, C_i = 2} z_i^2 \, d_w(i) \\ &\leq z^2_{1,min} \sum_{i:\, C_i = 1} d_w(i) + z^2_{2,max} \sum_{i:\, C_i = 2} d_w(i) = z^2_{1,min} V_1 + z^2_{2,max} V_2. \end{aligned}$$

This gives

$$z_{1,min}^2 + z_{2,max}^2 \geq \frac{1}{V_{max}}.$$

Hence, we obtain the following lower bound on the overall support

$$(z_{2,max} - z_{1,min})^2 \geq z_{2,max}^2 + z_{1,min}^2 \geq \frac{1}{V_{max}}. \tag{58}$$

Denoting the supports of class 1 and class 2 and the overall support as

$$S_1 = z_{1,max} - z_{1,min}, \qquad S_2 = z_{2,max} - z_{2,min}, \qquad S = z_{2,max} - z_{1,min},$$

we have from (57)

$$\overline{w}_{min}(S_1^2 + S_2^2) \leq \frac{1}{2}\,\mu\, S^2\, \beta_{max} V_{max}^b,$$

which yields the following upper bound on the total support of the two classes

$$S_1 + S_2 \leq \sqrt{2(S_1^2 + S_2^2)} \leq S\sqrt{\frac{\mu \beta_{max} V_{max}^b}{\overline{w}_{min}}}.$$

We can thus lower bound the separation $z_{2,min} - z_{1,max}$ as

$$z_{2,min} - z_{1,max} = S - (S_1 + S_2) \geq S\left(1 - \sqrt{\frac{\mu \beta_{max} V_{max}^b}{\overline{w}_{min}}}\right),$$

provided that $\mu < \overline{w}_{min}/(\beta_{max} V_{max}^b)$. From the lower bound on the overall support in (58), we lower bound the separation as follows

$$z_{2,min} - z_{1,max} \geq \frac{1}{\sqrt{V_{max}}}\left(1 - \sqrt{\frac{\mu \beta_{max} V_{max}^b}{\overline{w}_{min}}}\right).$$

Finally, since the separation of any embedding with dimension $d \geq 1$ is at least as much as the separation $z_{2,min} - z_{1,max}$ of the embedding of dimension $d = 1$, the above lower bound holds for any $d \geq 1$ as well. ■

B.3 Proof of Corollary 15

Proof The one-dimensional embedding z is given as the solution of the constrained optimization problem

$$z = \arg\min N(\xi) \text{ s.t. } D(\xi) = 1,$$

where

$$N(\xi) = \xi^T D_w^{1/2}(L_w - \mu L_b) D_w^{1/2}\xi, \qquad D(\xi) = \xi^T D_w \xi.$$

Defining the Lagrangian function

$$\Lambda(\xi, \lambda) = N(\xi) + \lambda(D(\xi) - 1)$$

at the optimal solution z, we have

$$\nabla_\xi \Lambda = \nabla_\lambda \Lambda = 0,$$

where ∇_ξ and ∇_λ respectively denote the derivatives of Λ with respect to ξ and λ. Thus, at $\xi = z$,

$$\frac{\partial \Lambda}{\partial \xi_i} = \frac{\partial N(\xi)}{\partial \xi_i} + \lambda \frac{\partial D(\xi)}{\partial \xi_i} = 0$$

for all $i = 1, \ldots, N$. From (50), the derivatives of $N(\xi)$ and $D(\xi)$ at z are given by

$$\left.\frac{\partial N(\xi)}{\partial \xi_i}\right|_{\xi=z} = \sum_{j \sim_w i} 2(z_i - z_j) w_{ij} - \mu \sum_{j \sim_b i} 2\left(\sqrt{\beta_i} z_i - \sqrt{\beta_j} z_j\right)\sqrt{\beta_i}\, w_{ij}$$
$$\left.\frac{\partial D(\xi)}{\partial \xi_i}\right|_{\xi=z} = 2\, d_w(i)\, z_i,$$

which yields

$$\sum_{j \sim_w i} (z_i - z_j) w_{ij} - \mu \sum_{j \sim_b i} \left(\sqrt{\beta_i} z_i - \sqrt{\beta_j} z_j\right)\sqrt{\beta_i}\, w_{ij} + \lambda\, d_w(i)\, z_i = 0 \tag{59}$$

for all i. At $i = i_{1,max}$, as z attains its maximal value $z_{1,max}$ for class 1, we have

$$\begin{aligned}\lambda\, d_w(i_{1,max})\, z_{1,max} &= \sum_{j \sim_w i_{1,max}} (z_j - z_{1,max}) w_{i_{1,max} j} \\ &\quad + \mu \sum_{j \sim_b i_{1,max}} \left(\sqrt{\beta_{i_{1,max}}} z_{1,max} - \sqrt{\beta_j} z_j\right)\sqrt{\beta_{i_{1,max}}}\, w_{i_{1,max} j} \\ &\leq -\mu\, \beta_{min}\, (z_{2,min} - z_{1,max})\, d_b(i_{1,max}).\end{aligned}$$

Hence

$$|z_{1,max}| = -z_{1,max} \geq \frac{\mu\, \beta_{min}\, (z_{2,min} - z_{1,max}) d_b(i_{1,max})}{\lambda\, d_w(i_{1,max})} \geq \frac{\mu\, \beta_{min}\, (z_{2,min} - z_{1,max})}{\lambda\, \beta_{max}}. \tag{60}$$

We proceed by deriving an upper bound for λ. The gradients of $N(\xi)$ and $D(\xi)$ are given by

$$\nabla_\xi N = 2D_w^{1/2}(L_w - \mu L_b) D_w^{1/2}\xi, \qquad \nabla_\xi D = 2D_w \xi.$$

From the condition $\nabla_\xi \Lambda = 0$ at $\xi = z$, we have

$$\begin{aligned} D_w^{1/2}(L_w - \mu L_b) D_w^{1/2} z + \lambda D_w z &= 0 \\ (L_w - \mu L_b) y + \lambda y &= 0.\end{aligned}$$

Since $y = D_w^{1/2} z$ is the unit-norm eigenvector of $L_w - \mu L_b$ corresponding to its smallest eigenvalue, the Lagrangian multiplier λ is given by

$$\lambda = -\lambda_{\min}(L_w - \mu L_b).$$

We can lower bound the minimum eigenvalue as

$$\lambda_{\min}(L_w - \mu L_b) \geq \lambda_{\min}(L_w) + \lambda_{\min}(-\mu L_b) = 0 - \mu\lambda_{\max}(L_b) \geq -2\mu$$

since the eigenvalues of a graph Laplacian are upper bounded by 2. This gives $\lambda \leq 2\mu$. Using this upper bound on λ in (60), we obtain

$$|z_{1,max}| \geq \frac{1}{2}\frac{\beta_{min}}{\beta_{max}}(z_{2,min} - z_{1,max}).$$

Repeating the same steps for $i = i_{2,min}$ following (59), one can similarly show that

$$z_{2,min} \geq \frac{1}{2}\frac{\beta_{min}}{\beta_{max}}(z_{2,min} - z_{1,max}).$$

■

References

B. J. C. Baxter. *The interpolation theory of radial basis functions*. PhD thesis, Cambridge University, Trinity College, 1992.

M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, June 2003.

P. J. Bickel and B. Li. Local polynomial regression on unknown manifolds. *Lecture Notes-Monograph Series*, 54:177–186, 2007.

A. Caponnetto and E. De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

T. J. Chin and D. Suter. Out-of-sample extrapolation of learned manifolds. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(9):1547–1556, 2008.

F. R. K. Chung. *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society, December 1996.

F. Cucker and S. Smale. On the mathematical foundations of learning. *Bulletin of the American Mathematical Society*, 39:1–49, 2002.

Y. Cui and L. Fan. A novel supervised dimensionality reduction algorithm: Graph-based fisher analysis. *Pattern Recognition*, 45(4):1471–1481, 2012.

D. L. Donoho and C. Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences of the United States of America*, 100(10):5591–5596, May 2003.

A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

X. He and P. Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2004. MIT Press.

R. Herbrich. Exact tail bounds for binomial distributed variables. *Online: Available at http://research.microsoft.com/apps/pubs/default.aspx?id=66854*, 1999.

A. Hernández-Aguirre, C. Koutsougeras, and B. P. Buckles. Sample complexity for function learning tasks through linear neural networks. *International Journal on Artificial Intelligence Tools*, 11(4):499–511, 2002.

Q. Hua, L. Bai, X. Wang, and Y. Liu. Local similarity and diversity preserving discriminant projection for face and handwriting digits recognition. *Neurocomputing*, 86:150–157, 2012.

A. N. Kolmogorov and V. M. Tihomirov. ε-entropy and ε-capacity of sets in functional spaces. *Amer. Math. Soc. Transl.*, 2(17):277–364, 1961.

S. Kpotufe. k-NN regression adapts to local intrinsic dimension. In *Proc. Advances in Neural Information Processing Systems 24*, pages 729–737, 2011.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. Advances in Neural Information Processing Systems 25*, pages 1106–1114, 2012.

S. R. Kulkarni and S. E. Posner. Rates of convergence of nearest neighbor estimation under arbitrary sampling. *IEEE Transactions on Information Theory*, 41(4):1028–1039, 1995.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pages 2278–2324, 1998.

B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003)*, pages 409–415, 2003.

B. Li, J. Liu, Z. Zhao, and W. Zhang. Locally linear representation fisher criterion. In *The 2013 International Joint Conference on Neural Networks*, pages 1–7, 2013.

S. Lin, X. Liu, Y. Rong, and Z. Xu. Almost optimal estimates for approximation and learning by radial basis function networks. *Machine Learning*, 95(2):147–164, 2014.

F. J. Narcowich, N. Sivakumar, and J. D. Ward. On condition numbers associated with radial-function interpolation. *Journal of Mathematical Analysis and Applications*, 186(2): 457 – 485, 1994.

S. A. Nene, S. K. Nayar, and H. Murase. Columbia Object Image Library (COIL-20). Technical report, Feb 1996.

P. Niyogi and F. Girosi. On the relationship between generalization error, hypothesis complexity, and sample complexity for radial basis functions. *Neural Computation*, 8:819–842, 1996.

B. Peherstorfer, D. Pflüger, and H. J. Bungartz. A sparse-grid-based out-of-sample extension for dimensionality reduction and clustering with laplacian eigenmaps. In *AI 2011: Proc. Advances in Artificial Intelligence - 24th Australasian Joint Conference*, pages 112–121, 2011.

C. Piret. *Analytical and Numerical Advances in Radial Basis Functions*. PhD thesis, University of Colorado, 2007.

H. Qiao, P. Zhang, D. Wang, and B. Zhang. An explicit nonlinear mapping for manifold learning. *IEEE T. Cybernetics*, 43(1):51–63, 2013.

B. Raducanu and F. Dornaika. A supervised non-linear dimensionality reduction approach for manifold learning. *Pattern Recognition*, 45(6):2432–2444, 2012.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

I. Steinwart, D. Hush, and C. Scovel. Optimal rates for regularized least squares regression. In *The 22nd Conference on Learning Theory*, 2009.

M. Sugiyama. Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. *Journal of Machine Learning Research*, 8:1027–1061, 2007.

J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 2000.

M. Vidyasagar. *A Theory of Learning and Generalization*. Springer-Verlag, Secaucus, NJ, USA, 2nd edition, 1997.

E. Vural and C. Guillemot. Out-of-sample generalizations for supervised manifold learning for classification. *IEEE Transactions on Image Processing*, 25(3):1410–1424, March 2016a.

E. Vural and C. Guillemot. A study of the classification of low-dimensional data with supervised manifold learning. *Technical report*. Available at https://arxiv.org/abs/1507.05880, 2016b.

R. Wang and X. Chen. Manifold discriminant analysis. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 429–436, 2009.

W. Yang, C. Sun, and L. Zhang. A multi-manifold discriminant analysis method for image feature extraction. *Pattern Recognition*, 44(8):1649–1657, 2011.

Z. Zhang and H. Zha. Principal manifolds and nonlinear dimension reduction via local tangent space alignment. *SIAM Journal of Scientific Computing*, 26:313–338, 2005.

Z. Zhang, M. Zhao, and T. Chow. Marginal semi-supervised sub-manifold projections with informative constraints for dimensionality reduction and recognition. *Neural Networks*, 36:97–111, 2012.

Journal of Machine Learning Research 18 (2018) 1-49 Submitted 9/15; Revised 4/17; Published 4/18

Probabilistic Preference Learning with the Mallows Rank Model

Valeria Vitelli VALERIA.VITELLI@MEDISIN.UIO.NO
Oslo Centre for Biostatistics and Epidemiology,
Department of Biostatistics, University of Oslo,
P.O.Box 1122 Blindern, NO-0317, Oslo, Norway

Øystein Sørensen OYSTEIN.SORENSEN.1985@GMAIL.COM
Oslo Centre for Biostatistics and Epidemiology,
Department of Biostatistics, University of Oslo,
P.O.Box 1122 Blindern, NO-0317, Oslo, Norway

Marta Crispino MARTA.CRISPINO@PHD.UNIBOCCONI.IT
Department of Decision Sciences, Bocconi University,
via Röntgen 1, 20100, Milan, Italy

Arnoldo Frigessi ARNOLDO.FRIGESSI@MEDISIN.UIO.NO
Oslo Centre for Biostatistics and Epidemiology,
University of Oslo and Oslo University Hospital,
P.O.Box 1122 Blindern, NO-0317, Oslo, Norway

Elja Arjas ELJA.ARJAS@HELSINKI.FI
Oslo Centre for Biostatistics and Epidemiology,
Department of Biostatistics, University of Oslo,
P.O.Box 1122 Blindern, NO-0317, Oslo, Norway

Editor: Francois Caron

Abstract

Ranking and comparing items is crucial for collecting information about preferences in many areas, from marketing to politics. The Mallows rank model is among the most successful approaches to analyze rank data, but its computational complexity has limited its use to a particular form based on Kendall distance. We develop new computationally tractable methods for Bayesian inference in Mallows models that work with any right-invariant distance. Our method performs inference on the consensus ranking of the items, also when based on partial rankings, such as top-k items or pairwise comparisons. We prove that items that none of the assessors has ranked do not influence the maximum a posteriori consensus ranking, and can therefore be ignored. When assessors are many or heterogeneous, we propose a mixture model for clustering them in homogeneous subgroups, with cluster-specific consensus rankings. We develop approximate stochastic algorithms that allow a fully probabilistic analysis, leading to coherent quantifications of uncertainties. We make probabilistic predictions on the class membership of assessors based on their ranking of just some items, and predict missing individual preferences, as needed in recommendation systems. We test our approach using several experimental and benchmark data sets.

Keywords: Incomplete rankings, Pairwise comparisons, Preference learning with uncertainty, Recommendation systems, Markov Chain Monte Carlo

1. Introduction

Various types of data have ranks as their natural scale. Companies recruit panels to rank novel products, market studies are often based on interviews where competing services or items are compared or ranked. In recent years, analyzing preference data collected over the internet (for example, movies, books, restaurants, political candidates) has been receiving much attention, and often these data are in the form of partial rankings.

Some typical tasks for rank or preference data are: (i) aggregate, merge, summarize multiple individual rankings to estimate the consensus ranking; (ii) predict the ranks of unranked items at individual level; (iii) partition the assessors into classes, each sharing a consensus ranking of the items, and classify new assessors to a class. In this paper we phrase all these tasks (and their combinations) in a unified Bayesian inferential setting, which allows us to also quantify posterior uncertainty of the estimates. Uncertainty evaluations of the estimated preferences and class memberships are a fundamental aspect of information in marketing and decision making. When predictions are too unreliable, actions based on these might better be postponed until more data are available and safer predictions can be made, so as not to unnecessarily annoy users or clients.

There exist many probabilistic models for ranking data which differ both in the data generation mechanism and in the parametric space. Two of the most commonly used are the Plackett-Luce, PL, (Luce, 1959; Plackett, 1975) and the Mallows models (Mallows, 1957). The PL model is a stage-wise probabilistic model on permutations, while the Mallows model is based on a distance function between rankings. Inferring the parameters of the PL distribution is typically done by maximum likelihood estimation, using a minorize/maximize algorithm (Hunter, 2004). A Bayesian approach was first proposed by Guiver and Snelson (2009). Caron and Teh (2012) perform Bayesian inference in a Plackett-Luce model with time-dependent preference probabilities, and further develop the framework in Caron et al. (2014), where a Dirichlet process mixture is used to cluster assessors based on their preferences. The parameters in the PL model are continuous, which gives to this model much flexibility. Volkovs and Zemel (2014) develop a generalization of the PL model, called multinomial preference model, which deals with pairwise preferences, even inconsistent ones, and extends to supervised problems. One difficulty of this method is the use of gradient optimization in a non-convex problem (which can lead to local optima), and the somewhat arbitrary way of imputing missing ranks. Compared to the PL model, the Mallows model has the advantage of being flexible in the choice of the distance function between permutations. It is also versatile in its ability to adapt to different kinds of data (pairwise comparisons, partial rankings). However, for some distances exact inference is very demanding, because the partition function normalizing the model is very expensive to compute. Therefore most work on the Mallows has been limited to a few particular distances, like the Kendall distance, for which the partition function can be computed analytically. Maximum Likelihood inference about the consensus ranking in the Mallows model is generally very difficult, and in many cases NP-hard, which lead to the development of heuristic algorithms. The interesting proposal of Lu and Boutilier (2014) makes use of the Generalized Repeated Insertion Model (GRIM), based on the EM algorithm, and allows also for data in the form of pairwise preferences. Their model focuses on the Kendall distance only, and it provides no uncertainty quantification. Another interesting EM-based approach is Khan et al. (2014), which

is driven by expectation propagation approximate inference, and scales to very large data sets without requiring strong factorization assumptions. Among probabilistic approaches, Meilă and Chen (2010) use Dirichlet process mixtures to perform Bayesian clustering of assessors in the Mallows model, but they again focus on the Kendall distance only. Jacques and Biernacki (2014) also propose clustering based on partial rankings, but in the context of the Insertion Sorting Rank (ISR) model. Hence, the approach is probabilistic but it is far from the general form of the Mallows, even though it has connections with the Mallows with Kendall distance. See Section 5 for a more detailed presentation of related work. For the general background on statistical methods for rank data, we refer to the excellent monograph by Marden (1995), and to the book by Alvo and Yu (2014).

The contributions of this paper are summarized as follows. We develop a Bayesian framework for inference in Mallows models that works with any right-invariant metric. In particular, the method is able to handle some of the right-invariant distances poorly considered in the existing literature, because of their well-known intractability. In this way the main advantage of the Mallows models, namely its flexibility in the choice of the distance, is fully exploited. We propose a Metropolis-Hastings iterative algorithm, which converges to the Bayesian posterior distribution, if the exact partition function is available. In case the exact partition function is not available, we propose to approximate it using an off-line importance sampling scheme, and we document the quality and efficiency of this approximation. Using data augmentation techniques, our method handles incomplete rankings, like the important cases of top-k rankings, pairwise comparisons, and ranks missing at random. For the common situation when the pool of assessors is heterogeneous, and cannot be assumed to share a common consensus, we develop a Bayesian clustering scheme which embeds the Mallows model. Our approach unifies clustering, classification and preference prediction in a single inferential procedure, thus leading to coherent posterior credibility levels of learned rankings and predictions. The probabilistic Bayesian setting allows us to naturally compute complex probabilities of interest, like the probability that an item has consensus rank higher than a given level, or the probability that the consensus rank of an item is higher than that of another item of interest. For incomplete rankings this can be done also at the individual assessor level, allowing for individual recommendations.

In Section 2, we introduce the Bayesian Mallows model for rank data. In Section 2.1, we discuss how the choice of the distance function influences the calculation of the partition function, and Section 2.2 is devoted to the choice of the prior distributions. In Sections 2.3 and 2.4, we show how efficient Bayesian computation can be performed for this model, using a novel leap-and-shift proposal distribution. The tuning of the hyperparameters is discussed in the Supplementary Material, Section 1. In Section 3 we develop and test an importance sampling scheme for computing the partition function, based on a pseudo-likelihood approximation of the Mallows model. We carefully test and study this importance sampling estimation of the partition function (Section 3.1), and the effect of this estimation on inference, both theoretically (Section 3.2) and by simulations (Section 3.3). Section 4 is dedicated to partial rankings and clustering of assessors. In Section 4.1 we extend the Bayesian Mallows approach to partial rankings, and we prove some results on the effects of unranked items on the consensus ranking (Section 4.1.1). Section 4.2 considers data in the form of ordered subsets or pairwise comparisons of items. In Section 4.3 we describe a mixture model to deal with the possible heterogeneity of assessors, finding cluster-

specific consensus rankings. Section 4.4 is dedicated to prediction in a realistic setup, which requires both the cluster assignment and personalized preference learning. We show that our approach works well in a simulation context. In Section 5 we review related methods which have been proposed in the literature, and compare by simulation some algorithms with our procedure (Section 5.1). In Section 6, we then move to the illustration of the performance of our method on real data: the selected case studies illustrate the different incomplete data situations considered. This includes the Sushi (Section 6.3) and Movielens (Section 6.4) benchmark data. Section 7 presents some conclusions and extensions.

2. A Bayesian Mallows Model for Complete Rankings

Assume we have a set of n items, labelled $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$. We first assume that each of N assessors ranks all items individually with respect to a considered feature. The ordering provided by assessor j is represented by $\mathbf{X}_j$, whose n components are items in $\mathcal{A}$. The item with rank 1 appears as the first element, up to the item with rank n appearing as the n-th element. The observations $\mathbf{X}_1, \ldots, \mathbf{X}_N$ are hence N permutations of the labels in $\mathcal{A}$. Let $R_{ij} = \mathbf{X}_j^{-1}(A_i),\ i = 1, \ldots, n,\ j = 1, \ldots, N$, denote the rank given to item A_i by assessor j, and let $\mathbf{R}_j = (R_{1j}, R_{2j}, \ldots, R_{nj}),\ j = 1, \ldots, N$, denote the ranking (that is the full set of ranks given to the items), of assessor j. Letting $\mathcal{P}_n$ be the set of all permutations of $\{1, \ldots, n\}$, we have $\mathbf{R}_j \in \mathcal{P}_n,\ j = 1, \ldots, N$. Finally, let $d(\cdot, \cdot) : \mathcal{P}_n \times \mathcal{P}_n \to [0, \infty)$ be a distance function between two rankings.

The Mallows model (Mallows, 1957) is a class of non-uniform joint distributions for a ranking $\mathbf{r}$ on $\mathcal{P}_n$, of the form $P(\mathbf{r}|\alpha, \boldsymbol{\rho}) = Z_n(\alpha, \boldsymbol{\rho})^{-1} \exp\{-(\alpha/n) d(\mathbf{r}, \boldsymbol{\rho})\} 1_{\mathcal{P}_n}(\mathbf{r})$, where $\boldsymbol{\rho} \in \mathcal{P}_n$ is the latent consensus ranking, α is a scale parameter, assumed positive for identification purposes, $Z_n(\alpha, \boldsymbol{\rho}) = \sum_{\mathbf{r} \in \mathcal{P}_n} e^{-\frac{\alpha}{n} d(\mathbf{r}, \boldsymbol{\rho})}$ is the partition function, and $1_S(\cdot)$ is the indicator function of the set S. We assume that the N observed rankings $\mathbf{R}_1, \ldots, \mathbf{R}_N$ are conditionally independent given α and $\boldsymbol{\rho}$, and that each of them is distributed according to the Mallows model with these parameters. The likelihood takes then the form

$$P(\mathbf{R}_1, \ldots, \mathbf{R}_N | \alpha, \boldsymbol{\rho}) = \frac{1}{Z_n(\alpha, \boldsymbol{\rho})^N} \exp\left\{ -\frac{\alpha}{n} \sum_{j=1}^{N} d(\mathbf{R}_j, \boldsymbol{\rho}) \right\} \prod_{j=1}^{N} \{1_{\mathcal{P}_n}(\mathbf{R}_j)\}. \tag{1}$$

For a given α, the maximum likelihood estimate of $\boldsymbol{\rho}$ is obtained by computing

$$\underset{\boldsymbol{\rho} \in \mathcal{P}_n}{\operatorname{argmax}} \frac{\exp\left\{ -\frac{\alpha}{n} \sum_{j=1}^{N} d(\mathbf{R}_j, \boldsymbol{\rho}) \right\}}{Z_n(\alpha, \boldsymbol{\rho})^N}. \tag{2}$$

For large n this optimization problem is not feasible, because the space of permutations has $n!$ elements. This has impact both on the computation of $Z_n(\alpha, \boldsymbol{\rho})$, and on the minimization of the sum in the exponential of (2), which is typically NP-hard (Bartholdi et al., 1989).

2.1 Distance Measures and Partition Function

Right-invariant distances (Diaconis, 1988) play an important role in the Mallows models. A right-invariant distance is unaffected by a relabelling of the items, which is a reasonable assumption in many situations. For any right-invariant distance it holds $d(\boldsymbol{\rho_1}, \boldsymbol{\rho_2}) =$

$d(\boldsymbol{\rho_1}\boldsymbol{\rho_2}^{-1}, \mathbf{1}_n)$, where $\mathbf{1}_n = \{1, 2, ..., n\}$, and therefore the partition function $Z_n(\alpha, \boldsymbol{\rho})$ of (1) is independent on the latent consensus ranking $\boldsymbol{\rho}$. We write $Z_n(\alpha, \boldsymbol{\rho}) = Z_n(\alpha) = \sum_{\mathbf{r}\in\mathcal{P}_n} \exp\{-\frac{\alpha}{n} d(\mathbf{r}, \mathbf{1}_n)\}$. All distances considered in this paper are right-invariant. Importantly, since the partition function $Z_n(\alpha)$ does not depend on the latent consensus $\boldsymbol{\rho}$, it can be computed off-line over a grid for α, given n (details in Section 3). For some choices of right-invariant distances, the partition function can be analytically computed. For this reason, most of the literature considers the Mallows model with Kendall distance (Lu and Boutilier, 2014; Meilă and Chen, 2010), for which a closed form of $Z_n(\alpha)$ is given in Fligner and Verducci (1986), or with the Hamming (Irurozki et al., 2014) and Cayley (Irurozki et al., 2016b) distances. There are important and natural right-invariant distances for which the computation of the partition function is not feasible, in particular the footrule (l_1) and the Spearman's (l_2) distances. For precise definitions of all distances involved in the Mallows model we refer to Marden (1995). Following Irurozki et al. (2016a), $Z_n(\alpha)$ can be written in a more convenient way. Since $d(\mathbf{r}, \mathbf{1}_n)$ takes only the finite number of discrete values $\mathcal{D} = \{d_1, ..., d_a\}$, where a depends on n and on the distance $d(\cdot, \cdot)$, we define $L_i = \{\mathbf{r} \in \mathcal{P}_n : d(\mathbf{r}, \mathbf{1}_n) = d_i\} \subset \mathcal{P}_n$, $i = 1, ..., a$, to be the set of permutations at the same given distance from $\mathbf{1}_n$, and $|L_i|$ corresponds to its cardinality. Then

$$Z_n(\alpha) = \sum_{d_i\in\mathcal{D}} |L_i| \exp\{-(\alpha/n)d_i\}. \tag{3}$$

In order to compute $Z_n(\alpha)$ one thus needs $|L_i|$, for all values $d_i \in \mathcal{D}$. In the case of the footrule distance, the set $\mathcal{D}$ includes all even numbers, from 0 to $\lfloor n^2/2 \rfloor$, and $|L_i|$ corresponds to the sequence A062869 available for $n \leq 50$ on the On-Line Encyclopedia of Integer Sequences (OEIS) (Sloane, 2017). In the case of Spearman's distance, the set $\mathcal{D}$ includes all even numbers, from 0 to $2\binom{n}{3}$, and $|L_i|$ corresponds to the sequence A175929 available for $n \leq 14$ in the OEIS. When the partition function is needed for larger values of n, we suggest an importance sampling scheme which efficiently approximates $Z_n(\alpha)$ to an arbitrary precision (see Section 3). An interesting asymptotic approximation for $Z_n(\alpha)$, when $n \to \infty$, has been studied in Mukherjee (2016), and we apply it in an example where $n = 200$ (see Section 6.4, and Section 2 in the Supplementary Material).

2.2 Prior Distributions

To complete the specification of the Bayesian model for the rankings $\mathbf{R}_1, \ldots, \mathbf{R}_N$, a prior for its parameters is needed. We assume a priori that α and $\boldsymbol{\rho}$ are independent.

An obvious choice for the prior for $\boldsymbol{\rho}$ in the context of the Mallows likelihood is to utilize the Mallows model family also in setting up a prior for $\boldsymbol{\rho}$, and let $\pi(\boldsymbol{\rho}) = \pi(\boldsymbol{\rho}|\alpha_0, \boldsymbol{\rho}_0) \propto \exp\left\{-\frac{\alpha_0}{n} d(\boldsymbol{\rho}, \boldsymbol{\rho}_0)\right\}$. Here α_0 and $\boldsymbol{\rho}_0$ are fixed hyperparameters, with $\boldsymbol{\rho}_0$ specifying the ranking that is a priori thought most likely, and α_0 controlling the tightness of the prior around $\boldsymbol{\rho}_0$. Since α_0 is fixed, $Z_n(\alpha_0)$ is a constant. Note that combining the likelihood with the prior $\pi(\boldsymbol{\rho}|\alpha_0, \boldsymbol{\rho}_0)$ above has the same effect on inference as involving an additional hypothetical assessor $j = 0$, say, who then provides the ranking $\mathbf{R}_0 = \boldsymbol{\rho}_0$ as data, with α_0 fixed.

If we were to elicit a value for α_0, we could reason as follows. Consider, for $\boldsymbol{\rho}_0$ fixed, the prior expectation $g_n(\alpha_0) := E_{\pi(\boldsymbol{\rho})}(d(\boldsymbol{\rho}, \boldsymbol{\rho}_0)|\alpha_0, \boldsymbol{\rho}_0)$. Because of the assumed right invariance

of the distance $d(\cdot, \cdot)$, this expectation is independent of $\boldsymbol{\rho}_0$, which is why $g_n(\cdot)$ depends only on α_0. Moreover, $g_n(\alpha_0)$ is obviously decreasing in α_0. For the footrule and Spearman distances, which are defined as sums of item specific deviations $|\rho_{0i} - \rho_i|$ or $|\rho_{0i} - \rho_i|^2$, $g_n(\alpha_0)$ can be interpreted as the expected (average, per item) error in the prior ranking $\pi(\boldsymbol{\rho}|\alpha_0, \boldsymbol{\rho}_0)$ of the consensus. A value for α_0 is now elicited by first choosing a target level τ_0, say, which would realistically correspond to such an a priori expected error size, and then finding the value α_0 such that $g_n(\alpha_0) = \tau_0$. This procedure requires numerical evaluation of the function $g_n(\alpha_0)$ over a range of suitable α_0 values. In this paper, we employ only the uniform prior $\pi(\boldsymbol{\rho}) = (n!)^{-1} 1_{\mathcal{P}_n}(\boldsymbol{\rho})$ in the space $\mathcal{P}_n$ of $n-$dimensional permutations, corresponding to $\alpha_0 = 0$.

For the scale parameter α we have in this paper used a truncated exponential prior, with density $\pi(\alpha|\lambda) = \lambda e^{-\lambda\alpha} 1_{[0,\alpha_{\max}]}(\alpha)/(1 - e^{-\lambda\alpha_{\max}})$, where the cut-off point $\alpha_{\max} < \infty$ is large compared to the values supported by the data. In practice, in the computations involving sampling of values for α, truncation was never applied. We show in Figure 3 of Section 3.3 on simulated data, that the inferences on $\boldsymbol{\rho}$ are almost completely independent of the choice of the value of λ. Also a theoretical argument for this is provided in that same section, although it is tailored more specifically to the numerical approximations of $Z_n(\alpha)$. For these reasons, in all our data analyses, we assigned λ a fixed value. We chose values for λ close to 0, depending on the complexity of the data, thus implying a prior density for α which is quite flat in the region supported in practice by the likelihood. If a more elaborate elicitation of the prior for α for some reason were preferred, this could be achieved by computing, by numerical integration, values of the function $E_{\pi(\alpha)}(g_n(\alpha)|\lambda)$, selecting a realistic target τ, and solving $E_{\pi(\alpha)}(g_n(\alpha)|\lambda) = \tau$ for λ. In a similar fashion as earlier, also $E_{\pi(\alpha)}(g_n(\alpha)|\lambda)$ can be interpreted as an expected (average, per item) error in the ranking, but now by *errors* is meant those made by the assessors, relative to the consensus, and expectation is with respect to the prior $\pi(\alpha|\lambda)$.

2.3 Inference

Given the prior distributions $\pi(\boldsymbol{\rho})$ and $\pi(\alpha)$, and assuming prior independence of these variables, the posterior distribution for $\boldsymbol{\rho}$ and α is given by

$$P(\boldsymbol{\rho}, \alpha|\mathbf{R}_1, \ldots, \mathbf{R}_N) \propto \frac{\pi(\boldsymbol{\rho})\pi(\alpha)}{Z_n(\alpha)^N} \exp\left\{-\frac{\alpha}{n}\sum_{j=1}^{N} d(\mathbf{R}_j, \boldsymbol{\rho})\right\}. \tag{4}$$

Often one is interested in computing posterior summaries of this distribution. One such summary is the marginal posterior mode of $\boldsymbol{\rho}$ (the maximum a posteriori, MAP) from (4), which does not depend on α, and in case of uniform prior for $\boldsymbol{\rho}$ coincides with the ML estimator of $\boldsymbol{\rho}$ in (2). The marginal posterior distribution of $\boldsymbol{\rho}$ is given by

$$P(\boldsymbol{\rho}|\mathbf{R}_1, \ldots, \mathbf{R}_N) \propto \pi(\boldsymbol{\rho}) \int_0^\infty \frac{\pi(\alpha)}{Z_n(\alpha)^N} \exp\left\{-\frac{\alpha}{n}\sum_{j=1}^{N} d(\mathbf{R}_j, \boldsymbol{\rho})\right\} \mathrm{d}\alpha. \tag{5}$$

Given the data, $\mathbf{R} = \{\mathbf{R}_1, \ldots, \mathbf{R}_N\}$ and the consensus ranking $\boldsymbol{\rho}$, the sum of distances, $T(\boldsymbol{\rho}, \mathbf{R}) = \sum_{j=1}^{N} d(\mathbf{R}_j, \boldsymbol{\rho})$, takes only a finite set of discrete values $\{t_1, t_2, ...t_m\}$, where m

depends on the distance $d(\cdot,\cdot)$, on the sample size N, and on n. Therefore, the set of all permutations $\mathcal{P}_n$ can be partitioned into the sets $H_i = \{\boldsymbol{r} \in \mathcal{P}_n : T(\boldsymbol{r},\ \mathbf{R}) = t_i\}$ for each distance t_i. These sets are level sets of the posterior marginal distribution in (5), as all $\boldsymbol{r} \in H_i$ have the same posterior marginal probability. The level sets do not depend on α but the posterior distribution shared by the permutations in each set does.

In applications, the interest often lies in computing posterior probabilities of more complex functions of the consensus $\boldsymbol{\rho}$, for example the posterior probability that a certain item has consensus rank lower than a given level ("among the top 5", say), or that the consensus rank of a certain item is higher than the consensus rank of another one. These probabilities cannot be readily obtained within the maximum likelihood approach, while the Bayesian setting very naturally allows to approximate any posterior summary of interest by means of a Markov Chain Monte Carlo algorithm, which at convergence samples from the posterior distribution (4).

2.4 Metropolis-Hastings Algorithm for Complete Rankings

In order to obtain samples from the posterior in equation (4), we iterate between two steps. In one step we update the consensus ranking. Starting with $\alpha \geq 0$ and $\boldsymbol{\rho} \in \mathcal{P}_n$, we first update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$ according to a distribution which is centered around the current rank $\boldsymbol{\rho}$.

Definition 1 *Leap-and-Shift Proposal (L&S). Fix an integer $L \in \{1,\dots,\lfloor (n-1)/2 \rfloor\}$ and draw a random number $u \sim \mathcal{U}\{1,\dots,n\}$. Define, for a given $\boldsymbol{\rho}$, the set of integers $\mathcal{S} = \{\max(1,\rho_u - L), \min(n,\rho_u + L)\} \setminus \{\rho_u\}$, $\mathcal{S} \subseteq \{1,\dots,n\}$, and draw a random number r uniformly in $\mathcal{S}$. Let $\boldsymbol{\rho}^* \in \{1,2,...n\}^n$ have elements $\rho^*_u = r$ and $\rho^*_i = \rho_i$ for $i \in \{1,\dots,n\} \setminus \{u\}$, constituting the leap step. Now, define $\Delta = \rho^*_u - \rho_u$ and the proposed $\boldsymbol{\rho}' \in \mathcal{P}_n$ with elements*

$$\rho'_i = \begin{cases} \rho^*_u & \text{if } \rho_i = \rho_u \\ \rho_i - 1 & \text{if } \rho_u < \rho_i \leq \rho^*_u \text{ and } \Delta > 0 \\ \rho_i + 1 & \text{if } \rho_u > \rho_i \geq \rho^*_u \text{ and } \Delta < 0 \\ \rho_i & \text{else}\ , \end{cases}$$

for $i = 1,\dots,n$, constituting the shift step.

The probability mass function associated to the transition is given by

$$\begin{aligned} P_L(\boldsymbol{\rho}'|\boldsymbol{\rho}) &= \sum_{u=1}^{n} P_L(\boldsymbol{\rho}'|U=u,\boldsymbol{\rho}) P(U=u) \\ &= \frac{1}{n}\sum_{u=1}^{n}\left\{ 1_{\{\boldsymbol{\rho}_{-u}\}}(\boldsymbol{\rho}^*_{-u}) \cdot 1_{\{0<|\rho_u - \rho^*_u|\leq L\}}(\boldsymbol{\rho}^*_u) \cdot \left[\frac{1_{\{L+1,\dots,n-L\}}(\boldsymbol{\rho}_u)}{2L} + \sum_{l=1}^{L} \frac{1_{\{l\}}(\boldsymbol{\rho}_u) + 1_{\{n-l+1\}}(\boldsymbol{\rho}_u)}{L+l-1} \right] \right\} \\ &+ \frac{1}{n}\sum_{u=1}^{n}\left\{ 1_{\{\boldsymbol{\rho}_{-u}\}}(\boldsymbol{\rho}^*_{-u}) \cdot 1_{\{|\rho_u - \rho^*_u|=1\}}(\boldsymbol{\rho}^*_u) \cdot \left[\frac{1_{\{L+1,\dots,n-L\}}(\boldsymbol{\rho}^*_u)}{2L} + \sum_{l=1}^{L} \frac{1_{\{l\}}(\boldsymbol{\rho}^*_u) + 1_{\{n-l+1\}}(\boldsymbol{\rho}^*_u)}{L+l-1} \right] \right\}, \end{aligned}$$

where $\boldsymbol{\rho}_{-u} = \{\rho_i;\ i \neq u\}$.

Proposition 1 *The leap-and-shift proposal $\boldsymbol{\rho}' \in \mathcal{P}_n$ is a local perturbation of $\boldsymbol{\rho}$, separated from $\boldsymbol{\rho}$ by a Ulam distance 1 .*

Proof From the definition and by construction, $\boldsymbol{\rho}^* \notin \mathcal{P}_n$, since there exist two indices $i \neq j$ such that $\rho^*_i = \rho^*_j$. The shift of the ranks by Δ brings $\boldsymbol{\rho}^*$ to $\boldsymbol{\rho}'$ back into $\mathcal{P}_n$. The Ulam distance $d(\boldsymbol{\rho},\boldsymbol{\rho}')$ is the number of edit operations needed to convert $\boldsymbol{\rho}$ to $\boldsymbol{\rho}'$, where each edit operation involves deleting a character and inserting it in a new place. This is equal to 1, following Gopalan et al. (2006). ■

The acceptance probability when updating $\boldsymbol{\rho}$ in the Metropolis-Hastings algorithm is

$$\min\left\{1, \frac{P_L(\boldsymbol{\rho}|\boldsymbol{\rho}')\pi(\boldsymbol{\rho}')}{P_L(\boldsymbol{\rho}'|\boldsymbol{\rho})\pi(\boldsymbol{\rho})} \exp\left[-\frac{\alpha}{n}\sum_{j=1}^{N}\left\{ d(\mathbf{R}_j,\boldsymbol{\rho}') - d(\mathbf{R}_j,\boldsymbol{\rho}) \right\} \right] \right\}. \tag{6}$$

The leap-and-shift proposal is not symmetric, thus the ratio $P_L(\boldsymbol{\rho}|\boldsymbol{\rho}')/P_L(\boldsymbol{\rho}'|\boldsymbol{\rho})$ does not cancel in (6). The parameter L is used for tuning this acceptance probability.

The term $\sum_{j=1}^{N}\{d(\mathbf{R}_j,\boldsymbol{\rho}') - d(\mathbf{R}_j,\boldsymbol{\rho})\}$ in (6) can be computed efficiently, since most elements of $\boldsymbol{\rho}$ and $\boldsymbol{\rho}'$ are equal. Let $\rho_i = \rho'_i$ for $i \in E \subset \{1,\dots,n\}$, and $\rho_i \neq \rho'_i$ for $i \in E^c$. For the footrule and Spearman distances, we then have

$$\sum_{j=1}^{N}\left\{ d(\mathbf{R}_j,\boldsymbol{\rho}') - d(\mathbf{R}_j,\boldsymbol{\rho}) \right\} = \sum_{j=1}^{N}\left\{ \sum_{i\in E^c} |R_{ij} - \rho'_i|^p - \sum_{i\in E^c} |R_{ij} - \rho_i|^p \right\}, \tag{7}$$

for $p \in \{1,2\}$. For the Kendall distance, instead, we get

$$\begin{aligned} &\sum_{j=1}^{N}\left\{ d(\mathbf{R}_j,\boldsymbol{\rho}') - d(\mathbf{R}_j,\boldsymbol{\rho}) \right\} = \\ &= \sum_{j=1}^{N}\left\{ \sum_{1\leq k<l\leq n} 1\left[(R_{kj} - R_{lj})(\rho'_k - \rho'_l) > 0 \right] - 1\left[(R_{kj} - R_{lj})(\rho_k - \rho_l) > 0 \right] \right\} \\ &= \sum_{j=1}^{N}\left\{ \sum_{k\in E^c\setminus\{n\}} \sum_{l\in\{E^c\cap\{l>k\}\}} 1\left[(R_{kj} - R_{lj})(\rho'_k - \rho'_l) > 0 \right] - 1\left[(R_{kj} - R_{lj})(\rho_k - \rho_l) > 0 \right] \right\}. \end{aligned}$$

Hence, by storing the set E^c at each MCMC iteration, the computation of (6) involves a sum over fewer terms, speeding up the algorithm consistently.

The second step of the algorithm updates the value of α. We sample a proposal α' from a lognormal distribution $\log\mathcal{N}(\log(\alpha),\sigma^2_\alpha)$ and accept it with probability

$$\min\left\{1, \frac{Z_n(\alpha)^N \pi(\alpha')\alpha'}{Z_n(\alpha')^N \pi(\alpha)\alpha} \exp\left[-\frac{(\alpha'-\alpha)}{n}\sum_{j=1}^{N} d(\mathbf{R}_j,\boldsymbol{\rho}) \right] \right\}, \tag{8}$$

where σ^2_α can be tuned to obtain a desired acceptance probability. A further parameter, named α_{jump}, can be used to update α only every α_{jump} updates of $\boldsymbol{\rho}$: the possibility to

tune this parameter ensures a better mixing of the MCMC in the different sparse data applications. The above described MCMC algorithm is summarized as Algorithm 1 of Appendix B. Note that the MCMC Algorithm 1 using the exact partition function $Z_n(\alpha)$ samples from the Mallows posterior in equation (4), as the number of MCMC iterations tends to infinity.

Section 3 investigates approximations of $Z_n(\alpha)$, and how they affect the MCMC and the estimate of the consensus $\boldsymbol{\rho}$. In Section 1 of the Supplementary Material we instead focus on aspects related to the practical choices involved in the use of our MCMC algorithm, and in particular we aim at defining possible strategies for tuning the MCMC parameters L and σ_α.

3. Approximating the Partition Function $Z_n(\alpha)$ via Off-line Importance Sampling

For Kendall's, Hamming and Cayley distances, the partition function $Z_n(\alpha)$ is available in close form, but this is not the case for footrule and Spearman distances. To handle these cases, we propose an approximation of the partition function $Z_n(\alpha)$ based on importance sampling. Since we focus on right-invariant distances, the partition function does not depend on $\boldsymbol{\rho}$. Hence, we can obtain an off-line approximation of the partition function on a grid of α values, interpolate it to yield an estimate of $Z_n(\alpha)$ over a continuous range, and then read off needed values to compute the acceptance probabilities very rapidly.

We study the convergence of the importance sampler theoretically (Section 3.2) and numerically (Sections 3.1, 3.3), with a series of experiments aimed at demonstrating the quality of the approximation, and its impact in inference. We here show the results obtained with the footrule distance, but we obtained similar results with the Spearman distance. We also summarize in the Supplementary Material (Section 2) a further possible approximation of $Z_n(\alpha)$, namely the asymptotic proposal in Mukherjee (2016).

We briefly discuss the pseudo-marginal approaches for tackling intractable Metropolis-Hastings ratios, which could in principle be an interesting alternative. We refer to Beaumont (2003), Andrieu and Roberts (2009), and Murray et al. (2012) for a full description of the central methodologies. The idea is to replace $P(\boldsymbol{\rho}, \alpha|\mathbf{R})$ in (4) with a non-negative unbiased estimator $\hat{P}$, such that for some $C > 0$ we have $\mathbb{E}[\hat{P}] = CP$. The approximate acceptance ratio then uses $\hat{P}$, but this results in an algorithm still targeting the exact posterior. An unbiased estimate of the posterior P can be obtained via importance sampling if it is possible to simulate directly from the likelihood. This is not the case in our model, as there are no algorithms available to sample from the Mallows model with, say, the footrule distance. Neither is use of exact simulation possible for our model. The approach in Murray et al. (2012) extends the model by introducing an auxiliary variable, and uses a proposal distribution in the MCMC such that the partition functions cancel. A useful proposal for this purpose would in our case be based on the Mallows likelihood, so that again one would need to be able to sample from it, which is not feasible.

Our suggestion is instead to estimate the partition function directly, using an Importance Sampling (IS) approach. For K rank vectors $\mathbf{R}^1, \ldots, \mathbf{R}^K$ sampled from an IS auxiliary

distribution $q(\mathbf{R})$, the unbiased IS estimate of $Z_n(\alpha)$ is given by

$$\hat{Z}_n(\alpha) = K^{-1} \sum_{k=1}^{K} \exp\{-(\alpha/n) d(\mathbf{R}^k, \mathbf{1}_n)\} q(\mathbf{R}^k)^{-1}. \tag{9}$$

The more $q(\mathbf{R})$ resembles the Mallows likelihood (1), the smaller is the variance of $\hat{Z}_n(\alpha)$. On the other hand, it must be computationally feasible to sample from $q(\mathbf{R})$. We use the following pseudo-likelihood approximation of the target (1). Let $\{i_1, \ldots, i_n\}$ be a uniform sample from $\mathcal{P}_n$, which gives the order of the pseudo-likelihood factorization. Then

$$P(\mathbf{R}|\mathbf{1}_n) = P(R_{i_1}|R_{i_2}, \ldots, R_{i_n}, \mathbf{1}_n) P(R_{i_2}|R_{i_3}, \ldots, R_{i_n}, \mathbf{1}_n) \cdots P(R_{i_{n-1}}|R_{i_n}, \mathbf{1}_n) P(R_{i_n}|\mathbf{1}_n),$$

and the conditional distributions are given by

$$\begin{aligned}
P\left(R_{i_n}|\mathbf{1}_n\right) &= \frac{\exp\left\{-(\alpha/n) d\left(R_{i_n}, i_n\right)\right\} \cdot 1_{[1,\ldots,n]}(R_{i_n})}{\sum_{r_n \in \{1,\ldots,n\}} \exp\left\{-(\alpha/n) d\left(r_n, i_n\right)\right\}}, \\
P\left(R_{i_{n-1}}|R_{i_n}, \mathbf{1}_n\right) &= \frac{\exp\left\{-(\alpha/n) d\left(R_{i_{n-1}}, i_{n-1}\right)\right\} \cdot 1_{[\{1,\ldots,n\}\setminus\{R_{i_n}\}]}(R_{i_{n-1}})}{\sum_{r_{n-1} \in \{1,\ldots,n\}\setminus\{R_{i_n}\}} \exp\left\{-(\alpha/n) d\left(r_{n-1}, i_{n-1}\right)\right\}}, \\
&\vdots \\
P\left(R_{i_2}|R_{i_3}, \ldots, R_{i_n}, \mathbf{1}_n\right) &= \frac{\exp\left\{-(\alpha/n) d\left(R_{i_2}, i_2\right)\right\} \cdot 1_{[\{1,\ldots,n\}\setminus\{R_{i_3},\ldots,R_{i_n}\}]}(R_{i_2})}{\sum_{r_2 \in \{1,\ldots,n\}\setminus\{R_{i_3},\ldots,R_{i_n}\}} \exp\left\{-(\alpha/n) d\left(r_2, i_2\right)\right\}}, \\
P\left(R_{i_1}|R_{i_2}, \ldots, R_{i_n}, \mathbf{1}_n\right) &= 1_{[\{1,\ldots,n\}\setminus\{R_{i_2},\ldots,R_{i_n}\}]}(R_{i_1}).
\end{aligned}$$

Each factor is a simple univariate distribution. We sample R_{i_n} first, and then conditionally on that, $R_{i_{n-1}}$ and so on. The k-th full sample $\mathbf{R}^k$ has probability $q(\mathbf{R}^k) = P(R^k_{i_n}|\mathbf{1}_n)$ $P(R^k_{i_{n-1}}|R^k_{i_n}, \mathbf{1}_n) \cdots P(R^k_{i_2}|R^k_{i_3}, \ldots, R^k_{i_n}, \mathbf{1}_n)$. We observe that this pseudo-likelihood construction is similar to the sequential representation of the Plackett-Luce model with a Mallows parametrization of probabilities.

Note that, in principle, we could sample rankings $\mathbf{R}^k$ from the Mallows model with a different distance than the one of the target model (for example Kendall), or use the pseudo-likelihood approach with a different "proposal distance" other than the target distance. We experimented with these alternatives, but keeping the pseudo-likelihood with the same distance as the one in the target was most accurate and efficient (results not shown). In what follows the distance in (9) is the same as the distance in (4).

3.1 Testing the Importance Sampler

We experimented by increasing the number K of importance samples in powers of ten, over a discrete grid of 100 equally spaced α values between 0.01 and 10 (this is the range of α which turned out to be relevant in all our applications, typically $\alpha < 5$). We produced a smooth partition function simply using a polynomial of degree 10. The ratio $\hat{Z}^K_n(\alpha)/Z_n(\alpha)$ as a function of α is shown in Figure 1 for $n = 10, 20, 50$ and when using different values of K: the ratio quickly approaches 1 when increasing K; for larger n, a larger K is needed to ensure precision, but $K = 10^6$ seems enough to give very precise estimates.

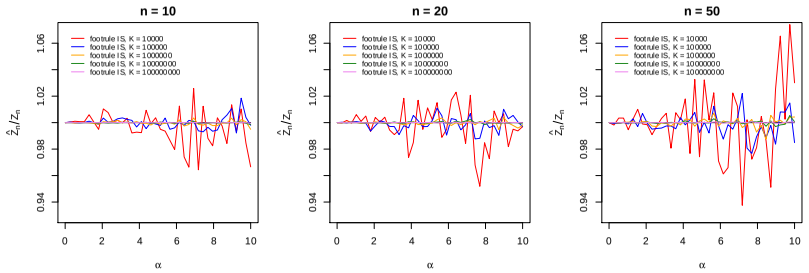


Figure 1: Ratio of the approximate partition function computed via IS to the exact, $\hat{Z}_n(\alpha)/Z_n(\alpha)$, as a function of α, when using the footrule distance. From left to right, $n = 10, 20, 50$; different colors refer to different values of K, as stated in the legend.

K	10^2	10^3	10^4	10^5	10^6	10^7	10^8
$n = 75$	152.036	0.921	0.373	0.084	0.056	0.005	0.004
$n = 100$	67.487	1.709	0.355	0.187	0.045	0.018	0.004

Table 1: Approximation of the partition function via the IS for the footrule model: maximum relative error ϵ_K from equation (10), between the current and the previous K, for $n = 75$ and 100.

When n is larger than 50, no exact expression for $Z_n(\alpha)$ is available. Then, we directly compare the estimated $\hat{Z}_n^K(\alpha)$ for increasing K, to check whether the estimates stabilize. We thus inspect the maximum relative error

$$\epsilon_K = \max_{\alpha} \left[\frac{\left| \hat{Z}_n^K(\alpha) - \hat{Z}_n^{K/10}(\alpha) \right|}{\left| \hat{Z}_n^{K/10}(\alpha) \right|} \right] \tag{10}$$

for $K = 10^2, \ldots, 10^8$. Results are shown in Table 1 for $n = 75$ and 100. For both values of n we see that the estimates quickly stabilize, and $K = 10^6$ appears to give good approximations. The computations shown here were performed on a desktop computer, and the off-line computation with $K = 10^6$ samples for $n = 10$ took less than 15 minutes, with no efforts for parallelizing the algorithm, which would be easy and beneficial. $K = 10^6$ samples for $n = 100$ were obtained on a 64-cores computing cluster in 12 minutes.

3.2 Effect of $\hat{Z}_n(\alpha)$ on the MCMC

In this Section we report theoretical results regarding the convergence of the MCMC, when using the IS approximation of the partition function.

Proposition 2 *Algorithm 1 of Appendix B using $\hat{Z}_n(\alpha)$ in (9) instead of $Z_n(\alpha)$ converges to the posterior distribution proportional to*

$$\frac{1}{\hat{C}(\mathbf{R})} \pi(\boldsymbol{\rho}) \pi(\alpha) \hat{Z}_n(\alpha)^{-N} \exp\left\{ -\frac{\alpha}{n} \sum_{j=1}^{N} d(\mathbf{R}_j, \boldsymbol{\rho}) \right\}, \tag{11}$$

with the normalizing factor $\hat{C}(\mathbf{R}) = \int \frac{\pi(\alpha)}{\hat{Z}_n(\alpha)^N} \sum_{\boldsymbol{\rho} \in \mathcal{P}_n} \pi(\boldsymbol{\rho}) \exp\left\{ -\frac{\alpha}{n} \sum_{j=1}^{N} d(\mathbf{R}_j, \boldsymbol{\rho}) \right\} d\alpha$.

Proof The acceptance probability of the MCMC in Algorithm 1 with the approximate partition function is given by (8) using $\hat{Z}_n(\alpha)$ in (9) instead of $Z_n(\alpha)$, which is exactly the acceptance probability needed for (11). ■

That $\hat{C}(\mathbf{R}) < \infty$ is an obvious consequence of our assumption, in Section 2.2, that the prior $\pi(\alpha)$ is supported by a finite interval $[0, \alpha_{\max}]$. The IS approximation $\hat{Z}_n(\alpha)$ converges to $Z_n(\alpha)$ as the number K of IS samples converges to infinity. In order to study this limit, let us change the notation to explicitly show this dependence and write $\hat{Z}_n^K(\alpha)$. Clearly, the approximate posterior (11) converges to the correct posterior (4) if K increases with N, $K = K(N)$, and

$$\lim_{N \to \infty} \left(\frac{\hat{Z}_n^{K(N)}(\alpha)}{Z_n(\alpha)} \right)^N = 1, \quad \text{for all } \alpha. \tag{12}$$

Proposition 3 *There exists a factor $c(\alpha, n, d(\cdot,\cdot))$ not depending on N, such that, if $K = K(N)$ tends to infinity as $N \to \infty$ faster than $c(\alpha, n, d(\cdot,\cdot)) \cdot N^2$, then (12) holds.*

Proof We see that

$$\left(\frac{\hat{Z}_n^{K(N)}(\alpha)}{Z_n(\alpha)} \right)^N = \exp\left\{ N \log\left(1 + \frac{\hat{Z}_n^{K(N)}(\alpha) - Z_n(\alpha)}{Z_n(\alpha)} \right) \right\}$$

tends to 1 in probability as $K(N) \to \infty$ when $N \to \infty$ if

$$\frac{\hat{Z}_n^{K(N)}(\alpha) - Z_n(\alpha)}{Z_n(\alpha)} \tag{13}$$

tends to 0 in probability faster than $1/N$. Since (9) is a sum of i.i.d. variables, there exists a constant $c = c(\alpha, n, d(\cdot,\cdot))$ depending on α, n and the distance $d(\cdot,\cdot)$ (but not on N) such that

$$\sqrt{K(N)} (\hat{Z}_n^{K(N)}(\alpha) - Z_n(\alpha)) \xrightarrow{\mathcal{L}} \mathcal{N}(0, c^2),$$

in law as $K(N) \to \infty$. Therefore, for (13) tending to 0 faster than $1/N$, it is sufficient that $K(N)$ grows faster than N^2. The speed of convergence to 1 of (12) depends on $c = c(\alpha, n, d(\cdot,\cdot))$. ■

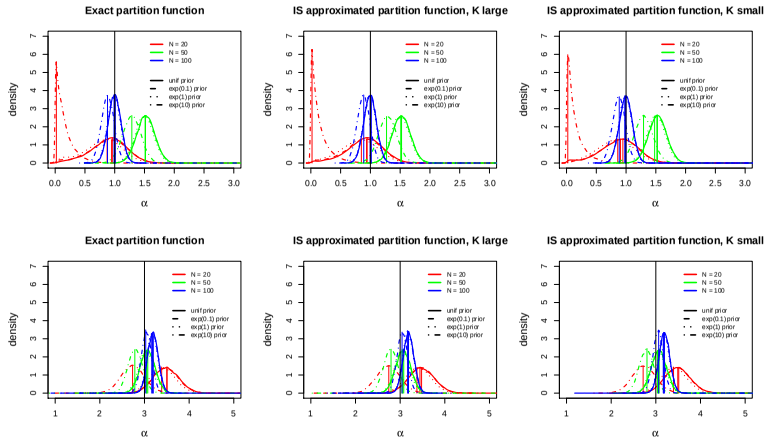


Figure 2: Results of the simulations described in Section 3.3, when $n = 20$. In each plot, posterior density of α (the black vertical line indicates α_{true}) obtained for various choices of N (different colors), and for different choices of the prior for α (different line types), as stated in the legend. From left to right, MCMC run with the exact $Z_n(\alpha)$, with the IS approximation $\hat{Z}_n^K(\alpha)$ with $K = 10^8$, and with the IS approximation $\hat{Z}_n^K(\alpha)$ with $K = 10^4$. First row: $\alpha_{\text{true}} = 1$; second row: $\alpha_{\text{true}} = 3$.

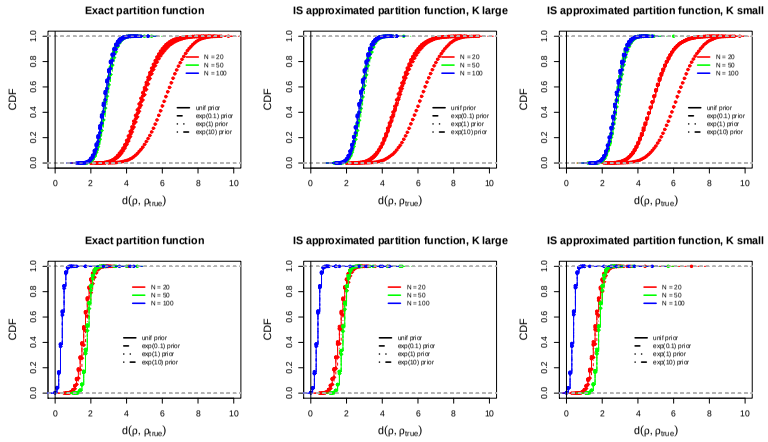


Figure 3: Results of the simulations described in Section 3.3, when $n = 20$. In each plot, posterior CDF of $d(\boldsymbol{\rho}, \boldsymbol{\rho}_{\text{true}})$ obtained for various choices of N (different colors), and for different choices of the prior for α (different line types), as stated in the legend. From left to right, MCMC run with the exact $Z_n(\alpha)$, with the IS approximation $\hat{Z}_n^K(\alpha)$ with $K = 10^8$, and with the IS approximation $\hat{Z}_n^K(\alpha)$ with $K = 10^4$. First row: $\alpha_{\text{true}} = 1$; second row: $\alpha_{\text{true}} = 3$.

3.3 Testing Approximations of the MCMC in Inference

We report results from extensive simulation experiments carried out in several different parameter settings, to investigate if our algorithm provides correct posterior inferences. In addition, we study the sensitivity of the posterior distributions to differences in the prior specifications, and demonstrate their increased precision when the sample size N grows. We explore the robustness of inference when using approximations of the partition function $Z_n(\alpha)$, both when obtained by applying our IS approach, and when using, for large n, the asymptotic approximation $Z_{\text{lim}}(\alpha)$ proposed in Mukherjee (2016). We focus here on the footrule distance since it allows us to explore all these different settings, being also the preferred distance in the experiments reported in Section 6. Some model parameters are kept fixed in the various cases: $\alpha_{\text{jump}} = 10$, $\sigma_\alpha = 0.15$, and $L = n/5$ (for the tuning of the two latter parameters, see the simulation study in the Supplementary Material, Section 1). Computing times for the simulations, performed on a laptop computer, varied depending on the value of n and N, from a minimum of $24''$ in the smallest case with $n = 20$ and $N = 20$, to a maximum of $3'22''$ for $n = 100$ and $N = 1000$.

First, we generated data from a Mallows model with $n = 20$ items, using samples from $N = 20, 50$, and 100 assessors, a setting of moderate complexity. The value of α_{true} was chosen to be either 1 or 3, and $\boldsymbol{\rho}_{\text{true}}$ was fixed at $(1, \ldots, n)$. To generate the data, we run the MCMC sampler (see Appendix C) for 10^5 burn-in iterations, and collected one sample every 100 iterations after that (these settings were kept in all data generations). In the analysis, we considered the performance of the method when using the IS approximation $\hat{Z}_n^K(\alpha)$ with $K = 10^4$ and 10^8, then comparing the results with those based on the exact $Z_n(\alpha)$. In each case, we run the MCMC for 10^6 iterations, with 10^5 iterations for burn-in. Finally, we varied the prior for α to be either the nonintegrable uniform or the exponential using hyperparameter values $\lambda = 0.1, 1$ and 10. The results are shown in Figures 2 for α and 3 for $\boldsymbol{\rho}$. As expected, we can see the precision and the accuracy of the marginal posterior distributions increasing, both for α and $\boldsymbol{\rho}$, with N becoming larger. For smaller values of α_{true}, the marginal posterior for α is more dispersed, and $\boldsymbol{\rho}$ is stochastically farther from $\boldsymbol{\rho}_{\text{true}}$. These results are remarkably stable against varying choices of the prior for α, even when the quite strong exponential prior with $\lambda = 10$ was used (with one exception: in the case of $N = 20$ the rather dispersed data generated by $\alpha_{\text{true}} = 1$ were not sufficient to overcome the control of the exponential prior with $\lambda = 10$, which favored even smaller values of α; see Figure 2, top panels). Finally and most importantly, we see that inference on both α and $\boldsymbol{\rho}$ is completely unaffected by the approximation of $Z_n(\alpha)$ already when $K = 10^4$.

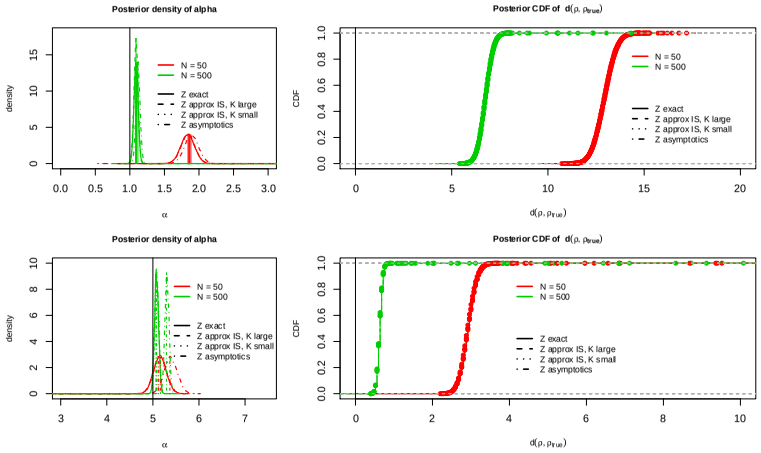


Figure 4: Results of the simulations described in Section 3.3, when $n = 50$. Left, posterior density of α (the black vertical line indicates α_{true}) obtained for various choices of N (different colors), and when using the exact, or different approximations to the partition function (different line types), as stated in the legend. Right, posterior CDF of $d(\boldsymbol{\rho}, \boldsymbol{\rho}_{\text{true}})$ in the same settings. First row: $\alpha_{\text{true}} = 1$; second row: $\alpha_{\text{true}} = 5$.

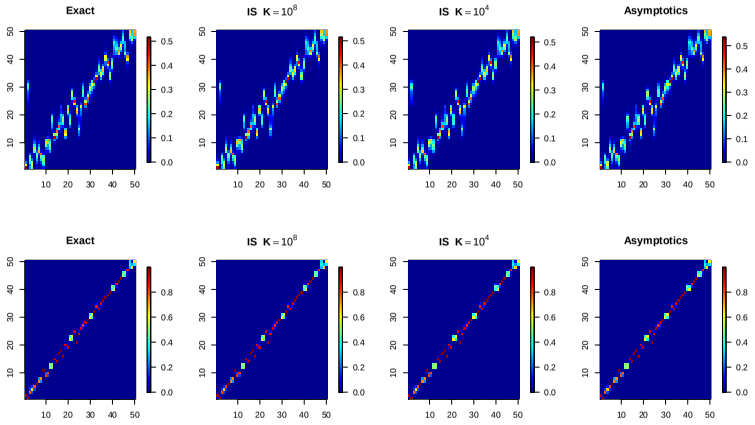


Figure 5: Results of the simulations described in Section 3.3, when $n = 50$ and $\alpha_{\text{true}} = 5$. In the x-axis items are ordered according to the true consensus $\boldsymbol{\rho}_{\text{true}}$. Each column j represents the posterior marginal density of item j in the consensus $\boldsymbol{\rho}$. Concentration along the diagonal is a sign of success of inference. From left to right, results obtained with the exact $Z_n(\alpha)$, with the IS approximation $\hat{Z}_n^K(\alpha)$ with $K = 10^8$, with the IS approximation $\hat{Z}_n^K(\alpha)$ with $K = 10^4$, and with $Z_{\text{lim}}(\alpha)$. First row: $N = 50$; second row: $N = 500$.

In a second experiment, we generated data using $n = 50$ items, $N = 50$ or 500 assessors, and scale parameter $\alpha_{\text{true}} = 1$ or 5. This increase in the value of n gave us some basis for comparing the results obtained by using the IS approximation of $Z_n(\alpha)$ with those from the asymptotic approximation $Z_{\text{lim}}(\alpha)$ of Mukherjee (2016), while still retaining also the possibility of using the exact $Z_n(\alpha)$. For the analysis, all the previous MCMC settings were kept, except for the prior for α: since results from $n = 20$ turned out to be independent of the choice of the prior, here we used the same exponential prior with $\lambda = 0.1$ in all comparisons (see the discussion in Section 2.2). The results are shown in Figures 4 and 5. Again, we observe substantially more accurate results for larger values of N and α_{true}. Concerning the impact of approximations to $Z_n(\alpha)$, we notice that, even in this case of larger n, the marginal posterior of $\boldsymbol{\rho}$ appears completely unaffected by the partition function not being exact (see Figure 4, right panels, and Figure 5). In the marginal posterior for α (Figure 4, left panels), there are no differences between using the IS approximations and the exact, but there is a difference between Z_{lim} and the other approximations: Z_{lim} appears to be systematically slightly worse.

Finally, we generated data from the Mallows model with $n = 100$ items, $N = 100$ or 1000 assessors, and using $\alpha_{\text{true}} = 5$ or 10. Because of this large value of n we were no longer able to compute the exact $Z_n(\alpha)$, hence we only compared results from the different

approximations. We kept the same MCMC settings as for $n = 50$, both in data generation and analysis. The results are shown in Figures 3 and 4 of the Supplementary Material, Section 3. Also in this case, we observe substantially more accurate estimates with larger values of N and α_{true}, establishing an overall stable performance of the method. Here, using the small number $K = 10^4$ of samples in the IS approximation has virtually no effect on the accuracy of the marginal posterior for α, while a small effect can be detected from using the asymptotic approximation (Figure 3 of the Supplementary Material, left panels). However, again, the marginal posterior for $\boldsymbol{\rho}$ appears completely unaffected by the considered approximations in the partition function (Figure 3, right panels, and Figure 4 of the Supplementary Material).

In conclusion, the main positive result from the perspective of practical applications was the relative lack of sensitivity of the posterior inferences to the specification of the prior for the scale parameter α, and the apparent robustness of the marginal posterior inferences on $\boldsymbol{\rho}$ on the choice of the approximation of the partition function $Z_n(\alpha)$. The former property was not an actual surprise, as it can be understood to be a consequence of the well-known Bernstein-von Mises principle: with sufficient amounts of data, the likelihood dominates the influence of the prior.

The second observation deserves a somewhat closer inspection, however. The marginal posterior $P(\alpha|\mathbf{R})$, considered in Figures 2 and 4 (left), and in Figure 3 (left) of the Supplementary Material, is obtained from the joint posterior (4) by simple summation over $\boldsymbol{\rho}$, then getting the expression

$$P(\alpha|\mathbf{R}) \propto \pi(\alpha)C(\alpha;\mathbf{R})/(Z_n(\alpha))^N, \tag{14}$$

where $C(\alpha;\mathbf{R}) = \sum_{\boldsymbol{\rho}\in\mathcal{P}_n} \pi(\boldsymbol{\rho}) \exp\left\{-\frac{\alpha}{n}\sum_{j=1}^N d(\mathbf{R}_j,\boldsymbol{\rho})\right\}$. For a proper understanding of the structure of the joint posterior and its modification (11), it is helpful to first factorize (4) into the product

$$P(\alpha,\boldsymbol{\rho}|\mathbf{R}) = P(\alpha|\mathbf{R})P(\boldsymbol{\rho}|\alpha,\mathbf{R}), \tag{15}$$

where then

$$P(\boldsymbol{\rho}|\alpha,\mathbf{R}) = [C(\alpha;\mathbf{R})]^{-1}\pi(\boldsymbol{\rho}) \exp\left\{-\frac{\alpha}{n}\sum_{j=1}^N d(\mathbf{R}_j,\boldsymbol{\rho})\right\}. \tag{16}$$

The joint posterior (11), which arises from replacing the partition function $Z_n(\alpha)$ by its approximation $\hat{Z}_n(\alpha)$, can be similarly expressed as the product

$$\hat{P}(\alpha,\boldsymbol{\rho}|\mathbf{R}) = \hat{P}(\alpha|\mathbf{R})P(\boldsymbol{\rho}|\alpha,\mathbf{R}), \tag{17}$$

where

$$\hat{P}(\alpha|\mathbf{R}) = [\hat{C}(\mathbf{R})]^{-1}(Z_n(\alpha)/\hat{Z}_n(\alpha))^N P(\alpha|\mathbf{R}). \tag{18}$$

This requires that the normalizing factor $\hat{C}(\mathbf{R})$ already introduced in (11), and here expressed as

$$\hat{C}(\mathbf{R}) \equiv \int (Z_n(\alpha)/\hat{Z}_n(\alpha))^N P(\alpha|\mathbf{R})d\alpha, \tag{19}$$

is finite. By comparing (15) and (17) we see that, under this condition, the posterior $\hat{P}(\alpha,\boldsymbol{\rho}|\mathbf{R})$ arises from $P(\alpha,\boldsymbol{\rho}|\mathbf{R})$ by changing the expression (14) of the marginal posterior for α into (18), while the conditional posterior $P(\boldsymbol{\rho}|\alpha,\mathbf{R})$ for $\boldsymbol{\rho}$, given α, remains the same in both cases. Thus, the marginal posteriors $P(\boldsymbol{\rho}|\mathbf{R})$ and $\hat{P}(\boldsymbol{\rho}|\mathbf{R})$ for $\boldsymbol{\rho}$ arise as mixtures of the same conditional posterior $P(\boldsymbol{\rho}|\alpha,\mathbf{R})$ with respect to two different mixing distributions, $P(\alpha|\mathbf{R})$ and $\hat{P}(\alpha|\mathbf{R})$.

It is obvious from (18) and (19) that $\hat{P}(\alpha|\mathbf{R}) = P(\alpha|\mathbf{R})$ would hold if the ratio $Z_n(\alpha)/\hat{Z}_n(\alpha)$ would be exactly a constant in α, and this would also entail the exact equality $\hat{P}(\boldsymbol{\rho}|\mathbf{R}) = P(\boldsymbol{\rho}|\mathbf{R})$. It was established in (12) that, in the IS scheme, $Z_n(\alpha)/\hat{Z}_n(\alpha) \to 1$ as $K \to \infty$. Thus, for large enough K, $(Z_n(\alpha)/\hat{Z}_n(\alpha))^N \approx 1$ holds as an approximation (see Proposition 3). Importantly, however, (18) shows that the approximation is only required to hold well on the effective support of $P(\alpha|\mathbf{R})$, and this support is narrow when N is large. This is demonstrated clearly in Figures 2 and 4 (left), and in Figure 3 (left) of the Supplementary Material. On this support, because of uniform continuity in α, also the integrand $P(\boldsymbol{\rho}|\alpha,\mathbf{R})$ in (16) remains nearly a constant. In fact, experiments (results not shown) performed by varying α over a much wider range of fixed values, while keeping the same $\mathbf{R}$, gave remarkably stable results for the conditional posterior $P(\boldsymbol{\rho}|\alpha,\mathbf{R})$. This contributes to the high degree of robustness in the posterior inferences on $\boldsymbol{\rho}$, making requirements of using large values of K much less stringent.

In Figures 3 and 4 (right), and in Figure 3 (right) of the Supplementary Material, we considered and compared the marginal posterior CDF's of the distance $d(\boldsymbol{\rho},\boldsymbol{\rho}_{\text{true}})$ under the schemes $P(\cdot|\mathbf{R})$ and $\hat{P}(\cdot|\mathbf{R})$. Using the shorthand $d^* = d(\boldsymbol{\rho},\boldsymbol{\rho}_{\text{true}})$, let

$$\begin{aligned}
F_{d^*}(x|\alpha,\mathbf{R}) &\equiv P(d(\boldsymbol{\rho},\boldsymbol{\rho}_{\text{true}}) \le x|\alpha,\mathbf{R}) = \sum_{\{\boldsymbol{\rho}:d(\boldsymbol{\rho},\boldsymbol{\rho}_{\text{true}})\le x\}} P(\boldsymbol{\rho}|\alpha,\mathbf{R}), \qquad (20)\\
F_{d^*}(x|\mathbf{R}) &\equiv \sum_{\{\boldsymbol{\rho}:d(\boldsymbol{\rho},\boldsymbol{\rho}_{\text{true}})\le x\}} P(\boldsymbol{\rho}|\mathbf{R}) = \int F_{d^*}(x|\alpha,\mathbf{R})P(\alpha|\mathbf{R})d\alpha,\\
\hat{F}_{d^*}(x|\mathbf{R}) &\equiv \sum_{\{\boldsymbol{\rho}:d(\boldsymbol{\rho},\boldsymbol{\rho}_{\text{true}})\le x\}} \hat{P}(\boldsymbol{\rho}|\mathbf{R}) = \int F_{d^*}(x|\alpha,\mathbf{R})\hat{P}(\alpha|\mathbf{R})d\alpha.
\end{aligned}$$

For example, in Figure 3 we display, for different priors, the CDF's $F_{d^*}(x|\mathbf{R})$ on the left, and $\hat{F}_{d^*}(x|\mathbf{R})$ in the middle and on the right, corresponding to two different IS approximations of the partition function. Like the marginal posteriors $P(\boldsymbol{\rho}|\mathbf{R})$ and $\hat{P}(\boldsymbol{\rho}|\mathbf{R})$ above, $F_{d^*}(x|\mathbf{R})$ and $\hat{F}_{d^*}(x|\mathbf{R})$ can be thought of as mixtures of the same function, here $F_{d^*}(x|\alpha,\mathbf{R})$, but with respect to two different mixing distributions, $P(\alpha|\mathbf{R})$ and $\hat{P}(\alpha|\mathbf{R})$. The same arguments, which were used above in support of the robustness of the posterior inferences on $\boldsymbol{\rho}$, apply here as well. Extensive empirical evidence for their justification is provided in Figures 3 and 4 (right), and in Figure 3 (right) of the Supplementary Material. Finally note that these arguments also strengthen considerably our earlier conclusion of the lack of sensitivity of the posterior inferences on $\boldsymbol{\rho}$ to the specification of the prior for α. For this, we only need to consider alternative priors, say, $\pi(\alpha)$ and $\hat{\pi}(\alpha)$, in place of the mixing distributions $P(\alpha|\mathbf{R})$ and $\hat{P}(\alpha|\mathbf{R})$.

4. Extensions to Partial Rankings and Heterogeneous Assessor Pool

We now relax two assumptions of the previous Sections, namely that each assessor ranks all n items and that the assessors are homogeneous, all sharing a common consensus ranking. This allows us to treat the important situation of pairwise comparisons, and of multiple classes of assessors, as incomplete data cases, within the same Bayesian Mallows framework.

4.1 Ranking of the Top Ranked Items

Often only a subset of the items is ranked: ranks can be missing at random, the assessors may only have ranked the, in-their-opinion, top-k items, or can be presented with a subset of items that they have to rank. These situations can be handled conveniently in our Bayesian framework, by applying data augmentation techniques. We start by explaining the method in the case of the top-k ranks, and then show briefly how it can be generalized to the other cases mentioned.

Suppose that each assessor j has ranked the subset of items $\mathcal{A}_j \subseteq \{A_1, A_2, \ldots, A_n\}$, giving them top ranks from 1 to $n_j = |\mathcal{A}_j|$. Let $R_{ij} = \mathbf{X}_j^{-1}(A_i)$ if $A_i \in \mathcal{A}_j$, while for $A_i \in \mathcal{A}_j^c$, R_{ij} is unknown, except for the constraint $R_{ij} > n_j$, $j = 1,\ldots,N$, and follows a symmetric prior on the permutations of $(n_j+1,\ldots,n)$. We define augmented data vectors $\tilde{\mathbf{R}}_1,\ldots,\tilde{\mathbf{R}}_N$ by assigning ranks to these non-ranked items randomly, using an MCMC algorithm, and do this in a way which is compatible with the rest of the data. Let $\mathcal{S}_j = \{\tilde{\mathbf{R}}_j \in \mathcal{P}_n : \tilde{R}_{ij} =$

$\mathbf{X}_j^{-1}(A_i)$ if $A_i \in \mathcal{A}_j\}$, $j = 1, \ldots, N$, be the set of possible augmented random vectors, that is the original partially ranked items together with the allowable "fill-ins" of the missing ranks. Our goal is to sample from the posterior distribution

$$P(\alpha, \boldsymbol{\rho}|\mathbf{R}_1, \ldots, \mathbf{R}_N) = \sum_{\tilde{\mathbf{R}}_1 \in \mathcal{S}_1} \cdots \sum_{\tilde{\mathbf{R}}_N \in \mathcal{S}_N} P\left(\alpha, \boldsymbol{\rho}, \tilde{\mathbf{R}}_1, \ldots, \tilde{\mathbf{R}}_N|\mathbf{R}_1, \ldots, \mathbf{R}_N\right).$$

Our MCMC algorithm alternates between sampling the augmented ranks given the current values of α and $\boldsymbol{\rho}$, and sampling α and $\boldsymbol{\rho}$ given the current values of the augmented ranks. For the latter, we sample from the posterior $P(\alpha, \boldsymbol{\rho}|\tilde{\mathbf{R}}_1, \ldots, \tilde{\mathbf{R}}_N)$ as in Section 2.4. For the former, fixing α and $\boldsymbol{\rho}$ and the observed ranks $\mathbf{R}_1, \ldots, \mathbf{R}_N$, we see that $\tilde{\mathbf{R}}_1, \ldots, \tilde{\mathbf{R}}_N$ are conditionally independent, and moreover, that each $\tilde{\mathbf{R}}_j$ only depends on the corresponding $\mathbf{R}_j$. This enables us to consider the sampling of new augmented vectors $\tilde{\mathbf{R}}'_j$ separately for each $j, j = 1, \ldots, N$. Specifically, given the current $\tilde{\mathbf{R}}_j$ (which embeds information contained in $\mathbf{R}_j$) and the current values for α and $\boldsymbol{\rho}$, $\tilde{\mathbf{R}}'_j$ is sampled in $\mathcal{S}_j$ from a uniform proposal distribution, meaning that the highest ranks from 1 to n_j have been reserved for the items in $\mathcal{A}_j$, while compatible ranks are randomly drawn for items in $\mathcal{A}_j^c$. The proposed $\tilde{\mathbf{R}}'_j$ is then accepted with probability

$$\min\left\{1, \exp\left[-\frac{\alpha}{n}\left(d(\tilde{\mathbf{R}}'_j, \rho) - d(\tilde{\mathbf{R}}_j, \rho)\right)\right]\right\}. \tag{21}$$

The MCMC algorithm described above and used in the case of partial rankings is given in Algorithm 3 of Appendix B. Our algorithm can also handle situations of generic partial ranking, where each assessor is asked to provide the mutual ranking of some subset $\mathcal{A}_j \subset \{A_1, ..., A_n\}$ consisting of $n_j \leq n$ items, not necessarily the top-n_j. In this case, we can only say that in $\tilde{\mathbf{R}}_j = (\tilde{R}_{1j}, ..., \tilde{R}_{nj})$ the order between items $A_i \in \mathcal{A}_j$ must be preserved as in $\mathbf{R}_j$, whereas the ranks of the augmented "fill-ins" $A_i \in \mathcal{A}_j^c$ are left open. More exactly, the latent rank vector $\tilde{\mathbf{R}}_j$ takes values in the set $\mathcal{S}_j = \{\tilde{\mathbf{R}}_j \in \mathcal{P}_n : \text{if } R_{i_1 j} < R_{i_2 j}, \text{ with } A_{i_1}, A_{i_2} \in \mathcal{A}_j \Rightarrow \tilde{R}_{i_1 j} < \tilde{R}_{i_2 j}\}$. The MCMC is then easily adjusted so that the sampling of each $\tilde{\mathbf{R}}_j$ is restricted to the corresponding $\mathcal{S}_j$, thus respecting the mutual rank orderings in the data.

4.1.1 Effects of Unranked Items on the top-k Consensus Ranking

In applications in which the number of items is large there are often items which none of the assessors included in their top-list. What is the exact role of such "left-over" items in the top-k consensus ranking of all items? Can we ignore such "left-over" items and consider only the items explicitly ranked by at least one assessor? In the following we first show that only items explicitly ranked by the assessors appear in top positions of the consensus ranking. We then show that, when considering the MAP consensus ranking, excluding the left-over items from the ranking procedure already at the start has no effect on how the remaining ones will appear in such consensus ranking.

For a precise statement of these results, we need some new notation. Suppose that assessor j has ranked a subset $\mathcal{A}_j$ of n_j items. Let $\mathcal{A} = \bigcup_{j=1,\ldots,N} \mathcal{A}_j$, and denote $n = |\mathcal{A}|$. Let n^* be the total number of items, including left-over items which have not been explicitly ranked by any assessor. Denote by $\mathcal{A}^* = \{A_i; i = 1, \ldots, n^*\}$ the collection of all items, and by $\mathcal{A}^c = \mathcal{A}^* \setminus \mathcal{A}$ the left-over items. Each rank vector $\mathbf{R}_j$ for assessor j contains, in some

order, the ranks from 1 to n_j given to items in $\mathcal{A}_j$. In the original data the ranks of all remaining items are left unspecified, apart from the fact that implicitly, for assessor j, they would have values which are at least as large as $n_j + 1$.

The results below are formulated in terms of the two different modes of analysis, which we need to compare and which correspond to different numbers of items being included. The first alternative is to include in the analysis the complete set $\mathcal{A}^*$ of n^* items, and to complement each data vector $\mathbf{R}_j$ by assigning (originally missing) ranks to all items which are not included in $\mathcal{A}_j$; their ranks will then form some permutation of the sequence $(n_j + 1, \ldots, n^*)$. We call this mode of analysis *full analysis*, and denote the corresponding probability measure by P_{n^*}. The second alternative is to include in the analysis only the items which have been explicitly ranked by at least one assessor, that is, items belonging to the set $\mathcal{A}$. We call this second mode *restricted analysis*, and denote the corresponding probability measure by P_n. The probability measure P_n is specified as before, including the uniform prior on the consensus ranking $\boldsymbol{\rho}$ across all $n!$ permutations of $(1, 2, \ldots, n)$, and the uniform prior of the unspecified ranks R_{ij} of items $A_i \in \mathcal{A}_j^c$ across the permutations of $(n_j + 1, \ldots, n)$. The definition of P_{n^*} is similar, except that then the uniform prior distributions are assumed to hold in the complete set $\mathcal{A}^*$ of items, that is, over permutations of $(1, 2, \ldots, n^*)$ and $(n_j + 1, \ldots, n^*)$, respectively. In the posterior inference carried out in both modes of analysis, the augmented ranks, which were not recorded in the original data, are treated as random variables, with values being updated as part of the MCMC sampling.

Proposition 4 *Consider two latent consensus rank vectors $\boldsymbol{\rho}$ and $\boldsymbol{\rho}'$ such that*

(i) in the ranking $\boldsymbol{\rho}$ all items in $\mathcal{A}$ have been included among the top-n-ranked, while those in $\mathcal{A}^c$ have been assigned ranks between $n + 1$ and n^*,

(ii) $\boldsymbol{\rho}'$ is obtained from $\boldsymbol{\rho}$ by a permutation, where the rank in $\boldsymbol{\rho}$ of at least one item belonging to $\mathcal{A}$ has been transposed with the rank of an item in $\mathcal{A}^c$.

Then, $P_{n^*}(\boldsymbol{\rho}|data) \geq P_{n^*}(\boldsymbol{\rho}'|data)$, for the footrule, Kendall and Spearman distances in the full analysis *mode.*

Remark. The above proposition says, in essence, that any consensus lists of top-n ranked items, which contains one or more items with their ranks completely missing in the data (that is, the item was not explicitly ranked by any of the assessors), can be improved *locally*, in the sense of increasing the associated posterior probability with respect to P_{n^*}. This happens by trading such an item in the top-n list against another, which had been ranked but which had not yet been selected to the list. In particular, the MAP estimate(s) for consensus ranking assign n highest ranks to explicitly ranked items in the data (which corresponds to the result in Meilă and Bao (2010) for Kendall distance). The following statement is an immediate implication of Proposition 4, following from a marginalization with respect to P_{n^*}.

Corollary 1 *Consider, for $k \leq n$, collections $\{A_{i_1}, A_{i_2}, \ldots, A_{i_k}\}$ of k items and the corresponding ranks $\{\rho_{i_1}, \rho_{i_2}, \ldots, \rho_{i_k}\}$. In* full analysis *mode, the maximal posterior probability $P_{n^*}(\{\rho_{i_1}, \rho_{i_2}, \ldots, \ \rho_{i_k}\} = \{1, 2, \ldots, k\}|data)$, is attained when $\{A_{i_1}, A_{i_2}, \ldots, A_{i_k}\} \subset \mathcal{A}$.*

Another consequence of Proposition 4 is the coincidence of the MAP estimates under the two probability measures P_n and P_{n^*}.

Corollary 2 *Denote by $\boldsymbol{\rho}^{MAP*}$ the MAP estimate for consensus ranking obtained in a* full analysis, $\boldsymbol{\rho}^{MAP*} := \text{argmax}_{\boldsymbol{\rho}\in\mathcal{P}_{n^*}} P_{n^*}(\boldsymbol{\rho}|data)$, *and by $\boldsymbol{\rho}^{MAP}$ the MAP estimate for consensus ranking obtained in a* restricted analysis, $\boldsymbol{\rho}^{MAP} := \text{argmax}_{\boldsymbol{\rho}\in\mathcal{P}_n} P_n(\boldsymbol{\rho}|data)$. *Then,* $\boldsymbol{\rho}^{MAP*}|_{i:A_i\in\mathcal{A}} \equiv \boldsymbol{\rho}^{MAP}$.

Remark. The above result is very useful in the context of applications, since it guarantees that the top-n items in the MAP consensus ranking do not depend on which version of the analysis is performed. Recall that a full analysis cannot always be carried out in practice, due to the fact that left-over items might be unknown, or their number might be too large for any realistic computation.

4.2 Pairwise Comparisons

In many situations, assessors compare pairs of items rather than ranking all or a subset of items. We extend our Bayesian data augmentation scheme to handle such data. Our approach is an alternative to Lu and Boutilier (2014), who treated preferences by applying their Repeated Insertions Model (RIM). Our approach is simpler, it is fully integrated into our Bayesian inferential framework, and it works for any right-invariant distance.

As an example of paired comparisons, assume assessor j stated the preferences $\mathcal{B}_j = \{A_1 \prec A_2, A_2 \prec A_5, A_4 \prec A_5\}$. Here $A_r \prec A_s$ means that A_s is preferred to A_r, so that A_s has a lower rank than A_r. Let $\mathcal{A}_j$ be the set of items constrained by assessor j, in this case $\mathcal{A}_j = \{A_1, A_2, A_4, A_5\}$. Differently from Section 4.1, the items which have been considered by each assessor are now not necessarily fixed to a given rank. Hence, in the MCMC algorithm, we need to propose augmented ranks which obey the partial ordering constraints given by each assessor, to avoid a large number of rejections, with the difficulty that none of the items is now fixed to a given rank. Note that we can also handle the case when assessors give ties as a result of some pairwise comparisons: in such a situation, each pair of items resulting in a tie is randomized to a preference at each data augmentation step inside the MCMC, thus correctly representing the uncertainty of the preference between the two items. None of the experiments included in the paper involves ties, thus this randomization is not needed.

We assume that the pairwise orderings in $\mathcal{B}_j$ are mutually compatible, and define by $\text{tc}(\mathcal{B}_j)$ the transitive closure of $\mathcal{B}_j$, containing all pairwise orderings of the elements in $\mathcal{A}_j$ induced by $\mathcal{B}_j$. In the example, $\text{tc}(\mathcal{B}_j) = \mathcal{B}_j \cup \{A_1 \prec A_5\}$. For the case of ordered subsets of items, the transitive closure is simply the single set of pairwise preferences compatible with the ordering, for example, $\{A_1 \prec A_2 \prec A_5\}$ yields $\text{tc}(\mathcal{B}_j) = \{A_1 \prec A_2, A_2 \prec A_5, A_1 \prec A_5\}$. The R packages `sets` (Meyer and Hornik, 2009) and `relations` (Meyer and Hornik, 2014) efficiently compute the transitive closure.

The main idea of our method for handling such data remains the same as in Section 4.1, and the algorithm is the same as Algorithm 3. However, here a "modified" leap-and-shift proposal distribution, rather than a uniform one, is used to sample augmented ranks which are compatible with the partial ordering constraint. Suppose that, from the latest step of the MCMC, we have a full augmented rank vector $\tilde{\mathbf{R}}_j$ for assessor j, which is

compatible with $\text{tc}(\mathcal{B}_j)$. Draw a random number u uniformly from $\{1,\dots,n\}$. If $A_u \in \mathcal{A}_j$, let $l_j = \max\{\tilde{R}_{kj} : A_k \in \mathcal{A}_j, k \neq u, (A_k \succ A_u) \in \text{tc}(\mathcal{B}_j)\}$, with the convention that $l_j = 0$ if the set is empty, and $r_j = \min\{\tilde{R}_{kj} : A_k \in \mathcal{A}_j, k \neq u, (A_k \prec A_u) \in \text{tc}(\mathcal{B}_j)\}$, with the convention that $r_j = n+1$ if the set is empty. Now complete the leap step by drawing a new proposal $\tilde{R}'_{uj}$ uniformly from the set $\{l_j+1,\dots,r_j-1\}$. Otherwise, if $A_u \in \mathcal{A}_j^c$, we complete the leap step by drawing $\tilde{R}'_{uj}$ uniformly from $\{1,\dots,n\}$. The shift step remains unchanged. Note that this modified leap-and-shift is symmetric.

4.3 Clustering Assessors Based on their Rankings of All Items

So far we have assumed that there exists a unique consensus ranking shared by all assessors. In many cases the assumption of homogeneity is unrealistic: the possibility of dividing assessors into more homogeneous subsets, each sharing a consensus ranking of the items, brings the model closer to reality. We then introduce a mixture of Mallows models, able to handle heterogeneity. We here assume that the data consist of complete rankings.

Let $z_1,\dots,z_N \in \{1,\dots,C\}$ assign each assessor to one of C clusters. The assessments within each cluster $c \in \{1,\dots,C\}$ are described by a Mallows model with parameters α_c and $\boldsymbol{\rho}_c$, the cluster consensus. Assuming conditional independence across the clusters, the augmented data formulation of the likelihood for the observed rankings $\mathbf{R}_1,\dots,\mathbf{R}_N$ is given by

$$P\left(\mathbf{R}_1,\dots,\mathbf{R}_N|\,\{\boldsymbol{\rho}_c,\alpha_c\}_{c=1,\dots,C}, z_1,\dots,z_N\right) = \prod_{j=1}^{N} \frac{1_{\mathcal{P}_n}(\mathbf{R}_j)}{Z_n(\alpha_{z_j})} \exp\left\{-\frac{\alpha_{z_j}}{n} d(\mathbf{R}_j,\boldsymbol{\rho}_{z_j})\right\}.$$

For the scale parameters, we assume the prior $\pi(\alpha_1,\dots,\alpha_C) \propto \prod_{c=1}^{C}\pi(\alpha_c)$, where $\pi(\alpha_c) = \lambda\exp(-\lambda\alpha_c)1_{[0,\alpha_{\max}]}(\alpha_c)/(1-e^{-\lambda\alpha_{\max}})$. We further assume that the cluster labels are a priori distributed according to $P(z_1,\dots,z_N|\tau_1,\dots,\tau_C) = \prod_{j=1}^{N}\tau_{z_j}$, where τ_c is the probability that an assessor belongs to the c-th subpopulation; $\tau_c \geq 0$, $c = 1,\dots,C$ and $\sum_{c=1}^{C}\tau_c = 1$. Finally $\tau_1,\dots,\tau_C$ are assigned the standard symmetric Dirichlet prior $\pi(\tau_1,\dots,\tau_C) = \Gamma(\psi C)\Gamma(\psi)^{-C}\prod_{c=1}^{C}\tau_c^{\psi-1}$, using the gamma function $\Gamma(\cdot)$.

The number of clusters C is often not known, and the selection of C can be based on different criteria. Here we inspect the posterior distribution of the within-cluster sum of distances of the observed ranks from the corresponding cluster consensus (see Section 6.3 for more details). This approach is a Bayesian version of the more classical within-cluster sum-of-squares criterion for model selection, and we expect to observe an elbow in the within-cluster distance posterior distribution as a function of C, identifying the optimal number of clusters.

Label switching is not explicitly handled inside our MCMC, to ensure full convergence of the chain (Jasra et al., 2005; Celeux et al., 2000). MCMC iterations are re-ordered after convergence is achieved, as in Papastamoulis (2015). The MCMC algorithm alternates between sampling $\boldsymbol{\rho}_1,\dots,\boldsymbol{\rho}_C$ and $\alpha_1,\dots,\alpha_C$ in a Metropolis-Hastings step, and $\tau_1,\dots,\tau_C$ and $z_1,\dots,z_N$ in a Gibbs sampler step. The former is straightforward, since $(\boldsymbol{\rho}_c,\alpha_c)_{c=1,\dots,C}$ are conditionally independent given $z_1,\dots,z_N$. In the latter, we exploit the fact that the Dirichlet prior for $\tau_1,\dots,\tau_C$ is conjugate to the multinomial conditional prior for $z_1,\dots,z_N$ given $\tau_1,\dots,\tau_C$. Therefore in the Gibbs step for $\tau_1,\dots,\tau_C$, we sample from $\mathcal{D}(\psi+n_1,\dots,\psi+$

$n_C)$, where $\mathcal{D}(\cdot)$ denotes the Dirichlet distribution and $n_c = \sum_{j=1}^{N} 1_c(z_j)$, $c = 1, \ldots, C$. Finally, in the Gibbs step for z_j, $j = 1, \ldots, N$, we sample from $P(z_j = c|\tau_c, \boldsymbol{\rho}_c, \alpha_c, R_j) \propto \tau_c P(\mathbf{R}_j|\boldsymbol{\rho}_c, \alpha_c) = \tau_c Z_n(\alpha_c)^{-1} \exp\{-(\alpha_c/n) d(\mathbf{R}_j, \boldsymbol{\rho}_c)\}$. The pseudo-code of the clustering algorithm is sketched in Algorithm 2 of Appendix B.

It is not difficult to treat situations where data are incomplete (in any way described before) and the assessors must be divided into separate clusters. Algorithms 2 and 3 are merged in an obvious way, by iterating between augmentation, clustering, and α and $\boldsymbol{\rho}$ updates. The MCMC algorithm for clustering based on partial rankings or pairwise preferences is sketched in Algorithm 4 of Appendix B.

4.4 Example: Preference Prediction

Consider a situation in which the assessors have expressed their preferences on a collection of items, by performing only partial rankings. Or, suppose that they have been asked to respond to some queries containing different sets of pairwise comparisons. One may then ask how the assessors would have ranked some subset of items of interest when such ranking could not be concluded directly from the data they provided. Sometimes the interest is to predict the assessors' top preferences, accounting for the possibility that such top lists could contain items which some assessors had not seen. Problems of this type are commonly referred to as *personalized ranking*, or *preference learning* (Fürnkranz and Hüllermeier, 2010), being a step towards *personalized recommendation*. There is a large and rapidly expanding literature describing a diversity of methods in this area.

Our framework, based on the Bayesian Mallows model, and its estimation algorithms as described in the previous Sections, form a principled approach for handling such problems. Assuming a certain degree of similarity in the individual preferences, and with different assessors providing partly complementary information, it is natural to try to borrow strength from such partial preference information from different assessors for forming a consensus. Expanding the model to include clusters allows handling heterogeneity that may be present in the assessment data (Francis et al., 2010). The Bayesian estimation procedure provides then the joint posterior distribution, expressed numerically in terms of the MCMC output consisting of sampled values of all cluster membership indicators, z_j, and of complete individual rankings, $\tilde{\mathbf{R}}_j$. For example, if assessor j did not compare A_1 to A_2, we might be interested in computing $P(A_1 \prec_j A_2|\text{data})$, the predictive probability that this assessor would have preferred item A_2 to item A_1. This probability is then readily obtained from the MCMC output, as a marginal of the posterior $P(\tilde{\mathbf{R}}_j|\text{data})$.

To illustrate how this is possible with our approach, we present a small simulated experiment, corresponding to a heterogeneous collection of assessors expressing some of their pairwise preferences, and then want to predict the full individual ranking $\tilde{\mathbf{R}}_j$ of all items, for all j. For this, we generated pairwise preference data from a mixture of Mallows models with footrule distance, using the procedure explained in Appendix C. We generated the data with $N = 200$, $n = 15$, $C = 3$, $\alpha_1, ..., \alpha_C = 4$, $\psi_1, ..., \psi_C = 50$, obtaining the true $\tilde{\mathbf{R}}_{j,\text{true}}$ for every assessor. Then, we assigned to each assessor j a different number, $T_j \sim \text{TruncPoiss}(\lambda_T, T_{\max})$, of pair comparisons, sampled from a truncated Poisson distribution with $\lambda_T = 20$, denoting by $T_{\max} = n(n-1)/2$ the total number of possible pairs

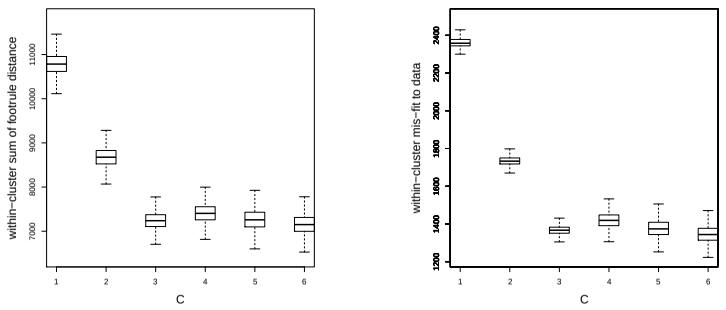


Figure 6: Results of the simulation in Section 4.4. Boxplots of the posterior distribution of the within-cluster sum of footrule distances (left), and of the within-cluster indicator of mis-fit to the data (right), for different choices of C.

from n items. Each pair comparison was then ordered according to the true $\tilde{\mathbf{R}}_{j,\text{true}}$. The average number of pairs per assessor was around 20, less than 20% of $T_{\max}$.

In the analysis, we run Algorithm 4 of Appendix B on these data, using the exact partition function, for 10^5 iterations (of which 10^4 were for burn-in). Separate analyses were performed for $C \in \{1, \ldots, 6\}$. Then, in order to inspect if our method correctly identified the true number of clusters we computed two quantities: the within-cluster sum of footrule distances, given by $\sum_{c=1}^{C} \sum_{j:z_j=c} d(\tilde{\mathbf{R}}_j, \boldsymbol{\rho}_c)$, and a within-cluster indicator of mis-fit to the data, $\sum_{c=1}^{C} \sum_{j:z_j=c} |\{B \in \text{tc}(\mathcal{B}_j) : B \text{ is not consistent with } \boldsymbol{\rho}_c\}|$, where a pair comparison $B \in \text{tc}(\mathcal{B}_j), B = (A_r \prec A_s)$ is not consistent with $\boldsymbol{\rho}_c$ if $\rho_{c,s} > \rho_{c,r}$. The number of such non-consistent pairs in $\mathcal{B}_j$ gives an indication of the mis-fit of the j-th assessor to its cluster. Notice that, while the latter measure takes into account the data directly, the former is based on the augmented ranks $\tilde{\mathbf{R}}_j$ only. Hence, the within-cluster sum of footrule distances could be more sensitive to possible misspecifications in $\tilde{\mathbf{R}}_j$ when the data are very sparse. Notice also that the second measure is a 'modified' version of the Kendall distance between the data and the cluster centers. The boxplots of the posterior distributions of these two quantities are shown in Figure 6: the two measures are very consistent in indicating a clear elbow at $C = 3$, thus correctly identifying the value we used to generate the data.

We then studied the success rates of correctly predicting missing individual pairwise preferences. A pairwise preference between items A_{i_1} and A_{i_2} was considered missing for assessor j if it was not among the sampled pairwise comparisons included in the data as either $A_{i_1} \prec_{j,\text{true}} A_{i_2}$ or $A_{i_2} \prec_{j,\text{true}} A_{i_1}$, nor could such ordering be concluded from the data indirectly by transitivity. Thus we computed, for all assessors j, the predictive probabilities $P(A_{i_1} \prec_j A_{i_2}|\text{data})$ for all pairs of items $\{A_{i_1}, A_{i_2}\}$ not ordered in $\text{tc}(\mathcal{B}_j)$. The rule for practical prediction was to always bet on the ordering with the larger predictive probability of these two probabilities, then at least 0.5. Each resulting predictive probability is a direct

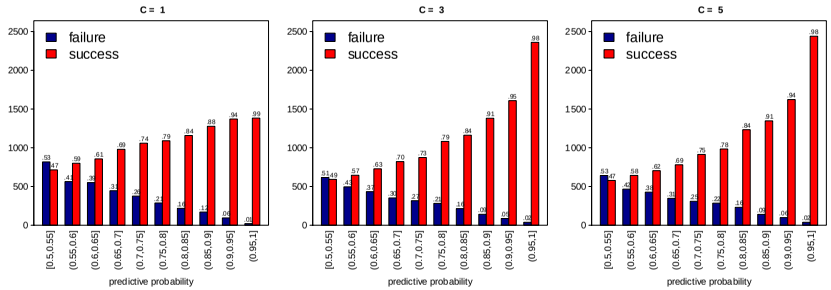


Figure 7: Results of the simulation in Section 4.4. Barplots of the frequency of successes (red columns) and failures (blue columns) obtained fixing $C = 1$ (left), 3 (middle), and 5 (right), for the data generated with $\lambda_T = 20$. For $C = 1$, 75% of all predictions were correct, for $C = 3$, 79.1%, and for $C = 5$, 79%.

quantification of the uncertainty in making the bet: a value close to 0.5 expresses a high degree of uncertainty, while a value close to 1 would signal greater confidence in that the bet would turn out right. In the experiment, these bets were finally compared to the orderings of the same pairs in the simulated true rankings $\tilde{\mathbf{R}}_{j,\text{true}}$. If they matched, this was registered as a success, and if not, then as a failure.

In Figure 7 are shown the barplots of the results from this experiment, expressed in terms of the frequency of successes (red columns) and failures (blue columns), obtained by combining the outcomes from all individual assessors. For this presentation, the predictive probabilities used for betting were grouped into the respective intervals $[0.50, 0.55], (0.55, 0.60], \ldots, (0.95, 1.00]$ on the horizontal axis, so that pair preferences become more difficult to predict the more one moves to the left, along the x-axis. On top of each column the percentage of successes, or failures, of the corresponding bets is shown. For the results considered on the left, the predictions were made without assuming a cluster structure ($C = 1$) in the analysis, in the middle graph the same number ($C = 3$) of clusters was assumed in the analysis as in the data generation, and on the right, we wanted to study whether assuming an even larger number ($C = 5$) of clusters in the analysis might influence the performance of our method for predicting missing preferences.

Two important conclusions can be made from the results of this experiment. First, from comparing the three graphs, we can see that not assuming a cluster structure ($C = 1$) in the data analysis led to an overall increased proportion of uncertain bets, in the sense of being based on predictive probabilities closer to the 0.5 end of the horizontal axis, than if either $C = 3$ or $C = 5$ was assumed. On the other hand, there is almost no difference between the graphs corresponding to $C = 3$ and $C = 5$. Thus, moderate overfitting of clusters neither improved nor deteriorated the quality of the predictions (this seems consistent with the very similar within-cluster distances in these two cases, shown in Figure 6). A second, and more interesting, observation is that, in all three cases considered, the predictive probabilities

used for betting turned out to be empirically very well calibrated (see, for example, Dawid (1982) and Little (2011)). For example, of the bets based on predictive probabilities in the interval $(0.70, 0.75]$, 74% were successful for $C = 1$, 73% when $C = 3$, and 75% when $C = 5$. By inspection, such correspondence can be seen to hold quite well on all intervals in all three graphs. That the same degree of empirical calibration holds also when an incorrect number of clusters was fitted to the data as with the correct one, signals a certain amount of robustness of this aspect towards variations in the modeling.

We repeated the same experiment with less data, namely using $\lambda_T = 10$. This gives an average number of pairs per assessor around 10% of $T_{\max}$. Results are displayed in Figure 5 of the Supplementary Material, Section 3. Predictive probabilities are still very well calibrated, but of course the quality of prediction is worse. Nonetheless, for $C = 3$, 76.8% of all predictions were correct.

5. Related Work

We briefly review the literature which uses the Mallows model, or is based on other probabilistic approaches, as these are most closely related to our method.

The Mallows model was studied almost exhaustively in the case of Kendall distance, of which the partition function is easy to compute. Among probabilistic approaches, one of the most interesting is Meilă and Chen (2010), who proposed a Dirichlet process mixture of the Generalized Mallows model of Fligner and Verducci (1986) over incomplete rankings. In this paper two Gibbs sampling techniques for estimating the posterior density were studied. This framework was further extended in Meilă and Bao (2010), who developed an algorithm for the ML estimation of their Generalized Mallows model for infinite rankings (IGM), based on Kendall distance. They also considered Bayesian inference with the conjugate prior, showing that such inference is much harder.

In terms of focus and aim, the proposal in Lu and Boutilier (2014) is very close to our approach: they develop a method to form clusters of assessors and perform preference learning and prediction from pairwise comparison data in the Mallows model framework. Their approach is connected to our extension to preference data (Section 4.2), but differs most notably in the general model and algorithm. Their generalized repeated insertion model (GRIM), based on Kendall distance only, generalizes the repeated insertion method for unconditional sampling of Mallows models of Doignon et al. (2004). Lu and Boutilier (2014) perform ML estimation of the consensus ranking using a method based on the EM algorithm, thus not providing uncertainty quantification for their estimates. Our target, on the other hand, is the full posterior distribution of the unknown consensus ranking. The fact that, for the uniform prior, the MAP estimates and the ML estimates coincide, establishes a natural link between these inferential targets. Two of our illustrations, in Sections 6.3 and 6.4, use the same data as in Lu and Boutilier (2014).

In the frequentist framework, the Mallows model with other distances than Kendall was studied by Irurozki et al. (2014) and Irurozki et al. (2016b), who also developed the `PerMallows` R package (Irurozki et al., 2016a). Moreover, mixtures of Mallows models have been used to analyze heterogeneous rank data by several authors. Murphy and Martin (2003) studied mixtures of Mallows with Kendall, footrule and Cayley distances, applying their method to the benchmark American Psychological Association (Diaconis, 1988)

election data set, where only $n = 5$ candidates (items) are ranked. The difficulties in the computation of the partition function for the footrule distance, which arise for larger values of n, were not discussed. Gormley and Murphy (2006) use mixtures of Plackett-Luce models in a maximum likelihood framework for clustering. Lee and Yu (2012) use mixtures of weighted distance-based models to cluster ranking data. Also Busse et al. (2007) proposed a mixture approach for clustering rank data, but focusing on the Kendall distance only.

Other probabilistic approaches, less related to the Mallows model, include the Insertion Sorting Rank (ISR) model of Jacques and Biernacki (2014). It is implemented in the R package `rankcluster` (Jacques et al., 2014), and allows clustering of partial rankings. Sun et al. (2012) developed a non-parametric probabilistic model on preferences, which can handle also heterogeneous assessors. This work extends the non-parametric kernel density estimation approach over rankings introduced by Lebanon and Mao (2008), enabling it then to handle ranking data of arbitrary incompleteness and tie structure. However, the approach is based on a random-censoring assumption, which could be easily violated in practice.

Among machine learning approaches, those pertaining to the area of learning to rank, or rank aggregation, are also related to ours. Their aim is to find the best consensus ranking by optimizing some objective function (for example Kemeny or Borda rankings), but they generally do not provide uncertainty quantifications of the derived point estimates. A simple comparison of our approach to two such methods is shown below, in Section 5.1.

5.1 Comparisons with Other Methods

The procedure we propose is Bayesian, and one of its strengths is its ability to quantify the uncertainty related to the parameter estimates and predictions. In order to compare our results with the ones obtained by other methods which provide only point estimates, we need to summarize the posterior density of the model parameters into a single point estimate, for example MAP, mode, mean, cumulative probability consensus. The cumulative probability (CP) consensus ranking is the ranking arising from the following sequential scheme: first select the item which has the maximum a posteriori marginal probability of being ranked 1st; then the item which has the maximum a posteriori marginal posterior probability of being ranked 1st or 2nd among the remaining ones, etc. The CP consensus can be seen as a sequential MAP. We generated the data from the Mallows model (for details refer to Appendix C) with Kendall distance, since this is the unique distance handled by existing competitors based on the Mallows model. We compare our procedure (here denoted by `BayesMallows`) with the following methods:

- `PerMallows` (Irurozki et al., 2016a): MLE of the Mallows and the Generalized Mallows models, with some right-invariant distance functions, but not footrule nor Spearman.
- `rankcluster` (Jacques et al., 2014): Inference for the Insertion Sorting Rank (ISR) model.
- `RankAggreg` (Pihur et al., 2009): Rank aggregation via several different algorithms. Here we use the Cross-Entropy Monte Carlo algorithm.
- Borda count (de Borda, 1781): Easy and classic way to aggregate ranks. Basically equivalent to the average rank method, thus not a probabilistic approach.

α_T	method	$\hat{\alpha}$ or $\hat{\pi}$	$\frac{1}{n}d(\hat{\boldsymbol{\rho}}, \boldsymbol{\rho}_T)$	$T(\hat{\boldsymbol{\rho}}, \mathbf{R})$
1	`BayesMallows` - CP	1.01 (0.22)	0.53 (0.26)	19.07 (0.54)
	`BayesMallows` - MAP		0.57 (0.31)	19.07 (0.56)
	`PerMallows`	1.10 (0.19)	0.54 (0.26)	19.12 (0.56)
	`rankcluster`	0.60 (0.02)	0.86 (0.34)	19.4 (0.58)
	`RankAggreg`	n.a.	0.66 (0.27)	19.25 (0.58)
	Borda	n.a.	0.54 (0.27)	19.12 (0.56)
2	`BayesMallows` - CP	2.05 (0.18)	0.17 (0.12)	16.29 (0.47)
	`BayesMallows` - MAP		0.18 (0.13)	16.28 (0.47)
	`PerMallows`	2.07 (0.17)	0.23 (0.13)	16.33 (0.46)
	`rankcluster`	0.66 (0.02)	0.37 (0.22)	16.52 (0.54)
	`RankAggreg`	n.a.	0.29 (0.14)	16.41 (0.49)
	Borda	n.a.	0.23 (0.14)	16.33 (0.46)
3	`BayesMallows` - CP	3.02 (0.07)	0.06 (0.08)	13.88 (0.5)
	`BayesMallows` - MAP		0.07 (0.09)	13.87 (0.5)
	`PerMallows`	3.02 (0.21)	0.09 (0.08)	13.9 (0.51)
	`rankcluster`	0.72 (0.01)	0.15 (0.11)	13.96 (0.49)
	`RankAggreg`	n.a.	0.14 (0.11)	13.94 (0.52)
	Borda	n.a.	0.09 (0.08)	13.91 (0.51)
4	`BayesMallows` - CP	3.96 (0.20)	0.02 (0.05)	11.83 (0.41)
	`BayesMallows` - MAP		0.02 (0.04)	11.83 (0.41)
	`PerMallows`	3.95 (0.20)	0.03 (0.05)	11.85 (0.4)
	`rankcluster`	0.76 (0.01)	0.08 (0.08)	11.9 (0.44)
	`RankAggreg`	n.a.	0.06 (0.05)	11.87 (0.42)
	Borda	n.a.	0.03 (0.05)	11.85 (0.4)

Table 2: Results of the simulations of Section 5.1. $\hat{\alpha}$ refers to the posterior mean (row: `BayesMallows`) or to MLE (row: `PerMallows`). $\hat{\pi}$ is the dispersion parameter of ISR. $\hat{\boldsymbol{\rho}}$ is the consensus ranking estimated by the different procedures: MAP (row: `BayesMallows` (MAP)), CP (row: `BayesMallows` (CP)), MLE (row: `PerMallows` and `rankcluster`), point estimate (row: `RankAggreg` and Borda). Standard deviations are reported in parenthesis. Parameters setting: $N = 100$, $n = 10$.

The results of the comparisons are shown in Table 2. The `BayesMallows` estimates are obtained through Algorithm 1 of Appendix B, with the available exact partition function corresponding to Kendall distance, and for 10^5 iterations (after a burn-in of 10^4 iterations). All quantities shown are averages over 50 independent repetitions of the whole simulation experiment. $\hat{\alpha}$ is the posterior mean (for `BayesMallows`) or the MLE (for `PerMallows`), while $\hat{\pi}$ is the MLE estimate of the dispersion parameter of ISR (for `rankcluster`). $\hat{\boldsymbol{\rho}}$ is the consensus ranking estimated by the different procedures: for `BayesMallows` it is either given by the CP consensus (`BayesMallows` - CP), or by the MAP (`BayesMallows` - MAP). We compare the goodness of fit of the methods by evaluating two quantities: first, the normalized Kendall distance between the estimated consensus ranking and the true one, used to generate the data, $d(\hat{\boldsymbol{\rho}}, \boldsymbol{\rho}_T)/n$. Second, the average of Kendall distances between the data points and the estimated consensus ranking, $T(\hat{\boldsymbol{\rho}},\ \mathbf{R}) = \frac{1}{N}\sum_{j=1}^{N} d(\hat{\boldsymbol{\rho}}, \mathbf{R}_j)$. This quantity makes sense here, being independent on the likelihood assumed by the different models.

The first remark about the results in Table 2 is the clear improvement of the performance in terms of $\frac{1}{n}d(\hat{\boldsymbol{\rho}}, \boldsymbol{\rho}_{\mathrm{T}})$, of all the methods, for increasing α. This obvious result is a consequence of the easier task of rank aggregation when the assessors are more concentrated around the consensus. Because the data were generated with the same model which `BayesMallows` and `PerMallows` used for inference, we expected that the Mallows-based methods would perform better than the rank aggregation methods we considered. The results of Table 2 confirm this claim: `BayesMallows` and `PerMallows` outperform the other rank aggregation methods, with the exception of Borda count, which gives the same results as `PerMallows`. This is not surprising, since the `PerMallows` MLE of the consensus is approximated though the Borda algorithm. Moreover, when the summary of the Bayesian posterior is the CP consensus, the performance of `BayesMallows`, both in terms of $\frac{1}{n}d(\hat{\boldsymbol{\rho}}, \boldsymbol{\rho}_{\mathrm{T}})$ and $T(\hat{\boldsymbol{\rho}}, \mathbf{R})$, was better than the others. This is another advantage of our approach on the competitors: being the output a full posterior distribution of the consensus, we can select any strategy to summarize it, possibly driven by the application at hand. To conclude, our approach gives slightly better results than the other existing methods, and in the worst cases the performance is still equivalent. In Section 6 we will compare inferential results on real data, not necessarily generated from the Mallows model.

6. Experiments

The experiments considered in this section illustrate the use of our approach in various situations corresponding to different data structures.

6.1 Meta-Analysis of Differential Gene Expression

Studies of differential gene expression between two conditions produce lists of genes, ranked according to their level of differential expression as measured by, for example, p-values. There is often little overlap between gene lists found by independent studies comparing the same condition. This situation raises the question of whether a consensus top list over all available studies can be found.

We handle this situation in our Bayesian Mallows model by considering each study $j \in \{1, \ldots, N\}$ to be an assessor, providing a top-n_j list of differentially expressed genes, which are the ranked items. This problem was studied by DeConde et al. (2006), Deng et al. (2014), and Lin and Ding (2009), who all used the same 5 studies comparing prostate cancer patients with healthy controls (Dhanasekaran et al., 2001; Luo et al., 2001; Singh et al., 2002; True et al., 2006; Welsh et al., 2001). We consider the same 5 studies, and we aim at estimating a consensus with uncertainty. Data consist of the top-25 lists of genes from each study, in total 89 genes. Here we perform a restricted analysis (see 4.1.1), and in this case $n_j = 25$ for all $j = 1, \ldots, 5$, and $n = 89$.

Table 3 shows the result of analyzing the five gene lists with the Mallows footrule model for partial data (Section 4.1). We run 20 different chains, for a total of 10^7 iterations (computing time was $16'4''$), and discarded the first $5 \cdot 10^4$ iterations of each as burn-in. For the partition function, we used the IS approximation $Z_n^K(\alpha)$ with $K = 10^7$, computed off-line on a grid of α's in $(0, 40]$. After some tuning, we set $L = 40$, $\sigma_\alpha = 0.95$, $\lambda = 0.05$ and $\alpha_{\mathrm{jump}} = 1$, and used the footrule distance. Like DeConde et al. (2006), Deng et al. (2014), and Lin and Ding (2009), our method ranked the genes HPN and AMACR first and

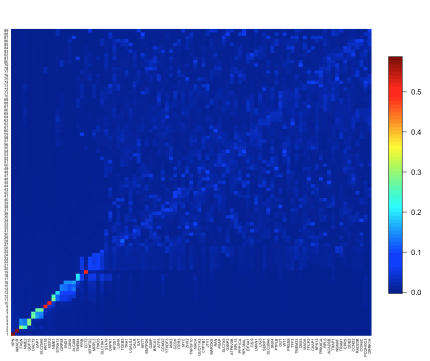


Figure 8: Heat plot of the posterior probabilities, for 89 genes, for being ranked as the $k-$th most preferred, for $k = 1, ..., 89$. On the x-axis the genes are ordered according to the estimated CP consensus.

Rank	MAP	$P(\rho \leq i)$	$P(\rho \leq 10)$	$P(\rho \leq 25)$
1	HPN	0.58	0.72	0.84
2	AMACR	0.59	0.69	0.8
3	NME2	0.26	0.56	0.64
4	GDF15	0.32	0.67	0.79
5	FASN	0.61	0.65	0.76
6	SLC25A6	0.19	0.63	0.71
7	OACT2	0.61	0.63	0.71
8	UAP1	0.62	0.64	0.74
9	KRT18	0.6	0.61	0.72
10	EEF2	0.64	0.64	0.75
11	GRP58	0.13	0.07	0.61
12	NME1	0.68	0.15	0.79
13	STRA13	0.49	0.06	0.56
14	ALCAM	0.33	0.05	0.65
15	SND1	0.51	0.07	0.71
16	CANX	0.59	0.07	0.64
17	TMEM4	0.34	0.05	0.58
18	DAPK1	0.15	0.04	0.21
19	CCT2	0.59	0.05	0.62
20	MRPL3	0.36	0.06	0.6
21	MTHFD2	0.43	0.06	0.58
22	PPIB	0.51	0.06	0.57
23	SLC19A1	0.42	0.06	0.53
24	FMO5	0.58	0.05	0.59
25	TRAM1	0.14	0.04	0.14

Table 3: Top-25 genes in the MAP consensus ranking from a total of 89 genes. The cumulative probability of each gene in the top-25 positions in the MAP of being in that position, or higher, is shown in the third column of the Table, $P(\rho \leq i)$. The probabilities of being among the top-10 and top-25 are also shown for each gene.

second in the MAP consensus ranking. The low value of the posterior mean of α, being 0.56 (mode 0.43, high posterior density, HPD, interval $(0.04, 1.29)$), is an indicator of a generally low level of agreement between the studies. In addition, the fact that $n > N$, and having partial data, both contribute to keeping α small. However, the posterior probability for each gene to be among the top-10 or top-25 is not so low, thus demonstrating that our approach can provide a valid criterion for consensus. In the hypothetical situation in which we had included in our analysis all n^* genes following a *full analysis* mode, with n^* being at least 7567, the largest number of genes included in in any of the five original studies (DeConde et al., 2006), this would have had the effect of making the posterior probabilities in Table 3 smaller. On the other hand, because of Corollary 2, the ranking order obtained from such hypothetical analysis based on all n^* genes would remain the same as in Table 3.

Next we compared the result shown in Table 3 with other approaches: Table 4 (left) reports results obtained with `RankAggreg` (Pihur et al., 2009), which is specifically targeted to meta-analysis problems, while in Table 4 (right) different aggregation methods implemented in `TopKLists` (Schimek et al., 2015) are considered. The results obtained via `RankAggreg` turned out unstable, with the final output changing in every run, and the list shown in Table 4 differs from that in Pihur et al. (2009). Overall, apart from the genes ranked to the top-2 places, there is still considerable variation in the exact rankings of

rank	CE algorithm	GA algorithm
1	HPN	HPN
2	AMACR	AMACR
3	FASN	NME2
4	GDF15	0ACT2
5	NME2	GDF15
6	0ACT2	FASN
7	KRT18	KRT18
8	UAP1	SLC25A6
9	NME1	UAP1
10	EEF2	SND1
11	STRA13	EEF2
12	ALCAM	NME1
13	GRP58	STRA13
14	CANX	ALCAM
15	SND1	GRP58
16	SLC25A6	TMEM4
17	TMEM4	CCT2
18	PPIB	FM05
19	CCT2	CANX
20	MRPL3	DYRK1A
21	MTHFD2	MTHFD2
22	SLC19A1	CALR
23	FM05	MRPL3
24	PRSS8	TRA1
25	NACA	NACA

rank	mean	median	geo.mean	l2norm
1	HPN	HPN	HPN	HPN
2	AMACR	AMACR	AMACR	AMACR
3	GDF15	FASN	FASN	GDF15
4	FASN	KRT18	GDF15	NME1
5	NME1	GDF15	NME2	FASN
6	KRT18	NME1	SLC25A6	KRT18
7	EEF2	EEF2	EEF2	EEF2
8	NME2	UAP1	0ACT2	NME2
9	0ACT2	CYP1B1	OGT	UAP1
10	SLC25A6	ATF5	KRT18	0ACT2
11	UAP1	BRCA1	NME1	SLC25A6
12	CANX	LGALS3	UAP1	STRA13
13	GRP58	MYC	CYP1B1	CANX
14	STRA13	PCDHGC3	ATF5	GRP58
15	SND1	WT1	CBX3	SND1
16	OGT	TFF3	SAT	ALCAM
17	ALCAM	MARCKS	CANX	TMEM4
18	CYP1B1	OS-9	BRCA1	MTHFD2
19	MTHFD2	CCND2	GRP58	MRPL3
20	ATF5	DYRK1A	MTHFD2	PPIB
21	CBX3	TRAP1	STRA13	OGT
22	SAT	FM05	LGALS3	CYP1B1
23	BRCA1	ZHX2	ANK3	SLC19A1
24	MRPL3	RPL36AL	GUCY1A3	ATF5
25	LGALS3	ITPR3	LDHA	CBX3

Table 4: Results given by the `RankAggreg` R package (left) and by the `TopKLists` R package (right).

the genes. Rather than considering such exact rankings, however, it may in practice be of more interest to see to what extent the same genes are shared between different top$-k$ lists. Here the results are more positive. For example, of the 10 genes on top of the MAP consensus list of Table 3, always 9 genes turned out to be in common with each of the lists of Table 4, with the exception of the median (column 3 of Table 4, right), where only 7 genes are shared. Column 4 of Table 3 provides additional support to the MAP selection of the top-10: all genes included in that list have posterior probability at least 0.56 for being among the top-10, while for those outside the list it is maximally 0.15.

In order to have a quantification of the quality of the different estimates, we compute the footrule distance for partial data (Critchlow, 2012, p. 30) between $\boldsymbol{\rho}$ and $\mathbf{R}_j$, averaged over the assessors, defined as follows

$$T_{\text{partial}}(\boldsymbol{\rho},\ \mathbf{R}) = \frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{n}|\nu_{R_{ij}} - \nu_{\rho_i}|,$$

where $\nu_{\boldsymbol{\rho}}$, $\nu_{\mathbf{R}_j} \in \mathcal{P}_n$ are equal to $\boldsymbol{\rho}$ and $\mathbf{R}_j$ in their top$-n_j$ ranks (top-25 in the case of gene lists), while the rank $\frac{n+n_j+1}{2}$ is assigned to the items whose rank in $\boldsymbol{\rho}$ and $\mathbf{R}_j$ is not in their top$-n_j$. Note that $\frac{n+n_j+1}{2}$ (equal to 57.5 in this case) is the average of the ranks of the excluded items. Table 5 reports the values of T_{partial} for the various methods. We notice that the minimum value is achieved by the Mallows MAP consensus list.

6.2 Beach Preference Data

Here we consider pair comparison data (Section 4.2) generated as follows: first we chose $n = 15$ images of tropical beaches, shown in Figure 9, such that they differ in terms of

	MAP	CE	GA	mean	median	geo.mean	l2norm
$T_{\text{partial}}(\boldsymbol{\rho},\ \mathbf{R})$	12.56	12.67	12.98	13.52	15.26	14.05	13.04

Table 5: Values of the average footrule distance for partial data T_{partial} between the partial gene lists and the different estimated consensus rankings.

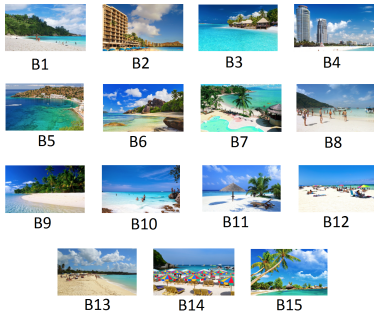


Figure 9: The 15 images used for producing the Beach data set.

$\boldsymbol{\rho}$	CP	$P(\rho_i \le i)$	95% HPDI
1	B9	0.81	(1,2)
2	B6	1	(1,2)
3	B3	0.83	(3,4)
4	B11	0.75	(3,5)
5	B15	0.68	(4,7)
6	B10	0.94	(4,7)
7	B1	1	(6,7)
8	B13	0.69	(8,10)
9	B5	0.55	(8,10)
10	B7	1	(8,10)
11	B8	0.41	(11,14)
12	B4	0.62	(11,14)
13	B14	0.81	(11,14)
14	B12	0.94	(12,15)
15	B2	1	(14,15)

Table 6: Results of the pair comparisons. Beaches arranged according to the CP consensus ordering together with the corresponding 95% highest posterior density intervals.

presence of building and people. For example, beach B9 depicts a very isolated scenery, while beach B2 presents a large hotel seafront.

The pairwise preference data were collected as follows. Each assessor was shown a sequence of 25 pairs of images, and asked on every pair the question: *"Which of the two beaches would you prefer to go to in your next vacation?"*. Each assessor was presented with a random set of pairs, arranged in random order. As there are 105 possible pairs, 25 pairs is less than 25% of the total. We collected $N = 60$ answers. Seven assessors did not answer to all questions, but we kept these responses as our method is able to analyze also incomplete data. Nine assessors returned orderings which contained at least one non-transitive pattern of comparisons. In this analysis we dropped the non-transitive patterns from the data. Systematic methods for dealing with non-transitive rank data will be considered elsewhere.

We run the MCMC for 10^6 iterations, and discarded the first 10^5 iterations as burn-in. We set $L = 2$, $\sigma_\alpha = 0.1$, $\lambda = 0.1$ and $\alpha_{\text{jump}} = 100$. Computing time was less than $2'$.

$\boldsymbol{\rho}$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
BT	B6	B9	B3	B11	B10	B15	B1	B5	B7	B13	B4	B8	B14	B12	B2
PR	B6	B9	B10	B15	B3	B1	B11	B13	B7	B5	B8	B12	B4	B14	B2

Table 7: Consensus ordering given by other methods: `BT` is the Bradley Terry given by the `BradleyTerry2` R package (Firth and Turner, 2012), `PR` is the popular Google PageRank output (Brin and Page, 1998) given by the `igraph` R package (Csardi and Nepusz, 2006). Most preferred to the left.

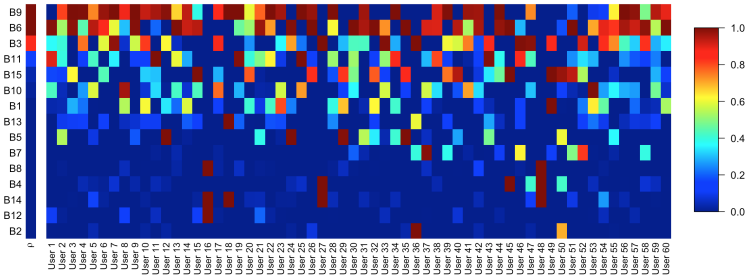


Figure 10: Posterior probability, for each beach, of being ranked among the top-3 in $\boldsymbol{\rho}$ (column 1), and in $\mathbf{R}_j$, $j = 1, ..., 60$ (next columns).

The posterior mean of α was $\mathbb{E}(\alpha|\text{data}) = 3.38\,(2.94, 3.82)$. In Table 6 we report the CP consensus ranking of the beaches (column 2), the cumulative probability of each item i to be in the top$-i$ positions, i.e., $P(\rho_i \leq i)$ (column 3), and the 95% HPDI for each item (column 4), which represents the posterior uncertainty. In Table 7 we give the consensus ranking obtained by two other methods, for comparison.

With our method we also estimate the latent full ranking of each assessor. Figure 10 was obtained as follows: in the separate column on the left, we display the posterior probability $P(\rho_{\text{B}i} \leq 3|\text{data})$ that a given beach Bi, $i = 1, ..., 15$, is among the top-3 in the consensus $\boldsymbol{\rho}$. In the other columns we show, for each beach Bi, the individual posterior probabilities $P(\tilde{R}_{j,\text{B}i} \leq 3|\text{data})$, of being among the top$-3$ for each assessor j, $j = 1, ..., 60$. We see for example that beach B5, which was ranked only 9th in the consensus, had, for 4 assessors, posterior probability very close to 1 of being included among their top-3 beaches.

6.3 Sushi Data

We illustrate clustering based on full rankings using the benchmark data set of sushi preferences collected across Japan (Kamishima, 2003), see also Lu and Boutilier (2014). $N = 5000$ people were interviewed, each giving a complete ranking of $n = 10$ sushi variants. Cultural differences among Japanese regions influence food preferences, so we expect the assessors to

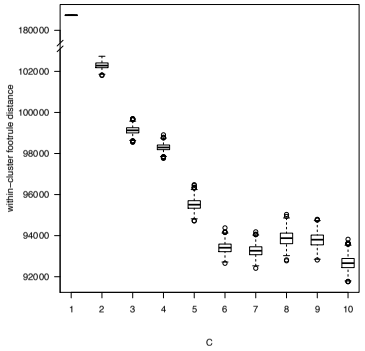


Figure 11: Results of the Sushi experiment. Boxplots of the posterior distributions of the within-cluster sum of footrule distances of assessors' ranks from the corresponding cluster consensus for different choices of C (note the y-axis break, for better visualization).

be clustered according to different shared consensus rankings. We analyzed the sushi data using mixtures of Mallows models (Section 4.3) with the footrule distance (with the exact partition function of the Mallows model, see Section 2.1). We run the MCMC for 10^6 iterations, and discarded the first 10^5 iterations as burn-in. After some tuning, we set $L = 1$, $\sigma_\alpha = 0.1$, $\lambda = 0.1$ and $\alpha_{\text{jump}} = 100$. In the Dirichlet prior for $\boldsymbol{\tau}$, we set the hyper-parameter $\psi = N/C$, thus favoring high-entropy distributions. Computing time varied depending on C, from order of minutes to an hour. For each possible number of clusters $C \in \{1, \ldots, 10\}$, we used a thinned subset of MCMC samples to compute the posterior footrule distance between $\boldsymbol{\rho}_c$ and the ranking of each assessor assigned to that cluster, $\sum_{c=1}^{C} \sum_{j:z_j=c} d(\mathbf{R}_j, \boldsymbol{\rho}_c)$. The posterior of this quantity, over all assessors and cluster centers, was then used for choosing the appropriate value for C, see Figure 11. We found an elbow at $C = 6$, which was then used to further inspect results.

Table 8 shows the results when the number of clusters is set to $C = 6$: for each cluster, the MAP estimates for $\boldsymbol{\tau}$ and $\boldsymbol{\alpha}$, together with their 95% HPDIs, are shown on the top of the table. Table 8 also shows the sushi items, arranged in cluster-specific lists according to the MAP consensus ordering (in this case equal to the CP consensus). Our results can be compared with the ones in Lu and Boutilier (2014) (Table 1 in Section 5.3.2): the correspondence of ours-Lu and Boutilier (2014) clusters could be 1-4, 2-1, 3-2, 4-5, 5-4, 6-0. Note that the dispersion parameter α in our Bayesian Mallows model is connected to the dispersion parameter ϕ in Lu and Boutilier (2014) by the link $\alpha = -n \log(\phi)$. Hence, we can also observe that the cluster-specific α values reported in Table 8 are quite comparable to the dispersion parameters of Lu and Boutilier (2014).

We investigate the stability of the clustering in Figure 12, which shows the heatplot of the posterior probabilities, for all 5000 assessors (on the x-axis), of being assigned to each of the

	$c=1$	$c=2$	$c=3$	$c=4$	$c=5$	$c=6$
τ_c	0.243 (0.23,0.26)	0.131 (0.12,0.14)	0.107 (0.1,0.11)	0.117 (0.11,0.12)	0.121 (0.11,0.13)	0.278 (0.27,0.29)
α_c	3.62 (3.52,3.75)	2.55 (2.35,2.71)	3.8 (3.42,4.06)	4.02 (3.78,4.26)	4.46 (4.25,4.68)	1.86 (1.77,1.94)
1	fatty tuna	shrimp	sea urchin	fatty tuna	fatty tuna	fatty tuna
2	sea urchin	sea eel	fatty tuna	salmon roe	tuna	tuna
3	salmon roe	egg	shrimp	tuna	tuna roll	sea eel
4	sea eel	squid	tuna	tuna roll	shrimp	shrimp
5	tuna	cucumber roll	squid	shrimp	squid	salmon roe
6	shrimp	tuna	tuna roll	egg	sea eel	tuna roll
7	squid	tuna roll	salmon roe	squid	egg	squid
8	tuna roll	fatty tuna	cucumber roll	cucumber roll	cucumber roll	sea urchin
9	egg	salmon roe	egg	sea eel	salmon roe	egg
10	cucumber roll	sea urchin	sea eel	sea urchin	sea urchin	cucumber roll

Table 8: Results of the Sushi experiment when setting $C = 6$. Sushi items arranged according to the MAP consensus ranking found from the posterior distribution of $\boldsymbol{\rho}_c$, $c = 1, \ldots, 6$. At the top of the Table, corresponding MAP estimates for $\boldsymbol{\tau}$ and $\boldsymbol{\alpha}$, with 95% HPDIs (in parenthesis). Results are based on 10^6 MCMC iterations.

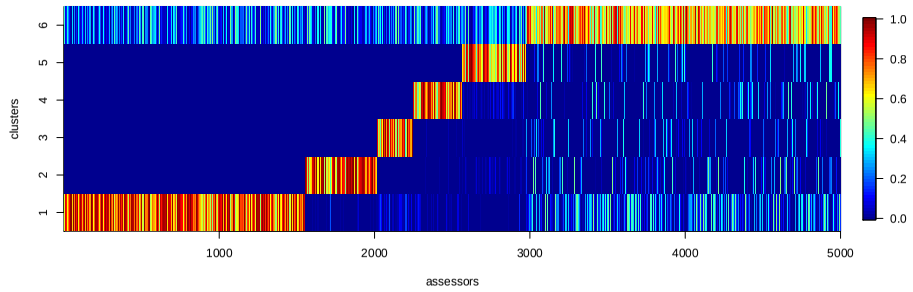


Figure 12: Heatplot of posterior probabilities for all 5000 assessors (on the x-axis) of being assigned to each cluster ($c = 1, \ldots, 6$ from bottom to top).

6 clusters in Table 8 (clusters $c = 1, \ldots, 6$ from bottom to top in Figure 12): most of these individual probabilities were concentrated on some particular preferred value of c among the six possibilities, indicating a reasonably stable behavior in the cluster assignments.

6.4 Movielens Data

The Movielens data set[1] contains movie ratings from 6040 users. In this example, we focused on the $n = 200$ most rated movies, and on the $N = 6004$ users who rated (not equally) at

1. www.grouplens.org/datasets/.

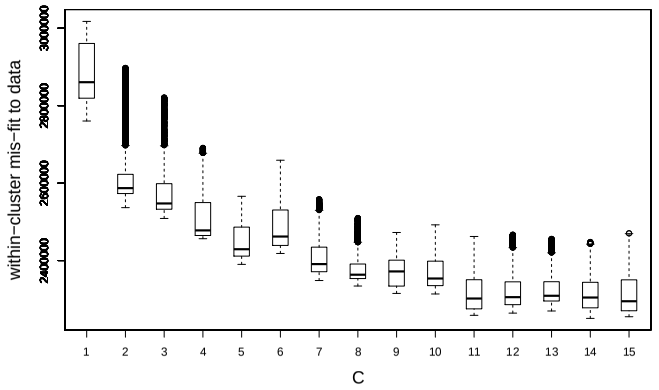


Figure 13: Results of the Movielens experiment. Boxplots of the posterior distributions of the within-cluster indicator of mis-fit to the data, as introduced in Section 4.4, for different choices of C.

least 3 movies. Each user had considered only a subset of the n movies (30.2 on average). We converted the ratings given by each user from a 1-5 scale to pairwise preferences as described in Lu and Boutilier (2014): each movie was preferred to all movies which the user had rated strictly lower. We selected users whose rating included at least 3 movies, because two of them were needed to create at least a pairwise comparison, and the third one was needed for prediction, as explained in the following.

Since we expected heterogeneity among users, due to age/gender/social factors/education, we applied the clustering scheme for pairwise preferences, with the footrule distance. Since $n = 200$, we used the asymptotic approximation for $Z_n(\alpha)$ described in Mukherjee (2016) and in Section 2 of the Supplementary Material. We run the MCMC for 10^5 iterations, after a burn-in of $5 \cdot 10^4$ iterations. We set: $L = 20$, $\sigma_\alpha = 0.05$, $\alpha_{\text{jump}} = 10$ and $\lambda = 0.1$, after some tuning. Note that the label switching problem only affects inference on cluster-specific parameters, but it does not affect predictive distributions (Celeux et al., 2006). We varied the number C of clusters in the set $\{1, \ldots, 15\}$, and inspected the within-cluster indicator of mis-fit to the data, $\sum_{c=1}^{C} \sum_{j:z_j=c} |\{B \in \text{tc}(\mathcal{B}_j) : B \text{ is not consistent with } \boldsymbol{\rho}_c\}|$, introduced in Section 4.4, see Figure 13: the posterior within-cluster indicator shows two possible elbows: $C = 5$, and $C = 11$. Hence, according to these criteria, both choices seemed initially conceivable. However, it is beyond the scope of this paper to discuss ways to decide the number of clusters.

In order to select one of these two models, we examined their predictive performance. Before converting ratings to preferences, we discarded for each user j one of the rated movies at random. Then, we randomly selected one of the other movies rated by the same user, and

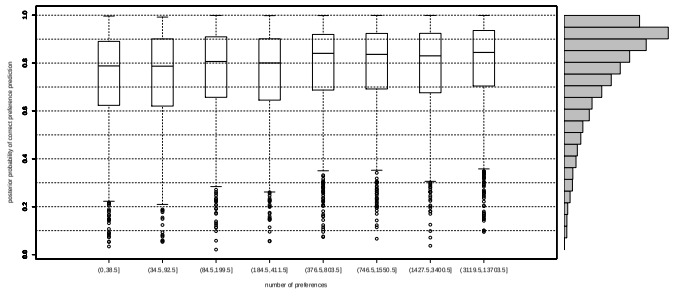


Figure 14: Results of the Movielens experiment. Boxplots of the posterior probability for correctly predicting the discarded preference conditionally on the number of preferences stated by the user, for the model with $C = 5$. The histogram on the right shows the marginal posterior probability for correct preference prediction.

used it to create a pairwise preference involving the discarded movie. This preference was then not used for inference. After running the Bayesian Mallows model, we computed for each user the predictive probabilities $P(\tilde{\mathbf{R}}_j|\text{data})$, and thereby the probabilities for correctly predicting the discarded preference. The median, across all users, of these probabilities was 0.8225 for the model with $C = 5$ clusters, and 0.796 for $C = 11$ clusters. Moreover, for $C = 5$, 88 % of these probabilities were higher than 0.5. These are very positive results, and they suggest that the predictive performance of the model with 5 clusters is slightly better than the one with 11 clusters. It appears that the larger number of clusters in the latter model leads to a slight overfitting, and this is likely to be the main cause of the loss in the predictive success. Figure 14 shows the boxplots of the posterior distribution of the probability for correct preference prediction of the left out comparison, stratified with respect to the number of preferences given by each user, for the model with $C = 5$. The histogram on the right shows the same posterior probability for correctly predicting the discarded preference for all users, for the same model, regardless of how many preferences each user had expressed. Interestingly, in this data, the predictive power is rather stable and high, irrespectively from how many movies the users rated. In other applications, we would expect the predictions to become better the more preferences are expressed by a user. In this case, a figure similar to Figure 14 could guide personal recommendation algorithms, which should not rely on estimated point preferences, if these are too uncertain, as happens for users who have given a few ratings only.

In Table 9 the MAP estimates for $\boldsymbol{\tau}$ and $\boldsymbol{\alpha}$, together with their 95% HPDIs, are shown at the top. The table also shows a subset of the movies, arranged in cluster-specific top-10 lists according to the CP consensus ranking, from the posterior distribution of $\boldsymbol{\rho}_c$, $c = 1, \ldots, 5$. We note that all α values correspond to a reasonable within-cluster variability. Moreover, the lists reported in Table 9 characterize the users in the same cluster as individuals sharing a reasonably well interpretable preference profile. Since in the Movielens data set additional

	$c = 1$	$c = 2$	$c = 3$	$c = 4$	$c = 5$
τ_c	0.325 (0.32,0.33)	0.219 (0.21,0.23)	0.156 (0.15,0.17)	0.145 (0.14,0.15)	0.155 (0.15,0.16)
α_c	2.53 (2.36,2.7)	3.33 (3.2,3.48)	2.58 (2.27,2.81)	1.87 (1.67,2.02)	2.68 (2.47,2.89)
1	A Christmas Story	Citizen Kane	The Sting	Indiana Jones (I)	Shawshank Redemption
2	Schindler's List	The Godfather	Dr. Strangelove	A Christmas Story	Indiana Jones (I)
3	The Godfather	Pulp Fiction	2001: A Space Odyssey	Star Wars (IV)	Braveheart
4	Casablanca	Dr. Strangelove	The Maltese Falcon	The Princess Bride	Star Wars (IV)
5	Star Wars (IV)	A Clockwork Orange	Casablanca	Schindler's List	Saving Private Ryan
6	Shawshank Redemption	Casablanca	Taxi Driver	The Matrix	The Green Mile
7	Saving Private Ryan	The Usual Suspects	Citizen Kane	Shawshank Redemption	Schindler's List
8	The Sting	2001: A Space Odyssey	Schindler's List	Indiana Jones (III)	The Sixth Sense
9	The Sixth Sense	American Beauty	Chinatown	The Sting	The Matrix
10	American Beauty	Star Wars (IV)	The Godfather	The Sixth Sense	Star Wars (V)

Table 9: Results of the Movielens experiment. Movies arranged according to the CP consensus ranking, from the posterior distribution of $\boldsymbol{\rho}_c$, $c = 1, \ldots, 5$.

information on the users is available, we compared the estimated cluster assignments with the age, gender, and the occupation of the users. While occupation showed no interesting patterns, the second and fifth clusters had more males than expected, in contrast to the first and fourth clusters which included more females than average, the former above 45 and the latter below 35 of age.

7. Discussion

In this paper, we developed a fully Bayesian hierarchical framework for the analysis of rank data. An important advantage of the Bayesian approach is that it offers coherently propagated and directly interpretable ways to quantify posterior uncertainties of estimates of any quantity of interest. Earlier Bayesian treatments of the Mallows rank model are extended in many ways: we develop an importance sampling scheme for $Z_n(\alpha)$ allowing the use of other distances than Kendall's, and our MCMC algorithm efficiently samples from the posterior distribution of the unknown consensus ranking and of the latent assessor-specific full rankings. We also develop various extensions of the model, motivated by applications in which data take particular forms.

The Mallows model performs very well with a large number of assessors N, as we show in the Sushi experiment of Section 6.3, and in the Movielens experiment of Section 6.4. On the other hand, it may not be computationally feasible when the number of items is extremely large, for example $n \geq 10^4$, which is not uncommon in certain applications (Volkovs and Zemel, 2014). For the footrule and Spearman distances, there exist asymptotic approximations for $Z_n(\alpha)$ as $n \to \infty$ (Mukherjee, 2016), which we successfully used in Section 6.4, although the MCMC algorithm converges slowly in such large spaces. Maximum likelihood estimation of $\boldsymbol{\rho}$ runs into the same problem when n gets large (Aledo et al., 2013; Ali and Meilă, 2012). Volkovs and Zemel (2014) developed the multinomial preference model (MPM) for cases with very large n, which can be efficiently computed by maximizing a concave log-likelihood function. The MPM thus seems a useful choice when n is very large and real time performance is needed.

All methods presented have been implemented in C++, and run efficiently on a desktop computer, with the exception of the Movielens experiment, which needed to be run on a cluster. Obtaining a sufficiently large sample from the posterior distribution takes from

a few seconds, for small problems, to several minutes, in the examples involving massive data augmentation. We are also working on distributed versions of the MCMC on parallel synchronous and asynchronous machines.

Many of the extensions we propose for solving specific problems (for example, clustering, preference prediction, pairwise comparisons) are needed jointly in real applications, as we illustrate for example in the Movielens data. Our general framework is flexible enough to handle such extensions.

There are many situations in which rankings vary over time, as in political surveys (Regenwetter et al., 1999) or book bestsellers (Caron and Teh, 2012). We have extended our approach to this setting (Asfaw et al., 2017). We assume to observe ranks at discrete time-points indexed by $t = 0, 1, \ldots, T$ and let $\boldsymbol{\rho}^{(t)}$ and $\alpha^{(t)}$ denote the parameters of the Mallows model at time t. Interestingly, this model allows for prediction (with uncertainty quantification) of rankings in future time instances.

A natural generalization of our model is to allow for item-specific α's. This is known as generalized Mallows's model, first implemented in Fligner and Verducci (1986), for Kendall and Cayley distances, and further extended in Meilă and Bao (2010), for Kendall distance only, to the Bayesian framework. To our knowledge, the Mallows model with footrule and Spearman has not yet been generalized to handle item-specific α's, mostly because of the obvious computational difficulties. Within our framework this appears as feasible.

Acknowledgments

Valeria Vitelli and Øystein Sørensen contributed equally to this paper and are joint first authors. Marta Crispino contributed to the project as a visiting PhD student at OCBE, University of Oslo, and was partially funded by Cariplo. The authors thank Tyler Lu and Craig Boutilier for their help with the Movielens data, and Magne Thoresen for helpful discussions. The authors would also like to thank the Editor and three anonymous reviewers for their useful comments and suggestions on a previous version of the paper.

Appendix A. Proofs of Results from Section 4.1.1

Proof of Proposition 4.
Having assumed the uniform prior across all permutations of latent consensus ranks, the desired result will hold if and only if $\sum_{j=1,\ldots,N} d(\mathbf{R}_j, \boldsymbol{\rho}) \leq \sum_{j=1,\ldots,N} d(\mathbf{R}_j, \boldsymbol{\rho}')$. This is true if $d(\mathbf{R}_j, \boldsymbol{\rho}) \leq d(\mathbf{R}_j, \boldsymbol{\rho}')$ holds separately for each assessor j, for $j = 1, \ldots, N$. We consider first the footrule distance d, and then show that the result holds also for the Kendall and Spearman distances. This proof follows Proposition 4 in Meilă and Bao (2010).

Suppose first, for simplicity, that all assessors have ranked the same n items, that is, $\mathcal{A}_1 = \mathcal{A}_2 = \ldots = \mathcal{A}_N = \mathcal{A}$. Later we allow the sets $\mathcal{A}_j$ of ranked items to be different for different assessors. Thus there are $n^* - n$ items, which nobody ranked in the original data.

We now introduce synthetic rankings for all these items as well, that is, we augment each $\mathbf{R}_j$ as recorded in the data by replacing the missing ranks of the items $A_i \in \mathcal{A}^c$ by some permutation of their possible ranks from $n+1$ to n^*. We then show that the desired

inequality holds regardless of how these ranks $\{R_{ij}, A_i \in \mathcal{A}^c\}$ were assigned. The proof is by induction, and it is carried out in several steps.

For the first step, let $\boldsymbol{\rho}$ be a rank vector were the ranks from 1 to n, in any order, have been assigned to the items in $\mathcal{A}$, and the ranks R_{ij} between $n+1$ and n^* are given to items in $\mathcal{A}^c$. Let $\boldsymbol{\rho}'$ be a rank vector obtained from $\boldsymbol{\rho}$ by a transposition of the ranks of two items, say, of $A_{i_0} \in \mathcal{A}^c$ and $A_{i_1} \in \mathcal{A}$, with $\rho_{i_0} = \rho'_{i_1} \geq n+1$ and $\rho_{i_1} = \rho'_{i_0} \leq n$. Fixing these two items, we want to show that $d(\mathbf{R}_j, \boldsymbol{\rho}) \leq d(\mathbf{R}_j, \boldsymbol{\rho}')$. For the footrule distance we have to show that $\sum_{i=1}^{n} |R_{ij} - \rho_i| \leq \sum_{i=1}^{n} |R_{ij} - \rho'_i|$. Since $\boldsymbol{\rho}$ and $\boldsymbol{\rho}'$ coincide for all their coordinates $i \neq i_0, i_1$, it is enough to compare here the terms $|R_{i_0 j} - \rho_{i_0}|$ and $|R_{i_1 j} - \rho_{i_1}|$ on the left to the corresponding terms $|R_{i_0 j} - \rho'_{i_0}|$ and $|R_{i_1 j} - \rho'_{i_1}|$ on the right. We need to distinguish between two situations:

(i) Suppose $R_{i_1 j} \leq \rho_{i_1}$. Then, $\rho'_{i_1} - R_{i_1 j} > \rho_{i_1} - R_{i_1 j}$. On the other hand, $\rho_{i_0} \geq n+1$ implies that $A_{i_0} \in \mathcal{A}^c$, and it is therefore ranked by assessor j with $R_{i_0 j} \geq n+1$. Therefore, $|R_{i_0 j} - \rho'_{i_0}| \geq |R_{i_0 j} - \rho_{i_0}|$. By combining these two results we get that $|R_{i_0 j} - \rho_{i_0}| + |R_{i_1 j} - \rho_{i_1}| \leq |R_{i_0 j} - \rho'_{i_0}| + |R_{i_1 j} - \rho'_{i_1}|$.

(ii) Now, suppose that $R_{i_1 j} > \rho_{i_1}$. Then, $R_{i_1 j} - \rho_{i_1} \leq n - \rho_{i_1} \leq R_{i_0 j} - \rho'_{i_0}$. Moreover, since $|R_{i_0 j} - \rho_{i_0}| \leq |R_{i_1 j} - \rho_{i_0}| = |R_{i_1 j} - \rho'_{i_1}|$, we have that again $|R_{i_0 j} - \rho_{i_0}| + |R_{i_1 j} - \rho_{i_1}| \leq |R_{i_0 j} - \rho'_{i_0}| + |R_{i_1 j} - \rho'_{i_1}|$ holds.

The same reasoning holds also for the Kendall distance, since the Kendall distance between the two rank vectors, which are obtained from each other by a transposition of a pair of items, is the same as the footrule distance. For the Spearman distance, we only need to form squares of the distance between pairs of items, and the inequality remains valid.

For the general step of the induction, suppose that $\boldsymbol{\rho}$ has been obtained from its original version with all items in $\mathcal{A}$ ranked to the first n positions, via a sequence of transpositions between items originally in $\mathcal{A}$ and items originally in $\mathcal{A}^c$. Let $\boldsymbol{\rho}'$ be a rank vector where one more transposition of this type from $\boldsymbol{\rho}$ to $\boldsymbol{\rho}'$ has been carried out. Then the argument of the proof can still be carried through, and the conclusion $d(\mathbf{R}_j, \boldsymbol{\rho}) \leq d(\mathbf{R}_j, \boldsymbol{\rho}')$ holds. This argument needs to be complemented by considering the uniform random permutations, corresponding to the assumed prior of the ranks originally missing in the data, across their possible values from $n+1$ to n^*. But this is automatic, because the conclusion holds separately for all permutations of such ranks.

Finally, the argument needs to be extended to the situation in which the sets $\mathcal{A}_j$ of ranked items can be different for different assessors. In this case we are led to consider, as a by-product of the data augmentation scheme, a joint distribution of the rank vectors $\{\tilde{\mathbf{R}}_j; j = 1, \ldots, N\}$. Here, for each j, the n_j items which were ranked first have been fixed by the data. The remaining $n - n_j$ items are assigned augmented random ranks with values between $n_j + 1$ and n, where the probabilities, corresponding to the model P_{n^*}, are determined by the inference from the assumed Mallows model and the data. The conclusion remains valid regardless of the particular way in which the augmentation was done, and so it holds also when taking an expectation with respect to P_{n^*}. ■

Proof of Corollary 2.
It follows from Proposition 4 that the n top ranks in $\boldsymbol{\rho}^{MAP*}$ are all assigned to items $A_i \in \mathcal{A}$.

Therefore, using shorthand $\boldsymbol{\rho}_{\mathcal{A}} = (\boldsymbol{\rho}_i; A_i \in \mathcal{A})$ and $\boldsymbol{\rho}_{\mathcal{A}^c} = (\boldsymbol{\rho}_i; A_i \in \mathcal{A}^c)$ we see that $\boldsymbol{\rho}^{MAP*}$ must be of the form $\boldsymbol{\rho}^{MAP*} = (\boldsymbol{\rho}^{MAP*}_{\mathcal{A}}, \boldsymbol{\rho}^{MAP*}_{\mathcal{A}^c}) = (\boldsymbol{\pi}, \boldsymbol{\pi}')$, where $\boldsymbol{\pi}$ is a permutation of the set $(1, 2, \ldots, n)$, and similarly $\boldsymbol{\pi}'$ is some permutation of $(n+1, \ldots, n^*)$.

To prove the statement, we show the following: (i) the posterior probabilities $P_{n^*}(\boldsymbol{\rho}_{\mathcal{A}} = \boldsymbol{\pi}, \boldsymbol{\rho}_{\mathcal{A}^c} = \boldsymbol{\pi}'|\text{data})$ and $P_{n^*}(\boldsymbol{\rho}_{\mathcal{A}} = \boldsymbol{\pi}|\boldsymbol{\rho}_{\mathcal{A}^c} = \boldsymbol{\pi}', \text{data})$ are invariant under permutations of $\boldsymbol{\pi}'$, and (ii) the latter conditional probabilities $P_{n^*}(\boldsymbol{\rho}_{\mathcal{A}} = \boldsymbol{\pi}|\boldsymbol{\rho}_{\mathcal{A}^c} = \boldsymbol{\pi}', \text{data})$ coincide with $P_n(\boldsymbol{\rho}_{\mathcal{A}} = \boldsymbol{\pi}|\text{data})$. As a consequence, a list of top-n items obtained from the *full analysis* estimate $\boldsymbol{\rho}^{MAP*}$ qualifies also as the *restricted analysis* estimate $\boldsymbol{\rho}^{MAP}$, and conversely, $\boldsymbol{\rho}^{MAP}$ can be augmented with any permutation $\boldsymbol{\pi}'$ of $(n+1, \ldots, n^*)$ to jointly form $\boldsymbol{\rho}^{MAP*}$.

The first part of (i) follows by noticing that the likelihood in the *full analysis*, when considering consensus rankings of the form $\boldsymbol{\rho} = (\boldsymbol{\rho}_{\mathcal{A}}, \boldsymbol{\rho}_{\mathcal{A}^c}) = (\boldsymbol{\pi}, \boldsymbol{\pi}')$, only depends on the observed data via $\boldsymbol{\pi}$. Since the assessors act independently, each imposing a uniform prior on their unranked items, also the posterior $P_{n^*}(\boldsymbol{\rho}_{\mathcal{A}} = \boldsymbol{\pi}, \boldsymbol{\rho}_{\mathcal{A}^c} = \boldsymbol{\pi}'|\text{data})$ will depend only on $\boldsymbol{\pi}$. The second part follows from the first, either by direct conditioning in the joint distribution, or by first computing the marginal $P_{n^*}(\boldsymbol{\rho}_{\mathcal{A}^c} = \boldsymbol{\pi}'|\text{data})$ by summation, and then dividing. (ii) follows then because, for both posterior probabilities, the sample space, the prior, and the likelihood are the same. ■

Appendix B. Pseudo-codes of the Algorithms

We here report the pseudo-codes of the algorithms. The available distance functions are: Kendall, footrule, Spearman. Cayley and Hamming are easy to implement. For Kendall, Cayley and Hamming, there is no need to run the IS to approximate $Z_n(\alpha)$, as the closed form is available (Fligner and Verducci, 1986). For footrule ($n \leq 50$) and Spearman ($n \leq 14$) the algorithm exploits the results presented in Section 2.1. For footrule ($n > 50$) and Spearman ($n > 14$) the IS procedure has to be run off-line, before the MCMC.

Algorithm 1: Basic MCMC Algorithm for Complete Rankings

input : $\mathbf{R}_1, \ldots, \mathbf{R}_N$; λ, σ_α, α_{jump}, L, $d(\cdot,\cdot)$, $Z_n(\alpha)$, M.
output: Posterior distributions of $\boldsymbol{\rho}$ and α.
Initialization of the MCMC: randomly generate $\boldsymbol{\rho}_0$ and α_0.

for $m \leftarrow 1$ **to** M **do**
  M-H step: update $\boldsymbol{\rho}$**:**
  sample: $\boldsymbol{\rho}' \sim \text{L\&S}(\boldsymbol{\rho}_{m-1}, L)$ and $u \sim \mathcal{U}(0,1)$
  compute: $ratio \leftarrow$ equation (6) with $\boldsymbol{\rho} \leftarrow \boldsymbol{\rho}_{m-1}$ and $\alpha \leftarrow \alpha_{m-1}$
  if $u < ratio$ **then** $\boldsymbol{\rho}_m \leftarrow \boldsymbol{\rho}'$
  else $\boldsymbol{\rho}_m \leftarrow \boldsymbol{\rho}_{m-1}$

  if $m \mod \alpha_{jump} = 0$ **then M-H step: update** α**:**
  sample: $\alpha' \sim \log \mathcal{N}(\alpha_{m-1}, \sigma^2_\alpha)$ and $u \sim \mathcal{U}(0,1)$
  compute: $ratio \leftarrow$ equation (8) with $\boldsymbol{\rho} \leftarrow \boldsymbol{\rho}_m$ and $\alpha \leftarrow \alpha_{m-1}$
  if $u < ratio$ **then** $\alpha_m \leftarrow \alpha'$
  else $\alpha_m \leftarrow \alpha_{m-1}$
end

Algorithm 2: MCMC Algorithm for Clustering Complete Rankings

input : $\mathbf{R}_1, \ldots, \mathbf{R}_N$; C, ψ, λ, σ_α, α_{jump}, L, $d(\cdot,\cdot)$, $Z_n(\alpha)$, M.
output: Posterior distributions of $\boldsymbol{\rho}_1, \ldots, \boldsymbol{\rho}_C$, $\alpha_1, \ldots, \alpha_C$, $\tau_1, \ldots, \tau_C$, $z_1, \ldots, z_N$.
Initialization of the MCMC: randomly generate $\boldsymbol{\rho}_{1,0}, \ldots, \boldsymbol{\rho}_{C,0}$, $\alpha_{1,0}, \ldots, \alpha_{C,0}$, $\tau_{1,0}, \ldots, \tau_{C,0}$, and $z_{1,0}, \ldots, z_{N,0}$.

for $m \leftarrow 1$ **to** M **do**
  Gibbs step: update $\tau_1, \ldots, \tau_C$
  compute: $n_c = \sum_{j=1}^N 1_c(z_{j,m-1})$, for $c = 1, \ldots, C$
  sample: $\tau_1, \ldots, \tau_C \sim \mathcal{D}(\psi + n_1, \ldots, \psi + n_C)$

  for $c \leftarrow 1$ **to** C **do**
    M-H step: update $\boldsymbol{\rho}_c$
    sample: $\boldsymbol{\rho}'_c \sim \text{L\&S}(\boldsymbol{\rho}_{c,m-1}, L)$ and $u \sim \mathcal{U}(0,1)$
    compute: $ratio \leftarrow$ equation (6) with $\boldsymbol{\rho} \leftarrow \boldsymbol{\rho}_{c,m-1}$ and $\alpha \leftarrow \alpha_{c,m-1}$, and where the sum is over $\{j : z_{j,m-1} = c\}$
    if $u < ratio$ **then** $\boldsymbol{\rho}_{c,m} \leftarrow \boldsymbol{\rho}'_c$
    else $\boldsymbol{\rho}_{c,m} \leftarrow \boldsymbol{\rho}_{c,m-1}$

    if $m \mod \alpha_{jump} = 0$ **then M-H step: update** α_c sample: $\alpha'_c \sim \mathcal{N}(\alpha_{c,m-1}, \sigma^2_\alpha)$ and $u \sim \mathcal{U}(0,1)$
    compute: $ratio \leftarrow$ equation (8) with $\boldsymbol{\rho} \leftarrow \boldsymbol{\rho}_{c,m}$ and $\alpha \leftarrow \alpha_{c,m-1}$, and where the sum is over $\{j : z_{j,m-1} = c\}$
    if $u < ratio$ **then** $\alpha_{c,m} \leftarrow \alpha'_c$
    else $\alpha_{c,m} \leftarrow \alpha_{c,m-1}$
  end

  Gibbs step: update $z_1, \ldots, z_N$
  for $j \leftarrow 1$ **to** N **do**
    foreach $c \leftarrow 1$ **to** C **do** compute cluster assignment probabilities: $p_{cj} = \frac{\tau_{c,m}}{Z_n(\alpha_{c,m})} \exp\left[\frac{-\alpha_{c,m}}{n} d(\mathbf{R}_j, \boldsymbol{\rho}_{c,m})\right]$
    sample: $z_{j,m} \sim \mathcal{M}(p_{1j}, \ldots, p_{Cj})$
  end
end

Algorithm 3: MCMC Algorithm for Partial Rankings or Pairwise Preferences

input : $\{\mathcal{S}_1, \ldots, \mathcal{S}_N\}$ or $\{\text{tc}(\mathcal{B}_1), \ldots, \text{tc}(\mathcal{B}_N)\}$; λ, σ_α, α_{jump}, L, $d(\cdot,\cdot)$, $Z_n(\alpha)$, M.
output: Posterior distributions of $\boldsymbol{\rho}$, α and $\tilde{\mathbf{R}}_1, \ldots, \tilde{\mathbf{R}}_N$.
Initialization of the MCMC: randomly generate $\boldsymbol{\rho}_0$ and α_0.

if $\{\mathcal{S}_1, \ldots, \mathcal{S}_N\}$ *among inputs* **then**
  foreach $j \leftarrow 1$ **to** N **do** randomly generate $\tilde{\mathbf{R}}^0_j$ in $\mathcal{S}_j$
else
  foreach $j \leftarrow 1$ **to** N **do** randomly generate $\tilde{\mathbf{R}}^0_j$ compatible with $\text{tc}(\mathcal{B}_j)$
end

for $m \leftarrow 1$ **to** M **do**
  M-H step: update $\boldsymbol{\rho}$**:**
  sample: $\boldsymbol{\rho}' \sim \text{L\&S}(\boldsymbol{\rho}_{m-1}, L)$ and $u \sim \mathcal{U}(0,1)$
  compute: $ratio \leftarrow$ equation (6) with $\boldsymbol{\rho} \leftarrow \boldsymbol{\rho}_{m-1}$ and $\alpha \leftarrow \alpha_{m-1}$
  if $u < ratio$ **then** $\boldsymbol{\rho}_m \leftarrow \boldsymbol{\rho}'$
  else $\boldsymbol{\rho}_m \leftarrow \boldsymbol{\rho}_{m-1}$

  if $m \mod \alpha_{jump} = 0$ **then M-H step: update** α**:**
  sample: $\alpha' \sim \mathcal{N}(\alpha_{m-1}, \sigma^2_\alpha)$ and $u \sim \mathcal{U}(0,1)$
  compute: $ratio \leftarrow$ equation (8) with $\boldsymbol{\rho} \leftarrow \boldsymbol{\rho}_m$ and $\alpha \leftarrow \alpha_{m-1}$
  if $u < ratio$ **then** $\alpha_m \leftarrow \alpha'$
  else $\alpha_m \leftarrow \alpha_{m-1}$

  M-H step: update $\tilde{\mathbf{R}}_1, \ldots, \tilde{\mathbf{R}}_N$**:**
  for $j \leftarrow 1$ **to** N **do**
    if $\{\mathcal{S}_1, \ldots, \mathcal{S}_N\}$ *among inputs* **then** sample: $\tilde{\mathbf{R}}'_j$ in $\mathcal{S}_j$ from the leap-and-shift distribution centered at $\tilde{\mathbf{R}}^{m-1}_j$
    else sample: $\tilde{\mathbf{R}}'_j$ from the leap-and-shift distribution centered at $\tilde{\mathbf{R}}^{m-1}_j$ and compatible with $\text{tc}(\mathcal{B}_j)$
    compute: $ratio \leftarrow$ equation (21) with $\boldsymbol{\rho} \leftarrow \boldsymbol{\rho}_m$, $\alpha \leftarrow \alpha_m$ and $\tilde{\mathbf{R}}_j \leftarrow \tilde{\mathbf{R}}^{m-1}_j$
    sample: $u \sim \mathcal{U}(0,1)$
    if $u < ratio$ **then** $\tilde{\mathbf{R}}^m_j \leftarrow \tilde{\mathbf{R}}'_j$
    else $\tilde{\mathbf{R}}^m_j \leftarrow \tilde{\mathbf{R}}^{m-1}_j$
  end
end

Algorithm 4: MCMC Algorithm for Clustering Partial Rankings or Pairwise Preferences

input : $\{\mathcal{S}_1, \ldots, \mathcal{S}_N\}$ or $\{\text{tc}(\mathcal{B}_1), \ldots, \text{tc}(\mathcal{B}_N)\}$; C, ψ, λ, σ_α, α_{jump}, L, $d(\cdot,\cdot)$, $Z_n(\alpha)$, M.
output: Posterior distributions of $\boldsymbol{\rho}_1, \ldots, \boldsymbol{\rho}_C$, $\alpha_1, \ldots, \alpha_C$, $\tau_1, \ldots, \tau_C$, $z_1, \ldots, z_N$, and $\tilde{\mathbf{R}}_1, \ldots, \tilde{\mathbf{R}}_N$.
Initialization of the MCMC:
randomly generate $\boldsymbol{\rho}_{1,0}, \ldots, \boldsymbol{\rho}_{C,0}$, $\alpha_{1,0}, \ldots, \alpha_{C,0}$, $\tau_{1,0}, \ldots, \tau_{C,0}$, and $z_{1,0}, \ldots, z_{N,0}$.

if $\{\mathcal{S}_1, \ldots, \mathcal{S}_N\}$ *among inputs* **then**
  foreach $j \leftarrow 1$ **to** N **do** randomly generate $\tilde{\mathbf{R}}_j^0$ in $\mathcal{S}_j$
else
  foreach $j \leftarrow 1$ **to** N **do** randomly generate $\tilde{\mathbf{R}}_j^0$ compatible with $\text{tc}(\mathcal{B}_j)$
end

for $m \leftarrow 1$ **to** M **do**
  Gibbs step: update $\tau_1, \ldots, \tau_C$
  compute: $n_c = \sum_{j=1}^N 1_c(z_{j,m-1})$, for $c = 1, \ldots, C$
  sample: $\tau_1, \ldots, \tau_C \sim \mathcal{D}(\psi + n_1, \ldots, \psi + n_C)$

  for $c \leftarrow 1$ **to** C **do**
    M-H step: update $\boldsymbol{\rho}_c$
    sample: $\boldsymbol{\rho}_c' \sim \text{L\&S}(\boldsymbol{\rho}_{c,m-1}, L)$ and $u \sim \mathcal{U}(0,1)$
    compute: $ratio \leftarrow$ equation (6) with $\boldsymbol{\rho} \leftarrow \boldsymbol{\rho}_{c,m-1}$ and $\alpha \leftarrow \alpha_{c,m-1}$, and where the sum is over $\{j : z_{j,m-1} = c\}$
    if $u < ratio$ **then** $\boldsymbol{\rho}_{c,m} \leftarrow \boldsymbol{\rho}_c'$
    else $\boldsymbol{\rho}_{c,m} \leftarrow \boldsymbol{\rho}_{c,m-1}$

    if $m \mod \alpha_{jump} = 0$ **then M-H step: update** α_c
    sample: $\alpha_c' \sim \mathcal{N}(\alpha_{c,m-1}, \sigma_\alpha^2)$ and $u \sim \mathcal{U}(0,1)$
    compute: $ratio \leftarrow$ equation (8) with $\boldsymbol{\rho} \leftarrow \boldsymbol{\rho}_{c,m}$ and $\alpha \leftarrow \alpha_{c,m-1}$, and where the sum is over $\{j : z_{j,m-1} = c\}$
    if $u < ratio$ **then** $\alpha_{c,m} \leftarrow \alpha_c'$
    else $\alpha_{c,m} \leftarrow \alpha_{c,m-1}$
  end

  Gibbs step: update $z_1, \ldots, z_N$
  for $j \leftarrow 1$ **to** N **do**
    foreach $c \leftarrow 1$ **to** C **do** compute cluster assignment probabilities:
      $p_{cj} = \frac{\tau_{c,m}}{Z_n(\alpha_{c,m})} \exp\left[\frac{-\alpha_{c,m}}{n} d(\tilde{\mathbf{R}}_j^{m-1}, \boldsymbol{\rho}_{c,m})\right]$
    sample: $z_{j,m} \sim \mathcal{M}(p_{1j}, \ldots, p_{Cj})$
  end

  M-H step: update $\tilde{\mathbf{R}}_1, \ldots, \tilde{\mathbf{R}}_N$**:**
  for $j \leftarrow 1$ **to** N **do**
    if $\{\mathcal{S}_1, \ldots, \mathcal{S}_N\}$ *among inputs* **then** sample: $\tilde{\mathbf{R}}_j'$ in $\mathcal{S}_j$ from the leap-and-shift distribution centered at $\tilde{\mathbf{R}}_j^{m-1}$
    else sample: $\tilde{\mathbf{R}}_j'$ from the leap-and-shift distribution centered at $\tilde{\mathbf{R}}_j^{m-1}$ and compatible with $\text{tc}(\mathcal{B}_j)$
    compute: $ratio \leftarrow$ equation (21) with $\boldsymbol{\rho} \leftarrow \boldsymbol{\rho}_{z_{j,m},m}$, $\alpha \leftarrow \alpha_{z_{j,m},m}$ and $\tilde{\mathbf{R}}_j \leftarrow \tilde{\mathbf{R}}_j^{m-1}$
    sample: $u \sim \mathcal{U}(0,1)$
    if $u < ratio$ **then** $\tilde{\mathbf{R}}_j^m \leftarrow \tilde{\mathbf{R}}_j'$
    else $\tilde{\mathbf{R}}_j^m \leftarrow \tilde{\mathbf{R}}_j^{m-1}$
  end
end

Appendix C. Sample from the Mallows Model

We here explain our proposed procedure to sample rankings from the Mallows model.

To sample full rankings $\mathbf{R}_1, ..., \mathbf{R}_N \sim \text{Mallows}(\boldsymbol{\rho}, \alpha)$, we use the following scheme (sketched in Algorithm 5). We run a basic Metropolis-Hastings algorithm with fixed consensus $\boldsymbol{\rho} \in \mathcal{P}_n$, $\alpha > 0$ and with a given distance measure, $d(\cdot,\cdot)$, until convergence. Once convergence is achieved, we continue sampling, and store the so obtained rankings at regular intervals (large enough to achieve independence) until we have reached the desired data dimension.

In case of heterogeneous rankings, we sample from Algorithm 6. As inputs, we give the number of clusters C, the fixed consensuses $\boldsymbol{\rho}_1, ..., \boldsymbol{\rho}_C$, the fixed $\alpha_1, ..., \alpha_C$, the hyper-

Algorithm 5: MCMC Sampler for full rankings

input : $\boldsymbol{\rho}$, α, d, N, L
output: $\mathbf{R}_1, ..., \mathbf{R}_N$
Initialization of the MCMC: randomly generate $\mathbf{R}_{1,0}, ..., \mathbf{R}_{N,0}$
for $m \leftarrow 1$ **to** M **do**
  for $j \leftarrow 1$ **to** N **do**
    sample $\mathbf{R}_j' \sim \text{L\&S}(\mathbf{R}_{j,m-1}, L)$
    compute: $ratio = \frac{P_L(\mathbf{R}_j|\mathbf{R}_j')}{P_L(\mathbf{R}_j'|\mathbf{R}_j)} \exp\left\{-\frac{\alpha}{n}\sum_{j=1}^N \left[d(\mathbf{R}_j', \boldsymbol{\rho}) - d(\mathbf{R}_j, \boldsymbol{\rho})\right]\right\}$ with $\mathbf{R}_j \leftarrow \mathbf{R}_{j,m-1}$
    sample: $u \sim \mathcal{U}(0,1)$
    if $u < ratio$ **then**
      $\mathbf{R}_{j,m} \leftarrow \mathbf{R}_j'$
    else
      $\mathbf{R}_{j,m} \leftarrow \mathbf{R}_{j,m-1}$
    end
  end
end

parameter $\boldsymbol{\psi} = (\psi_1, ..., \psi_C)$ of the Dirichlet density over the proportion of assessors in the clusters, and $d(\cdot,\cdot)$. The algorithm then returns the rankings $\mathbf{R}_1, ..., \mathbf{R}_N$, sampled from a Mixture of Mallows models, as well as the the cluster assignments $z_1, ..., z_N$.

Algorithm 6: MCMC Sampler for full rankings with clusters

input : C, $\boldsymbol{\rho}_{1:C}$, $\alpha_{1:C}$, $\boldsymbol{\psi}$, d, N, L
output: $\mathbf{R}_1, ..., \mathbf{R}_N$ and $z_1, ..., z_N$
Initialization of the MCMC: randomly generate $\mathbf{R}_{1,0}, ..., \mathbf{R}_{N,0}$
randomly generate $\tau_1, ..., \tau_C \sim \text{Dir}(\boldsymbol{\psi})$
randomly generate $z_1, ..., z_N \sim \text{Mn}(1, \tau_1, ..., \tau_C)$
for $m \leftarrow 1$ **to** M **do**
  for $c \leftarrow 1$ **to** C **do**
    compute: $N_c = \sum_{j=1}^N 1\!\!1_c(z_j)$,
    sample N_c ranks with Algortihm 5
  end
end

For generating top-k rankings, we simply generate $\mathbf{R}_1, ..., \mathbf{R}_N$ with Algorithm 5, and then keep only the top-k items. In case of clusters, we do the same as above, but starting with Algorithm 6.

Finally, to sample the sets of pairwise comparisons $\mathcal{B}_1, ..., \mathcal{B}_N$, we first generate $\mathbf{R}_1, ..., \mathbf{R}_N$ with Algorithm 5. We then select the number $T_1, ..., T_N$ of pairwise comparisons that each assessor will evaluate[2]. Finally, given $\mathbf{R}_1, ..., \mathbf{R}_N$ and $T_1, ..., T_N$, we randomly sample without replacement T_j pairs (for each assessor $j = 1, \ldots, N$) from the collection of all possible $n(n-1)/2$ pairs, and obtain pairwise preferences by ordering all pairs according to $\mathbf{R}_j$. For generating pairwise comparisons with clusters, we follow the previous procedure, but starting with Algorithm 6.

2. Here it is possible to choose the same number of comparisons $T_j = T \leq n(n-1)/2$, $\forall j = 1, ..., N$, but also to have a different number of pairs per assessor. In this paper, for a given mean parameter λ_T, we independently sample $T_1, ..., T_N \sim \text{TruncPoiss}(\lambda_T, n(n-1)/2)$.

References

J. A. Aledo, J. A. Gàmez, and D. Molina. Tackling the rank aggregation problem with evolutionary algorithms. *Applied Mathematics and Computation*, 222:632 – 644, 2013.

A. Ali and M. Meilă. Experiments with Kemeny ranking: What works when? *Mathematical Social Sciences*, 64(1):28 – 40, 2012.

M. Alvo and P. L. H. Yu. *Statistical Methods for Ranking Data*. Frontiers in Probability and the Statistical Sciences. Springer, New York, NY, USA, 2014.

C. Andrieu and G. O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725, 2009.

D. Asfaw, V. Vitelli, Ø. Sørensen, E. Arjas, and A. Frigessi. Time-varying rankings with the Bayesian Mallows model. *Stat*, 6(1):14–30, 2017.

J. Bartholdi, C. Tovey, and M. A. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6(2):157–165, 1989.

M.A. Beaumont. Estimation of population growth or decline in genetically monitored populations. *Genetics*, 164(3):1139–1160, 2003.

S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1):107–117, 1998.

L. M. Busse, P. Orbanz, and J. M. Buhmann. Cluster analysis of heterogeneous rank data. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 113–120, New York, NY, USA, 2007. ACM.

F. Caron and Y. W. Teh. Bayesian nonparametric models for ranked data. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1520–1528. Curran Associates, Inc., 2012.

F. Caron, Y. W. Teh, and T. B. Murphy. Bayesian nonparametric Plackett-Luce models for the analysis of preferences for college degree programmes. *The Annals of Applied Statistics*, 8(2):1145–1181, 2014.

G. Celeux, M. Hurn, and C. Robert. Computational and inferential difficulties with mixture posterior distribution. *Journal of the American Statistical Association*, 95(451):957–970, 2000.

G. Celeux, F. Forbes, C. P. Robert, and D. M. Titterington. Deviance information criteria for missing data models. *Bayesian Analysis*, 1(4):651–674, 2006.

D. E. Critchlow. *Metric methods for analyzing partially ranked data*, volume 34. Springer Science and Business Media, 2012.

G. Csardi and T. Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006. URL `http://igraph.sf.net`.

A. P. Dawid. The well-calibrated Bayesian. *Journal of the American Statistical Association*, 77(379):605–610, 1982.

J. C. de Borda. Mémoire sur les élections au scrutin, histoire de l'académie royale des sciences. *Paris, France*, 1781.

R. P. DeConde, S. Hawley, S. Falcon, N. Clegg, B. Knudsen, and R. Etzioni. Combining results of microarray experiments: A rank aggregation approach. *Statistical Applications in Genetics and Molecular Biology*, 5(1):Article 15, 2006.

K. Deng, S. Han, K. J. Li, and J. S. Liu. Bayesian aggregation of order-based rank data. *Journal of the American Statistical Association*, 109(507):1023–1039, 2014.

S. M. Dhanasekaran, T. R. Barrette, D. Ghosh, R. Shah, S. Varambally, K. Kurachi, K. J. Pienta, M. A. Rubin, and A. M. Chinnaiyan. Delineation of prognostic biomarkers in prostate cancer. *Nature*, 412:822–826, 2001.

P. Diaconis. *Group representations in probability and statistics*, volume 11 of *Lecture Notes - Monograph Series*. Institute of Mathematical Statistics, Hayward, CA, USA, 1988.

J. P. Doignon, A. Pekeč, and M. Regenwetter. The repeated insertion model for rankings: Missing link between two subset choice models. *Psychometrika*, 69(1):33–54, 2004.

D. Firth and H. L. Turner. Bradley-Terry models in R: the BradleyTerry2 package. *Journal of Statistical Software*, 48(9), 2012.

M. A. Fligner and J. S. Verducci. Distance based ranking models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 48(3):359–369, 1986.

B. Francis, R. Dittrich, and R. Hatzinger. Modeling heterogeneity in ranked responses by nonparametric maximum likelihood: how do Europeans get their scientific knowledge? *The Annals of Applied Statistics*, 4(4):2181–2202, 2010.

J. Fürnkranz and E. Hüllermeier. *Preference learning: An introduction*. Springer, 2010.

P. Gopalan, T.S. Jayram, R. Krauthgamer, and R. Kumar. Approximating the longest increasing sequence and distance from sortedness in a data stream. Research Microsoft Publications, 2006.

I. C. Gormley and T. B. Murphy. Analysis of Irish third-level college applications data. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 169(2):361–379, 2006.

J. Guiver and E. Snelson. Bayesian inference for Plackett-Luce ranking models. In *proceedings of the 26th annual international conference on machine learning*, pages 377–384. ACM, 2009.

D. R. Hunter. MM algorithms for generalized Bradley-Terry models. *The Annals of Statistics*, 32(1):384–406, 2004.

E. Irurozki, B. Calvo, and A. Lozano. Sampling and learning the Mallows and generalized Mallows models under the Hamming distance. *Bernoulli (submitted)*, 2014.

E. Irurozki, B. Calvo, and A. Lozano. PerMallows: An R package for Mallows and generalized Mallows models. *Journal of Statistical Software*, 71, 2016a.

E. Irurozki, B. Calvo, and A. Lozano. Sampling and learning the Mallows and generalized Mallows models under the Cayley distance. *Methodology and Computing in Applied Probability*, 2016b.

J. Jacques and C. Biernacki. Model-based clustering for multivariate partial ranking data. *Journal of Statistical Planning and Inference*, 149:201–217, 2014.

J. Jacques, Q. Grimonprez, and C. Biernacki. Rankcluster: An R package for clustering multivariate partial rankings. *The R Journal*, 6(1):10, 2014.

A. Jasra, C.C. Holmes, and D.A. Stephens. Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling. *Statistical Science*, 20(1):50–67, 2005.

T. Kamishima. Nantonac collaborative filtering: Recommendation based on order responses. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 583–588, New York, NY, USA, 2003. ACM.

M. E. Khan, Y. J. Ko, and M. Seeger. Scalable collaborative Bayesian preference learning. In *AISTATS*, volume 14, pages 475–483, 2014.

G. Lebanon and Y. Mao. Non-parametric modeling of partially ranked data. *Journal of Machine Learning Research*, 9:2401–2429, 2008.

P. H. Lee and P. L. H. Yu. Mixtures of weighted distance-based models for ranking data with applications in political studies. *Computational Statistics & Data Analysis*, 56(8): 2486–2500, 2012.

S. Lin and J. Ding. Integration of ranked lists via cross entropy Monte Carlo with applications to mRNA and microRNA studies. *Biometrics*, 65(1):9–18, 2009.

R. Little. Calibrated Bayes, for statistics in general, and missing data in particular. *Statistical Science*, 26(2):162–174, 2011.

T. Lu and C. Boutilier. Effective sampling and learning for Mallows models with pairwise-preference data. *Journal of Machine Learning Research*, 15:3783–3829, 2014.

R.D. Luce. *Individual choice behavior: A theoretical analysis*. Wiley, New York, NY, USA, 1959.

J. Luo, D. J. Duggan, Y. Chen, J. Sauvageot, C. M. Ewing, M. L. Bittner, J. M. Trent, and W. B. Isaacs. Human prostate cancer and benign prostatic hyperplasia: Molecular dissection by gene expression profiling. *Cancer Research*, 61(12):4683–4688, 2001.

C. L. Mallows. Non-null ranking models. I. *Biometrika*, 44(1/2):114–130, 1957.

J. I. Marden. *Analyzing and Modeling Rank Data*, volume 64 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, Cambridge, MA, USA, 1995.

M. Meilă and L. Bao. An exponential model for infinite rankings. *Journal of Machine Learning Research*, 11:3481–3518, 2010.

M. Meilă and H. Chen. Dirichlet process mixtures of generalized Mallows models. In *Proceedings of the Twenty-Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-10)*, pages 358–367, Corvallis, OR, USA, 2010. AUAI Press.

D. Meyer and K. Hornik. Generalized and customizable sets in R. *Journal of Statistical Software*, 31(2):1–27, 2009.

D. Meyer and K. Hornik. relations: Data structures and algorithms for relations. R package version 0.6-3, 2014. URL `http://CRAN.R-project.org/package=relations`.

S. Mukherjee. Estimation in exponential families on permutations. *The Annals of Statistics*, 44(2):853–875, 2016.

T. B. Murphy and D. Martin. Mixtures of distance-based models for ranking data. *Computational Statistics & Data Analysis*, 41(3–4):645 – 655, 2003.

I. Murray, Z. Ghahramani, and D. MacKay. MCMC for doubly-intractable distributions. *arXiv preprint arXiv:1206.6848*, 2012.

P. Papastamoulis. label.switching: An R package for dealing with the label switching problem in MCMC outputs. arXiv:1503.02271v1, 2015.

V. Pihur, S. Datta, and S. Datta. RankAggreg, an R package for weighted rank aggregation. *BMC bioinformatics*, 10(1):62, 2009.

R. L. Plackett. The analysis of permutations. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 24(2):193–202, 1975.

M. Regenwetter, J. C. Falmagne, and B. Grofman. A stochastic model of preference change and its application to 1992 presidential election panel data. *Psychological Review*, 106 (2):362–384, 1999.

M. G. Schimek, E. Budinská, K. G. Kugler, V. Švendová, J. Ding, and S. Lin. TopKLists: a comprehensive R package for statistical inference, stochastic aggregation, and visualization of multiple omics ranked lists. *Statistical Applications in Genetics and Molecular Biology*, 14(3):311–316, 2015.

D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie, E. S. Lander, M. Loda, P. W. Kantoff, T. R. Golub, and W. R. Sellers. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1(2):203 – 209, 2002. ISSN 1535-6108.

N. J. A. Sloane. The On-Line Encyclopedia of Integer Sequences, 2017. URL `http://oeis.org`.

M. Sun, G. Lebanon, and P. Kidwell. Estimating probabilities in recommendation systems. *Journal of the Royal Statistical Society, Series C*, 61(3):471–492, 2012.

L. True, I. Coleman, S. Hawley, C.Y. Huang, D. Gifford, R. Coleman, T. M. Beer, E. Gelmann, M. Datta, E. Mostaghel, B. Knudsen, P. Lange, R. Vessella, D. Lin, L. Hood, and P. S. Nelson. A molecular correlate to the Gleason grading system for prostate adenocarcinoma. *Proceedings of the National Academy of Sciences*, 103(29):10991–10996, 2006.

M. N. Volkovs and R. S. Zemel. New learning methods for supervised and unsupervised preference aggregation. *Journal of Machine Learning Research*, 15:1135–1176, 2014.

J. B. Welsh, L. M. Sapinoso, A. I. Su, S. G. Kern, J. Wang-Rodriguez, C. A. Moskaluk, H. F. Frierson, and G. M. Hampton. Analysis of gene expression identifies candidate markers and pharmacological targets in prostate cancer. *Cancer Research*, 61(16):5974–5978, 2001.

Journal of Machine Learning Research 18 (2018) 1-40 Submitted 9/15; Revised 2/17; Published 4/18

Robust Topological Inference: Distance To a Measure and Kernel Distance

Frédéric Chazal FREDERIC.CHAZAL@INRIA.FR
Inria Saclay - Ile-de-France
Alan Turing Bldg, Office 2043
1 rue Honoré d'Estienne d'Orves
91120 Palaiseau, FRANCE

Brittany Fasy BRITTANY@CS.MONTANA.EDU
Computer Science Department
Montana State University
357 EPS Building
Montana State University
Bozeman, MT 59717

Fabrizio Lecci FABRZIO.LECCI@GMAIL.COM
New York, NY

Bertrand Michel BERTRAND.MICHEL@EC-NANTES.FR
Ecole Centrale de Nantes
Laboratoire de mathématiques Jean Leray
1 Rue de La Noe
44300 Nantes FRANCE

Alessandro Rinaldo ARINALDO@CMU.EDU
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213

Larry Wasserman LARRY@CMU.EDU
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213

Editor: Mikhail Belkin

Abstract

Let P be a distribution with support S. The salient features of S can be quantified with persistent homology, which summarizes topological features of the sublevel sets of the distance function (the distance of any point x to S). Given a sample from P we can infer the persistent homology using an empirical version of the distance function. However, the empirical distance function is highly non-robust to noise and outliers. Even one outlier is deadly. The distance-to-a-measure (DTM), introduced by Chazal et al. (2011), and the kernel distance, introduced by Phillips et al. (2014), are smooth functions that provide useful topological information but are robust to noise and outliers. Chazal et al. (2015) derived concentration bounds for DTM. Building on these results, we derive limiting distributions and confidence sets, and we propose a method for choosing tuning parameters.

Keywords: Topological data analysis, persistent homology, RKHS.

1. Introduction

Figure 1 shows three complex point clouds, based on a model used for simulating cosmology data. Visually, the three samples look very similar. Below the data plots are the persistence diagrams, which are summaries of topological features defined in Section 2. The persistence diagrams make it clearer that the third data set is from a different data generating process than the first two.

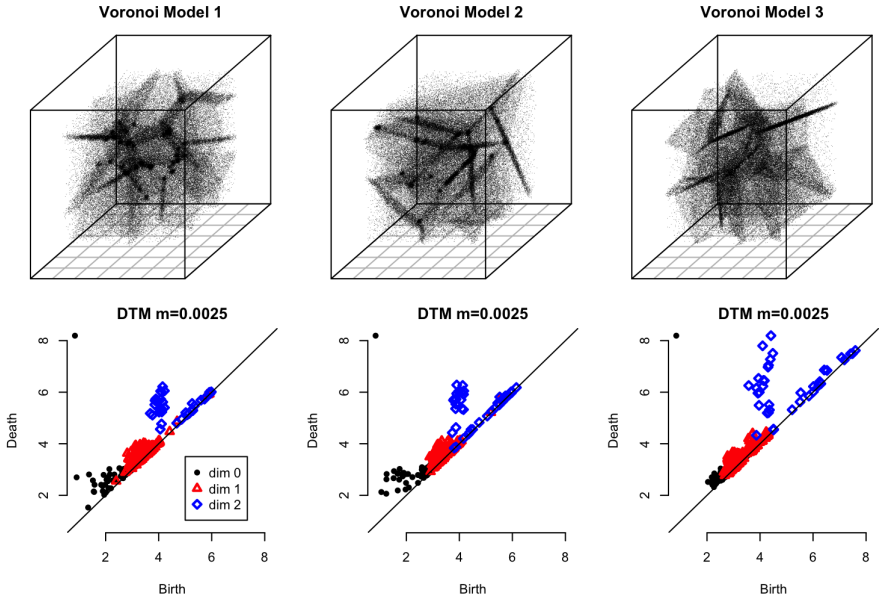


Figure 1: The first two datasets come from the same data generating mechanism. In the third one, the particles are more concentrated around the walls of the Voronoi cells. Although the difference is not clear from the scatterplots, it is evident from the persistence diagrams of the sublevel sets of the distance-to-measure functions. See Example 3 for more details on the Voronoi Models.

This is an example of how topological features can summarize structure in point clouds. The field of topological data analysis (TDA) is concerned with defining such topological features; see Carlsson (2009). When performing TDA, it is important to use topological measures that are robust to noise. This paper explores some of these robust topological measures.

Let P be a distribution with compact support $S \subset \mathbb{R}^d$. One way to describe the shape of S is by using homology. Roughly speaking, the homology of S measures the topological features of S, such as the connected components, the holes, and the voids. A more nuanced way to describe the shape of S is using persistent homology, which is a multiscale version of homology. To describe persistent homology, we begin with the distance function $\Delta_S \colon \mathbb{R}^d \to \mathbb{R}$ for S which is defined by

$$\Delta_S(x) = \inf_{y \in S} \|x - y\|. \tag{1}$$

The sublevel sets $L_t = \{x : \ \Delta_S(x) \leq t\}$ provide multiscale topological information about S. As t varies from zero to ∞, topological features — connected components, loops, voids — are born and die. Persistent homology quantifies the evolution of these topological features as a function of t. See Figure 2. Each point on the persistence diagram represents the birth and death time of a topological feature.

Given a sample $X_1, \ldots, X_n \sim P$, the empirical distance function is defined by

$$\widehat{\Delta}(x) = \min_{X_i} \|x - X_i\|. \tag{2}$$

If P is supported on S, and has a density bounded away from zero and infinity, then $\widehat{\Delta}$ is a consistent estimator of Δ_S, i.e., $\sup_x |\widehat{\Delta}(x) - \Delta_S(x)| \xrightarrow{P} 0$. However, if there are outliers, or noise, then $\widehat{\Delta}(x)$ is no longer consistent. Figure 3 (bottom) shows that a few outliers completely change the distance function. In the language of robust statistics, the empirical distance function has breakdown point zero.

A more robust approach is to estimate the persistent homology of the super-level sets of the density p of P. As long as P is concentrated near S, we expect the level sets of p to provide useful topological information about S. Specifically, some level sets of p are homotopic to S under weak conditions, and this implies that we can estimate the homology of S. Note that, in this case, we are using the persistent homology of the super-level sets of p, to estimate the homology of S. This is the approach suggested by Bubenik (2015), Fasy et al. (2014b) and Bobrowski et al. (2014). A related idea is to use persistent homology based on a kernel distance (Phillips et al., 2014). In fact, the sublevel sets of the kernel distance are a rescaling of the super-level sets of p, so these two ideas are essentially equivalent. We discuss this approach in Section 5.

A different approach, more closely related to the distance function, but robust to noise, is to use the *distance-to-a-measure (DTM)*, $\delta \equiv \delta_{P,m}$, from Chazal et al. (2011); see Section 2. An estimate $\widehat{\delta}$ of δ is obtained by replacing the true probability measure with the empirical probability measure P_n, or with a deconvolved version of the observed measure Caillerie et al. (2011). One then constructs a persistence diagram based on the sub-level sets of the DTM. See Figure 1. This approach is aimed at estimating the persistent homology of S. (The DTM also suggests new approaches to density estimation; see Biau et al. (2011).)

The density estimation approach and the DTM are both trying to probe the topology of S. But the former is using persistent homology to estimate the homology of S, while the DTM is directly trying to estimate the persistent homology of distance function of S. We discuss this point in detail in Section 9.1.

In this paper, we explore some statistical properties of these methods. In particular:

1. We show that $\sqrt{n}(\widehat{\delta}^2(x) - \delta^2(x))$ converges to a Gaussian process. (Theorem 5).
2. We show that the bootstrap provides asymptotically valid confidence bands for δ. This allows us to identify significant topological features. (Theorem 19).
3. We find the limiting distribution of a key topological quantity called the bottleneck distance. (Section 4.1).
4. We also show that, under additional assumptions, there is another version of the bootstrap—which we call the bottleneck bootstrap—that provides more precise inferences. (Section 6).
5. We show similar results for the kernel distance. (Section 5).
6. We propose a method for choosing the tuning parameter m for DTM and the bandwidth h for the kernel distance. (Section 7.1).
7. We show that the DTM and the kernel density estimator (KDE) both suffer from boundary bias and we suggest a method for reducing the bias. (Section 7.2).

Notation. $B(x, \epsilon)$ is a Euclidean ball of radius ϵ, centered at x. We define $A \oplus \epsilon = \bigcup_{x \in A} B(x, \epsilon)$, the union of ϵ-balls centered at points in A. If x is a vector then $||x||_\infty = \max_j |x_j|$. Similarly, if f is a real-valued function then $||f||_\infty = \sup_x |f(x)|$. We write $X_n \rightsquigarrow X$ to mean that X_n converges in distribution to X, and we use symbols like $c, C, \ldots,$ as generic positive constants.

Remark: The computing for the examples in this paper were done using the R package **TDA**. See Fasy et al. (2014a). The package can be downloaded from `http://cran.r-project.org/web/packages/TDA/index.html`.

Remark: In this paper, we discuss the DTM which uses a smoothing parameter m and the kernel density estimator which uses a smoothing bandwidth h. Unlike in traditional function estimation, we do not send these parameters to zero as n increases. In TDA, the topological features created with a fixed smoothing parameter are of interest. Thus, all the theory in this paper treats the smoothing parameters as being bounded away from 0. See also Section 4.4 in Fasy et al. (2014b). In Section 7.1, we discuss the choice of these smoothing parameters.

2. Background

In this section, we define several distance functions and distance-like functions, and we introduce the relevant concepts from computational topology. For more detail, we refer the reader to Edelsbrunner and Harer (2010).

2.1 Distance Functions and Persistent Homology

Let $S \subset \mathbb{R}^d$ be a compact set. The *homology* of S characterizes certain topological features of S, such as its connected components, holes, and voids. *Persistent homology* is a multiscale version of homology. Recall that the distance function Δ_S for S is

$$\Delta_S(x) = \inf_{y \in S} \|x - y\|. \tag{3}$$

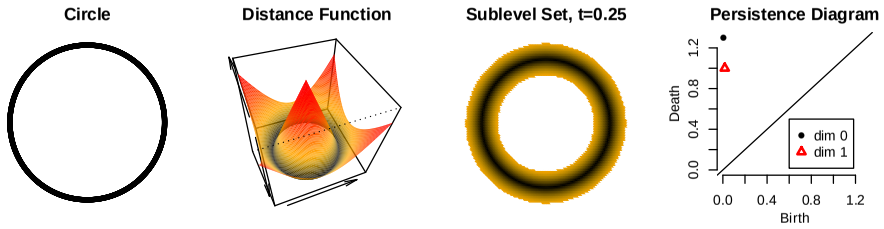


Figure 2: The left plot shows a one-dimensional curve. The second plot is the distance function. The third plot shows a typical sublevel set of the distance function. The fourth plot is the persistence diagram which shows the birth and death times of loops (triangles) and connected components (points) of the sublevel sets.

Let $L_t = \{x : \ \Delta_S(x) \leq t\}$. We will refer to the parameter t as "time."

Given the nested family of the sublevel sets of Δ_S, the topology of L_t changes as t increases: new connected components can appear, existing connected components can merge, cycles and cavities can appear or be filled, etc. Persistent homology tracks these changes, identifies *features* and associates an *interval* or *lifetime* (from t_{birth} to t_{death}) to them. For instance, a connected component is a feature that is born at the smallest t such that the component is present in L_t, and dies when it merges with an older connected component. Intuitively, the longer a feature persists, the more relevant it is.

A feature, or more precisely its lifetime, can be represented as a segment whose extremities have abscissae t_{birth} and t_{death}; the set of these segments is called the *barcode* of Δ_S. An interval can also be represented as a point in the plane with coordinates $(u, v) = (t_{\text{birth}}, t_{\text{death}})$. The set of points (with multiplicity) representing the intervals is called the *persistence diagram* of Δ_S. Note that the diagram is entirely contained in the half-plane above the diagonal defined by $u = v$, since death always occurs after birth. This diagram is well-defined for any compact set S (Chazal et al. (2012), Theorem 2.22). The most persistent features (supposedly the most important) are those represented by the points furthest from the diagonal in the diagram, whereas points close to the diagonal can be interpreted as (topological) noise.

Figure 2 shows a simple example. Here, the points on the circle are regarded as a subset of $\mathbb{R}^2$. At time zero, there is one connected component and one loop. As t increases, the loop dies.

Let S_1 and S_2 be compact sets with distance functions Δ_1 and Δ_2 and diagrams D_1 and D_2. The bottleneck distance between D_1 and D_2 is defined by

$$W_\infty(D_1, D_2) = \min_{g:\ D_1 \to D_2} \sup_{z \in D_1} \|z - g(z)\|_\infty, \tag{4}$$

where the minimum is over all bijections between D_1 and D_2. In words, the bottleneck distance is the maximum distance between the points of the two diagrams, after minimizing over all possible pairings of the points (including the points on the diagonals).

A fundamental property of persistence diagrams is their *stability*. According to the Persistence Stability Theorem (Cohen-Steiner et al. (2005); Chazal et al. (2012))

$$W_\infty(D_1, D_2) \leq ||\Delta_1 - \Delta_2||_\infty = H(S_1, S_2). \tag{5}$$

Here, H is the Hausdorff distance, namely,

$$H(A, B) = \inf\Big\{\epsilon : \ A \subset B \oplus \epsilon \text{ and } B \subset A \oplus \epsilon\Big\},$$

where we recall that $A \oplus \epsilon = \bigcup_{x \in A} B(x, \epsilon)$. More generally, the definition of persistence diagrams and the above stability theorem do not restrict to distance functions but also extend to families of sublevel sets (resp. upper-level sets) of functions defined on $\mathbb{R}^d$ under very weak assumption. We refer the reader to Edelsbrunner and Harer (2010); Chazal et al. (2009, 2012) for a detailed exposition of the theory.

Given a sample $X_1, \ldots, X_n \sim P$, the empirical distance function is defined by

$$\widehat{\Delta}(x) = \min_{X_i} \|x - X_i\|. \tag{6}$$

Lemma 1 (Lemma 4 in Fasy et al., 2014b) *Suppose that P is supported on S, and has a density bounded away from zero and infinity. Then*

$$\sup_x |\widehat{\Delta}(x) - \Delta_S(x)| \xrightarrow{P} 0.$$

See also Cuevas and Rodríguez-Casal (2004). The previous lemma justifies using $\widehat{\Delta}$ to estimate the persistent homology of sublevel sets of Δ_S. In fact, the sublevel sets of $\widehat{\Delta}$ are just unions of balls around the observed data. That is,

$$L_t = \big\{x : \ \widehat{\Delta}(x) \leq t\big\} = \bigcup_{i=1}^n B(X_i, t).$$

The persistent homology of the union of the balls as t increases may be computed by creating a combinatorial representation (called a Cech complex) of the union of balls, and then applying basic operations from linear algebra (Edelsbrunner and Harer, 2010, Sections VI.2 and VII.1).

However, as soon as there is noise or outliers, the empirical distance function becomes useless, as illustrated in Figure 3. More specifically, suppose that

$$P = \pi R + (1 - \pi)(Q \star \Phi_\sigma), \tag{7}$$

where $\pi \in [0, 1]$, R is an outlier distribution (such as a uniform on a large set), Q is supported on S, $\star$ denotes convolution, and Φ_σ is a compactly supported noise distribution with scale parameter σ.

Recovering the persistent homology of Δ_S exactly (or even the homology of S) is not possible in general since the problem is under-identified. But we would still like to find a function that is similar to the distance function for S. The empirical distance function fails miserably even when π and σ are small. Instead, we now turn to the DTM.

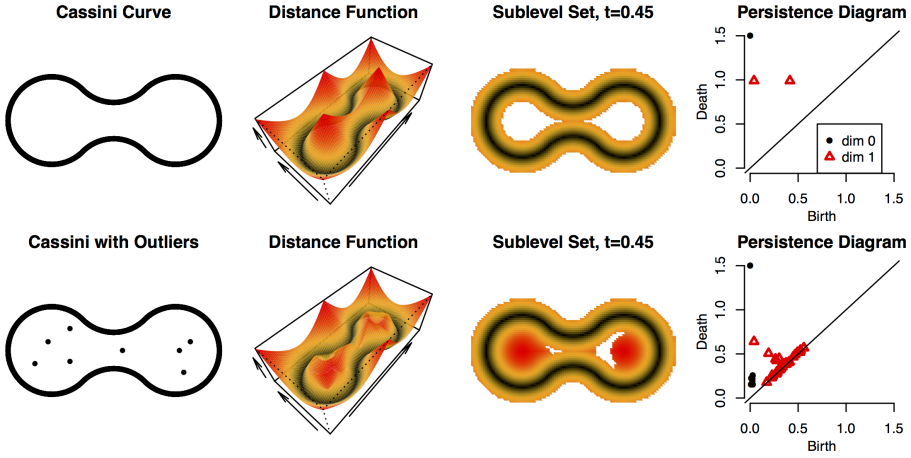


Figure 3: Top: data on the Cassini curve, the distance function $\widehat{\Delta}$, a typical sublevel set $\{x : \widehat{\Delta}(x) \leq t\}$ and the resulting persistence diagram. Bottom: the effect of adding a few outliers. Note that the distance function and persistence diagram are dramatically different.

2.2 Distance to a Measure

Given a probability measure P, for $0 < m < 1$, the *distance-to-measure (DTM)* at resolution m (Chazal et al., 2011) is defined by

$$\delta(x) \equiv \delta_{P,m}(x) = \sqrt{\frac{1}{m}\int_0^m (G_x^{-1}(u))^2 du}, \tag{8}$$

where $G_x(t) = P(\|X - x\| \leq t)$. Alternatively, the DTM can be defined using the cdf of the squared distances, as in the following lemma:

Lemma 2 (Chazal et al., 2015) *Let $F_x(t) = P(\|X - x\|^2 \leq t)$. Then*

$$\delta_{P,m}^2(x) = \frac{1}{m}\int_0^m F_x^{-1}(u) du.$$

Proof For any $0 < u < 1$,

$$\begin{aligned}\left[G_x^{-1}(u)\right]^2 &= \inf\left\{t^2 : G_x(t) \geq u\right\} = \inf\left\{t^2 : P(\|X - x\| \leq t) \geq u\right\} \\ &= \inf\left\{t : P(\|X - x\|^2 \leq t) \geq u\right\} = \inf\left\{t : F_x(t) \geq u\right\} = F_x^{-1}(u).\end{aligned}$$

Therefore

$$\delta_{P,m}^2(x) = \frac{1}{m}\int_0^m (G_x^{-1}(u))^2 du = \frac{1}{m}\int_0^m F_x^{-1}(u) du.$$

■

Given a sample $X_1, \ldots, X_n \sim P$, let P_n be the probability measure that puts mass $1/n$ on each X_i. It is easy to see that the distance to the measure P_n at resolution m is

$$\widehat{\delta}^2(x) \equiv \delta_{P_n,m}^2(x) = \frac{1}{k}\sum_{X_i \in N_k(x)} \|X_i - x\|^2, \tag{9}$$

where $k = \lceil mn \rceil$ and $N_k(x)$ is the set containing the k nearest neighbors of x among $X_1, \ldots, X_n$. We will use $\widehat{\delta}$ to estimate δ.

Now we summarize some important properties of the DTM, all of which are proved in Chazal et al. (2011) and Buchet et al. (2013). First, recall that the *Wasserstein distance of order p* between two probability measures P and Q is given by

$$W_p(P, Q) = \inf_J \left(\int \|x - y\|^p \, dJ(x, y)\right)^{1/p}, \tag{10}$$

where the infimum is over all joint distributions J for (X, Y) such that $X \sim P$ and $Y \sim Q$. We say that P satisfies the (a, b)-*condition* if there exist $a, b > 0$ such that, for every x in the support of P and every $\epsilon > 0$,

$$P\Big(B(x, \epsilon)\Big) \geq a\epsilon^b. \tag{11}$$

This means that the support does not have long, thin components.

Theorem 3 (Properties of DTM) *The following properties hold:*

1. *The distance to measure is 1-Lipschitz: for any probability measure P on $\mathbb{R}^d$ and any $(x, x') \in \mathbb{R}^d$,*
$$|\delta_{P,m}(x) - \delta_{P,m}(x')| \leq \|x - x'\|.$$
2. *If Q satisfies (11) and is supported on a compact set S, then*
$$\sup_x |\delta_{Q,m}(x) - \Delta_S(x)| \leq a^{-1/b} m^{1/b}. \tag{12}$$
In particular, $\sup_x |\delta_{Q,m}(x) - \Delta_S(x)| \to 0$ as $m \to 0$.
3. *If P and Q are two distributions, then*
$$\sup_x |\delta_{P,m}(x) - \delta_{Q,m}(x)| \leq \frac{1}{\sqrt{m}} W_2(P, Q). \tag{13}$$
4. *If Q satisfies (11) and is supported on a compact set S and P is another distribution (not necessarily supported on S), then*
$$\sup_x |\delta_{P,m}(x) - \Delta_S(x)| \leq a^{-1/b} m^{1/b} + \frac{1}{\sqrt{m}} W_2(P, Q) \tag{14}$$
Hence, if $m \asymp W_2(P, Q)^{2b/(2+b)}$, then $\sup_x |\delta_{P,m}(x) - \Delta_S(x)| = O(W_2(P, Q)^{2/(2+b)})$.

5. Let D_P be the diagram from $\delta_{P,m}$ and let D_Q be the diagram from $\delta_{Q,m}$, then

$$W_\infty(D_P, D_Q) \leq ||\delta_{P,m} - \delta_{Q,m}||_\infty. \tag{15}$$

For any compact set $A \subset \mathbb{R}^d$, let $r(A)$ denotes the radius of the smallest enclosing ball of A centered at zero:

$$r(A) = \inf\{r > 0 : A \subset B(0,r)\}.$$

We conclude this section by bounding the distance between the diagrams $D_{\delta_{P,m}}$ and D_{Δ_S}.

Lemma 4 (Comparison of Diagrams) *Let $P = \pi R + (1-\pi)(Q \star \Phi_\sigma)$ where Q is supported on S and satisfies (11), R is uniform on a compact set $A \subset \mathbb{R}^d$ and $\Phi_\sigma = N(0, \sigma^2 I)$. Then,*

$$W_\infty\left(D_{\delta_{P,m}}, D_{\Delta_S}\right) \leq a^{-1/b} m^{1/b} + \frac{\pi\sqrt{r(A)^2 + 2r(S)^2 + 2\sigma^2} + \sigma}{\sqrt{m}}.$$

Proof We first apply the stability theorem and parts 4 and 5 of the previous result:

$$W_\infty\left(D_{\delta_{P,m}}, D_{\Delta_S}\right) \leq a^{-1/b} m^{1/b} + \frac{1}{\sqrt{m}} W_2(P,Q).$$

The term $W_2(P,Q)$ can be upper bounded as follows:

$$W_2(P,Q) \leq W_2(P, Q \star \Phi_\sigma) + W_2(Q \star \Phi_\sigma, Q)$$

These two terms can be bounded with simple transport plans. Let Z be a Bernoulli random variable with parameter π. Let X and Y be random variables with distributions R and $Q \star \Phi_\sigma$. We take these three random variables to be independent. Then, the random variable V defined by $V = ZX + (1-Z)Y$ has for distribution the mixture distribution P. By definition of W_2, one has

$$\begin{aligned} W_2^2(P, Q \star \Phi_\sigma) &\leq \mathbb{E}\left(\|V - Y\|^2\right) \\ &\leq \mathbb{E}\left(|Z|^2\right)\mathbb{E}\left(\|X - Y\|^2\right), \end{aligned}$$

by definition of V and by independence of Z and $X - Y$. Next, we have $\mathbb{E}\left(\|X\|^2\right) \leq r(A)^2$ and $\mathbb{E}\left(\|Y\|^2\right) \leq 2[r(S)^2 + \sigma^2]$. Thus

$$W_2^2(P, Q \star \Phi_\sigma) \leq \pi^2\left(r(A)^2 + 2r(S)^2 + 2\sigma^2\right).$$

It can be checked in a similar way that $W_2(Q \star \Phi_\sigma, Q) \leq \sigma$ (see for instance the proof of Proposition 1 in Caillerie et al. (2011)) and the Lemma is proved. ■

Remark: Note that when π and σ are small (and m tends to 0) we see that the diagrams $D_{\delta_{P,m}}$ and D_{Δ_S} are close.

3. Limiting Distribution of the Empirical DTM

In this section, we find the limiting distribution of $\widehat{\delta}$ and we use this to find confidence bands for $\delta(x)$. We start with the pointwise limit.

Let $\delta(x) \equiv \delta_{P,m}(x)$ and $\widehat{\delta}(x) \equiv \delta_{P_n,m}(x)$, as defined in the previous section.

Theorem 5 (Convergence to Normal Distribution) *Let P be some distribution in $\mathbb{R}^d$. For some fixed x, assume that F_x is differentiable at $F_x^{-1}(m)$, for $m \in (0,1)$, with positive derivative $F_x'(F_x^{-1}(m))$. Then we have*

$$\sqrt{n}(\widehat{\delta}^2(x) - \delta^2(x)) \rightsquigarrow N(0, \sigma_x^2), \tag{16}$$

where

$$\sigma_x^2 = \frac{1}{m^2}\int_0^{F_x^{-1}(m)}\int_0^{F_x^{-1}(m)} [F_x(s \wedge t) - F_x(s)F_x(t)]\, ds\, dt.$$

Remark 6 *Note that assuming that F_x is differentiable is not a strong assumption. According to the Lebesgue differentiation theorem on $\mathbb{R}$, it will be satisfied as soon as the push forward measure of P by the function $\|x - \cdot\|^2$ is absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}$.*

Proof From Lemma 2,

$$\delta^2(x) = \frac{1}{m}\int_0^m (G_x^{-1}(t))^2 dt = \frac{1}{m}\int_0^m F_x^{-1}(t)dt$$

where $G_x(t) = \mathbb{P}(\|X - x\| \leq t)$ and $F_x(t) = \mathbb{P}(\|X - x\|^2 \leq t)$. So

$$\sqrt{n}(\widehat{\delta}^2(x) - \delta^2(x)) = \frac{1}{m}\int_0^m \sqrt{n}[\widehat{F}_x^{-1}(t) - F_x^{-1}(t)]dt. \tag{17}$$

First suppose that $\widehat{F}_x^{-1}(m) > F_x^{-1}(m)$. Then, by integrating "horizontally" rather than

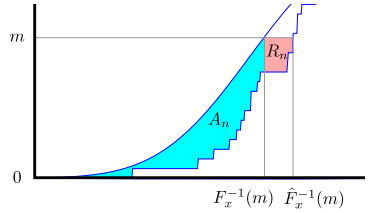


Figure 4: The integral of (17) can be decomposed into two parts, A_n and R_n.

"vertically", we can split the integral into two parts, as illustrated in Figure 4:

$$\begin{aligned} \frac{1}{m}\int_0^m \sqrt{n}[\widehat{F}_x^{-1}(t) - F_x^{-1}(t)]dt &= \frac{1}{m}\int_0^{F_x^{-1}(m)} \sqrt{n}[F_x(t) - \widehat{F}_x(t)]dt + \frac{1}{m}\int_{F_x^{-1}(m)}^{\widehat{F}_x^{-1}(m)} \sqrt{n}[m - \widehat{F}_x(t)]dt \\ &\equiv A_n(x) + R_n(x) \end{aligned} \tag{18}$$

Next, it can be easily checked that (18) is also true when $\widehat{F}_x^{-1}(m) < F_x^{-1}(m)$ if we take $\int_a^b f(u)du := -\int_b^a f(u)du$ when $a > b$. Now, since F_x is differentiable at m, we have that $\left|F_x^{-1}(m) - \widehat{F}_x^{-1}(m)\right| = O_P(1/\sqrt{n})$, see for instance Corollary 21.5 in van der Vaart (2000). According to the DKW (Dvoretsky-Kiefer-Wolfowitz) inequality we have that $\sup_t \left|F_x(t) - \widehat{F}_x(t)\right| = O_P(\sqrt{1/n})$ and thus

$$|R_n| \leq \frac{\sqrt{n}}{m} \left|F_x^{-1}(m) - \widehat{F}_x^{-1}(m)\right| \sup_t \left|F_x(t) - \widehat{F}_x(t)\right| = o_P(1).$$

Next, note that $\sqrt{n}[F_x(t) - \widehat{F}_x(t)] \rightsquigarrow \mathbb{B}(t)$, where $\mathbb{B}(t)$ is a Gaussian process with covariance function $[F_x(s \wedge t) - F_x(s)F_x(t)]$ (See, for example, van der Vaart and Wellner (1996)). By taking the integral, which is a bounded operator, we have that

$$A_n \rightsquigarrow \int_0^{F_x^{-1}(m)} \mathbb{B}(t)\, dt \stackrel{d}{=} N(0, \sigma_x^2),$$

where

$$\sigma_x^2 = \frac{1}{m^2} \int_0^{F_x^{-1}(m)} \int_0^{F_x^{-1}(m)} [F_x(s \wedge t) - F_x(s)F_x(t)]\, ds\, dt.$$

■

Now, we consider the functional limit of the distance to measure, on a compact domain $\mathcal{X} \subset \mathbb{R}^d$. The functional convergence of the DTM requires assumptions on the regularity of the quantile functions F_x^{-1}. We say that $\omega_x : (0,1) \to \mathbb{R}^+$ is a *modulus of (uniform) continuity* of F_x^{-1} if, for any $u \in (0,1)$,

$$\sup_{(m,m') \in (0,1)^2\,,\, |m'-m|<u} |F_x^{-1}(m') - F_x^{-1}(m)| \leq \omega_x(u),$$

with $\lim_{u \to 0} \omega_x(u) = \omega_x(0) = 0$. We say that $\omega_{\mathcal{X}} : (0,1) \to \mathbb{R}^+$ is an *uniform modulus of continuity* for the family of quantiles functions $(F_x^{-1})_{\mathcal{X}}$ if, for any $u \in (0,1)$ and any $x \in \mathcal{X}$,

$$\sup_{(m,m') \in (0,1)^2\,,\, |m'-m|<u} |F_x^{-1}(m') - F_x^{-1}(m)| \leq \omega_{\mathcal{X}}(u),$$

with $\lim_{u \to 0} \omega_{\mathcal{X}}(u) = \omega_{\mathcal{X}}(0) = 0$. When such modulus of continuity ω exists, note that it always can be chosen nondecreasing and this allows us to consider its generalized inverse ω^{-1}.

One may ask if the existence of the uniform modulus of continuity over a compact domain $\mathcal{X}$ is a strong assumption or not. To answer this issue, let us introduce the following assumption:

$(H_{\omega,\mathcal{X}})$: for any $x \in \mathcal{X}$, the push forward measure P_x of P by $\|x - \cdot\|^2$ is supported on a finite closed interval.

Note that Assumption $(H_{\omega,\mathcal{X}})$ is not very strong. For instance it is satisfied for a measure P supported on a compact and connected manifold, with P_x absolutely continuous for the Hausdorff measure on P. The following Lemma derives from general results on quantile functions given in Bobkov and Ledoux (2014) (see their Appendix A); the lemma shows that a uniform modulus of continuity for the quantiles exists under Assumption $(H_{\omega,\mathcal{X}})$.

Lemma 7 (Existence of Uniform Modulus of Continuity) *Let $\mathcal{X}$ be a compact domain and let P be a measure with compact support in $\mathbb{R}^d$. Assume that Assumption $(H_{\omega,\mathcal{X}})$ is satisfied. Then there exists a uniform modulus of continuity for the family of quantile functions F_x^{-1} over $\mathcal{X}$.*

Proof Let $x \in \mathcal{X}$. According to Propositions A.7 and A.12 in Bobkov and Ledoux (2014), Assumption $(H_{\omega,\mathcal{X}})$ is equivalent to assuming the existence of a uniform modulus of continuity of F_x^{-1} (it tends to zero at zero). We can then define ω_x on $(0,1)$ by

$$u \in (0,1) \mapsto \omega_x(u) := \sup_{(m,m') \in (0,1)^2\,,\, |m'-m|<u} \left|F_x^{-1}(m') - F_x^{-1}(m)\right|.$$

According to Lemma 8, we have that for any $(x, x') \in \mathcal{X}^2$:

$$\left|F_{x'}^{-1}(m) - F_x^{-1}(m)\right| \leq C\|x' - x\|, \tag{19}$$

where C only depends on P and $\mathcal{X}$. According to (19), for any $(m, m') \in (0,1)^2$, and for any $(x, x') \in \mathcal{X}^2$:

$$\left|F_x^{-1}(m') - F_x^{-1}(m)\right| \leq \left|F_{x'}^{-1}(m') - F_{x'}^{-1}(m)\right| + 2C\|x' - x\|.$$

By taking the supremum over the m and the m' such that $|m' - m| < u$, it yields:

$$\omega_x(u) \leq \omega_{x'}(u) + 2C\|x' - x\|,$$

and $x \mapsto \omega_x(u)$ is thus Lipschitz at any u. For any $u \in (0,1)$, let

$$\omega_{\mathcal{X}}(u) := \sup_{x \in \mathcal{X}} \omega_x(u),$$

which is finite because the function $x \mapsto \omega_x(u)$ is continuous on the compact $\mathcal{X}$ for any $u \in (0,1)$. We only need to prove that $\omega_{\mathcal{X}}$ is continuous at 0. Let $(u_n) \in (0,1)^{\mathbb{N}}$ be a decreasing sequence to zero. Since $\omega_{\mathcal{X}}$ is a non decreasing function, $\omega_{\mathcal{X}}(u_n)$ has a limit. For any $n \in \mathbb{N}$, there exists a point $x_n \in \mathcal{X}$ such that $\omega_{\mathcal{X}}(u_n) = \omega_{x_n}(u_n)$. Let $x_{\phi(n)}$ be a subsequence which converges to $\bar{x} \in \mathcal{X}$. According to (19),

$$\begin{aligned} \omega_{\mathcal{X}}(u_{\phi(n)}) &\leq \left|\omega_{x_{\phi(n)}}(u_{\phi(n)}) - \omega_{\bar{x}}(u_{\phi(n)})\right| + \left|\omega_{\bar{x}}(u_{\phi(n)})\right| \\ &\leq C\left\|x_{\phi(n)} - \bar{x}\right\| + \left|\omega_{\bar{x}}(u_{\phi(n)})\right| \end{aligned}$$

which gives that $\omega_{\mathcal{X}}(u_{\phi(n)})$ and $\omega_{\mathcal{X}}(u_n)$ both tend to zero because $\omega_{\bar{x}}$ is continuous at zero. Thus $\omega_{\mathcal{X}}$ is continuous at zero and the Lemma is proved. ■

We will also need the the following result, which shows that on any compact domain $\mathcal{X}$, the function $x \mapsto F_x^{-1}(m)$ is Lipschitz. For a domain $\mathcal{X} \in \mathbb{R}^d$, a probability P and a level m, we introduce the quantity $q_{P,\mathcal{X}}(m) \in \bar{\mathbb{R}}$, defined by

$$q_{P,\mathcal{X}}(m) := \sup_{x \in \mathcal{X}} F_x^{-1}(m).$$

Lemma 8 (Lipschitz Lemma) *Let P be a measure on $\mathbb{R}^d$ and let $m \in (0,1)$. Then, for any $(x, x') \in \mathbb{R}^d$,*

$$\left|\sqrt{F_{x'}^{-1}(m)} - \sqrt{F_x^{-1}(m)}\right| \leq \|x' - x\|.$$

Moreover, if $\mathcal{X}$ is a compact domain in $\mathbb{R}^d$, then $q_{P,\mathcal{X}}(m) < \infty$ and for any $(x, x') \in \mathcal{X}^2$:

$$\left|F_{x'}^{-1}(m) - F_x^{-1}(m)\right| \leq 2\sqrt{q_{P,\mathcal{X}}(m)}\ \|x' - x\|.$$

Proof Let $(x, a) \in \mathbb{R}^2$, note that

$$B\left(x, \sqrt{F_x^{-1}(m)}\right) \subseteq B\left(x + a, \sqrt{F_x^{-1}(m)} + \|a\|\right),$$

which implies

$$m = \mathbb{P}\left[B\left(x, \sqrt{F_x^{-1}(m)}\right)\right] \leq \mathbb{P}\left[B\left(x + a, \sqrt{F_x^{-1}(m)} + \|a\|\right)\right].$$

Therefore $\sqrt{F_{x+a}^{-1}(m)} \leq \sqrt{F_x^{-1}(m)} + \|a\|$. Similarly,

$$m = \mathbb{P}\left[B\left(x + a, \sqrt{F_{x+a}^{-1}(m)}\right)\right] \leq \mathbb{P}\left[B\left(x, \sqrt{F_{x+a}^{-1}(m)} + \|a\|\right)\right],$$

which implies $\sqrt{F_x^{-1}(m)} \leq \sqrt{F_{x+a}^{-1}(m)} + \|a\|$.

Let $\mathcal{X}$ be a compact domain of $\mathbb{R}^d$, then according to the previous result for some fixed $x \in \mathcal{X}$ and for any $x' \in \mathcal{X}$, $\sqrt{F_{x'}^{-1}(m)} \leq \|x' - x\| + \sqrt{F_x^{-1}(m)}$ which is bounded on $\mathcal{X}$. The last statement follows from the fact that $|x - y| = |\sqrt{x} - \sqrt{y}|\ |\sqrt{x} + \sqrt{y}|$. ■

We are now in position to state the functional limit of the distance to measure to the empirical measure.

Theorem 9 (Functional Limit) *Let P be a measure on $\mathbb{R}^d$ with compact support. Let $\mathcal{X}$ be a compact domain on $\mathbb{R}^d$ and $m \in (0,1)$. Assume that there exists a uniform modulus of continuity $\omega_{\mathcal{X}}$ for the family $(F_x^{-1})_{\mathcal{X}}$. Then $\sqrt{n}(\widehat{\delta}^2(x) - \delta^2(x)) \rightsquigarrow \mathbb{B}(x)$ for a centered Gaussian process $\mathbb{B}(x)$ with covariance kernel*

$$\kappa(x, y) = \frac{1}{m^2}\int_0^{F_x^{-1}(m)}\int_0^{F_y^{-1}(m)}\left(\mathbb{P}\left[B(x, \sqrt{t}) \cap B(y, \sqrt{s})\right] - F_x(t)F_y(s)\right) ds\, dt.$$

Remark 10 *Note that the functional limit is valid for any value of $m \in (0,1)$. A local version of this result could be also proposed by considering the (local) modulii of continuity of the quantile functions at m. For the sake of clarity, we prefer to give a global version.*

Remark 11 *By the delta method (as described in Section 4) a similar result holds for $\sqrt{n}(\widehat{\delta}(x) - \delta(x))$ as long as $\inf_x \delta(x) > 0$.*

Proof In the proof of Theorem 5 we showed that $\sqrt{n}(\widehat{\delta}^2(x) - \delta^2(x)) = A_n(x) + R_n(x)$ where

$$A_n(x) = \frac{1}{m}\int_0^{F_x^{-1}(m)} \sqrt{n}[F_x(t) - \widehat{F}_x(t)]dt$$
$$R_n(x) = \frac{1}{m}\int_{F_x^{-1}(m)}^{\widehat{F}_x^{-1}(m)} \sqrt{n}[m - \widehat{F}_x(t)]dt.$$

First, we show that $\sup_{x\in\mathcal{X}}|R_n(x)| = o_P(1)$. Then we prove that $A_n(x)$ converges to a Gaussian process.

Note that $|R_n(x)| \leq \frac{\sqrt{n}}{m}|S_n(x)||T_n(x)|$ where

$$S_n(x) = \left|F_x^{-1}(m) - \widehat{F}_x^{-1}(m)\right|, \quad T_n(x) = \sup_t\left|F_x(t) - \widehat{F}_x(t)\right|.$$

Let $\xi_i \sim$Uniform (0,1), for $i = 1, \ldots, n$ and let H_n be their empirical distribution function. Define $k = mn$. Then $\widehat{F}_x^{-1}(m) \stackrel{d}{=} F_x^{-1}(\xi_{(k)}) = F_x^{-1}\left(H_n^{-1}(m)\right)$, where $\xi_{(k)}$ is the kth order statistic. Thus, for any $m > 0$ and any $x \in \mathcal{X}$:

$$\begin{aligned}
\mathbb{P}\left(|S_n(x)| > \epsilon\right) &= \mathbb{P}\left(|F_x^{-1}(H_n^{-1}(m)) - F_x^{-1}(m)| > \epsilon\right) \\
&\leq \mathbb{P}\left(\omega_{\mathcal{X}}(|m - H_n^{-1}(m)|) > \epsilon\right) \\
&\leq \mathbb{P}\left(|m - H_n^{-1}(m)| > \omega_{\mathcal{X}}^{-1}(\epsilon)\right) \\
&\leq 2\exp\left\{-\frac{n\left[\omega_{\mathcal{X}}^{-1}(\epsilon)\right]^2}{m}\frac{1}{1 + \frac{2\omega_{\mathcal{X}}^{-1}(\epsilon)}{3m}}\right\} \qquad (20)
\end{aligned}$$

In the last line we used inequality 1 page 453 and Point (12) of Proposition 1 page 455 of Shorack and Wellner (2009). Note that $\omega_{\mathcal{X}}^{-1}(\epsilon) > 0$ for any $\varepsilon > 0$ because $\omega_{\mathcal{X}}$ is assumed to be continuous at zero by definition.

Fix $\varepsilon > 0$. There exists an absolute constant $C_{\mathcal{X}}$ such that there exists an integer $N \leq C_{\mathcal{X}}\varepsilon^{-d}$ and N points $(x_1, \ldots, x_N)$ laying in $\mathcal{X}$ such that $\bigcup_{j=1\ldots N} B_j \supseteq \mathcal{X}$, where $B_j = B(x_j, \varepsilon)$. Now, we apply Lemma 8 with P, and with P_n and we find that for any $x \in B_j$:

$$\left|F_x^{-1}(m) - F_{x_j}^{-1}(m)\right| \leq 2\sqrt{q_{P,\mathcal{X}}(m)}\ \varepsilon \quad \text{and} \quad \left|\widehat{F}_x^{-1}(m) - \widehat{F}_{x_j}^{-1}(m)\right| \leq 2\sqrt{q_{P_n,\mathcal{X}}(m)}\ \varepsilon.$$

Thus, for any $x \in B_j$,

$$\begin{aligned}
\left|F_x^{-1}(m) - \widehat{F}_x^{-1}(m)\right| &\leq \left|F_x^{-1}(m) - F_{x_j}^{-1}(m)\right| + \left|F_{x_j}^{-1}(m) - \widehat{F}_{x_j}^{-1}(m)\right| + \left|\widehat{F}_{x_j}^{-1}(m) - \widehat{F}_x^{-1}(m)\right| \\
&\leq 2\left[\sqrt{q_{P,\mathcal{X}}(m)} + \sqrt{q_{P_n,\mathcal{X}}(m)}\right]\varepsilon + |F_{x_j}^{-1}(m) - \widehat{F}_{x_j}^{-1}(m)| \\
&\leq C\varepsilon + |F_{x_j}^{-1}(m) - \widehat{F}_{x_j}^{-1}(m)| \qquad (21)
\end{aligned}$$

where C is a positive constant which only depends on $\mathcal{X}$ and P. Using a union bound together with (20) , we find that

$$P\left(\sup_{x\in\mathcal{X}}|S_n(x)|>2C\varepsilon\right)\leq P\left(\sup_{j=1\dots N}|S_n(x_j)|>C\varepsilon\right)$$
$$\leq 2C_{\mathcal{X}}\varepsilon^{-d}\exp\left\{-\frac{n\left[\omega_{\mathcal{X}}^{-1}(C\epsilon)\right]^2}{m}\frac{1}{1+\frac{2\omega_{\mathcal{X}}^{-1}(C\epsilon)}{3m}}\right\}.$$

Thus, $\sup_{x\in\mathcal{X}}|S_n(x)|=o_P(1)$. Then

$$\sup_{x\in\mathcal{X}}|T_n(x)|=\sup_{x\in\mathcal{X}}\sup_t|\widehat{F}_x(t)-F_x(t)|$$
$$=\sup_{x\in\mathcal{X}}\sup_t\left|\mathbb{P}_n\left(B(x,\sqrt{t}\right)-\mathbb{P}\left(B(x,\sqrt{t}\right)\right|$$
$$\leq\sup_{B\in\mathcal{B}_d}|\mathbb{P}_n(B)-\mathbb{P}(B)|=O_P\left(\sqrt{\frac{d}{n}}\right)\tag{22}$$

where $\mathcal{B}_d$ is the set of balls in $\mathbb{R}^d$ and we used the Vapnik-Chervonenkis theorem. Finally, we obtain that

$$\sup_{x\in\mathcal{X}}|R_n(x)|\leq\frac{\sqrt{n}}{m}\sup_{x\in\mathcal{X}}|S_n(x)|\sup_{x\in\mathcal{X}}|T_n(x)|=o_P(1).\tag{23}$$

Since $\sup_{x\in\mathcal{X}}|R_n(x)|=o_P(1)$, it only remains to prove that the process A_n converges to a Gaussian process.

Now, we consider the process A_n on $\mathcal{X}$. Let us denote $\nu_n:=\sqrt{n}(P_n-P)$ the empirical process. Note that

$$A_n(x)=\frac{1}{m}\nu_n\left(\int_0^{F_x^{-1}(m)}I_{\|x-X\|^2\leq t}dt\right)=\frac{1}{m}\nu_n\left(f_x\right)$$

where $f_x(y):=\left[F_x^{-1}(m)-\|x-y\|^2\right]\wedge 0$. For any $(x,x')\in\mathcal{X}$ and any $y\in\mathbb{R}^d$, we have

$$|f_x(y)-f_x(y)|\leq|F_x^{-1}(m)-F_{x'}^{-1}(m)|+\|x-x'\|\left[\|x\|+\|x'\|+2\|y\|\right]$$
$$\leq 2\left[r(\mathcal{X})+\|y\|+\sqrt{q_{P,\mathcal{X}}(m)}\right]\|x-x'\|$$

Since P is compactly supported, then the collection of functions $(f_x)_{x\in\mathcal{X}}$ is P-Donsker (see for instance 19.7 in van der Vaart (2000)) and $A_n(x)\rightsquigarrow\mathbb{B}(x)$ for a centered Gaussian process $\mathbb{B}(x)$ with covariance kernel

$$\kappa(x,y)=\mathrm{Cov}(A_n(x),A_n(y))=\mathbb{E}[A_n(x)A_n(y)]$$
$$=\frac{1}{m^2}\int_0^{F_x^{-1}(m)}\int_0^{F_y^{-1}(m)}\mathbb{E}\left[\left(\widehat{F}_x(t)-F_x(t)\right)\left(\widehat{F}_y(s)-F_y(s)\right)\right]\,ds\,dt$$
$$=\frac{1}{m^2}\int_0^{F_x^{-1}(m)}\int_0^{F_y^{-1}(m)}\left(\mathbb{P}\left[B(x,\sqrt{t})\cap B(y,\sqrt{s})\right]-F_x(t)F_y(s)\right)\,ds\,dt.$$

■

4. Hadamard Differentiability and The Bootstrap

In this section, we use the bootstrap to get a confidence band for δ. Define c_α by

$$\mathbb{P}(\sqrt{n}||\widehat{\delta}-\delta||_\infty>c_\alpha)=\alpha.$$

Let $X_1^*,\dots,X_n^*$ be a sample from the empirical measure P_n and let $\widehat{\delta}^*$ be the corresponding empirical DTM. The bootstrap estimate $\widehat{c}_\alpha$ is defined by

$$\mathbb{P}(\sqrt{n}||\widehat{\delta}^*-\widehat{\delta}||_\infty>\widehat{c}_\alpha\mid X_1,\dots,X_n)=\alpha.$$

As usual, $\widehat{c}_\alpha$ can be approximated by Monte Carlo. Below we show that this bootstrap is valid. It then follows that

$$\mathbb{P}\left(||\delta-\widehat{\delta}||_\infty<\frac{\widehat{c}_\alpha}{\sqrt{n}}\right)\to 1-\alpha.$$

A different approach to the bootstrap is considered in Section 6.

To prepare for our next result, let $\mathcal{B}$ denote the class of all closed Euclidean balls in $\mathbb{R}^d$ and let $\mathbb{B}$ denote the P-Brownian bridge on $\mathcal{B}$, i.e. the centered Gaussian process on $\mathcal{B}$ with covariance function $\kappa(B,C)=P(B\cap C)-P(B)P(C)$, with B,C in $\mathcal{B}$. We will denote with $\mathbb{B}_x(r)$ the value of $\mathbb{B}$ at $B(x,r)$, the closed ball centered at $x\in\mathbb{R}^d$ and with radius $r>0$.

Theorem 12 (Bootstrap Validity) *Let P be a measure on $\mathbb{R}^d$ with compact support S, $m\in(0,1)$ be fixed and $\mathcal{X}$ be a compact domain in $\mathbb{R}^d$. Assume that $F_{P,x}=F_x$ is differentiable at $F_x^{-1}(m)$ and that there exist a constant $C>0$ such that for all small $\eta\in\mathbb{R}$,*

$$\sup_{x\in\mathcal{X}}\left|F_x\left(F_x^{-1}(m)\right)-F_x(F_x^{-1}(m)+\eta)\right|<\epsilon\quad\textit{implies}\quad|\eta|<C\epsilon.\tag{24}$$

for all $x\in\mathcal{X}$. Then, $\sup_{x\in\mathcal{X}}\sqrt{n}\left|\left(\widehat{\delta}^*(x)\right)^2-\left(\widehat{\delta}(x)\right)^2\right|$ converges in distribution to

$$\sup_{x\in\mathcal{X}}\left|\frac{1}{m}\int_0^{F_x^{-1}(m)}\mathbb{B}_x(u)du\right|$$

conditionally given $X_1,X_2,\dots$, in probability.

We will establish the above result using the functional delta method, which entails showing that the distance to measure function is Hadamard differentiable at P. In fact, the proof further shows that the process

$$x\in\mathcal{X}\mapsto\sqrt{n}\left(\delta^2(x)-\widehat{\delta}^2(x)\right),$$

converges weakly to the Gaussian process

$$x\in\mathcal{X}\mapsto-\frac{1}{m}\int_0^{F_x^{-1}(m)}\mathbb{B}_x(u)du.$$

Remark 13 *This result is consistent with the result established in Theorem 9, but in order to establish Hadamard differentiability, we use a slightly different assumption. Theorem 9 is proved by assuming a uniform modulus of continuity on the quantile functions F_x^{-1} whereas in Theorem 12 a uniform lower bound on the derivatives is required. These two assumptions are consistent: they both say that F_x^{-1} is well-behaved in a neighborhood of m for all x. However, the condition used in Theorem 12 is stronger than the condition used in Theorem 9.*

Proof [Proof of Theorem 12] Let us first give the definition of Hadamard differentiability, for which we refer the reader to, e.g., Section 3.9 of van der Vaart and Wellner (1996). A map ϕ from a normed space $(\mathcal{D}, \|\cdot\|_{\mathcal{D}})$ to a normed space $(\mathcal{E}, \|\cdot\|_{\mathcal{E}})$ is Hadamard differentiable at the point $x \in \mathcal{D}$ if there exists a continuous linear map $\phi'_x : \mathcal{D} \to \mathcal{E}$ such that

$$\left\| \frac{\phi(x + th_t) - \phi(x)}{t} - \phi'_x(h) \right\|_{\mathcal{E}} \to 0, \tag{25}$$

whenever $\|h_t - h\|_{\mathcal{D}} \to 0$ as $t \to 0$.

We also recall the functional delta method (see, e.g. van der Vaart and Wellner, 1996, Theorem 3.9.4): suppose that T_n takes values in $\mathcal{D}$, $r_n \to \infty$, $r_n(T_n - \theta) \rightsquigarrow T$, and suppose that ϕ is Hadamard differentiable at θ. Then $r_n(\phi(T_n) - \phi(\theta)) \rightsquigarrow \phi'_\theta(T)$. Moreover, by Theorem 3.9.11 of van der Vaart and Wellner (1996) the bootstrap has the same limit. More precisely, given $X_1, X_2, \ldots$, we have that $r_n(\phi(T_n^*) - \phi(T_n))$ converges conditionally in distribution to $\phi'_\theta(T)$, in probability. This implies the validity of the bootstrap confidence sets.

We define $\mathcal{M}$ to be the space of finite, σ-finite signed measures on $(\mathbb{R}^d, \mathcal{B}^d)$ supported on the compact set S and the mapping $\|\cdot\|_{\mathcal{B}} \colon \mathcal{M} \mapsto \mathbb{R}$ given by

$$\|\mu\|_{\mathcal{B}} = \sup_{B \in \mathcal{B}} |\mu(B)|, \quad \mu \in \mathcal{M}.$$

In Lemma 14 we show that this is a normed space.

For our purposes, instead of using $\mathcal{M}$ it will be convenient to work with the equivalent space of the evaluations of all $\mu \in \mathcal{M}$ over the balls $\mathcal{B}$. Formally, let $\ell^\infty(\mathcal{B})$ denote the normed space of bounded functions on $\mathcal{B}$ equipped with the supremum norm. Then, by Lemma 14, the mapping from $\mathcal{M}$ into $\ell^\infty(\mathcal{B})$ given by

$$\mu \mapsto (\mu \colon \mathcal{B} \to [0, 1]) \tag{26}$$

is a bijection on its image, which we will denote by $\mathcal{D}$. By definition, the supremum norm on $\mathcal{D}$ is exactly the norm $\|\cdot\|_{\mathcal{B}}$, so that $\mathcal{D} \subset \ell^\infty(\mathcal{B})$ equipped with the supremum norm is a normed space. With a slight abuse of notation, we will identify measures in $\mathcal{M}$ with the corresponding points in $\mathcal{D}$ and write $\mu \in \mathcal{D}$ to denote the signed measure μ corresponding to the point $\{\mu(B), B \in \mathcal{B}\}$ in $\mathcal{D}$.

The advantage of using the space $\mathcal{D}$ instead of $\mathcal{M}$ is that the convergence of the empirical process $(\sqrt{n}(\mathbb{P}_n(B) - P(B)) : \ B \in \mathcal{B})$ to the Brownian bridge takes place in $\mathcal{D}$, as required by the delta-method for the bootstrap (see van der Vaart and Wellner, 1996, Theorem 3.9.).

For a signed measure μ in $\mathcal{M}$, $x \in \mathbb{R}^d$ and $r > 0$, we set $F_{\mu,x}(r) = \mu\left(B(x, \sqrt{r})\right)$. Notice if P is a probability measure, then $F_{P,x}$ is the c.d.f. of the univariate random variable $\|X - x\|^2$, with $X \sim P$. For a general $\mu \in \mathcal{M}$, $F_{\mu,x}$ is a cadlag function, though not monotone. For any $m \in \mathbb{R}$, μ in $\mathcal{M}$ and $x \in \mathbb{R}^d$, set

$$F^{-1}_{\mu,x}(m) = \inf\left\{ r > 0 \colon \mu(B(x, \sqrt{r})) \geq m \right\},$$

where the infimum over the empty set is define to be ∞. If P is a probability measure and $m \in (0, 1)$ then $F^{-1}_{P,x}(m)$ is just the m-th quantile of the random variable $\|X - x\|^2$, $X \sim P$.

Fix a $m \in (0, 1)$ and let $\mathcal{M}_m = \mathcal{M}_m(\mathcal{X})$ denote the subset of $\mathcal{M}$ consisting of all finite signed measure μ such that, there exists a value of $r > 0$ for which $\inf_{x \in \mathcal{X}} \mu\left(B(x, \sqrt{r})\right) \geq m$. Thus, for any $\mu \in \mathcal{M}_m$ and $x \in \mathcal{X}$, $F^{-1}_{\mu,x}(m) < \infty$. Let $\mathcal{D}_m$ be the image of $\mathcal{M}_m$ by the mapping (26).

Let $\mathcal{E}$ the set of bounded, real-valued function on $\mathcal{X}$, a normed space with respect to the sup norm. Finally, we define $\phi \colon \mathcal{D}_m \to \mathcal{E}$ to be the mapping

$$\mu \in \mathcal{D}_m \mapsto \phi(\mu)(x) = F^{-1}_{\mu,x}(m) - \frac{1}{m} \int_0^{F^{-1}_{\mu,x}(m)} F_{\mu,x}(u) du, \quad x \in \mathcal{X} \tag{27}$$

Notice that if P is a probability measure, simple algebra shows that $\phi(P)(x)$ is the square value of the distance to measure of P at the point x, i.e. $\delta_P^2(x)$; see Figure 5.

Below we will show that, for any probability measure P, the mapping (27) is Hadamard differentiable at P.

For an arbitrary $Q \in \mathcal{D}$, let $\{Q_t\}_{t>0} \subset \mathcal{D}$ be a sequence of signed measure such that $\lim_{t \to 0} \|Q_t - Q\|_{\mathcal{B}} = 0$ and such that $P + tQ_t \in \mathcal{D}_m$ for all t. Sequences of this form exist: since $\|tQ_t\|_{\mathcal{B}} \to 0$ as $t \to 0$, for any arbitrary $0 < \epsilon < 1 - m$ and all t small enough,

$$\inf_{x \in \mathcal{X}} (P + tQ_t)\left(B\left(x, F^{-1}_{P,x}(m + \epsilon)\right)\right) \geq m + \epsilon/2.$$

By the boundedness of $\mathcal{X}$ and compactness of S, this implies that there exists a number $r > 0$ such that

$$\inf_x (P + tQ_t)\left(B\left(x, r\right)\right) \geq m,$$

so the image of $P + tQ_t$ by (27) is an element of $\mathcal{E}$ (i.e. it is a bounded function).

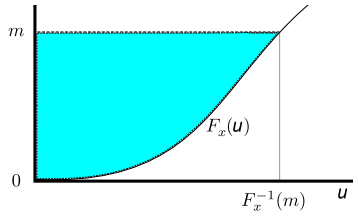


Figure 5: The integral $\int_0^m F^{-1}_{P,x}(u)du$ is equivalent to $mF^{-1}_{P,x}(m) - \int_0^{F^{-1}_{P,x}(m)} F_{P,x}(u)du$.

For sake of readability, below we will write $F_{x,t}$ and $F^{-1}_{x,t}(m)$ for $F_{P+tQ_t,x}$ and $F^{-1}_{P+tQ_t,x}(m)$, respectively, and F_x for $F_{P,x}$. Also, for each $x \in \mathcal{X}$ and $z \in \mathbb{R}_+$ we the set $\mathcal{A}_{x,z} = \{y : \|y - x\|^2 \leq z\}$ and $F_{x,t}(z) = (P + tQ_t)(\mathcal{A}_{x,z})$.

Thus,

$$\phi(P)(x) = \delta_P^2(x) = F_x^{-1}(m) - \frac{1}{m}\int_0^{F_x^{-1}(m)} F_x(u)du$$

and

$$\phi(P + tQ_t)(x) = F_{x,t}^{-1}(m) - \frac{1}{m}\int_0^{F_{x,t}^{-1}(m)} F_{x,t}(u)du.$$

Some algebra show that, for any x,

$$\frac{\phi(P)(x) - \phi(P + tQ_t)(x)}{t} = \frac{F_x^{-1}(m) - F_{x,t}^{-1}(m)}{t} - \frac{A(x,t)}{mt}, \tag{28}$$

where

$$A(x,t) = \begin{cases} \int_0^{F_x^{-1}(m)} [F_x(u) - F_{x,t}(u)]\, du - \int_{F_x^{-1}(m)}^{F_{x,t}^{-1}(m)} F_{x,t}(u)du & \text{if } F_x^{-1}(m) \le F_{x,t}^{-1}(m) \\ \int_0^{F_x^{-1}(m)} [F_x(u) - F_{x,t}(u)]\, du + \int_{F_{x,t}^{-1}(m)}^{F_x^{-1}(m)} F_{x,t}(u)du & \text{if } F_x^{-1}(m) > F_{x,t}^{-1}(m). \end{cases}$$

To demonstrate Hadamard differentiability (see 25), we will prove that, as $t \to 0$, the expression in (28), as a bounded function of $x \in \mathcal{X}$, will converge in $\mathcal{E}$ to the bounded function

$$x \in \mathcal{X} \mapsto -\frac{1}{m}\int_0^{F_x^{-1}(m)} Q(\mathcal{A}_{x,u})du.$$

Towards that end, we have, for all t and any $x \in \mathcal{X}$,

$$\begin{aligned} \frac{A(x,t)}{t} &= \frac{1}{t}\left[\int_0^{F_x^{-1}(m)} tQ_t(\mathcal{A}_{x,u})du - \int_{F_x^{-1}(m)}^{F_{x,t}^{-1}(m)} (P + tQ_t)(\mathcal{A}_{x,u})du\right] \\ &= \int_0^{F_x^{-1}(m)} Q_t(\mathcal{A}_{x,u})du - \frac{1}{t}\int_{F_x^{-1}(m)}^{F_{x,t}^{-1}(m)} P(\mathcal{A}_{x,u})du - \int_{F_x^{-1}(m)}^{F_{x,t}^{-1}(m)} (Q_t)(\mathcal{A}_{x,u})du \\ &\equiv A_1(x,t) - A_2(x,t) - A_3(x,t), \end{aligned}$$

where, for $a < b$, we write $\int_b^a = -\int_a^b$.

To handle the three terms appearing in the last display, we use Lemma 15 below which shows that $\sup_{x\in\mathcal{X}} \left|m - F_x\left(F_{x,t}^{-1}(m)\right)\right| = O(t)$ and that $\sup_{x\in\mathcal{X}} |F_x^{-1}(m) - F_{x,t}^{-1}(m)| = O(t)$ as $t \to 0$.

We now analyze the terms $A_1(x,t)$, $A_2(x,t)$ and $A_3(x,t)$ separately.

- **Term $A_1(x,t)$.** As $t \to 0$, $Q_t \to Q$ and, uniformly in $x \in \mathcal{X}$ and $z > 0$, $|Q_t(\mathcal{A}_{x,z})| \le |Q_t(S)| = |Q(S)| + o(1) < \infty$. Furthermore, $\sup_{x\in\mathcal{X}} F_x^{-1}(m) < \infty$ by compactness of $\mathcal{X}$ and S. Therefore, using the dominated convergence theorem,

$$\lim_{t\to 0}\sup_{x\in\mathcal{X}}\left|\frac{A_1(x,t)}{m} - \frac{1}{m}\int_0^{F_x^{-1}(m)} Q(\mathcal{A}_{x,u})du\right| = 0. \tag{29}$$

- **Term $A_2(x,t)$.** Since $P(\mathcal{A}_{x,u})$ is non-decreasing in u for all x, we have

$$\frac{F_{x,t}^{-1}(m) - F_x^{-1}(m)}{t} \times \min\left\{m, F_x(F_{x,t}^{-1}(m))\right\} \le A_2(x,t) \le \frac{F_{x,t}^{-1}(m) - F_x^{-1}(m)}{t} \times \max\left\{m, F_x(F_{x,t}^{-1}(m))\right\}.$$

  Using (32), we conclude that

$$\lim_{t\to 0}\sup_{x\in\mathcal{X}}\left|\frac{F_x^{-1}(m) - F_{x,t}^{-1}(m)}{t} - \frac{A_2(x,t)}{m}\right| = 0. \tag{30}$$

- **Term $A_3(x,t)$.** Finally, since $|Q_t(S)| \le |Q(S)| + o(1)$ as $t \to 0$ and using (35), we obtain

$$\sup_x |A_3(x,t)| = O\Big(\sup_x \left|F_{x,t}^{-1}(m) - F_x^{-1}(m)\right|\Big) = o(1) \tag{31}$$

  as $t \to 0$.

Therefore, from (28), (29), (30), and (31),

$$\lim_{t\to 0}\sup_x\left|\frac{\phi(P)(x) - \phi(P + tQ_t)(x)}{t} + \frac{1}{m}\int_0^{F_x^{-1}(m)} Q(\mathcal{A}_{x,u})du\right| = 0,$$

which shows that

$$x \in \mathcal{X} \mapsto -\frac{1}{m}\int_0^{F_x^{-1}(m)} Q(\mathcal{A}_{x,u})du$$

is the Hadamard derivative of δ^2 at P.

The statement of the theorem now follows from an application of Theorem 3.9.11 in van der Vaart and Wellner (1996) and the fact that, since $\mathcal{B}$ is a Donsker class, the empirical process $(\sqrt{n}(\mathbb{P}_n(B) - P(B)) : B \in \mathcal{B})$ converges to the Brownian bridge $\mathbb{B}$ on $\mathcal{B}$ with covariance kernel $\kappa(B,C) = P(B \cap C) - P(B)P(C)$. ■

Lemma 14 (Normed Space) *The pair $(\mathcal{M}, \|\cdot\|_{\mathcal{B}})$ is a normed space.*

Proof It is clear that $\mathcal{M}$ is closed under addition and scalar multiplication, and so it is a linear space. We then need to show that the mapping $\|\cdot\|_{\mathcal{B}}$ is a norm. It is immediate to see that it absolutely homogeneous and satisfies the triangle inequality: for any μ and ν in $\mathcal{M}$ and $c \in \mathbb{R}$, $\|c\mu\|_{\mathcal{B}} = |c|\|\mu\|_{\mathcal{B}}$ and $\|\mu + \nu\|_{\mathcal{B}} \le \|\mu\|_{\mathcal{B}} + \|\nu\|_{\mathcal{B}}$. It remains to prove that $\|\mu\|_{\mathcal{B}} = 0$ if and only if μ is identically zero, i.e. $\mu(A) = 0$ for all Borel sets A. One direction is immediate: if $\|\mu\|_{\mathcal{B}} > 0$, then there exists a ball B such that $\mu(B) \neq 0$, so that $\mu \neq 0$. For the other direction, assume that $\mu \in \mathcal{M}$ is such that $\|\mu\|_{\mathcal{B}} = 0$. By the Jordan decomposition, μ can be represented as the difference of two singular, non-negative finite measures: $\mu = \mu_+ - \mu_-$. The condition $\mu(B) = 0$ for all $B \in \mathcal{B}$ is equivalent to $\mu_+(B) = \mu_-(B)$ for all $B \in \mathcal{B}$. We will show that this further implies that the supports of μ_+ and μ_-, denoted with S_+ and S_- respectively, are both empty, and therefore that μ is identically zero. Indeed, recall that the support of a Borel measure λ over a topological space $\mathbb{X}$ is the set of points $x \in \mathbb{X}$ all of whose open neighborhoods have positive λ-measure.

In our setting this is equivalent to the set of points in $\mathbb{R}^d$ such that all open balls centered at those points have positive measure, which in turn is equivalent to the set of points such that all closed balls centered at those points have positive measure. Therefore, using the fact that $\mu^+(B) = \mu^-(B)$, for all $B \in \mathcal{B}$,

$$S_+ = \left\{x \in \mathbb{R}^d \colon \mu_+(B(x,r)) > 0, \forall r > 0\right\} = \left\{x \in \mathbb{R}^d \colon \mu_-(B(x,r)) > 0, \forall r > 0\right\} = S_-.$$

where $B(X,r) = \{y \in \mathbb{R}^d \colon \|y - x\| \leq r\}$. It then follows that S_+ and S_- must be empty, for otherwise μ_+ and μ_- would be mutually singular, non-zero measures with the same support, a contradiction. ■

Lemma 15 *Under the assumptions of the theorem and as $t \to 0$,*

$$\sup_{x \in \mathcal{X}} \left| m - F_x\left(F_{x,t}^{-1}(m)\right)\right| = O(t), \tag{32}$$

and

$$\sup_{x \in \mathcal{X}} |F_x^{-1}(m) - F_{x,t}^{-1}(m)| = O(t). \tag{33}$$

Proof Set $\mathcal{A}_{x,t} = \{y \colon \|y - x\|^2 \leq F_{x,t}^{-1}(m)\}$ and let γ_t be any positive, decreasing function of t such that $\gamma_t = o(t)$ as $t \to 0$. Then, for all small enough t and all $x \in \mathcal{X}$, the set $\mathcal{A}_{x,t,\gamma_t} = \{y \colon \|y - x\|^2 \leq F_{x,t}^{-1}(m) - \gamma_t\}$ is non-empty and

$$F_x\left(F_{x,t}^{-1}(m) - \gamma_t\right) + tQ_t(\mathcal{A}_{x,t,\gamma_t}) \leq m \leq F_x\left(F_{x,t}^{-1}(m)\right) + tQ_t(\mathcal{A}_{x,t}), \tag{34}$$

because $P(\mathcal{A}_{x,t}) = F_x(F_{x,t}^{-1}(m))$ and $P(\mathcal{A}_{x,t,\gamma}) = F_x(F_{x,t}^{-1}(m) - \gamma_t)$. Rearranging and using the bound $\sup_{x\in\mathcal{X}} F_x(F_{x,t}^{-1}(m) - \gamma_t) = \sup_{x\in\mathcal{X}} F_x(F_{x,t}^{-1}(m)) + O(\gamma_t)$, which holds for all small enough t, we obtain that, for all such values of t,

$$\sup_{x \in \mathcal{X}} \left| m - F_x\left(F_{x,t}^{-1}(m)\right)\right| \leq t \sup_{x \in \mathcal{X}} \max\left\{|Q_t(\mathcal{A}_{x,t})|, |Q_t(\mathcal{A}_{x,t,\gamma_t})|\right\} + o(t) = tO\left(|Q(S)|\right),$$

since $|Q_t(S)| = |Q(S)| + o(1)$ as $t \to 0$. This establishes (32). Next, by the monotonicity of F_x for each $x \in \mathcal{X}$ and the facts – both implied by (24) – that $m = F_x(F_x^{-1}(m))$ and $\inf_{x\in\mathcal{X}} F_x'(F_x^{-1}(m))$ is bounded away from 0, (32) yields that

$$\lim_{t \to 0} \sup_{x \in \mathcal{X}} |F_x^{-1}(m) - F_{x,t}^{-1}(m)| = 0. \tag{35}$$

Combining the last display with the bound

$$\sup_{x \in \mathcal{X}} \left| F_x\left(F_x^{-1}(m)\right) - F_x\left(F_{x,t}^{-1}(m)\right)\right| = tO\left(|Q(S)|\right)$$

and assumption (24) again, we obtain that

$$\sup_{x \in \mathcal{X}} \left|F_x^{-1}(m) - F_{x,t}^{-1}(m)\right| \leq CtO(|Q(S)|) = O(t),$$

for all t small enough, where C is the constant in (24). This completes the proof of (33). ■

4.1 Significance of Topological Features

Fasy et al (2014) showed how to use the bootstrap to test the significance of a topological feature. They did this for distance functions and density estimators but the same idea works for DTM as we now explain. We assume in this section that the support of the distribution is contained in a compact set. The supremum norm refers to the supremum over this set.

Given a feature with birth and death time (u, v), we will say that the feature is significant if $|v - u| > 2c_\alpha/\sqrt{n}$ where c_α is defined by

$$\mathbb{P}(\sqrt{n}\|\widehat{\delta}(x) - \delta(x)\|_\infty > c_\alpha) = \alpha.$$

In particular, c_α can be estimated from the bootstrap as we showed in the previous section. Specifically, define $\widehat{c}_\alpha$ by

$$\mathbb{P}(\sqrt{n}\|\widehat{\delta}^*(x) - \widehat{\delta}(x)\|_\infty > \widehat{c}_\alpha | X_1, \ldots, X_n) = \alpha.$$

Then $\widehat{c}_\alpha$ is a consistent estimate of c_α.

To see why this makes sense, let $\mathcal{D}$ be the set of all persistence diagrams. Let $D \equiv D_\delta$ be the true diagram and let $\widehat{D} \equiv D_{\widehat{\delta}}$ be the estimated diagram. Let

$$\mathcal{C}_n = \left\{E \in \mathcal{D} \colon\ W_\infty(\widehat{D}, E) \leq \frac{\widehat{c}_\alpha}{\sqrt{n}}\right\}.$$

Then

$$\mathbb{P}(D \in \mathcal{C}_n) = \mathbb{P}\left(W_\infty(D, \widehat{D}) \leq \frac{\widehat{c}_\alpha}{\sqrt{n}}\right) \geq \mathbb{P}(\sqrt{n}\|\widehat{\delta}(x) - \delta(x)\|_\infty \leq \widehat{c}_\alpha) \to 1 - \alpha$$

as $n \to \infty$. Now $|v - u| > 2\widehat{c}_\alpha/\sqrt{n}$ if and only if the feature cannot be matched to the diagonal for any diagram in $\mathcal{C}$. (Recall that the diagonal corresponds to features with zero lifetime.)

We can visualize the significant features by putting a band of size $2c_\alpha/\sqrt{n}$ around the diagonal of $\widehat{D}$. See Figure 6.

5. Theory for Kernels

In this section, we consider an alternative to the DTM, namely, kernel based methods. This includes the kernel distance and the kernel density estimator.

Phillips et al. (2014) suggest using the kernel distance for topological inference. Given a kernel $K(x, y)$, the kernel distance between two probability measures P and Q is

$$D_K(P,Q) = \sqrt{\iint K(x,y)dP(x)dP(y) + \iint K(x,y)dQ(x)dQ(y) - 2\iint K(x,y)dP(x)dQ(y)}.$$

It can be shown that $D_K(P,Q) = \|\mu_P - \mu_Q\|$ for vectors μ_P and μ_Q in an appropriate reproducing kernel Hilbert space (RKHS). Such distances are popular in machine learning; see Sriperumbudur et al. (2009), for example.

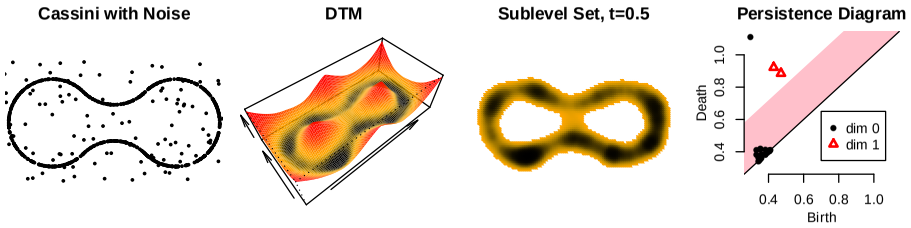


Figure 6: The left plot shows a sample from the Cassini curve together with a few outliers. The second plot is the empirical DTM. The third plot is one sub-level set of the DTM. The last plot is the persistence diagram. Points not in the shaded band are significant features. Thus, this method detects one significant connected component and two significant loops in the sublevel set filtration of the empirical DTM function.

Given a sample $X_1, \ldots, X_n \sim P$, let P_n be the probability measure that puts mass $1/n$ on each X_i. Let ϑ_x be the Dirac measure that puts mass one on x. Phillips et al. (2014) suggest using the discrete kernel distance

$$\widehat{D}_K(x) \equiv D_K(P_n, \vartheta_x) = \sqrt{\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} K(X_i, X_j) + K(x,x) - \frac{2}{n}\sum_{i=1}^{n} K(x, X_i)} \tag{36}$$

for topological inference. This is an estimate of the population quantity

$$D_K(x) \equiv D_K(P, \vartheta_x) = \sqrt{\iint K(z,y)dP(z)dP(y) + K(x,x) - 2\int K(x,y)dP(y)}.$$

The most common choice of kernel is the Gaussian kernel $K(x,y) \equiv K_h(x,y) = \exp\left(-\frac{\|x-y\|^2}{2h^2}\right)$, which has one tuning parameter h. We recall that, in topological inference, we generally do not let h tend to zero. The reason is that topological features can be detected with $h > 0$ and keeping h bounded way from 0 reduces the variance of the estimator. See the related discussion in Section 4.4 of Fasy et al. (2014b).

Recall that the kernel density estimator is defined by

$$\widehat{p}_h(x) = \frac{1}{n(\sqrt{2\pi}h)^d}\sum_i K(x, X_i).$$

Let $p_h(x) = \mathbb{E}[\widehat{p}_h(x)]$. We see that

$$\begin{aligned}
\widehat{D}_K^2(x) &= h^d\left(\frac{(\sqrt{2\pi})^d}{n}\sum_i \widehat{p}_h(X_i) + h^{-d}K(0,0) - 2(\sqrt{2\pi})^d\widehat{p}_h(x)\right) \\
&= h^d\left(\frac{(\sqrt{2\pi})^d}{n}\sum_i [\widehat{p}_h(X_i) - p_h(X_i)] + \frac{(\sqrt{2\pi})^d}{n}\sum_i p_h(X_i) + h^{-d}K(0,0) - 2(\sqrt{2\pi})^d\widehat{p}_h(x)\right) \\
&= (\sqrt{2\pi})^d h^d(c + o_P(1)) + O_P\left(\sqrt{\frac{\log n}{n}}\right) + K(0,0) - 2(\sqrt{2\pi}h)^d\widehat{p}_h(x).
\end{aligned}$$

Here, we used the fact that $n^{-1}\sum_{i=1}^n p_h(X_i) = c + o_P(1)$ and $||\widehat{p}_h - p_h||_\infty = O_P(\sqrt{\log n/n})$ where $c = \int p_h p$.

We see that up to small order terms, the sublevel sets of $D_K(x)$ are a rescaled version of the super-level sets of the kernel density estimator. Hence, the kernel distance approach and the density estimator approach are essentially the same, up to a rescaling. However, D_K^2 has some nice properties; see Phillips et al. (2014).

The limiting properties of $\widehat{D}_K^2(x)$ follow immediately from well-known properties of kernel density estimators. In fact, the conditions needed for $\widehat{D}_K^2$ are weaker than for the DTM.

Theorem 16 (Limiting Behavior of Kernel Distance) *We have that*

$$\sqrt{n}(\widehat{D}_K^2 - D_K^2) \rightsquigarrow \mathbb{B},$$

where $\mathbb{B}$ is a Brownian bridge. The bootstrap version converges to the same limit, conditionally almost surely.

The proof of the above theorem is based on the aforementioned equivalence of D_K to the rescaled density function and the well known fact that $\sqrt{n}(\widehat{p}_h(x) - p_h(x))$ converges to a Brownian bridge. This theorem justifies using the bootstrap to construct L_∞ bands for $p_h = \mathbb{E}(\widehat{p}_h)$ or D_K.

As we mentioned before, for topological inference, we keep the bandwidth h fixed. Thus, it is important to keep in mind that we view $\widehat{p}_h$ as an estimate of $p_h(x) = \mathbb{E}[\widehat{p}_h(x)] = \int K_h(x,u)dP(u)$.

6. The Bottleneck Bootstrap

More precise inferences can be obtained by directly bootstrapping the persistence diagram. Let $X_1^*, \ldots, X_n^*$ be as before a sample from the empirical measure P_n and let $\widehat{D}^*$ be the (random) persistence diagram defined on this point cloud. Define $\widehat{t}_\alpha$ by

$$\mathbb{P}(\sqrt{n}W_\infty(\widehat{D}^*, \widehat{D}) > \widehat{t}_\alpha \,|\, X_1, \ldots, X_n) = \alpha. \tag{37}$$

The quantile $\widehat{t}_\alpha$ can be estimated by Monte Carlo. We then use a band of size $2\widehat{t}_\alpha$ on the diagram D.

In the following, we show that $\widehat{t}_\alpha$ consistently estimates the population value t_α defined by

$$\mathbb{P}(\sqrt{n}W_\infty(\widehat{D}, D) > t_\alpha) = \alpha. \tag{38}$$

The reason why the bottleneck bootstrap can lead to more precise inferences than the functional bootstrap from the previous section is that the functional bootstrap uses the fact that $W_\infty(\widehat{D}, D) \leq ||\widehat{\delta} - \delta||_\infty$ and finds an upper bound for $||\widehat{\delta} - \delta||_\infty$. But in many cases the inequality is not sharp so the confidence set can be very conservative. Moreover, we can obtain different critical values for different dimensions (connected components, loops, voids, ...) and so the inferences are tuned to the specific features we are estimating. See Figure 7.

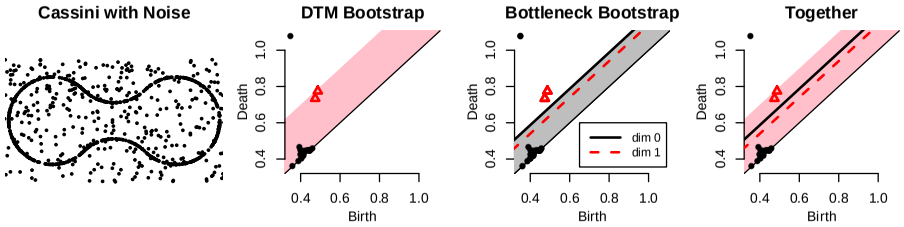


Figure 7: The left plot shows a sample from the Cassini curve together with a few outliers. The second plot shows the DTM persistence diagram with a 95% confidence band constructed using the method of Section 4. The third plot shows the same persistence diagram with two 95% confidence bands constructed using the bottleneck bootstrap with zero-dimensional features and one-dimensional features. The fourth plot shows the three confidence bands at the same time. In Section 8, we use this compact form to show multiple confidence bands.

Although the bottleneck bootstrap can be used with either the DTM or the KDE, we shall only prove its validity for the KDE. First, we need the following result. For any function p, let $g = \nabla p$ denote its gradient and let $H = \nabla^2 p$ denotes its Hessian. We say that x is a critical point if $g(x) = (0, \ldots, 0)^T$. We then call $p(x)$ a critical value. A function is Morse if the Hessian is non-degenerate at each critical point. The Morse index of a critical point x is the number of negative eigenvalues of $H(x)$.

Lemma 17 (Stability of Critical Points) *Let p be a density with compact support S. Assume that S is a d-dimensional compact submanifold of $\mathbb{R}^d$ with boundary. Assume p is a Morse function with finitely many, distinct, critical values with corresponding critical points $c_1, \ldots, c_k$. Also assume that p is of class $\mathcal{C}^2$ on the interior of S, continuous and differentiable with non vanishing gradient on the boundary of S. Then, there exist $\epsilon_0 > 0$ and $c > 0$ such that for all $0 < \epsilon < \epsilon_0$, there exists $\eta \geq c\epsilon$ such that, for any density q with*

support S satisfying

$$\sup_x |p(x) - q(x)| < \eta, \qquad \sup_x |\nabla p(x) - \nabla q(x)| < \eta, \qquad \sup_x |\nabla^2 p(x) - \nabla^2 q(x)| < \eta,$$

q is a Morse function with exactly k critical points $c'_1, \ldots, c'_k$ say, and, after a suitable re-labeling of indices,

$$\max_j ||c_j - c'_j|| \leq \epsilon.$$

Moreover, c_j and c'_j have the same Morse index.

Proof This lemma is a consequence of classical stability properties of Morse functions. First, from Theorem 5.31, p.140 in Banyaga and Hurtubise (2004) and Proposition II.2.2, p.79 in Golubitsky and Guillemin (1986), there exists $\epsilon_1 > 0$ such that if q is at distance less than ϵ_1 in the $\mathcal{C}^2$ topology (i.e. such that the sup-norm of $p - q$ and its first and second derivatives are bounded by ϵ_1) then q is a Morse function. Moreover, there exist two diffeomorphisms $h : \mathbb{R} \to \mathbb{R}$ and $\phi : S \to S$ such that $q = h \circ p \circ \phi$. As the notion of critical point and of index are invariant by diffeomorphism, p and q have the same number of critical points with same index. More precisely, the critical points of q are the points $c'_i = \phi^{-1}(c_i)$.

Now let $\epsilon > 0$ be small enough such that $2\epsilon < \min_{i \neq j} \|c_i - c_j\|$, and for any $i \neq j$, $p(B(c_i, \epsilon)) \cap p(B(c_j, \epsilon)) = \emptyset$. Then $\eta_1 = \eta_1(\epsilon) = \min_{i \neq j} d(p(B(c_i, \epsilon)), p(B(c_j, \epsilon)))$ where $d(A, B) = \min_{a \in A, b \in B} |a - b|$ and $\eta_2 = \eta_2(\epsilon) = \inf\{\|\nabla p(x)\| : x \in S \setminus \cup_{i=1}^k B(c_i, \epsilon)\}$ are both positive. If q satisfies the assumptions of the lemma for any $0 < \eta \leq \min(\eta_1, \eta_2)$, then the critical values of q have to be in $\cup_i p(B(c_i, \epsilon))$ and the critical points c'_i have to be in $\cup_i B(c_i, \epsilon)$.

More precisely, notice that since p is a Morse function, for ϵ small enough, $\eta_2 = O(\epsilon)$, and, for any $i \in \{1, \cdots, k\}$, the Taylor series of ∇p about c_j yields

$$\nabla p(x) = H_i(x - c_i) + \|x - c_i\| r(x - c_i),$$

where $r(z) \to 0$ as $\|z\| \to 0$ and H_i is the Hessian of p at c_i. Let $\lambda_{\min}$ be the smallest absolute eigenvalue of the Hessians at all the critical points. Since p is a Morse function, the matrix H_i is full rank and $\lambda_{\min}$ is positive. As a consequence, for all $x \in S \setminus \cup_{i=1}^k B(c_i, \epsilon)$ and ϵ small enough, $\|\nabla p(x)\| \geq \frac{\lambda_{\min}}{2}\epsilon$. Since η_1 is a non-increasing function of ϵ, we have that, for ϵ small enough, $\eta = \eta_2 \geq \frac{\lambda_{\min}}{2}\epsilon$.

To conclude the proof of the lemma, we need to prove that each ball $B(c_i, \epsilon)$ contains exactly one critical point of q. Indeed, for $t \in [0, 1]$, the functions $q_t(x) = p(x) + t(q(x) - p(x))$ are Morse functions satisfying the same properties as q. Now, since each c_i is a non-degenerate point of p, it follows from the continuity of the critical points (see, e.g. Prop. 4.6.1 in Demazure (2013)) that, restricting ϵ if necessary, there exist smooth functions $c_i : [0, 1] \to S$, $c_i(0) = c_i, c_i(1) = c'_i$ such that $c_i(t)$ is the unique critical point of q_t in $B(c_i, \epsilon)$. Moreover, since all the q_t are Morse functions and since the Hessian of q_t at $c_i(t)$ is a continuous function of t, then for any $t \in [0, 1]$, $c_i(t)$ is a non-degenerate critical point of q_t with same index as c_i. ■

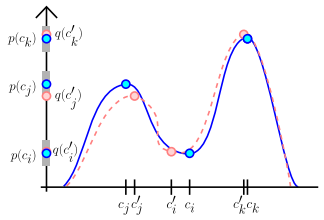


Figure 8: This figure illustrates the assumptions of Lemma 18. The functions p and q are shown in solid blue and dashed pink, respectively. The grey regions on the y-axis represent the sets $p(c) \pm b$ for critical points c of p.

Consider now two smooth functions such that the critical points are close, as illustrated in Figure 8. Next we show that, in this circumstance, the bottleneck distance takes a simple form.

Lemma 18 (Critical Distances) *Let p and q be two Morse functions as in Lemma 17, with finitely many critical points $C = \{c_1, \ldots, c_k\}$ and $C' = \{c'_1, \ldots, c'_k\}$ respectively. Let D_p and D_q be the persistence diagrams from the upper level set (i.e. super level sets) filtrations of p and q respectively and let $a = \min_{i \neq j} |p(c_i) - p(c_j)|$ and $b = \max_j |p(c_j) - q(c'_j)|$. If $b \leq a/2 - ||p - q||_\infty$ and $a/2 > 2||p - q||_\infty$, then $W_\infty(D_p, D_q) = b$.*

Proof The topology of the upper level sets of the Morse functions p and q only changes at critical values (Theorem 3.1 in Milnor (1963)). As a consequence the non-diagonal points of D_p (resp. D_q) have their coordinates among the set $\{p(c_1), \ldots, p(c_k)\}$ (resp. $\{q(c'_1), \ldots, p(c'_k)\}$) and each $p(c_i)$ is the coordinate of exactly one point in D_p. Moreover, the pairwise distances between the points of D_p are lower bounded by a and all non-diagonal points of D_p are at distance at least a from the diagonal. From the persistence stability theorem Cohen-Steiner et al. (2005); Chazal et al. (2012), $W_\infty(D_p, D_q) \leq ||p - q||_\infty$. Since $a > 4||p - q||_\infty$ and $a \geq 2b + 2||p - q||_\infty$, the (unique) optimal matching realizing the bottleneck distance $W_\infty(D_p, D_q)$ is such that if $(p(c_i), p(c_j)) \in D_p$ then it is matched to the point $(q(c'_i), q(c'_j))$ which thus have to be in D_q. It follows that $W_\infty(D_p, D_q) = b$. ■

Now we establish the limiting distribution of $\sqrt{n} W_\infty(\widehat{D}, D)$.

Theorem 19 (Limiting Distribution) *Let $p_h(x) = \mathbb{E}[\widehat{p}_h(x)]$, where $\widehat{p}_h(x)$ is the Kernel Density Estimator evaluated in x. Assume that p_h is a Morse function with two uniformly bounded continuous derivatives and finitely many critical points $c = \{c_1, \ldots, c_k\}$. Let D be the persistence diagram of the upper level sets of p_h and let $\widehat{D}$ be the diagram of upper level sets of $\widehat{p}_h$. Then*

$$\sqrt{n} W_\infty(\widehat{D}, D) \rightsquigarrow ||Z||_\infty$$

where $Z = (Z_1, \ldots, Z_k) \sim N(0, \Sigma)$ and

$$\Sigma_{jk} = \int K_h(c_j, u) K_h(c_k, u) dP(u) - \int K_h(c_j, u) dP(u) \int K_h(c_k, u) dP(u).$$

Proof Let $\widehat{c} = \{\widehat{c}_1, \widehat{c}_2, \ldots\}$ be the set of critical points of $\widehat{p}_h$. Let g and H be the gradient and Hessian of p_h. Let $\widehat{g}$ and $\widehat{H}$ be the gradient and Hessian of $\widehat{p}_h$. By a standard concentration of measure argument (and recalling that the support is compact), for any $\eta > 0$ there is an event $A_{n,\eta}$ such that, on $A_{n,\eta}$,

$$\sup_x ||\widehat{p}_h^{(i)}(x) - p_h^{(i)}(x)|| < \eta \tag{39}$$

for $i = 0, 1, 2$, and $\mathbb{P}(A_{n,\eta}^c) \leq e^{-nc\eta^2}$. This is proved for $i = 0$ in Rao (1983), Giné and Guillou (2002), Yukich (1985), and the same proof gives the results for $i = 1, 2$. It follows that $\sup_x ||g(x) - \widehat{g}(x)|| = O_P(1/\sqrt{n})$ and $\sup_x ||H(x) - \widehat{H}(x)||_{\max} = O_P(1/\sqrt{n})$.

For η smaller than a fixed value η_0, we can apply Lemma 17, we get that on $A_{n,\eta}$, $\widehat{c}$ and c have the same number of elements and can be indexed so that

$$\max_{j=1,\ldots,k} \|\widehat{c}_j - c_j\| \leq \frac{\eta}{C}$$

where C is the same constant is in Lemma 17. We then take $\eta_n := \sqrt{\frac{\log n}{n}}$ and we consider the events $A_n := A_{n,\eta_n}$. Then, for n large enough, on A_n we get

$$\max_{j=1,\ldots,k} \|\widehat{c}_j - c_j\| = O\left(\sqrt{\frac{\log n}{n}}\right)$$

whereas $P(A_n^c) = o(1)$. In the following, we thus can restrict to A_n.

The critical values of p_h are $v = (v_1 \equiv p_h(c_1), \ldots, v_k \equiv p_h(c_k))$ and the critical values of $\widehat{p}_h$ are $\widehat{v} = (\widehat{v}_1 \equiv \widehat{p}_h(\widehat{c}_1), \ldots, \widehat{v}_k \equiv \widehat{p}_h(\widehat{c}_k))$. Now we use Lemma 18 to conclude that $W_\infty(\widehat{D}, D) = \max_j \|\widehat{v}_j - v_j\|_\infty$ for n large enough. Hence,

$$W_\infty(\widehat{D}, D) = \max_{j=1,\ldots,k} |\widehat{p}_h(\widehat{c}_j) - p_h(c_j)|.$$

Then, using a Taylor expansion, for each j,

$$\widehat{p}_h(\widehat{c}_j) = \widehat{p}_h(c_j) + (\widehat{c}_j - c_j)^T \widehat{g}(c_j) + O(||\widehat{c}_j - c_j||^2).$$

Since $g(c_j) = (0, \ldots, 0)$ we can write the last equation as

$$\widehat{p}_h(\widehat{c}_j) = \widehat{p}_h(c_j) + (\widehat{c}_j - c_j)^T (\widehat{g}(c_j) - g(c_j)) + O(||\widehat{c}_j - c_j||^2).$$

So,

$$\begin{aligned}
\sqrt{n}(\widehat{v}_j - v_j) &= \sqrt{n}(\widehat{p}_h(\widehat{c}_j) - p_h(c_j)) \\
&= \sqrt{n}(\widehat{p}_h(c_j) - p_h(c_j)) + \sqrt{n}(\widehat{c}_j - c_j)^T (\widehat{g}(c_j) - g(c_j)) + O(||\widehat{c}_j - c_j||^2) \\
&= \sqrt{n}(\widehat{p}_h(c_j) - p_h(c_j)) + \sqrt{n}(\widehat{c}_j - c_j)^T (\widehat{g}(c_j) - g(c_j)) + o(1/\sqrt{n}).
\end{aligned}$$

For the second term, note that $\sqrt{n}(\widehat{c}_j - c_j) = O(\log n)$ and $(\widehat{g}(c_j) - g(c_j)) = O_P(1/\sqrt{n})$. So

$$\sqrt{n}(\widehat{v}_j - v_j) = \sqrt{n}(\widehat{p}_h(c_j) - p_h(c_j)) + o_P(1).$$

Therefore,

$$\sqrt{n}W_\infty(\widehat{D}, D) = \sqrt{n}\max_j |\widehat{v}_j - v_j| = \max_j |\sqrt{n}(\widehat{p}_h(c_j) - p_h(c_j))| + o_P(1).$$

By the multivariate Berry-Esseen theorem (Bentkus, 2003),

$$\sup_A |\mathbb{P}(\sqrt{n}(\widehat{p}_h(c) - p_h(c)) \in A) - \mathbb{P}(Z \in A)| \leq \frac{C_1}{\sqrt{n}}$$

where the supremum is over all convex sets $A \in \mathbb{R}^k$, $C_1 > 0$ depends on k and the third moment of $h^{-d}K(x - X/h)$ (which is finite since h is fixed and the support is compact), $Z = (Z_1, \ldots, Z_k) \sim N(0, \Sigma)$ and

$$\Sigma_{jk} = \int K_h(c_j, u)K_h(c_k, u)dP(u) - \int K_h(c_j, u)dP(u) \int K_h(c_k, u)dP(u).$$

Hence,

$$\sup_t \left| \mathbb{P}\Big(\max_j |\sqrt{n}(\widehat{p}_h(c_j) - p_h(c_j))| \leq t\Big) - \mathbb{P}(||Z||_\infty \leq t) \right| \leq \frac{C_1}{\sqrt{n}}.$$

By Lemma 18, $W_\infty(\widehat{D}, D) = \max_j |\widehat{v}_j - v_j|$. The result follows. ■

Let

$$\widehat{F}_n(t) = \mathbb{P}(\sqrt{n}W_\infty(\widehat{D}, D) \leq t).$$

Let $X_1^*, \ldots, X_n^* \sim P_n$ where P_n is the empirical distribution. Let $\widehat{D}^*$ be the diagram from $\widehat{p}_h^*$ and let

$$\widehat{F}_n(t) = \mathbb{P}\Big(\sqrt{n}W_\infty(\widehat{D}^*, \widehat{D}) \leq t \;\Big|\; X_1, \ldots, X_n\Big)$$

be the bootstrap approximation to F_n.

Next we show that the bootstrap quantity $F_n(t)$ converges to the same limit as $F_n(t)$.

Corollary 20 *Assume the same conditions as the last theorem. Then,*

$$\sup_t |\widehat{F}_n(t) - F_n(t)| \xrightarrow{P} 0.$$

Proof The proof is essentially the same as the proof of Theorem 19 except that $\widehat{p}_h$ replaces p_h and $\widehat{p}_h^*$ replaces $\widehat{p}_h$. Using the same notations as in the proof of Theorem 19, we note that on the set A_n, for n larger than a fixed value n_0, the function $\widehat{p}_h$ is a Morse function with two uniformly bounded continuous derivatives and finitely many critical points $\widehat{c} = \{\widehat{c}_1, \ldots, \widehat{c}_k\}$. We can restrict the analysis to the sequence of events A_n since $P(A_n)$ tends

to zero. Assuming that A_n is satisfied, using the same argument as in Theorem 19, we get that:

$$\sup_t \left| \mathbb{P}\Big(\max_j |\sqrt{n}(\widehat{p}_h^*(\widehat{c}_j) - \widehat{p}_h(\widehat{c}_j))| \leq t \;\Big|\; X_1, \ldots, X_n\Big) - \mathbb{P}(||\widetilde{Z}||_\infty \leq t) \right| \leq \frac{C_2^*}{\sqrt{n}}$$

where $\widetilde{Z} \sim N(0, \widehat{\Sigma})$ with

$$\widehat{\Sigma}_{jk} = \frac{1}{n}\sum_i K_h(\widehat{c}_j, X_i)K_h(\widehat{c}_k, X_i) - \frac{1}{n}\sum_i K_h(\widehat{c}_j, X_i)\frac{1}{n}\sum K_h(\widehat{c}_k, X_i)$$

and C_2^* depends on the empirical third moments of $h^{-d}K((x - X^*)/h)$. There exists an upper bound C_2 on C_2^* that only depends on K and P. Since $\max_{j,k} |\widehat{\Sigma}_{j,k} - \Sigma_{j,k}| = O_P(\log n/\sqrt{n})$ and $\max_j \|\widehat{c}_j - c_j\| = O_P(\log n/\sqrt{n})$, we conclude that

$$\sup_t |\mathbb{P}(||\widetilde{Z}||_\infty \leq t) - \mathbb{P}(||Z||_\infty \leq t)| = O_P\left(\frac{\log n}{\sqrt{n}}\right).$$

Then

$$\sup_t |\widehat{F}_n(t) - F_n(t)| \leq \sup_t |\widehat{F}_n(t) - \mathbb{P}(||\widetilde{Z}||_\infty \leq t)| + \sup_t |\mathbb{P}(||\widetilde{Z}||_\infty \leq t) - \mathbb{P}(||Z||_\infty \leq t)| + \sup_t |F_n(t) - \mathbb{P}(||Z||_\infty \leq t)| = O_P\left(\frac{\log n}{\sqrt{n}}\right).$$

The result follows. ■

7. Extensions

In this section, we discuss how to deal with three issues that can arise: choosing the parameters, correcting for boundary bias, and dealing with noisy data.

7.1 A Method for Choosing the Smoothing Parameter

An unsolved problem in topological inference is how to choose the smoothing parameter m (or h). Guibas et al. (2013) suggested tracking the evolution of the persistence of the homological features as the tuning parameter varies. Here we make this method more formal, by selecting the parameter that maximizes the total amount of significant persistence.

Let $\ell_1(m), \ell_2(m), \ldots,$ be the lifetimes of the features at scale m. Let $c_\alpha(m)/\sqrt{n}$ be the significance cutoff at scale m. We define two quantities that measure the amount of significant information using parameter m:

$$N(m) = \#\left\{i : \ \ell(i) > \frac{c_\alpha(m)}{\sqrt{n}}\right\}, \quad S(m) = \sum_i \left[\ell_i - \frac{c_\alpha(m)}{\sqrt{n}}\right]_+.$$

These measures are small when m is small since $c_\alpha(m)$ is large. On the other hand, they are small when m is large since then all the features are smoothed out. Thus we have a kind of topological bias-variance trade-off. We choose m to maximize $N(m)$ or $S(m)$. The same idea can be applied to the kernel distance and kernel density estimator. See the example in Figure 9.

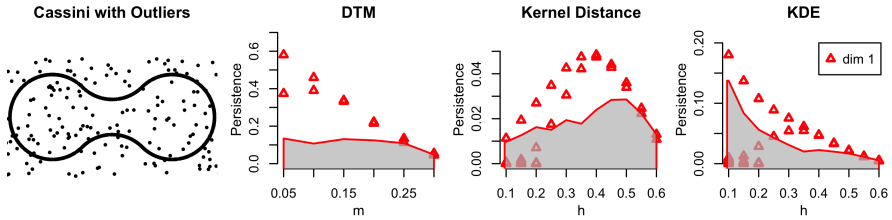


Figure 9: Max Persistence Method with Bottleneck Bootstrap Bands for 1-dimensional features. DTM: $\text{argmax}_m N(m) = \{0.05, 0.10, 0.15, 0.20\}$, $\text{argmax}_m S(m) = 0.05$; Kernel Distance: $\text{argmax}_h N(h) = \{0.25, 0.30, 0.35, 0.40, 0.45, 0.50\}$, $\text{argmax}_h S(h) = 0.35$; KDE: $\text{argmax}_h N(h) = \{0.25, 0.30, 0.35, 0.40, 0.45, 0.50\}$, $\text{argmax}_h S(h) = 0.3$ The plots show how to choose the smoothing parameters to maximize the number of significant features. The red triangles are the lifetimes of the features versus the tuning parameter.The red line is the significance cutoff.

7.2 Boundary Bias

It is well known that kernel density estimators suffer from boundary bias. For topological inference, this bias manifests itself in a particular form and the same problem affects the DTM. Consider Figure 10. Because of the bounding box, many of the loops are incomplete. The result is that, using either the DTM or the KDE we will miss many of the loops.

There is a large literature on reducing boundary bias in the kernel density estimation literature. Perhaps the simplest approach is to reflect the data around the boundaries (see for example Schuster (1958)). But there is a simpler fix for topological inference: we merely need to close the loops at the boundary. This can be done by adding points uniformly around the boundary.

7.3 Two Methods for Improving Performance

We can improve the performance of all the methods if we can mitigate the outliers and noise. Here we suggest two methods to do this. We focus on the kernel density estimator.

First, a simple method to reduce the number of outliers is to truncate the density, that is, we eliminate $\{X_i : \widehat{p}(X_i) < t\}$ for some threshold t. Then we re-estimate the density.

Secondly, we sharpen the data as described in Choi and Hall (1999) and Hall and Minnotte (2002). The idea of sharpening is to move each data point X_i slightly in the direction of the gradient $\nabla\widehat{p}(X_i)$ and then re-estimate the density. The authors show that this reduces the bias at peaks in the density which should make it easier to find topological features. It can be seen that the sharpening method amounts to running one or more steps if the mean-shift algorithm. This is a gradient ascent which is intended to find modes of the density estimator. Given a point x, we move x to

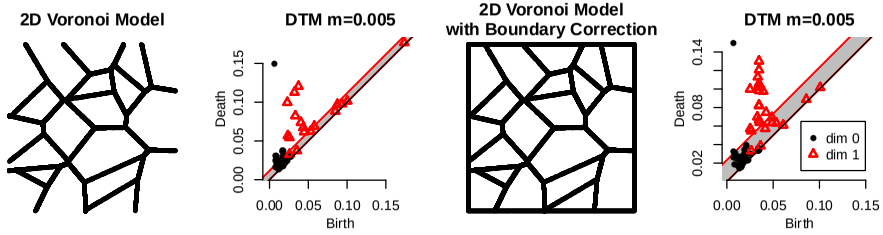


Figure 10: First: 10,000 points sampled from a 2D Voronoi model with 20 nuclei. Second: the corresponding persistence diagram of sublevel sets of the distance to measure function. Note that only 9 loops are detected as significant. Third: 2,000 points have been added on the boundary of the square delimiting the Voronoi model. Fourth: now the corresponding persistence diagram shows 16 significant loops.

$$\frac{\sum_i X_i K_h(x, X_i)}{\sum_i K_h(x, X_i)},$$

which is simply the local average centered at x. For data sharpening, we do one (or a few) iterations of this to each data point X_i. Then the density is re-estimated.
In fact, we could also use the subspace constrained mean shift algorithm (SCMS) which moves points towards ridges of the density; see Ozertem and Erdogmus (2011).
Figure 11 shows these methods applied to a simple example.

8. Examples

Example 1 (Noisy Grid) *The data in Figure 12 are 10,000 data points on a 2D grid. We add Gaussian noise plus 1,000 outliers and compute the persistence diagrams of Kernel Density Estimator, Kernel distance, and Distance to Measure. The pink bands show 95% confidence sets obtained by bootstrapping the corresponding functions. The black lines show 95% confidence bands obtained with the bottleneck bootstrap for dimension 0, while the red lines show 95% confidence bands obtained with the bottleneck bootstrap for dimension 1. The Distance to Measure, which is less sensitive to the density of the points, correctly captures the topology of the data. The Kernel Distance and KDE find some extra significant connected component, corresponding to high density regions at the intersection of the grid.*

Example 2 (Soccer) *Figure 13 shows the field position of two soccer players. The data come from body-sensor traces collected during a professional soccer game in late 2013 at the Alfheim Stadium in Tromso, Norway. The data are sampled at 20 Hz. See Pettersen et al. (2014). Although the data is a function observed over time, we treat it as a point cloud. Points on the boundary of the field have been added to avoid boundary bias. The DTM*

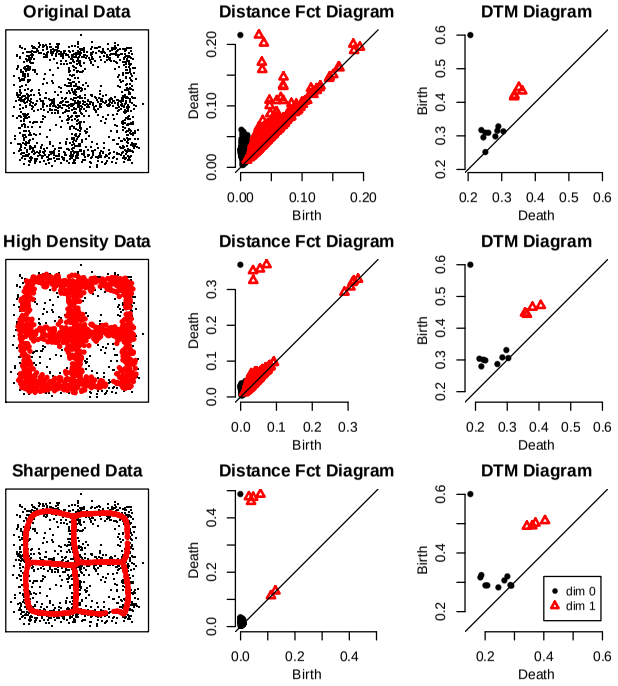


Figure 11: Top: 1,300 points sampled along a 2×2 grid with Gaussian noise; the diagram of the distance function shows many loops due to noise. Middle: the red points are the high density data (density > 0.15); the corresponding diagram of the distance function correctly captures the 4 loops, plus a few features with short lifetime. Bottom: the red points represent the sharpened high density data; now most of the noise in the corresponding diagram is eliminated. Note that the diagram of the distance to measure function does a good job with the original data. The bottom left plot shows a slight improvement, in the sense that the persistence of the 4 loops has increased.

captures the difference between the two players: the defender leaves one big portion of the filed uncovered (1 significant loop in the persistence diagram), while the midfielder does not cover the 4 corners (4 significant loops). Nonetheless, the Kernel distance, which is more sensible to the density of these points, fails to detect significant topological features.

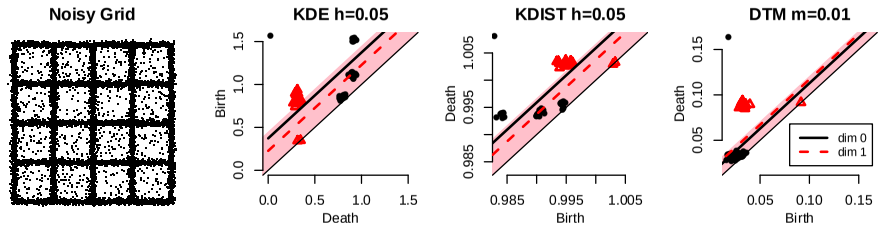


Figure 12: 10,000 data points on a 2D grid and the corresponding persistence diagrams of Kernel Density Estimator, Kernel distance, and Distance to Measure. For more details see Example 1.

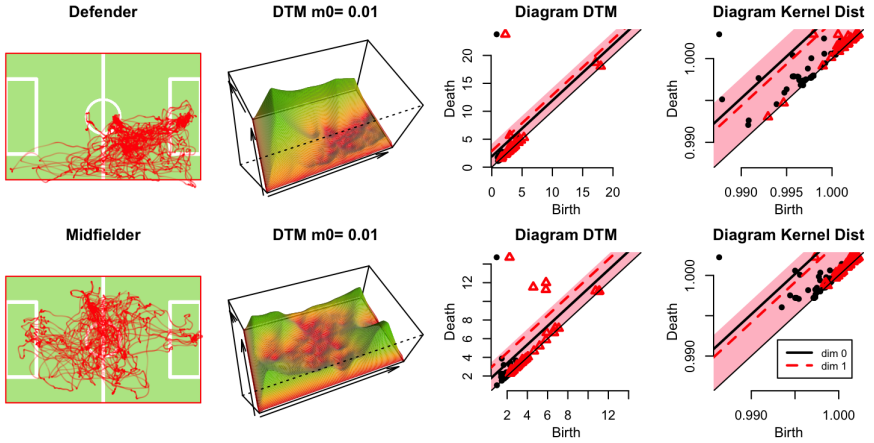


Figure 13: Top: data for a defender. We show the DTM, the digram for the DTM and the digram for the kernel distance. Bottom: same but for a midfielder. The midfielder data has more loops.

Example 3 (Voronoi Models) *Given k points (nuclei) $\{z_1, \ldots, z_k\} \subset \mathbb{R}^3$, let the Voronoi region R_k be $R_k = \left\{x \in \mathbb{R}^3 : \|x - z_k\| \leq \|x - x_j\| \text{ for all } j \neq k\right\}$. The Voronoi regions $R_1, \ldots, R_k$ partition the space, forming what is known as the Voronoi diagram. A face is formed by the intersection of 2 adjacent Voronoi regions; a line is formed at the intersection of two faces and a node is formed at the intersection of two or more lines.*

We will sample points around the the nodes, lines and faces that are formed at the intersection of the Voronoi regions. A Voronoi wall model is a sampling scheme that returns points within or around the Voronoi faces. Similarly, by sampling points exclusively around the lines or exclusively around the nodes, we can construct Voronoi filament models and Voronoi cluster models.

These models were introduced by Icke and van de Weygaert (1991) to mimic key features of cosmological data; see also van de Weygaert et al. (2011).

In this example we generate data from filament models and wall models using the basic definition of Voronoi diagram, computed on a fine grid in $[0, 50]^3$. *We also add random Gaussian noise. See Figure 14: the first two rows show 100K particles concentrated around the filaments of 8 and 64 Voronoi cells, respectively. The last two rows show 100K particles concentrated around the walls of 8 and 64 Voronoi cells. 60K points on the boundary of the boxes have been added to mitigate boundary bias. For each model we present the persistent diagrams of the distance function, distance to measure and kernel density estimator. We chose the smoothing parameters by maximizing the quantity* $S(\cdot)$, *defined in Section 7.1.*

The diagrams illustrate the evolution of the filtrations for the three different functions: at first, the connected components appear (black points in the diagrams); then they merge forming loops (red triangle), that eventually evolve into 3D voids (blue squares).

The persistence diagrams of the three functions allow us to distinguish the different models (see Figure 1 for a less trivial example) and the confidence bands, generated using the bootstrap method of Section 4.1, allow us to separate the topological signal from the topological noise. In general, the DTM performs better than the KDE, which is more affected by the high density of points around the nodes and filaments. For instance, this is very clear in the third row of Figure 14. The DTM diagram correctly captures the topology of the Voronoi wall model with 8 nuclei: one connected component and 8 voids are significant, while the remaining homological features fall into the band and are classified as noise.

9. Discussion

In this paper, we showed how the DTM and KDE can be used for robust topological inference. Further, we showed how to use the bootstrap to identify topological features that are distinguishable from noise. We conclude by discussing two issues: comparing DTM and KDE, and using persistent homology versus selecting a single level set.

9.1 Comparison of DTM and Kernel Distance

The DTM and the KDE have the same broad aim: to provide a means for extracting topological features from data. However, these two methods are really focused on different goals. Consider again the model $P = \pi R + (1 - \pi)(Q \star \Phi_\sigma)$ and let S be the support of Q. As before, we assume that S is a "small set" meaning that either it has dimension $k < d$ or that it is full dimensional but has small Lebesgue measure. When π and σ are small, the persistent homology of the upper level sets of the density p will be dominated by features corresponding to the homology of S. In other words, we are using the persistent homology of $\{p > t\}$ to learn about the homology of S. In contrast, the DTM is aimed at estimating the persistent homology of S. Both are useful, but they have slightly different goals.

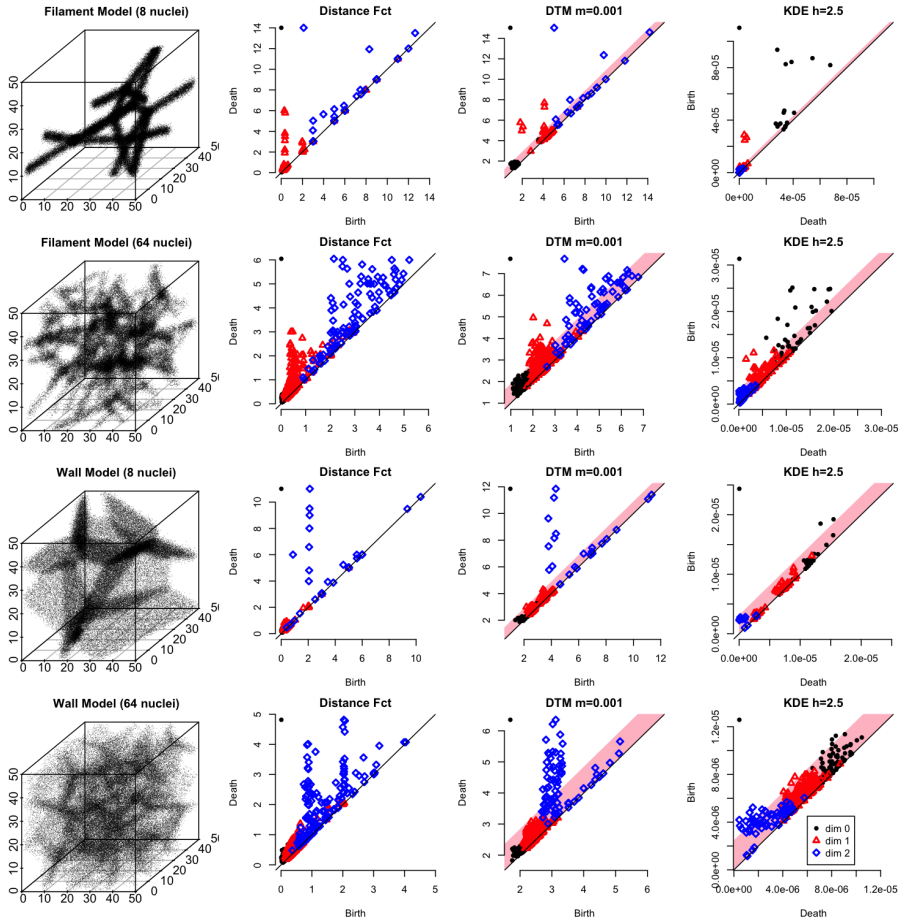


Figure 14: Data from four Voronoi foam models. In each case we show the diagrams of the distance function, the DTM and the KDE. A boundary correction was included.

This also raises the intriguing idea of extracting more information from both the KDE and DTM by varying more parameters. For example, if we look at the sets $\{p_h > t\}$ for fixed t but varying h, we get information very similar to that of the DTM. Conversely, for

the DTM, we can vary the tuning parameter m. There are many possibilities here which we will investigate in future work.

9.2 Persistent Homology Versus Choosing One Level Set

We have used the persistent homology of the upper level sets $\{\widehat{p}_h > t\}$ to probe the homology of S. This is the approach used in Bubenik (2015) and Phillips et al. (2014).

Bobrowski et al (2014) suggest a different approach. They select a particular level set $\{p > t\}$ and they form a robust estimate of the homology of this one level set. They have a data-driven method for selecting t. (This approach is only one part of the paper. They also consider persistent homology.)

They make two key assumptions. The first is that there exists $A < B$ such that $\{p > t\}$ is homotopic to S for all $A < t < B$. (If two sets are homotopic, then they have the same homology.) This is a very reasonable assumption. In the mixture model $P = \pi R + (1 - \pi)(Q \star \Phi_\sigma)$ this assumption will be satisfied when S is a small set and when π and σ are small. In this case, persistent homology will also work well: the dominant features in the persistence diagram will correspond to the homology of S.

Bobrowski et al (2014) make an additional assumption. They assume that the dimension k of S is known and that the rank of the k^{th} homology group is 0 for all $t > B$. This assumption is critical for their approach to choosing a single level set. Currently, it is not clear how strong this assumption is. In future work, we plan to compare the robustness of the single-level approach versus persistent homology.

9.3 Future Work

Lastly, we would like to mention that several issues deserve future attention. In particular, the methods we discussed for choosing the tuning parameters, for mitigating boundary bias and for sharpening the data, all deserve further investigation.

In a companion paper we will show how the ideas presented in this work can be used to develop hypothesis tests for comparing point clouds.

Acknowledgments

The authors are grateful to Jérome Dedecker for pointing out the key decomposition (18) of the DTM. The authors also would like to thank Jessi Cisewski and Jisu Kim for their comments and two referees for helpful suggestions. We would like to acknowledge support for this project from ANR-13-BS01-0008, NSF CAREER Grant DMS 1149677, Air Force Grant FA95500910373 and NSF Grant DMS-0806009.

References

A. Banyaga and D. Hurtubise. *Lectures on Morse Homology*. Kluwer Academic Publishers, 2004.

Vidmantas Bentkus. On the dependence of the berry–esseen bound on dimension. *Journal of Statistical Planning and Inference*, 113(2):385–402, 2003.

Gérard Biau, Frédéric Chazal, David Cohen-Steiner, Luc Devroye, Carlos Rodriguez, et al. A weighted k-nearest neighbor density estimate for geometric inference. *Electronic Journal of Statistics*, 5:204–237, 2011.

S. Bobkov and M. Ledoux. One-dimensional empirical measures, order statistics and Kantorovich transport distances. *Preprint*, 2014.

Omer Bobrowski, Sayan Mukherjee, and Jonathan Taylor. Topological consistency via kernel estimation. *arXiv preprint arXiv:1407.5272*, 2014.

Peter Bubenik. Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 16:77–102, 2015.

Mickaël Buchet, Frédéric Chazal, Steve Y Oudot, and Donald R Sheehy. Efficient and robust topological data analysis on metric spaces. *arXiv preprint arXiv:1306.0039*, 2013.

Claire Caillerie, Frédéric Chazal, Jérôme Dedecker, and Bertrand Michel. Deconvolution for the wasserstein metric and geometric inference. *Electronic Journal of Statistics*, 5: 1394–1423, 2011.

Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46 (2):255–308, 2009.

F. Chazal, D. Cohen-Steiner, M. Glisse, L.J. Guibas, and S.Y. Oudot. Proximity of persistence modules and their diagrams. In *SCG*, pages 237–246, 2009. ISBN 978-1-60558-501-7. doi: http://doi.acm.org/10.1145/1542362.1542407.

F. Chazal, V. de Silva, M. Glisse, and S. Oudot. The structure and stability of persistence modules. *arXiv preprint arXiv:1207.3674*, 2012.

Frédéric Chazal, David Cohen-Steiner, and Quentin Mérigot. Geometric inference for probability measures. *Foundations of Computational Mathematics*, 11(6):733–751, 2011.

Frédéric Chazal, Pascal Massart, and Bertrand Michel. Rates of convergence for robust geometric inference. Technical report, ArXiv preprint 1505.07602, 2015.

D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. In *SCG*, pages 263–271, 2005.

Antonio Cuevas and Alberto Rodríguez-Casal. On boundary estimation. *Advances in Applied Probability*, 36(2):340–354, 2004.

M. Demazure. *Bifurcations and Catastrophes: Geometry of Solutions to Nonlinear Problems*. Springer-Verlag, 2013.

Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. American Mathematical Society, 2010.

Brittany Terese Fasy, Jisu Kim, Fabrizio Lecci, and Clement Maria. Introduction to the R package TDA. *arXiv preprint arXiv: 1411.1830*, 2014a.

Brittany Terese Fasy, Fabrizio Lecci, Alessandro Rinaldo, Larry Wasserman, Sivaraman Balakrishnan, and Aarti Singh. Confidence sets for persistence diagrams. *The Annals of Statistics*, 42(6):2301–2339, 2014b.

Evarist Giné and Armelle Guillou. Rates of strong uniform consistency for multivariate kernel density estimators. In *Annales de l'Institut Henri Poincare (B) Probability and Statistics*, volume 38, pages 907–921. Elsevier, 2002.

M. Golubitsky and V. Guillemin. *Stable Mappings and Their Singularities*. Springer-Verlag, 1986.

Leonidas Guibas, Dmitriy Morozov, and Quentin Mérigot. Witnessed k-distance. *Discrete & Computational Geometry*, 49(1):22–45, 2013.

Vincent Icke and Rien van de Weygaert. The galaxy distribution as a voronoi foam. *Quarterly Journal of the Royal Astronomical Society*, 32:85–112, 1991.

J/. Milnor. *Morse Theory*. Number 51. Princeton University Press, 1963.

Umut Ozertem and Deniz Erdogmus. Locally defined principal curves and surfaces. *The Journal of Machine Learning Research*, 12:1249–1286, 2011.

Svein Arne Pettersen, Dag Johansen, Håvard Johansen, Vegard Berg-Johansen, Vamsidhar Reddy Gaddam, Asgeir Mortensen, Ragnar Langseth, Carsten Griwodz, Håkon Kvale Stensland, and Pål Halvorsen. Soccer video and player position dataset. In *Proceedings of the 5th ACM Multimedia Systems Conference*, pages 18–23. ACM, 2014.

Jeff M. Phillips, Bei Wang, and Yan Zheng. Geometric inference on kernel density estimates. *arXiv preprint arXiv:1307.7760*, 2014.

B. L. S. Prakasa Rao. *Nonparametric Functional Estimation*. Probability and Mathematical Statistics. Academic Press, Orlando, FL, 1983.

E. Schuster. Incorporating support constraints into nonparametric estimators of densities. *Communications in Statistics*, A(14):1123–1136, 1958.

Galen R Shorack and Jon A Wellner. *Empirical processes with applications to statistics*, volume 59. SIAM, 2009.

Bharath K Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Gert RG Lanckriet, and Bernhard Schölkopf. Kernel choice and classifiability for rkhs embeddings of probability distributions. In *NIPS*, pages 1750–1758, 2009.

Rien van de Weygaert, Gert Vegter, Herbert Edelsbrunner, Bernard JT Jones, Pratyush Pranav, Changbom Park, Wojciech A Hellwing, Bob Eldering, Nico Kruithof, EGP Bos, et al. Alpha, Betti and the megaparsec universe: On the topology of the cosmic web. In *Transactions on Computational Science XIV*, pages 60–101. Springer-Verlag, 2011.

Aad W. van der Vaart. *Asymptotic Statistics*, volume 3. Cambridge UP, 2000.

Aad W. van der Vaart and Jon A. Wellner. *Weak Convergence*. Springer, 1996.

JE Yukich. Laws of large numbers for classes of functions. *Journal of multivariate analysis*, 17(3):245–260, 1985.

Journal of Machine Learning Research 18 (2018) 1-25 Submitted 9/15; Revised 1/18; Published 5/18

Training Gaussian Mixture Models at Scale via Coresets

Mario Lucic LUCIC@INF.ETHZ.CH
Department of Computer Science
ETH Zurich
Universitätstrasse 6, 8092 Zürich, Switzerland

Matthew Faulkner MNFAULK@GMAIL.COM
Department of Electrical Engineering and Computer Sciences
Caltech
1200 E California Blvd, Pasadena, California 91125

Andreas Krause KRAUSEA@ETHZ.CH
Department of Computer Science
ETH Zurich
Universitätstrasse 6, 8092 Zürich, Switzerland

Dan Feldman DANNYF.POST@GMAIL.COM
Department of Computer Science
University of Haifa
199 Aba Khoushy Ave. Mount Carmel, Haifa, Israel

Editor: Sanjoy Dasgupta

Abstract

How can we train a statistical mixture model on a massive data set? In this work we show how to construct *coresets* for mixtures of Gaussians. A coreset is a weighted subset of the data, which guarantees that models fitting the coreset also provide a good fit for the original data set. We show that, perhaps surprisingly, Gaussian mixtures admit coresets of size polynomial in dimension and the number of mixture components, while being *independent* of the data set size. Hence, one can harness computationally intensive algorithms to compute a good approximation on a significantly smaller data set. More importantly, such coresets can be efficiently constructed both in distributed and streaming settings and do not impose restrictions on the data generating process. Our results rely on a novel reduction of statistical estimation to problems in computational geometry and new combinatorial complexity results for mixtures of Gaussians. Empirical evaluation on several real-world data sets suggests that our coreset-based approach enables significant reduction in training-time with negligible approximation error.

Keywords: Gaussian mixture models, coresets, streaming and distributed computation

1. Introduction

We consider the problem of training statistical mixture models, in particular mixtures of Gaussians on massive data sets. In contrast to parameter estimation for models with compact sufficient statistics, mixture models generally require inference over latent variables, which in turn depends on the full data set. Such data sets are often distributed across a cluster of machines, or arrive in a data stream, and have to be processed with limited memory. In this paper, we show that Gaussian mixture models (GMMs) admit small *coresets*: A weighted subset of the data which guarantees that models fitting the coreset will also provide a good fit for the original data set. As a result, solving the estimation problem on the coreset $\mathcal{C}$ is almost as good as solving the estimation problem on the large data set $\mathcal{X}$. Critically, we show that the size of these coresets is *independent* of the size of the data set.

We focus on training mixtures of λ-semi-spherical Gaussians, where the covariance matrix Σ_i of each component $i \in [k]$ has eigenvalues bounded in $[\lambda, 1/\lambda]$. More formally, given a data set $\mathcal{X}$ of n points in $\mathbb{R}^d$, some $\varepsilon > 0$ and an integer $k \geq 1$, one can efficiently construct a weighted set $\mathcal{C} \subseteq \mathcal{X}$ of $\Theta\left(d^4 k^6 \varepsilon^{-2} \lambda^{-4}\right)$ points, such that for any mixture of k λ-semi-spherical Gaussians $\theta = [(w_i, \mu_i, \Sigma_i)]_{i=1}^k$ it holds that the log-likelihood $\ln P(\mathcal{X} \mid \theta)$ of $\mathcal{X}$ under θ is approximated by the (properly weighted) log-likelihood $\ln P(\mathcal{C} \mid \theta)$ of $\mathcal{C}$ under θ to arbitrary accuracy as $\varepsilon \to 0$. Moreover, these coresets can be efficiently constructed in parallel (using a merge-reduce style computation), as well as in the streaming setting using space and update time per point polynomial in d, k, λ^{-1}, ε^{-1}, $\log n$ and $\log(1/\delta)$.

2. Background and Problem Statement

We first discuss the problem of parameter estimation of Gaussian mixture models by maximum likelihood estimation. We then turn our attention to the problem of approximating the log-likelihood using a weighted subset of the data set and formally define the desired coreset property.

2.1 Fitting Gaussian Mixture Models by Maximum Likelihood Estimation

Given a data set $\mathcal{X} = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ we consider fitting a mixture of Gaussians $\theta = [(w_1, \mu_1, \Sigma_1), \ldots, (w_k, \mu_k, \Sigma_k)]$, that is, the distribution

$$P(x \mid \theta) = \sum_{i=1}^{k} w_i \mathcal{N}(x; \mu_i, \Sigma_i)$$

where $w_1, \ldots, w_k \geq 0$ are the mixture weights and $\sum_i w_i = 1$. The i-th mixture component is modelled as a multivariate Normal distribution parametrized by mean $\mu_i \in \mathbb{R}^d$ and covariance $\Sigma_i \in \mathbb{R}^{d \times d}$,

$$\mathcal{N}(x; \mu_i, \Sigma_i) = \frac{1}{\sqrt{|2\pi\Sigma_i|}} \exp\left(-\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)\right).$$

Assuming the data was generated i.i.d., the negative log-likelihood of the data is

$$\mathcal{L}(\mathcal{X} \mid \theta) = -\sum_j \ln P(x_j \mid \theta),$$

and we wish to obtain the maximum likelihood estimate (MLE) of the parameters

$$\theta^* = \operatorname*{argmin}_{\theta \in \mathfrak{C}} \mathcal{L}(\mathcal{X} \mid \theta),$$

where $\mathfrak{C}$ is a set of constraints ensuring that degenerate solutions are avoided. Hereby, for a symmetric matrix $\mathbf{A}$, let $\mathrm{spec}(\mathbf{A})$ be the set of all eigenvalues of $\mathbf{A}$. We define $\mathfrak{C} = \mathfrak{C}_\lambda = \{\theta = [(w_i, \mu_i, \Sigma_i)]_{i=1}^k \mid \forall_i : \mathrm{spec}(\Sigma_i) \subseteq [\lambda, 1/\lambda]\}$ for some small $\lambda \in (0, 1)$.

2.2 Approximating the Log-likelihood

Ideally, we would like to obtain $(1+\varepsilon)$-multiplicative approximation for the likelihood

$$\prod_{x\in\mathcal{X}} P(x \mid \theta)$$

which implies an additive ε error for the sum of log-likelihoods. What kind of approximation accuracy may we hope to expect? Notice that there is a non-trivial issue of scale: Suppose we have a MLE θ^* for $\mathcal{X}$, and let $\alpha > 0$. Then straightforward linear algebra shows that we can obtain a MLE θ^*_α for a scaled data set $\alpha D = \{\alpha x : x \in \mathcal{X}\}$ by simply scaling all means by α, and covariance matrices by α^2. For the log-likelihood, however, it holds that $\frac{1}{n}\mathcal{L}(\alpha D \mid \theta^*_\alpha) = d\ln\alpha + \frac{1}{n}\mathcal{L}(\mathcal{X} \mid \theta^*)$. Therefore, optimal solutions on one scale can be efficiently transformed to optimal solutions on a different scale, while maintaining the same *additive error*. Thus, we cannot expect to obtain a $(1+\varepsilon)$-multiplicative approximation to the likelihood since any algorithm which achieves absolute error ε at any scale could be used to compute parameter estimates (for means, covariances) with arbitrarily small error, simply by applying the algorithm to a scaled data set and transforming back the obtained solution.

An alternative, scale-invariant approach, may be to strive towards a *multiplicative error* $(1+\varepsilon)$ for the sum of log-likelihoods. Unfortunately, this goal is also hard to achieve: Choosing a scaling parameter α such that $d\ln\alpha + \mathcal{L}(\mathcal{X} \mid \theta^*) = 0$ would require any algorithm that achieves any bounded multiplicative error to essentially incur *no error at all* when evaluating $\mathcal{L}(\alpha\mathcal{X} \mid \theta^*)$. The above observations hold even for the case $k = 1$ and $\Sigma = I$, where the mixture θ consists of a single Gaussian, and the log-likelihood is the sum of squared distances to a point μ and an additive term.

Motivated by the scaling issues discussed above, our goal is to approximate the data set $\mathcal{X}$ by a weighted set $C = \{(\gamma_1, \mathbf{x}'_1), \ldots, (\gamma_m, \mathbf{x}'_m)\} \subseteq \mathbb{R}_+ \times \mathbb{R}^d$ such that $\mathcal{L}(\mathcal{X} \mid \theta) \approx \mathcal{L}(\mathcal{C} \mid \theta)$ for all $\theta \in \mathfrak{C}_\lambda$, where we define

$$\mathcal{L}(\mathcal{C} \mid \theta) = -\sum_i \gamma_i \ln P(\mathbf{x}'_i \mid \theta).$$

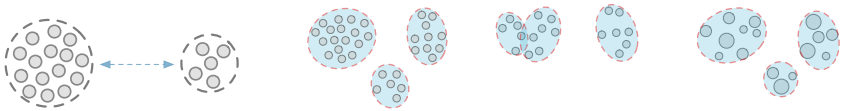
The key idea is to decompose the negative log-likelihood into a data-dependent term, and a data-independent term, with the goal of approximating the latter with a coreset. To this end, we apply the following decomposition suggested by Arora and Kannan (2005):

$$\begin{aligned}\mathcal{L}(\mathcal{X} \mid \theta) &= -\sum_{j=1}^{n} \ln \sum_{i=1}^{k} \frac{w_i}{\sqrt{|2\pi\Sigma_i|}} \exp\left(-\frac{1}{2}(x_j-\mu_i)^T\Sigma_i^{-1}(x_j-\mu_i)\right) \\ &= -n\ln Z(\theta) + \operatorname{cost}(\mathcal{X},\theta),\end{aligned}$$

where $Z(\theta) = \sum_i \frac{w_i}{\sqrt{|2\pi\Sigma_i|}}$ is a normalizer, and the function cost is defined as

$$\operatorname{cost}(\mathcal{X},\theta) = -\sum_{j=1}^{n} \ln \sum_{i=1}^{k} \frac{w_i}{Z(\theta)\sqrt{|2\pi\Sigma_i|}} \exp\left(-\frac{1}{2}(x_j-\mu_i)^T\Sigma_i^{-1}(x_j-\mu_i)\right).$$

Hereby, $Z(\theta)$ is a normalizing term which can be computed *exactly* and independently of the set $\mathcal{X}$. Furthermore, function $\operatorname{cost}(\mathcal{X},\theta)$ captures all dependencies of $\mathcal{L}(\mathcal{X} \mid \theta)$ on $\mathcal{X}$.



(a) Simple failure case (b) Full data set (c) Uniform sample (d) Coreset

Figure 1: Figure (a) illustrates a simple example of well-separated Gaussians for which uniform subsampling fails arbitrarily badly. Consider two spherical Gaussian mixtures with weights $w_1 = 1/\sqrt{n}$ and $w_2 = 1 - 1/\sqrt{n}$. Unless the number of samples $m \in \Theta(\sqrt{n})$, the uniform sample will consist only of points from the first Gaussian, with high probability. Hence, moving the means of the Gaussians apart, the difference in MLEs on the full data set and the one on the uniform subsample can be made arbitrarily high. Figure (b) shows a GMM with 3 components fit on the full data set, (c) the model fit on the uniform subsample, and (d) the model fit on a coreset. The uniform subsample is likely to miss small clusters in presence of unbalanced data.

2.3 Coresets for Semi-spherical Gaussian Mixtures

We proceed by showing that it suffices to approximate $\operatorname{cost}(\mathcal{X},\theta)$ uniformly over $\theta \in \mathfrak{C}$.

Definition 1 *We call a weighted data set $\mathcal{C}$ a (k,ε)-*coreset *for another (possibly weighted) set $\mathcal{X} \subset \mathbb{R}^d$, if for all mixtures $\theta \in \mathfrak{C}$ of k Gaussians it holds that*

$$(1-\varepsilon)\operatorname{cost}(\mathcal{X},\theta) \leq \operatorname{cost}(\mathcal{C},\theta) \leq (1+\varepsilon)\operatorname{cost}(\mathcal{X},\theta).$$

Hereby, $\operatorname{cost}(\mathcal{C},\theta)$ is generalized to weighted data sets $\mathcal{C}$ in the natural way (weighing the contribution of each summand $\mathbf{x}'_j \in \mathcal{C}$ by its weight γ_j). Thus, as $\varepsilon \to 0$, for a sequence of (k,ε)-coresets $\mathcal{C}_\varepsilon$ we have that

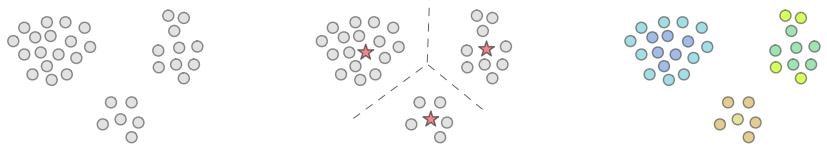
$$\sup_{\theta\in\mathfrak{C}} |\mathcal{L}(\mathcal{C}_\varepsilon \mid \theta) - \mathcal{L}(\mathcal{X} \mid \theta)| = \sup_{\theta\in\mathfrak{C}} |\operatorname{cost}(\mathcal{C}_\varepsilon,\theta) - \operatorname{cost}(\mathcal{X} \mid \theta)| \to 0$$

which implies that $\mathcal{L}(\mathcal{C}_\varepsilon \mid \theta)$ uniformly approximates $\mathcal{L}(\mathcal{X} \mid \theta)$ (over $\theta \in \mathfrak{C}$).

The main motivation for constructing a (k,ε)-coreset $\mathcal{C}$ is to reduce the problem of fitting a mixture model on $\mathcal{X}$ to one of fitting a model on $\mathcal{C}$, since the optimal solution $\theta_\mathcal{C}$ is a good approximation (in terms of log-likelihood) of θ^*. While finding the optimal $\theta_\mathcal{C}$ is a difficult problem, one can use a (weighted) variant of the EM algorithm to find a good solution. Moreover, if $|\mathcal{C}| \ll |\mathcal{X}|$, running EM on $\mathcal{C}$ is orders of magnitude faster than running EM on $\mathcal{X}$.

3. Efficient Coreset Construction

We start by contrasting the coreset approach with the "naive" approach of fitting the model on a uniform subsample. We show that the uniform subsampling approach can perform arbitrarily badly, while explicit worst-case guarantees can be given for the coreset based approach. We then present a simple coreset construction algorithm and conclude with a bound on the sufficient coreset size.



(a) Data set (b) k-means approximation (c) Distribution $q(x)$

Figure 2: Illustration of the coreset construction on a synthetic data set. Figure (a) shows the original data set and (b) the k-means bicriteria approximation. Figure (c) illustrates the computed sensitivity (blue-low, orange-high). The coreset sampling probabilities are inversely proportional to the size of the cluster which results in more representative samples.

3.1 Naive Approach via Uniform Sampling

A naive approach towards approximating the log-likelihood of $\mathcal{X}$ is to just pick a subset $\mathcal{C}$ uniformly at random. Unfortunately, such a simple strategy may result in arbitrary bad approximations of the log-likelihood as illustrated by the following example. Suppose the data set is generated from a mixture of two spherical Gaussians ($\Sigma_i = \mathbf{I}$) with weights $w_1 = \frac{1}{\sqrt{n}}$ and $w_2 = 1 - \frac{1}{\sqrt{n}}$. Unless $m = \Omega(\sqrt{n})$ points are sampled, with constant probability no data point generated from the second Gaussian is sampled. By moving the means of the Gaussians apart, $\mathcal{L}(\mathcal{X} \mid \theta_{\mathcal{C}})$ can be made arbitrarily worse than $\mathcal{L}(\mathcal{X} \mid \theta_{\mathcal{X}})$, where $\theta_{\mathcal{C}}$ and $\theta_{\mathcal{X}}$ are MLEs on $\mathcal{C}$ and $\mathcal{X}$ respectively. Thus, even for two well-separated Gaussians, uniform subsampling can perform arbitrarily poorly which is illustrated in Figure 2. The problem becomes even more pronounced as k and d grow.

This example already suggests that we must devise a sampling scheme that adaptively selects representative points from all "clusters" present in the data set. However, this implies that obtaining a coreset requires solving a chicken-and-egg problem, where we need to understand the density of the data to obtain the coreset, but simultaneously would like to use the coreset for density estimation.

3.2 Better Approximation via Importance Sampling

The key idea behind the coreset construction is that we can break the chicken-and-egg problem by first obtaining a rough approximation of the problem on $\mathcal{X}$ and use it to construct a non-uniform sampling scheme. This non-uniform sampling can be understood as an importance-weighted estimate of the log-likelihood $\mathcal{L}(\mathcal{X} \mid \theta)$, where the weights are optimized in order to reduce the variance. Feldman and Langberg (2011) successfully used the same idea to construct coresets for geometric clustering problems.

The critical insight is that, even though we seek to fit GMMs, it suffices to consider approximate solutions to the k-means clustering of $\mathcal{X}$ to construct a good sampling strategy. Intuitively, such a solution partitions $\mathcal{X}$ into Voronoi cells for which we can compute the density and the variance. Hence, one can bias the sampling towards less dense regions of $\mathcal{X}$. We prove that the resulting sampling strategy yields valid coresets in Appendix A.3. Our main result is the following Theorem which establishes the sufficient sample size which

Algorithm 1 Coreset

require: Data set $\mathcal{X}$, bicriteria approximation $\mathcal{B}$, approximation factor α, coreset size m.
for $j \leftarrow 1$ **to** $|\mathcal{B}|$
  $\mathcal{X}_j \leftarrow$ Set of points from $\mathcal{X}$ closest to point $\mathcal{B}_j$. Ties may be broken arbitrarily.
for $j \leftarrow 1$ **to** $|\mathcal{B}|$, **for each** $x \in \mathcal{X}_j$
  $s(x) \leftarrow \alpha \mathrm{d}(x, \mathcal{B})^2 + \frac{2\alpha}{|\mathcal{X}_j|} \sum_{x' \in \mathcal{X}_j} \mathrm{d}(x', \mathcal{B})^2 + \frac{2}{|\mathcal{X}_j|} \sum_{x' \in \mathcal{X}} \mathrm{d}(x', \mathcal{B})^2$
for each $x \in \mathcal{X}$
  $q(x) \leftarrow \frac{s(x)}{\sum_{x' \in \mathcal{X}} s(x')}$
$\mathcal{C} \leftarrow$ Sample m weighted points from $\mathcal{X}$, where each point x is sampled with probability $q(x)$ and assigned a weight $\frac{1}{m \cdot q(x)}$.
return $\mathcal{C}$

Algorithm 2 K-Means++

require: Data set $\mathcal{X}$, number of clusters k.
$\mathcal{B} \leftarrow \{$Sample $x \in \mathcal{X}$ uniformly at random$\}$
for $j \leftarrow 2$ **to** k
  for x **in** $\mathcal{X}$
    $p(x) \leftarrow \frac{\mathrm{d}^2(x, \mathcal{B})}{\sum_{x' \in \mathcal{X}} \mathrm{d}^2(x', \mathcal{B})}$
  Sample $x \in \mathcal{X}$ with probability $p(x)$ and add it to $\mathcal{B}$.
return $\mathcal{B}$

Algorithm 3 Adaptive sampling

require: $\mathcal{X}$, k, failure probability δ.
$R \leftarrow D, \mathcal{B} \leftarrow \emptyset, c \leftarrow \lceil 10dk \ln(1/\delta) \rceil$
while $|R| > c$
  $S \leftarrow$ Sample c points uniformly from R
  $P \leftarrow \lceil |R|/2 \rceil$ points from R closest to S
  $R \leftarrow R \setminus P$
  $\mathcal{B} \leftarrow \mathcal{B} \cup S$
$\mathcal{B} \leftarrow \mathcal{B} \cup R$
return $\mathcal{B}$

is independent of n, and only polynomial in k, d and ε. For clarity, we define $\mathrm{d}(x, \mathcal{B}) = \min_{b \in \mathcal{B}} ||x - b||$ and $\phi(\mathcal{X}, \mathcal{B}) = \sum_{x \in \mathcal{X}} \mathrm{d}(x, \mathcal{B})^2$.

Theorem 2 *Let $\mathcal{X} \subset \mathbb{R}^d$, $\delta \in (0,1)$, $\varepsilon \in (0, 1/2)$, $k \in \mathbb{N}$, and $\lambda \in (0,1)$. Let $\mathcal{B}_1, \ldots, \mathcal{B}_p$ be the outputs of $p = \lceil \log_2 1/\delta \rceil$ independent runs of Algorithm 2 with input $\mathcal{X}$ and k. Let $\mathcal{C}$ be the output of Algorithm 1 with $\alpha = 16(\log_2 k + 2)$, $\mathcal{B}^\star = \operatorname{argmin}_{i \in \{1,\ldots,p\}} \phi(\mathcal{X}, \mathcal{B}_i)$ and coreset size*

$$m \geq c \cdot \frac{d^4 k^6 + k^2 \log \frac{1}{\delta}}{\lambda^4 \varepsilon^2},$$

where $c > 0$ is an absolute constant. Then, with probability at least $1 - \delta$, the set $\mathcal{C}$ is a (k, ε)-coreset of $\mathcal{X}$.

To establish the result we first reduce the MLE problem to the Euclidean space where the log-likelihood contribution of a data point given a model $\theta \in \mathfrak{C}$ can be expressed in a purely geometric manner. We then apply a coreset construction framework introduced by Feldman and Langberg (2011), which formalizes the intuition that one should sample the points with a potentially large impact more often. We then show that to compute the sampling distribution $q(x)$ it suffices to consider a rough approximation so the k-means clustering of $\mathcal{X}$. To guarantee uniform convergence over $\mathfrak{C}$ we bound the combinatorial complexity of the family of functions induced by the MLE of GMMs and show that it has a polynomial dependency on k and d. The full proof is presented in the Appendix.

Several implementation choices are available. Firstly, we prove that one can use any (α, β)-*bicriteria solution* to k-means (α approximate with respect to the optimal k-means clustering, using βk centers). The suggested algorithm, `k-means++`, provides a solution $\mathcal{B}$ of size k with approximation $\mathcal{O}(\log_2 k)$ in expectation in time $\mathcal{O}(nkd)$ and results in coresets of size *independent* of the data set size (Theorem 2). For other bicriteria approximation algorithms offering different tradeoffs in terms of computational complexity and the approximation guarantee we refer the reader to Bachem et al. (2016a,b), Makarychev et al. (2016) and Feldman and Langberg (2011) presented in Algorithm 3. Furthermore, all steps of the algorithm are parallelizable. The bicriteria approximation can be computed via k-means|| (Bahmani et al., 2012) and other quantities (i.e. $s(x)$, $q(x)$) can be computed in parallel. Finally, the importance sampling step can be implemented to run in constant-time per point with linear-time preprocessing (Vose, 1991).

4. Fitting a GMM on the Coreset using Weighted EM

Once the coreset $\mathcal{C}$ is constructed, we need to fit a mixture model that takes into account the point weights. Since the coreset size is independent of the cardinality of $\mathcal{X}$ we can (at least from the perspective of developing a polynomial time algorithm) afford to use a more "expensive" method. In geometric clustering problems such as k-means or k-median, where data points are hard-assigned to the closest cluster (point, subspace, etc.), it is possible to find the *optimal* clustering via exhaustive search, by simply considering all possible partitions of the coreset, and picking the best one. This procedure – constructing a coreset of size independent of n, and then using exhaustive search on the coreset – yields a (randomized) polynomial time approximation scheme (PTAS): It is guaranteed to achieve multiplicative error $1+\varepsilon$, in time which is polynomial in n, but exponential in all other quantities (in particular $1/\varepsilon$). For mixture models, this exhaustive search algorithm is not feasible, since points are not hard-assigned to a cluster, but "soft-assigned" (according to the cluster membership probabilities). One approach, which we employ in our experiments, is to use a natural generalization of the EM algorithm, which takes the coreset weights into account. The details are presented in Algorithm 4 and the derivation of the EM update equations is presented in Appendix B.

Since the EM algorithm is applied on a significantly smaller data set, it can be initialized using multiple random restarts. In our experiments, we show that running weighted EM on the coreset typically leads to comparable performance (in terms of log-likelihood) as running EM on the full data set. A practical issue when using EM to fit a Gaussian mixture model is to ensure non-degeneracy which occurs when the MLE estimate for the (co)variance is zero implying infinite density and infinite log-likelihood (Bishop, 2006). The generally accepted remedy is to constrain the variances to be greater than some small apriori chosen value $\lambda > 0$ which is the strategy we employ in the experimental evaluation (Bishop, 2006). The alternative is to consider a fully Bayesian framework whereby one introduces a prior on the covariance matrices which induces large penalties for small values of the (co)variance in the resulting *maximum a posteriori* estimation problem (Murphy, 2012).

Algorithm 4 EM for GMMs

require: Data set $\mathcal{X}$, point weights γ, number of components k, prior threshold λ.
$\eta \leftarrow$ Weighted-K-Means$(\mathcal{X}, \gamma, k)$ ▷ $\eta_{ij} = 1$ iff x_i was assigned to cluster j.
$w, \mu, \Sigma \leftarrow$ Maximization$(\mathcal{X}, \eta, \lambda)$
while not converged
  $\eta \leftarrow$ Expectation$(\mathcal{X}, \gamma, w, \mu, \Sigma)$
  $w, \mu, \Sigma \leftarrow$ Maximization$(\mathcal{X}, \eta, \lambda)$
return w, μ, Σ

Algorithm 5 Expectation

require: Data set $\mathcal{X}$, point weights γ, component weights w, means μ, covariances Σ.
$z \leftarrow 0_n$
for $i \in [n]$, $j \in [k]$
  $\eta_{ij} \leftarrow w_j \mathcal{N}(x_i \mid \mu_j, \Sigma_j)$
  $z_i \leftarrow z_i + \eta_{ij}$
for $i \in [n]$, $j \in [k]$
  $\eta_{ij} \leftarrow \gamma_i \eta_{ij} \,/\, z_i$
return η

Algorithm 6 Maximization

require: $\mathcal{X}$, responsibilities η, threshold λ.
$z \leftarrow 0_k, \mu \leftarrow 0_{k \times d}, \Sigma \leftarrow 0_{k \times d \times d}$
for $j \in [k], i \in [n]$
  $z_j \leftarrow z_j + \eta_{ij}$
  $\mu_j \leftarrow \mu_j + \eta_{ij} x_j$
  $\Sigma_j \leftarrow \Sigma_j + \eta_{ij}(x_i - \mu_j)(x_i - \mu_j)^T$
for $j \in [k]$
  $w_j \leftarrow z_j / ||z||_1$
  $\mu_j \leftarrow \mu_j \,/\, z_j$
  $\Sigma_j \leftarrow \Sigma_j \,/\, z_j + I_\lambda$
return (w, μ, Σ)

5. Streaming and Parallel Computation

One advantage of coresets is that they can be constructed in parallel, as well as in a streaming setting where data points arrive one by one, and it is impossible to remember the entire data set due to memory constraints. The key insight is that coresets satisfy certain composition properties, which have previously been used by Har-Peled and Mazumdar (2004) for streaming and parallel construction of coresets for geometric clustering problems such as k-median and k-means.

1. Let $\mathcal{C}_1$ be a (k, ε)-coreset for $\mathcal{X}_1$, and $\mathcal{C}_2$ be a (k, ε)-coreset for $\mathcal{X}_2$. Then $\mathcal{C}_1 \cup \mathcal{C}_2$ is a (k, ε)-coreset for $\mathcal{X}_1 \cup \mathcal{X}_2$.
2. Let $\mathcal{C}$ be a (k, ε)-coreset for $\mathcal{X}$, and $\mathcal{C}'$ be a (k, δ)-coreset for $\mathcal{C}$. Then $\mathcal{C}'$ is a $(k, (1+\varepsilon)(1+\delta) - 1)$-coreset for $\mathcal{X}$.

5.1 Streaming Computation

In the streaming setting, we assume that points arrive one-by-one, but we do not have enough memory to remember the entire data set. Thus, we wish to maintain a coreset over time, while keeping only a small subset of $\mathcal{O}(\log n)$ coresets in memory, where n is the number of points seen. The idea is to construct and store in memory a coreset for every block of $\text{poly}(d, k, \lambda^{-1}, \varepsilon^{-1})$ consecutive points arriving in a stream (Bentley and Saxe, 1980; Har-Peled and Mazumdar, 2004). When we have two ε-coresets in memory we first merge them

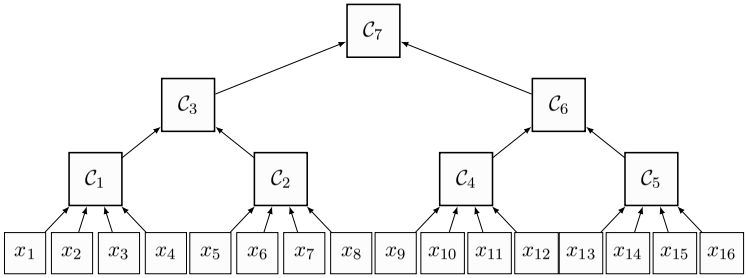


Figure 3: Coreset construction in the streaming setting. Black arrows indicate "merge-and-compress" operations. The (intermediate) coresets $\mathcal{C}_1, \ldots, \mathcal{C}_7$ are enumerated in the order in which they would be generated in the streaming case. In the parallel case, $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_4$ and $\mathcal{C}_5$ would be constructed in parallel, followed by parallel construction of $\mathcal{C}_3$ and $\mathcal{C}_6$, finally resulting in $\mathcal{C}_7$.

which results in a (k, ε)-coreset via property (1). To maintain a small size, we compress them by computing a single coreset from the merged coresets via property (1) which increases the error. A naive approach that merges and compresses immediately as soon as two coresets have been constructed can incur an exponential increase in approximation error. Fortunately, it is possible to organize the merge-and-compress operations in a binary tree of height $\mathcal{O}(\log n)$, where we need to store in memory a single coreset for each level of the tree (Feldman et al., 2013b, Theorem 10.1.).

Consider Figure 3 which illustrates the tree computation for an example data set. In the following, ε-coreset denotes a (ε, k)-coreset (k is fixed). In the first step, we construct a ε-coreset for $x_1, \ldots, x_4$. We then construct a ε-coreset for $x_5, \ldots, x_8$. At this point we have two ε-coresets and their union is, by property (1), a ε-coreset for $x_1, \ldots, x_8$. By property (2) a coreset $\mathcal{C}_3$ of the union of those two coresets is a 4ε-coreset for $x_1, \ldots, x_8$ since $(1+\varepsilon)^2 \leq 1 + 4\varepsilon$, for $0 \leq \varepsilon \leq 1$. Hence, we discard coresets $\mathcal{C}_1$ and $\mathcal{C}_2$ and keep only $\mathcal{C}_3$ in memory. We apply the same approach for $x_9, \ldots, x_{16}$ and obtain $\mathcal{C}_6$, a 4ε-coreset for $x_9, \ldots, x_{16}$. Once again, we obtained two coresets at the same level of the tree ($\mathcal{C}_3$ and $\mathcal{C}_6$) and merging them we obtain a 4ε-coreset for $x_1, \ldots, x_{16}$. By property (2), the coreset of the union of $\mathcal{C}_3$ and $\mathcal{C}_6$ is a 13ε-coreset for the whole data set since $(1+4\varepsilon)(1+\varepsilon) \leq 1+13\varepsilon$. Finally, only coreset C_7 is kept in memory.

In general, to ensure an error of ε, it suffices that the intermediate coreset error is bounded by $\varepsilon' = \frac{\varepsilon}{6 \log_2 n}$ as the height of the tree is at most $\lceil \log_2 n \rceil$ and $(1+\varepsilon')^{\lceil \log n \rceil} = (1 + \frac{\varepsilon}{6 \log n})^{\lceil \log n \rceil} \leq e^{\frac{\varepsilon}{6}} \leq 1 + \frac{\varepsilon}{3}.(1+\varepsilon')^{\lceil \log_2 n \rceil} \leq 1 + \frac{\varepsilon}{3}$. Thus, by property (2), a $(\varepsilon/3)$-coreset of the union of all coresets in memory has the approximation error bounded by $(1+\varepsilon/3)^2 \leq 1+\varepsilon$. As we do not know n *a priori*, we compute coresets for data batches of exponentially increasing size – the total time and space requirements are dominated by the last batch whose size is upper bounded by n (Feldman et al., 2013b).

Theorem 3 *A (k, ε)-coreset for a stream of n points in $\mathbb{R}^d$ can be computed for the λ-semi-spherical GMM using update time per point and memory* $\mathrm{poly}(d, k, \lambda^{-1}, \varepsilon^{-1}, \log n, \log(1/\delta))$ *with probability at least $1 - \delta$.*

In order to construct a coreset for the union of two (weighted) coresets, we use weighted versions of Algorithms 1 and 2, where we consider a weighted point as copies of a non-weighted point (possibly with fractional weight).

5.2 Distributed Computation

Using the same ideas from the streaming model, a (non-parallel) coreset construction can be transformed into a parallel one. We partition the data and compute a coreset for each partition independently. We then in parallel merge via property (1) two coresets, and compute a single coreset for every pair of such coresets exploiting the property (2). Continuing in this manner yields a process that takes $\mathcal{O}(\log n)$ iterations of parallel computation. This computation is naturally suited for map-reduce (Dean and Ghemawat, 2004) style computations, where the map tasks compute coresets for disjoint parts of $\mathcal{X}$, and the reduce tasks perform the merge-and-compress operations. Figure 3 illustrates this parallel construction.

Theorem 4 *A (k, ε)-coreset for a set of n points in $\mathbb{R}^d$ can be computed for the λ-semi-spherical GMM using m machines in time* $(n/m) \cdot \mathrm{poly}(d, k, \lambda^{-1}, \varepsilon^{-1}, \log(1/\delta), \log n)$ *with probability at least $1 - \delta$.*

Furthermore, if we have enough memory on one of the machines we can apply a simpler algorithm. First, partition the data to m machines and compute a $(\varepsilon/3)$-coreset on each, producing coresets $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m$. Then the union $C = \cup_{i=1}^m \mathcal{C}_i$ is a $(\varepsilon/3)$-coreset for the whole data set and its size is bounded by $m \cdot \max_i |\mathcal{C}_i|$. To obtain a coreset of size independent of the number of machines m, it suffices to construct a $(\varepsilon/3)$-coreset of $\mathcal{C}$.

Finally, Feldman and Tassa (2015) unified the streaming and distributed approaches when the data set size is unknown apriori. They propose splitting the input stream into several smaller streams whereby each machine proceeds to construct the coreset tree for its own stream.

6. Experimental Evaluation

The goal of this section is to demonstrate the effectiveness of using coresets for training Gaussian mixture models. To this end, we compare our coreset based approach to the "naive" approach of uniformly subsampling the data using models trained on the full data set as a baseline.

We construct coresets and uniform subsamples of sizes ranging from 1'000 to 10'000. For uniform subsampling, we subsample the data set and fit the model using Algorithm 4 where we set the weight of each point to one. For the coreset based approach with k mixture components we first construct a bicriteria approximation using Algorithm 2 and construct the coreset using Algorithm 1. Finally, we fit a GMM on the weighed coreset using Algorithm 4. We stop iterating between EM steps if the number of iterations is greater than 100, or the relative log-likelihood changed is smaller than 10^{-3} and apply prior thresholding with $\lambda = 0.001$ (Section 4).

We compare the median results from 200 runs for the subsampling methods and *best* out of 100 runs on the full data set. For each run we store the sampling time, solving time and the log-likelihood C on the test set. Finally, we calculate the *relative error*

$\eta = |(C - C_{full})/C_{full}|$ for both uniform subsampling and coresets. For each data set we use 80% of the data for training and the remaining 20% for computing the error. We perform the evaluation on four real-world data sets and summarize our observations as follows:

1. Higgs. Contains 11'000'000 instances describing signal processes which produce Higgs bosons and background processes which do not (Baldi et al., 2014). We consider the first two principal components components and fit GMMs with 150 components. For $m = 10'000$ the coreset based approach leads to a speedup of 57.5× with a relative error of 4%. At the same time, uniform subsampling leads to a relative error of 12.2%.

2. csn. Contains 80'000 instances with 17 features extracted from acceleration data recorded from volunteers carrying and operating their phone in normal conditions (Faulkner et al., 2011). We fit GMM with 100 components. For $m = 10'000$ the coreset based approach leads to a speedup of 7× with a relative error of 6.6%. At the same time, uniform subsampling leads to a relative error of 58.7%.

3. Kdd. Contains 145'000 instances with 74 features measuring the match between a protein and a native sequence. We fit GMMs with 10 components. For $m = 10'000$ the coreset based approach leads to a speedup of 8.3× with a relative error of 2.7%. At the same time, uniform subsampling leads to a relative error of 16.8%. We observe that models trained on the uniform subsample may be more sensitive to initialization.

4. msyp. Contains 515'345 instances with 90 features which represent timbre average and timbre covariance for mostly western, commercial tracks ranging from 1922–2011. We consider the top 25 principal components and fit GMMs with 50 components. For $m = 5'000$ the coreset based approach leads to a speedup of 78× with a relative error of 2.3%. At the same time, uniform subsampling leads to a relative error of 3.9%.

We observe that, for a fixed subsample size, coresets enjoy smaller approximation errors and obtain significant speedups with respect to solving the problem on the full data set.

7. Related Work

This paper is an extended version of Feldman et al. (2011) and provides significant improvements over the prior work. The theoretical analysis is executed directly on the negative log-likelihood function. This in turn leads to a novel importance sampling sampling strategy as well as a more practical algorithm with linear running time in n. Furthermore, we prove that one can use *any* bicriteria approximation for the k-means clustering problem as a basis for the importance sampling scheme. Overall, we are able to construct larger coresets in less time (approximately two orders of magnitude) and significantly improve the experimental results. To ensure the uniform convergence over $\mathfrak{C}$, we present new proof for the complexity (pseudo-dimension) of a mixture of Gaussians by forging a link to VC analysis of neural networks. The empirical evaluation considers data sets two orders of magnitude larger than those in prior work.

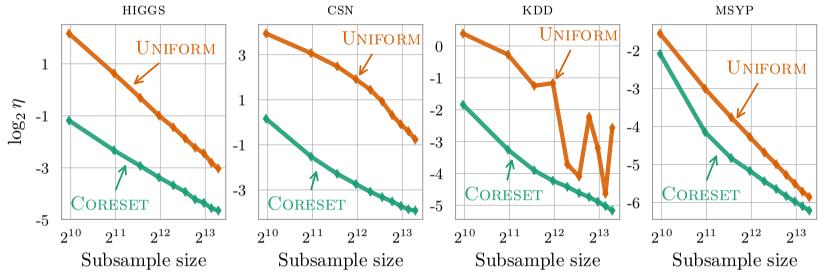


Figure 4: Median relative error η with respect to the best models trained on the full data set (100 runs) with respect to the coreset/subsample (200 runs) For a fixed subsample size, models trained on coresets outperform models trained on the uniform subsample. Furthermore, we observe that the uniform subsampling may be much more sensitive to initialization (KDD).

7.1 Learning Mixtures of Gaussians

The fundamental problem of learning Gaussian mixture models has received a great deal of interest. Dasgupta (1999) was the first to show that, given common covariance, bounded eccentricity, a bound on the smallest component weight, as well as a separation which scales with $\Omega(\sqrt{d})$, parameters of an unknown GMM θ can be estimated in polynomial time, with arbitrary accuracy ε, given i.i.d. samples from θ. Subsequent works relax the assumption on separation to $d^{1/4}$ (Dasgupta and Schulman, 2000) and $k^{1/4}$ (Vempala and Wang, 2004). Feldman et al. (2006b) provide the first result that does not require any separation, but assumes that the Gaussians are axis-aligned. Moitra and Valiant (2010) and Belkin and Sinha (2010) prove that arbitrary GMMs with fixed number of components can be learned in polynomial time and sample complexity (but exponential dependence on k). Anandkumar et al. (2012, 2014) demonstrate that a spectral decomposition technique yields consistent parameter estimates from low-order observable moments, without additional separation assumptions. The aforementioned results hinge on non-degeneracy, that is, that the component means indeed span a k-dimensional subspace and the vector w has strictly positive entries. The problem of fitting a mixture model with near-optimal log-likelihood for arbitrary data is studied by Arora and Kannan (2005). They provide a polynomial-time approximation scheme, provided that the Gaussians are identical spheres. In contrast, as detailed in Section 2, our results make only mild assumptions about the Gaussian components and allow one to explicitly trade-off the coreset size and the assumption strength. Critically, none of the algorithms described above applies to the streaming or parallel setting.

7.2 Approximation Algorithms via Coresets

Existence and construction of coresets have been investigated for a number of problems in computational geometry (Agarwal et al., 2005; Czumaj and Sohler, 2007) and have been used to great effect for a host of geometric and graph problems, including k-median (Har-

Peled and Mazumdar, 2004), k-means (Feldman et al., 2007), k-center (Har-Peled and Varadarajan, 2004), k-line median (Feldman et al., 2006a), subspace approximation (Feldman et al., 2006a; Mahoney and Drineas, 2009), (k, m)-segment mean (Feldman et al., 2012), PCA and projective clustering (Feldman et al., 2013b), distributed k-means and k-median (Balcan et al., 2013), dictionary learning (Feldman et al., 2013a), k-segmentation of streaming data (Rosman et al., 2014), non-parametric estimation (Bachem et al., 2015), and clustering with Bregman divergences (Lucic et al., 2016b). A framework that generalizes and improves several of these results has appeared in Feldman and Langberg (2011). Notably, coresets also imply streaming algorithms for many of these problems (Har-Peled and Mazumdar, 2004; Agarwal et al., 2005; Frahling and Sohler, 2005; Feldman et al., 2007). Recently, coresets were leveraged to establish a space-time-data-risk tradeoff in the context of unsupervised learning (Lucic et al., 2015). Promising results in the context of empirical risk minimization have been demonstrated by Reddi et al. (2015). This paper uses the sensitivity framework of Feldman and Langberg (2011) and the quadratic dependency on the total sensitivity can be reduced to near-linear via Braverman et al. (2016). For a survey of the recent results we refer the reader to Bachem et al. (2017b); Phillips (2016).

8. Conclusion

We have shown how to efficiently construct coresets for estimating parameters of Gaussian mixture models by exploiting a connection between statistical estimation and clustering problems in computational geometry. We prove existence of coresets of size *independent* of the original data set size. To our knowledge, our results provide the first rigorous guarantees for obtaining compressed ε-approximations of the log-likelihood of mixture models for large data sets. The coreset construction algorithm is based on a simple importance sampling scheme and and has linear running time in n. We demonstrate that, by exploiting certain closure properties of coresets, it is possible to construct them in parallel, or in a single pass through a stream of data, using only $\text{poly}(d, k, \lambda^{-1}, \varepsilon^{-1}, \log n, \log(1/\delta))$ space and update time. Critically, our coresets provide guarantees for any given (possibly unstructured) data, without assumptions on the distribution or model that generated it. In an empirical evaluation on several real-world data sets we observe a reduction in computational time of *up to two orders of magnitude*, while achieving a hold-out set likelihood competitive with the models trained on the full data set.

There are two interesting open problems: Is it possible to compute a coreset (i) for any set of k mixture of Gaussians, i.e., whose size is independent of λ, and (ii) of size independent of d, as in Barger and Feldman (2016).

Acknowledgments

We thank Olivier Bachem for invaluable discussions, suggestions and comments. This research was partially supported by ONR grant N00014-09-1-1044, NSF grants CNS-0932392, IIS-0953413, DARPA MSEE grant FA8650-11-1-7156, and the Zurich Information Security Center.

Appendix A. Bounding the Coreset Size

Let $f : \mathcal{X} \times \mathfrak{C} \to \mathbb{R}_+$ be defined as

$$f_\theta(x) = -\ln\left(\sum_{i=1}^{k} \frac{w_i}{Z(\theta)\sqrt{|2\pi\Sigma_i|}} \exp\left(-\frac{1}{2}\left\|{\Sigma_i}^{-1/2}(x-\mu_i)\right\|_2^2\right)\right),$$

where $Z(\theta) = \sum_i w_i/\sqrt{|2\pi\Sigma_i|}$. For a point $x \in \mathcal{X}$ and a *query* θ – a parametrization of the mixture model – $f_\theta(x)$ measures the contribution of the point x to the log-likelihood. Intuitively, to select the points for the coreset, we would like to perform importance sampling on points $x \in \mathcal{X}$, such that we sample points proportionally to the impact on the log-likelihood. Langberg and Schulman (2010) show that it suffices to perform importance sampling with respect to *sensitivity* – the worst-case contribution of a point.

Definition 5 (Sensitivity) *Sensitivity of $x \in \mathcal{X}$ w.r.t. $\mathcal{F} = \{f_\theta(\cdot) \mid \theta \in \mathfrak{C}\}$ is defined as*

$$\sigma_{\mathfrak{C}}(x) = \sup_{\theta \in \mathfrak{C}} \frac{f_\theta(x)}{\frac{1}{|\mathcal{X}|}\sum_{x' \in \mathcal{X}} f_\theta(x')}.$$

Total sensitivity is defined as $\mathfrak{S} = \frac{1}{|\mathcal{X}|}\sum_{x \in \mathcal{X}} \sigma_{\mathfrak{C}}(x)$.

For the Gaussian mixture model, the *query space* $\mathfrak{C}$ is the space of all possible GMMs with k components in d dimensions. While the exact sensitivity is hard to compute, Langberg and Schulman (2010) show that any uniform upper bound $s_{\mathfrak{C}}(x)$ to $\sigma_{\mathfrak{C}}(x)$ can be used. Looser bounds will lead to larger coresets, so one should aim to provide the tightest bound possible.

A.1 Total Sensitivity by Reduction to the Euclidean Space

The following lemma bounds the difference in log-likelihood contribution of two points for any fixed λ-semi-spherical GMM.

Lemma 6 *Let $\mathfrak{C}_\lambda$ be the family of λ-semi-spherical Gaussian mixtures of k components. For a fixed $\theta \in \mathfrak{C}_\lambda$ define $f_\theta : \mathcal{X} \to [0, +\infty)$ as*

$$f_\theta(x) = -\ln\sum_{i=1}^{k} \frac{w_i}{Z(\theta)\sqrt{|2\pi\Sigma_i|}} \exp\left(-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)\right),$$

where $Z(\theta) = \sum_{i=1}^{k} \frac{w_i}{\sqrt{|2\pi\Sigma_i|}}$. Then, for every $x, y \in \mathbb{R}^d$ it holds that

$$f_\theta(x) \leq \frac{1}{\lambda}\|x-y\|_2^2 + 2f_\theta(y).$$

Proof Let $a, x \in \mathbb{R}^d$. By the weak triangle inequality for a fixed $i \in [k]$,

$$\left\|\Sigma_i^{-1/2}(x-\mu_i)\right\|_2^2 \leq 2\left\|\Sigma_i^{-1/2}(x-a)\right\|_2^2 + 2\left\|\Sigma_i^{-1/2}(a-\mu_i)\right\|_2^2. \quad (1)$$

Let UDU^T denote the SVD of Σ_i and note that U is an orthogonal matrix. As such,

$$\left\|\Sigma_i^{-1/2}(x-a)\right\|_2 = \left\|UD^{-1/2}U^T(x-a)\right\|_2 = \left\|D^{-1/2}U^T(x-a)\right\|_2$$
$$\leq \frac{\left\|U^T(x-a)\right\|_2}{\sqrt{\lambda}} \leq \frac{\|x-a\|_2}{\sqrt{\lambda}},$$

which combined with (1) yields

$$\left\|\Sigma_i^{-1/2}(x-\mu_i)\right\|_2^2 \leq \frac{2}{\lambda}\|x-a\|_2^2 + 2\left\|\Sigma_i^{-1/2}(a-\mu_i)\right\|_2^2.$$

As such,

$$f_\theta(x) \leq -\ln\left(\sum_{i=1}^k \frac{w_i}{Z(\theta)\sqrt{|2\pi\Sigma_i|}}\exp\left(-\frac{1}{\lambda}\|x-a\|_2^2 - \left\|\Sigma_i^{-1/2}(a-\mu_i)\right\|_2^2\right)\right)$$
$$= \frac{1}{\lambda}\|x-a\|_2^2 - \ln\left(\sum_{i=1}^k \frac{w_i}{Z(\theta)\sqrt{|2\pi\Sigma_i|}}\exp\left(-\frac{1}{2}\left\|\Sigma_i^{-1/2}(a-\mu_i)\right\|_2^2\right)^2\right)$$
$$\leq \frac{1}{\lambda}\|x-a\|_2^2 - \ln\left(\sum_{i=1}^k \frac{w_i}{Z(\theta)\sqrt{|2\pi\Sigma_i|}}\exp\left(-\frac{1}{2}\left\|\Sigma_i^{-1/2}(a-\mu_i)\right\|_2^2\right)\right)^2$$
$$= \frac{1}{\lambda}\|x-a\|_2^2 + 2f_\theta(a),$$

by Jensen's inequality and the fact that

$$\sum_{i=1}^k \frac{w_i}{Z(\theta)\sqrt{|2\pi\Sigma_i|}} = \frac{1}{Z(\theta)}\sum_{i=1}^k \frac{w_i}{\sqrt{|2\pi\Sigma_i|}} = 1.$$

■

The critical insight necessary to compute an upper-bound on the sensitivity of each point is to note that the denominator in (5) can be lower-bounded by a rough approximation to the optimal k-means clustering of $\mathcal{X}$.

Lemma 7 *Let $\mathfrak{C}_\lambda$ be the family of λ-semi-spherical Gaussian mixtures with k components. Let $\mathcal{X} \subset \mathbb{R}^d$, and $C^\star = \min_{C\subset\mathbb{R}^{d\times k}} \phi(\mathcal{X}, C)$. Let $\mathcal{B} \subset \mathbb{R}^{d\times\beta}$ such that $\phi(\mathcal{X},\mathcal{B}) \leq \alpha\phi(\mathcal{X}, C^\star)$. The sensitivity of $x \in \mathcal{X}$ with respect to $\mathfrak{C}_\lambda$,*

$$\sigma(x) = \sup_{\theta\in\mathfrak{C}_\lambda} \frac{f_\theta(x)}{\frac{1}{|\mathcal{X}|}\sum_{x'\in\mathcal{X}} f_\theta(x')},$$

is bounded by

$$\sigma(x) \leq s(x) = |\mathcal{X}|\frac{2}{\lambda^2}\left(\frac{\alpha \mathrm{d}(x,\mathcal{B})^2}{\sum_{x'\in\mathcal{X}}\mathrm{d}(x',\mathcal{B})^2} + \frac{2\alpha}{|\mathcal{X}_j|}\frac{\sum_{x'\in\mathcal{X}_j}\mathrm{d}(x',\mathcal{B})^2}{\sum_{x'\in\mathcal{X}}\mathrm{d}(x',\mathcal{B})^2} + \frac{2}{|\mathcal{X}_j|}\right).$$

Furthermore, it holds that

$$\mathfrak{S} \leq \frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}} s(x) = \frac{1}{\lambda^2}(6\alpha + 4\beta).$$

Proof Set $\mathcal{B}$ partitions $\mathcal{X}$ into β Voronoi cells, $\mathcal{X}_1, \ldots, \mathcal{X}_\beta$. Consider some $x \in \mathcal{X}_j$ and $\theta \in \mathfrak{C}_\lambda$. By Lemma 6 it holds that

$$\frac{f_\theta(x)}{\sum_{x'\in\mathcal{X}} f_\theta(x')} \leq \underbrace{\frac{1}{\lambda}\frac{\mathrm{d}(x,\mathcal{B})^2}{\sum_{x'\in\mathcal{X}} f_\theta(x')}}_{\mathtt{u}} + \underbrace{\frac{2f_\theta(y)}{\sum_{x'\in\mathcal{X}} f_\theta(x')}}_{\mathtt{v}}. \tag{2}$$

We now upper-bound the right hand side. To bound $\mathtt{u}$, let $\Sigma_i = UDU^T$ be the singular value decomposition of Σ_i and μ_i the corresponding mean. Since U is a rotation matrix

$$\left\|\Sigma_i^{-1/2}(x-\mu_i)\right\|_2 = \left\|UD^{-1/2}U^T(x-\mu_i)\right\|_2 = \left\|D^{-1/2}U^T(x-\mu_i)\right\|_2$$
$$\geq \sqrt{\lambda}\left\|U^T(x-\mu_i)\right\|_2 = \sqrt{\lambda}\|x-\mu_i\|_2 \geq \sqrt{\lambda}\mathrm{d}(x,\mu)$$

where $\mu = \{\mu_1, \ldots, \mu_k\}$. Noting that $Z(\theta)$ is a normalization constant it follows that

$$f_\theta(x) \geq -\ln\left(\sum_{i=1}^k \frac{w_i}{Z(\theta)\sqrt{|2\pi\Sigma_i|}}\exp\left(-\frac{\lambda}{2}\mathrm{d}(x,\mu)^2\right)\right) = \frac{\lambda}{2}\mathrm{d}(x,\mu)^2$$

Summing over $x' \in \mathcal{X}$ yields

$$\sum_{x'\in\mathcal{X}} f_\theta(x') \geq \frac{\lambda}{2}\sum_{x'\in\mathcal{X}}\mathrm{d}(x,\mu)^2 \geq \frac{\lambda}{2}\min_{C\subset\mathbb{R}^{d\times k}}\sum_{x'\in\mathcal{X}}\mathrm{d}(x',C)^2 \geq \frac{\lambda}{2\alpha}\sum_{x'\in\mathcal{X}}\mathrm{d}(x',\mathcal{B})^2,$$

where the last inequality follows by the definition of $\mathcal{B}$. Thus,

$$\mathtt{u} = \frac{1}{\lambda}\frac{\mathrm{d}(x,\mathcal{B})^2}{\sum_{x'\in\mathcal{X}} f_\theta(x')} \leq \frac{2\alpha}{\lambda^2}\frac{\mathrm{d}(x,\mathcal{B})^2}{\sum_{x'\in\mathcal{X}}\mathrm{d}(x',\mathcal{B})^2}. \tag{3}$$

To upper-bound $\mathtt{v}$ consider again the Voronoi partitioning of $\mathcal{X}$ induced by $\mathcal{B}$. Let $x \in \mathcal{X}$ with the corresponding cell $\mathcal{X}_j \subseteq \mathcal{X}$, and $y = \mathcal{B}_j$ such that $\mathrm{d}(x,\mathcal{B}) = \|x-y\|_2$ (y induced the cell $\mathcal{X}_j$). By swapping x and y in Lemma 6 we have

$$f_\theta(y) \leq \frac{1}{\lambda}\mathrm{d}(x,\mathcal{B})^2 + 2f_\theta(x)$$

and summing over $x' \in \mathcal{X}_j$ yields

$$|\mathcal{X}_j| f_\theta(y) \leq \frac{1}{\lambda}\sum_{x'\in\mathcal{X}_j}\mathrm{d}(x',\mathcal{B})^2 + 2\sum_{x'\in\mathcal{X}_j} f_\theta(x')$$
$$\leq \frac{1}{\lambda}\sum_{x'\in\mathcal{X}_j}\mathrm{d}(x',\mathcal{B})^2 + 2\sum_{x'\in\mathcal{X}} f_\theta(x').$$

where the last inequality follows from $f_\theta(x) \geq 0$. Hence,

$$\frac{f_\theta(y)}{\sum_{x'\in\mathcal{X}} f_\theta(x')} \leq \frac{1}{\lambda}\frac{\sum_{x'\in\mathcal{X}_j}\mathrm{d}(x',\mathcal{B})^2}{|\mathcal{X}_j|\sum_{x'\in\mathcal{X}} f_\theta(x')} + \frac{2}{|\mathcal{X}_j|}$$
$$\leq \frac{2\alpha}{\lambda^2}\frac{\sum_{x'\in\mathcal{X}_j}\mathrm{d}(x',\mathcal{B})^2}{|\mathcal{X}_j|\sum_{x'\in\mathcal{X}}\mathrm{d}(x',\mathcal{B})^2} + \frac{2}{|\mathcal{X}_j|}. \tag{4}$$

Since the choice of θ was arbitrary, by applying the obtained bounds (3) and (4) to (2) it follows that

$$\begin{aligned}\sigma(x) &\leq |\mathcal{X}|\frac{2}{\lambda^2}\left(\frac{\alpha \mathrm{d}(x,\mathcal{B})^2}{\sum_{x'\in\mathcal{X}}\mathrm{d}(x',\mathcal{B})^2}+\frac{2\alpha}{|\mathcal{X}_j|}\frac{\sum_{x'\in\mathcal{X}_j}\mathrm{d}(x',\mathcal{B})^2}{\sum_{x'\in\mathcal{X}}\mathrm{d}(x',\mathcal{B})^2}+\frac{2\lambda^2}{|\mathcal{X}_j|}\right)\\ &\leq |\mathcal{X}|\frac{2}{\lambda^2}\left(\frac{\alpha \mathrm{d}(x,\mathcal{B})^2}{\sum_{x'\in\mathcal{X}}\mathrm{d}(x',\mathcal{B})^2}+\frac{2\alpha}{|\mathcal{X}_j|}\frac{\sum_{x'\in\mathcal{X}_j}\mathrm{d}(x',\mathcal{B})^2}{\sum_{x'\in\mathcal{X}}\mathrm{d}(x',\mathcal{B})^2}+\frac{2}{|\mathcal{X}_j|}\right)\\ &:= s(x)\end{aligned}$$

since $\lambda \in (0,1)$. Hence, the total sensitivity may be bounded by

$$\mathfrak{S} \leq \frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}} s(x) = \frac{1}{\lambda^2}(6\alpha+4\beta).$$

■

An attractive property of this result is that the total sensitivity is independent of $|\mathcal{X}|$. Furthermore, to compute the bound we only need a bicriteria approximation to the k-means objective which can be computed in linear time with the popular k-Means++ algorithm (Arthur and Vassilvitskii, 2007) for which $\alpha = \mathcal{O}(\log_2 k)$ and $\beta = k$ resulting in total sensitivity of $\mathcal{O}\big(k/\lambda^2\big)$. As shown in Lucic et al. (2016a), this bound on the total sensitivity is tight (up to a constant) as there exists a data set $\mathcal{X}$ for which $\mathfrak{S} \in \Theta(k)$.

A.2 Pseudo-dimension of Gaussian Mixtures

The other key factor in bounding the coreset size is the combinatorial complexity of the function family $\mathcal{F}$ induced by the maximum likelihood estimation of the Gaussian mixture. More complex models require more samples for uniform convergence. We first introduce the required definitions to ensure that the exposition is self-contained.

Definition 8 (VC dimension) *Let $\mathcal{X}$ be a ground set and $\mathcal{F}$ be a set of functions from $\mathcal{X}$ to $\{0,1\}$. Fix a set $S = \{x_1,\ldots,x_n\} \subset \mathcal{X}$ and a function $f \in \mathcal{F}$. We call $S_f = \{x_i \in S \mid f(x_i) = 1\}$ the induced subset of S by f. A subset $S = \{x_1,\ldots,x_n\}$ of $\mathcal{X}$ is shattered by $\mathcal{F}$ if $|\{S_f \mid f \in \mathcal{F}\}| = 2^n$. VC dimension of $\mathcal{F}$ is the size of the largest subset of $\mathcal{X}$ shattered by $\mathcal{F}$. If $\mathcal{F}$ can shatter sets of arbitrary size VC dimension of $\mathcal{F}$ is ∞.*

These notions naturally extend to functions mapping to $\mathbb{R}$ (or a subset thereof).

Definition 9 (Pseudo-dimension) *Let $\mathcal{X}$ be a ground set and $\mathcal{F}$ be a set of functions from $\mathcal{X}$ to the interval $[0,1]$. Fix a set $S = \{x_1,\ldots,x_n\} \subset \mathcal{X}$, a set of reals $R = \{r_1,\ldots,r_n\}, r_i \in [0,1]$ and a function $f \in \mathcal{F}$. We call $S_f = \{x_i \in S \mid f(x_i) \geq r_i\}$ the induced subset of S formed by f and R. Subset S with associated values R is shattered by $\mathcal{F}$ if $|\{S_f \mid f \in \mathcal{F}\}| = 2^n$. Pseudo-dimension of $\mathcal{F}$ is the cardinality of the largest shattered subset of $\mathcal{X}$. If $\mathcal{F}$ can shatter sets of arbitrary size pseudo-dimension of $\mathcal{F}$ is ∞.*

Clearly, for every space of a given pseudo-dimension we can construct a space with the same VC dimension as formalized by the following lemma.

Lemma 10 *For any $f \in \mathcal{F}$ let B_f be the indicator function of the region below or on the graph of f, i.e. $B_f(x,y) = \mathrm{sgn}(f(x)-y)$. The pseudo-dimension of $\mathcal{F}$ is precisely the VC-dimension of the subgraph class $B_{\mathcal{F}} = \{B_f \mid f \in \mathcal{F}\}$.*

For the Gaussian mixture model, each f_θ consists exclusively of applications of the exponential function and arithmetic operations on real numbers. The problem of bounding the combinatorial complexity of such function classes has received a great deal of interest which culminated in the following result (Anthony and Bartlett, 2009, Theorem 8.14):

Theorem 11 *Let h be a function from $\mathbb{R}^m \times \mathbb{R}^d$ to $\{0,1\}$, determining the class*

$$\mathcal{H} = \{x \mapsto h(\,x) : \theta \in \mathbb{R}^m\}.$$

Suppose that h can be computed by an algorithm that takes as input the pair $(\theta, x) \in \mathbb{R}^m \times \mathbb{R}^d$ and returns $h(\theta, x)$ after no more than t of the following operations:

(i) the exponential function $\alpha \mapsto e^\alpha$ on real numbers,

(ii) the arithmetic operations $+, -, \times,$ and $/$ on real numbers,

(iii) jumps conditioned on $>, \geq, <, \leq, =,$ and $\neq$ comparisons of real numbers, and

(iv) output $0, 1$.

If the t operations include no more than p in which the exponential function is evaluated, then the VC-dimension of $\mathcal{H}$ is $\mathcal{O}\big(m^2p^2 + mp(t + \log mp)\big)$.

Theorem 12 *Let $\mathfrak{C}_\lambda$ be the family of λ-semi-spherical Gaussian mixtures with k components. Let $\mathcal{X} \subset \mathbb{R}^d$ and $m = k(d+1)(d+2)/2 - 1$. Define $f_\theta : \mathcal{X} \to [0,\infty)$ as*

$$f_\theta(x) = -\ln\left(\sum_{i=1}^k \frac{w_i}{Z(\theta)\sqrt{|2\pi\Sigma_i|}}\exp\left(-\frac{1}{2}\left\|{\Sigma_i}^{-1/2}(x-\mu_i)\right\|_2^2\right)\right),$$

where $Z(\theta) = \sum_i \frac{w_i}{\sqrt{|2\pi\Sigma_i|}}$. Let $\mathcal{F} = \{f_\theta(x) \mid \theta \in \mathfrak{C}_\lambda \subset \mathbb{R}^m\}$. Then, $\dim \mathcal{F} \in \mathcal{O}\big(d^4k^4\big)$.

Proof Let $\theta \in \mathfrak{C}_\lambda$ and $r \in \mathbb{R}$. Define $h : \mathbb{R}^{m+1} \times \mathbb{R}^d \to \{0,1\}$ such that $h_{\theta,r}(x) = 1$ iff $f_\theta(x) \geq r$. Let the corresponding function class be defined as

$$\mathcal{H} = \{h_{\theta,r}(\cdot) \mid h_{\theta,r} : \mathcal{X} \to \{0,1\} \mid \theta \in \mathfrak{C}_\lambda \subset \mathbb{R}^m, r \in \mathbb{R}\}.$$

For a fixed query θ, all $\Sigma_i^{-1}, \forall i \in [k]$ are well defined as $\theta \in \mathfrak{C}_\lambda$. Furthermore, $\tilde{w}_i = w_i/(Z(\theta)\sqrt{|2\pi\Sigma_i|}) \in \mathbb{R}, \forall i \in [k]$ and $\sum_{i=1}^k \tilde{w}_i = 1$. As shown in Figure 5 the function $h_{\theta,r}(x)$ can be evaluated using $t = \mathcal{O}(m)$ arithmetic operations out of which exactly $k+1$ are evaluations of the exponential function. Furthermore, it can be evaluated without using the natural logarithm by computing e^{-r}. By Lemma 10 and Theorem 11 we have

$$\dim \mathcal{F} = \dim_{\mathrm{VC}} \mathcal{H} \in \mathcal{O}\big(m^2k^2 + mk(t + \log mk)\big) \in \mathcal{O}\big(k^4d^4\big).$$

■

The lower-bound of $\Omega(kd^2)$ was established by Akama and Irie (2011). It is an open problem whether this gap can be closed further in the general setting.

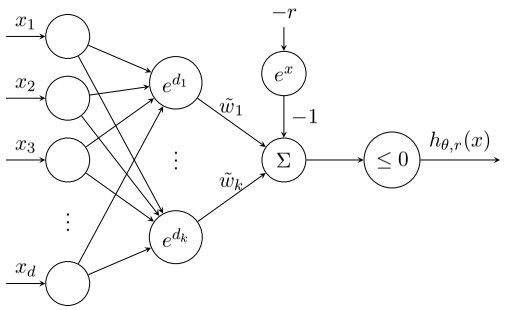


Figure 5: Feed-forward network which calculates $h_{\theta,r}(x)$ whereby the immediate nodes compute the exponents of quadratic forms, and the resulting sum is compared to zero.

A.3 Sufficient Coreset Size

Given the pseudo-dimension $\dim \mathcal{F}$ and some upper bound on the total sensitivity $\mathfrak{S}$ we may bound the coreset size by using the following theorem from Bachem et al. (2017b).

Theorem 13 (Coreset size) *Let $\varepsilon > 0$ and $\delta \in (0,1)$. Let $\mathcal{X}$ be a weighted data set, $\mathcal{Q}$ the set of all possible queries and $f_Q(x) : \mathcal{X} \times \mathcal{Q} \to \mathbb{R}_{\geq 0}$ a cost function. Let $s(x) : \mathcal{X} \to \mathbb{R}_{\geq 0}$ denote any upper bound on the sensitivity $\sigma(x)$ and define $S = \sum_{i=1}^{n} \mu_{\mathcal{X}}(x) s(x)$. Let $\mathfrak{C}$ be a sample of m points from $\mathcal{X}$ with replacement where each point $x \in \mathcal{X}$ is sampled with probability $q(x) = \frac{\mu_{\mathcal{X}}(x)s(x)}{S}$ and each point $x \in \mathfrak{C}$ is assigned the weight $\mu_{\mathfrak{C}}(x) = \frac{\mu_{\mathcal{X}}(x)}{mq(x)}$. Let $\mathcal{F} = \left\{ \frac{\mu_{\mathcal{X}}(\cdot) f_Q(\cdot)}{\mathrm{cost}(\mathcal{X},Q) S q(\cdot)} \mid Q \in \mathcal{Q} \right\}$ and $d' = \dim \mathcal{F}$. Then, the set $\mathfrak{C}$ is an ε-coreset of $\mathcal{X}$ with probability at least $1 - \delta$ for*

$$m \geq \frac{cS^2}{\varepsilon^2} \left(d' + \log \frac{1}{\delta} \right),$$

where $c > 0$ is an absolute constant.

A bound on the coreset size can also be obtained by applying the main theorem of Feldman and Langberg (2011), where one needs to bound the primal shattering dimension instead of the pseudo-dimension. For a detailed discussion of the effects introduced by this difference we refer the reader to Bachem et al. (2017a).

Now we are ready to present the proof of the Theorem 2 which states that, under natural assumptions, we can uniformly approximate the log-likelihood of the model trained on the coreset and the likelihood of the model trained on the full data set as $\varepsilon \to 0$.

Theorem 2 *Let $\mathcal{X} \subset \mathbb{R}^d$, $\delta \in (0,1)$, $\varepsilon \in (0, 1/2)$, $k \in \mathbb{N}$, and $\lambda \in (0,1)$. Let $\mathcal{B}_1, \ldots, \mathcal{B}_p$ be the outputs of $p = \lceil \log_2 1/\delta \rceil$ independent runs of Algorithm 2 with input $\mathcal{X}$ and k. Let $\mathcal{C}$ be the output of Algorithm 1 with $\alpha = 16(\log_2 k + 2)$, $\mathcal{B}^\star = \mathrm{argmin}_{i \in \{1,\ldots,p\}} \phi(\mathcal{X}, \mathcal{B}_i)$ and coreset size*

$$m \geq c \cdot \frac{d^4 k^6 + k^2 \log \frac{1}{\delta}}{\lambda^4 \varepsilon^2},$$

where $c > 0$ is an absolute constant. Then, with probability at least $1 - \delta$, the set $\mathcal{C}$ is a (k, ε)-coreset of $\mathcal{X}$.

Proof Since all p runs are independent, it holds that

$$\mathbb{P}\left[\phi(\mathcal{X}, \mathcal{B}^\star) > t\right] = \mathbb{P}\left[\min_{i \in \{1,\ldots,p\}} \phi(\mathcal{X}, \mathcal{B}_i) > t\right] = \Pi_{i=1}^{p} \mathbb{P}\left[\phi(\mathcal{X}, \mathcal{B}_i) > t\right] \leq \left(\frac{\mathbb{E}[\phi(\mathcal{X}, \mathcal{B}_1)]}{t} \right)^p$$

by Markov's inequality. By Theorem 5 of Arthur and Vassilvitskii (2007), it holds that

$$\mathbb{E}[\phi(\mathcal{X}, \mathcal{B}_1)] \leq 8(\log_2 k + 2)\phi(\mathcal{X}, OPT).$$

Hence, for $p = \lceil \log_2 1/\delta \rceil$, with probability at least $1 - \delta$,

$$\phi(\mathcal{X}, \mathcal{B}^\star) \leq 16(\log_2 k + 2)\phi(\mathcal{X}, OPT).$$

By Lemma 7 we have that

$$s(x) = |\mathcal{X}| \frac{2}{\lambda^2} \left(\frac{\alpha \mathrm{d}(x, \mathcal{B})^2}{\sum_{x' \in \mathcal{X}} \mathrm{d}(x', \mathcal{B})^2} + \frac{2\alpha}{|\mathcal{X}_j|} \frac{\sum_{x' \in \mathcal{X}_j} \mathrm{d}(x', \mathcal{B})^2}{\sum_{x' \in \mathcal{X}} \mathrm{d}(x', \mathcal{B})^2} + \frac{2}{|\mathcal{X}_j|} \right) \geq \sigma(x)$$

for each $x \in \mathcal{X}$ and

$$\mathfrak{S} \leq \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} s(x) = \frac{4\alpha + 2\beta}{\lambda^2} \in \mathcal{O}\left(\frac{k}{\lambda^2} \right) \tag{5}$$

since $\alpha = \mathcal{O}(\log_2 k)$ and $\beta = k$. We conclude the proof by instantiating Theorem 13 with $\mu_{\mathcal{X}}(x) = 1/|\mathcal{X}|$, Theorem 12 and the total sensitivity bound (5). ■

A.4 Directly Approximating the Log-Likelihood

Under additional assumptions on the eigenvalues we can derive a stronger result directly relating the approximated log-likelihood with the true likelihood.

Theorem 14 *Let the conditions of Theorem 2 hold. If $\prod_{\lambda_j \in \mathrm{spec}(\Sigma_i)} \lambda_j \geq \frac{1}{(2\pi)^d}$ we have*

$$|\mathcal{L}(\mathcal{X} \mid \theta) - \mathcal{L}(\mathcal{C} \mid \theta)| \leq \varepsilon \mathcal{L}(\mathcal{X} \mid \theta).$$

Proof By Theorem 2, for $\alpha = 16(\log_2 k + 2)$, $\beta = k$, $m \in \Theta\big(d^4 k^6 \lambda^{-4} \varepsilon^{-2}\big)$, with probability at least $1 - \delta$, it holds that

$$(1 - \varepsilon)\,\mathrm{cost}(\mathcal{X}, \theta) \leq \mathrm{cost}(\mathcal{C}, \theta) \leq (1 + \varepsilon)\,\mathrm{cost}(\mathcal{X}, \theta).$$

By assumption that all eigenvalues are sufficiently large, namely $\prod_{\lambda_j \in \mathrm{spec}(\Sigma_i)} \lambda_j \geq \frac{1}{(2\pi)^d}$, for all components i, the log-normalizer $\ln Z(\theta)$ is negative since

$$Z(\theta) = \sum_i \frac{w_i}{\sqrt{|2\pi\Sigma_i|}} \leq \max_i \frac{1}{\sqrt{|2\pi\Sigma_i|}} = \max_i \frac{1}{\sqrt{(2\pi)^d \prod_{\lambda_j \in \mathrm{spec}(\Sigma_i)} \lambda_j}} \leq 1.$$

Hence,

$$\begin{aligned}\mathcal{L}(\mathcal{C} \mid \theta) &= -n \ln Z(\theta) + \text{cost}(\mathcal{C}, \theta) \leq -n \ln Z(\theta) + (1+\varepsilon)\,\text{cost}(\mathcal{X}, \theta) \\ &\leq (1+\varepsilon)\big(-n \ln Z(\theta) + \text{cost}(\mathcal{X}, \theta)\big) = (1+\varepsilon)\mathcal{L}(\mathcal{X} \mid \theta),\end{aligned}$$

and similarly,

$$\begin{aligned}\mathcal{L}(\mathcal{C} \mid \theta) &= -n \ln Z(\theta) + \text{cost}(\mathcal{C}, \theta) \geq -n \ln Z(\theta) + (1-\varepsilon)\,\text{cost}(\mathcal{X}, \theta) \\ &\geq (1-\varepsilon)\big(-n \ln Z(\theta) + \text{cost}(\mathcal{X}, \theta)\big) = (1-\varepsilon)\mathcal{L}(\mathcal{X} \mid \theta).\end{aligned}$$

■

Appendix B. Convergence of Weighted EM for Gaussian Mixtures

We present the EM update equations for fitting a weighted set of points using Algorithm 4. Since we are interested in an MLE we begin by stating the necessary conditions for a stationary point of

$$\mathcal{L}(\mathcal{C} \mid \theta) = -n \ln Z(\theta) + \text{cost}(\mathcal{C}, \theta) = -\sum_i \gamma_i \ln P(\mathbf{x}'_i \mid \theta).$$

By assumption, all covariance matrices are non-singular. Taking the derivative of $\mathcal{L}(\mathcal{C} \mid \theta)$ with respect to μ_i and Σ_i and setting it equal to zero yields

$$\mu_i = \frac{1}{N_i}\sum_{j=1}^{n} \eta_{i,j} x_j \quad \text{and} \quad \Sigma_i = \frac{1}{N_i}\sum_{j=1}^{n} \eta_{i,j}(x_j - \mu_i)(x_j - \mu_i)^T$$

where

$$N_i = \sum_{j=1}^{n} \eta_{i,j}\gamma_j \quad \text{and} \quad \eta_{i,j} = \gamma_j \frac{w_i \mathcal{N}(\mathbf{x}'_j; \mu_i, \Sigma_i)}{\sum_\ell w_\ell \mathcal{N}(\mathbf{x}'_j; \mu_\ell, \Sigma_\ell)}.$$

To find the mixing weights we minimize the negative log-likelihood under the constraint

$$\sum_{i=1}^{k} w_i = 1, w_i \geq 0, i = 1, \ldots, k.$$

To this end we introduce a Lagrange multiplier λ and minimize $\mathcal{L}(\mathcal{C} \mid \theta) + \lambda(\sum_{i=1}^{k} w_i - 1)$. Setting the derivative with respect to w_i to zero yields

$$w_i = \frac{N_i}{\sum_{j=1}^{n} \eta_{i,j}}$$

As expected, the only difference to the non-weighted version of the EM algorithm is that the updates are now scaled proportionally to the weight of each point. As shown in Dempster et al. (1977) this algorithm will converge to a stationary point.

References

Pankaj K Agarwal, Sariel Har-Peled, and Kasturi R Varadarajan. Geometric approximation via coresets. *Combinatorial and Computational Geometry*, 52:1–30, 2005.

Yohji Akama and Kei Irie. VC dimension of ellipsoids. *arXiv preprint arXiv:1109.4347*, 2011.

Animashree Anandkumar, Daniel J Hsu, and Sham M Kakade. A method of moments for mixture models and hidden markov models. In *Conference On Learning Theory (COLT)*, 2012.

Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research (JMLR)*, 15(1):2773–2832, 2014.

Martin Anthony and Peter L Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 2009.

Sanjeev Arora and Ravi Kannan. Learning mixtures of separated nonspherical Gaussians. *Annals of Applied Probability*, 15(1A):69–92, 2005.

David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Symposium on Discrete Algorithms (SODA)*, pages 1027–1035. SIAM, 2007.

Olivier Bachem, Mario Lucic, and Andreas Krause. Coresets for nonparametric estimation - the case of DP-means. In *International Conference on Machine Learning (ICML)*, 2015.

Olivier Bachem, Mario Lucic, S. Hamed Hassani, and Andreas Krause. Approximate k-means++ in sublinear time. In *Conference on Artificial Intelligence (AAAI)*, 2016a.

Olivier Bachem, Mario Lucic, S. Hamed Hassani, and Andreas Krause. Fast and provably good seedings for k-means. In *Neural Information Processing Systems (NIPS)*, 2016b.

Olivier Bachem, Mario Lucic, and Andreas Krause. Scalable and distributed clustering via lightweight coresets. *arXiv preprint arXiv:1702.08248*, 2017a.

Olivier Bachem, Mario Lucic, and Andreas Krause. Practical coreset constructions for machine learning. *arXiv preprint*, 2017b.

Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii. Scalable k-means++. *Proceedings of the VLDB Endowment*, 5(7):622–633, 2012.

Maria-Florina Balcan, Steven Ehrlich, and Yingyu Liang. Distributed k-means and k-median clustering on general topologies. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1995–2003, 2013.

Pierre Baldi, Peter Sadowski, and Daniel Whiteson. Searching for exotic particles in high-energy physics with Deep learning. *Nature Communications*, 5, 2014.

Artem Barger and Dan Feldman. k-means for streaming and distributed big sparse data. In *International Conference on Data Mining (ICDM)*, pages 342–350. SIAM, 2016.

Mikhail Belkin and Kaushik Sinha. Polynomial learning of distribution families. In *Foundations of Computer Science (FOCS)*, pages 103–112. IEEE, 2010.

Jon Louis Bentley and James B Saxe. Decomposable searching problems I. Static-to-dynamic transformation. *Journal of Algorithms*, 1(4):301–358, 1980.

Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

Vladimir Braverman, Dan Feldman, and Harry Lang. New frameworks for offline and streaming coreset constructions. *arXiv preprint arXiv:1612.00889*, 2016.

Artur Czumaj and Christian Sohler. Sublinear-time approximation algorithms for clustering via random sampling. *Random Structures & Algorithms*, 30(1-2):226–256, 2007.

Sanjoy Dasgupta. Learning mixtures of Gaussians. In *Foundations of Computer Science (FOCS)*, pages 634–644. IEEE, 1999.

Sanjoy Dasgupta and Leonard J Schulman. A two-round variant of EM for Gaussian mixtures. In *Uncertainty in Artificial Intelligence (UAI)*, pages 152–159, 2000.

Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. In *Symposium on Operating System Design and Implementation (OSDI)*, 2004.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

Matthew Faulkner, Michael Olson, Rishi Chandy, Jonathan Krause, K. Mani Chandy, and Andreas Krause. The next big one: Detecting earthquakes and other rare events from community-based sensors. In *ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2011.

Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *Symposium on Theory of Computing (STOC)*, pages 569–578. ACM, 2011.

Dan Feldman and Tamir Tassa. More constraints, smaller coresets: Constrained matrix approximation of sparse big data. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 249–258. ACM, 2015.

Dan Feldman, Amos Fiat, and Micha Sharir. Coresets for weighted facilities and their applications. In *Foundations of Computer Science (FOCS)*, pages 315–324. IEEE, 2006a.

Dan Feldman, Morteza Monemizadeh, and Christian Sohler. A PTAS for k-means clustering based on weak coresets. In *Symposium on Computational Geometry (SoCG)*, pages 11–18. ACM, 2007.

Dan Feldman, Matthew Faulkner, and Andreas Krause. Scalable training of mixture models via coresets. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2142–2150, 2011.

Dan Feldman, Cynthia Sung, and Daniela Rus. The single pixel GPS: learning big data signals from tiny coresets. In *Advances in Geographic Information Systems*, pages 23–32. ACM, 2012.

Dan Feldman, Micha Feigin, and Nir Sochen. Learning big (image) data via coresets for dictionaries. *Journal of Mathematical Imaging and Vision*, 46(3):276–291, 2013a.

Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size coresets for k-means, PCA and projective clustering. In *Symposium on Discrete Algorithms (SODA)*, pages 1434–1453. SIAM, 2013b.

Jon Feldman, Rocco A Servedio, and Ryan O'Donnell. PAC learning axis-aligned mixtures of Gaussians with no separation assumption. In *Learning Theory*, pages 20–34. Springer, 2006b.

Gereon Frahling and Christian Sohler. Coresets in dynamic geometric data streams. In *Symposium on Theory of Computing (STOC)*, pages 209–217. ACM, 2005.

Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In *Symposium on Theory of Computing (STOC)*, pages 291–300. ACM, 2004.

Sariel Har-Peled and Kasturi R Varadarajan. High-dimensional shape fitting in linear time. *Discrete & Computational Geometry*, 32(2):269–288, 2004.

Michael Langberg and Leonard J Schulman. Universal ε-approximators for integrals. In *Symposium on Discrete Algorithms (SODA)*, pages 598–607. SIAM, 2010.

Mario Lucic, Mesrob I. Ohannessian, Amin Karbasi, and Andreas Krause. Tradeoffs for space, time, data and risk in unsupervised learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 663–671, 2015.

Mario Lucic, Olivier Bachem, and Andreas Krause. Linear-time outlier detection via sensitivity. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2016a.

Mario Lucic, Olivier Bachem, and Andreas Krause. Strong coresets for hard and soft Bregman clustering with applications to exponential family mixtures. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1–9, 2016b.

Michael W Mahoney and Petros Drineas. CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences (PNAS)*, 106(3):697–702, 2009.

Konstantin Makarychev, Yury Makarychev, Maxim Sviridenko, and Justin Ward. A Bi-Criteria Approximation Algorithm for k-Means. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2016)*, volume 60, Dagstuhl, Germany, 2016.

Ankur Moitra and Gregory Valiant. Settling the polynomial learnability of mixtures of Gaussians. In *Foundations of Computer Science (FOCS)*, 2010.

Kevin P Murphy. Machine learning: A probabilistic perspective. 2012.

Jeff M Phillips. Coresets and sketches. *arXiv preprint arXiv:1601.00617*, 2016.

Sashank J Reddi, Barnabás Póczos, and Alex Smola. Communication efficient coresets for empirical loss minimization. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2015.

Guy Rosman, Mikhail Volkov, Dan Feldman, John W Fisher III, and Daniela Rus. Coresets for k-segmentation of streaming data. In *Advances in Neural Information Processing Systems (NIPS)*, pages 559–567, 2014.

Santosh Vempala and Grant Wang. A spectral algorithm for learning mixture models. *Journal of Computer and System Sciences*, 68(4):841–860, 2004.

Michael D. Vose. A linear algorithm for generating random numbers with a given distribution. *IEEE Transactions on software engineering*, 17(9):972–975, 1991.

Journal of Machine Learning Research 18 (2018) 1-27 Submitted 11/15; Revised 1/18; Published 4/18

Gradient Estimation with Simultaneous Perturbation and Compressive Sensing

Vivek S. Borkar borkar.vs@gmail.com
Department of Electrical Engineering
Indian Institute of Technology Bombay
Mumbai 400076, India

Vikranth R. Dwaracherla vikrantha@stanford.edu
Department of Electrical Engineering
Stanford, USA

Neeraja Sahasrabudhe neeraja@iisermohali.ac.in
Department of Mathematical Sciences
Indian Institute of Science Education and Research, Mohali
SAS Nagar 140306, India

Editor: Sujay Sanghavi

Abstract

We propose a scheme for finding a "good" estimator for the gradient of a function on a high-dimensional space with few function evaluations, for applications where function evaluations are expensive and the function under consideration is not sensitive in all coordinates locally, making its gradient almost sparse. Exploiting the latter aspect, our method combines ideas from Spall's Simultaneous Perturbation Stochastic Approximation with compressive sensing. We theoretically justify its computational advantages and illustrate them empirically by numerical experiments. In particular, applications to estimating gradient outer product matrix as well as standard optimization problems are illustrated via simulations.

Keywords: Gradient estimation; Compressive sensing; Sparsity; Gradient descent; Gradient outer product matrix.

1. Introduction

Estimating the gradient of a given function (with or without noise) is often an important part of problems in reinforcement learning, optimization and manifold learning. In reinforcement learning, policy-gradient methods are used to obtain an unbiased estimator for the gradient. The policy parameters are then updated with increments proportional to the estimated gradient (Sutton et. al, 2000). The objective is to learn a locally optimum policy. REINFORCE and PGPE methods (policy gradients with parameter-based exploration) are popular instances of this approach (Zhao et. al, 2012) for details and comparisons, (Grondman et. al, 2012) for a survey on policy gradient methods in the context of actor-critic algorithms). In manifold learning, various finite difference methods have been explored for gradient estimation (Mukherjee, Wu and Zhou, 2010; Wu et. al, 2010). The idea is to use the estimated gradient to find the lower dimensional manifold where the given function actually lives. Optimization, i.e., finding maximum or minimum of a function, is a ubiquitous problem that appears in many fields wherein one seeks zeroes of the gradient. But the gradient itself might be hard to compute. Gradient estimation techniques prove particularly useful in such scenarios.

A further theoretical justification is facilitated by the results of Austin (2016). In Austin (2016), it was shown that given a connected and locally connected metric probability space (X, d, μ) (i.e., X is a compact metric space with metric d and μ is a probability measure on the Borel σ-algebra of (X, d)), under suitable conditions, any function $f : X^n \mapsto \mathbb{R}$ is close (in $L^1(\mu^n)$) to a function on a lower dimensional factor space obtained by averaging out the remaining arguments w.r.t. the corresponding product of μ (see Austin, 2016, Theorem 1.1). As a special case, a similar fact can be proved for real-values 1-Lipschitz functions on $\mathbb{R} \setminus \mathbb{Z}$ with metric $|\cdot|_\infty^n$ (see Austin, 2016, Theorem 1.2). This suggests that sparse gradients can be expected for functions on high dimensional spaces with adequate regularity conditions.

Over the years gradient estimation has also become an interesting problem in its own right. One would expect that the efficiency of a given method for gradient estimation also depends on the properties of function f. We consider one such class of problems in this paper. Suppose we have a continuously differentiable function $f : \mathbb{R}^n \mapsto \mathbb{R}$ where n is large, such that the gradient ∇f lives mostly in a lower dimensional subspace. This means that one can throw out most of the coordinates of ∇f in a suitable local basis without incurring too much error. In this case, computing $\frac{\partial f}{\partial x_i}$ $\forall i$ is clearly a waste of means. If in addition the function evaluations are expensive, most gradient estimation methods become inefficient. Such is the case, e.g., if a single function evaluation is the output of a large time-consuming simulation. This situation is our specific focus. The problem of expensive function evaluations does not seem to have attracted much attention in machine learning literature, though there has been quite a lot of work on this theme in other communities such as engineering and operations research (Joseph and Murthy, 2017; Xu, Caramanis and Mannor, 2016). Most methods, however, focus on learning a good surrogate for the original function (see Jones, Schonlau and Welch, 1998; Pandita, Bilionis and Panchal, 2016; Shan and Wang, 2010).

To handle the first issue, ideas from compressive sensing can be applied. Compressive sensing theory tells us that an s-sparse vector can be reconstructed from $m \sim s \log(n/s)$ measurements. This means that one does not need the information about ∇f in all n directions, a much smaller number of measurements would suffice. These ideas are frequently used in signal as well as image processing (see Chan et. al, 2008; Duarte et. al, 2008). To remedy the latter difficulty, we use an idea from Simultaneous Perturbation Stochastic Approximation (SPSA) due to Spall (Spall, 1992), viz., the Simultaneous Perturbation (SP).

We begin by explaining the proposed method for gradient estimation. Important ideas and results from compressive sensing and SPSA that are relevant to this work are discussed in Section 2.1 and Section 2.2 respectively. We state the main result in Section 2.3. Section 3 presents applications to manifold learning and optimization with simulated examples.

Some notational preliminaries are as follows. By $\|\cdot\|_1$ and $\|\cdot\|$ we denote the l_1 and l_2 norms in $\mathbb{R}^n$ respectively. By abuse of notation, we also denote the Frobenius norm for matrices over $\mathbb{R}$ by $\|\cdot\|$. Throughout, 'a.s.' stands for 'almost surely', i.e., with probability one.

2. Gradient Estimation: Combining Compressive Sensing and SP

As mentioned above, if function evaluations are expensive, SP works well to avoid the problem of computing function multiple times. However, if the gradient is sparse it makes sense to use the ideas of compressive sensing to our advantage. Combining these two techniques helps us overcome the problem of too many function evaluations and also exploit the sparse structure of the gradient. The idea is to use SP to get sufficient number of observations to be able to recover the gradient via l_1-minimization. We describe the method in detail in the following sub-sections.

2.1 Compressive Sensing

Assume that $\nabla f \in \mathbb{R}^n$ is an approximately sparse vector. The idea of compressive sensing is based on the fact that typically a sparse vector contains much less information or complexity than its apparent dimension. Therefore one should be able to reconstruct ∇f with considerable accuracy with much less information than that of order n. We will make these ideas more precise in the forthcoming discussion on compressive sensing. We state all the results for vectors in $\mathbb{R}^n$. All of these results also hold for vectors over $\mathbb{C}$. We start by defining what we mean by sparse vectors.

Definition 1 (Sparsity) *The support of a vector $x \in \mathbb{R}^n$ is defined as:*

$$supp(x) := \{j \in [n] : x_j \neq 0\}.$$

where $[n] = \{1, 2, \ldots, n\}$. The vector $x \in \mathbb{R}^n$ is called s-sparse if at most s of its entries are nonzero, i.e., if

$$\|x\|_0 := card(supp(x)) \leq s.$$

We assume that the observed data $y \in \mathbb{R}^m$ is related to the original vector $x \in \mathbb{R}^n$ via $Ax = y$ for some matrix $A \in \mathbb{R}^{m \times n}$, where $m < n$. In other words, we have a linear measurement process for observing x. The theory of compressive sensing tells us that if x is sparse, then it can be recovered from y by solving a convex optimization problem. In particular, given a suitable matrix A and appropriate m, the following l_1-minimization problem recovers x exactly.

$$\min_{z \in \mathbb{R}^n} \|z\|_1 \text{ subject to } y = Az \tag{1}$$

where $y = Ax$ are the m observations. These ideas were introduced by E. Candés and T. Tao in their seminal paper on near-optimal signal reconstruction (Candés and Tao, 2006). In this paper, the authors proved that the matrices suitable for the recovery need to have what is called the restricted isometry property (RIP). A large class of random matrices satisfy the RIP with quantifiable 'high probability' and are therefore suitable for reconstruction via l_1-minimization. In particular, subgaussian matrices have been shown to have RIP with high probability and are suitable for the aforementioned reconstruction scheme for $m \sim s \log(n/s)$. This gives the explicit relationship between the sparsity level s, the dimension of the original vector n and the dimension of the observed data m. In recent times some work has been done to construct deterministic matrices with this restricted isometry property (Bandeira et. al, 2013). The current known lower bound on m for

deterministic matrices is of the order of s^2 where s is the sparsity. Thus random matrices are a better choice for linear measurement for reconstruction via compressive sensing if one is willing to settle for probabilistic guarantees.

For the scope of this paper, we consider robust recovery options using Gaussian random matrices, i.e., matrices whose entries are realizations of independent standard normal random variables.

Remark 2 *Matrices with more structure such as random partial Fourier matrix or more generally, bounded orthonormal systems $\{\phi_i\}_{i=1}^N$ can also be used as meaurement matrices for compressive sensing techniques. Given a random draw of such a matrix with associated constant $K_0 \geq 1$ (where K_0 is the bound on $\|\phi_i\|_\infty$), a fixed s-sparse vector x can be reconstructed via l_1-minimization with high probability provided $m \geq CK_0^2 s \log n$. For more details on random sampling matrices in compressive sensing (see Foucart and Rauhut, 2013, Chap. 12).*

The crucial point here is that it is enough that the given vector is sparse in some basis. A more detailed discussion on various aspects of compressive sensing can be found in Foucart and Rauhut (2013). In real-life situations the measurements are almost always noisy. A more general statement of problem in (1), that takes into account bounded noise in measurement, bounded in norm by η, is given by:

$$\min_{z \in \mathbb{R}^n} \|z\|_1 \text{ subject to } \|Az - y\| \leq \eta \tag{2}$$

It may also happen that the original vector x is not sparse but is close to a sparse vector. In other words, we would like the reconstruction scheme to be robust and stable. Foucart and Rauhut (2013, Theorem 9.13) gives explicit error bounds for stable and robust recovery where A is a subgaussian matrix. The bound is expressed in terms of $\sigma_s(x) := \inf\{\|x - z\| : z \in \mathbb{R}^n \text{ is } s\text{-sparse}\}$, the distance of x from the nearest s-sparse vector, and the measurement error. See Candés, Romberg and Tao (2006); Candés et. al (2005); Kabanava and Rauhut (2015) for more on robust and stable recovery via compressive sensing.

We assume that our observations $y = (y_1, \ldots, y_m)$ are noisy. The following theorem gives an error bound on the reconstruction from noisy measurements using a Gaussian matrix.

Theorem 3 (Theorem 9.20 in (Foucart and Rauhut, 2013)) *Let $x \in \mathbb{R}^n$ be a s-sparse vector. Let $M \in \mathbb{R}^{m \times n}$ be a randomly drawn Gaussian. Assume that noisy measurements $y = Mx + \xi$ are taken with $\|\xi\| \leq \eta$. If for $0 < \epsilon < 1$ and some $\tau > 0$,*

$$\frac{m^2}{m+1} \geq 2s \left(\sqrt{\log(en/s)} + \sqrt{\frac{\log(\epsilon^{-1})}{s}} + \frac{\tau}{\sqrt{s}} \right)^2, \tag{3}$$

then with probability at least $1 - \epsilon$ every minimizer $\hat{x}$ of $\|z\|_1$ subject to $\|Mz - Mx\| \leq \eta$ satisfies

$$\|x - x^*\| \leq \frac{2\eta}{\tau}.$$

See Theorem 9.29 in Foucart and Rauhut (2013) for a statement for stable and robust recovery via Gaussian matrices.

2.2 Simultaneous Perturbation Stochastic Approximation

As discussed above we have a fairly good reconstruction of a sparse gradient ∇f given a sufficient number of observations $\{y_i\}$. However, as mentioned before, the problem often is the unavailability of these observations. Even though observations for ∇f are not readily available, one may compute y_i's using the available information, that is, noisy measurements of the function f. Note that we have, however, assumed that the function evaluations are computationally expensive. We will now address this issue of estimating ∇f with low computational overheads.

Let e_i denote the i^{th} coordinate direction for $1 \leq i \leq n$. We consider the finite difference approximation

$$\frac{\partial f(x(k))}{\partial x_i} \approx \frac{f(x(k)+\delta e_i) - f(x(k)-\delta e_i)}{2\delta}$$

where $x(k) = (x_1(k), \ldots, x_n(k))$ and $\delta > 0$. By Taylor's theorem, the error of estimation is $O(\delta \|\nabla^2 f(x(k))\|)$ where $\nabla^2 f$ denotes the Hessian. This estimate requires $2n$ function evaluations. Replacing the 'two sided differences' $(f(x(k)+\delta e_i) - f(x(k)-\delta e_i))/2$ above by 'one sided differences' $(f(x(k)+\delta e_i) - f(x(k)))$ reduces this to $n+1$, which is still large for large n. Given that we have assumed f to be such that the function evaluations are computationally expensive, an alternative method is desirable. We use the method devised by Spall (Spall, 1992) in the context of stochastic gradient descent, known as Simultaneous Perturbation Stochastic Approximation (SPSA).

Recall the stochastic gradient descent scheme (Borkar, 2008)

$$x(k+1) = x(k) + a(k)\left[-\nabla f(x(k)) + M(k+1)\right], \tag{4}$$

where:

- $\{M(k)\}$ is a square-integrable martingale difference sequence, viz., a sequence of zero mean random variables with finite second moment satisfying

  $$E\left[M(k+1)|x(m), M(m), m \leq k\right] = 0 \;\forall\; k \geq 0,$$

  i.e., it is uncorrelated with the past. We assume that it also satisfies

  $$\sup_k E\left[\|M(k+1)\|^2 | x(m), M(m), m \leq k\right] < \infty, \tag{5}$$

- $\{a(k)\}$ are step-sizes satisfying

  $$a(k) > 0 \;\forall k,\; \sum_k a(k) = \infty,\; \sum_k a(k)^2 < \infty. \tag{6}$$

The term in square bracket in (4) stands for a noisy measurement of the gradient. Under mild technical conditions, $x(k)$ can be shown to converge a.s. to a local minimum of f (Borkar, 2008). The idea is that the incremental adaptation due to the slowly decreasing step-size $a(k)$ averages out the noise $\{M(k)\}$, rendering this a close approximation of

the classical gradient descent with vanishing error (Borkar, 2008). In practice the noisy gradient is often unavailable and one has to use an approximation $\widehat{\nabla f}$ thereof using noisy evaluations of f, e.g., the aforementioned finite difference approximations, which lead to the Kiefer-Wolfowitz scheme. That is where the SP scheme comes in. We describe this next.

Let $\{\Delta_i(k), 1 \leq i \leq n, k \geq 0\}$ be i.i.d. zero mean random variables such that

- $\Delta(k) = (\Delta_1(k), \ldots \Delta_n(k))$ is independent of $M(\ell), \ell \leq k+1$.
- $P(\Delta_i(k) = 1) = P(\Delta_i(k) = -1) = 1/2$.

Then by Taylor's theorem, we have that for $\delta > 0$:

$$\frac{f(x(k)+\delta\Delta(k)) - f(x(k))}{\delta\Delta_i(k)} \approx \frac{\partial f}{\partial x_i}(x(k)) + \sum_{j\neq i} \frac{\partial f}{\partial x_i}(x(k))\frac{\Delta_j}{\Delta_i}. \tag{7}$$

Note that since Δ_j's are i.i.d. zero mean random variables, we have for $j \neq i$,

$$\mathbb{E}\left[\frac{\partial f}{\partial x_i}(x(k))\frac{\Delta_j}{\Delta_i}\Big| x(m), M(m), m \leq k-1\right] = 0.$$

Hence for the purpose of stochastic gradient descent, the second term in (7) acts as a zero mean noise (i.e., martingale difference) term that can be clubbed with $M(k+1)$ as martingale difference noise and gets averaged out by the iteration. This serves our purpose, since the above scheme requires only two function evaluations per iterate given by

$$x_i(k+1) = x_i(k) + a(k)\left[-\frac{f(x(k)+\delta\Delta(k)) - f(x(k))}{\delta\Delta_i(k)}\right] + M_i(k+1).$$

Our idea is to generate $\widetilde{\nabla f}$ according to the scheme discussed above.

It should be mentioned that Spall also introduced another approximation based on a single function evaluation (see Borkar, 2008, chap. 10). But this suffers from numerical issues due to the 'small divisor' problem, so we do not pursue it here.

2.3 Main result

As mentioned in the introduction, the idea is to combine the SP and compressive sensing to obtain a sparse approximation of ∇f. Note that while SP gives an estimate with zero-mean error, the final estimate of gradient obtained after compressive sensing may not be unbiased. To avoid the error from piling up we need to average out the error at SP stage. We propose the following algorithm for estimating gradient of f. Let a_i denote the row vectors of A.

Algorithm 1 Gradient Estimation at some $x \in \mathbb{R}^n$ with SP and Compressive Sensing

Initialization:

$A = (a_{ij})_{m\times n} \leftarrow$ random Gaussian matrix.

$\delta \leftarrow$ small positive scalar.

- $y_i^\ell = \frac{f(x+\delta\sum_j^m \Delta_j^\ell a_j) - f(x)}{\delta\Delta_i^\ell}$ for $i = 1,\dots,m;\ \ell = 1,\dots,k,$
  where Δ_j^ℓ are i.i.d. zero mean Bernoulli random variables taking values in $\{-1, 1\}$.
- Set $\bar{y}_i := \frac{\sum_{\ell=1}^k y_i^\ell}{k},\ i = 1,\dots,m.$
- $y = (\bar{y}_1,\dots,\bar{y}_m) = A\nabla f(x) + \zeta$ where ζ denotes the bounded error with bound $\|\zeta\| \le \eta$.
- Solve the l_1-minimization problem stated below to obtain $\widetilde{\nabla f}$:

$$\text{minimize } \|z\|_1 \ \text{ subject to } \ \|Az - y\| \le \eta$$

Output: estimated gradient $\widetilde{\nabla f}(x)$.

There are several algorithms for the l_1-minimization problem that appears in the last step of the algorithm above. A detailed discussion of these algorithms, including the Homotopy method (used in Section 3 for the l_1-minimization step), can be found in Yang et. al (2010) and Foucart and Rauhut (2013, Chap. 15).
The following theorem states that with high probability such an approximation is "close" to the actual gradient.

Theorem 4 *Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be a continuously differentiable function with bounded sparse gradient. Then for $m \in \mathbb{N}$ such that it satisfies the bound in (3), $0 < \epsilon << \frac{1}{m}$, and given $\delta > 0$ (as in (7)) and $\tau > 0$ (as in Theorem 3), ∇f can be estimated by a sparse vector $\widetilde{\nabla f}$ such that with probability at least $1 - \epsilon m$,*

$$\|\widetilde{\nabla f} - \nabla f\| < \frac{2t}{\tau}$$

where $t > 2mO(\delta)$.

Proof Let $A \in \mathbb{R}^{m\times n}$ be a Gaussian matrix such that m satisfies (3). Then, following the same idea as in (7), we have:

$$\begin{aligned} y_i &= \frac{f(x+\delta\sum_{j=1}^m \Delta_j a_j) - f(x)}{\delta\Delta_i} \\ &= \langle\nabla f(x), a_i\rangle + \sum_{j\neq i}\frac{\Delta_j\langle\nabla f, a_i\rangle}{\Delta_i} + O(\delta). \end{aligned}$$

So we get

$$y = A\nabla f + \text{ 'error'}, \tag{8}$$

where we quantify the 'error' below.

The above computation is carried out k times independently, keeping the matrix A fixed and choosing the random vector Δ according to the distribution defined in Section 2.2. The reason for this additional averaging is as follows. The reconstruction in compressive sensing need not give an unbiased estimate, since it performs a nonlinear (minimization) operation. Thus it is better to do some pre-processing of the SP estimate (which is nearly, i.e., modulo the $O(\delta)$ term, unbiased) to reduce its variance. We do so by repeating it k times with independent perturbations and taking its arithmetic mean. This may seem to defeat our original objective of reducing function evaluations, but the k required to get reasonable error bounds is not large as our analysis shows later, and the computational saving is still significant (see 'Remark 5' below).

Denote by y^l the measurement obtained at l^{th} iteration of SP. The error for a single iteration is given by

$$\eta^l = \left(\sum_{j\neq 1}\frac{\Delta_j^l\langle\nabla f, a_1\rangle}{\Delta_1^l} + O(\delta),\dots,\sum_{j\neq m}\frac{\Delta_j^l\langle\nabla f, a_m\rangle}{\Delta_m^l} + O(\delta)\right).$$

Denote by $X_{ij}^l = \frac{\Delta_j^l\langle\nabla f, a_i\rangle}{\Delta_i^l},\ j \neq i.$ X_{ij}^l are zero-mean conditionally (given past iterates) independent random variables.

The error vector after k iterations is given by

$$\begin{aligned} \eta &= \frac{1}{k}\sum_{l=1}^k \eta^l \\ &= \frac{1}{k}\sum_{l=1}^k\left(\sum_{j\neq 1}X_{1j}^l + O(\delta),\dots,\sum_{j\neq m}X_{mj}^l + O(\delta)\right) \\ &= \left(\frac{1}{k}\sum_{l=1}^k\sum_{j\neq 1}X_{1j}^l + O(\delta),\dots,\frac{1}{k}\sum_{l=1}^k\sum_{j\neq m}X_{mj}^l + O(\delta)\right). \end{aligned} \tag{9}$$

In order to apply the ideas from compressive sensing as in Theorem 3, we need to have a bound on the error $\|\eta\|$. This is obtained as follows. Let $K > 0$ be a constant such that the $O(\delta)$ term above is bounded in absolute value by $K\delta$. K can, e.g., be a bound on $\|\nabla^2 f\|\|A\|$ by the mean value theorem, where we use the Frobenius norm. Choose $C \ge \sup|\langle\nabla f, a_i\rangle|$

and $t > 2mK\delta$. Then, by Hoeffding's inequality we have,

$$\begin{aligned} P(\|\eta\| \geq t) &\leq \sum_{i=1}^{m} P\left(\left|\frac{1}{k}\sum_{l=1}^{k}\sum_{j\neq i} X_{ij}^{l} + O(\delta)\right| > t/m\right) \\ &\leq \sum_{i=1}^{m} P\left(\left|\sum_{l=1}^{k}\sum_{j\neq i} X_{ij}^{l}\right| > kt/2m\right) \\ &\leq 2me^{-\frac{kt^2}{2m^2(m-1)C^2}}. \end{aligned}$$

Choose the number of iterations, $k > \frac{2m^3C^2}{t^2}\log\left(\frac{2}{\epsilon}\right)$. Then,

$$P(\|\eta\| \geq t) \leq \epsilon m. \tag{10}$$

We define $\widetilde{\nabla f}$ to be the reconstruction of the gradient using m measurements. That is, $\widetilde{\nabla f}$ solves the following optimization problem:

$$\min_{z\in\mathbb{R}^n} \|z\|_1 \text{ subject to } \quad \|Az - y\| \leq t,$$

where y is as in (8). Our claim then follows from the bound in (10) and Theorem 3. ■

Remark 5 *Note that the minimum number of iterations of SP required to obtain a "good" estimate of ∇f is given by*

$$\begin{aligned} k &> \frac{2m^3C^2}{t^2}\log\left(\frac{2}{\epsilon}\right) \\ &\geq \frac{mC^2}{2K^2\delta^2}\log\left(\frac{2}{\epsilon}\right) \\ &\geq \frac{s\tilde{C}}{\delta^2}\log\left(\frac{n}{s}\right)\log\left(\frac{2}{\epsilon}\right). \end{aligned}$$

for a suitable constant $\tilde{C}$.

The above $\widetilde{\nabla f}$ can now be used as an effective gradient in various problems involving gradients of high-dimensional functions. Three such applications are discussed in the next section.

3. Applications

We consider the application of our method to manifold learning and optimization problems. The gradient estimates obtained using our method can be used to estimate the gradient outer product matrix or can be plugged into an optimization scheme. In the former case, along with an example, we also provide error bounds on the estimated and actual gradient outer product matrix. For the latter case, we look at an example and provide suitable

modifications to existing algorithms to achieve faster convergence. Algorithm 1 described in Section 2.3 is used for gradient estimation.

As mentioned before, there are various algorithms available for carrying out the l_1-minimization. Here we use the homotopy method (see Donoho and Tsaig, 2008; Foucart and Rauhut, 2013; Yang et. al, 2010). Homotopy method solves the quadratically constrained l_1-minimization problem (2) by considering the following l_1-regularized least squares functional:

$$F_\lambda(x) = \frac{1}{2}\|Az - y\| + \lambda\|z\|_1 \quad \text{for } \lambda > 0 \tag{11}$$

The algorithm traces the piece-wise linear and continuous solution path $\lambda \mapsto x_\lambda$ and at each step, an element is added or removed from the support set of the current minimizer. If the minimizer x^* of (1) is unique then the minimizer x_λ of (11) converges to x^*. The idea is to start with a large λ such that the minimizer x_λ of (11) is zero and then trace the solution trajectory in the direction of decreasing λ.

We consider a Gaussian matrix A and get $y(n) \quad \leftarrow A\nabla f(x(n)) +$ error as obtained in equation (8). The last step is to obtain $\widetilde{\nabla f}(x)$ via l_1-recovery from observations y and Gaussian random matrix A using the homotopy method described above. All the simulations were performed on MATLAB using the available toolbox for l_1-minimization. (Berkeley database: http://www.eecs.berkeley.edu/ yang/software/ l1benchmark/).

Consider a function $f : \mathbb{R}^{25000} \mapsto \mathbb{R}$ given by $f(x) = x^T MM^T x$ where, M is 25000×3-dimensional matrix with 3 non-zero elements per column. Let A be a random Gaussian matrix that is used for measurement. We consider $m = 50$ measurements.

Figure 1 shows the performance of the proposed method with varying number of SP iterations. Figures 2 and 3 show the comparison between our method and naive SP for estimating gradient with gradually increasing number of iterations for averaging over SP (The quantity 'k' in (9)). As mentioned earlier, since the gradient is assumed to be sparse, using naive SP to compute derivative in each direction seems wasteful. Although the error diminishes as the number of iterations for SP increase, the proposed method combining compressive sensing with SP consistently performs better.

Figures 4 and 5 show that the proposed method works well with higher sparsity levels too. It also shows that with higher s, performance of naive SP improves. This is expected. The extremely high error in the naive SP method (especially for small s) is owing to the fact that the actual gradient is extremely sparse and in the beginning SP method ends up populating almost all the coordinates. That contributes to the high percentage of error as seen in the aforementioned figures.

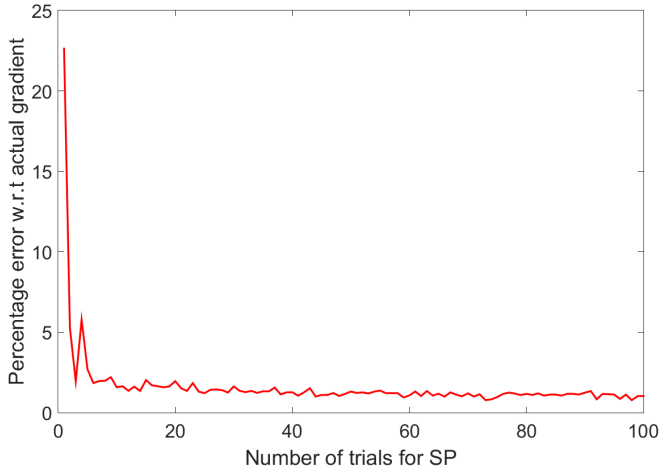


Figure 1: Percentage error of $\|\nabla f - \widetilde{\nabla f}\|$ in our method, with varying number of iterations k for SP. Here, $n = 25000, s = 3$ and $m = 50$.

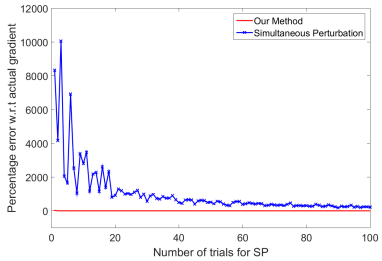


Figure 2: Performance of the proposed algorithm vs. the SP method with varying number iterations k at SP step. Here $n = 25000, s = 3$ and $m = 50$

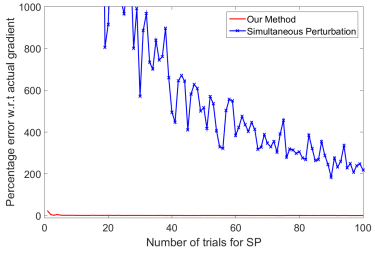


Figure 3: A closer look at Figure 2 : Performance of the proposed algorithm vs. the SP method with varying number iterations k at SP step. Here $n = 25000, s = 3$ and $m = 50$.

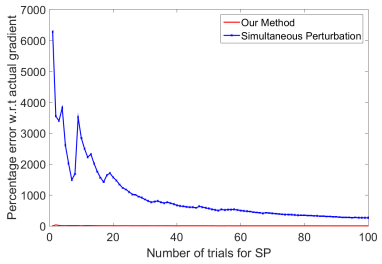


Figure 4: Performance of the proposed algorithm vs. the SP method with varying number iterations k at SP step. Here $s = 50$ and $m = 500$.

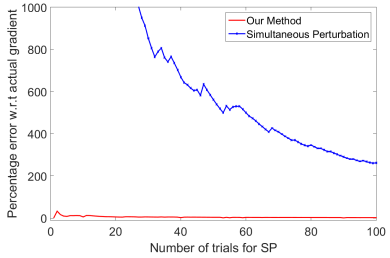


Figure 5: A closer look at Figure 4: Performance of the proposed algorithm vs. the SP method with varying number iterations k at SP step. Here $s = 50$ and $m = 500$.

Before we consider specific applications, we illustrate how the percentage error of estimated gradient with varying k for different sparsity levels s. For appropriately large m, for small k the error is high (this matches with the discussion in Remark 5). As k increases the error is much less. As long as m satisfies (3), the compressive sensing results apply. Figures 6 and 7 show the behaviour of the proposed method with variation in the sparsity, but with constant number of observations. We consider 10000-dimensional vector with $m = 50$ observations. As expected, for a fixed m, as the sparsity increases, increasing k no longer helps as the compressive sensing results do not apply and the error increases. $f(x) = x(1)^2 + \ldots + x(s)^2$ was used as a test function for both the simulations.

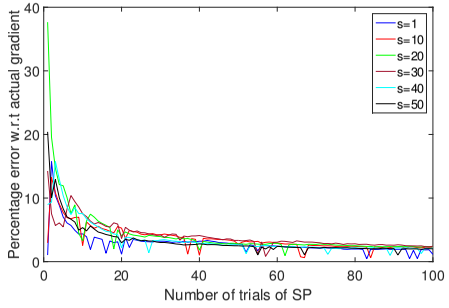


Figure 6: Performance of the proposed algorithm with variation k for different sparsity levels.

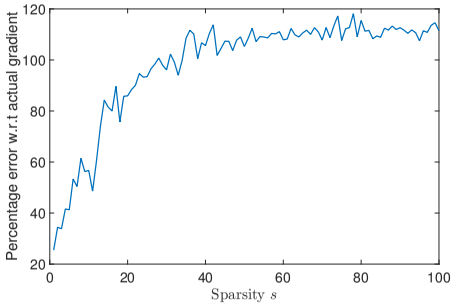


Figure 7: Performance of the proposed algorithm with variation in sparsity.

3.1 Manifold Learning: Estimating e.d.r. space

Consider the following semi-parametric model

$$Y = f(X) + \epsilon$$

where ϵ is noise and f is a smooth function $\mathbb{R}^n \mapsto \mathbb{R}^m$ of the form $f(X) = g(b_1^T X, \ldots, b_d^T X)$. Define by B the matrix $(b_1, b_2, \ldots, b_d)^T$. B maps the data to a d-dimensional relevant subspace. This means that the function f depends on a subspace of smaller dimension given by Range(B) (Note that this is essentially the local view in manifold learning : B can vary with location.). The vectors or the directions given by the vectors b_i are called the effective dimension reducing directions or e.d.r. The question is: how to find the matrix B? It turns out that if f doesn't vary in some direction v, then $v \in Null(E_X[G])$ where G is the gradient outer product matrix defined as

$$G = [[G_{ij}]] \text{ where } G_{ij} = \Big\langle \frac{\partial f}{\partial x_i}(X), \frac{\partial f}{\partial x_j}(X) \Big\rangle$$

and $E_X[\,\cdot\,]$ denotes the expectation over X. Lemma 1 from (Wu et. al, 2010) stated below implies that to find the e.d.r. directions it is enough to compute $E_X[G]$.

Lemma 6 *Consider the semi-parametric model*

$$Y = g(b_1^T X, \ldots, b_d^T X) + \epsilon, \tag{12}$$

where ϵ represents zero mean finite variance noise. Then the expected gradient outer product (EGOP) matrix G is of rank at most d. Furthermore, if $\{v_1, \ldots, v_d\}$ are the eigenvectors associated to the nonzero eigenvalues of G, the following holds:

$$Span(B) = Span(v_1, \ldots, v_d).$$

Clearly, calculating $E_X[G(X)]$ is computationally heavy. We therefore try to estimate this matrix. Several methods are known for estimating the EGOP and this has been a very popular problem in statistics for a while. The idea of using EGOP for obtaining e.d.r. originated in Li (1991). While there are other methods based on inverse regression etc., most of the efforts have been directed towards getting an efficient way to estimate gradients in order to finally estimate EGOP (see Xia et. al, 2002). In Mukherjee, Wu and Zhou (2010), the authors use their method of gradient estimation for this purpose. The idea is to use sample observations $\{f(x_i)\}$ for $\{x_i\}$ in a neighborhood of the given point x and minimize over z the error

$$\frac{1}{n^2} \sum_{i,j=1}^{n} w_{ij}[y_i - f(x_j) - \langle z, (x_i - x_j)\rangle]^2,$$

where $w_{ij} \geq 0$ are weights ('kernel') that favor locality $x_i \approx x$ and are typically Gaussian, with regularization in a reproducing kernel Hilbert space (RKHS). The minimizer then is the desired estimate. In Trivedi et. al (2014) a rather simple rough estimator using directional derivative along each coordinate direction is provided. The authors demonstrate that for the purpose of finding e.d.r., a rough estimate such as theirs suffices. We also propose a method via gradient estimation. Take $\widehat{G}$ to be the matrix defined by

$$\widehat{G}_{ij} = \Big\langle \widetilde{\frac{\partial f}{\partial x_i}}, \widetilde{\frac{\partial f}{\partial x_j}} \Big\rangle.$$

In other words, $\widehat{G} = \widetilde{\nabla f} \widetilde{\nabla f}^T$, where $\widetilde{\nabla f}$ denotes the estimate of ∇f obtained by algorithm 1. We impose our previous restrictions on f. That is, the function evaluations at any point are expensive and the gradient of f is sparse. In this case we propose an estimate for $E_X[G]$ by the mean of $\widehat{G}$ over a sample of r points given by the set $\chi = \{(x_i, f(x_i))\}_{1 \leq i \leq r}$. By $\langle\, \cdot\, \rangle$, we shall denote the empirical mean over the sample set χ. Thus,

$$\langle G(X) \rangle = \frac{1}{r} \sum_{x_i \in \chi} \nabla f(x_i) \nabla f(x_i)^T$$

and

$$\langle \widehat{G}(X) \rangle = \frac{1}{r} \sum_{x_i \in \chi} \widetilde{\nabla f}(x_i) \widetilde{\nabla f}(x_i)^T.$$

Theorem 7 *Let $f : \mathbb{R}^n \mapsto \mathbb{R}^k$ from the semi-parametric model in (12) be a continuously differentiable function with bounded sparse gradient. Then, for $0 < \epsilon << \frac{1}{m}$ and some $\tau > 0$, with probability at least $1 - \epsilon m$,*

$$\Big\| E_X[G] - \langle \widehat{G} \rangle \Big\| < \frac{6R^2}{\sqrt{r}} \left(\sqrt{\ln n} + \sqrt{\ln \frac{1}{\epsilon}} \right) + \frac{2t}{\tau} \left(\frac{2t}{\tau} + 2R \right)$$

where r is the sample size, R is such that $\|\nabla f\| \leq R$, t is as in Theorem 4 and $m \in \mathbb{N}$ is such that it satisfies the bound in (3).

The proof closely follows the line of argument in Trivedi et. al (2014).

Proof Note that,

$$\|E_X[G(X)] - \langle \widehat{G}(X)\rangle\| \leq \|E_X[G(X)] - \langle G(X)\rangle\| + \|\langle G(X)\rangle - \langle \widehat{G}(X)\rangle\|.$$

The idea is to bound each term. We use concentration inequality for sum of random matrices (see Trivedi et. al, 2014, Lemma 1) and (Tropp, 2012) for more general results, to claim that for $\epsilon > 0$,

$$\|E_X[G(X)] - \langle G(X)\rangle\| \leq \frac{6R^2}{\sqrt{r}}\left(\sqrt{\ln n} + \sqrt{\ln \frac{1}{\epsilon}}\right)$$

with probability $\geq 1 - \epsilon$. For the second term, it is enough to show that it is bounded for any single sample point x. Observe that for any two vectors v and w, $\|vv^T - ww^T\| \leq \|(v-w)(v+w)^T\|$ Using this we get, for a fixed x,

$$\begin{aligned}
\|G(x) - \widehat{G}(x)\| &= \|\nabla f(x)\nabla f(x)^T - \widetilde{\nabla} f(x)\widetilde{\nabla} f(x)^T\| \\
&\leq \|\nabla f(x) - \widetilde{\nabla} f(x)\|\|\nabla f(x) + \widetilde{\nabla} f(x)\| \\
&\leq \|\nabla f(x) - \widetilde{\nabla} f(x)\|\left(\|\nabla f(x) - \widetilde{\nabla} f(x)\| + 2\|\nabla f(x)\|\right) \\
&\leq \frac{2t}{\tau}\left(\frac{2t}{\tau} + 2R\right)
\end{aligned}$$

with probability $\geq 1 - \epsilon m$, where the last inequality is obtained by applying the bound from Theorem 4. ■

We now simulate an example to illustrate the decay of the error $\|\langle G(X)\rangle - \langle \widehat{G}(X)\rangle\|_F$ (See Figure 8). Consider a function $f : R^{25000} \mapsto R^3$ given by: $f_i(x) = x^T M_i M_i^T x$ where, M_i is a 25000-dimensional vector with 3 non-zero elements and $f_i(x)$ corresponds to the i^{th} dimension of $f(x)$. A 25000×100 Gaussian random matrix is used for the compressive sensing part of the algorithm. The plot of percentage in normed error between $\langle G(X)\rangle$ and $\langle \widehat{G}(X)\rangle$, i.e. $\|\langle G(X)\rangle - \langle \widehat{G}(X)\rangle\|_F^2$ is shown below by varying number of samples $r = 1$ to 25. Remember that due to the bias at compressive sensing step, we need to average out the gradient estimation error at SP step. This is done in $k = 100$ iterations.

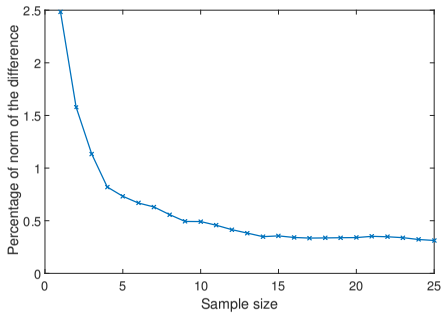


Figure 8: Percentage error in $\|\langle G(X)\rangle - \langle \widehat{G}(X)\rangle\|$ with number of samples r varying from 1 to 25. Here, $n = 25000, s = 3, m = 100$ and $k = 100$.

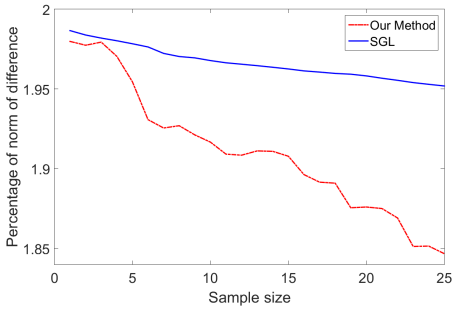


Figure 9: Comparison of percentage error of $\|\langle G(X)\rangle - \langle \widehat{G}(X)\rangle\|$ computed by plugging gradient estimates by proposed method vs SGL method. Here, $n = 10000, s = 30, m = 100$ and $k = 100$. Number of samples for SGL method are 100.

Learning e.d.r. by estimating the gradient using the method proposed in this paper was compared with the SGL (Sparse Gradient Learning) method proposed in (Mukherjee, Wu and Zhou, 2010) using the same function as above and an exponential kernel (See `http://www2.stat.duke.edu/~sayan/soft.html` for details). This is illustrated in Figure 9. Here, $n = 10000$ and the measurement matrix is a 10000×100 Gaussian matrix. The SP step is averaged over 100 iterations. 100 samples were considered for the SGL method with the neighborhood radius of of 0.05. Sparsity of the gradient vector is 30.

3.2 Optimization

We consider next a typical problem of function minimization, but only consider a function with sparse gradient. In other words, we want to minimize $f(x)$ where

$$f : \mathbb{R}^n \mapsto \mathbb{R}$$

is a continuously differentiable real-valued Lipschitz function such that function evaluation at a point in $\mathbb{R}^n$ is typically expensive. We also assume that n is large and that ∇f is sparse. In addition, we assume that the critical points of f (i.e., the zeros of ∇f) are isolated. (This is generically true unless there is overparametrization.) The idea is to use the stochastic gradient scheme (4) with the standard assumptions (5), (6). It follows from the theory of stochastic approximation (see Borkar, 2008, chap. 2) that under above conditions, the solution of the random difference equation (4) tracks with probability one the trajectory of the solution of a limiting o.d.e. as long as the iterates remain bounded, which they do under mild additional conditions on f. Following (Borkar, 2008, chap. 2), we use this so called 'o.d.e. approach' which states that the algorithm will a.s. converge to the equilibria of the limiting o.d.e., which is

$$\dot{x}(t) = -\nabla f(x(t)). \tag{13}$$

For this, f itself serves as the Lyapunov function, leading to the conclusion that the trajectories of (13) and therefore a.s., the iterates of (4) will converge to one of its equilibria, viz., the critical points of f. In fact under additional conditions on the noise, it will converge to a (possibly random) stable equilibrium thereof, viz., a local minimum (*ibid.*, Chapter 4).

The stochastic gradient scheme requires $\nabla f(x)$ at each iteration. The problem often is the unavailability of $\nabla f(x)$, as already noted. It is therefore important to have a good method for estimating the gradient. Typically one would obtain noisy measurements and hence the estimate will have a non-zero error η. It is known that if the error remains small, the iterates converge a.s. to a small neighbourhood of some point in the set of equilibria of (13). We analyze the resultant error below. Also, the error obtained in SP is zero-mean modulo higher order terms, so one can even take an empirical average over a few separate estimates in order to reduce variance. For high dimensional problems, the number of function evaluations remains still small as compared with, e.g., the classical Kiefer-Wolfowitz scheme. We use Theorem 4 to justify using SP (Simultaneous Perturbation Stochastic Approximation) combined with compressed sensing to obtain an approximation for the gradient and then use the above scheme to minimize f.

Consider the following stochastic approximation scheme:

$$x_{n+1} = x_n + a(n)[-\nabla f(x_n) + M_{n+1} + \eta(n)] \tag{14}$$

where $\{\eta(n)\}$ is the additional error arising due to the error in gradient estimation. That is, $\widetilde{\nabla} f(x_n) = \nabla f(x_n) - \eta(n)$. If $\sup_n \|\eta(n)\| < \epsilon_0$ for some small ϵ_0, then the iterates of (14) converge to a small neighbourhood A of some point x^* in $H = \{x : \nabla f(x) = 0\}$ (see Tadic and Doucet, 2017) and (Borkar, 2008, chap. 10)). This is ensured by a Lyapunov argument as follows. The limiting o.d.e. is of the form

$$\dot{x}(t) = -\nabla f(x(t)) + \breve{\eta}(t)$$

for some measurable $\breve{\eta}(\cdot)$ with $\|\breve{\eta}(t)\| \leq \epsilon_0 \ \forall t$. Then

$$\frac{d}{dt} f(x(t)) = -\|\nabla f(x(t))\|^2 + \langle \nabla f(x(t)), \breve{\eta}(t) \rangle,$$

which is < 0 as long as $\|\nabla f(x(t))\|^2 > |\langle \nabla f(x(t)), \breve{\eta}(t) \rangle|$. Therefore $x(t)$ will converge to the set

$$\{x : \|\nabla f(x)\| \leq \epsilon_0\}.$$

Assume that the Hessian $\nabla^2 f(x^*)$ is positive definite, which is generically so for isolated local minima. Then for A small enough, the lowest eigenvalue $\lambda_m(x)$ of $\nabla^2 f(x)$ for $x \in A$ is > 0. By mean value theorem, $\nabla f(x) = \nabla^2 f(x')(x - x^*)$ for some $x' \in A$, so $\|\nabla f(x)\| \geq \lambda_m(x')\|x - x^*\|$. Thus there is convergence to a ball of radius $\frac{\epsilon_0}{\lambda_m}$ around x^*. (A statement to this result without the estimate on the radius of the ball is contained in Theorem 1 of (Hirsch, 1989).) Thus we have:

Theorem 8 *The stochastic gradient scheme*

$$x_{n+1} = x_n + a(n)[-\widetilde{\nabla} f(x_n) + M_{n+1}]$$

a.s. converges to a ball of radius $O(\epsilon_0)$ centered at some local minimum of f, where $\widetilde{\nabla} f$ is the reconstructed gradient as in Theorem 4 and ϵ_0 is a bound on $\|\widetilde{\nabla} f - \nabla f\|$.

Proof The claim is immediate from the above observations about the perturbed differential equation and Theorem 6, pp. 58-59, (Borkar, 2008). □

See Tadic and Doucet (2017) for a finer analysis. Also, observe that we have only discussed asymptotic convergence above. For real-life optimization problems, however, we must ensure that the scheme in (14) converges to a neighbourhood of x^* in finite time. This is indeed true and recent concentration-type results (see Kamal, 2010; Thoppe and Borkar , 2015)) strengthen the theoretical basis for plugging $\widetilde{\nabla} f(x)$ in place of $\nabla f(x)$ in stochastic gradient descent schemes. The results in Kamal (2010) involve estimates on lock-in probability, i.e., the probability of convergence to a stable equilibrium given that the iterates visit its domain of attraction. An estimate on the number of steps needed to be within a prescribed neighborhood of the desired limit set with a prescribed probability is also obtained. Specifically, the result states that if the n_0th iterate is in the domain of attraction of a stable equilibrium x^*, then after a certain number of additional steps, the iterates remain in a small tube around the differential equation trajectory converging to x^* with probability exceeding

$$1 - \mathrm{O}\left(e^{-\frac{C}{(\sum_{m=n_0}^{\infty} a(m)^2)^{\frac{1}{4}}}} \right),$$

ipso facto implying an analogous claim for the probability of remaining in a small neighborhood of x^* after a certain number of iterates. We refer the reader to Kamal (2010) for details. In Thoppe and Borkar (2015), an improvement on this estimate is proved under additional regularity conditions on ∇f (twice continuous differentiability) using Alekseev's

formula. We have omitted the details of both the cases as it needs much additional notation to replicate them here. These would, however, apply to the exact stochastic gradient descent. Since we have an additional error due to approximate gradient as in the preceding theorem, we need to combine the results of *ibid.* with the above theorem to make a weaker claim regarding how small the neighborhood of x^* in question can be. Furthermore, these claims are about iterates which are in the domain of attraction of a stable equilibrium. This, however, is not a problem, as 'avoidance of traps' results as in Section 4.3 of Borkar (2008) (also see Benaim, 1996; Brandiére and Duflo, 1996; Pemantle, 1990) ensure that if the noise is rich enough in a certain precise sense, unstable equilibria are avoided with probability one.

Remark 9 *Note that the gradient descent is a stochastic approximation scheme which itself averages out the noise. So in principle the averaging over k steps at the SP stage in the original algorithm can be skipped. This means that for a stochastic gradient descent scheme, we cut down the cost of function evaluation even further. The simulations in the next section confirm that good results are obtained without averaging over SP iterations. There is, however, a standard trade-off involved between per step computation / speed of convergence, and fluctuations (equivalently, variance) of the estimates: any additional averaging improves the latter at the expense of the former.*

3.3 Numerical experiments

We compare following three algorithms.

1. *Actual Gradient Descent*

   This is the classical stochastic gradient descent with exact gradient.

Algorithm 2 Stochastic Gradient Decent with Compressive Sensing

---

Initialization:

$x(0) = x_{initial}, A \leftarrow random\ Gaussian\ matrix$
$a(n)$ be a sequence that satisfies the properties of stepsize listed above.

Iteration: *Repeat until convergence criteria is met at* $n = n^{\#}$. *At* n^{th} *iteration:*

$$\begin{aligned} y(n) &\leftarrow A\nabla f(x(n)) + error \\ \widetilde{\nabla f}(x(n)) &\leftarrow l_1 - \ minimization\ with\ Homotopy(y(n), A) \\ x(n+1) &\leftarrow x(n) - a(n)[\widetilde{\nabla f}(x(n))] \end{aligned}$$

Output: *Approximate minimizer of* f *i.e.* $x(n^{\#})$

Here $Homotopy(y(n), A)$ denotes the l_1-recovery from observations $y(n)$ and Gaussian random matrix A using the homotopy method.

2. *Accelerated Gradient Method*

   Accelerated gradient scheme was proposed by Nesterov (Nesterov, 1983). While Gradient Descent algorithm has a rate of convergence of order $1/s$ after s steps, Nesterov's method achieves a rate of order $1/s^2$. We implement the method here to achieve an improvement in the time complexity further. The idea is to replace the n^{th} iteration above by the following.

At n^{th} iteration:

$$\begin{aligned} y(n) &\leftarrow A\nabla f(x(n)) + \text{error} \\ \widetilde{\nabla f}(x(n)) &\leftarrow l_1 - \ \text{minimization with } Homotopy(y(n), A) \\ z(n+1) &\leftarrow x(n) - a(n)[\widetilde{\nabla f}(x(n))] \\ x(n+1) &\leftarrow (1-\gamma(n))z(n+1) + \gamma(n)z(n) \end{aligned}$$

where, λ and γ are as follows:

$$\lambda(0) = 0,\ \lambda(n) = \frac{1+\sqrt{1+4\lambda^2(n-1)}}{2},\ \text{and } \gamma(n) = \frac{1-\lambda(n)}{\lambda(n+1)}.$$

This gives us faster convergence towards the minimum.

3. *Adaptive Method*

   Another way to achieve a faster convergence rate is to perform the l_1-minimization adaptively with the gradient descent. The idea is to again use the homotopy method for l_1-minimization but this part of the algorithm is run for very few iterations. The intermediate approximation of ∇f is then used for performing the stochastic gradient descent. As expected, the errors are high in the beginning but the convergence is faster.

We consider the following function to test our algorithms:

$$f(x) = (x^T M_1^T M_1 x)^3 + (x^T M_2^T M_2 x)^2 + x^T M_1^T M_2 x \tag{15}$$

where, function $f : \mathbb{R}^n \mapsto \mathbb{R}$ and M_1, M_2 are $n \times s$ random matrices. This is to ensure sparsity of the gradient. Here, $n = 25000$ and number of non-zero entries in each column of M_1 and M_2 are $s = 3$. Number of measurements, $m = 500$. A is a $n \times m$ random Gaussian matrix.

Figure 10 show the comparisons between various algorithms described above for the same function.

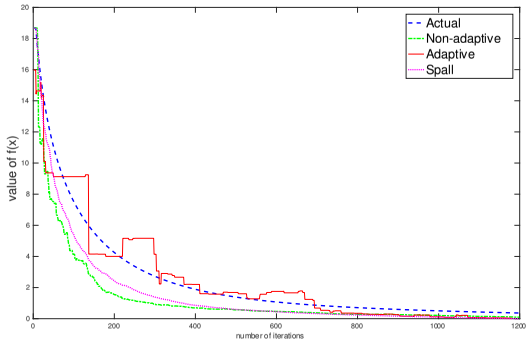


Figure 10: Comparison of Gradient descent with actual gradient ∇f and estimated gradient $\widetilde{\nabla f}$ using non-adaptive, adaptive schemes and Spall's SPSA. Here, $n = 25000, s = 3$ and $m = 500$

Algorithm Used	n = 1000 (sec)	n = 10000 (sec)	n = 25000 (sec)
Adaptive Method	4.3	62.3	85.4
Without Adaptive	50.572	612.3	1004.5
Spall's SPSA	25.0	459.8	864.5
Actual Gradient Method	64.1	4641.12	7526

As expected, adaptive method turns out to be faster compared to the non-adaptive method which in turn is much faster than the algorithm that computes actual gradients. Incidentally, the classical scheme all but converges in under 1000 iterations. Even so it takes more time than the other two which take more iterations. This is because of the heavy per iterate computation for the classical scheme. From the above table it is clear that as the dimensionality of the problem increases, adaptive method proves more and more useful compared to the other two algorithms.

We also compared our method with the method proposed in (Mukherjee, Wu and Zhou, 2010) (See Figure 11). The function in (15) is used for the comparison with $n = 10000$ and $s = 50$. Figure 12 is a scaled down version with $n = 10000, s = 100$ and $m = 500$. Number of samples for the SGL method were 10, chosen with neighbourhood radius 0.05.

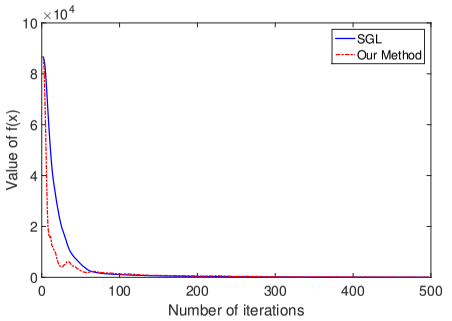


Figure 11: Comparison of Gradient descent using estimated gradient from proposed method vs the SGL method proposed in (Mukherjee, Wu and Zhou, 2010). Here, $n = 10000, s = 50$ and $m = 500$. Number of samples for the SGL method = 10.

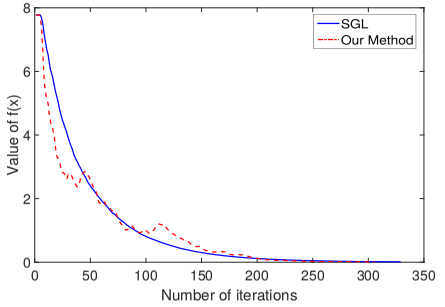


Figure 12: Comparison of Gradient descent using estimated gradient from proposed method vs the SGL method proposed in (Mukherjee, Wu and Zhou, 2010). Here, $n = 10000, s = 100$ and $m = 500$. Number of samples for the SGL method = 10.

The time taken by the SGL method and our method was 993 and 83 seconds respectively. As mentioned earlier, the aim of this paper is to provide a good estimation of gradient when the function evaluations are expensive. In such cases, our method would provide a significant gain in terms of function evaluations needed. While in this example we do see a significant improvement in time taken for the estimation, there is no a priori reason to always expect it. It will indeed be the case when the function evaluations are 'expensive' in terms of the time they take. One expects this to be so when the ambient dimension is high.

3.3.1 An Example: Longitude Estimation

In this section, we test our method on real data. Gradient estimation technique proposed in this paper is applied on UJIIndoorLoc Data Set (Torres-Sospedra et. al, 2014) to estimate the longitude information from signal strengths of 520 wireless access points. The Data-set has 19937 samples for training and 1111 samples for testing. We assume that longitude l is linearly dependent on signal strengths of access points x. Let $\theta \in \mathbb{R}^n$ be the vector assigning weights of signal strengths to each access points. Then, $l(\theta) = \theta^T x$. We recover θ by minimizing regularized l_1 norm of the error:

$$f(\theta) = \sum_{i=1}^{M} \frac{1}{M} \|l_{actual} - l(\theta)\|_1 + \lambda \|\theta\|$$

where, l_{actual} denotes the actual longitude. In this example, $M = 19937$, $n = 520$ and $m = 100$, where m is as in theorem 2.3.

Note that we do not have a closed form expression for the gradient of $f(\theta)$. We compare the proposed method with Spall's SPSA. Parameters like k (number of trials of SP) and iteration for l_1 minimization are chosen so that both methods converge empirically in least number of iterations. Taking into account the higher error in SPSA, we take number of trials of SP to be 20 for Spall's SPSA and 10 for our method. Maximum number of iterations in Homotopy method are limited to 50.

Algorithm:	Proposed Method	Spall's SPSA
Mean percentage error on training data	5.43	5.54
Mean percentage error on test data	5.55	5.48
Time taken to train (sec)	16.77	84.81

We can see that both methods obtain similar performance but the method proposed in this paper is much faster. Figure 13 shows the training performance of both methods. Figures 14, 15 show percentage error in reconstruction of training and test data respectively. Percentage error is sorted for better visualization.

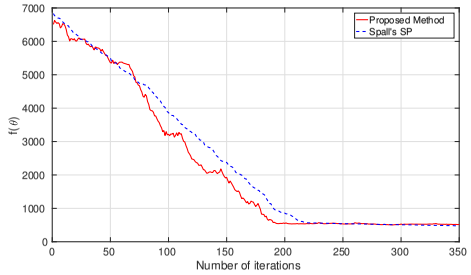


Figure 13: Optimizing $f(\theta)$ using Spall's SPSA and proposed method.

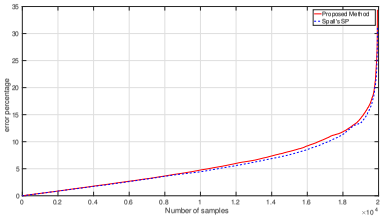


Figure 14: Sorted percentage error on train data.

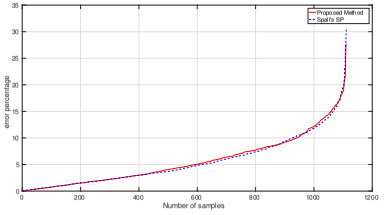


Figure 15: Sorted percentage error on test data.

4. Concluding remarks

We have proposed an estimation scheme for gradient in high dimensions that combines ideas from Spall's SPSA with compressive sensing and thereby tries to economize on the number of function evaluations. This has theoretical justification by the results of (Austin, 2016). Our method can be extremely useful when the function evaluation is very expensive, e.g., when a single evaluation is the output of a long simulation. This situation does not seem to have been addressed much in literature. In very high dimensional problems with sparse gradient, computing estimates for partial derivatives in every direction is inefficient because of the large number of function evaluations needed. SP simplifies the problem of repeated function evaluation by concentrating on a single *random* direction at each step. When the gradient vectors in such cases live in a lower dimensional subspace, it also makes sense to exploit ideas from compressive sensing. We have computed the error bound in this case and have also shown theoretically that this kind of estimation of gradient works well with high probability for the gradient descent problems and in other high dimensional problems such as estimating EGOP in manifold learning where gradients are actually low-dimensional and gradient estimation is relevant. Simulations show that our method works much better than pure SP.

Acknowledgments

We thank GPU Centre of Excellence, IIT Bombay for providing us with the facility to carry out simulations and Prof. Chandra Murthy of Indian Institute of Science for helpful discussions regarding compressive sensing.

References

T. Austin. On the failure of concentration for the ℓ_∞-ball. *Israel Journal of Mathematics* 211(1):221-238, 2016.

S. A. Bandeira, M. Fickus, G. D. Mixon, and P. Wong. The road to deterministic matrices with the restricted isometry property. *Journal of Fourier Analysis and Applications* 19(6):1123-1149, 2013.

M. Benaim, A dynamical system approach to stochastic approximation, *SIAM Journal of Control and Optimization* 34(2):437-472, 1996.

O. Brandiére and M. Duflo, Les algorithmes stochastiques contourment-ils les pièges?, *Annals de l'Institut Henri Poincaré* 32(3):395-427, 1996.

Vivek S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Hindustan Book Agency, New Delhi, and Cambridge University Press, Cambridge, UK, 2008.

E. Candés and T. Tao. Near optimal signal recovery from random projections: universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406-5425, 2006.

E. Candés, J. Romberg J. and T. Tao. Stable signal recovery from incomplete and inaccurate measurements, *Comm. Pure Appl. Math.*, 59(8):1207-1223, 2006.

E. Candés, M. Rudelson, T. Tao and R. Vershynin. Error correction via linear programming. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 295-308, 2005.

L. W. Chan, K. Charan, D. Takhar, K. F. Kelly, R. G. Baraniuk, G. Richard and D. M. Mittleman. A single-pixel terahertz imaging system based on compressed sensing. *Applied Physics Letters*, 93(12):121105-121105-3, 2008.

D. L. Donoho and Y. Tsaig. Fast solution of l_1-norm minimization problems when the solution may be sparse, *IEEE Transactions on Information Theory*, 54:4789-4812, 2008.

M. F. Duarte, M. A. Davenport, T. Dharmpal, J. N. Laska, T. Sun, K. F. Kelly and R. G. Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, 25(2):83-91, March 2008.

S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing*, Birkhäuser, New York, 2013.

I. Grondman, L. Busoniu, G. A. D. Lopes and R. Babuska. A Survey of actor-critic reinforcement learning: standard and natural policy gradients. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 42(6):1291-1307, Nov. 2012.

M. W. Hirsch. Convergent Activation Dynamics in Continuous Time Networks. *Neural Networks*, 2(5): 331-349, 1989.

D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black box functions. *Journal of Global Optimization*, 13(4):455-492, 1998.

G. Joseph and C. R. Murthy. A Non-iterative Online Bayesian Algorithm for the Recovery of Temporally Correlated Sparse Vectors. *IEEE Transactions on Signal Processing*, 65(20):5510-5525, 2017.

M. Kabanava and H. Rauhut. Analysis l_1-recovery with frames and gaussian measurements. *Acta Applicandae Mathematicae*, 140(1):173-195, 2015.

S. Kamal. On the convergence, lock-in probability, and sample complexity of stochastic approximation. *SIAM Journal on Control and Optimization*, 48(8):5178-5192, 2010.

K. C. Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316-327, 1991.

S. Mukherjee, Q. Wu, and D. X. Zhou. Learning gradients on manifolds. *Bernoulli*, 16(1):181-207, 2010.

Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372-376, 1983.

P. Pandita, I. Bilionis and J. Panchal. Extending Expected Improvement for High-dimensional Stochastic Optimization of Expensive Black-Box Functions. Journal of Mechanical Design 138(11):111412, 2016.

R. Pemantle. Nonconvergence to unstable points in urn models and stochastic approximation. *Annals of Probability*, 18(2):698-712, 1990.

S. Shan and G. Gary Wang. Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Structural and Multidisciplinary Optimization*, 41(2), 219, 2010.

J. C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332-341, 1992.

R. S. Sutton, D. McAllester, S. Singh and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems*, 12:1057-1063, MIT Press, 2000.

V. B. Tadic and A. Doucet. Asymptotic bias of stochastic gradient search. *The Annals of Applied Probability* 27(6):3255-3304, 2017.

G. Thoppe and V. S. Borkar. A concentration bound for stochastic approximation via Alexeev's formula. *arXiv*:1506-08657v2 [math.OC], 2015.

J. Torres-Sospedra, R. Montoliu, A. Martnez-Us, J. P. Avariento, T. J. Arnau, M. Benedito-Bordonau, and J. Huerta. UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems. *Indoor Positioning and Indoor Navigation (IPIN), 2014 International Conference*, 261-270, 2014.

S. Trivedi, J. Wang, S. Kpotufe and G. Shakhnarovich. A consistent estimator of the expected gradient outerproduct. *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*:819-828, July 2014.

J. A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389-434, 2012.

Q. Wu, J. Guinney, M. Maggioni and S. Mukherjee. Learning gradients : predictive models that infer geometry and statistical dependence. *Journal of Machine Learning Research*, 11(1922):2175-2198, 2010.

Y. Xia, H. Tong, W. Li, and L.-X. Zhu. An adaptive estimation of dimension reduction space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(3):363-410, 2002.

H. Xu, C. Caramanis and S. Mannor. Statistical optimization in high dimensions. *Operations research*, 64(4):958-979, 2016.

A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Fast l_1-minimization algorithms and an application in robust face recognition: a review. *ICIP*, 2010.

T. Zhao, H. Hachiya, G. Niu and M. Sugiyama. Analysis and improvement of policy gradient estimation. *Neural Networks*, 26:118-129, 2012.

Journal of Machine Learning Research 18 (2018) 1-38 Submitted 11/15; Revised 12/17; Published 4/18

Principled Selection of Hyperparameters in the Latent Dirichlet Allocation Model

Clint P. George CLINTPG@UFL.EDU
Informatics Institute
University of Florida
Gainesville, FL 32611, *USA*

Hani Doss DOSS@STAT.UFL.EDU
Department of Statistics
University of Florida
Gainesville, FL 32611, *USA*

Editor: David Blei

Abstract

Latent Dirichlet Allocation (LDA) is a well known topic model that is often used to make inference regarding the properties of collections of text documents. LDA is a hierarchical Bayesian model, and involves a prior distribution on a set of latent topic variables. The prior is indexed by certain hyperparameters, and even though these have a large impact on inference, they are usually chosen either in an ad-hoc manner, or by applying an algorithm whose theoretical basis has not been firmly established. We present a method, based on a combination of Markov chain Monte Carlo and importance sampling, for estimating the maximum likelihood estimate of the hyperparameters. The method may be viewed as a computational scheme for implementation of an empirical Bayes analysis. It comes with theoretical guarantees, and a key feature of our approach is that we provide theoretically-valid error margins for our estimates. Experiments on both synthetic and real data show good performance of our methodology.

Keywords: Empirical Bayes inference, latent Dirichlet allocation, Markov chain Monte Carlo, model selection, topic modelling.

1. Introduction

Latent Dirichlet Allocation (LDA, Blei et al. 2003) is a model that is used to describe high-dimensional sparse count data represented by feature counts. Although the model can be applied to many different kinds of data, for example collections of annotated images and social networks, for the sake of concreteness, here we focus on data consisting of a collection of documents. Suppose we have a corpus of documents, say a collection of news articles, and these span several different topics, such as sports, medicine, politics, etc. We imagine that for each word in each document, there is a latent (i.e. unobserved) variable indicating a topic from which that word is drawn. There are several goals, but two principal ones are to recover an interpretable set of topics, and to make inference on the latent topic variables for each document.

To describe the LDA model, we first set up some terminology and notation. There is a vocabulary $\mathcal{V}$ of V words; typically, this is taken to be the union of all the words in all the documents of the corpus, after removing stop (i.e. uninformative) words. (Throughout, we use "word" to refer to either an actual word, or to a phrase, such as "heart attack"; LDA has implementations that deal with each of these.) There are D documents in the corpus, and for $d = 1, \ldots, D$, document d has n_d words, $w_{d1}, \ldots, w_{dn_d}$. The order of the words is considered uninformative, and so is neglected. Each word is represented as an index vector of dimension V with a 1 at the s^{th} element, where s denotes the term selected from the vocabulary. Thus, document d is represented by the matrix $\boldsymbol{w}_d = (w_{d1}, \ldots, w_{dn_d})$ and the corpus is represented by the list $\boldsymbol{w} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_D)$. The number of topics, K, is finite and known. By definition, a topic is a distribution over $\mathcal{V}$, i.e. a point in the simplex $\mathbb{S}_V = \{a \in \mathbb{R}^V : a_1, \ldots, a_V \geq 0, \sum_{j=1}^V a_j = 1\}$. For $d = 1, \ldots, D$, for each word w_{di}, z_{di} is an index vector of dimension K which represents the latent variable that denotes the topic from which w_{di} is drawn. The distribution of $z_{d1}, \ldots, z_{dn_d}$ will depend on a document-specific variable θ_d which indicates a distribution on the topics for document d.

We will use $\mathrm{Dir}_L(a_1, \ldots, a_L)$ to denote the finite-dimensional Dirichlet distribution on the simplex $\mathbb{S}_L$. Also, we will use $\mathrm{Mult}_L(b_1, \ldots, b_L)$ to denote the multinomial distribution with number of trials equal to 1 and probability vector $(b_1, \ldots, b_L)$. We will form a $K \times V$ matrix $\boldsymbol{\beta}$, whose t^{th} row is the t^{th} topic (how $\boldsymbol{\beta}$ is formed will be described shortly). Thus, $\boldsymbol{\beta}$ will consist of vectors $\beta_1, \ldots, \beta_K$, all lying in $\mathbb{S}_V$. The LDA model is indexed by hyperparameters $\eta \in (0, \infty)$ and $\boldsymbol{\alpha} \in (0, \infty)^K$. It is represented graphically in Figure 1, and described formally by the following hierarchical model:

1. $\beta_t \overset{\text{iid}}{\sim} \mathrm{Dir}_V(\eta, \ldots, \eta)$, $t = 1, \ldots, K$.
2. $\theta_d \overset{\text{iid}}{\sim} \mathrm{Dir}_K(\boldsymbol{\alpha})$, $d = 1, \ldots, D$, and the θ_d's are independent of the β_t's.
3. Given $\theta_1, \ldots, \theta_D$, $z_{di} \overset{\text{iid}}{\sim} \mathrm{Mult}_K(\theta_d)$, $i = 1, \ldots, n_d$, $d = 1, \ldots, D$, and the D matrices $(z_{11}, \ldots, z_{1n_1}), \ldots, (z_{D1}, \ldots, z_{Dn_D})$ are independent.
4. Given $\boldsymbol{\beta}$ and the z_{di}'s, the w_{di}'s are independently drawn from the row of $\boldsymbol{\beta}$ indicated by z_{di}, $i = 1, \ldots, n_d$, $d = 1, \ldots, D$.

From the description of the model, we see that there is a latent topic variable for every word that appears in the corpus. Thus it is possible that a document spans several topics. Also, because $\boldsymbol{\beta}$ is chosen once, at the top of the hierarchy, it is shared among the D documents. Thus the model encourages different documents to share the same topics, and moreover, all the documents in the corpus share a single set of topics defined by $\boldsymbol{\beta}$.

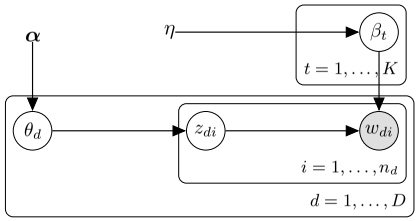


Figure 1: Graphical model representation for LDA. Nodes denote random variables, shaded nodes denote observed variables, edges denote conditional dependencies, and plates denote replicated processes.

Let $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_D)$, $\boldsymbol{z}_d = (z_{d1}, \ldots, z_{dn_d})$ for $d = 1, \ldots, D$, $\boldsymbol{z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_D)$, and let $\boldsymbol{\psi} = (\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{z})$. The model is indexed by the hyperparameter vector $h = (\eta, \boldsymbol{\alpha}) \in (0, \infty)^{K+1}$. For any given h, lines 1–3 induce a prior distribution on $\boldsymbol{\psi}$, which we will denote by ν_h. Line 4 gives the likelihood. The words $\boldsymbol{w}$ are observed, and we are interested in $\nu_{h,\boldsymbol{w}}$, the posterior distribution of $\boldsymbol{\psi}$ given $\boldsymbol{w}$ corresponding to ν_h. In step 2 it is common to take the distribution of the θ_d's to

be a symmetric Dirichlet, although arbitrary Dirichlets are sometimes used. Our model allows for arbitrary Dirichlets, for the sake of generality, but in all our examples we use symmetric Dirichlets because a high-dimensional hyperparameter can cause serious problems. We return to this point at the end of Section 2.1.

The hyperparameter vector h is not random, and must be selected in advance. It has a strong effect on the distribution of the parameters of the model. For example, when η is large, the topics tend to be probability vectors which spread their mass evenly among many words in the vocabulary, whereas when η is small, the topics tend to put most of their mass on only a few words. Also, in the special case where $\boldsymbol{\alpha} = (\alpha, \ldots, \alpha)$, so that $\mathrm{Dir}_K(\boldsymbol{\alpha})$ is a symmetric Dirichlet indexed by the single parameter α, when α is large, each document tends to involve many different topics; on the other hand, in the limiting case where $\alpha \to 0$, each document involves a single topic, and this topic is randomly chosen from the set of all topics.

The preceding paragraph is about the effect of h on the prior distribution of the parameters. We may think about the role of h on statistical inference by considering posterior distributions. Let g be a function of the parameter $\boldsymbol{\psi}$. For example, $g(\boldsymbol{\psi})$ might be the indicator of the event $\|\theta_i - \theta_j\| \leq \epsilon$, where i and j are the indices of two particular documents, ϵ is some user-specified small number, and $\|\cdot\|$ denotes ordinary Euclidean distance in $\mathbb{R}^K$. In this case, the value of $g(\boldsymbol{\psi})$ gives a way of determining whether the topics for documents i and j are nearly the same ($g(\boldsymbol{\psi}) = 1$), or not ($g(\boldsymbol{\psi}) = 0$). Of interest then is the posterior probability $\nu_{h,\boldsymbol{w}}(\|\theta_i - \theta_j\| \leq \epsilon)$, which is given by the integral $\int g(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi})$. In another example, the function g might be taken to measure the distance between two topics of interest. In Section 2.4 we demonstrate empirically that the posterior expectation given by the integral $\int g(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi})$ can vary considerably with h.

To summarize: the hyperparameter h can have a strong effect not only on the prior distribution of the parameters in the model, but also on their posterior distribution; therefore it is important to choose it carefully. Yet in spite of the very widespread use of LDA, there is no method for choosing the hyperparameter that has a firm theoretical basis. In the literature, h is sometimes selected in some ad-hoc or arbitrary manner. A principled way of selecting it is via maximum likelihood: we let $m_{\boldsymbol{w}}(h)$ denote the marginal likelihood of the data as a function of h, and use $\hat{h} = \arg\max_h m_{\boldsymbol{w}}(h)$ which is, by definition, the empirical Bayes choice of h. We will write $m(h)$ instead of $m_{\boldsymbol{w}}(h)$ unless we need to emphasize the dependence on $\boldsymbol{w}$. Unfortunately, the function $m(h)$ is analytically intractable: $m(h)$ is the likelihood of the data with all latent variables integrated or summed out, and from the hierarchical nature of the model, we see that $m(h)$ is a very large sum, because we are summing over all possible values of $\boldsymbol{z}$. Blei et al. (2003) propose estimating $\arg\max_h m(h)$ via a combination of the EM algorithm and "variational inference" (VI-EM). Very briefly, $\boldsymbol{w}$ is viewed as "observed data," and $\boldsymbol{\psi}$ is viewed as "missing data." Because the "complete data likelihood" $p_h(\boldsymbol{\psi}, \boldsymbol{w})$ is available, the EM algorithm is a natural candidate for estimating $\arg\max_h m(h)$, since $m(h)$ is the "incomplete data likelihood." But the E-step in the algorithm is infeasible because it requires calculating an expectation with respect to the intractable distribution $\nu_{h,\boldsymbol{w}}$. Blei et al. (2003) substitute an approximation to this expectation. Unfortunately, because there are no useful bounds on the approximation, and because the approximation is used at every iteration of the algorithm, there are no results regarding the theoretical properties of this method. Wallach (2006) (see also Wallach (2008)) proposed a "Gibbs-EM" algorithm, in which the E-step is approximated by a Markov chain Monte Carlo estimate. This method can perform well empirically but, as for the VI-EM algorithm, its theoretical validity has not been established. The

advantages and disadvantages of these two approximations to the EM algorithm are discussed in Section 5.1.

Wallach et al. (2009) give an overview of a class of methods for estimating a combination of some parameter components and some hyperparameter components and, in principle, these procedures could be adapted to the problem of estimating h. The methods they present differ from EM-based approaches in two fundamental respects: (1) they work with an objective function which is not the marginal likelihood function $m(h)$, but rather a measure of the "predictive performance of the LDA model indexed by h," and (2) evaluation of their objective function at hyperparameter value h requires running a Markov chain, and this has to be done "for each value of h" before doing the maximization of the objective function, which can impose a heavy computational burden. This paper is discussed further in Section 5.

Another approach for dealing with the problem of having to make a choice of the hyperparameter vector is the fully Bayes approach, in which we simply put a prior on the hyperparameter vector, that is, add one layer to the hierarchical model. For example, we can either put a flat prior on each component of the hyperparameter, or put a gamma prior instead. While this approach can be useful, there are reasons why one may want to avoid it. On the one hand, if we put a flat prior then one problem is that we are effectively skewing the results towards large values of the hyperparameter components. A more serious problem is that the posterior may be improper. In this case, insidiously, if we use Gibbs sampling to estimate the posterior, it is possible that all conditionals needed to implement the sampler are proper; but Hobert and Casella (1996) have shown that the Gibbs sampler output may not give a clue that there is a problem. On the other hand, if we use a gamma prior, then at least in the case of a symmetric Dirichlet on the θ_d's, we have not made things any easier: we have to specify four gamma hyperparameters. Another reason to avoid the fully Bayes approach is that, in broad terms, the general interest in empirical Bayes methods arises in part from a desire to select a specific value of the hyperparameter vector because this gives a model that is more parsimonious and interpretable. This point is discussed more fully (in a general context) in George and Foster (2000) and Robert (2001, Chapter 7).

In the present paper we show that while it is not possible to compute $m(h)$ itself, it is nevertheless possible, with a single MCMC run, to estimate the entire function $m(h)$ up to a multiplicative constant. Before proceeding, we note that if c is a constant, then the information regarding h given by the two functions $m(h)$ and $cm(h)$ is the same: the same value of h maximizes both functions, and the second derivative matrices of the logarithm of these two functions are identical. In particular, the Hessians of the logarithm of these two functions at the maximum (i.e. the observed Fisher information) are the same and, therefore, the standard point estimates and confidence regions based on $m(h)$ and $cm(h)$ are identical. Let g be a function of $\boldsymbol{\psi}$ and let $I(h) = \int g(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi})$ denote the posterior expectation of $g(\boldsymbol{\psi})$. We also show that it is possible to estimate the entire function $I(h)$ with a single MCMC run.

As we will see in Section 2, our approach for estimating $m(h)$ up to a single multiplicative constant and $I(h)$ has two requirements: (i) we need a formula for the ratio $\nu_{h_1}(\boldsymbol{\psi})/\nu_{h_2}(\boldsymbol{\psi})$ for any two hyperparameter values h_1 and h_2, and (ii) for any hyperparameter value h, we need an ergodic Markov chain whose invariant distribution is the posterior $\nu_{h,\boldsymbol{w}}$. This paper is organized as follows. In Section 2 we explain our method for estimating the function $m(h)$ up to a single multiplicative constant (and hence its argmax) and for estimating the family of posterior expectations $\{I(h),\, h \in \mathcal{H}\}$; and we also explain how to form error margins for our estimates, paying particular attention to theoretical underpinnings. Additionally, we provide the formula for the ratio $\nu_{h_1}(\boldsymbol{\psi})/\nu_{h_2}(\boldsymbol{\psi})$.

In Section 3 we consider synthetic data sets generated from a simple model in which h is low dimensional and known, and we show that our method correctly estimates the true value of h. In Section 4 we describe two Markov chains which satisfy requirement (ii) above. In Section 5 we compare, both theoretically and empirically, the various methods of estimating the maximizer of the marginal likelihood function, in terms of accuracy. Then we compare the various choices of the hyperparameter that are used in the literature—those that are ad-hoc and those that estimate the maximizer of the marginal likelihood function—through a standard criterion that is used to evaluate topic models, and we show that our method performs favorably. In Section 6 we make some concluding remarks, and the Appendix contains some of the technical material that is needed in the paper.

2. Estimation of the Marginal Likelihood up to a Multiplicative Constant and Estimation of Posterior Expectations

This section consists of four parts. In Section 2.1 we show how the marginal likelihood function can be estimated (up to a constant) with a single MCMC run. Section 2.2 concerns estimation of the posterior expectation of a function g of the parameter $\boldsymbol{\psi}$, given by the integral $\int g(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi})$, and which depends on h. We show how the entire family of posterior expectations $\{I(h),\, h \in \mathcal{H}\}$ can be estimated with a single MCMC run. In Section 2.3 we explain that the simple estimates given in Sections 2.1 and 2.2 can have large variances, and we present estimates which are far more reliable. In Section 2.4 we illustrate our methodology on a corpus created from Wikipedia.

Let $\mathcal{H} = (0, \infty)^{K+1}$ be the hyperparameter space. For any $h \in \mathcal{H}$, ν_h and $\nu_{h,\boldsymbol{w}}$ are prior and posterior distributions, respectively, of the vector $\boldsymbol{\psi} = (\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{z})$, for which some components are continuous and some are discrete. We will use $\ell_{\boldsymbol{w}}(\boldsymbol{\psi})$ to denote the likelihood function (which is given by line 4 of the LDA model).

2.1 Estimation of the Marginal Likelihood up to a Multiplicative Constant

Note that $m(h)$ is the normalizing constant in the statement "the posterior is proportional to the likelihood times the prior," i.e.

$$\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi}) = \frac{\ell_{\boldsymbol{w}}(\boldsymbol{\psi})\nu_h(\boldsymbol{\psi})}{m(h)}.$$

Now suppose that we have a method for constructing a Markov chain on $\boldsymbol{\psi}$ whose invariant distribution is $\nu_{h,\boldsymbol{w}}$ and which is ergodic. Two Markov chains which satisfy these criteria are discussed in later in this section. Let $h_* \in \mathcal{H}$ be fixed but arbitrary, and let $\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots$ be an ergodic Markov chain with invariant distribution $\nu_{h*,\boldsymbol{w}}$. For any $h \in \mathcal{H}$, as $n \to \infty$ we have

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n} \frac{\nu_h(\boldsymbol{\psi}_i)}{\nu_{h_*}(\boldsymbol{\psi}_i)} &\xrightarrow{\text{a.s.}} \int \frac{\nu_h(\boldsymbol{\psi})}{\nu_{h_*}(\boldsymbol{\psi})}\, d\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi}) \\
&= \frac{m(h)}{m(h_*)} \int \frac{\ell_{\boldsymbol{w}}(\boldsymbol{\psi})\nu_h(\boldsymbol{\psi})/m(h)}{\ell_{\boldsymbol{w}}(\boldsymbol{\psi})\nu_{h_*}(\boldsymbol{\psi})/m(h_*)}\, d\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi}) \\
&= \frac{m(h)}{m(h_*)} \int \frac{\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi})}{\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi})}\, d\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi}) \\
&= \frac{m(h)}{m(h_*)}.
\end{aligned} \tag{2.1}$$

The almost sure convergence statement in (2.1) follows from ergodicity of the chain. (There is a slight abuse of notation in (2.1) in that we have used $\nu_{h_*,\boldsymbol{w}}$ to denote a probability measure when we write $d\nu_{h_*,\boldsymbol{w}}$, whereas in the integrand, ν_h, ν_{h_*}, and $\nu_{h_*,\boldsymbol{w}}$ refer to probability densities.) The significance of (2.1) is that this result shows that we can estimate the entire family $\{m(h)/m(h_*),\, h \in \mathcal{H}\}$ with a single Markov chain run. Since $m(h_*)$ is a constant, the remarks made in Section 1 apply, and we can estimate $\arg\max_h m(h)$. The usefulness of (2.1) stems from the fact that the average on the left side involves *only the priors*, so we effectively bypass having to deal with the posterior distributions.

The development in (2.1) is not new (although we do not know who first noticed it), and the estimate on the left side of (2.1) is not the one we will ultimately use (cf. Section 2.3); we present (2.1) primarily for motivation. Note that (2.1) is generic, i.e. it is not specific to the LDA model: it is potentially valid for any Bayesian model for which we have a data vector $\boldsymbol{w}$, a corresponding likelihood function $\ell_{\boldsymbol{w}}(\boldsymbol{\psi})$, and parameter vector $\boldsymbol{\psi}$ having prior ν_h, with hyperparameter $h \in \mathcal{H}$. We now discuss carefully the scope of its applicability. In order to be able to use (2.1) to obtain valid estimators of the family $m(h)/m(h_*),\, h \in \mathcal{H}$, we need the following.

C1 A closed-form expression for the ratio of densities $\nu_h(\boldsymbol{\psi})/\nu_{h_*}(\boldsymbol{\psi})$ for some fixed $h_* \in \mathcal{H}$.

C2 A method for generating an ergodic Markov chain with invariant distribution $\nu_{h_*,\boldsymbol{w}}$.

We need C1 in order to write down the estimators, and we need C2 for the estimators to be valid.

For notational convenience, let $B_n(h) = (1/n)\sum_{i=1}^{n}[\nu_h(\boldsymbol{\psi}_i)/\nu_{h_*}(\boldsymbol{\psi}_i)]$, and define $B(h) = m(h)/m(h_*)$. In order to use (2.1) to obtain a valid estimator, together with a confidence interval (confidence set, if $\dim(h) > 1$) for $\arg\max_h m(h)$, we need in addition the following.

C3 A result that says that the convergence in the first line of (2.1) is uniform in h.

C4 A result that says that if $G_n(h)$ is the centered and scaled version of the estimate on the left side of (2.1) given by $G_n(h) = n^{1/2}(B_n(h) - B(h))$, then $G_n(\cdot)$ converges in distribution to a Gaussian process indexed by h.

We now explain these last two conditions. Generally speaking, for real-valued functions f_n and f defined on $\mathcal{H}$, the pointwise convergence condition $f_n(h) \to f(h)$ for each $h \in \mathcal{H}$ does not imply that $\arg\max_h f_n(h) \to \arg\max_h f(h)$. Indeed, counterexamples are easy to construct, and in Section A.2 of the Appendix we provide a simple one. In order to be able to conclude that $\arg\max_h f_n(h) \to \arg\max_h f(h)$, which is a global condition, we need the convergence of f_n to f to be uniform (this is discussed rigorously in Section A.2 of the Appendix). Hence we need C3. Regarding confidence intervals (or sets) for $\arg\max_h f(h)$, we note that $B_n(h)$ is simply an average, and so under suitable regularity conditions, it satisfies a central limit theorem (CLT). However, we are not interested in a central limit theorem for $B_n(h)$, but rather in a CLT for $\arg\max_h B_n(h)$. In this regard, C4 is a "uniform in h CLT" that is necessary to obtain a CLT of the form $n^{1/2}(\arg\max_h B_n(h) - \arg\max_h B(h)) \xrightarrow{d} \mathcal{N}(0, \Sigma)$ for some positive definite matrix Σ, which is what is needed to form confidence sets for $\arg\max_h B(h)$. Again, this is discussed rigorously in Section A.2 of the Appendix.

We now discuss Conditions C1–C4. Condition C1 is satisfied by the LDA model, and in Section A.1 of the Appendix we show that the ratio of densities ν_h/ν_{h_*} is given by

$$\frac{\nu_h(\boldsymbol{\psi})}{\nu_{h_*}(\boldsymbol{\psi})} = \left[\prod_{d=1}^{D}\left(\frac{\Gamma\left(\sum_{j=1}^K \alpha_j\right)}{\prod_{j=1}^K \Gamma(\alpha_j)}\frac{\prod_{j=1}^K \Gamma(\alpha_j^*)}{\Gamma\left(\sum_{j=1}^K \alpha_j^*\right)}\prod_{j=1}^K \theta_{dj}^{\alpha_j-\alpha_j^*}\right)\right]\left[\prod_{j=1}^K\left(\frac{\Gamma(V\eta)}{\Gamma(\eta)^V}\frac{\Gamma(\eta^*)^V}{\Gamma(V\eta^*)}\prod_{t=1}^V \beta_{jt}^{\eta-\eta^*}\right)\right], \tag{2.2}$$

where $h_* = (\eta^*, \boldsymbol{\alpha}^*)$.

There are many extensions and variants of the LDA model described in Section 1—too many to even list them all here—and versions of (2.2) can be obtained for these models. This has to be done separately for each case. The features of the LDA model that make it possible to obtain a ratio of densities formula are that it is a hierarchical model, and at every stage the distributions are explicitly finite dimensional. For other models, a ratio of densities formula is obtainable routinely as long as these features exist: when they do, we have a closed-form expression for the prior distribution $\nu_h(\boldsymbol{\psi})$, and hence a closed-form expression for the ratio $\nu_h(\boldsymbol{\psi})/\nu_{h_*}(\boldsymbol{\psi})$.

Unfortunately, a ratio of densities formula is not always available. A prominent example is the "Hierarchical Dirichlet Processes" model introduced in Teh et al. (2006), which effectively allows infinitely many topics but with finitely many realized in any given document. Very briefly, in this model, for word i in document d, there is an unobserved topic ψ_{di}. The latent topic vector $\boldsymbol{\psi}_d = (\psi_{d1}, \ldots, \psi_{dn_d})$ has a complicated joint distribution with strength of dependence governed by a hyperparameter h_1 (the precision parameter of the Dirichlet process in the middle of the hierarchy), and the D vectors $\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_D$ also have a complicated dependence structure, with strength of dependence governed by a hyperparameter h_2 (the precision parameter of the Dirichlet process at the top of the hierarchy). The parameter vector for the model is $\boldsymbol{\psi} = (\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_D)$ and the hyperparameter is $h = (h_1, h_2)$. Unfortunately, the joint (prior) distribution of $\boldsymbol{\psi}$ is not available in closed form, and our efforts to obtain a formula for $\nu_h(\boldsymbol{\psi})/\nu_{h_*}(\boldsymbol{\psi})$ have been fruitless.

Regarding Condition C2, we note that Griffiths and Steyvers have developed a "collapsed Gibbs sampler" (CGS) which runs over the vector $\boldsymbol{z}$. The invariant distribution of the CGS is the conditional distribution of $\boldsymbol{z}$ given $\boldsymbol{w}$. The CGS cannot be used directly, because to apply (2.1) we need a Markov chain on the triple $(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{z})$, whose invariant distribution is $\nu_{h_*,\boldsymbol{w}}$. In Section 4 we obtain the conditional distribution of $(\boldsymbol{\beta}, \boldsymbol{\theta})$ given $\boldsymbol{z}$ and $\boldsymbol{w}$, and we show how to sample from this distribution. Therefore, given a Markov chain $\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(n)}$ generated via the CGS, we can form triples $(\boldsymbol{z}^{(1)}, \boldsymbol{\beta}^{(1)}, \boldsymbol{\theta}^{(1)}), \ldots, (\boldsymbol{z}^{(n)}, \boldsymbol{\beta}^{(n)}, \boldsymbol{\theta}^{(n)})$, and it is easy to see that this sequence forms a Markov chain with invariant distribution $\nu_{h_*,\boldsymbol{w}}$. We will refer to this Markov chain as the Augmented Collapsed Gibbs Sampler, and use the acronym ACGS. In Section 4 we show that the ACGS is not only geometrically ergodic, but actually is uniformly ergodic. We also show how to sample from the conditional distribution of $\boldsymbol{z}$ given $(\boldsymbol{\beta}, \boldsymbol{\theta})$ and $\boldsymbol{w}$. This enables us to construct a two-cycle Gibbs sampler which runs on the pair $(\boldsymbol{z}, (\boldsymbol{\beta}, \boldsymbol{\theta}))$. We will refer to this chain as the Grouped Gibbs Sampler, and use the acronym GGS. Either the ACGS or the GGS may be used, and we see that Condition C2 is satisfied for the LDA model.

The theorem below pertains to the LDA model and states that for this model, $\arg\max_h B_n(h)$ converges to $\arg\max_h m(h)$ almost surely, and that $\arg\max_h B_n(h)$ satisfies a CLT. The theorem also gives a procedure for constructing confidence sets for $\arg\max_h m(h)$. The result is explicit. Therefore, given a desired level of precision, we can determine the Markov chain length needed to estimate $\arg\max_h m(h)$ with that level of precision. The proof of the theorem is in Section A.2 of the Appendix. The theorem is valid under some natural and mild regularity conditions which

are given in the Appendix. (We have relegated the regularity conditions and a discussion of their significance to the Appendix in order to avoid making the present section too technical.) Let p be the dimension of h. So $p = 2$ if we take the distribution of the θ_d's to be a symmetric Dirichlet, and $p = K + 1$ if we allow this distribution to be an arbitrary Dirichlet.

Theorem 1 *Let $\psi_1, \psi_2, \ldots$ be generated according to the Augmented Collapsed Gibbs Sampling algorithm described above, let $B_n(h)$ be the estimate on the left side of* (2.1)*, and assume that Conditions A1–A6 in Section A.2 of the Appendix hold. Then:*

1. $\arg\max_h B_n(h) \xrightarrow{a.s.} \arg\max_h m(h)$.
2. $n^{1/2}\big(\arg\max_h B_n(h) - \arg\max_h m(h)\big) \xrightarrow{d} \mathcal{N}_p(0, \Sigma)$ *for some positive definite matrix* Σ.
3. *Let $\widehat{\Sigma}_n$ be the estimate of Σ obtained by the method of batching described in Section A.2 of the Appendix. Then $\widehat{\Sigma}_n \xrightarrow{a.s.} \Sigma$, and in particular $\widehat{\Sigma}_n$ is invertible for large n. Consequently, the ellipse $\mathcal{E}$ given by*

$$\mathcal{E} = \big\{h : (\arg\max_u B_n(u) - h)^\top \widehat{\Sigma}_n^{-1} (\arg\max_u B_n(u) - h) \le \chi^2_{p,.95}/n\big\}$$

is an asymptotic 95% confidence set for $\arg\max_h m(h)$. Here, $\chi^2_{p,.95}$ denotes the .95 quantile of the chi-square distribution with p degrees of freedom.

Remark 2 *The mathematical development in the proof requires a stipulation (Condition A5) which says that the distinguished point h_* is not quite arbitrary: if we specify $\mathcal{H} = [\eta^{(L)}, \eta^{(U)}] \times [\alpha_1^{(L)}, \alpha_1^{(U)}] \times \cdots \times [\alpha_K^{(L)}, \alpha_K^{(U)}]$, then h_* must satisfy*

$$\eta^* < 2\eta^{(L)} \quad \text{and} \quad \alpha_j^* < 2\alpha_j^{(L)},\ j = 1, \ldots, K. \tag{2.3}$$

(Condition (2.3) *is replaced by the obvious simpler analogue in the case of symmetric Dirichlets.) Thus, Condition A5 provides guidelines regarding the user-selected value of h_*.*

Remark 3 *Part 3 suggests that one should not use arbitrary Dirichlets when K is large. The ellipse is centered at $\arg\max_u B_n(u)$ and the lengths of its principal axes are governed by the term $\chi^2_{p,.95}$. When $p = K + 1$ and K is large, $\chi^2_{K+1,.95}$ is on the order of $K + 1$, and the confidence set for the empirical Bayes estimate of h is then huge. In other words, when we use an arbitrary Dirichlet, our estimates are very inaccurate.*

Actually, the problem that arises when $\dim(h) = K$ is not limited to our Monte Carlo scheme for estimating $\arg\max_h m(h)$. There is a fundamental problem, which has to do with the fact that there is not enough information in the corpus to estimate a high-dimensional h. Suppose we view the D documents as being drawn from some idealized population generated according to the LDA model indexed by h_0. Leaving aside computational issues, suppose we are able to calculate $\hat{h} = \arg\max_h m(h)$, the maximum likelihood estimate of h_0, to infinite accuracy. Standard asymptotics give $D^{1/2}(\hat{h} - h_0) \xrightarrow{d} \mathcal{N}_p(0, \Omega^{-1})$ as $D \to \infty$, where Ω is the Fisher information matrix. Therefore, a 95% confidence set for h_0 is given by the ellipse $\big\{h : (\hat{h} - h)^\top \Omega (\hat{h} - h) \le \chi^2_{K+1,.95}/D\big\}$, and we see that for high-dimensional h, D must be very large for us to be able to accurately estimate h_0.

2.2 Estimation of the Family of Posterior Expectations

Let g be a function of $\boldsymbol{\psi}$, and let $I(h) = \int g(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi})$ be the posterior expectation of $g(\boldsymbol{\psi})$ when the prior is ν_h. Suppose that we are interested in estimating $I(h)$ for all $h \in \mathcal{H}$. Proceeding as we did for estimation of the family of ratios $\{m(h)/m(h_*),\, h \in \mathcal{H}\}$, let $h_* \in \mathcal{H}$ be fixed but arbitrary, and let $\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots$ be an ergodic Markov chain with invariant distribution $\nu_{h*,\boldsymbol{w}}$. To estimate $\int g(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi})$, the obvious approach is to write

$$\int g(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi}) = \int g(\boldsymbol{\psi}) \frac{\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi})}{\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi})}\, d\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi}) \tag{2.4}$$

and then use the importance sampling estimate $(1/n)\sum_{i=1}^n g(\boldsymbol{\psi}_i)[\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi}_i)/\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi}_i)]$. This will not work because we do not know the normalizing constants for $\nu_{h,\boldsymbol{w}}$ and $\nu_{h_*,\boldsymbol{w}}$. This difficulty is handled by rewriting $\int g(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi})$, via (2.4), as

$$\begin{aligned}
\int g(\boldsymbol{\psi}) \frac{\ell_{\boldsymbol{w}}(\boldsymbol{\psi})\nu_h(\boldsymbol{\psi})/m(h)}{\ell_{\boldsymbol{w}}(\boldsymbol{\psi})\nu_{h_*}(\boldsymbol{\psi})/m(h_*)}\, d\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi}) &= \frac{m(h_*)}{m(h)} \int g(\boldsymbol{\psi}) \frac{\nu_h(\boldsymbol{\psi})}{\nu_{h_*}(\boldsymbol{\psi})}\, d\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi}) \\
&= \frac{\frac{m(h_*)}{m(h)} \int g(\boldsymbol{\psi}) \frac{\nu_h(\boldsymbol{\psi})}{\nu_{h_*}(\boldsymbol{\psi})}\, d\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi})}{\frac{m(h_*)}{m(h)} \int \frac{\nu_h(\boldsymbol{\psi})}{\nu_{h_*}(\boldsymbol{\psi})}\, d\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi})} \qquad (2.5a) \\
&= \frac{\int g(\boldsymbol{\psi}) \frac{\nu_h(\boldsymbol{\psi})}{\nu_{h_*}(\boldsymbol{\psi})}\, d\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi})}{\int \frac{\nu_h(\boldsymbol{\psi})}{\nu_{h_*}(\boldsymbol{\psi})}\, d\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi})}, \qquad (2.5b)
\end{aligned}$$

where in (2.5a) we have used the fact that the integral in the denominator is just 1, in order to cancel the unknown constant $m(h_*)/m(h)$ in (2.5b). The idea to express $\int g(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi})$ in this way was proposed in a different context by Hastings (1970). Expression (2.5b) is the ratio of two integrals with respect to $\nu_{h_*,\boldsymbol{w}}$, each of which may be estimated from the sequence $\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots, \boldsymbol{\psi}_n$. We may estimate the numerator and the denominator by

$$\frac{1}{n}\sum_{i=1}^n g(\boldsymbol{\psi}_i)[\nu_h(\boldsymbol{\psi}_i)/\nu_{h_*}(\boldsymbol{\psi}_i)] \quad \text{and} \quad \frac{1}{n}\sum_{i=1}^n [\nu_h(\boldsymbol{\psi}_i)/\nu_{h_*}(\boldsymbol{\psi}_i)]$$

respectively. Thus, if we let

$$w_i^{(h)} = \frac{\nu_h(\boldsymbol{\psi}_i)/\nu_{h_*}(\boldsymbol{\psi}_i)}{\sum_{e=1}^n [\nu_h(\boldsymbol{\psi}_e)/\nu_{h_*}(\boldsymbol{\psi}_e)]},$$

then these are weights, and we see that the desired integral may be estimated by the weighted average

$$\hat{I}(h) = \sum_{i=1}^n g(\boldsymbol{\psi}_i) w_i^{(h)}. \tag{2.6}$$

The significance of this development is that it shows that with a single Markov chain run, we can estimate the entire family of posterior expectations $\{I(h),\, h \in \mathcal{H}\}$. As was the case for the estimate on the left side of (2.1), the estimate (2.6) is remarkable in its simplicity. To compute it, we need to know only the ratio of the *priors*, and not the posteriors.

2.3 Serial Tempering

Unfortunately, (2.6) suffers a serious defect: unless h is close to h_*, ν_h can be nearly singular with respect to ν_{h_*} over the region where the $\boldsymbol{\psi}_i$'s are likely to be, resulting in a very unstable estimate. A similar remark applies to the estimate on the left side of (2.1). In other words, there is effectively a "radius" around h_* within which one can safely move. To state the problem more explicitly: there does not exist a single h_* for which the ratios $\nu_h(\boldsymbol{\psi})/\nu_{h_*}(\boldsymbol{\psi})$ have small variance simultaneously for all $h \in \mathcal{H}$. One way of dealing with this problem is to select J fixed points $h_1, \ldots, h_J \in \mathcal{H}$ that "cover" $\mathcal{H}$ in the sense that for every $h \in \mathcal{H}$, ν_h is "close to" at least one of $\nu_{h_1}, \ldots, \nu_{h_J}$. We then replace ν_{h_*} in the denominator by $(1/J)\sum_{j=1}^J b_j \nu_{h_j}$, for some suitable choice of positive constants $b_1, \ldots, b_J$. Operating intuitively, we say that for any $h \in \mathcal{H}$, because there exists at least one j for which ν_h is close to ν_{h_j}, the variance of $\nu_h(\boldsymbol{\psi})/[(1/J)\sum_{j=1}^J b_j \nu_{h_j}(\boldsymbol{\psi})]$ is small; hence the variance of $\nu_h(\boldsymbol{\psi})/[(1/J)\sum_{j=1}^J b_j \nu_{h_j}(\boldsymbol{\psi})]$ is small simultaneously for all $h \in \mathcal{H}$. Whereas for the estimates (2.1) and (2.6) we need a Markov chain with invariant distribution is $\nu_{h*,\boldsymbol{w}}$, in the present situation we need a Markov chain whose invariant distribution is the mixture $(1/J)\sum_{j=1}^J \nu_{h_j,\boldsymbol{w}}$. This approach may be implemented by a methodology called serial tempering (Marinari and Parisi (1992); Geyer and Thompson (1995)), originally developed for the purpose of improving mixing rates of certain Markov chains that are used to simulate physical systems in statistical mechanics. However, it can be used for a very different purpose, namely to increase the range of values over which importance sampling estimates have small variance. We now summarize this methodology, in the present context, and show how it can be used to produce estimates that are stable over a wide range of h values. Our explanations are detailed, because the material is not trivial and because we wish to deal with estimates of both marginal likelihood and posterior expectations. The reader who is not interested in the detailed explanations can skip the rest of this subsection with no loss regarding understanding the rest of the material in this paper, and simply regard serial tempering as a black box that produces estimates of the marginal likelihood (up to a constant) and of posterior expectations (cf. (2.1) and (2.6)) that are stable over a wide h-region.

To simplify the discussion, suppose that in line 2 of the LDA model we take $\boldsymbol{\alpha} = (\alpha, \ldots, \alpha)$, i.e. $\mathrm{Dir}_K(\boldsymbol{\alpha})$ is a symmetric Dirichlet, so that $\mathcal{H}$ is effectively two-dimensional, and suppose that we take $\mathcal{H}$ to be a bounded set of the form $\mathcal{H} = [\eta_L, \eta_U] \times [\alpha_L, \alpha_U]$. Our goal is to generate a Markov chain with invariant distribution $(1/J)\sum_{j=1}^J \nu_{h_j,\boldsymbol{w}}$. The updates will sample different components of this mixture, with jumps from one component to another. We now describe this carefully. Let Ψ denote the state space for $\boldsymbol{\psi}$. Recall that $\boldsymbol{\psi}$ has some continuous components and some discrete components. To proceed rigorously, we will take ν_h and $\nu_{h,\boldsymbol{w}}$ to all be densities with respect to a measure μ on Ψ. Define $\mathcal{L} = \{1, \ldots, J\}$, and for $j \in \mathcal{L}$, suppose that Φ_j is a Markov transition function on Ψ with invariant distribution equal to the posterior $\nu_{h_j,\boldsymbol{w}}$. On occasion we will write ν_j instead of ν_{h_j}. This notation is somewhat inconsistent, but we use it in order to avoid having double and triple subscripts. We have $\nu_{h,\boldsymbol{w}} = \ell_{\boldsymbol{w}}\, \nu_h/m(h)$ and $\nu_{h_j,\boldsymbol{w}} = \ell_{\boldsymbol{w}}\, \nu_j/m(h_j)$, $j = 1, \ldots, J$.

Serial tempering involves considering the state space $\mathcal{L} \times \Psi$, and forming the family of distributions $\{P_\zeta,\, \zeta \in \mathbb{R}^J\}$ on $\mathcal{L} \times \Psi$ with densities

$$p_\zeta(j, \boldsymbol{\psi}) \propto \ell_{\boldsymbol{w}}(\boldsymbol{\psi})\nu_j(\boldsymbol{\psi})/\zeta_j. \tag{2.7}$$

(To be pedantic, these are densities with respect to $\mu \times \sigma$, where σ is counting measure on $\mathcal{L}$.) The vector ζ is a tuning parameter, which we discuss shortly. For any value of ζ, by standard methods involving the Metropolis-Hastings algorithm, we can generate a Markov chain having invariant dis-

tribution equal to (2.7). If we take $\zeta_j = am(h_j)$ for $j = 1, \ldots, J$, where a is an arbitrary constant, then the ψ-marginal of p_ζ is exactly $(1/J)\sum_{j=1}^J \nu_{h_j,\boldsymbol{w}}$, so we can generate a Markov chain with the desired invariant distribution. Unfortunately, the values $m(h_1), \ldots, m(h_J)$ are unknown (our objective is precisely to estimate them). It will turn out that for any value of ζ, a Markov chain with invariant distribution (2.7) enables us to estimate the vector $(m(h_1), \ldots, m(h_j))$ up to a constant, and the closer ζ is to a constant multiple of $(m(h_1), \ldots, m(h_j))$, the better is our estimate. This gives rise to a natural iterative procedure for estimating $(m(h_1), \ldots, m(h_j))$. We now give the details.

Let $\Gamma(j, \cdot)$ be a Markov transition function on $\mathcal{L}$. In our context, we would typically take $\Gamma(j, \cdot)$ to be the uniform distribution on $\mathcal{N}_j$, where $\mathcal{N}_j$ is a set consisting of the indices of the h_l's which are close to h_j. Serial tempering is a Markov chain on $\mathcal{L} \times \Psi$ which can be viewed as a two-block Metropolis-Hastings (i.e. Metropolis-within-Gibbs) algorithm, and is run as follows. Suppose that the current state of the chain is $(L_{i-1}, \boldsymbol{\psi}_{i-1})$.

- A new value $j \sim \Gamma(L_{i-1}, \cdot)$ is proposed. We set $L_i = j$ with the Metropolis probability

$$\rho = \min\left\{1, \frac{\Gamma(j, L_{i-1})}{\Gamma(L_{i-1}, j)} \frac{\nu_j(\boldsymbol{\psi}_{i-1})/\zeta_j}{\nu_{L_{i-1}}(\boldsymbol{\psi}_{i-1})/\zeta_{L_{i-1}}}\right\}, \tag{2.8}$$

  and with the remaining probability we set $L_i = L_{i-1}$.

- Generate $\boldsymbol{\psi}_i \sim \Phi_{L_i}(\boldsymbol{\psi}_{i-1}, \cdot)$.

By standard arguments, the density (2.7) is an invariant density for the serial tempering chain. A key observation is that the ψ-marginal density of p_ζ is

$$f_\zeta(\boldsymbol{\psi}) = (1/c_\zeta)\sum_{j=1}^J \ell_{\boldsymbol{w}}(\boldsymbol{\psi})\nu_j(\boldsymbol{\psi})/\zeta_j, \qquad \text{where} \qquad c_\zeta = \sum_{j=1}^J m(h_j)/\zeta_j. \tag{2.9}$$

Suppose that $(L_1, \boldsymbol{\psi}_1), (L_2, \boldsymbol{\psi}_2), \ldots$ is a serial tempering chain. To estimate $m(h)$, consider

$$\widehat{M}_\zeta(h) = \frac{1}{n}\sum_{i=1}^n \frac{\nu_h(\boldsymbol{\psi}_i)}{(1/J)\sum_{j=1}^J \nu_j(\boldsymbol{\psi}_i)/\zeta_j}. \tag{2.10}$$

Note that this estimate depends only on the $\boldsymbol{\psi}$-part of the chain. Assuming that we have established that the chain is ergodic, we have

$$\begin{aligned}\widehat{M}_\zeta(h) &\xrightarrow{\text{a.s.}} \int \frac{\nu_h(\boldsymbol{\psi})}{(1/J)\sum_{j=1}^J \nu_j(\boldsymbol{\psi})/\zeta_j} \frac{\sum_{j=1}^J \ell_{\boldsymbol{w}}(\boldsymbol{\psi})\nu_j(\boldsymbol{\psi})/\zeta_j}{c_\zeta}\, d\mu(\boldsymbol{\psi}) \\ &= \int \frac{\ell_{\boldsymbol{w}}(\boldsymbol{\psi})\nu_h(\boldsymbol{\psi})}{c_\zeta/J}\, d\mu(\boldsymbol{\psi}) \\ &= \frac{m(h)}{c_\zeta/J}.\end{aligned} \tag{2.11}$$

This means that for any ζ, the family $\{\widehat{M}_\zeta(h),\, h \in \mathcal{H}\}$ can be used to estimate the family $\{m(h),\, h \in \mathcal{H}\}$, up to a single multiplicative constant.

To estimate the family of integrals $\{\int g(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi}),\, h \in \mathcal{H}\}$, we proceed as follows. Let

$$\widehat{U}_\zeta(h) = \frac{1}{n}\sum_{i=1}^n \frac{g(\boldsymbol{\psi}_i)\nu_h(\boldsymbol{\psi}_i)}{(1/J)\sum_{j=1}^J \nu_j(\boldsymbol{\psi}_i)/\zeta_j}. \tag{2.12}$$

By ergodicity we have

$$\begin{aligned}\widehat{U}_\zeta(h) &\xrightarrow{\text{a.s.}} \int \frac{g(\boldsymbol{\psi})\nu_h(\boldsymbol{\psi})}{(1/J)\sum_{j=1}^J \nu_j(\boldsymbol{\psi})/\zeta_j} \frac{\sum_{j=1}^J \ell_{\boldsymbol{w}}(\boldsymbol{\psi})\nu_j(\boldsymbol{\psi})/\zeta_j}{c_\zeta}\, d\mu(\boldsymbol{\psi}) \\ &= \int \frac{\ell_{\boldsymbol{w}}(\boldsymbol{\psi})g(\boldsymbol{\psi})\nu_h(\boldsymbol{\psi})}{c_\zeta/J}\, d\mu(\boldsymbol{\psi}) \\ &= \frac{m(h)}{c_\zeta/J}\int g(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi}).\end{aligned} \tag{2.13}$$

Combining the convergence statements (2.13) and (2.11), we see that

$$\hat{I}^{\text{st}}_\zeta(h) \coloneqq \frac{\widehat{U}_\zeta(h)}{\widehat{M}_\zeta(h)} \xrightarrow{\text{a.s.}} \int g(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi}). \tag{2.14}$$

Suppose that for some constant a, we have

$$(\zeta_1, \ldots, \zeta_J) = a(m(h_1), \ldots, m(h_J)). \tag{2.15}$$

Then $c_\zeta = J/a$, and as noted earlier, $f_\zeta(\boldsymbol{\psi}) = (1/J)\sum_{j=1}^J \nu_{h_j,\boldsymbol{w}}(\boldsymbol{\psi})$, i.e. the ψ-marginal of p_ζ (see (2.9)) gives equal weight to each of the component distributions in the mixture. (Expressing this slightly differently, if (2.15) is true, then the invariant density (2.7) becomes $p_\zeta(j, \boldsymbol{\psi}) = (1/J)\nu_{h_j,\boldsymbol{w}}(\boldsymbol{\psi})$, so the L-marginal distribution of p_ζ gives mass $(1/J)$ to each point in $\mathcal{L}$.) Therefore, for large n, the proportions of time spent in the J components of the mixture are about the same, a feature which is essential if serial tempering is to work well. In practice, we cannot arrange for (2.15) to be true, because $m(h_1), \ldots, m(h_J)$ are unknown. However, the vector $(m(h_1), \ldots, m(h_J))$ may be estimated (up to a multiplicative constant) iteratively as follows. If the current value is $\zeta^{(t)}$, then set

$$\big(\zeta_1^{(t+1)}, \ldots, \zeta_J^{(t+1)}\big) = \big(\widehat{M}_{\zeta^{(t)}}(h_1), \ldots, \widehat{M}_{\zeta^{(t)}}(h_J)\big). \tag{2.16}$$

From the convergence result (2.11), we get $\widehat{M}_{\zeta^{(t)}}(h_j) \xrightarrow{\text{a.s.}} m(h_j)/a_{\zeta^{(t)}}$, where $a_{\zeta^{(t)}}$ is a constant, i.e. (2.15) is nearly satisfied by $\big(\zeta_1^{(t+1)}, \ldots, \zeta_J^{(t+1)}\big)$. To determine the number of iterations needed, at each iteration we record the proportions of time spent in the J different components of the mixture, i.e. the vector $\big((1/n)\sum_{i=1}^n I(L_i = 1), \ldots, (1/n)\sum_{i=1}^n I(L_i = J)\big)$, and we stop the iteration when this vector is nearly uniform. In all our examples, three or four iterations were sufficient. Pseudocode is given in Algorithm 1.

To sum up, we estimate the family of marginal likelihoods (up to a constant) and the family of posterior expectations as follows. First, we obtain the vector of tuning parameters ζ via the iterative scheme given by (2.16). To estimate the family of marginal likelihoods (up to a constant) we use

Algorithm 1: Serial tempering. See the discussion in Sec. 2.3 and Appendices A.1 and 4.

Data: Observed words $\boldsymbol{w}$
Result: A Markov chain on $\mathcal{L} \times \Psi$

specify $h_1, \ldots, h_J \in \mathcal{H}$;
initialize $\zeta_1^{(1)}, \ldots, \zeta_J^{(1)}$;
initialize $\boldsymbol{\psi}_0 = (\boldsymbol{\beta}^{(0)}, \boldsymbol{\theta}^{(0)}, \boldsymbol{z}^{(0)})$, L_0;
compute count statistics n_{dk} and m_{dkv}, for $d = 1, \ldots, D$, $k = 1, \ldots, K$, $v = 1, \ldots, V$;
for *tuning iteration* $t = 1, \ldots$ **do**
  for *MCMC iteration* $i = 1, \ldots$ **do**
    `// The Metropolis-Hastings update`
    `// Set` L_i `via the probability` ρ `given by` (2.8)
    propose index $j \sim \Gamma(L_{i-1}, \cdot)$;
    sample $U \sim \text{Uniform}(0, 1)$;
    if $U < \rho$ **then**
      set $L_i = j$;
    else
      set $L_i = L_{i-1}$;
    `// Generate` $\boldsymbol{\psi}_i = (\boldsymbol{\beta}^{(i)}, \boldsymbol{\theta}^{(i)}, \boldsymbol{z}^{(i)}) \sim \Phi_{L_i}(\boldsymbol{\psi}_{i-1}, \cdot)$
    for *document* $d = 1, \ldots, D$ **do**
      for *word* w_{dr}, $r = 1, \ldots, n_d$ **do**
        sample topic index $z_{dr}^{(i)}$ via the CGS (Griffiths and Steyvers, 2004);
        update count statistics n_{dk} and m_{dkv} according to $z_{dr}^{(i)}$ and w_{dr};
    for *topic* $k = 1, \ldots, K$ **do**
      sample topic $\beta_k^{(i)}$ via (4.5);
    for *document* $d = 1, \ldots, D$ **do**
      sample the distribution on topics $\theta_d^{(i)}$ via (4.5);
  `// Update tuning parameters` $\zeta_1, \ldots, \zeta_J$
  compute the estimates $\widehat{M}_{\zeta^{(t)}}(h_1), \ldots, \widehat{M}_{\zeta^{(t)}}(h_J)$ via (2.10) using $\boldsymbol{\psi}_i$ and $\zeta_1^{(t)}, \ldots, \zeta_J^{(t)}$;
  set $\big(\zeta_1^{(t+1)}, \ldots, \zeta_J^{(t+1)}\big) = \big(\widehat{M}_{\zeta^{(t)}}(h_1), \ldots, \widehat{M}_{\zeta^{(t)}}(h_J)\big)$;

$\widehat{M}_\zeta(h)$ defined in (2.10), and to estimate the family of posterior expectations we use $\hat{I}_\zeta^{\text{st}}(h) = \widehat{U}_\zeta(h)/\widehat{M}_\zeta(h)$ (see (2.12) and (2.10)).

We point out that it is possible to estimate the family of marginal likelihoods (up to a constant) by

$$\widetilde{M}_\zeta(h) = \frac{1}{n}\sum_{i=1}^{n} \frac{\nu_h(\boldsymbol{\psi}_i)}{\nu_{L_i}(\boldsymbol{\psi}_i)/\zeta_{L_i}}. \tag{2.17}$$

Note that $\widetilde{M}_\zeta(h)$ uses the sequence of pairs $(L_1, \boldsymbol{\psi}_1), (L_2, \boldsymbol{\psi}_2), \ldots$, and not just the sequence $\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots$. To see why (2.17) is a valid estimator, observe that by ergodicity we have

$$\begin{aligned}
\widetilde{M}_\zeta(h) &\xrightarrow{\text{a.s.}} \iint \frac{\nu_h(\boldsymbol{\psi})}{\nu_L(\boldsymbol{\psi})/\zeta_L} \cdot \left[\frac{1}{c_\zeta}\ell_{\boldsymbol{w}}(\boldsymbol{\psi})\nu_L(\boldsymbol{\psi})/\zeta_L\right] d\mu(\boldsymbol{\psi})\, d\sigma(L) \\
&= \iint \frac{m(h)}{c_\zeta}\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi})\, d\mu(\boldsymbol{\psi})\, d\sigma(L) \\
&= J\frac{m(h)}{c_\zeta}.
\end{aligned} \tag{2.18}$$

(Note that the limit in (2.18) is the same as the limit in (2.11).) Similarly, we may estimate the integral $\int g(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi})$ by the ratio

$$\tilde{I}_\zeta^{\text{st}}(h) = \sum_{i=1}^{n} \frac{g(\boldsymbol{\psi}_i)\nu_h(\boldsymbol{\psi}_i)}{\nu_{L_i}(\boldsymbol{\psi}_i)/\zeta_{L_i}} \Big/ \sum_{i=1}^{n} \frac{\nu_h(\boldsymbol{\psi}_i)}{\nu_{L_i}(\boldsymbol{\psi}_i)/\zeta_{L_i}}.$$

The estimate $\tilde{I}_\zeta^{\text{st}}(h)$ is also based on the pairs $(L_1, \boldsymbol{\psi}_1), (L_2, \boldsymbol{\psi}_2), \ldots$, and it is easy to show that $\tilde{I}_\zeta^{\text{st}}(h) \xrightarrow{\text{a.s.}} \int g(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi})$.

The estimates $\widetilde{M}_\zeta(h)$ and $\tilde{I}_\zeta^{\text{st}}(h)$ are the ones that are used by Marinari and Parisi (1992) and Geyer and Thompson (1995), but $\widehat{M}_\zeta(h)$ and $\hat{I}_\zeta^{\text{st}}(h)$ appear to significantly outperform $\widetilde{M}_\zeta(h)$ and $\tilde{I}_\zeta^{\text{st}}(h)$ in terms of accuracy. We demonstrate this in Section 2.4.

Remark 4 *Theorem 1 continues to be true when we use the serial tempering chain, as opposed to the simple ACGS. The needed changes are that in the statement of the theorem B_n is replaced with $\widetilde{M}_\zeta$, and Condition A5 is replaced by the following. If $h^{(1)}, \ldots, h^{(J)}$ are the grid points used in running the serial tempering chain, then the stipulation on h_* given by* (2.3) *is satisfied by $h^{(j)}$ for at least one index j. See the proof of Theorem 1 in the Appendix.*

Globally-Valid Confidence Bands for $\{I(h),\, h \in \mathcal{H}\}$ Based on Serial Tempering Here we explain how to form confidence bands for the family $\{I(h),\, h \in \mathcal{H}\}$ based on $\{\hat{I}_\zeta^{\text{st}}(h),\, h \in \mathcal{H}\}$. Our arguments are informal, and we focus primarily on the algorithm for constructing the bands. The proof that the method works is given in Section A.3 of the Appendix. We will write $\hat{I}$ instead of $\hat{I}_\zeta^{\text{st}}$ to lighten the notation. Suppose that $\sup_{h\in\mathcal{H}} n^{1/2}|\hat{I}(h) - I(h)|$ has a limiting distribution as $n \to \infty$ (in the Appendix we explain why such a result is true), and suppose that we know the .95 quantile of this distribution, i.e. we know the value $c_{.95}$ such that

$$P\Big(\sup_{h\in\mathcal{H}} n^{1/2}|\hat{I}(h) - I(h)| \le c_{.95}\Big) = .95. \tag{2.19}$$

In this case we may rewrite (2.19) as

$$P\Big(\hat{I}(h) - c_{.95}/n^{1/2} \le I(h) \le \hat{I}(h) + c_{.95}/n^{1/2} \text{ for all } h \in \mathcal{H}\Big) = .95,$$

meaning that the band $\hat{I}(h) \pm c_{.95}/n^{1/2}$ is a globally-valid confidence band for $\{I(h),\, h \in \mathcal{H}\}$. (In contrast, for a pointwise band $(L(h), U(h))$, $h \in \mathcal{H}$, we can only make the statement $P\big(L(h) \le$

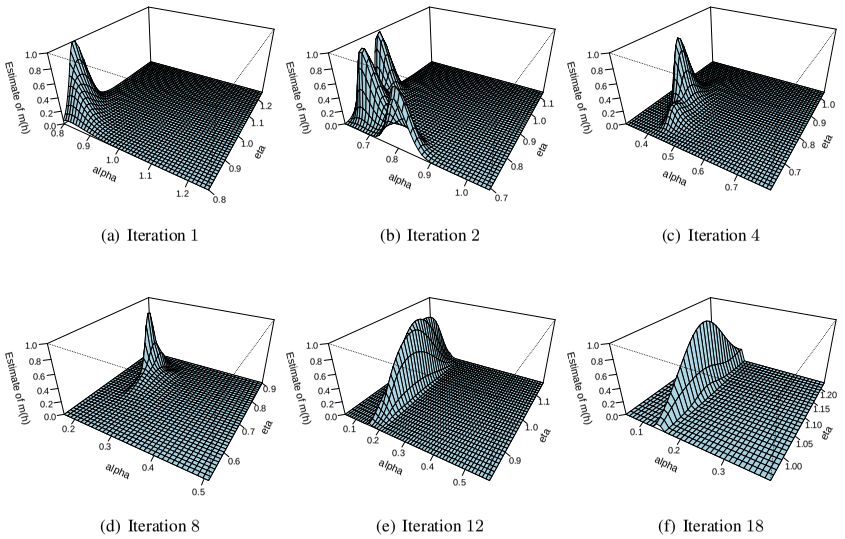
$I(h) \le U(h)\big) = .95$ for each $h \in \mathcal{H}$, and we cannot make any statement regarding simultaneous coverage.)

The difficulty is in obtaining $c_{.95}$, and we now show how this quantity can be estimated through the method of batching, which is described as follows. The sequence $\psi_1, \ldots, \psi_n$ is broken up into J consecutive pieces of equal lengths called batches. For $j = 1, \ldots, J$, let $\hat{I}_j(h)$ be the estimate of $I(h)$ produced by batch j. Now the $\hat{I}_j(h)$'s are each formed from a sample of size n/J. Informally, if n is large and n/J is also large, then for $j = 1, \ldots, J$, $\sup_{h\in\mathcal{H}}(n/J)^{1/2}|\hat{I}_j(h) - I(h)|$ and $\sup_{h\in\mathcal{H}} n^{1/2}|\hat{I}(h) - I(h)|$ have approximately the same distribution. Therefore, to estimate $c_{.95}$, we let $S_j = \sup_{h\in\mathcal{H}}(n/J)^{1/2}|\hat{I}_j(h) - I(h)|$, and as our estimate of $c_{.95}$ we use the 95^{th} percentile of the sequence $S_1, \ldots, S_J$. Unfortunately, the S_j's are not available, because they involve $I(h)$, which is unknown. So instead we use $\mathcal{S}_j = \sup_{h\in\mathcal{H}}(n/J)^{1/2}|\hat{I}_j(h) - \hat{I}(h)|$, in which we have substituted $\hat{I}(h)$ for $I(h)$. To conclude, let $\mathcal{S}_{[1]} \le \mathcal{S}_{[2]} \le \cdots \le \mathcal{S}_{[J]}$ denote the ordered values of the sequence $\mathcal{S}_1, \ldots, \mathcal{S}_J$. We estimate $c_{.95}$ via $\mathcal{S}_{[.95J]}$, and our 95% globally-valid confidence band for $\{I(h),\, h \in \mathcal{H}\}$ is $\big\{\hat{I}(h) \pm \mathcal{S}_{[.95J]}/n^{1/2},\, h \in \mathcal{H}\big\}$. In the Appendix we show that the probability that the entire function $\{I(h),\, h \in \mathcal{H}\}$ lies inside the band converges to .95 as $n \to \infty$. There are conditions on J: we need that $J \to \infty$ and $n/J \to \infty$; a good choice is $J = n^{1/2}$. The Markov chain length n should be chosen such that the band is acceptably narrow.

Iterative Scheme for Choosing the Grid The performance of serial tempering depends crucially on the choice of grid points $h_1, \ldots, h_J$, and it is essential that $\arg\max_h m(h)$ be close to at least one of the grid points, for the reason discussed at the beginning of this section. This creates a circular problem: the ideal grid is one that is centered or nearly centered at $\arg\max_h m(h)$, but $\arg\max_h m(h)$ is unknown. The problem is compounded by the fact that the grid has to be "tight," i.e. the points $h_1, \ldots, h_J$ need to be close together. This is because when the corpus is large, if h_j and $h_{j'}$ are not close, then for $j \ne j'$, ν_{h_j} and $\nu_{h_{j'}}$ are nearly singular (each is a product of a large number of terms—see (2.2)). In the serial tempering chain, this near singularity causes the proposal $j \sim \Gamma(L_{i-1}, \cdot)$ (see (2.8)) to have high probability of being rejected, and the chain does not mix well. To deal with this problem, we use an iterative scheme which proceeds as follows. We initialize the experiment with a fixed $h^{(0)}$ (for example $h^{(0)} = (1, 1)$) and a subgrid that "covers" $h^{(0)}$ (for example a subgrid with convex hull equal to $[1/2, 2] \times [1/2, 2]$). We then subsample a small set of documents from the corpus and run the serial tempering chain to find the estimate of the maximizer of the marginal likelihood for the subsampled corpus, using the current grid setting. We iterate: at iteration t, we set $h^{(t)}$ to be the estimate of the maximizer obtained from the previous iteration, and select a subgrid that covers $h^{(t)}$. As the iteration number t increases, the grid is made more narrow, and the number of subsampled documents is increased. This scheme works because in the early iterations the number of documents is small, so the near-singularity problem does not arise, and we can use a wide grid. In our experience, decreasing the dimensions of the α- and η-grids by 10% and increasing the number of subsampled documents by 10% at each iteration works well. It is very interesting to note that convergence may occur before the subsample size is equal to the number of documents in the corpus, in which case there is no need to ever deal with the entire corpus, and in fact this is typically what happens, unless the corpus is small. (By "convergence" we mean that $h^{(t)}$ is nearly the same as the values from the previous iterations.) Of course, for small corpora the near-singularity problem does not arise, and the iterative scheme can be skipped entirely.

To illustrate the scheme, we generated a corpus according to the LDA model with $D = 10^5$, $K = 50$, $V = 500$, $n_d = 80$ for all d, and $h_{\text{true}} = (\eta, \alpha) = (.8, .2)$, and ran the scheme using

Markov chains of length $n = 50{,}000$ and grids of size $J = 100$. As will be clear shortly, our results would have been identical if D had been *any* number bigger than 10^5. Figure 2 shows the marginal likelihood surfaces as the iterations progress. At iteration 1, the α-value of the maximizer is outside the convex hull of the grid, and at the second iteration, the grid is centered at that point. Figure 3 gives precise information on the number of subsampled documents (left panel), and the lower and upper endpoints of the α- and η-values used in the grids, as the iterations progress (right panel). The right panel also gives α- and η-values of the estimate of the argmax as the iterations progress. As can be seen from Figure 3, the scheme has effectively converged after about 18 iterations, and at convergence the number of subsampled documents is only 200.



(a) Iteration 1 (b) Iteration 2 (c) Iteration 4

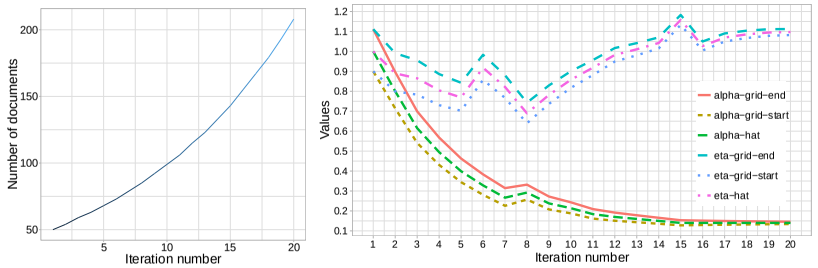
(d) Iteration 8 (e) Iteration 12 (f) Iteration 18

Figure 2: Values of $\widehat{M}(h)$ for iterations 1, 2, 4, 8, 12, 18 using a synthetic corpus generated according to the LDA model with $K = 20$, $n_d = 100$ for each d, $V = 100$, and $h_{\text{true}} = (.8, .2)$.

Serial tempering is a method for enhancing the simple estimator (2.1) which works well when $\dim(h)$ is low. The method does not scale well when $\dim(h)$ increases. In Section 6 we discuss this issue and present an idea on a different way to enhance (2.1) when h is high dimensional.

2.4 Illustration on a Wikipedia Corpus

In Section 1 we mentioned that the hyperparameter h has a strong effect on the prior distribution of the parameters in the model. Here we show empirically that it has a strong impact on the posterior distribution, and hence on inference based on this posterior distribution. To this end, we considered a corpus of articles from Wikipedia, constructed as follows. When a Wikipedia article is created, it



(a) Number of documents (b) Grid configurations and α- and η-values of estimate of the argmax

Figure 3: Iterations in the serial tempering scheme used on the synthetic corpus in Figure 2: left panel gives the number of documents subsampled at each iteration; right panel gives the specifications for the grid at each iteration.

is typically tagged to one or more categories, one of which is the "primary category." The corpus consists of 8 documents from the category *Leopardus*, 8 from the category *Lynx*, and 7 from *Prionailurus*. There are 303 words in the vocabulary, and the total number of words in the corpus is 7788. We took $K = 3$, so implicitly we envisage a topic being induced by each of the three categories. The corpus is quite small, but it is challenging to analyze because the topics are very close to each other, so in the posterior distribution there is a great deal of uncertainty regarding the latent topic indicator variables, and this is why we chose this data set. (In our analysis of this corpus, we treat the articles as unlabeled, i.e. we act as if for each article we don't know the category from which the article is taken.) As mentioned in Section 1, two quantities of interest are the posterior probability that the topic indicator variables for documents i and j are close, i.e. $\nu_{h,\boldsymbol{w}}(\|\theta_i - \theta_j\| \leq \epsilon)$, and the posterior expectation of the distance between topics i and j, which is given by the integral $\int \|\beta_i - \beta_j\| \, d\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi})$. Figure 4 gives plots of estimates of these posterior probabilities and expectations, as h varies, together with 95% globally-valid confidence sets. The plots clearly show that these posterior probabilities and expectations vary considerably with h.

Each plot was constructed from a serial tempering chain, using the methodology described in Section 2.3. Details regarding the chain and the plots are as follows. We took the sequence $h_1, \ldots, h_J$ to consist of an 11×20 grid of 220 evenly-spaced values over the region $(\eta, \alpha) \in [.6, 1.1] \times [.15, 1.1]$. For each hyperparameter value h_j ($j = 1, \ldots, 220$), we took Φ_j to be the Markov transition function of the Augmented Collapsed Gibbs Sampler alluded to earlier and described in detail in Section 4 (in all our experiments we used the Augmented Collapsed Gibbs Sampler, but the Grouped Gibbs Sampler gives results which are very similar). We took the Markov transition function $K(j, \cdot)$ on $\mathcal{L} = \{1, \ldots, 220\}$ to be the uniform distribution on $\mathcal{N}_j$ where $\mathcal{N}_j$ is the subset of $\mathcal{L}$ consisting of the indices of the h_l's that are neighbors of the point h_j. (An interior point has eight neighbors, an edge point has five, and a corner point has three.)[1]

1. Software for implementation of our algorithms as well as datasets we use are available as an R package at `https://github.com/clintpgeorge/ldamcmc`

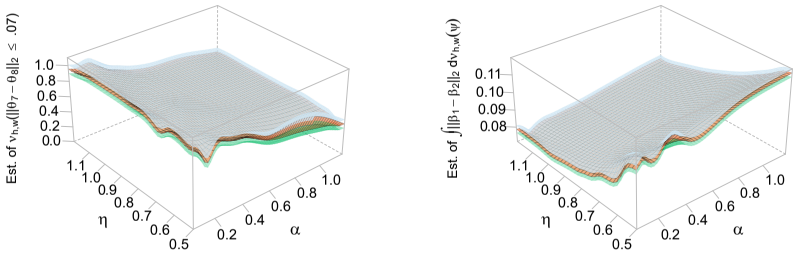
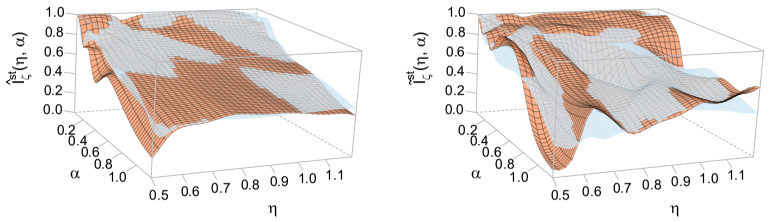


Figure 4: Variability of posterior probabilities and expectations for the Cats corpus from Wikipedia. Left panel: estimate of the posterior probability that documents 7 and 8 have essentially the same topics, in the sense that $\|\theta_7 - \theta_8\| \leq .07$, as h varies. Right panel: estimate of the posterior expectation of the (Euclidean, i.e. L_2) distance between topics 1 and 2 as h varies.

In Section 2.3, we stated that $\widehat{M}_\zeta(h)$ and $\hat{I}^{\text{st}}_\zeta(h)$ appear to significantly outperform $\widetilde{M}_\zeta(h)$ and $\tilde{I}^{\text{st}}_\zeta(h)$ in terms of accuracy. We now provide some evidence for this, and we will deal with the estimates of $I(h)$ (a comparison of $\widehat{M}_\zeta(h)$ and $\widetilde{M}_\zeta(h)$ is given in George (2015)). We considered the Wikipedia Cats corpus described above, and we took $I(h) = \nu_{h,\boldsymbol{w}}(\|\theta_7 - \theta_8\| \leq .07)$. We calculated $\hat{I}^{\text{st}}_\zeta(h)$ twice, using two different seeds, and also calculated $\tilde{I}^{\text{st}}_\zeta(h)$ twice, using two different seeds, in every case using the same h-range that was used in Figure 4. The four surfaces were constructed via four independent serial tempering experiments, each involving two iterations (each of length 50,000 after a short burn-in period) to form the tuning parameter ζ, which was given initial value $\zeta^{(0)} = (\zeta_1^{(0)}, \ldots, \zeta_{220}^{(0)}) = (1, \ldots, 1)$, and one final iteration (of length 100,000) to form the estimate of $I(h)$. Figure 5(a) shows the two estimates $\hat{I}^{\text{st}}_\zeta(h)$, and Figure 5(b) shows the two estimates $\tilde{I}^{\text{st}}_\zeta(h)$. The figures show that the two independent estimates $\hat{I}^{\text{st}}_\zeta(h)$ are close to each other, whereas the two independent estimates $\tilde{I}^{\text{st}}_\zeta(h)$ are not.

Although the variability of $\hat{I}^{\text{st}}_\zeta(h)$ is significantly smaller than that of $\tilde{I}^{\text{st}}_\zeta(h)$, the figures perhaps don't show this very clearly because a visual comparison of two surfaces is not easy. Therefore, we extracted two one-dimensional slices from each panel in Figure 5, which we used to create Figure 6. The figure shows the values of the two versions of $\hat{I}^{\text{st}}_\zeta(\eta, \alpha)$ and the two versions of $\tilde{I}^{\text{st}}_\zeta(\eta, \alpha)$ when η is fixed at .70 (two left panels); and it shows these plots when η is fixed at 1.00 (two right panels). The superiority of $\hat{I}^{\text{st}}_\zeta$ over $\tilde{I}^{\text{st}}_\zeta$ is striking. We mention that, ostensibly, $\widehat{M}_\zeta(h)$ and $\hat{I}^{\text{st}}_\zeta(h)$ require more computation, but the quantities $(1/J) \sum_{j=1}^{J} \nu_j(\boldsymbol{\psi}_i)/\zeta_j$, $i = 1, \ldots, n$ are calculated once, and stored. Doing this essentially eliminates the increased computing cost.



(a) $\hat{I}^{\text{st}}_\zeta(\eta, \alpha)$ (b) $\tilde{I}^{\text{st}}_\zeta(\eta, \alpha)$

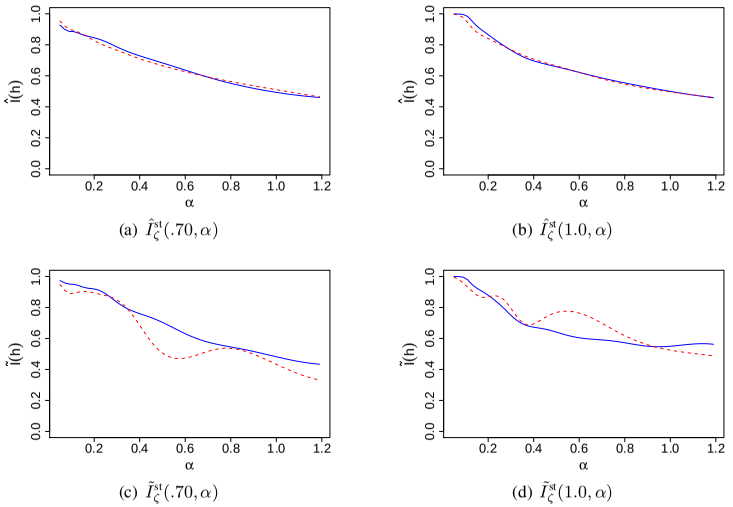
Figure 5: Comparison of the variability of $\hat{I}^{\text{st}}_\zeta$ and $\tilde{I}^{\text{st}}_\zeta$. Left panel shows two independent estimates of $I(\eta, \alpha) = \nu_{h,\boldsymbol{w}}(\|\theta_7 - \theta_8\| \leq .07)$ using $\hat{I}^{\text{st}}_\zeta(\eta, \alpha)$. Right panel uses $\tilde{I}^{\text{st}}_\zeta$ instead of $\hat{I}^{\text{st}}_\zeta$.

3. Empirical Assessment of the Estimator of the Argmax

Consider the LDA model with a given hyperparameter value, which we will denote by h_{true}, and suppose we carry out steps 1–4 of the model, where in the final step we generate the corpus $\boldsymbol{w}$. The maximum likelihood estimate of h is $\hat{h} = \arg\max_h m(h)$ and, as we mentioned earlier, for any constant a, known or unknown, $\arg\max_h m(h) = \arg\max_h am(h)$. As noted earlier, the family $\{\widehat{M}_\zeta(h),\, h \in \mathcal{H}\}$, where $\widehat{M}_\zeta(h)$ is given by (2.10), may be used to estimate the family $\{m(h),\, h \in \mathcal{H}\}$ up to a multiplicative constant. So we may use $\arg\max_h \widehat{M}_\zeta(h)$ to estimate $\hat{h}$.

Recall that $B_n(h)$ is the estimate of $m(h)/m(h_*)$ given by the left side of equation (2.1). In theory, $\arg\max_h B_n(h)$ can also be used. However, as we pointed out earlier, $B_n(h)$ is stable only for h close to h_*—a similar remark applies to $\hat{I}(h)$—and unless the region of hyperparameter values of interest is small, we would not use $B_n(h)$ and $\hat{I}(h)$, and we would use estimates based on serial tempering instead. We have included the derivations of $B_n(h)$ and $\hat{I}(h)$ primarily for motivation, as these makes it easier to understand the development of the serial tempering estimates. In Section 2.4 we presented an experiment which strongly suggested that $\hat{I}^{\text{st}}_\zeta(h)$ is significantly better than $\tilde{I}^{\text{st}}_\zeta(h)$ in terms of variance. George (2015) gives experimental evidence that, analogously, $\widehat{M}_\zeta(h)$ is significantly better than $\widetilde{M}_\zeta(h)$. Therefore, for the rest of this paper, we use only $\widehat{M}_\zeta(h)$ and $\hat{I}^{\text{st}}_\zeta(h)$.

Here we present the results of some experiments which demonstrate good performance of $\hat{\hat{h}} := \arg\max_h \widehat{M}_\zeta(h)$ as an estimate of h_{true}. We took $\boldsymbol{\alpha} = (\alpha, \ldots, \alpha)$, i.e. $\text{Dir}_K(\boldsymbol{\alpha})$ is a symmetric Dirichlet, so that the hyperparameter in the model reduces to $h = (\eta, \alpha) \in (0, \infty)^2$. Our experiment is set up as follows: the vocabulary size is $V = 40$, the number of documents is $D = 400$, the document lengths are $n_d = 80$, $d = 1, \ldots, D$, and the number of topics is $K = 8$. We used four settings for the hyperparameter under which we generate the model: h_{true} is taken to be $(.25, .25)$, $(.25, 4)$, $(4, .25)$, and $(4, 4)$. We estimated the marginal likelihood surfaces (up to a constant) on an



(a) $\hat{I}^{\text{st}}_\zeta(.70, \alpha)$ (b) $\hat{I}^{\text{st}}_\zeta(1.0, \alpha)$

(c) $\tilde{I}^{\text{st}}_\zeta(.70, \alpha)$ (d) $\tilde{I}^{\text{st}}_\zeta(1.0, \alpha)$

Figure 6: Two one-dimensional views of the plots in Figure 5. Each of the top two panels shows two independent estimates of $I(\eta, \alpha)$, using $\hat{I}^{\text{st}}_\zeta(\eta, \alpha)$. For the left panel, $\eta = 0.70$, and for the right panel, $\eta = 1.00$. The bottom two panels use $\tilde{I}^{\text{st}}_\zeta$ instead of $\hat{I}^{\text{st}}_\zeta$. The plots show that the variability of $\hat{I}^{\text{st}}_\zeta$ is much smaller than that of $\tilde{I}^{\text{st}}_\zeta$.

evenly-spaced 50×50 grid of 2500 values using $\widehat{M}_\zeta(h)$ calculated from a serial tempering chain implemented as follows. The size of the subgrid was taken to be $11 \times 11 = 121$, and we used ten iterations of the iterative scheme described in Section 2.3 to form the final subgrid. The subgrid for each of the four corpora is shown in the first section of the supplementary document George and Doss. For each hyperparameter value h_j ($j = 1, \ldots, 121$), we took Φ_j to be the Markov transition function of the Augmented Collapsed Gibbs sampler. We took the Markov transition function $K(j, \cdot)$ on $\mathcal{L} = \{1, \ldots, 121\}$ to be the uniform distribution on $\mathcal{N}_j$ where $\mathcal{N}_j$ is the subset of $\mathcal{L}$ consisting of the indices of the h_l's that are neighbors of the point h_j. We obtained the value ζ^{final} via three iterations of the scheme given by (2.16), in which we ran the serial tempering chain in each tuning iteration for 100,000 iterations after a short burn-in period, and we initialized $\zeta^{(0)} = (\zeta^{(0)}_1, \ldots, \zeta^{(0)}_{121}) = (1, \ldots, 1)$. Using ζ^{final}, we ran the final serial tempering chain for the same number of iterations as in the tuning stage.

Figure 7 gives plots of the estimates $\widehat{M}_\zeta(h)$ and also of their Monte Carlo standard errors (MCSE) for the four specifications of h_{true}. We computed these standard error estimates using the method of batch means, which is implemented in the R package `mcmcse` (Flegal et al., 2016); they are valid pointwise, as opposed to globally, over the h-region of interest. They indicate that the accuracy of $\widehat{M}_\zeta(\cdot)$ is adequate over the entire h-range for each of the four cases of h_{true}. (We

produced error margins that are valid locally, as opposed to globally, because it is of interest to see the regions where the variability is high.) In the supplementary document George and Doss we show plots of the occupancy times for the 121 components of the mixture distribution. For each of the four values of h_{true}, these occupancy times are close to uniform, indicating adequate mixing. We note that $\arg\max_h \widetilde{M}_\zeta(h)$ can be obtained through a grid search from the plots in Figure 7, which is what we did in this particular illustration, but in practice these plots don't need to be generated, and $\arg\max_h \widehat{M}_\zeta(h)$ can be found very quickly through standard optimization algorithms such as those that work through gradient-based approaches (which are very easy to implement here, since $\dim(h)$ is only 2). These algorithms take very little time because they require calculation of $\widehat{M}_\zeta(\cdot)$ for only a few values of h. For the case where $\dim(h)$ is large, we mention in particular Bergstra and Bengio (2012), who argue that random search is more efficient than grid search when only a few components of h matter. As can be seen from the figure, $\arg\max_h \widehat{M}_\zeta(h)$ provides fairly good estimates of h_{true}. This experiment involves modest sample sizes; when we increase the number of documents, the surfaces become more peaked, and $\hat{\hat{h}}$ is closer to h_{true} (experiments not shown).

George (2015) shows that estimates based on $\widetilde{M}_\zeta$ also provide good estimates of h_{true}, and he compares the $\widetilde{M}_\zeta$ and the $\widehat{M}_\zeta$ estimates. From his comparison, we can conclude that the extent of the superiority of the estimates based on $\widehat{M}_\zeta$ is about the same on the synthetic corpora of the present section as in the real data illustration of Section 2.4.

4. Construction of Two Markov Chains with Invariant Distribution $\nu_{h_*,\boldsymbol{w}}$

In order to develop Markov chains on $\boldsymbol{\psi} = (\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{z})$ whose invariant distribution is the posterior $\nu_{h,\boldsymbol{w}}$, we first express the posterior in a convenient form. We start with the familiar formula

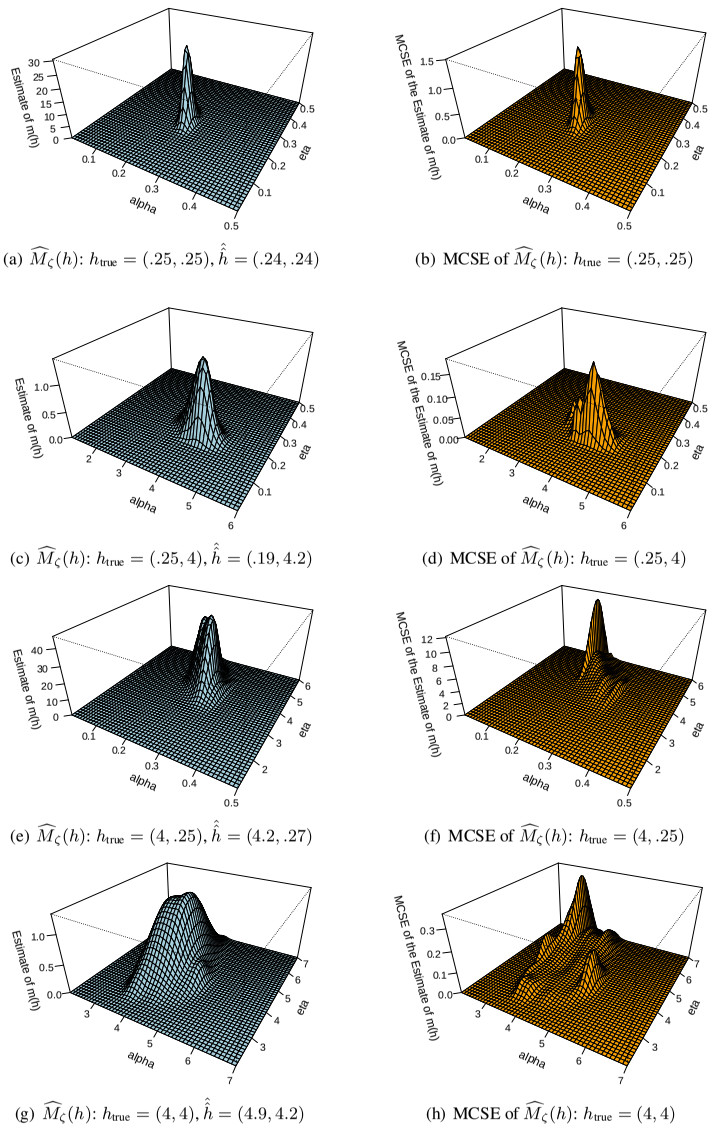
$$\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi}) \propto \ell_{\boldsymbol{w}}(\boldsymbol{\psi})\nu_h(\boldsymbol{\psi}), \tag{4.1}$$

where the likelihood $\ell_{\boldsymbol{w}}(\boldsymbol{\psi}) = p^{(h)}_{\boldsymbol{w}\,|\,\boldsymbol{z},\boldsymbol{\theta},\boldsymbol{\beta}}(\boldsymbol{w}\,|\,\boldsymbol{z},\boldsymbol{\theta},\boldsymbol{\beta})$ is given by line 4 of the LDA model statement. For $d = 1,\ldots,D$ and $j = 1,\ldots,K$, let $S_{dj} = \{i : 1 \le i \le n_d \text{ and } z_{dij} = 1\}$, which is the set of indices of all words in document d whose latent topic variable is j. With this notation, from line 4 of the model statement we have

$$\begin{aligned} p^{(h)}_{\boldsymbol{w}\,|\,\boldsymbol{z},\boldsymbol{\theta},\boldsymbol{\beta}}(\boldsymbol{w}\,|\,\boldsymbol{z},\boldsymbol{\theta},\boldsymbol{\beta}) &= \prod_{d=1}^{D}\prod_{i=1}^{n_d}\prod_{j:z_{dij}=1}^{}\prod_{t=1}^{V}\beta_{jt}^{w_{dit}} = \prod_{d=1}^{D}\prod_{j=1}^{K}\prod_{t=1}^{V}\prod_{i\in S_{dj}}\beta_{jt}^{w_{dit}} \\ &= \prod_{d=1}^{D}\prod_{j=1}^{K}\prod_{t=1}^{V}\beta_{jt}^{\sum_{i\in S_{dj}} w_{dit}} = \prod_{d=1}^{D}\prod_{j=1}^{K}\prod_{t=1}^{V}\beta_{jt}^{m_{djt}}, \end{aligned} \tag{4.2}$$

where $m_{djt} = \sum_{i\in S_{dj}} w_{dit}$ counts the number of words in document d for which the latent topic is j and the index of the word in the vocabulary is t. Recalling the definition of n_{dj} given just before (A.1), and noting that $\sum_{i\in S_{dj}} w_{dit} = \sum_{i=1}^{n_d} z_{dij}w_{dit}$, we see that

$$m_{djt} = \sum_{i=1}^{n_d} z_{dij}w_{dit} \quad \text{and} \quad \sum_{t=1}^{V} m_{djt} = n_{dj}. \tag{4.3}$$



(a) $\widehat{M}_\zeta(h)$: $h_{\text{true}} = (.25, .25)$, $\hat{\hat{h}} = (.24, .24)$ (b) MCSE of $\widehat{M}_\zeta(h)$: $h_{\text{true}} = (.25, .25)$

(c) $\widehat{M}_\zeta(h)$: $h_{\text{true}} = (.25, 4)$, $\hat{\hat{h}} = (.19, 4.2)$ (d) MCSE of $\widehat{M}_\zeta(h)$: $h_{\text{true}} = (.25, 4)$

(e) $\widehat{M}_\zeta(h)$: $h_{\text{true}} = (4, .25)$, $\hat{\hat{h}} = (4.2, .27)$ (f) MCSE of $\widehat{M}_\zeta(h)$: $h_{\text{true}} = (4, .25)$

(g) $\widehat{M}_\zeta(h)$: $h_{\text{true}} = (4, 4)$, $\hat{\hat{h}} = (4.9, 4.2)$ (h) MCSE of $\widehat{M}_\zeta(h)$: $h_{\text{true}} = (4, 4)$

Figure 7: $\widehat{M}_\zeta(h)$ and MCSE of $\widehat{M}_\zeta(h)$ for four values of h_{true}. In each case, $\hat{\hat{h}}$ is close to h_{true}.

Plugging the likelihood (4.2) and the prior (A.1) into (4.1), and absorbing Dirichlet normalizing constants into an overall constant of proportionality, we have

$$\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi}) \propto \left[\prod_{d=1}^{D}\prod_{j=1}^{K}\prod_{t=1}^{V}\beta_{jt}^{m_{djt}}\right]\left[\prod_{d=1}^{D}\prod_{j=1}^{K}\theta_{dj}^{n_{dj}}\right]\left[\prod_{d=1}^{D}\prod_{j=1}^{K}\theta_{dj}^{\alpha_j-1}\right]\left[\prod_{j=1}^{K}\prod_{t=1}^{V}\beta_{jt}^{\eta-1}\right]. \tag{4.4}$$

The expression for $\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi})$ above also appears in the unpublished report Fuentes et al. (2011).

The Conditional Distributions of $(\boldsymbol{\beta},\boldsymbol{\theta})$ Given $\boldsymbol{z}$ and of $\boldsymbol{z}$ Given $(\boldsymbol{\beta},\boldsymbol{\theta})$

All distributions below are conditional distributions given $\boldsymbol{w}$, which is fixed, and henceforth this conditioning is suppressed in the notation. Note that in (4.4), the terms m_{djt} and n_{dj} depend on $\boldsymbol{z}$. By inspection of (4.4), we see that given $\boldsymbol{z}$,

$$\begin{aligned}&\theta_1,\ldots,\theta_D \text{ and } \beta_1,\ldots,\beta_K \text{ are all independent,}\\ &\theta_d \sim \mathrm{Dir}_K\big(n_{d1}+\alpha_1,\ldots,n_{dK}+\alpha_K\big),\\ &\beta_j \sim \mathrm{Dir}_V\big(\textstyle\sum_{d=1}^{D} m_{dj1}+\eta,\ldots,\sum_{d=1}^{D} m_{djV}+\eta\big).\end{aligned} \tag{4.5}$$

From (4.4) we also see that

$$\begin{aligned}p^{(h)}_{\boldsymbol{z}\,|\,\boldsymbol{\theta},\boldsymbol{\beta}}(\boldsymbol{z}\,|\,\boldsymbol{\theta},\boldsymbol{\beta}) &\propto \prod_{d=1}^{D}\prod_{j=1}^{K}\left(\left[\prod_{t=1}^{V}\beta_{jt}^{m_{djt}}\right]\theta_{dj}^{n_{dj}}\right)\\ &= \prod_{d=1}^{D}\prod_{i=1}^{n_d}\prod_{j=1}^{K}\left[\prod_{t=1}^{V}\beta_{jt}^{z_{dij}w_{dit}}\theta_{dj}^{z_{dij}w_{dit}}\right] &(4.6)\\ &= \prod_{d=1}^{D}\prod_{i=1}^{n_d}\prod_{j=1}^{K}\left[\prod_{t=1}^{V}\big(\beta_{jt}\theta_{dj}\big)^{w_{dit}}\right]^{z_{dij}}, &(4.7)\end{aligned}$$

where (4.6) follows from (4.3). Let $p_{dij} = \prod_{t=1}^{V}\big(\beta_{jt}\theta_{dj}\big)^{w_{dit}}$. By inspection of (4.7) we see immediately that given $(\boldsymbol{\theta},\boldsymbol{\beta})$,

$$\begin{aligned}&z_{11},\ldots,z_{1n_1},z_{21},\ldots,z_{2n_2},\ldots,z_{D1},\ldots,z_{Dn_D} \text{ are all independent,}\\ &z_{di} \sim \mathrm{Mult}_K(p_{di1},\ldots,p_{diK}).\end{aligned} \tag{4.8}$$

The conditional distribution of $(\boldsymbol{\beta},\boldsymbol{\theta})$ given by (4.5) can be used, in conjunction with the CGS of Griffiths and Steyvers (2004), to create a Markov chain on $\boldsymbol{\psi}$ whose invariant distribution is $\nu_{h,\boldsymbol{w}}$: if $\boldsymbol{z}^{(1)},\boldsymbol{z}^{(2)},\ldots$ is the CGS, then for $l=1,2,\ldots$, we generate $(\boldsymbol{\beta}^{(l)},\boldsymbol{\theta}^{(l)})$ from $p^{(h)}_{\boldsymbol{\theta},\boldsymbol{\beta}\,|\,\boldsymbol{z}}(\cdot\,|\,\boldsymbol{z}^{(l)})$ given by (4.5) and form $(\boldsymbol{z}^{(l)},\boldsymbol{\beta}^{(l)},\boldsymbol{\theta}^{(l)})$—this is what we have called the Augmented CGS. The CGS is uniformly ergodic (Theorem 1 of Chen and Doss (2017)) and an easy argument shows that the resulting ACGS is therefore also uniformly ergodic (and in fact, the rate of convergence of the ACGS is exactly the same as that of the CGS; see Diaconis et al. (2008, Lemma 2.4)).

The two conditionals (4.5) and (4.8) also enable a direct construction of a two-cycle Gibbs sampler that runs on the pair $(\boldsymbol{z},(\boldsymbol{\beta},\boldsymbol{\theta}))$—this is what we have called the Grouped Gibbs Sampler. This Gibbs sampler has the very attractive feature that it can be parallelized: From (4.5), we see that

given $\boldsymbol{z}$ and $\boldsymbol{w}$, the θ_d's and β_t's are all independent, so can be updated simultaneously by different processors; and from (4.8), we see that given $(\boldsymbol{\beta},\boldsymbol{\theta})$ and $\boldsymbol{w}$, all the components of $\boldsymbol{z}$ are independent, so can also be updated simultaneously by different processors. This scheme was noted earlier by Newman et al. (2009), who dismissed it on the grounds that the Collapsed Gibbs Sampler has superior mixing properties because, according to Liu et al. (1994), collapsing improves the mixing rate. However, the theorem from Liu et al. (1994) that Newman et al. (2009) are citing does not apply to the present situation. To be specific, Liu et al. (1994) consider a Gibbs sampling situation involving three variables X, Y, and Z. They show that a Gibbs sampler on the pair (X,Y) (with Z integrated out), which they call a collapsed Gibbs sampler, is superior to a Gibbs sampler on the triple (X,Y,Z). But for the LDA model, the CGS on $\boldsymbol{z}=(z_{11},\ldots,z_{1n_1},\ldots,z_{D1},\ldots,z_{Dn_D})$ is not a collapsed version of the Gibbs sampler that runs on the pair $\big(\boldsymbol{z},(\boldsymbol{\beta},\boldsymbol{\theta})\big)$ in any sense, so which of the two Gibbs samplers is superior in terms of mixing rate is an open question. George (2015) compared the mixing rates for various parameters empirically, and found that the mixing rate for the CGS is faster, but not much faster. A paper based on George (2015) that studies this Grouped Gibbs Sampler, including its mixing rate and computational complexity, is under preparation (Doss and George, 2017).

5. Evaluation: Choice of Estimator of $\arg\max_h m(h)$ and Resulting Model Fit

The maximizer of the marginal likelihood, $\hat{h}=\arg\max_h m(h)$, may be estimated via the MCMC scheme described in the present paper, or by some version of the EM algorithm (VI-EM or Gibbs-EM). Our main goal in this section is two-fold. (1) We show empirically that neither the VI-EM nor the Gibbs-EM method provides estimates of $\hat{h}$ that are as accurate as ours, and we briefly discuss why theoretically neither VI-EM nor Gibbs-EM, at least in its current implementation, can be expected to work correctly. We also compare VI-EM to Gibbs-EM in terms of accuracy, which to the best of our knowledge has not been done before, and compare VI-EM, Gibbs-EM, and our estimator in terms of speed. This is done in Section 5.1. (2) We consider some of the default choices of h used in the literature that use ad-hoc (i.e. non-principled) criteria. We look at model fit and show empirically that when we use any of the three estimates of $\hat{h}$ (VI-EM, Gibbs-EM, or our serial tempering method), model fit is better than if we use any of the ad-hoc choices. This is done in Section 5.2.

5.1 Comparison of Methods for Estimating $\arg\max_h m(h)$

For uniformity of notation, let $\hat{\hat{h}}_{\mathrm{ST}}$, $\hat{\hat{h}}_{\mathrm{VEM}}$, and $\hat{\hat{h}}_{\mathrm{GEM}}$ be the estimates of $\hat{h}$ formed from serial tempering MCMC, VI-EM, and Gibbs-EM, respectively, and recall that $\hat{\hat{h}}_{\mathrm{ST}}=\hat{\hat{h}}=\arg\max_h \widehat{M_\zeta}(h)$.

VI-EM The estimate $\hat{\hat{h}}_{\mathrm{VEM}}$ proposed by Blei et al. (2003) is obtained as follows. If $h^{(k)}$ is the current value of h, the E-step of the EM algorithm is to calculate $E_{h^{(k)}}\big(\log(p_h(\boldsymbol{\psi},\boldsymbol{w}))\big)$, where $p_h(\boldsymbol{\psi},\boldsymbol{w})$ is the joint distribution of $(\boldsymbol{\psi},\boldsymbol{w})$ under the LDA model indexed by h, and the subscript to the expectation indicates that the expectation is taken with respect to $\nu_{h^{(k)},\boldsymbol{w}}$. This step is infeasible because $\nu_{h^{(k)},\boldsymbol{w}}$ is analytically intractable. We consider $\{q_\phi,\ \phi\in\Phi\}$, a (finite-dimensional) parametric family of analytically tractable distributions on $\boldsymbol{\psi}$, and within this family, we find the distribution, say q_{ϕ_*}, which is "closest" to $\nu_{h^{(k)},\boldsymbol{w}}$. Let $Q(h)$ be the expected value of $\log(p_h(\boldsymbol{\psi},\boldsymbol{w}))$

with respect to q_{ϕ_*}. We view $Q(h)$ as a proxy for $E_{h^{(k)}}\big(\log(p_h(\boldsymbol{\psi}, \boldsymbol{w}))\big)$, and the M-step is then to maximize $Q(h)$ with respect to h, to produce $h^{(k+1)}$. The maximization is done analytically.

The implementation of the EM algorithm through variational inference methods outlined above describes what Blei et al. (2003) do *conceptually*, but not exactly. Actually, Blei et al. (2003) apply VI-EM to a model that is different from ours. In that model, $\boldsymbol{\beta}$ is viewed as a fixed but unknown parameter, to be estimated, and the latent variable is $\boldsymbol{\vartheta} = (\boldsymbol{\theta}, \boldsymbol{z})$. Thus, the observed and missing data are, respectively, $\boldsymbol{w}$ and $\boldsymbol{\vartheta}$, and the marginal likelihood is a function of two variables, h and $\boldsymbol{\beta}$. Abstractly speaking, the description of VI-EM given above is exactly the same. We implemented VI-EM to the version of the LDA model considered in this paper, by modifying the Blei et al. (2003) code. While VI-EM can handle very large corpora with many topics, there are no theoretical results regarding convergence of the sequence $h^{(k)}$ to $\arg\max_h m(h)$, and VI-EM has the following problems: it may have poor performance if the approximation of $\nu_{h^{(k)},\boldsymbol{w}}$ by q_{ϕ_*} is not good; and if the likelihood surface is multimodal, as in Figure 7(e), then it can fail to find the global maximum (as is the case for all EM-type algorithms and also gradient-based approaches).

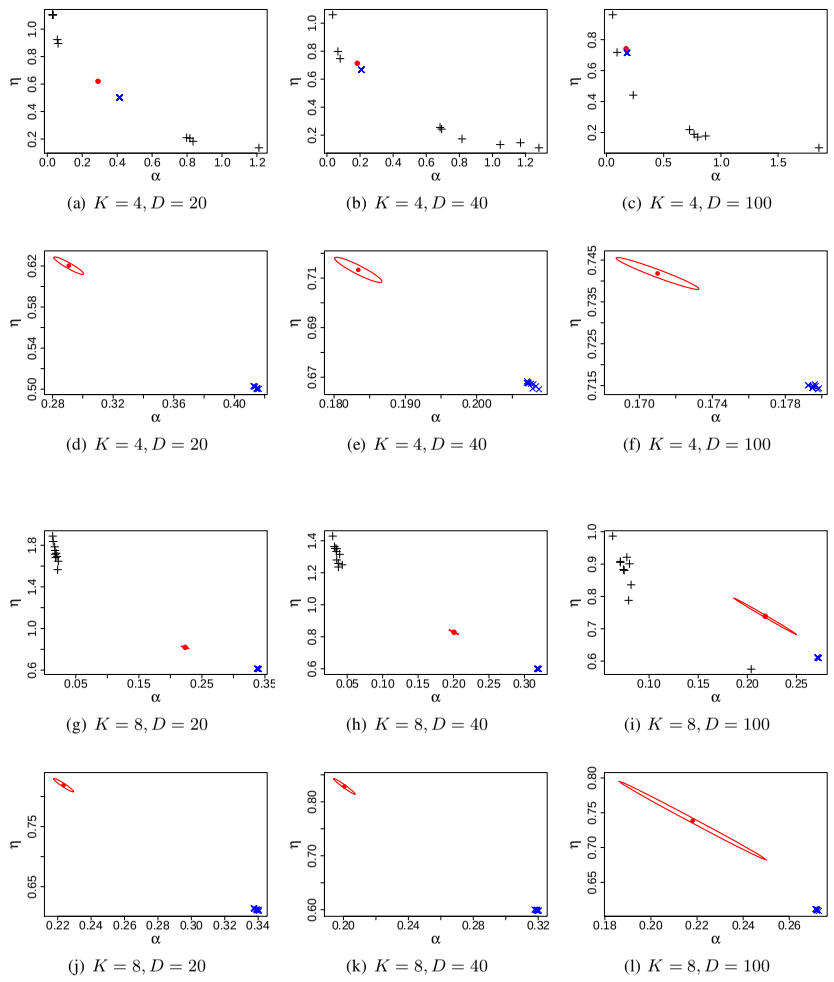
Gibbs-EM Monte Carlo EM (MC-EM), in which the E-step is replaced by a Monte Carlo estimate, dates back to Wei and Tanner (1990), and was introduced to the machine learning community in Andrieu et al. (2003). As mentioned earlier, since an error is introduced at every iteration, there is no reason to expect that the algorithm will converge at all, let alone to the true maximizer of the likelihood. In fact, Wei and Tanner (1990) recognized this problem and suggested that the Markov chain length be increased at every iteration of the EM algorithm. We will let m_k denote the MC length at the k^{th} iteration. Convergence of MC-EM (of which the Gibbs-EM algorithm of Wallach (2008) is a special case) is a nontrivial issue. It was studied by Fort and Moulines (2003), who showed that a minimal condition is that $m_k \to \infty$ at the rate of k^a, for some $a > 1$. However, they do not give guidelines for choosing a. Other conditions imposed in Fort and Moulines (2003) are fairly stringent, and it is not clear whether they are satisfied in the LDA model. In the current implementation of Gibbs-EM (Wallach, 2006), the latent variable is taken to be $\boldsymbol{z}$ (because the standard Markov chain used to estimate posterior distributions in this model is the CGS). At the k^{th} iteration, a Markov chain $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{m_k}$ with invariant distribution equal to the posterior distribution of $\boldsymbol{z}$ given $\boldsymbol{w}$ is generated, and the function $G(h) = (1/m_k)\sum_{i=1}^{m_k} \log(p_h(\boldsymbol{z}_i, \boldsymbol{w}))$ must be maximized. This is done by solving the equation $\nabla G(h) = 0$ using fixed-point iteration, and because $\nabla G(h)$ is computationally intractable, an approximation (Minka, 2003) is used (in effect, a lower bound to $G(h)$ is found, and the lower bound is what is maximized). This approximation introduces a second potential problem for Gibbs-EM. A third potential problem is that, as for VI-EM, the iterations may get stuck near a local maximum when the likelihood surface is multimodal.

To evaluate the performance of the VI-EM, Gibbs-EM, and serial tempering MCMC methods of estimating $\hat{h}$, we generated small synthetic corpora according to the LDA model with the following specifications: the true hyperparameter value is $h = (\eta, \alpha) = (.8, .2)$, the vocabulary size is $V = 20$, the number of words in each document is $n_d = 80$, the number of topics is $K = 4$ and 8, and the number of documents is $D = 20$, 40, and 100, for a total of 6 specifications. For each specification, we formed $\hat{\hat{h}}_{\text{ST}}$, $\hat{\hat{h}}_{\text{VEM}}$, and $\hat{\hat{h}}_{\text{GEM}}$. For $\hat{\hat{h}}_{\text{GEM}}$, we used the algorithm given in Wallach (2006), in which the Markov chain is the CGS. We took the number of cycles of the Gibbs sampler to be 10,000—this is considerably greater than the default value of 20 in the MALLET package (McCallum, 2002); and we formed 10 independent estimates, using 10 different initial values. Likewise, for VI-EM we formed 10 estimates using 10 different initial values. For the

serial tempering estimate, our principal goal was to form a confidence set for $\hat{h}$, and we did this as follows. We ran 10 independent serial tempering chains, for which the sequence $h_1, \ldots, h_J$ consisted of a 7×9 grid of 63 values over the region $(\eta, \alpha) \in [.6, .9] \times [.1, .3]$ (this region was obtained from a small number of iterations of the iterative scheme described in Section 2.3), and the rest of the specifications were the same as those described in the experiments of Section 2.4; each chain was run for 100,000 iterations. Let $\hat{\hat{h}}_{\text{ST}}^{[\ell]}$ be the estimate of $\hat{h}$ formed from serial tempering chain ℓ, for $\ell = 1, \ldots, 10$. According to Theorem 1 in Section 2.1 and Remark 4 in Section 2.3, the independent variables $\hat{\hat{h}}_{\text{ST}}^{[1]}, \ldots, \hat{\hat{h}}_{\text{ST}}^{[10]}$ are approximately bivariate normally distributed with mean vector $\hat{h}$. Therefore, they can be used to form a 95% confidence ellipse for $\hat{h}$, based on Hotelling's T^2 distribution (this ellipse is simply the two-dimensional analogue of the standard t-interval, which is based on the t-distribution). The confidence set could also have been formed from a single long chain, using the method described in Theorem 1; the two methods use about the same computational resources. Figure 8 shows the results, and we make two general observations.

1. From the plots in rows 1 and 3 (plots (a), (b), (c), (g), (h), and (i)), we see that the VI-EM method does not perform well: in each of the 6 cases, the estimates are far from the true value, $\arg\max_h m(h)$, and also strongly depend on the starting values. We created plots (d), (e), (f), (j), (k), and (l), which are zoomed-in versions of plots (a), (b), (c), (g) (h), (i), respectively; these magnify a region which contains the serial tempering estimate and associated confidence ellipse. We see that while the plots in rows 1 and 3 show that the Gibbs-EM estimates greatly outperform the VI-EM estimates (they are both closer to the true value and less dependent on the starting value), the zoomed-in plots in rows 2 and 4 show that the Gibbs-EM points are far from being inside the 95% confidence ellipse. We carried out some experiments in which we followed the recommendations in Fort and Moulines (2003) to increase the number of cycles in the Gibbs sampling inner loop. Specifically, we took $m_1 = 2^7$ and doubled the length of the Gibbs sampler run with every iteration, i.e. we took $m_n = 2^{(6+n)}$, $n = 1, \ldots, 20$. Unfortunately, this did not give significant improvement. The Gibbs-EM estimates were never close to being inside the ellipse, The problem could be with our rate of increase, or that Gibbs-EM simply does not produce consistent estimates, or with the implementation of the maximization step (which uses an approximation).
2. Both Gibbs-EM and VI-EM improve as the number of documents, D, increases. A possible explanation of this is that as D increases, generally speaking the EM algorithm converges faster because the likelihood surface becomes more peaked. Of course, the larger the value of D, the weaker is the effect of the choice of h—this is the Bernstein-von Mises Theorem (see Freedman (1999) and the references therein), which loosely speaking states that as $D \to \infty$, the data swamp the prior.

To assess the computational burden, we computed $\hat{\hat{h}}_{\text{VEM}}$, $\hat{\hat{h}}_{\text{GEM}}$, and $\hat{\hat{h}}_{\text{ST}}$ for the six corpora we considered. For $\hat{\hat{h}}_{\text{VEM}}$, each run consisted of 100 EM iterations and in each EM iteration there were 100 variational inference iterations. For $\hat{\hat{h}}_{\text{GEM}}$, each run consisted of 50 EM iterations and in each EM iteration the CGS was run for 11,000 cycles, of which the first 1000 were deleted as burn-in. For $\hat{\hat{h}}_{\text{ST}}$, each run consisted of 2 tuning iterations with a chain length of 51,000 cycles, and a final iteration with a chain length of 101,000 cycles; and in each case, the first 1000 cycles were deleted as burn-in. Our experiments were conducted through the R programming language, using Rcpp, on a 3.70GHz quad core Intel Xeon Processor E5-1630V3. Table 1 gives the results. From the table,



(a) $K = 4, D = 20$ (b) $K = 4, D = 40$ (c) $K = 4, D = 100$

(d) $K = 4, D = 20$ (e) $K = 4, D = 40$ (f) $K = 4, D = 100$

(g) $K = 8, D = 20$ (h) $K = 8, D = 40$ (i) $K = 8, D = 100$

(j) $K = 8, D = 20$ (k) $K = 8, D = 40$ (l) $K = 8, D = 100$

Figure 8: Plots of estimates of $\hat{h}$ for the 6 corpora described in text. Points marked $\times$ are estimates formed by Gibbs-EM, points marked $+$ are estimates formed by VI-EM. A point marked $\bullet$ is the average of 10 independent estimates of $\hat{h}$ formed via ST chains, and the ellipse is a 95% confidence set for $\hat{h}$ formed from the 10 estimates. The three plots in row 2 are zoomed-in versions of the three plots in row 1, magnifying a region which contains the ST estimate, so the ellipse becomes visible. Similarly, the plots in row 4 are zoomed-in versions of the plots in row 3.

we see that the time for $\hat{\hat{h}}_{\mathrm{ST}}$ is about seven times the time for $\hat{\hat{h}}_{\mathrm{GEM}}$, and the time for $\hat{\hat{h}}_{\mathrm{GEM}}$ is about 55 times the time for $\hat{\hat{h}}_{\mathrm{VEM}}$. These numbers are not as extreme as they look, because for both $\hat{\hat{h}}_{\mathrm{GEM}}$ and $\hat{\hat{h}}_{\mathrm{ST}}$ we could have gotten comparable results with much smaller chain lengths.

K	D	Time for $\hat{\hat{h}}_{\mathrm{VEM}}$	Time for $\hat{\hat{h}}_{\mathrm{GEM}}$	Time for $\hat{\hat{h}}_{\mathrm{ST}}$
8	100	.34	11.64	90.55
8	40	.12	6.08	45.87
8	20	.04	2.96	31.35
4	100	.19	10.73	49.75
4	40	.08	4.88	25.71
4	20	.04	2.23	16.72

Table 1: Length of time, in minutes, it takes to compute the VI-EM, Gibbs-EM, and serial tempering estimates of $\hat{h}$ for six corpora.

5.2 Comparison of Model Fit: Empirical Bayes Choice vs. Ad-Hoc Choices of the Hyperparameter

In the literature, the following choices for $h = (\eta, \alpha)$ have been presented: $h_{\mathrm{DG}} = (0.1, 50/K)$, used in Griffiths and Steyvers (2004); $h_{\mathrm{DA}} = (0.1, 0.1)$, used in Asuncion et al. (2009); and $h_{\mathrm{DR}} = (1/K, 1/K)$, used in the `Gensim` topic modelling package (Řehůřek and Sojka, 2010), a well-known package used in the topic modelling community. These choices are ad-hoc, and not based on any particular principle.

Criterion for Model Fit The criterion we use is a score that is inversely related to the so-called "perplexity" score which is sometimes used in the machine learning literature. When applied to the LDA context, the score is obtained as follows. For $d = 1, \ldots, D$, let $\boldsymbol{w}_{(-d)}$ denote the corpus consisting of all the documents except for document d. To evaluate a given model (in our case the LDA model indexed by a given h), in essence we see how well the model based on $\boldsymbol{w}_{(-d)}$ predicts document d, the held-out document. We do this for $d = 1, \ldots, D$, and take the geometric mean (Wallach et al., 2009). We formalize this as follows. The predictive likelihood of h for the held-out document is

$$L_d(h) = \int \ell_{\boldsymbol{w}_d}(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}_{(-d)}}(\boldsymbol{\psi}), \tag{5.1}$$

where $\ell_{\boldsymbol{w}_d}(\boldsymbol{\psi})$ is the likelihood of $\boldsymbol{\psi}$ for the held-out document d, and $\nu_{h,\boldsymbol{w}_{(-d)}}$ is the posterior distribution of $\boldsymbol{\psi}$ given $\boldsymbol{w}_{(-d)}$. We form the score $S(h) = \left[\prod_{d=1}^{D} L_d(h)\right]^{1/D}$. Two different values of hyperparameter h are compared via their scores. Conceptually, it is easy to estimate $L_d(h)$ by direct Monte Carlo: let $\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots$ be an ergodic Markov chain with invariant distribution $\nu_{h,\boldsymbol{w}_{(-d)}}$. We then approximate the integral by $(1/n)\sum_{i=1}^{n} \ell_{\boldsymbol{w}_d}(\boldsymbol{\psi}_i)$. Care needs to be exercised, however, because in (5.1), the variable $\boldsymbol{\psi}$ in the term $\ell_{\boldsymbol{w}_d}(\boldsymbol{\psi})$ has a dimension that is different than that of the variable $\boldsymbol{\psi}$ in the rest of the integral. Chen (2015) gives a careful description of an MCMC scheme for estimating the integral in (5.1).

Real Datasets Here we compare the fit of LDA models based on various choices of the hyperparameter, on several corpora of real documents. We created two sets of document corpora, one from the 20Newsgroups dataset[2], and the other from the English Wikipedia. The 20Newsgroups dataset is commonly used in the machine learning literature for experiments on applications of text classification and clustering algorithms. It contains approximately 20,000 articles that are partitioned relatively evenly across 20 different newsgroups or categories. We created the second set of corpora from web articles downloaded from the English Wikipedia, with the help of the MediaWiki API[3].

We created the 20Newsgroups corpora as follows. We formed five subsets of the 20Newsgroups dataset, which we call C-1–C-5, with the feature that the articles within the subsets are increasingly difficult to distinguish: for corpus C-1 the topics for the different articles are very different, and for corpus C-5 the topics for the different articles are similar. For each article, we took its true topic label to be the newsgroup to which the article is assigned. Thus, for corpora C-1–C-5, it becomes increasingly difficult to place the articles into the correct newsgroup. We built corpus C-1 from a random subset of articles from the 20Newsgroups categories Medicine, Christianity, and Baseball; these three categories are highly unrelated and easily recognizable from article texts. We built corpus C-2 from a random subset of articles from the categories Automobiles, Motorcycles, Baseball, and Hockey (all four of these categories are classified under the super-category Recreation in the 20Newsgroups dataset), and we built corpus C-3 from a random subset of articles from the categories Cryptography, Electronics, Medicine, and Space (all four of these categories are classified under the super-category Science in the 20Newsgroups dataset). Compared to the categories in corpus C-1, the categories in corpora C-2 and C-3 are moderately related. Lastly, we created corpus C-4 using articles under the categories Autos and Motorcycles, and corpus C-5 using articles under the categories PC Hardware and Mac Hardware. In corpora C-4 and C-5, the corresponding categories are closely related to each other and hard to distinguish from article texts.

We created the Wikipedia corpora as follows. When a Wikipedia article is created, it is typically tagged to one or more categories, one of which is the "primary category." For each article, we took its true topic label to be the primary category label for the article. We created corpus C-6 from a subset of the Wikipedia articles under the categories *Leopardus*, *Lynx*, and *Prionailurus* and corpus C-7 from a subset of the Wikipedia articles under the categories *Acinonyx*, *Leopardus*, *Prionailurus*, and *Puma*. All the categories of corpora C-6 and C-7 are part of the Wikipedia supercategory *Felines*. We created corpus C-8 from a subset of the Wikipedia articles under the categories *Coyotes*, *Jackals*, and *Wolves*. All three categories of corpus C-8 are under the Wikipedia supercategory *Canis*. Finally, we created corpus C-9 from a subset of the Wikipedia articles under the categories *Eagles*, *Falco (genus)*, *Falconry*, *Falcons*, *Harriers*, *Hawks*, *Kites*, and *Owls*. All eight categories of corpus C-9 are subcategories of the Wikipedia category *Birds of Prey*. For each of the four Wikipedia corpora that we created, the categories of the articles are closely related to each other, and fairly hard to distinguish from article texts.

Table 2 gives some information on the nine corpora we created. In the table, the column labeled V gives the vocabulary size for each corpus, the column labeled N gives the total number of words for each corpus, and the column labeled Categories gives newsgroup categories for each 20Newsgroup corpus, and Wikipedia categories for each Wikipedia corpus. The numbers shown in parentheses next to the category names are the number of documents associated with the corresponding categories. For each corpus, we took the number of topics K to be equal to the number of categories for the corpus.

2. http://qwone.com/~jason/20Newsgroups
3. http://www.mediawiki.org/wiki/API:Query

Corpus	Categories	V	N
C-1	sci.med (50), soc.religion.christian (50), rec.sport.baseball (50)	807	12,092
C-2	rec.autos (50), rec.motorcycles (50), rec.sport.baseball (50), rec.sport.hockey (50)	1,061	16,579
C-3	sci.crypt (50), sci.electronics (50), sci.med (50), sci.space (50)	1,033	15,828
C-4	rec.autos (50), rec.motorcycles (50)	488	6,602
C-5	comp.sys.ibm.pc.hardware (50), comp.sys.mac.hardware (50)	502	7,454
C-6	Leopardus (8), Lynx (8), Prionailurus (7)	303	7,788
C-7	Acinonyx (6), Leopardus (8), Prionailurus (7), Puma (8)	622	12,831
C-8	Coyotes (7), Jackals (7), Wolves (8)	447	9,212
C-9	Eagles (62), Falco (genus) (45), Falconry (52), Falcons (10), Harriers (21), Hawks (16), Kites (22), Owls (76)	1,369	116,135

Table 2: Corpora created from the 20Newsgroups dataset and the Wikipedia pages.

Comparison of Model Fit We now compare the performance of the LDA models indexed by $\hat{\hat{h}}_{\text{ST}}$, $\hat{\hat{h}}_{\text{GEM}}$, $\hat{\hat{h}}_{\text{VEM}}$, h_{DR}, h_{DA}, and h_{DG} for corpora C-1–C-9, using the estimate of the score $S(h)$, which we denote by $\widehat{S}(h)$, described in the beginning of this subsection. Details regarding how $\hat{\hat{h}}_{\text{ST}}$ was computed and regarding its accuracy are given in the supplementary document George and Doss. The actual values of $\hat{\hat{h}}_{\text{ST}}$, $\hat{\hat{h}}_{\text{GEM}}$, and $\hat{\hat{h}}_{\text{VEM}}$, are also given in George and Doss.

To compute $\widehat{S}(h)$ for a corpus, for every held-out document, we used Chen's (2015) method with a full Gibbs sampling chain of length 2,000, after discarding a short burn-in period. Table 3 gives the ratios $\widehat{S}(h)/\widehat{S}(\hat{\hat{h}}_{\text{ST}})$, where h is $\hat{\hat{h}}_{\text{GEM}}$, $\hat{\hat{h}}_{\text{VEM}}$, h_{DR}, h_{DA}, and h_{DG}, for all nine corpora. From the table, we make three main observations: (1) Any of the estimates of $\hat{\hat{h}}$ are better than any of the ad-hoc choices, uniformly, and by wide margins. (2) Within the estimates of $\hat{\hat{h}}$, ST does better than either GEM or VEM on the whole, although not in every case, and when it is outperformed, it is not by much. (3) As a general pattern, the lack of fit of the models indexed by the ad-hoc choices of h is worse for the Wikipedia corpora than for the 20Newsgroups corpora. The Wikipedia corpora may be considered "difficult," in the sense that for these corpora the articles are very similar, and thus hard to distinguish from article texts. On the other hand, within the group of estimates of $\hat{\hat{h}}$, it is not clear what are the characteristics of a corpus which affect the fit—there may be factors, beyond similarity of the documents, that are relevant.

Implementation Details To compute $\widehat{M}_{\zeta}(h)$, we implemented the serial tempering scheme described in Section 2 as follows. The size of the subgrid was taken to be $7 \times 13 = 91$, and we used six iterations of the iterative scheme described in Section 2.3 to form the final subgrid, using Markov chains of length 10,000. For the run using the final subgrid, we used three iterations of the scheme given by (2.16) to obtain ζ^{final}, with a Markov chain length of 50,000 per iteration (after a short burn-in period). The final run, using ζ^{final}, also used a Markov chain length of 50,000. To estimate the standard error of $\widehat{M}_{\zeta}(h)$, we used the method of batch means, which is implemented by the R package `mcmcse` in Flegal et al. (2016). Diagnostics that establish that the serial tempering

Corpus	$\hat{\hat{h}}_{\text{GEM}}$	$\hat{\hat{h}}_{\text{VEM}}$	h_{DR}	h_{DA}	h_{DG}
C-1	6.78×10^{-01}	4.83×10^{-01}	3.54×10^{-01}	$1.11 \times 10^{+00}$	8.24×10^{-04}
C-2	5.11×10^{-01}	8.19×10^{-01}	5.23×10^{-01}	2.52×10^{-02}	7.21×10^{-05}
C-3	9.86×10^{-01}	5.58×10^{-01}	2.98×10^{-01}	1.41×10^{-01}	1.33×10^{-02}
C-4	8.21×10^{-01}	7.71×10^{-01}	3.48×10^{-01}	1.22×10^{-01}	6.66×10^{-02}
C-5	9.98×10^{-01}	$1.62 \times 10^{+00}$	4.58×10^{-01}	1.61×10^{-01}	9.36×10^{-02}
C-6	$2.48 \times 10^{+00}$	$1.12 \times 10^{+01}$	7.31×10^{-03}	5.71×10^{-06}	6.57×10^{-08}
C-7	4.39×10^{-01}	$7.82 \times 10^{+00}$	5.34×10^{-03}	1.51×10^{-10}	1.89×10^{-14}
C-8	$2.04 \times 10^{+00}$	6.40×10^{-01}	9.90×10^{-04}	1.77×10^{-09}	3.29×10^{-12}
C-9	$1.04 \times 10^{+00}$	1.75×10^{-02}	2.17×10^{-02}	7.04×10^{-03}	5.56×10^{-09}

Table 3: Ratios of the estimates of the fit criterion $S(h)$ to estimate of $S(\hat{\hat{h}}_{\text{ST}})$ for five choices of h, for all nine corpora. A small number indicates a lack of fit, thus a poor choice of h, and by this criterion, all ad-hoc choices perform poorly.

chain mixes adequately are given in the supplementary document George and Doss. Table 4 gives the time it took to compute $\hat{\hat{h}}_{\text{VEM}}$, $\hat{\hat{h}}_{\text{GEM}}$, and $\hat{\hat{h}}_{\text{ST}}$, for three of the real corpora used in this section.

Corpus	K	$\hat{\hat{h}}_{\text{VEM}}$	$\hat{\hat{h}}_{\text{GEM}}$	$\hat{\hat{h}}_{\text{ST}}$
C-2	4	0.18	7.72	409.75
C-4	2	0.10	3.78	195.90
C-9	8	1.20	59.12	287.97

Table 4: Execution times, in minutes, for three corpora, on a 3.70GHz quad core Intel Xeon Processor E5-1630V3.

It is natural to ask why it has not been noted before that VI-EM and Gibbs-EM sometimes perform poorly. Evaluations have been typically done through a model fit criterion such as the one we used in this subsection, and to the best of our knowledge the literature has not given an assessment of how close $\hat{\hat{h}}_{\text{VEM}}$ and $\hat{\hat{h}}_{\text{GEM}}$ are to h_{true} for corpora generated from an LDA model indexed by h_{true}, as is done in Section 5.1.

6. Discussion

Inference from LDA depends heavily on the choice of hyperparameters used to fit the model. To estimate the hyperparameters, we view the analytically intractable $\hat{h} = \arg\max_h m(h)$, which is a function of the document corpus itself, as the gold standard, and we have developed a methodology for estimating $\hat{h}$. The basis for our approach is a stable method, based on a single serial tempering Markov chain, for estimating the entire marginal likelihood function $m(h)$ (up to a constant). For a given function of the parameters of the model, essentially the same method enables us to estimate the entire family of posterior expectations of the parameters as the hyperparameter varies, and

this feature enables us to carry out an analysis of sensitivity of our inference with respect to the hyperparameters.

Hyperparameter selection is a simple form of model selection and we note that, generally speaking, in carrying out model selection there are two competing goals. One goal is to select the correct model, and the other goal is to select the model that "provides the best inference." These two goals are not the same. The second goal is particularly relevant when the document corpus is a real data set, i.e. the corpus is not necessarily generated from the LDA model, and we use LDA as a convenient model through which to make inference. Selection of the hyperparameter via maximization of the marginal likelihood is akin to maximum likelihood estimation and, as such, should have the standard properties of maximum likelihood estimates. We will avoid giving a technical explanation of this last fact, and instead state it informally as follows: for a corpus generated according to the LDA model indexed by h_{true}, if the corpus is large, then $\hat{h}$ is close to h_{true}. So the empirical Bayes method achieves the first goal by its very nature, and we have verified this empirically in Section 3. The evaluation in Section 5 shows (at least empirically) that the empirical Bayes method also accomplishes the second goal.

A Fully Bayes Approach to Empirical Bayes Inference For serial tempering to work, it is necessary for the grid points $h_1, \ldots, h_J$ to cover $\mathcal{H}$. Unfortunately, when $\dim(\mathcal{H})$ is large, the value of J that is needed is huge, and the approach breaks down. Here we discuss an entirely different method. Although there is no inherent limitation on $\dim(\mathcal{H})$ for the method to work, we view it as useful for the case where $\dim(\mathcal{H})$ is moderate: we re-iterate our caution stated in Remark 3 of Section 2.1 that it is not advisable to use a high-dimensional h.

Suppose $\mathcal{H}$ is a bounded hyper-rectangle. We put a uniform distribution on $\mathcal{H}$, denoted $u(h)$, and in this fully-Bayes situation the parameter is now $(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{z}, h)$. The marginal posterior distribution of h is then $\pi(h) \propto m_{\boldsymbol{w}}(h)u(h) \propto m_{\boldsymbol{w}}(h)$, and we see that $\arg\max_h m_{\boldsymbol{w}}(h) = \arg\max_h \pi(h)$. Suppose that $(\boldsymbol{\beta}^{(1)}, \boldsymbol{\theta}^{(1)}, \boldsymbol{z}^{(1)}, h^{(1)}), \ldots, (\boldsymbol{\beta}^{(n)}, \boldsymbol{\theta}^{(n)}, \boldsymbol{z}^{(n)}, h^{(n)})$ is a Markov chain whose invariant distribution is the posterior distribution of $(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{z}, h)$ given $\boldsymbol{w}$ [see Wallach (2008)]. From the marginal sequence $h^{(1)}, \ldots, h^{(n)}$ we may estimate $\pi(h)$ via a multivariate density estimator, and hence $\arg\max_h \pi(h)$. Call this estimate $\bar{h}$. We then use (2.1), with $\bar{h}$ as the value of h_*, to estimate $m_{\boldsymbol{w}}(h)$ in a small neighborhood of $\bar{h}$, which is all that we need in order to estimate $\arg\max_h m_{\boldsymbol{w}}(h)$. In effect, $\bar{h}$ is an initial coarse estimate of $\arg\max_h m_{\boldsymbol{w}}(h)$, and (2.1) is then used to fine-tune it. We hope to develop this idea fully in future work.

Acknowledgments

We are grateful to three referees for their very helpful constructive criticism. This work is supported by the International Center for Automated Research at the UF Levin College of Law, NSF Grant DMS-11-06395, and NIH grant P30 AG028740.

Appendix

A.1 A Likelihood Ratio Formula for the Parameters in the LDA Model

To obtain the ratio of densities formula (2.2), we note that from the hierarchical nature of the LDA model we have

$$\nu_h(\boldsymbol{\psi}) = \nu_h(\boldsymbol{\beta},\boldsymbol{\theta},\boldsymbol{z}) = p^{(h)}_{\boldsymbol{z}\,|\,\boldsymbol{\theta},\boldsymbol{\beta}}(\boldsymbol{z}\,|\,\boldsymbol{\theta},\boldsymbol{\beta})\, p^{(h)}_{\boldsymbol{\theta}}(\boldsymbol{\theta})\, p^{(h)}_{\boldsymbol{\beta}}(\boldsymbol{\beta})$$

in self-explanatory notation, where $p^{(h)}_{\boldsymbol{z}\,|\,\boldsymbol{\theta},\boldsymbol{\beta}}$, $p^{(h)}_{\boldsymbol{\theta}}$, and $p^{(h)}_{\boldsymbol{\beta}}$ are given by lines 3, 2, and 1, respectively, of the LDA model. Let $n_{dj} = \sum_{i=1}^{n_d} z_{dij}$, i.e. n_{dj} is the number of words in document d that are assigned to topic j. Using the Dirichlet and multinomial distributions specified in lines 1–3 of the model, we obtain

$$\nu_h(\boldsymbol{\psi}) = \left[\prod_{d=1}^{D}\prod_{j=1}^{K}\theta_{dj}^{n_{dj}}\right]\left[\prod_{d=1}^{D}\left(\frac{\Gamma(\sum_{j=1}^{K}\alpha_j)}{\prod_{j=1}^{K}\Gamma(\alpha_j)}\prod_{j=1}^{K}\theta_{dj}^{\alpha_j-1}\right)\right]\left[\prod_{j=1}^{K}\left(\frac{\Gamma(V\eta)}{\Gamma(\eta)^V}\prod_{t=1}^{V}\beta_{jt}^{\eta-1}\right)\right]. \tag{A.1}$$

We now apply (A.1) to ν_h and ν_{h_*} and obtain (2.2).

A.2 Proof of Theorem 1

The convergence in (2.1) holds for each fixed h; however, $\arg\max_h B_n(h)$ depends on the function $B_n(\cdot)$. Before proving Theorem 1 we provide an example to show that if f_n and f are real-valued functions, then convergence of f_n to f pointwise does not imply convergence of $\arg\max_h f_n(h)$ to $\arg\max_h f(h)$. In our example, the domain of the functions is the interval $[0,1]$, and the functions are displayed in Figure 9. The functions f_n and f are identical on the interval $[2/n, 1]$. Clearly $f_n(h) \to f(h)$ for each $h \in [0,1]$, but $\arg\max_h f_n(h) = 1/n$ while $\arg\max_h f(h) = .9$.

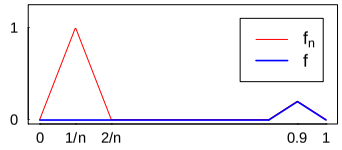


Figure 9: Non-convergence of the argmax.

Theorem 1 refers to the regularity conditions below.

A1 The hyperparameter space $\mathcal{H}$ is compact.

A2 The maximizer of $m(\cdot)$ is unique (thus it makes sense to talk about $\arg\max_h m(h)$).

A3 The maximizer of $m(\cdot)$ is in $\mathcal{H}$.

A4 For each n, the maximizer of $B_n(\cdot)$ is unique (thus we can talk about $\arg\max_h B_n(\cdot)$).

A5 The point $h_* = (\eta^*, \boldsymbol{\alpha}^*)$ satisfies $2\eta - \eta^* > 0$ and $2\alpha_j - \alpha_j^* > 0$, $j = 1,\ldots,K$ for all $h = (\eta, \boldsymbol{\alpha}) \in \mathcal{H}$.

A6 The marginal likelihood function $m(\cdot)$ is twice continuously differentiable in $\mathcal{H}$, and the $p \times p$ Hessian matrix $\nabla_h^2 m(\arg\max_h m(h))$ is nonsingular.

Proof of Part 1 Note the following:

- In Section 4 we showed that the ACGS is uniformly ergodic. So in particular, it is Harris ergodic.

- The LDA model is an exponential family with parameter (η, α) (Section 3.3 of Wainwright and Jordan (2008)).
- In their Remark 1, Doss and Park (2018) show that if $\{\nu_h,\ h \in \Omega\}$ is an exponential family, where Ω is the natural parameter space, and if $\mathcal{H}$ is a compact subset of the interior of Ω, then $\int \sup_{h\in\mathcal{H}}(\nu_h/\nu_{h_*})\, d\nu_{h_*,\boldsymbol{w}} < \infty$. (In empirical process theory, finiteness of this integral is the main condition that is needed to obtain uniformity in the Law of Large Numbers.)
- In the context of the present situation, Theorem 3 of Doss and Park (2018) states that under Harris ergodicity of the sequence $\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots$ and finiteness of $\int \sup_{h\in\mathcal{H}}(\nu_h/\nu_{h_*})\, d\nu_{h_*,\boldsymbol{w}}$, the convergence in (2.1) is uniform on $\mathcal{H}$, i.e. Condition C3 of Section 2.1 holds.
- Suppose that f_n, $n = 1, 2, \ldots$ and f are real-valued functions defined on a compact subset X of Euclidean space. Suppose further that f is continuous and that each of f_n, $n = 1, 2, \ldots$ and f has a unique maximizer. Under these conditions, uniform convergence of f_n to f on X implies $\arg\max_{x\in\mathsf{X}} f_n(x) \to \arg\max_{x\in\mathsf{X}} f(x)$. Verification of this fact is routine. A detailed proof is given in Lemma 1 of Doss and Park (2018).

Combining these facts, we see that under A1–A4, $\arg\max_h B_n(h) \to \arg\max_h m(h)$ with probability one (Assumptions A5 and A6 are not needed for Part 1 of the theorem.) □

Proof of Part 2 Theorem 4 of Doss and Park (2018) asserts the asymptotic normality stated in Part 2 of Theorem 1 under A1–A4 and A6, the condition

$$\text{for every } h \in \mathcal{H} \text{ there exists } \epsilon > 0 \text{ such that } \int \|\nabla_h(\nu_h/\nu_{h_*})\|^{2+\epsilon}\, d\nu_{h_*,\boldsymbol{w}} < \infty, \tag{A.2}$$

where $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^p$, and the condition that the Markov chain used is geometrically ergodic. Using standard calculus, we can check that if $2\eta - \eta^* > 0$ and $2\alpha_j - \alpha_j^* > 0$, $j = 1,\ldots,K$, then for sufficiently small ϵ the integral in (A.2) is finite. Thus, Condition A5 implies (A.2). As mentioned in the proof of Part 1 of the theorem, the ACGS is uniformly ergodic; so in particular, it is geometrically ergodic. Thus, we have established the asymptotic normality stated in Part 2 of the theorem under A1–A6. □

Proof of Part 3 The variance matrix Σ in Part 2 is analytically intractable, but fortunately is easy to estimate via the method of batching, as follows. For $j = 1,\ldots,J$, let $h^{[j]}$ be the estimate of the argmax produced from batch j, and let $h^{[\cdot]}$ be the estimate of the argmax produced from the entire sequence. The batch-based estimate is $\widehat{\Sigma}_n = (n/J)\big\{[1/(J-1)]\sum_{j=1}^{J}\big(h^{[j]} - h^{[\cdot]}\big)\big(h^{[j]} - h^{[\cdot]}\big)^\top\big\}$. (The quantity inside the braces is essentially the sample covariance matrix of $h^{[1]},\ldots,h^{[J]}$, except that we use $h^{[\cdot]}$ instead of the average of $h^{[1]},\ldots,h^{[J]}$ as the centering value; and the term n/J is the number of samples per batch.) Estimates of the covariance matrix based on batching are consistent under very general conditions which include that $J \to \infty$ as $n \to \infty$. The literature recommends taking $J = n^{1/2}$; see Flegal et al. (2008) and also Jones et al. (2006). Invertibility of $\widehat{\Sigma}_n$ for large n follows from positive definiteness of Σ and the convergence $\widehat{\Sigma}_n \xrightarrow{\text{a.s.}} \Sigma$; in fact we have $\widehat{\Sigma}_n^{-1} \xrightarrow{\text{a.s.}} \Sigma^{-1}$. Therefore, applying Part 2, we get

$$n^{1/2}\big(\arg\max_h B_n(h) - \arg\max_h m(h)\big)\widehat{\Sigma}_n^{-1} n^{1/2}\big(\arg\max_h B_n(h) - \arg\max_h m(h)\big)^\top \xrightarrow{d} \chi^2_p,$$

which establishes the statement regarding the ellipse. □

Proof of Theorem 1 for the Serial Tempering Chain The main change in the proof is that the requirement (A.2) is replaced with

$$\text{for every } h \in \mathcal{H} \text{ there exists } \epsilon > 0 \text{ such that } \int \|\nabla_h(\nu_h/\nu_\zeta)\|^{2+\epsilon}\, df_\zeta < \infty, \tag{A.3}$$

where $\nu_\zeta = (1/J)\sum_{j=1}^{J} \nu_j(\boldsymbol{\psi}_i)/\zeta_j$, and f_ζ is given by (2.9). It is easy to see that (A.3) is satisfied if the stipulation on h_* given by (2.3) holds for $h^{(j)}$ for at least one index j. □

A.3 Proof of Validity of the Confidence Band for $\{I(h),\, h \in \mathcal{H}\}$

In addition to assuming that $J \to \infty$ and $n/J \to \infty$, we will need the following conditions:

A7 The stipulation on h_* given by (2.3) holds for $h^{(j)}$ for at least one index j.

A8 The function g satisfies the moment condition

$$\text{for every } h \in \mathcal{H} \text{ there exists } \epsilon > 0 \text{ such that } \int \left(g\frac{\nu_h}{\nu_\zeta}\right)^{2+\epsilon} df_\zeta < \infty.$$

Note that A8 is automatically satisfied if A7 holds and g is bounded (for example if g is an indicator function, as in Section 2.4). In the following, we will assume Conditions A1, A7, and A8.

The heart of the proof is the assertion that $\sup_{h\in\mathcal{H}} n^{1/2}\big|\hat{I}^{\text{st}}_\zeta(h) - I(h)\big|$ has a limiting distribution as $n \to \infty$, and we show this in three steps:

1. We observe that for each h, $n^{1/2}\big(\hat{I}^{\text{st}}_\zeta(h) - I(h)\big)$ has an asymptotic normal distribution.
2. We show that more can be said, and that the stochastic process $\big\{n^{1/2}\big(\hat{I}^{\text{st}}_\zeta(h) - I(h)\big),\, h \in \mathcal{H}\big\}$ converges in distribution to a mean-zero Gaussian process indexed by h.
3. We conclude from Step 2 that $\sup_{h\in\mathcal{H}} n^{1/2}\big|\hat{I}^{\text{st}}_\zeta(h) - I(h)\big|$ has a limiting distribution as $n \to \infty$.

We now provide the details.

1. Note that $\hat{I}^{\text{st}}_\zeta(h)$, defined in (2.14), is a ratio of $\widehat{M}_\zeta(h)$ and $\widehat{U}_\zeta(h)$, which are given by (2.10) and (2.12), respectively. Each of these is an average of a function of $\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_n$, so we have a bivariate central limit theorem, as follows. For economy of notation, let $U_i^{(h)}$ be the summands in (2.12) and let $M_i^{(h)}$ be the summands in (2.10). We have

$$n^{1/2}\begin{pmatrix} \frac{1}{n}\sum_{i=1}^{n} U_i^{(h)} - \frac{m(h)}{c_\zeta/J}\int g\, d\nu_{h,\boldsymbol{w}} \\ \frac{1}{n}\sum_{i=1}^{n} M_i^{(h)} - \frac{m(h)}{c_\zeta/J} \end{pmatrix} \xrightarrow{d} \mathcal{N}_2(0, \Sigma_h),$$

where Σ_h is a covariance matrix. (If the $\boldsymbol{\psi}$'s were an iid sequence, then Σ_h would be simply the covariance matrix of the pair $(U_1^{(h)}, M_1^{(h)})$; however, in the present situation, Σ_h is the more complicated covariance matrix that arises in the Markov chain central limit theorem.) Therefore, by the delta method applied to the function $\varphi\colon \mathbb{R}^2 \to \mathbb{R}$ defined by $\varphi(u, m) = u/m$, we have

$$n^{1/2}\left(\frac{\sum_{i=1}^{n} U_i^{(h)}}{\sum_{i=1}^{n} M_i^{(h)}} - \int g\, d\nu_{h,\boldsymbol{w}}\right) \xrightarrow{d} \mathcal{N}\big(0, (\nabla\varphi)^\top \Sigma_h \nabla\varphi\big), \tag{A.4}$$

where the gradient $\nabla\varphi$ is evaluated at $\big((1/n)\sum_{i=1}^{n} U_i^{(h)}, (1/n)\sum_{i=1}^{n} M_i^{(h)}\big)$. Now note that the quantity to the left of the "$\xrightarrow{d}$" sign in (A.4) is precisely $n^{1/2}\big(\hat{I}^{\text{st}}_\zeta(h) - I(h)\big)$.

2. To extend convergence in distribution for each fixed h to convergence as a stochastic process, we use Part 4 of Theorem 6 of Doss and Park (2018). Because we assume Conditions A1, A7, and A8 and because the distributions of the latent parameters in the LDA model form an exponential family, the regularity conditions for that theorem are satisfied, and we conclude that $n^{1/2}\big(\hat{I}^{\text{st}}_\zeta(\cdot) - I(\cdot)\big) \xrightarrow{d} G(\cdot)$, where $G(\cdot)$ is a mean-zero Gaussian process indexed by h. Here, convergence in distribution takes place in $C(\mathcal{H})$, the space of continuous real-valued functions defined on $\mathcal{H}$, endowed with the sup-norm topology.
3. The map $T\colon C(\mathcal{H}) \to [0, 1]$ defined by $T(f) = \sup_{h\in\mathcal{H}} |f(h)|$ is continuous, so from Step 2 we conclude that $\sup_{h\in\mathcal{H}} n^{1/2}\big|\hat{I}^{\text{st}}_\zeta(h) - I(h)\big| \xrightarrow{d} \sup_{h\in\mathcal{H}} |G(h)|$.

Substitution of the $\mathcal{S}_j$'s for the S_j's is valid under the assumption that $J \to \infty$, convergence in probability of $\mathcal{S}_{\lceil .95J\rceil}$ to $c_{.95}$ is a consequence of the condition $n/J \to \infty$, and the validity of the bands now follows. The literature's recommendation of $J = n^{1/2}$ is made in the different context of estimating the variance of an average, not for forming globally-valid confidence bands; nevertheless, in our experience this choice works well also in the present situation.

References

Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50:5–43, 2003.

Arthur Asuncion, Max Welling, Padhraic Smyth, and Yee Whye Teh. On smoothing and inference for topic models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 27–34, Arlington, Virginia, United States, 2009. AUAI Press.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

Zhe Chen. *Inference for the Number of Topics in the Latent Dirichlet Allocation Model via Bayesian Mixture Modelling*. PhD thesis, University of Florida, 2015.

Zhe Chen and Hani Doss. Inference for the number of topics in the latent Dirichlet allocation model via Bayesian mixture modelling. Technical report, Department of Statistics, University of Florida, 2017.

Persi Diaconis, Kshitij Khare, and Laurent Saloff-Coste. Gibbs sampling, exponential families and orthogonal polynomials (with discussion). *Statistical Science*, 23:151–178, 2008.

Hani Doss and Clint P. George. Theoretical and empirical evaluation of a grouped Gibbs sampler for parallel computation in the LDA model. Technical report, Department of Statistics, University of Florida, 2017.

Hani Doss and Yeonhee Park. An MCMC approach to empirical Bayes inference and Bayesian sensitivity analysis via empirical processes. *Annals of Statistics* (to appear), 2018.

James M. Flegal, Murali Haran, and Galin L. Jones. Markov chain Monte Carlo: Can we trust the third significant figure? *Statistical Science*, 23:250–260, 2008.

James M. Flegal, John Hughes, and Dootika Vats. *mcmcse: Monte Carlo Standard Errors for MCMC*. Riverside, CA and Minneapolis, MN, 2016. R package version 1.2-1.

Gersende Fort and Eric Moulines. Convergence of the Monte Carlo expectation maximization for curved exponential families. *The Annals of Statistics*, 31:1220–1259, 2003.

David Freedman. Wald Lecture: On the Bernstein-von Mises theorem with infinite-dimensional parameters. *The Annals of Statistics*, 27:1119–1141, 1999.

Claudio Fuentes, Vikneshwaran Gopal, George Casella, Clint P. George, Taylor C. Glenn, Joseph N. Wilson, and Paul D. Gader. Product partition models for Dirichlet allocation. Technical report, Department of Computer and Information Science and Engineering, University of Florida, 2011.

Clint P. George. *Latent Dirichlet Allocation: Hyperparameter Selection and Applications to Electronic Discovery*. PhD thesis, University of Florida, 2015.

Clint P. George and Hani Doss. Supplement to "Principled selection of hyperparameters in the latent Dirichlet allocation model," 2018.

Edward I. George and Dean P. Foster. Calibration and empirical Bayes variable selection. *Biometrika*, 87:731–747, 2000.

Charles J. Geyer and Elizabeth A. Thompson. Annealing Markov chain Monte Carlo with applications to ancestral inference. *Journal of the American Statistical Association*, 90:909–920, 1995.

Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101:5228–5235, 2004.

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

James P. Hobert and George Casella. The effect of improper priors on Gibbs sampling in hierarchical linear mixed models. *Journal of the American Statistical Association*, 91(436):1461–1473, 1996.

Galin L. Jones, Murali Haran, Brian S. Caffo, and Ronald Neath. Fixed-width output analysis for Markov chain Monte Carlo. *Journal of the American Statistical Association*, 101:1537–1547, 2006.

J. S. Liu, W. H. Wong, and A. Kong. Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes. *Biometrika*, 81:27–40, 1994.

Enzo Marinari and Giorgio Parisi. Simulated tempering: A new Monte Carlo scheme. *Europhysics Letters*, 19:451–458, 1992.

Andrew Kachites McCallum. MALLET: A machine learning for language toolkit. `http://mallet.cs.umass.edu`, 2002.

Thomas P. Minka. Estimating a Dirichlet distribution, 2003. URL `http://research.microsoft.com/~minka/papers/dirichlet/`.

David Newman, Arthur Asuncion, Padhraic Smyth, and Max Welling. Distributed algorithms for topic models. *Journal of Machine Learning Research*, 10:1801–1828, 2009.

Radim Řehůřek and Petr Sojka. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, 2010. ELRA.

Christian P. Robert. *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*. Springer-Verlag, New York, 2001.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101:1566–1581, 2006.

Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

Hanna M. Wallach. Topic modeling: Beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 977–984, New York, NY, USA, 2006. ACM.

Hanna M. Wallach. *Structured Topic Models for Language*. PhD thesis, University of Cambridge, 2008.

Hanna M Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1105–1112. ACM, 2009.

Greg C. G. Wei and Martin A. Tanner. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85: 699–704, 1990.

Journal of Machine Learning Research 18 (2018) 1-17 Submitted 9/17; Revised 2/18; Published 4/18

Deep Learning the Ising Model Near Criticality

Alan Morningstar aormorningstar@gmail.com
Roger G. Melko rgmelko@uwaterloo.ca
Perimeter Institute for Theoretical Physics
Waterloo, Ontario, N2L 2Y5, Canada
and
Department of Physics and Astronomy
University of Waterloo
Waterloo, Ontario, N2L 3G1, Canada

Editor: Amos Storkey

Abstract

It is well established that neural networks with deep architectures perform better than shallow networks for many tasks in machine learning. In statistical physics, while there has been recent interest in representing physical data with generative modelling, the focus has been on shallow neural networks. A natural question to ask is whether deep neural networks hold any advantage over shallow networks in representing such data. We investigate this question by using unsupervised, generative graphical models to learn the probability distribution of a two-dimensional Ising system. Deep Boltzmann machines, deep belief networks, and deep restricted Boltzmann networks are trained on thermal spin configurations from this system, and compared to the shallow architecture of the restricted Boltzmann machine. We benchmark the models, focussing on the accuracy of generating energetic observables near the phase transition, where these quantities are most difficult to approximate. Interestingly, after training the generative networks, we observe that the accuracy essentially depends only on the number of neurons in the first hidden layer of the network, and not on other model details such as network depth or model type. This is evidence that shallow networks are more efficient than deep networks at representing physical probability distributions associated with Ising systems near criticality.

Keywords: deep learning, restricted Boltzmann machine, deep belief network, deep Boltzmann machine

1. Introduction

It is empirically well supported that neural networks with deep architectures perform better than shallow networks for certain machine learning tasks involving data with complex, hierarchal structures (Hinton and Salakhutdinov, 2006). This has led to an intense focus on developing deep neural networks for use on data sets related to natural images, speech, video, social networks, etc. with demonstrable success. In regard to theoretical understanding, among the most profound questions facing the modern machine learning community is why, and under what conditions, are deep neural networks superior to shallow (single-hidden-layer) models. Further, does the success of deep architectures in extracting features from conventional big data translate into arenas with other highly complex data sets, such as those encountered in the physical sciences?

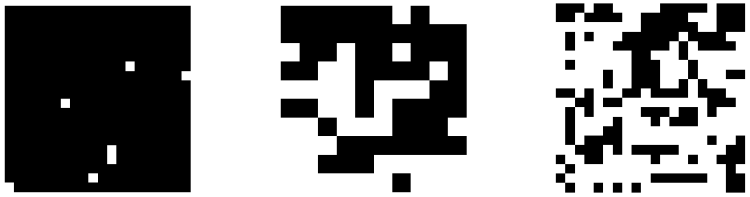
Very recently, the statistical physics community has become engaged in exploring the representational power of generative neural networks such as the restricted Boltzmann machine (RBM; Carleo and Troyer, 2017; Wetzel, 2017). There, the "curse of dimensionality" is manifest in the size of state space (Carrasquilla and Melko, 2017), which for example, grows as 2^N for a system of N binary (or *Ising*) variables. The central problem in much of statistical physics is related to determining and sampling the probability distribution of this state space. Therefore, the possibility of efficiently modelling a physical probability distribution with a generative neural network has broad implications for condensed matter, materials physics, quantum computing, and other areas of the physical sciences involving the statistical mechanics of N-body systems.

The universal approximation theorem, while establishing the capability of neural networks as generative models in theory, only requires the very weak assumption of exponentially large resources (Le Roux and Bengio, 2008; Montúfar and Morton, 2015). In both quantum and classical statistical physics applications, the goal is to calculate macroscopic *features* (for example, a heat capacity or a susceptibility) from the N-body equations. For this, an *efficient* representation is desired that demands computer resources which scale polynomially, typically in the number of particles or lattice sites, $\mathcal{O}(N^c)$ where c is a constant. In this case, "resources" could refer to memory or time; for example, the number of model parameters, the size of the data set for training, or the number of samples required of the model to accurately reproduce physical features.

Thus we can refine our question: are deep neural networks more efficient than shallow networks when modelling physical probability distributions, and if so, why? In this paper we address the dependence of generative modelling on network depth directly for the prototypical physical system of the two-dimensional (2D) lattice Ising model. The most important feature of this system is the existence of a phase transition (or *critical point*), with subtle physical features that are theoretically known to require a non-trivial multi-scale description. We define criteria for the accuracy of a generative model, based on the convergence of the energy and energy fluctuations of samples generated from it, to the exact physical values calculated near the phase transition. Using various types of generative neural networks with a multitude of widths and depths, we find the surprising result that accurate modelling depends only on the *architecture* of the network—defined as the set of integers specifying the number of neurons in each network layer. Furthermore, we show that the accuracy of the neural network only depends significantly on the number of hidden units in the first hidden layer. This illustrates that in this physical example, depth does not contribute to the representational efficiency of a generative neural network, even near a non-trivial scale-invariant phase transition.

2. The 2D Ising System Near Criticality

In order to benchmark the representational power of generative neural networks, we replace naturally occurring data common to machine learning applications with synthetic data for a system intimately familiar to statistical physicists, the 2D Ising model. This system is defined on an N-site square lattice, with binary variables (spins) $x_i \in \{0, 1\}$ at each lattice



(a) $T < T_c$ (b) $T \approx T_c$ (c) $T > T_c$

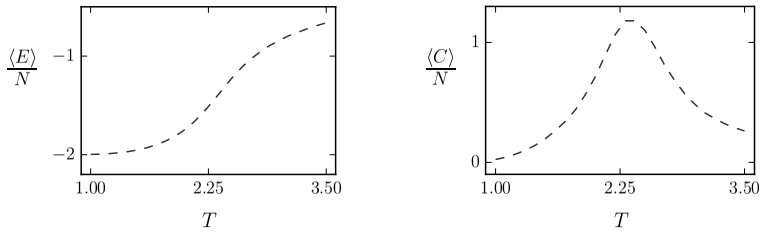
Figure 1: Representative Ising spin configurations on a $N = 20 \times 20$ site lattice. In (a) is the ferromagnetic phase below T_c; (b) is the critical point or phase transition; and (c) is above T_c in the paramagnetic phase. Such configurations play the role of 2D images for the training of generative neural networks in this paper.

site i. The physical probability distribution defining the Ising model is the Boltzmann distribution

$$q(x) = \frac{1}{Z} \exp(-\beta H(x)) \tag{1}$$

determined by the Hamiltonian (or physical energy) $H(x) = -\sum_{\langle i,j \rangle} \sigma_i \sigma_j$, where $\sigma_i = 2x_i - 1$. Here, Z is the normalization factor (or partition function), and $\langle i, j \rangle$ denotes nearest-neighbour sites. As a function of inverse temperature $\beta = 1/k_B T$ (with $k_B = 1$), the Ising system displays two distinct phases, with a phase transition at $T_c = 2/\log\left(1+\sqrt{2}\right) \approx 2.2693$ in the thermodynamic limit where $N \to \infty$ (Kramers and Wanniers, 1941; Onsager, 1944). In this limit, the length scale governing fluctuations—called the *correlation length* ξ—diverges to infinity resulting in emergent macroscopic behavior such as singularities in measured observables like heat capacity or magnetic susceptibility. Phase transitions with a diverging ξ are called *critical points*. Criticality makes understanding macroscopic phenomena (or features) from the microscopic interactions encoded in $H(x)$ highly non-trivial. The most successful theoretical approach is the renormalization group (RG), which is a muli-scale description where features are systematically examined at successively deeper coarse-grained levels of scale (Wilson and Fisher, 1972). Physicists have begun to explore in ernest the conceptual connection between deep learning and the RG (Mehta and Schwab, 2014; Koch-Janusz and Ringel, 2017).

We concentrate on two-dimensional lattices with finite N, where remnants of the true thermodynamic ($N \to \infty$) phase transition are manifest in physical quantities. We are interested in approximately modelling the physical distribution $q(x)$, both at and near this phase transition, using generative neural networks. Real-space spin configurations at fixed temperatures play the role of data sets for training and testing the generative models of the next section. Configurations—such as those illustrated in Figure 1—are sampled from Equation 1 using standard Markov Chain Monte Carlo techniques described elsewhere; see for example, Newman and Barkema (1999). For a given temperature T and lattice size N, training and testing sets of arbitrary size can be produced with relative ease, allowing us



(a) Energy per Ising spin. (b) Heat capacity per Ising spin.

Figure 2: Physical data for the energy and heat capacity of an $N = 64$ square-lattice Ising system, calculated using Markov Chain Monte Carlo.

to explore the representational power of different generative models without the concern of regularization due to limited data.

In order to quantify the representational power of a given neural network, we use the standard recently introduced by Torlai and Melko (2016), which is the faithful recreation of physical observables from samples of the visible layer of a fully-trained generative neural network. As a reference, observables such as average energy E and heat capacity $C = \partial E/\partial T$ are calculated from the physical distribution, that is, $\langle E \rangle = Z^{-1} \sum_x q(x) H(x)$ and $\langle C \rangle = (\langle E^2 \rangle - \langle E \rangle^2)/T^2$. Using shallow RBMs, Torlai and Melko examined the convergence of these and other physical observables for 2D Ising systems on different size lattices. For a lattice of N Ising spins, the number of hidden neurons in the RBM, N_{h_1} (see below), was increased until the observables modelled by the RBM approached the exact, physical values calculated by Monte Carlo. In the RBM, N also represents the number of visible units. Thus, the ratio $\alpha \equiv N_{h_1}/N$—describing the network resources per lattice site—serves as a convergence parameter measuring the overall efficiency of this machine learning approach. It was shown by Chen et al. (2017) that this ratio is bounded above by $\alpha = 2$ for the same 2D Ising model we consider in this paper, a bound consistent with the results of numerical simulations detailed in sections below.

In Figure 2 we illustrate the energy E and heat capacity C calculated via Monte Carlo for the $N = 64$ square-lattice Ising model. Near the critical point $T_c \approx 2.2693$, C retains a finite-size remnant of the divergence expected when $N \to \infty$. As observed by Torlai, it is here where the accuracy of the modelled neural network data suffers most from an insufficient number of hidden units. In this paper, we therefore concentrate on the convergence of E and C near T_c as a function of model parameters.

3. Generative Graphical Models

In this section, we briefly introduce the shallow and deep stochastic neural networks that serve as generative graphical models for reproducing Ising observables. These stochastic neural networks are made up of a set of binary neurons (nodes) which can be in the state 0

or 1, each with an associated bias, and weighted connections (edges) to other neurons. For this type of model, neurons are partitioned into two classes, visible and hidden neurons, such that the model defines a joint probability distribution $p(v, h)$ over the visible and hidden neurons, v and h respectively (or h_1, h_2, and so on for multiple hidden layers). The weights and biases of a model—collectively referred to as "model parameters", and indicated by θ—determine the state of each neuron given the state of other connected neurons. They therefore define the model and parametrize the probability distribution $p(v, h)$ given a fixed architecture, which we define to specify only the number of neurons in each layer that makes up the network. A model's parameters can be trained such that its distribution over the visible units matches a desired probability distribution $q(v)$, for example, the thermal distribution of Ising variables ($v = x$) in Equation 1. More explicitly, given data samples from $q(v)$, θ are adjusted so that

$$p(v) \equiv \sum_{\{h\}} p(v, h) \approx q(v).$$

Once training is complete, θ are fixed, and the states of both visible and hidden neurons can be importance sampled with block Gibbs sampling, as long as the graph is bipartite. Since the purpose of the model distribution is a faithful representation of the physical distribution, this allows us to calculate physical estimators from the neural network, for example, $\langle E \rangle = \|\mathcal{S}\|^{-1} \sum_{v \in \mathcal{S}} H(v)$, where $\mathcal{S}$ is a set of configurations importance sampled from the network's visible nodes. It is the convergence of these estimators—in particular, $\langle E \rangle$ and $\langle C \rangle$—to the true values obtained via Monte Carlo that we will focus on in later sections.

Training such generative models requires data, which in this paper is the binary spin configurations of an N-site 2D lattice Ising model. Training often presents practical limitations that dictate which types of stochastic neural network are effective. The major practical limitation is the problem of inferring the state of the hidden neurons given the state of the visible layer, which is often clamped to values from the data during training. Therefore, we only use models in which such inference can be done efficiently. Below we outline these models, which we compare and contrast for the task of modelling a physical distribution in Section 4.

3.1 Restricted Boltzmann Machines

When visible and hidden nodes are connected in an undirected graph by edges (representing a weight matrix), the model is called a *Boltzmann machine* (Ackley et al., 1985). The original Boltzmann machine with all-to-all connectivity suffers from intractable inference problems. When the architecture is restricted so that nodes are arranged in layers of visible and hidden units, with no connection between nodes in the same layer, this problem is alleviated. The resulting model, which is depicted in Figure 3, is known as an RBM (Smolensky, 1986). The joint probability distribution defining the RBM is

$$p(v, h) = \frac{\exp\left(b^T v + c^T h + v^T W h\right)}{Z},$$

where Z is a normalization constant or partition function, b and c are biases for the visible and hidden neurons respectively, and W is the matrix of weights associated to the connec-

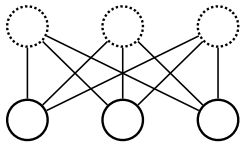


Figure 3: The restricted Boltzmann machine graph. The solid circles at bottom represent visible nodes, v. The dashed circles at top represent hidden nodes, h. Straight lines are the weights W.

tions between visible and hidden neurons. This is known as an energy-based model, because it follows the Boltzmann distribution for the "energy" function

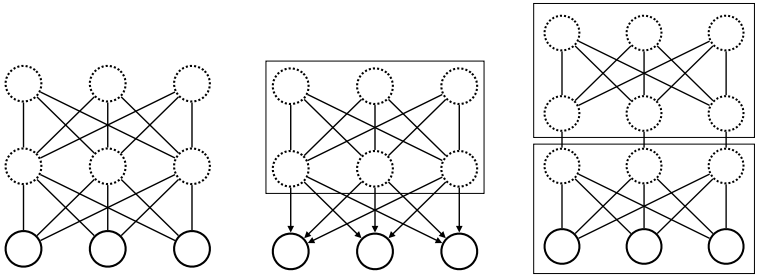
$$E_{\text{RBM}}(v, h) = -b^T v - c^T h - v^T W h.$$

In order to train the RBM (Hinton, 2012), the negative log-likelihood $\mathcal{L}$ of the training data set $\mathcal{D}$ is minimized over the space of model parameters θ, where $\mathcal{L}(\mathcal{D}) = -\frac{1}{\|\mathcal{D}\|} \sum_{v \in \mathcal{D}} \log p(v)$. In other words, the model parameters of the RBM are tuned such that the probability of generating a data set statistically similar to the training data set is maximized. This is done via stochastic gradient descent, where instead of summing over $v \in \mathcal{D}$, the negative log-likelihood of a mini-batch $\mathcal{B}$ (with $\|\mathcal{B}\| \ll \|\mathcal{D}\|$) is minimized. The mini-batch is changed every update step, and running through all $\mathcal{B} \in \mathcal{D}$ constitutes one epoch of training. The gradient of the negative log-likelihood must be calculated with respect to weights and biases, and consists of two terms as described in Appendix A; the data-dependent correlations, and the model-dependent correlations. The latter term is computed approximately by running a Markov chain through a finite number k steps starting from the mini-batch data. This approximation is called *contrastive divergence* (CD-k, Hinton, 2002) which we use to train the models in this paper. In order to use the CD-k learning algorithm, efficient block Gibbs sampling must be possible. Because of the restricted nature of the RBM, the hidden variables are conditionally independent given the visible variables, and the posterior factorizes. The state of the hidden units can then be exactly inferred from a known visible configuration in the RBM, using the conditional probability

$$p(h_i = 1|v) = \sigma\left(\left(c^T + v^T W\right)_i\right), \tag{2}$$

with a similar expression for $p(v_i = 1|h)$. For reference, we also provide the derivation of this conditional probability distribution in Appendix A.

Once trained, samples of the RBM's probability distribution can be generated by initializing v to random values, then running Gibbs sampling as a Markov Chain ($v_0 \mapsto h_0 \mapsto v_1 \mapsto h_1 \cdots$). Focussing on the visible configurations thus produced, like standard Monte Carlo after a sufficient warm-up (or equilibration) period, they will converge to importance samples of $p(v)$, allowing for the approximation of physical estimators, such as $\langle E \rangle$ or $\langle C \rangle$. Before showing results for these, we further discuss deep generalizations of the RBM in the next sections.



(a) DBM (b) DBN (c) DRBN

Figure 4: The deep generalizations of the restricted Boltzmann machine. Dashed circles represent hidden nodes, with h_1 in the layer above v, h_2 in the layer above h_1, and so on. Directed (undirected) edges depict a one-way (two-way) dependence of the neurons they connect. A closed rectangle indicates a constituent restricted Boltzmann machine.

3.2 Deep Boltzmann Machine

A *deep Boltzmann machine* (DBM; Salakhutdinov and Hinton, 2009) has one visible and two or more hidden layers of neurons, which we label $v, h_1, h_2, \cdots$ (Figure 4). It is a deep generalization of the restricted Boltzmann machine, and is similarly an energy-based model, with the energy for the two-hidden-layer case being

$$E_{\text{DBM}}(v, h_1, h_2) \quad = \quad -b^T v - c^T h_1 - d^T h_2 - v^T W_0 h_1 - h_1^T W_1 h_2.$$

Sampling $p(h_1, h_2|v)$ cannot be done exactly as in the case of the RBM. This is because the conditional probabilities of the neurons are

$$\begin{aligned} p(v_i = 1|h_1) &= \sigma\left(\left(b^T + W_0 h_1\right)_i\right) \\ p((h_1)_i = 1|v, h_2) &= \sigma\left(\left(c^T + v^T W_0 + W_1 h_2\right)_i\right) \\ p((h_2)_i = 1|h_1) &= \sigma\left(\left(d^T + h_1^T W_1\right)_i\right). \end{aligned}$$

However, notice that the even and odd layers are conditionally independent. That is, for a three layer DBM, h_1 may be sampled given knowledge of $\{v, h_2\}$, and $\{v, h_2\}$ may be sampled given the state of h_1. Therefore, in order to approximately sample $p(h_1, h_2|v_0)$ for some v_0, successive Gibbs sampling of even and odd layers with the visible neurons clamped to v_0 is performed. Under such conditions, the state of the hidden neurons will equilibrate to samples of $p(h_1, h_2|v_0)$. This is how the inference $v_0 \mapsto h_1, h_2$ is performed in our training procedure. In order to generate model-dependent correlations, the visible neurons are unclamped and Gibbs sampling proceeds for k steps. The parameter updates for the DBM are similar to the CD-k algorithm for the RBM described in Section 3.1 and explicitly given in Appendix A (Salakhutdinov and Hinton, 2009).

Pre-training the DBM provides a reasonable initialization of the weights and biases by training a stack of RBM's and feeding their weights and biases into the DBM. Our pre-training procedure was inspired by Hinton and Salakhutdinov (2012), however it varies slightly from their approach because the two-layer networks used in our numerical results below are small enough to be trained fully without ideal pre-training. Our procedure goes as follows for a two-layer DBM. First, an RBM is trained on the training data set $\mathcal{D}$, and is then used to infer a hidden data set $\mathcal{H}$ via sampling the distribution $p(h|v \in \mathcal{D})$. A second RBM is then trained on $\mathcal{H}$. Weights from the first and second RBMs initialize the weights in the first and second layers of the DBM respectively. DBM biases are similarly initialized from the RBM stack, however in the case where the biases in the first hidden layer of the DBM can be taken from either the hidden layer in the first RBM, or the visible layer in the second RBM, these two options are averaged. We emphasize that this procedure is not ideal for DBMs with more than three layers.

Once fully trained, the probability distribution associated to the DBM can be sampled, similarly to the RBM, by initializing all neurons randomly, then running the Markov chain even layers $\mapsto$ odd layers $\mapsto$ even layers $\cdots$ etc. until the samples generated on the visible neurons are uncorrelated with the initial conditions of the Markov chain.

3.3 Deep Belief Network

The *deep belief network* (DBN) of Figure 4 is another deep generalization of the RBM. It is not a true energy-based model, and only the deepest layer of connections are undirected. The deepest two hidden layers of the DBN form an RBM, with the layers connecting the hidden RBM to the visible units forming a feed forward neural network. The DBN is trained by first performing layer-wise training of a stack of RBMs as detailed by Hinton et al. (2006), and similarly to the aforementioned pre-training procedure in Section 3.2. The model parameters of the deepest hidden RBM are used to initialize the deepest two layers of the DBN. The remaining layers of the DBN take their weights and biases from the weights and visible biases of the corresponding RBMs in the pre-training stack. Note, there exists a fine tuning procedure called *the wake-sleep algorithm* (Hinton et al., 1995), which further adjusts the model parameters of the DBN in a non-layer-wise fashion. However, performing fine tuning of the DBN was found to have no influence on the results described below.

Once trained, a DBN can be used to generate samples by initializing the hidden RBM randomly, running Gibbs sampling in the hidden RBM until convergence to the model distribution is achieved, then propagating the configuration of the RBM neurons downwards to the visible neurons of the DBN using the same conditional probabilities used in an RBM, Equation 2.

3.4 Deep Restricted Boltzmann Network

The *deep restricted Boltzmann network* (DRBN) of Figure 4 is a simple, deep generalization of the RBM which, like the DBN, is also not a true energy-based model (Hu et al., 2016).

Inference in the DRBN is exactly the same as in the RBM, but with many layers,

$$\begin{aligned} p((x_l)_i|x_{l-1}) &= \sigma\left(\left(b_l^T + x_{l-1}^T W_{l-1}\right)_i\right) \\ p((x_{l-1})_i|x_l) &= \sigma\left(\left(b_{l-1}^T + W_{l-1}x_l\right)_i\right), \end{aligned}$$

where $x_0 \equiv v$, $x_1 \equiv h_1$ etc. However, Gibbs sampling in the DRBN is performed by doing a complete up (down) pass, where the state of neurons in all layers of the DRBN are inferred sequentially given the state of the bottom (top) neurons. The CD-k algorithm is applied to train the DRBN, as it was for the RBM, by first inferring data-dependent correlations from some initial data v_0, then performing k steps of Gibbs sampling in order to generate the model-dependent correlations.

Once trained, the DRBN is sampled by initializing the visible neurons randomly, then performing Gibbs sampling until the visible samples converge to samples of the model dependent distribution $p(v)$.

4. Training and Results

In this section, we investigate the ability of the generative graphical models, introduced in the previous section, to represent classical statistical mechanical probability distributions, with a particular interest in the comparison between deep and shallow models near the phase transition of the two-dimensional Ising system.

In order to produce training data, we use standard Markov chain Monte Carlo techniques with a combination of single-site Metropolis and Wolff cluster updates, where one Monte Carlo step consists of N single-site updates and one cluster update, and N is the number of sites on the lattice. Importance sampling thus obtained 10^5 independent spin configurations for each $T \in [1.0, 3.5]$ in steps of $\Delta T = 0.1$. An equilibration time of N^3 Monte Carlo steps and a decorrelation time of N steps were used, where in this paper we concentrate on $N = 64$ only.

Using these data sets, we trained shallow (RBM) and deep (DBM, DBN, DRBN) generative models to learn the underlying probability distribution of the Monte Carlo data for each value of T. Similar to the work done by Torlai and Melko (2016), restricted Boltzmann machines of architecture (N, N_{h_1}) for $N = 64$ and various N_{h_1} were trained. Further, we trained two-hidden-layer deep models of architecture (N, N_{h_1}, N_{h_2}) in order to investigate the efficiency with which deep architectures can represent the thermal distribution of Ising systems. Networks with two hidden layers were chosen to reduce the large parameter space to be explored, while still allowing us to quantify the difference between shallow and deep models. Training hyper-parameters for each model are given in Table 1, and values of N_{h_1} and N_{h_2} used in this work were $\{8, 16, 24, 32, 40, 48, 56, 64\}$.

In order to quantify how well the models captured the target physical probability distribution, each model was used to generate a set $\mathcal{S}$ of 10^4 spin configurations on which the observables $\langle E\rangle$ and $\langle C\rangle$ were computed and compared to the exact values from Monte Carlo simulations.

We begin by benchmarking our calculations on the case of the shallow RBM, studied previously by Torlai and Melko (2016). Similar to the results reported in that work, we find that for all temperatures an RBM with $N_{h_1} = N$ accurately captures the underlying Ising model distribution, a result consistent with the upper bound of $N_{h_1} \leq 2N$ found by Chen

hyperparameter	RBM	DBM	DBN	DRBN
k	10	10	10	5
equilibration steps	NA	10	NA	NA
training epochs	4×10^3	3×10^3	3×10^3	3×10^3
learning rate	5×10^{-3}	10^{-3}	10^{-4}	10^{-4}
mini-batch size	10^2	10^2	10^2	10^2

Table 1: Values of training hyper-parameters for each model, trained with contrastive divergence CD-k.

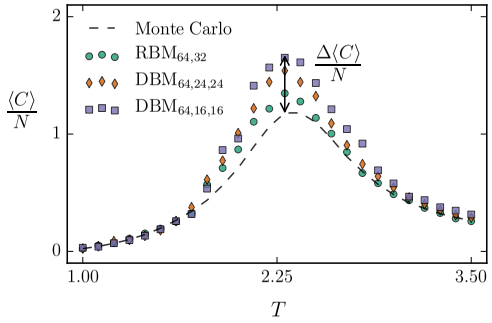


Figure 5: Measurements of heat capacity per Ising spin on samples generated by trained shallow and deep models of architecture $(64, 32)$, $(64, 16, 16)$, and $(64, 24, 24)$. Measurements are compared to Monte Carlo values of the data on which the models were trained.

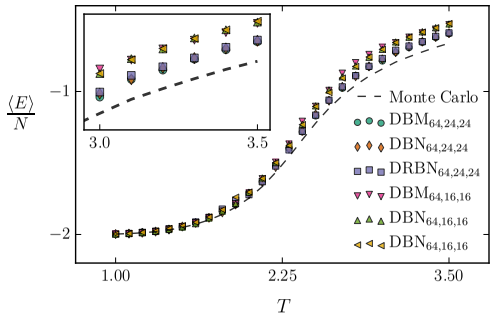


Figure 6: Measurements of energy E per Ising spin on samples generated by trained shallow and deep models of various architectures. Measurements are compared to Monte Carlo values of the data on which the models were trained.

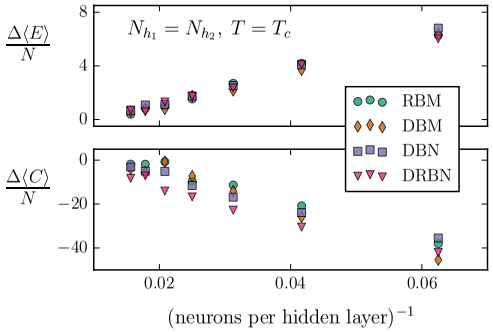


Figure 7: The error in reproducing the energy and heat capacity per Ising spin of the training data as a function of the number of hidden neurons per layer for all model types.

et al. (2017) for the 2D Ising model. RBMs with less hidden neurons fail to completely capture the observables of the data, particularly near the critical region defined by $T \approx T_c$. Furthermore, we find here that all deep models with architecture $N_{h_1} = N_{h_2} = N$ can also accurately represent the probability distribution generated by the $N = 64$ Ising Hamiltonian at any temperature. Below we concentrate on $N_{h_1}, N_{h_2} < N$ in order to systematically investigate and compare the representational power of each model type.

Figure 5 shows a heat capacity calculated from the spin configurations on the visible nodes. As an illustrative example, we present a shallow RBM with 32 hidden units, compared to two different DBMs each with two hidden layers. To reduce parameter space, we let $N_{h_1} = N_{h_2}$ for these deep models. From this figure it is clear that the DBM with the same number of total hidden neurons ($\mathrm{DBM}_{64,16,16}$) performs worse than the DBM with approximately the same total number of model parameters ($\mathrm{DBM}_{64,24,24}$) as the RBM. Remarkably, both deep networks are outperformed by the shallow RBM, suggesting it is more efficient to put additional network resources into the first hidden layer, than to add more layers in a deep architecture.

In Figure 6 we examine $\langle E\rangle$ for deep models with two different architectures, namely $N_{h_1} = N_{h_2} = 16$ and $N_{h_1} = N_{h_2} = 24$. In this plot, two families of curves become apparent, associated with these two different network architectures. From this, it appears that between all the different deep models, the differences in performance correspond only to the number and arrangement of hidden nodes (i.e. the "architecture" of the network), and not to the specific model type and training procedure, which varies considerably between DBM, DBN, and DRBN.

To further investigate this point, we next compare RBMs with deep models having the same number of hidden units in each hidden layer ($N_{h_1} = N_{h_2}$) as the RBMs do in h_1. Figure 7 compares the percent error at $T = T_c$ in reproducing the physical observables

energy and heat capacity. In this figure, we see the expected monotonic trends of both $\Delta\langle E\rangle$ and $\Delta\langle C\rangle$, which are defined as the difference between the generated quantity and its exact value at the critical point (see Figure 5). This demonstrates that, for all models, the accuracy of reconstructing physical observables increases with the number of neurons per hidden layer. More surprisingly however, since values of $\Delta\langle E\rangle$ and $\Delta\langle C\rangle$ are not significantly different across model types with equal architecture, it can also be said that adding a second hidden layer to an RBM does not improve its ability to generate accurate observables. This is the case even when the deep networks have greater total network resources (neurons or model parameters) than the shallow RBMs, as they do in Figure 7. This demonstrates again that the most efficient way to increase representational power in this case is not to add a second hidden layer, thereby increasing the depth of the neural network, but to increase the size of the only hidden layer in a shallow network.

We confirm this in a striking way in Figure 8, where we plot both $\langle E\rangle$ and $\langle C\rangle$ at the critical temperature for samples generated from deep models of architecture $(64, 24, N_{h_2})$, in comparison to an RBM with $N_{h_1} = 24$ and the exact Monte Carlo values. The fact that there is no clear systematic improvement in accuracy as N_{h_2} increases confirms that the accuracy of deep neural networks for the Ising system is essentially independent of the second hidden layer.

5. Conclusion

We have investigated the use of generative neural networks as an approximate representation of the many-body probability distribution of the two-dimensional Ising system, using the restricted Boltzmann machine and its deep generalizations; the deep Boltzmann machine, deep belief network, and deep restricted Boltzmann network. We find that it is

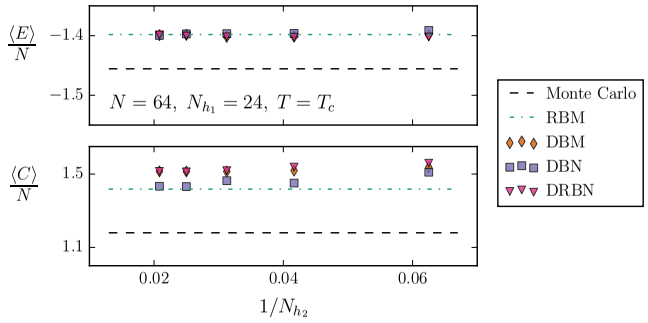


Figure 8: Energy E and heat capacity C per Ising spin at the critical temperature as measured on samples generated from deep models with various number of neurons in the second hidden layer. N_{h_1} is fixed, and model values are compared to Monte Carlo and an RBM, which serves as the $N_{h_2} = 0$ reference point.

possible to accurately represent the physical probability distribution, as determined by reproducing physical observables such as energy or heat capacity, with both shallow and deep generative neural networks. However we find that the most efficient representation of the Ising system—defined to be the most accurate model given some fixed amount of network resources—is given by a shallow RBM, with only one hidden layer.

In addition, we have also found that the performance of a deep generative network is independent of what specific model was used. Rather, the accuracy in reproducing observables is entirely dependent on the network architecture, and even more remarkably, on the number of units in the first hidden layer only. This result is consistent with that of Raghu et al. (2017), who find that trained networks are most sensitive to changes in the weights of the first hidden layer. Additionally, in the wake of recent developments in information based deep learning by Tishby and Zaslavsky (2015), we remark that such behavior may be a signature of the information bottleneck in neural networks.

It is interesting to view our results in the context of the empirically well-supported fact that deep architectures perform better for supervised learning of complex, hierarchal data (LeCun et al., 2015). In statistical mechanical systems there is often a very simple pattern of local, translation-invariant couplings between degrees of freedom. While there are multi-scale techniques in physics such as the renormalization group for analyzing physical systems, no obvious hierarchy of features exists to be leveraged by generative, deep learning algorithms. While it can be speculated that the power of deep networks also translates to the realm of unsupervised, generative models, for natural data sets commonly used by the machine learning community it is difficult to obtain quantitative measures of accuracy (Hinton et al., 2006; Le Roux and Bengio, 2008; Salakhutdinov and Hinton, 2009; Hu et al., 2016). Therefore, modelling physics data offers an opportunity to quantitatively measure

and compare the performance of deep and shallow models, due to the existence of physical observables such as energy and heat capacity.

However, considering our results, it is clear that there may be a fundamental difference between statistical physics data and real-world data common to machine learning practices, otherwise the established success of deep learning would more easily translate to applications of representing physics systems, such as thermal Ising distributions near criticality. It is interesting to ask what precisely this difference is.

The lines of inquiry addressed in this paper could also be continued by extending this analysis to consider the efficient representation of quantum many-body wave functions by deep neural networks (Torlai et al., 2017), and by investigating in more detail the nature of information flow in neural-network representations of physical probability distributions.

Acknowledgments

We would like to thank J. Carrasquilla, G. Torlai, B. Kulchytskyy, L. Hayward Sierens, P. Ponte, M. Beach, and A. Golubeva for many useful discussions. This research was supported by Natural Sciences and Engineering Research Council of Canada, the Canada Research Chair program, and the Perimeter Institute for Theoretical Physics. Simulations were performed on resources provided by the Shared Hierarchical Academic Research Computing Network (SHARCNET). Research at Perimeter Institute is supported through Industry Canada and by the Province of Ontario through the Ministry of Research and Innovation.

Appendix A.

The gradient of the negative log-likelihood must be calculated with respect to weights and biases; for example, $\nabla_W(\mathcal{L}) = \langle vh^T \rangle_{p(h|v)} - \langle vh^T \rangle_{p(v,h)}$, where $\langle f \rangle_{P(x)}$ denotes the expectation value of f with respect to the probability density $P(x)$. The first term in the gradient is referred to as the data-dependent correlations, and is tractable as long as sampling $p(h|v)$ is efficient, which it is for RBMs. However, the second term, referred to as model-dependent correlations, is ideally computed by running the Markov chain $v_0 \mapsto h_0 \mapsto v_1 \mapsto h_1 \cdots$ an infinite number of steps in order to sample the model distribution without bias due to the initial conditions v_0. This cannot be done exactly, and the standard approximation to estimate the model-dependent term is to run the Markov chain for only $k < \infty$ steps from an initial value of v_0 given by the mini-batch data—in practice, k of order 1 may do. For concreteness, the full update rules for the weights and biases of an RBM are

then given by $\theta \mapsto \theta + \eta\Delta\theta$, where

$$\Delta W = \|\mathcal{B}\|^{-1} \sum_{v_0 \in \mathcal{B}} v_0 h_0^T - v_k h_k^T \quad \text{(A.1)}$$

$$\Delta b = \|\mathcal{B}\|^{-1} \sum_{v_0 \in \mathcal{B}} v_0 - v_k \quad \text{(A.2)}$$

$$\Delta c = \|\mathcal{B}\|^{-1} \sum_{v_0 \in \mathcal{B}} h_0 - h_k, \quad \text{(A.3)}$$

and η is a hyper-parameter of the learning algorithm called the *learning rate*.

In order to evaluate the parameter updates in Equations A.1 to A.3, conditional probabilities in an RBM such as $p(h_i = 1|v)$ must be known. Therefore, here we provide a derivation of the conditional probability distribution $p(h_i = 1|v) = \sigma\left(\left(c^T + v^T W\right)_i\right)$ of an RBM.

Firstly, the marginal distribution $p(v)$ is given by

$$\begin{aligned} p(v) &= \sum_h p(v,h) \\ &= \frac{1}{Z} \exp\left(b^T v + \sum_i \log\left(1 + \exp\left(\left(c^T + v^T W\right)_i\right)\right)\right) \\ &\equiv \frac{1}{Z} \exp(-\mathcal{F}(v)), \end{aligned}$$

where $\mathcal{F}(v)$ is known as the "free energy". One can interpret training an RBM as fitting $\mathcal{F}(v)$ to the Hamiltonian $H(v)$ of the training data set, sampled from Equation 1, a perspective taken by previous studies in physics (see Huang and Wang, 2017). Next, using Bayes' theorem, $p(h|v) = \frac{1}{Z'} \prod_i \exp\left((c^T + v^T W)_i h_i\right)$, meaning that the conditional probability factors, $p(h|v) = \prod_i p(h_i|v)$. Finally, by requiring normalization, the conditional activation probabilities are

$$p(h_i = 1|v) = \sigma\left(\left(c^T + v^T W\right)_i\right)$$

where $\sigma(x) = (1+\exp(-x))^{-1}$. The activation probabilities $p(v_i = 1|h)$ are derived similarly.

References

David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985.

Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355, 2017.

Juan Carrasquilla and Roger G. Melko. Machine learning phases of matter. *Nat Phys*, advance online publication, 02 2017.

Jing Chen, Song Cheng, Haidong Xie, Lei Wang, and Tao Xiang. On the equivalence of restricted boltzmann machines and tensor network states. *arXiv:1701.04831*, January 2017.

Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Comput.*, 14(8):1771–1800, August 2002.

Geoffrey E. Hinton. *A Practical Guide to Training Restricted Boltzmann Machines*, pages 599–619. Springer, Berlin, Heidelberg, 2012.

Geoffrey E. Hinton and Ruslan R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, July 2006.

Geoffrey E Hinton and Ruslan R Salakhutdinov. A better way to pretrain deep boltzmann machines. In *Advances in Neural Information Processing Systems 25*, pages 2447–2455. Curran Associates, Inc., 2012.

Geoffrey E. Hinton, Peter Dayan, Brendan J. Frey, and Radford M. Neal. The "wake-sleep" algorithm for unsupervised neural networks. *Science*, 268:1158–1161, 1995.

Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, July 2006.

Hengyuan Hu, Lisheng Gao, and Quanbin Ma. Deep restricted boltzmann networks. *CoRR:1611.07917*, 2016.

Li Huang and Lei Wang. Accelerated monte carlo simulations with restricted boltzmann machines. *Phys. Rev. B*, 95:035105, Jan 2017.

Maciej Koch-Janusz and Zohar Ringel. Mutual Information, Neural Networks and the Renormalization Group. *arXiv:1704.06279*, April 2017.

Hendrick A. Kramers and Gregory H. Wanniers. Statistics of two-dimensional ferromagnet. *Physical Review*, 60:252–262, 1941.

Nicolas Le Roux and Yoshua Bengio. Representational power of restricted boltzmann machines and deep belief networks. *Neural Comput.*, 20(6):1631–1649, June 2008.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, 05 2015.

Pankaj Mehta and David J. Schwab. An exact mapping between the variational renormalization group and deep learning. *arXiv:1410.3831*, 2014.

Guido Montúfar and Jason Morton. Discrete restricted boltzmann machines. *JMLR*, 16: 653–672, 2015.

Mark E.J. Newman and Gerard T. Barkema. *Monte Carlo Methods in Statistical Physics*. Clarendon Press, 1999.

Lars Onsager. Crystal statistics. i. a two-dimensional model with an order-disorder transition. *Physical Review, Series II*, 65:117–149, 1944.

Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 2847–2854, Sydney, Australia, 06–11 Aug 2017. PMLR.

Ruslan R. Salakhutdinov and Geoffrey E. Hinton. Deep boltzmann machines. In *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5, pages 448–455, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR.

Paul Smolensky. Information Processing in Dynamical Systems: Foundations of Harmony Theory ; CU-CS-321-86. *Computer Science Technical Reports*, 315, 1986.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. *CoRR:1503.02406*, 2015.

Giacomo Torlai and Roger G. Melko. Learning thermodynamics with boltzmann machines. *Phys. Rev. B*, 94:165134, Oct 2016.

Giacomo Torlai, Guglielmo Mazzola, Juan Carrasquilla, Matthias Troyer, Roger Melko, and Giuseppe Carleo. Many-body quantum state tomography with neural networks. *arXiv:1703.05334*, 2017.

Sebastian J. Wetzel. Unsupervised learning of phase transitions: from principal component analysis to variational autoencoders. *arXiv:1703.02435*, 2017.

Kenneth G. Wilson and Michael E. Fisher. Critical exponents in 3.99 dimensions. *Phys. Rev. Lett.*, 28:240–243, Jan 1972.

Journal of Machine Learning Research 18 (2018) 1-6 Submitted 10/17; Revised 2/18; Published 4/18

pomegranate: Fast and Flexible Probabilistic Modeling in Python

Jacob Schreiber jmschr@cs.washington.edu
Paul G. Allen School of Computer Science
University of Washington
Zheng-Chu Seattle, WA 98195-4322, USA

Editor: Balazs Kegl

Abstract

We present pomegranate, an open source machine learning package for probabilistic modeling in Python. Probabilistic modeling encompasses a wide range of methods that explicitly describe uncertainty using probability distributions. Three widely used probabilistic models implemented in pomegranate are general mixture models, hidden Markov models, and Bayesian networks. A primary focus of pomegranate is to abstract away the complexities of training models from their definition. This allows users to focus on specifying the correct model for their application instead of being limited by their understanding of the underlying algorithms. An aspect of this focus involves the collection of additive sufficient statistics from data sets as a strategy for training models. This approach trivially enables many useful learning strategies, such as out-of-core learning, minibatch learning, and semi-supervised learning, without requiring the user to consider how to partition data or modify the algorithms to handle these tasks themselves. pomegranate is written in Cython to speed up calculations and releases the global interpreter lock to allow for built-in multithreaded parallelism, making it competitive with—or outperform—other implementations of similar algorithms. This paper presents an overview of the design choices in pomegranate, and how they have enabled complex features to be supported by simple code. The code is available at `https://github.com/jmschrei/pomegranate`

Keywords: probabilistic modeling, Python, Cython, machine learning, big data.

1. Introduction

The Python ecosystem is becoming increasingly popular for the processing and analysis of data. This popularity is in part due to easy-to-use libraries such as numpy (van der Walt et al., 2011), scipy (Jones et al., 2001), and matplotlib (Hunter, 2007) that aim to provide fast general purpose functionality. However, equally important are the libraries that are built on top of these to provide higher level functionality, such as pandas (McKinney, 2010) for data analysis, scikit-image (van der Walt et al., 2014) for computer vision, Theano (Theano Development Team, 2016) for efficient evaluation of mathematical expressions, gensim (Řehůřek and Sojka, 2010) for topic modeling in natural language processing, and countless others. Naturally, many machine learning packages have also been developed for Python, including those that implement classic machine learning algorithms, such as scikit-learn (Pedregosa et al., 2011), mlpy (Albanese et al., 2012), shogun (Sonnenburg et al., 2017), and xgboost (Chen and Guestrin, 2016).

pomegranate fills a gap in the Python ecosystem that encompasses building probabilistic machine learning models that utilize maximum likelihood estimates for parameter updates. There are several packages that implement certain probabilistic models in this style individually, such as hmmlearn for hidden Markov models, libpgm for Bayesian networks, and scikit-learn for Gaussian mixture models and naive Bayes models. However, pomegranate implements a wider range of probabilistic models and does so in a more modular fashion than these other packages, having two main effects. The first is that the addition of a new probability distribution in pomegranate allows for all models to be built using that distribution immediately. The second is that improvements to one aspect of pomegranate immediately propagate to all models that would use that aspect. For example, when GPU support was added to multivariate Gaussian distributions, this immediately meant that all models with multivariate Gaussian emissions could be GPU accelerated without any additional code. pomegranate currently includes a library of basic probability distributions, naive Bayes classifiers, Bayes classifiers, general mixture models, hidden Markov models, Bayesian networks, Markov chains, as well as implementations of factor graphs and k-means++/|| that can be used individually but primarily serve as helpers to the primary models.

There are several already existing Python libraries that implement Bayesian methods for probabilistic modeling. These include, but are not limited to, PyMC3 (Salvatier et al., 2016), PyStan (Stan Development Team, 2016), Edward (Tran et al., 2016), pyro (Inc., 2017), and emcee (Foreman-Mackey et al., 2013). Bayesian approaches typically represent each model parameter as its own probability distribution, inherently capturing the uncertainty in that parameter, whereas maximum likelihood approaches typically represent each model parameter as a single value. An example of this distinction is that a mixture model can either be represented as a set of probability distributions and a vector of prior probabilities, or as a set of probability distributions that themselves have probability distributions over their respective parameters (such as the mean and standard deviation, should these distributions be normal distributions) and as a dirichlet distribution representing the prior probabilities. The first representation typically specifies models that are faster to both train and perform inference with, while the second is illustrative of the type of models one could build with packages that implement Bayesian methods, such as PyMC3. Both representations have strengths and weaknesses, but pomegranate implements models falling solely in the first representation.

pomegranate was designed to be easy to use while not sacrificing on computational efficiency. Models can either be specified by writing out each of the components individually if known beforehand, or learned directly from data if not. Key features, such as out-of-core learning and parallelization, can be toggled for each model independently of the definition or method calls, typically by simply passing in an optional parameter. The core computational bottlenecks are written in Cython and release the global interpreter lock (GIL), enabling multi-threaded parallelism that typically Python modules cannot take advantage of. Lastly, linear algebra operations such as matrix-matrix multiplications are implemented using BLAS with the ability to toggle a GPU if present.

All comparisons were run on a computational server with 24 Intel Xeon CPU E5-2650 cores with a clock speed of 2.2 GHz, a Tesla K40c GPU, and 256 GB of RAM running CentOS 6.9. The software used was pomegranate v0.8.1 and scikit-learn v0.19.0. pomegranate can be installed using `pip install pomegranate` or `conda install pomegranate` on all platforms. Pre-built wheels are available for Windows builds, removing the sometimes difficult requirement of a working compiler.

2. The API

pomegranate provides a simple and consistent API for all implemented models that mirrors the scikit-learn API as closely as possible. The most important methods are `fit`, `from_samples`, `predict` and `probability`. The `fit` method will use the given data and optional weights to update the parameters of an already initialized model, using either maximum-likelihood estimates (MLE) or expectation-maximization (EM) as appropriate. In contrast, the `from_samples` method will create a model directly from data in a manner similar to scikit-learn's `fit` method. For simple models like single distributions this corresponds only to MLE on the input data, but for most other models this corresponds to an initialization step plus a call to `fit`. This initialization can range from using k-means for mixture models to structure learning for Bayesian networks. The `predict` method returns the posterior estimate $argmax_M P(M|D)$, identifying the most likely component of

the model for each sample. The `probability` method returns the likelihood of the data given the model $P(D|M)$. The other methods include `predict_proba` which returns the probability of each component for each sample $P(M|D)$, `predict_log_proba` which returns the log of the previous value, and `summarize` and `from_summaries` that jointly implement the learning strategies detailed below.

3. Key Features

pomegranate supports many learning strategies that can be employed during training, including out-of-core learning for massive data sets, semi-supervised learning for data sets with a mixture of labeled and unlabeled data, and minibatch learning. In addition, one can employ multithreaded parallelism or a GPU for data-parallel speedups. These features are made possible by separating out the collection of sufficient statistics from a data set (using the `summarize` method) from the actual parameter update step (using the `from_summaries` method).

Sufficient statistics are the smallest set of numbers needed to calculate some statistic on a data set. As an example, fitting a normal distribution to data involves the calculation of the mean and the variance. The sufficient statistics for the mean and the variance are the sum of the weights of the points seen so far $\left(\sum_{i=1}^{n} w_i\right)$, the sum of the weighted samples $\left(\sum_{i=1}^{n} w_i X_i\right)$, and the sum of the weighted samples squared $\left(\sum_{i=1}^{n} w_i X_i^2\right)$. The mean and variance can then be directly calculated from these three numbers using the following two equations:

$$\mu = \frac{\sum_{i=1}^{n} w_i X_i}{\sum_{i=1}^{n} w_i} \qquad \sigma^2 = \frac{\sum_{i=1}^{n} w_i X_i^2}{\sum_{i=1}^{n} w_i} - \left(\frac{\sum_{i=1}^{n} w_i X_i}{\sum_{i=1}^{n} w_i}\right)^2 \tag{1}$$

Out-of-core Learning: The additive nature of the sufficient statistics means that if one were to summarize two batches of data successively and then add the sufficient statistics together, they would get the same sufficient statistics as if they were calculated from the full data set. This presents an intuitive way to handle data sets that are too large to fit in memory, by chunking the data set into batches that do fit in memory and summarizing them successively, adding the calculated sufficient statistics together afterwards. This can be done by passing in a `batch_size` parameter to your training method, for example `model.fit(X, batch_size=10000)` would train a pre-initialized model on more data than can fit in memory by successively summarizing batches of size 10,000 until the full data set has been seen. The `summarize` and `from_summaries` methods can also be used independently to implement custom out-of-core strategies.

Minibatch Learning: A natural extension of the out-of-core strategy is minibatch learning, where a parameter update is done after one or a few batches, instead of the full data set. This is in contrast to batch methods that calculate an update using the entire data set, and stochastic methods that typically update using only a single sample. Minibatching can be specified by passing values to both `batch_size` and `batches_per_epoch` parameters when using `fit` or `from_summaries`, where the `batches_per_epoch` is the number of batches to consider before making an update.

Semi-supervised Learning: Semi-supervised learning is the task of fitting a model to a mixture of both labeled and unlabeled data. Typically this arises in situations where labeled data is sparse, but unlabeled data is plentiful, and one would like to make use of both to learn an informed model.

pomegranate supports semi-supervised learning for `HiddenMarkovModel`, `BayesClassifier`, and `NaiveBayes` models as a combination of EM and MLE. Models are initialized using MLE on the labeled data. Next, a version of EM is used that combines the sufficient statistics calculated from the labeled data using MLE with the sufficient statistics calculated from the unlabeled data using EM at each iteration until convergence. This is automatically toggled whenever -1 is present in the label set, following scikit-learn conventions.

This EM-based approach compares favorably to scikit-learn. To demonstrate, we generate a data set of 100k samples in 10 dimensions from 2 overlapping Gaussian ellipses with means of 0 and 1 respectively and standard deviations of 2. It took pomegranate ∼0.04s to learn a Gaussian naive Bayes model with 10 iterations of EM, ∼0.2s to learn a multivariate Gaussian Bayes classifier with a full covariance matrix with 10 iterations of EM, whereas the scikit-learn label propagation model with a RBF kernel did not converge after ∼220s and 1000 iterations, and took ∼2s with a knn kernel with 7 neighbors. Both pomegranate models achieved validation accuracies over 0.75, whereas the scikit-learn models did no better than chance.

Parallelism: Another benefit of the use of additive sufficient statistics is that it presents a clear data-parallel way to parallelize model fitting. Simply, one would divide the data into several batches and calculate the sufficient statistics for each batch locally. These sufficient statistics can then be added together back on the main job and all parameters updated accordingly. This is implemented by dividing the data into batches and running `summarize` on each of them using separate threads and then running `from_summaries` after all threads finish. Typically, the global interpreter lock (GIL) in Python prevents multiple threads from running in parallel in the same python process. However, since the computationally intensive aspects are written in Cython the GIL can be released, allowing for multiple threads to run at once. On a synthetic data set with 3M samples with 1K dimensions it takes ∼65 seconds to train a Gaussian naive Bayes classifier using pomegranate with 1 thread, but only ∼17 seconds with 8 threads. For comparison, it takes ∼53 seconds to train a Gaussian naive Bayes classifier using scikit-learn. On another synthetic data set with 2M samples and 150 dimensions it takes pomegranate ∼470s to learn a Gaussian mixture model with a full covariance matrix with 1 thread, ∼135s with 4 threads, ∼57s with 16 threads, and ∼200s using a GPU. Lastly, we compared the speed at which pomegranate and hmmlearn could train a 10 state dense Gaussian hidden Markov model with diagonal covariance matrices. On a synthetic data set of 100 sequences, each containing 1,000 10 dimensional observations, it took hmmlearn ∼25s to run five iterations of Baum-Welch training, while it only took pomegranate ∼13s with 1 thread, ∼4s with 4 threads, and ∼2s with 16 threads.

4. Discussion

pomegranate aims to fill a niche in the Python ecosystem that exists between classic machine learning methods and Bayesian methods by serving as an implementation of flexible probabilistic models. The design choices that were made early on while building pomegranate allowed for a great number of useful features to be added later on without significant effort.

A clear area of improvement in the future is the handling of missing values, because many probabilistic models can intuitively modify the EM algorithm to infer these missing values. For example, when trying to learn a Bayesian network over a data set with missing values, one can identify the best structure over the incomplete data set, infer the missing values, and relearn the structure, iterating until convergence. Given the prevalence of missing data in the real world, extending pomegranate to handle missing data efficiently is a priority.

Acknowledgments

We would like to first acknowledge all of the contributors and users of pomegranate, whom without this project would not be possible. We would also like to acknowledge Adam Novak, who wrote the first iteration of the hidden Markov model code. Lastly, we would also like to acknowledge Dr. William Noble for suggestions and guidance during development. This work was partially supported by NSF IGERT grant DGE-1258485.

References

Davide Albanese, Roberto Visintainer, Stefano Merler, Samantha Riccadonna, Giuseppe Jurman, and Cesare Furlanello. mlpy: Machine learning python, 2012.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016. URL `http://arxiv.org/abs/1603.02754`.

D. Foreman-Mackey, D. W. Hogg, D. Lang, and J. Goodman. emcee: The mcmc hammer. *PASP*, 125:306–312, 2013. doi: 10.1086/670067.

John D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science and Engineering*, pages 90–95, 2007.

Uber Technologies Inc. pyro. https://github.com/uber/pyro, 2017.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001. URL `http://www.scipy.org/`.

Wes McKinney. Data structures for statistical computing in python. *Proceedings of the 9th Python in Science Conference*, pages 51–56, 2010.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Olivier Grisel Bertrand Thirion, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesna. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, pages 2825–2830, 2011.

Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

John Salvatier, Thomas V. Wiecki, and Christopher Fonnesbeck. Probabilistic programming in python using pymc3. *PeerJ Computer Science*, 2:e55, April 2016. ISSN 2376-5992. doi: 10.7717/peerj-cs.55. URL `https://doi.org/10.7717/peerj-cs.55`.

Soeren Sonnenburg, Heiko Strathmann, Sergey Lisitsyn, Viktor Gal, Fernando J. Iglesias García, Wu Lin, Soumyajit De, Chiyuan Zhang, frx, tklein23, Evgeniy Andreev, JonasBehr, sploving, Parijat Mazumdar, Christian Widmer, Pan Deng / Zora, Saurabh Mahindre, Abhijeet Kislay, Kevin Hughes, Roman Votyakov, khalednasr, Sanuj Sharma, Alesis Novik, Abinash Panda, Evangelos Anagnostopoulos, Liang Pang, Alex Binder, serialhex, Esben Sørig, and Björn Esser. shogun-toolbox/shogun: Shogun 6.0.0 - Baba Nobuharu, April 2017. URL `https://doi.org/10.5281/zenodo.556748`.

Stan Development Team. Pystan: the python interface to stan, 2016.

Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016. URL `http://arxiv.org/abs/1605.02688`.

Dustin Tran, Alp Kucukelbir, Adji B. Dieng, Maja Rudolph, Dawen Liang, and David M. Blei. Edward: A library for probabilistic modeling, inference, and criticism. *arXiv preprint arXiv:1610.09787*, 2016.

Stéfan van der Walt, Chris Colbert, and Gaël Varoquaux. The numpy array: A structure for efficient numerical computation. *Journal of Machine Learning Research*, pages 22–30, 2011.

Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: Image processing in python. *PeerJ*, 2014.

Journal of Machine Learning Research 18 (2018) 1-29 Submitted 11/17; Revised 3/18; Published 4/18

Maximum Principle Based Algorithms for Deep Learning

Qianxiao Li LIQIX@IHPC.A-STAR.EDU.SG
Institute of High Performance Computing
Agency for Science, Technology and Research
1 Fusionopolis Way, Connexis North, Singapore 138632

Long Chen XIDONGLC@PKU.EDU.CN
Peking University
Beijing, China, 100080

Cheng Tai CHENGTAI@PKU.EDU.CN
Beijing Institute of Big Data Research
and Peking University
Beijing, China, 100080

Weinan E WEINAN@MATH.PRINCETON.EDU
Princeton University
Princeton, NJ 08544, USA,
Beijing Institute of Big Data Research
and Peking University
Beijing, China, 100080

Editor: Yoshua Bengio

Abstract

The continuous dynamical system approach to deep learning is explored in order to devise alternative frameworks for training algorithms. Training is recast as a control problem and this allows us to formulate necessary optimality conditions in continuous time using the Pontryagin's maximum principle (PMP). A modification of the method of successive approximations is then used to solve the PMP, giving rise to an alternative training algorithm for deep learning. This approach has the advantage that rigorous error estimates and convergence results can be established. We also show that it may avoid some pitfalls of gradient-based methods, such as slow convergence on flat landscapes near saddle points. Furthermore, we demonstrate that it obtains favorable initial convergence rate per-iteration, provided Hamiltonian maximization can be efficiently carried out - a step which is still in need of improvement. Overall, the approach opens up new avenues to attack problems associated with deep learning, such as trapping in slow manifolds and inapplicability of gradient-based methods for discrete trainable variables.

Keywords: deep learning, optimal control, Pontryagin's maximum principle, method of successive approximations

1. Introduction

Supervised learning using deep neural networks has become an increasingly successful tool in modern machine learning applications (Bengio, 2009; Schmidhuber, 2015; LeCun et al., 2015; Goodfellow et al., 2016). Efficient training methods of very deep neural networks, however, remain an active area of research. The most commonly applied training method is stochastic gradient descent (Robbins and Monro, 1951; Bottou, 2010) and its variants (Duchi et al., 2011; Zeiler, 2012; Kingma and Ba, 2014; Johnson and Zhang, 2013), where incremental updates to the trainable parameters are performed using gradient information computed via back-propagation (Kelley, 1960; Bryson, 1975). While efficient to implement, the incremental updates to the parameter tend to be slow, especially in the initial stages of the training. Moreover, other than the computation of gradients through back-propagation, the specific structure of deep neural networks is not exploited. These observations point to the question of whether there exists alternative training methods tailored to deep neural networks.

In a series of papers, we introduce an alternative approach by exploring the optimal control viewpoint of deep learning (E, 2017). Our focus will be on ideas and algorithms derived from the powerful Pontryagin's maximum principle (Boltyanskii et al., 1960; Pontryagin, 1987), which has two major components: the Hamiltonian dynamics and the condition that at each time the optimal parameters maximize the Hamiltonian. The second component suggests that optimization can be performed independently at different layers. One can also derive an explicit error control estimate based on the maximum principle (see Lemma 2 below).

In this first paper, we will consider the simplest context in which the deep neural networks are replaced by continuous (or discretized) dynamical systems, and devise numerical algorithms that are based on the optimality conditions in the Pontryagin's maximum principle. This leads to a new approach for training deep learning models that have certain advantages, such as fast initial descent and resilience to stalling in flat landscapes. An additional advantage is that one has a good control of the error through explicit estimates.

The rest of the paper is organized as follows. In Section 2, we present a dynamical systems viewpoint of function approximation and deep learning. We then discuss the necessary optimality conditions, which is the well-known Pontryagin's maximum principle. In Section 3 and 4, we discuss numerical methods to solve the necessary conditions and obtain error estimates and convergence guarantees. Using benchmarking examples, we then compare our method with traditional gradient-descent based methods for optimizing deep neural networks in Section 5. In Section 6, we discuss and compare our work with existing literature. Conclusion and outlook are given in Section 7.

2. Function Approximation by Dynamical Systems

We start with a description of the (continuous) dynamical systems approach to machine learning (see E 2017). The essential task of supervised learning is to approximate some function

$$F : \mathcal{X} \to \mathcal{Y}$$

which maps inputs in $\mathcal{X} \subset \mathbb{R}^d$ (e.g. images, time-series) to labels in $\mathcal{Y}$ (categories, numerical predictions). Given a collection of K sample input-label pairs $\{x^i, y^i = F(x^i)\}_{i=1}^K$, one aims to approximate F using these data points. In the dynamical systems framework, we consider the inputs $x = (x^1, \ldots, x^K) \in \mathbb{R}^{d\times K}$ as the initial condition of a system of ordinary

differential equations

$$\dot{X}_t^i = f(t, X_t^i, \theta_t), \quad X_0^i = x^i, \quad 0 \le t \le T, \tag{1}$$

where $\theta : [0,T] \to \Theta \subset \mathbb{R}^p$, represents the control (training) parameters and $X_t = (X_t^1, \dots, X_t^K) \in \mathbb{R}^{d\times K}$ for all $t \in [0,T]$. The form of f is chosen as part of the machine learning model. For example, in deep learning, f is typically the composition (in either order) of a linear transformation and a component-wise nonlinear function (the activation function). For the solution to (1) to exist for any θ, we shall assume hereafter that f and $\nabla_x f$ are continuous in t, x, θ. Note that weaker but more complicated conditions can be considered (Clarke, 2005). For the i^{th} input sample, the prediction of the "network" is a deterministic transformation of the terminal state $g(X_T^i)$ for some $g : \mathbb{R}^d \to \mathcal{Y}$, which we can view collectively as a function of the initial state (input) x^i and the control parameters (weights) θ. The dynamics (1) are decoupled across samples except for the dependence on the control θ. We shall consider quite a general space of controls

$$\mathcal{U} := \{\theta : [0,T] \to \Theta : \theta \text{ is Lebesgue measurable}\}.$$

The aim is to select θ from $\mathcal{U}$ so that $g(X_T^i)$ most closely resembles y^i for $i = 1, \dots, K$. To this end, we define a loss function $\Phi : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ which is minimized when its arguments are equal, and we consider minimizing $\sum_i \Phi(g(x_T^i), y^i)$. Since g is fixed, we shall absorb it into the definition of the loss function by defining $\Phi_i(\cdot) := \Phi(\cdot, y_i)$. Then, the supervised learning problem in our framework is

$$\begin{aligned} &\min_{\theta\in\mathcal{U}} \sum_{i=1}^K \Phi_i(X_T^i) + \int_0^T L(\theta_t)dt, \\ &\dot{X}_t^i = f(t, X_t^i, \theta_t), \quad X_0^i = x^i, \quad 0 \le t \le T, \quad i = 1, \dots, K, \end{aligned} \tag{2}$$

where $L : \Theta \to \mathbb{R}$ is a running cost, or the regularizer[1]. We note here that alternatively, we can formulate the supervised learning problem more generally in terms of optimal control in function spaces, see Appendix A.

Problem (2) is a special case of a class of general optimal control problem for ordinary differential equations (Bertsekas, 1995; Athans and Falb, 2013). The advantage of this formulation is that we can write down and study the optimality conditions of (2) entirely in continuous time and derive numerical algorithms that can subsequently be discretized. In other words, we *optimize, then discretize*, as opposed to the traditional reverse approach in deep learning.

As was suggested in E (2017), deep residual networks (He et al., 2016) can be considered as the forward Euler discretization of the continuous approach described above. In this connection, the algorithms presented in this paper can also be formulated in the context of deep residual networks. For general deep neural networks, although one can also formulate similar algorithms, it is not clear at this moment that PMP holds and these algorithms are valid (e.g. converge to the right solution) in the general setting. This issue will be studied in future work.

1. We can also make L depend on X_t, but for simplicity of presentation and the fact that most current machine learning models do not regularize the states, we shall omit this general case.

The optimization problem (2) can be solved by first discretizing it into a discrete problem (a feed-forward neural network) and then applying back propagation and gradient descent approaches commonly used in deep learning. However, here we will present an alternative approach. Hereafter, for simplicity of notation we shall set $K = 1$ drop the scripts i on all functions, noting that analogous results can be obtained in the general case since the dynamics and loss functions are decoupled across samples. Equivalently, we can think of this as effectively concatenating all K sample inputs into a single input vector of dimension $d \times K$ and redefine our dynamics accordingly. Hence, all results remain valid if we perform full-batch training. The case of mini-batch training is discussed in Section 4.3.

2.1 Pontryagin's Maximum Principle

In this section, we introduce a set of necessary conditions for optimal solutions of (2), known as the Pontryagin's Maximum Principle (PMP) (Boltyanskii et al., 1960; Pontryagin, 1987). This shall pave way for an alternative numerical algorithm to train (2) and its discrete-time counter-part.

To begin with, we define the *Hamiltonian* $H : [0,T] \times \mathbb{R}^d \times \mathbb{R}^d \times \Theta \to \mathbb{R}$ given by

$$H(t, x, p, \theta) := p \cdot f(t, x, \theta) - L(\theta).$$

Theorem 1 (Pontryagin's Maximum Principle) *Let $\theta^* \in \mathcal{U}$ be an essentially bounded optimal control, i.e. a solution to (2) with $\operatorname{ess\,sup}_{t\in[0,T]} \|\theta_t^*\|_\infty < \infty$ (ess sup denotes the essential supremum). Denote by X^* the corresponding optimally controlled state process. Then, there exists an absolutely continuous co-state process $P^* : [0,T] \to \mathbb{R}^d$ such that the Hamilton's equations*

$$\dot{X}_t^* = \nabla_p H(t, X_t^*, P_t^*, \theta_t^*), \qquad X_0^* = x, \tag{3}$$

$$\dot{P}_t^* = -\nabla_x H(t, X_t^*, P_t^*, \theta_t^*), \qquad P_T^* = -\nabla\Phi(X_T^*), \tag{4}$$

are satisfied. Moreover, for each $t \in [0,T]$, we have the Hamiltonian maximization condition

$$H(t, X_t^*, P_t^*, \theta_t^*) \ge H(t, X_t^*, P_t^*, \theta) \text{ for all } \theta \in \Theta. \tag{5}$$

The proof of the PMP and its variants can be found in any optimal control theory reference, e.g. Athans and Falb (2013); Bertsekas (1995); Liberzon (2012). Some generalizations can be found in Clarke (2005) and references therein. For example, the requirement of the continuity of f with respect to t can be replaced by a much weaker measurability requirement if one assumes more conditions on $\nabla_x f$. In the statement of Theorem 1, we omitted a technicality involving an abnormal multiplier: the terminal condition for P^* should be $P_T^* = -\lambda\nabla\Phi(X_T^*)$ and the Hamiltonian should be defined as $H(t, x, p, \theta) = p \cdot f(t, x, \theta) - \lambda L(\theta)$ for some $\lambda \ge 0$ (abnormal multiplier) that we can choose. When we are forced to always take $\lambda = 0$, the problem is *singular* and in a sense ill-posed (Athans and Falb, 2013). On the contrary, if we can take a positive λ, we can then rescale the equation for P^* so that we can take $\lambda = 1$ without loss of generality. We shall hereafter assume that this is the case.

A few remarks are in order. First, Equation 3, 4 and 5 allow us to solve for the unknowns X^*, P^*, θ^* simultaneously as a function of t. In this sense, the resulting optimal control θ^* is *open-loop* and is not in a feed-back form $\theta_t^* = \theta^*(X_t^*)$. The latter is of closed-loop type and are typically obtained from dynamic programming and the Hamilton-Jacobi-Bellman formalism (Bellman, 2013). In this sense, the PMP gives a weaker control. However, open-loop solutions are sufficient for neural network applications, where the trained weights and biases are fixed and only depend on the layer number and not the inputs.

PMP can be regarded as a (highly non-trivial) generalization of the calculus of variations to non-smooth settings (since we only assume θ^* to be measurable). Perhaps more familiar to the optimization community, the PMP is related to the Karush-Kuhn-Tucker (KKT) conditions for non-linear constrained optimization. Indeed, we can view (2) as a non-linear program over the function space $\mathcal{U}$ where the constraint is the ODE (1). In this sense, the co-state process P^* plays the role of a continuous-time analogue of Lagrange multipliers. The key difference between the PMP and the KKT conditions (besides the lack of inequality constraints on the state) is the Hamiltonian maximization condition (5), which is stronger than a typical first-order condition that assumes smoothness with respect to θ (e.g. $\nabla_\theta H = 0$). In particular, the PMP says that H is not only stationary, but globally maximized at an optimal control - which is a much stronger statement if H is not concave. Moreover, the PMP makes minimal assumptions on the parameter space Θ; the PMP holds even when f is non-smooth with respect to θ, or worse, when Θ is a discrete subset of $\mathbb{R}^p$.

Last, we emphasize that the PMP is only a necessary condition, hence there can be cases where solutions to the PMP is not actually globally optimal for (2). Nevertheless, in practice the PMP is often strong enough to give good solution candidates, and when certain convexity assumptions are satisfied the PMP becomes sufficient (Bressan and Piccoli, 2007). In the next section, we will discuss numerical methods that can be used to solve the PMP.

3. Method of Successive Approximations

Now, our strategy is to devise numerical algorithms for training (2) via solving the PMP (Equation 3, 4 and 5). We derive and analyze algorithms entirely in continuous time, which allows us to characterize errors estimates and convergence in a more transparent fashion.

There are many methods for the numerical solution of the PMP, including two-point boundary value problem method (Bryson, 1975; Roberts and Shipman, 1972), and collocation methods (Betts, 1998) coupled with general non-linear programming techniques (Bertsekas, 1999; Bazaraa et al., 2013). See (Rao, 2009) for a more recent review. However, many of these methods concern small-scale problems typically encountered in control applications (e.g. trajectory optimization of spacecrafts) and do not scale well to modern machine learning problems with a large number of state and control variables. One exception is the method of successive approximations (MSA) (Chernousko and Lyubushin, 1982), which is an iterative method based on alternating propagation and optimization steps. We first introduce the simplest form of the MSA.

3.1 Basic MSA

Observe that (3) is simply the equation

$$\dot{X}_t^* = f(t, X_t^*, \theta_t^*),$$

and is independent of the co-state P^*. Therefore, we may proceed in the following manner. First, we make an initial guess of the optimal control $\theta^0 \in \mathcal{U}$. For each $k = 0, 1, 2, \ldots$, we first solve (3)

$$\dot{X}_t^{\theta^k} = f(t, X_t^{\theta^k}, \theta_t^k), \quad X_0^{\theta^k} = x. \tag{6}$$

for X^{θ^k}, which then allows us to solve (4)

$$\dot{P}_t^{\theta^k} = -\nabla_x H(t, X_t^{\theta^k}, P_t^{\theta^k}, \theta_t^k), \quad P_T^{\theta^k} = -\nabla\Phi(X_T^{\theta^k}), \tag{7}$$

to get P^{θ^k}. Finally, we use the maximization condition (5) to set

$$\theta_t^{k+1} = \arg\max_{\theta\in\Theta} H(t, X_t^{\theta^k}, P_t^{\theta^k}, \theta),$$

for $t \in [0, T]$. The algorithm is summarized in Algorithm 1.

Algorithm 1: Basic MSA

```
Initialize: θ^0 ∈ U ;
for k = 0 to #Iterations do
    Solve Ẋ_t^{θ^k} = f(t, X_t^{θ^k}, θ_t^k),   X_0^{θ^k} = x;
    Solve Ṗ_t^{θ^k} = −∇_x H(t, X_t^{θ^k}, P_t^{θ^k}, θ_t^k),   P_T^{θ^k} = −∇Φ(X_T^{θ^k});
    Set θ_t^{k+1} = arg max_{θ∈Θ} H(t, X_t^{θ^k}, P_t^{θ^k}, θ) for each t ∈ [0, T];
end
```

As is the case with the maximum principle, MSA consists of two major components: the forward-backward Hamiltonian dynamics and the maximization for the optimal parameters at each time. An important feature of MSA is that the Hamiltonian maximization step is decoupled for each $t \in [0, T]$. In the language of deep learning, the optimization step is decoupled for different network layers and only the Hamiltonian ODEs (Step 3,4 of Algorithm 1) involve propagation through the layers. This allows the parallelization of the maximization step, which is typically the most time-consuming step.

It has been shown that the basic MSA converges for a restricted class of linear quadratic regulators (Aleksandrov, 1968). However, in general it tends to diverge, especially if a bad initial θ^0 is chosen (Aleksandrov, 1968; Chernousko and Lyubushin, 1982). Our goal now is to modify the basic MSA to control its divergent behavior. Before we do so, it is important to understand why the MSA diverges, and in particular, the relationship between the maximization step in Algorithm 1 and the optimization problem (2).

3.2 Error Estimate for the Basic MSA

For each $\theta \in \mathcal{U}$, let us denote

$$J(\theta) := \Phi(X_T^\theta) + \int_0^T L(\theta_t)dt,$$

where X^θ satisfies (6). Our goal is to minimize $J(\theta)$. We show in the following Lemma the relationship between the values of J and the Hamiltonian maximization step. We start by making the following assumptions.

(A1) Φ is twice continuously differentiable, with Φ and $\nabla\Phi$ satisfying a Lipschitz condition, i.e. there exists $K > 0$ such that

$$|\Phi(x) - \Phi(x')| + \|\nabla\Phi(x) - \nabla\Phi(x')\| \le K\|x - x'\|,$$

for all $x, x' \in \mathbb{R}^d$.

(A2) $f(t, \cdot, \theta)$ is twice continuously differentiable in x, with $f, \nabla_x f$ satisfying a Lipschitz condition in x uniformly in θ and t, i.e. there exists $K > 0$ such that

$$\|f(t, x, \theta) - f(t, x', \theta)\| + \|\nabla_x f(t, x, \theta) - \nabla_x f(t, x', \theta)\|_2 \le K\|x - x'\|,$$

for all $x, x' \in \mathbb{R}^d$ and $t \in [0, T]$. Note that $\|\cdot\|_2$ denotes the induced 2-norm.

With these assumptions, we have the following estimate:

Lemma 2 *Suppose (A1)-(A2) holds. Then, there exists a constant $C > 0$ such that for any $\theta, \phi \in \mathcal{U}$,*

$$\begin{aligned} J(\phi) \le & J(\theta) - \int_0^T \Delta_{\phi,\theta} H(t)dt \\ & + C\int_0^T \|f(t, X_t^\theta, \phi_t) - f(t, X_t^\theta, \theta_t)\|^2 dt \\ & + C\int_0^T \|\nabla_x H(t, X_t^\theta, P_t^\theta, \phi_t) - \nabla_x H(t, X_t^\theta, P_t^\theta, \theta_t)\|^2 dt, \end{aligned}$$

where X^θ, P^θ satisfy Equations 6, 7 respectively and $\Delta H_{\phi,\theta}$ denotes the change in Hamiltonian

$$\Delta H_{\phi,\theta}(t) := H(t, X_t^\theta, P_t^\theta, \phi_t) - H(t, X_t^\theta, P_t^\theta, \theta_t).$$

Proof See Appendix B for the proof and discussion on relaxing the assumptions. ■

In essence, Lemma 2 says that the Hamiltonian maximization step in MSA (step 5 in Algorithm 1) is in some sense the optimal descent direction for J. However, the last two terms on the right hand side indicates that this descent can be nullified if substituting ϕ for θ incurs too much error in the Hamiltonian dynamics (step 3,4 in Algorithm 1). In other words, the last two integrals measure the degree of satisfaction of the Hamiltonian dynamics (3), (4), which can be viewed as a feasibility condition, when one replaces θ by ϕ. Hence, we shall hereafter refer to these errors as *feasibility errors*. The divergence of the basic MSA happens when the feasibility errors blow up. Armed with this understanding, we can then modify the basic MSA to ensure convergence.

3.3 Extended PMP and Extended MSA

As discussed previously in Lemma 2, the decrement of J is ensured if we can control the feasibility errors in the Hamiltonian dynamics in steps 3,4 of Algorithm 1. To this end, we employ a similar idea to augmented Lagrangians (Hestenes, 1969). Fix some $\rho > 0$ and introduce the augmented Hamiltonian

$$\begin{aligned} \tilde{H}(t, x, p, \theta, v, q) := & H(t, x, p, \theta) - \frac{1}{2}\rho\|v - f(t, x, \theta)\|^2 \\ & - \frac{1}{2}\rho\|q + \nabla_x H(t, x, p, \theta)\|^2. \end{aligned} \tag{8}$$

Then, we have the following set of alternative necessary conditions for optimality:

Proposition 3 (Extended PMP) *Suppose that θ^* is an essentially bounded solution to the optimal control problem (2). Then, there exists an absolutely continuous co-state process P^* such that the tuple $(X_t^*, P_t^*, \theta_t^*)$ satisfies the necessary conditions*

$$\dot{X}_t^* = \nabla_p \tilde{H}(t, X_t^*, P_t^*, \theta_t^*, \dot{X}_t^*, \dot{P}_t^*), \qquad X_0^* = x, \tag{9}$$

$$\dot{P}_t^* = -\nabla_x \tilde{H}(t, X_t^*, P_t^*, \theta_t^*, \dot{X}_t^*, \dot{P}_t^*), \qquad P_T^* = -\nabla_x \Phi(X_T^*), \tag{10}$$

$$\tilde{H}(t, X_t^*, P_t^*, \theta_t^*, \dot{X}_t^*, \dot{P}_t^*) \ge \tilde{H}(t, X_t^*, P_t^*, \theta, \dot{X}_t^*, \dot{P}_t^*), \qquad \theta \in \Theta, t \in [0, T]. \tag{11}$$

Proof If θ^* is optimal, then by the PMP there exists a co-state process P^* such that (3), (4) and (5) are satisfied. Then, for all $t \in [0, T]$ and $\theta \in \Theta$ we have

$$\begin{aligned} \nabla_x \tilde{H}(t, X_t^*, P_t^*, \theta, \dot{X}_t^*, \dot{P}_t^*) = & \nabla_x H(t, X_t^*, P_t^*, \theta), \\ \nabla_p \tilde{H}(t, X_t^*, P_t^*, \theta, \dot{X}_t^*, \dot{P}_t^*) = & \nabla_p H(t, X_t^*, P_t^*, \theta), \end{aligned}$$

which implies that (9) and (10) are satisfied. Lastly, we can write

$$\begin{aligned} & \tilde{H}(t, X_t^*, P_t^*, \theta, \dot{X}_t^*, \dot{P}_t^*) \\ = & H(t, X_t^*, P_t^*, \theta) - \frac{1}{2}\rho\|\dot{X}_t^* - f(t, X_t^*, \theta)\|^2 - \frac{1}{2}\rho\|\dot{P}_t^* + \nabla_x H(t, X_t^*, P_t^*, \theta)\|^2. \end{aligned}$$

For each t, θ^* maximizes all three terms on the RHS simultaneously, and hence (11) is also satisfied. ■

Compared with the usual PMP, the extended PMP is a weaker necessary condition. However, the advantage is that maximization of $\tilde{H}$ naturally penalizes errors in the Hamiltonian dynamical equations, and hence we should expect MSA applied to the extended PMP to converge for large enough ρ. Note that the Hamiltonian equation steps do not change (since the added terms have no effect on optimal solutions) and the only change is the maximization step. The extended MSA (E-MSA) algorithm is summarized in Algorithm 2.

To establish convergence, define

$$\mu_k := \int_0^T \Delta H_{\theta^{k+1},\theta^k}(t)dt \ge 0.$$

If $\mu_k = 0$, then from the Hamiltonian maximization step (11) we must have

$$0 = -\mu_k \le -\frac{1}{2}\rho\int_0^T \|f(t,X_t^{\theta^k},\theta_t^{k+1}) - f(t,X_t^{\theta^k},\theta_t^k)\|^2 dt$$
$$-\frac{1}{2}\rho\int_0^T \|\nabla_x H(t,X_t^{\theta^k},P_t^{\theta^k},\theta_t^{k+1}) - \nabla_x H(t,X_t^{\theta^k},P_t^{\theta^k},\theta_t^k)\|^2 dt. \le 0.$$

and so

$$\max_\theta \tilde{H}(X_t^{\theta^k},P_t^{\theta^k},\theta,\dot{X}_t^{\theta^k},\dot{P}_t^{\theta^k}) = \tilde{H}(X_t^{\theta^k},P_t^{\theta^k},\theta_k,\dot{X}_t^{\theta^k},\dot{P}_t^{\theta^k}),$$

i.e. $(X^{\theta^k},P^{\theta^k},\theta^k)$ satisfy the extended PMP. In other words, the quantity $\mu_k \ge 0$ measures the distance from a solution of the extended PMP, and if it equals 0, then we have a solution. We now prove the following result that guarantees the convergence of the extended MSA (Algorithm 2).

Theorem 4 *Let (A1)-(A2) be satisfied and $\theta^0 \in \mathcal{U}$ be any initial measurable control with $J(\theta^0) < +\infty$. Suppose also that $\inf_{\theta\in\mathcal{U}} J(\theta) > -\infty$. Then, for ρ large enough, we have under Algorithm 2,*

$$J(\theta^{k+1}) - J(\theta^k) \le -D\mu_k.$$

for some constant $D > 0$ and

$$\lim_{k\to 0}\mu_k = 0,$$

i.e. the extended MSA algorithm converges to the set of solutions of the extended PMP.

Proof Using Lemma 2 with $\theta \equiv \theta^k, \phi \equiv \theta^{k+1}$, we have

$$J(\theta^{k+1}) - J(\theta^k) \le -\mu_k + C\int_0^T \|f(t,X_t^{\theta^k},\theta_t^{k+1}) - f(t,X_t^{\theta^k},\theta_t^k)\|^2 dt$$
$$+ C\int_0^T \|\nabla_x H(t,X_t^{\theta^k},P_t^{\theta^k},\theta_t^{k+1}) - \nabla_x H(t,X_t^{\theta^k},P_t^{\theta^k},\theta_t^k)\|^2 dt.$$

From the Hamiltonian maximization step in Algorithm 2, we know that

$$H(t,X_t^{\theta^k},P_t^{\theta^k},\theta_t^k) \le H(t,X_t^{\theta^k},P_t^{\theta^k},\theta_t^{k+1})$$
$$-\frac{1}{2}\rho\|f(t,X_t^{\theta^k},\theta_t^{k+1}) - f(t,X_t^{\theta^k},\theta_t^k)\|^2$$
$$-\frac{1}{2}\rho\|\nabla_x H(t,X_t^{\theta^k},P_t^{\theta^k},\theta_t^{k+1}) - \nabla_x H(t,X_t^{\theta^k},P_t^{\theta^k},\theta_t^k)\|^2.$$

Hence, we have

$$J(\theta^{k+1}) - J(\theta^k) \le -(1-\frac{2C}{\rho})\mu_k.$$

Pick $\rho > 2C$, then we indeed have $J(\theta^{k+1}) - J(\theta^k) \le -D\mu_k$ with $D = (1-\frac{2C}{\rho}) > 0$. Moreover, we can rearrange and sum the above expression to get

$$\sum_{k=0}^M \mu_k \le D^{-1}(J(\theta^0) - J(\theta^{M+1})) \le D^{-1}(J(\theta^0) - \inf_{\theta\in\mathcal{U}} J(\theta)),$$

and hence $\sum_{k=0}^\infty \mu_k < +\infty$, which implies $\mu_k \to 0$ and the extended MSA converges to a solution of the extended PMP. ■

Algorithm 2: Extended MSA

1 Initialize: $\theta^0 \in \mathcal{U}$. Hyper-parameter: ρ ;
2 **for** $k = 0$ **to** *#Iterations* **do**
3 Solve $\dot{X}_t^{\theta^k} = f(t,X_t^{\theta^k},\theta_t^k), \quad X_0^{\theta^k} = x$;
4 Solve $\dot{P}_t^{\theta^k} = -\nabla_x H(t,X_t^{\theta^k},P_t^{\theta^k},\theta_t^k), \quad P_T^{\theta^k} = -\nabla\Phi(X_T^{\theta^k})$;
5 Set $\theta_t^{k+1} = \arg\max_{\theta\in\Theta} \tilde{H}(t,X_t^{\theta^k},P_t^{\theta^k},\theta,\dot{X}_t^{\theta^k},\dot{P}_t^{\theta^k})$ for each $t \in [0,T]$;
6 **end**

4. Discrete-Time Formulation

In the previous section, we discussed the PMP and MSA in the continuous-time setting, where we showed that an appropriately extended version (E-MSA) converges to a solution of an extended PMP. Here, we shall discuss the discretized versions of PMP, MSA and E-MSA, as well as their connections to deep residual networks and back-propagation.

4.1 Discrete-Time PMP and Discrete-Time MSA

Applying Euler-discretization to Equation 1, we get

$$x_{n+1} = x_n + \delta f_n(x_n,\vartheta_n), \quad x_0 = x,$$

for $n = 0,\dots,N-1$, with $\delta = T/N$ (step-size), $x_n := X_{n\delta}$, $\vartheta_n := \theta_{n\delta}$ and $f_n(\cdot) := f(n\delta,\cdot)$. Then, the discrete-time analogue of the control problem (2) is

$$\min_{\{\vartheta_0,\dots,\vartheta_{N-1}\}\in\Theta^N} \Phi(x_N) + \delta\sum_{n=0}^{N-1} L(\vartheta_n),$$
$$x_{n+1} = x_n + \delta f_n(x_n,\vartheta_n), \quad x_0 = x, \quad 0 \le n \le N-1. \tag{12}$$

Observe that barring the constant δ, this is exactly the supervised learning problem for deep residual networks[2]. Therefore, when suitably discretized, one expects that the E-MSA provides a means to train residual neural networks via the solution of the extended PMP.

We now write down formally the discretized form of the PMP. Let us use the shorthand $g_n(x_n,\vartheta_n) := x_n + \delta f_n(x_n,\vartheta_n)$. Define the scaled discrete Hamiltonian

$$H_n(x,p,\vartheta) = p\cdot g_n(x,\vartheta) - \delta L(\vartheta).$$

Then, a discrete-time PMP is the following set of conditions:

$$x_{n+1}^* = g_n(x_n^*,\vartheta_n^*), \qquad x_0^* = x,$$
$$p_n^* = \nabla_x H_n(x_n^*,p_{n+1}^*,\vartheta_n), \qquad p_N^* = -\nabla_x\Phi(x_N^*),$$
$$H_n(x_n^*,p_{n+1}^*,\vartheta_n^*) \ge H_n(x_n^*,p_{n+1}^*,\vartheta), \qquad \vartheta\in\Theta, \quad n = 0,\dots,N-1.$$

2. If we pick ReLU activations (Hahnloser et al., 2000), then δ can be absorbed into ϑ

The issue of whether the PMP holds for discrete time dynamical systems is a delicate one and there are known counterexamples (Butkovsky, 1963; Jackson and Horn, 1965; Nahorski et al., 1984). Nevertheless, they must hold approximately for small time step size and this is the situation we will consider in the current paper. We expect Lemma 2, which implies monotonicity of the E-MSA algorithm, to hold in the discrete-time case under appropriate conditions. We leave a rigorous analysis of these statements to future work. For numerical experiments presented in the next section, we shall almost always work with residual networks that can be regarded as discretizations of continuous networks so that the PMP holds approximately at least (Halkin, 1966).

For completeness, we summarize the discrete-time version of E-MSA in Algorithm 3. Note that for residual networks ($g_n = x_n + \delta f_n$), this is equivalent to a forward Euler discretization on the state equation and a backward Euler discretization on the co-state equation in Algorithm 2. As before, the Hamiltonian maximization step is decoupled across layers and can be carried out in parallel.

Algorithm 3: Discrete-time E-MSA

1 Initialize: Initialize: $\vartheta_n^0 \in \Theta_n$, $n = 0, \ldots, N-1$. Hyper-parameter: ρ ;
2 **for** $k = 0$ **to** *#Iterations* **do**
3 Set $x_0^{\vartheta^k} = x$;
4 **for** $n = 0$ **to** $N-1$ **do**
5 $x_{n+1}^{\vartheta^k} = g_n(x_n^{\vartheta^k}, \vartheta_n^k)$;
6 **end**
7 Set $p_N^{\vartheta^k} = -\nabla\Phi(x_N^{\vartheta^k})$;
8 **for** $n = N-1$ **to** 0 **do**
9 $p_n^{\vartheta^k} = \nabla_x H_n(x_n^{\vartheta^k}, p_{n+1}^{\vartheta^k}, \vartheta_n^k)$;
10 **end**
11 **for** $n = 0$ **to** $N-1$ **do**
12 Set $\vartheta_n^{k+1} = \arg\max_{\vartheta\in\Theta_n} H_n(x_n^{\vartheta^k}, p_{n+1}^{\vartheta^k}, \vartheta) - \frac{1}{2}\rho\|x_{n+1}^{\vartheta^k} - g_n(x_n^{\vartheta^k}, \vartheta)\|_2^2 - \frac{1}{2}\rho\|p_n^{\vartheta^k} - \nabla_x H_n(x_n^{\vartheta^k}, p_{n+1}^{\vartheta^k}, \vartheta)\|_2^2$;
13 **end**
14 **end**

4.2 Relationship to Gradient Descent with Back-propagation

We note an interesting relationship of the MSA with classical gradient descent with back-propagation (Kelley, 1960; Bryson, 1975; LeCun et al., 1998). We have shown in Lemma 2 that the divergence of MSA can be attributed to the large errors in the Hamiltonian dynamics terms caused by the maximization step, which involve drastic changes in parameter values. Assuming each Θ_n is a continuum and g_n, L are differentiable in ϑ_n, a simple fix is to make the maximization step "soft": we replace step 12 in Algorithm 3 with a gradient ascent step:

$$\vartheta_n^{k+1} = \vartheta_n^k + \eta\nabla_\vartheta H_n(x_n^{\vartheta^k}, p_{n+1}^{\vartheta^k}, \vartheta_n^k), \tag{13}$$

for some small learning rate η. We now show that in the discrete-time setting, this is equivalent to the classical gradient descent with back-propagation.

Proposition 5 *The basic MSA in discrete-time (Algorithm 3 with $\rho = 0$) with step 12 replaced by* (13) *is equivalent to gradient descent with back-propagation.*

Proof Recall that the Hamiltonian is

$$H_n := p_{n+1} \cdot g_n(x_n, \vartheta_n) - \delta L(\vartheta_n),$$

and the total loss function is $J(\vartheta) = \Phi(x_N) + \delta\sum_{n=0}^{N-1} L(\vartheta_n)$. It is easy to see that $p_n = -\nabla_{x_n}\Phi(x_N)$ by working backwards from $n = N$ and the fact that $\nabla_{x_n} x_{n+1} = \nabla_x g_n(x_n, \vartheta_n)$. Then,

$$\begin{aligned}\nabla_{\vartheta_n} J(\vartheta) =& \nabla_{x_{n+1}}\Phi(x_N) \cdot \nabla_{\vartheta_n} x_{n+1} + \delta\nabla_{\vartheta_n} L(\vartheta_n)\\ =& -p_{n+1} \cdot \nabla_{\vartheta_n} g_n(x_n, \vartheta_n) + \delta\nabla_{\vartheta_n} L(\vartheta_n)\\ =& -\nabla_{\vartheta_n} H_n.\end{aligned}$$

Hence, (13) is simply the gradient descent step

$$\vartheta_n^{k+1} = \vartheta_n^k - \eta\nabla_{\vartheta_n} J(\vartheta^k).$$

■

As the proposition shows, gradient descent with back-propagation can be seen as a modification of the basic MSA by replacing the Hamiltonian maximization step with a gradient ascent step. However, we note that the PMP (and MSA convergence) holds, at least in continuous-time, even when differentiability with respect to ϑ is not satisfied, and hence is more general than the classical back-propagation. In fact, the PMP formalism shows that the back-propagation of information through a deep network is handled by the co-state equation and there is no requirement or relationship to the gradients with respect to the trainable parameters. In other words, optimization is performed at each layer separately (with or without gradient information), and propagation is independent of optimization.

4.3 A Remark on Mini-batch Algorithms

So far, our discussion has focused on full-batch algorithms, where the input x represents the full set of training inputs. As modern supervised learning tasks typically involve a large number of training samples, usually the optimization problem has to be solved in mini-batches, where at each iteration we sub-sample m input-label pairs and optimize the parameters θ (or ϑ in discrete time) based on losses evaluated on these pairs. In the context of continuous-time PMP, we can write the batch version of the three necessary conditions as

$$\begin{aligned}\dot{X}_t^{i,*} &= \nabla_p H(t, X_t^{i,*}, P_t^{i,*}, \theta_t^*), & X_0^{i,*} &= x^i,\\ \dot{P}_t^{i,*} &= -\nabla_x H(t, X_t^{i,*}, P_t^{i,*}, \theta_t^*), & P_T^{i,*} &= -\nabla\Phi^i(X_T^{i,*}),\\ \theta_t^* &= \arg\max_{\theta\in\Theta} \sum_{i=1}^M H(t, X_t^{i,*}, P_t^{i,*}, \theta), & t &\in [0,T],\end{aligned}$$

for samples $i = 1, \dots, M$. We omit for brevity the equivalent expressions for discrete-time. In particular, notice that the propagation steps are decoupled across samples, and hence can be carried out independently. The only difference is the maximization step, where in a mini-batch setting we would evaluate instead

$$\arg\max_{\theta \in \Theta} \sum_{i=1}^{m} H(t, X_t^{i,*}, P_t^{i,*}, \theta).$$

If m is large enough and the samples are independently and identically drawn, then uniform law of large numbers (Jennrich, 1969) holds under fairly general conditions and ensures that the mini-batch mean of Hamiltonians converges uniformly in θ to the full-batch sum. Hence, maximization performed on the mini-batch sum should be close to the actual maximization on the full Hamiltonian. Rigorous error estimates for the mini-batch version of our algorithm is out of the scope of the current work, and we use instead numerical results in Section 5 to demonstrate that the algorithm can also be carried out in a mini-batch fashion.

5. Numerical Experiments

In this section, we investigate the performance of E-MSA compared with the usual gradient-based approaches, namely stochastic gradient descent and its variants: Adagrad (Duchi et al., 2011) and Adam (Kingma and Ba, 2014). To illustrate key properties of E-MSA, we shall begin by investigating some synthetic examples. First, we consider a simple one-dimensional function approximation problem where we want to approximate $F(x) = \sin(x)$ for $x \in [-\pi, \pi]$ using a continuous time dynamical system. Let $T = 5$ and consider

$$\dot{X}_t = f(X_t, \theta_t) = \tanh(W_t X_t + b_t),$$

where $\theta_t = (W_t, b_t) \in \mathbb{R}^{5\times 5} \times \mathbb{R}^5$, i.e. a continuous analogue of a fully connected feed forward neural networks with 5 nodes per layer. To match dimensions, we shall concatenate the input x to form a five dimensional vector of identical components, which is now the initial condition to the dynamical system on $\mathbb{R}^d$. The output of the network is $\sum_{i=1}^5 X_T^i$. and we define the loss function due to one sample to be $\Phi(X_T) = (\sum_{i=1}^5 X_T^i - \sin(x))^2$. For multiple samples, we average the loss function over all samples in the usual way. We apply E-MSA with discretization size $\delta = 0.25$ (giving 20 layers) and compute the Hamiltonian maximization step using 10 iterations of limited memory BFGS method (L-BFGS) (Liu and Nocedal, 1989). In Figure 1(a), we compare the results with gradient descent based optimization approaches, where we observe that E-MSA has favorable convergence rate per-iteration. More interestingly, it is well-known that gradient descent may suffer slow convergence at flat regions or near saddle-points, where the gradients become very small and optimization may stall for a long time. This often occurs as a result of poor initialization of weights and biases (Sutskever et al., 2013). Here, we simulate this scenario by initializing all weights and biases (W_t, b_t) to be 0 and observe the optimization process. We see from Figure 1(b) that gradient descent based methods are more easily stalled at flat regions. We calculated numerically the eigenvalues of the Hessian at this region, which confirms that this is indeed very close to a saddle point. On the other hand, the Hamiltonian maximization in E-MSA can quickly escape the locally flat regions. One possible reason is that second-order information employed by L-BFGS can off-set the small gradients and provide larger updates.

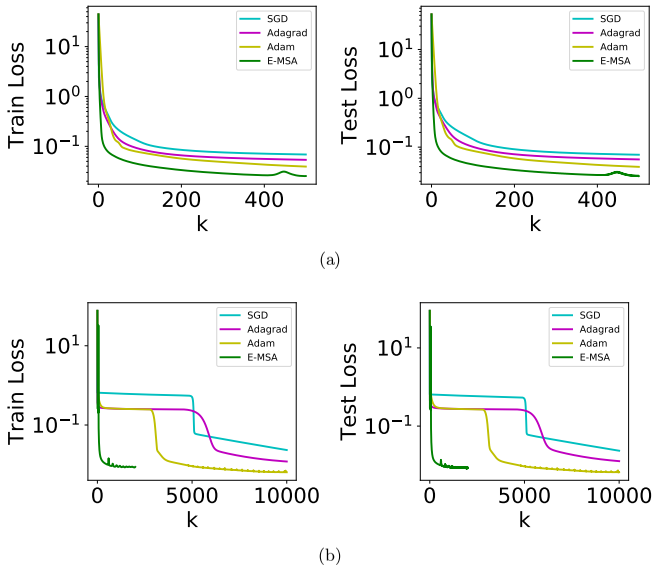


Figure 1: Comparison of E-MSA with gradient-based methods for approximating the sine function with a continuous, 5-dimensional dynamical system. A training and test set of 1000 samples each are used. (a) Loss function vs iterations for a good initialization, where weights are initialized with truncated random normal variables with standard deviation 0.1 and biases are initialized as constants equal to 0.1. We see that E-MSA has good convergence rate per iteration. (b) We use a poor initialization by setting all weights and biases to 0. We observe that gradient descent based methods tend to become stuck whereas E-MSA are better at escaping these slow manifolds, provided that ρ is well chosen (=1.0 in this case).

Next, we consider a familiar supervised learning test problem: the MNIST data set (LeCun, 1998) for handwritten digit recognition, with 55000 training samples and 10000 test samples. We employ a continuous dynamical system that resembles a (residual) convolution neural network (LeCun and Bengio, 1995) when discretized. More concretely, at each t we consider the map $f(t, x, \theta) = \tanh(W \star x + b)$ where W is a 3×3 convolution filter with 32 input and output channels. To match dimensions, we introduce two projection layers at the input (consisting of convolution, point-wise non-linearities followed by 2×2 max-pooling). We also use a fully-connected classification layer as the final layer, with softmax cross-entropy loss. Note that the input projection layers and fully-connected output layers

are not of residual form, but we can nevertheless apply Algorithm 3 with the appropriate g. We use a total of 10 layers (2 projections, 1 fully-connected and 7 residual layers with $\delta = 0.5$, i.e $T = 3.5$). The model is trained with mini-batch sizes of 100 using E-MSA and gradient-descent based methods, namely SGD, Adagrad, and Adam. For E-MSA, we approximately solve the Hamiltonian maximization step using either 10 iterations of L-BFGS. Note that since we have decoupled the layers through the PMP, the L-BFGS step used to maximize H is tractable since it involves much fewer parameters than directly minimizing J. Figure 2 compares the performance of E-MSA with the other gradient-descent based methods, where we observe that E-MSA has good performance per-iteration, especially at early stages of training. However, we also show in Figure 3 that the wall-clock performance of our methods are not currently competitive, because the Hamiltonian maximization step is time consuming and the performance gains per iteration is outweighed by the running time. Note that wall-clock times are compared on the CPU for fairness since we did not use a GPU implementation of L-BFGS. As a further test, we train the same model on a different data set, the fashion MNIST data set (Xiao et al., 2017), where we again observe similar phenomena (see Figure 4). Experiments on more complex data sets such as ImageNet (Deng et al., 2009) with larger residual networks is a direction of future work. In particular, this may require further improvements to the Hamiltonian maximization step current handled by direct minimization with L-BFGS, which can be significantly slower (on a wall-clock basis) for larger networks and data sets.

6. Discussion and Related Work

We commence this section by highlighting the distinguishing features of E-MSA from traditional gradient-descent based training methods. First, the formulations of PMP and E-MSA do not involve gradient information with respect to the trainable parameters. In fact, Theorem 1 and Algorithm 2 remain valid even when the trainable parameters can only take values in a discrete set. Second, due to a more drastic argmax step taken at each iteration, E-MSA tends to have better convergence rates at the early steps of training, as observed in our numerical experiments (Section 5). Third, in the PMP formalism, the Hamiltonian equations for the state and co-state are the "forward and backward propagations", whereas given the state and co-state values, the optimization step is decoupled across layers. This allows one to potentially parallelize the often time-consuming optimization step. Moreover, from Lemma 2, we show that as long as the Hamiltonian is sufficiently increased in a layer without causing too much loss in the Hamiltonian dynamics feasibility conditions, we can ensure decrement of the loss function. This is the reason why we can use a small number of iterations of L-BFGS at each step. Moreover, this suggests that the argmax updates need not happen synchronously, i.e. the optimization in each layer can be a separate thread or process that computes the argmax and updates that layer's parameters independent of other layers. The propagation may also potentially be allowed to happen asynchronously as long as updates are sufficiently frequent. We leave a rigorous analysis of an asynchronous version of the current approach to future work. In summary, the main strength of the PMP (over e.g. solving the KKT conditions using gradient methods) is that PMP says that at the optimum, the Hamiltonian is not only stationary (KKT), but globally maximized. This hints that heuristic global optimization methods can be applied to H to obtain algorithms that

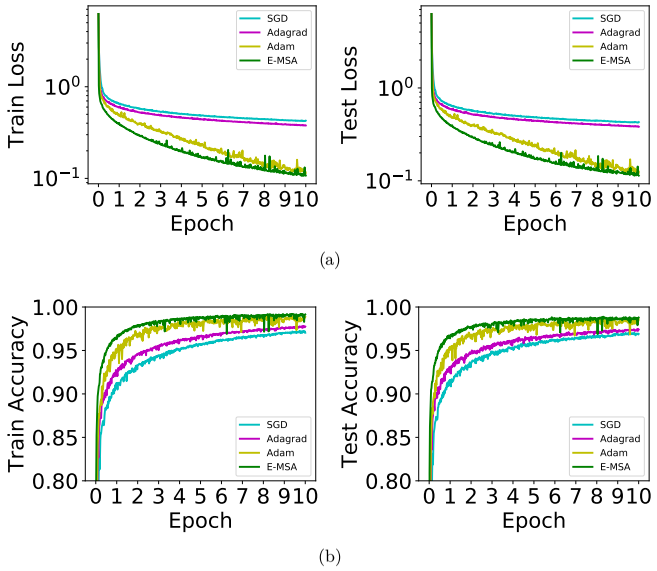


Figure 2: Comparison of E-MSA with gradient-based methods for the residual CNN on the MNIST data set. Mini-batch size of 100 is used so that each epoch of training consists of 550 iterations. (a) Train and test Loss vs epoch. (b) Train and test accuracy vs epoch. For each case, we tuned the associated hyper-parameters on a coarse grid for optimal performance. We observe that per-iteration, E-MSA performs favorably, at least at early times. This shows that if the augmented Hamiltonian can be efficiently maximized, we may obtain good performance.

are very different in behavior compared with gradient-descent based approaches. Again, Lemma 2 ensures that such heuristic global maximization need only be approximate.

As it currently stands, our experiments in Section 5 demonstrate that the Hamiltonian maximization step in E-MSA gives very different behavior compared with gradient-descent based methods. When the Hamiltonian is sufficiently maximized, we indeed obtain favorable performance compared with gradient descent based methods. Furthermore, we saw in Figure 1 that Hamiltonian maximization may avoid pitfalls such as a very flat landscape. Overall, the key to whether E-MSA (and other methods based on solving the PMP) will eventually constitute a replacement for gradient-descent based algorithm lies in the question of whether efficient Hamiltonian maximization can be performed at reasonable computational costs. Although this is still a non-convex optimization problem, it is much simpler than the original training problem because: (1) Optimization in the layers are decoupled and hence parameter space is greatly reduced; (2) The Hamiltonian is formally similar across

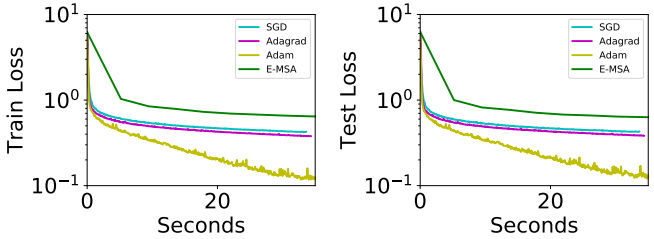


Figure 3: Comparison of E-MSA with gradient-based methods for the residual CNN on the MNIST data set on a wall-clock basis. We observe that currently, the gains per iteration is outweighed by the additional computational costs. Note that we did not use a GPU implementation for the L-BFGS algorithm used to maximize the augmented Hamiltonian, hence the wall-clock time for E-MSA is expected to be improved. Nevertheless, we expect that more efficient Hamiltonian maximization algorithms must be developed for E-MSA to out-perform gradient-based methods in terms of wall-clock efficiency.

different layers, loss functions and models, so specialized algorithms may be designed; (3) The Hamiltonian does not need to be maximized exactly, thus fast heuristic methods (Lee and El-Sharkawi, 2008) or learning (Andrychowicz et al., 2016; Jaderberg et al., 2016; Czarnecki et al., 2017) can potentially be used to perform this. All these are worthy of future exploration in order to make E-MSA truly competitive.

Next, we put our work in perspective by discussing related work in the optimal control, optimization and deep learning literature. First, the work on numerical algorithms for the solution of optimal control problem is abundant (see e.g. Rao 2009 for a survey). Many of the state-of-the-art techniques in the control theory literature assume a moderately small problem size, so that conventional non-linear programming techniques (Bertsekas, 1999; Bazaraa et al., 2013) as well as shooting (Roberts and Shipman, 1972) and collocation methods (Betts, 1998) produce efficient algorithms. This is usually not the case for large-scale machine learning problems, where often, the only scalable approach is to rely on iterative updates to the parameters. This is the reason for our focus on the MSA algorithms (Chernousko and Lyubushin, 1982), as they are straight-forward to implement and typically have linear scaling in computational complexity with respect to the input and parameter sizes. The basic MSA is discussed in Krylov and Chernousko (1962), and a number of improved variants are discussed in Chernousko and Lyubushin (1982) and references therein. For example, a popular improvement is based on needle-perturbations, where controls are varied on small intervals at each iteration. While convergent, the main issue with the needle-perturbation approach is the requirement of a sufficiently fine mesh (i.e. many layers in the discretized network), which impacts computational speed. A possible solution is the use of adaptive meshes, which is a future direction we plan to investigate. Our variant of the MSA presented in this work differs from classical approaches (Chernousko and Lyubushin, 1982) mainly in the sense that we solve a weaker sufficient condition (extended

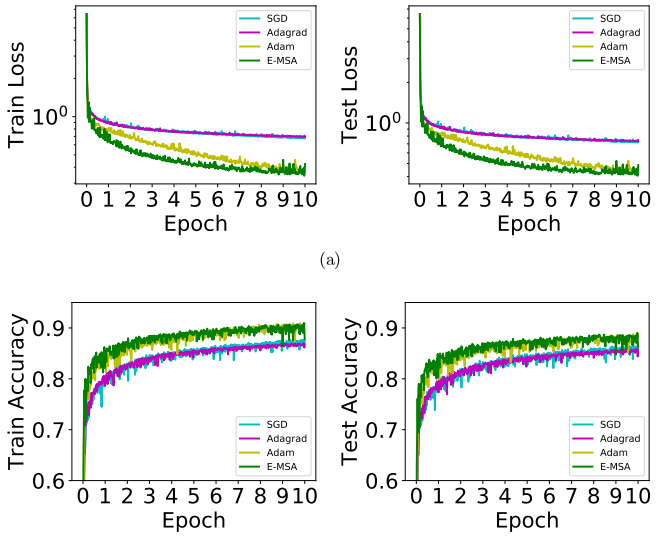


Figure 4: Comparison of E-MSA with gradient-based methods for the residual CNN on the fashion MNIST data set. We use the same network structure and mini-batch sizes as in Figure 2. The hyper-parameters have to be slightly re-tuned. (a) Train and test Loss vs epoch. (b) Train and test accuracy vs epoch. Again, we observe E-MSA performs favorably per-iteration.

PMP, Proposition 3), which then allows us to control errors in the Hamiltonian dynamical equations at every iteration without going into finer mesh-sizes. The regularization terms proportional to ρ is similar to the heuristic modifications suggested in Lyubushin (1982) by regularizing the distance between θ^k and θ^{k+1}, but we do not have to assume convexity of Θ or that f is Lipschitz in θ.

In the optimization literature, our work shares some similarity with the recently proposed ADMM methods (Taylor et al., 2016) for training deep neural networks, where the authors also considered necessary conditions with Lagrange multipliers that can decouple optimization across layers. The main difference in our work is that the PMP gives a stronger necessary condition (Hamiltonian maximization) that also applies to general parameter spaces (e.g., discrete, or bounded with non-linear constraints). Our modification of the basic MSA in terms of the augmented Hamiltonian is inspired by the method of augmented Lagrangians often applied in constrained optimization (Hestenes, 1969). The idea of viewing an initially discrete system as the discretization of a continuous-time system

has been explored in Li et al. (2017) in the form of stochastic optimization. Our current work is also in this flavor, but for neural network models.

In deep learning, there are a few works that share our perspective of deep neural networks as a discretization of a dynamical system. We note that the connection between the PMP and back-propagation has been pointed out qualitatively in LeCun (1988) and in the development of back-propagation (Bryson, 1975; Baydin et al., 2015), although to the best of our knowledge, this work is the first attempt to translate numerical algorithms for the PMP into training algorithms for deep learning that goes beyond gradient descent. The treatment of machine learning as function approximation via a dynamical system has been presented in E (2017). The recent work of Haber and Ruthotto (2017); Chang et al. (2017) also propose the dynamical systems viewpoint, and the authors used continuous-time tools to address stability issues. In contrast, our work focuses on the optimization aspects centered around the PMP. We also mention other recent approaches to decouple optimization in deep neural networks, such as synthetic gradients (Jaderberg et al., 2016; Czarnecki et al., 2017) and proximal back-propagation (Frerix et al., 2017).

7. Conclusion and Outlook

In this paper, we discuss the viewpoint that deep residual neural networks can be viewed as discretization of a continuous-time dynamical system, and hence supervised deep learning can be regarded as solving an optimal control problem in continuous time. We explore a concrete consequence of this connection, by modifying the classical method of successive approximations for solving optimal control problems (in particular the PMP) into a method for solving a weaker sufficient condition (extended PMP). We prove the convergence of the resulting algorithm (E-MSA) and test it on various benchmark problems, where we observe that the E-MSA algorithm performs favorably on a per-iteration basis, especially at early stages of training, compared with gradient-based approaches such as SGD, Adagrad and Adam.

There are many avenues of future research. On the algorithmic side, it is necessary to further improve the computational efficiency of the E-MSA, in particular the Hamiltonian maximization step. Moreover, adaptive selection of ρ depending on iteration number and/or layer can be explored, e.g. by designing adaptive tuning schemes using control theoretic tools (Li et al., 2017). Also, it is desirable to formulate and analyze the PMP and E-MSA from a discrete-time perspective in order to broaden the method's application. From a modeling perspective, viewing deep neural networks as continuous-time dynamical systems is useful in the sense that it allows one to think of neural network architectures as dynamical objects. Indeed, at each training iteration of the E-MSA, we do not have to use the same discretization scheme to compute the Hamiltonian dynamical equations. Also, as the PMP and E-MSA assume little structure on the parameter space Θ, it will also be interesting to apply the E-MSA to train neural networks that have discrete weights (e.g. those that can only take on binary values). Such networks have the advantage of fast inference speed and small memory requirement. However, training such networks is a challenge and most existing techniques rely on approximating or thresholding the derivatives (Courbariaux et al., 2015, 2016). With the PMP and MSA, we may be able to directly train discrete networks in a principled way.

Acknowledgments

The work of W. E is supported in part by Major Program of NNSFC under grant 91130005, ONR grant N00014-13-1-0338, DOE grants DE-SC0008626 and DE-SC0009248 Q. Li is supported by the Agency for Science, Technology and Research, Singapore.

Appendix A. Function Space Formulation

In this section, we give an alternative, non-rigorous formulation of the supervised learning problem as an optimal control problem on function spaces. This provides an alternative formulation of (continuous-time) deep learning that does not make reference to a specific set of input-outputs, but rather their conditional distributions. The idea is to consider the control of a continuity equation that describes the evolution of probability densities. Hereafter, we proceed formally by assuming all differentiability and integrability conditions are satisfied.

We would like to approximate, using a dynamical systems approach, some target joint probability density $\rho(x, y)$, where $x \in \mathcal{X} \subset \mathbb{R}^d$ is a sample input and $y \in \mathcal{Y}$ is the corresponding label. In the case where the labels are deterministically determined by the samples, i.e. there exists $F : \mathcal{X} \to \mathcal{Y}$ such that $y = F(x)$, we would have $\rho(x, y) = \overline{\rho}(x)\delta(F(x) - y)$. Here, $\overline{\rho}(x)$ is the marginal density of $\rho(x, y)$. In general, we can write $\rho(x, y) = \rho(y|x)\overline{\rho}(x)$.

As before, the idea is to consider passing the inputs through a dynamical system

$$\dot{X}_t = f(t, X_t, \theta_t), \qquad X_0 = x. \tag{14}$$

We begin with a guess of a conditional density $\rho_0(y|x)$ of y given x. In the deterministic case, we may set $\rho_0(y|x) = \delta(y - F_0(x))$ for some $F_0 : \mathcal{X} \to \mathcal{Y}$ (this is like the last layer of the neural network, be it a regressor or a classifier). Note that F_0 is potentially very different from F, so that $\rho_0(\cdot|x)$ is far from our target $\rho(\cdot|x)$.

To improve this approximation, we drive the initial condition by the controllable dynamical system (14). That is, we define the approximation at time t of $\rho(y|x)$ to be $\rho_t(y|x) := \langle \rho_0(y|\cdot), u_t \rangle$, with u_t denoting the probability density of X_t at time t (push-forward distribution of X_t according to (14)). It is well-known that u_t follows the continuity equation, or Liouville equation (Gibbs, 2014); or forward Kolmogorov equation in stochastic processes, but with zero noise (Risken, 1996),

$$\frac{d}{dt}u_t = -\mathrm{div}(f(t, \cdot, \theta_t)u_t), \qquad u_0 = \delta_x, \tag{15}$$

where $\mathrm{div} u = \sum_i \partial u / \partial x_i$ is the divergence operator and $\delta_x(x') = \delta(x - x')$ is a point-mass at x. We shall assume that $u_t \in \mathcal{H} \subset L^2(\mathbb{R}^d)$ for some function space $\mathcal{H}$, for all $t \in (0, T]$.

The goal now is to adjust $\theta \in \mathcal{U}$ so that $\rho_t(\cdot|x)$ is close to $\rho(\cdot|x)$. To this end, we define a differentiable loss function $\Phi(\rho_1, \rho_2)$ that measures distances between two conditional densities ρ_1, ρ_2 (e.g., L^2 loss, K-L divergence). Then, the learning problem can be formulated

as the following optimal control problem:

$$\min_{\theta\in\mathcal{U}} \mathbb{E}_{x\sim\overline{\rho}}\left[\Phi(\rho_t(\cdot|x),\rho(\cdot|x)) + \int_0^T L(\theta_t)dt\right],$$
$$\frac{d}{dt}u_t = -\mathrm{div}(f(t,\cdot,\theta_t)u_t), \qquad u_0 = \delta_x. \tag{16}$$

As before, L is a regularizer on the trainable parameters. Now, (16) is an optimal control problem on the function space $\mathcal{H}$.

We now write down formally a set of necessary conditions for optimality, in the form of the Pontryagin's maximum principle, for the present function-space control problem (16). Define the Hamiltonian functional $H : [0,T]\times\mathcal{H}\times\mathcal{H}\times\Theta\to\mathbb{R}$

$$\begin{aligned} H(t,u,v,\theta) &:= -\langle v, \mathrm{div}(f(t,\cdot,\theta)u)\rangle - L(\theta) \\ &= -\int_{\mathbb{R}^d} v(x)\sum_{i=1}^d \frac{\partial}{\partial x_i}(f(t,x,\theta)u(x))dx - L(\theta). \end{aligned}$$

Then, the Pontryagin's maximum principle for this system is expected to take the form: let $\theta^*\in\mathcal{U}$ be an optimal control, then there exists a co-state process $v_t\in\mathcal{H}$ such that

$$\begin{aligned} &\frac{d}{dt}u_t^* = D_v H(t,u_t^*,v_t^*,\theta_t^*), && u_0^* = \delta_x, \\ &\frac{d}{dt}v_t^* = -D_u H(t,u_t^*,v_t^*,\theta_t^*), && v_T^* = -D_u\Phi(\langle\rho_0,u_T^*\rangle,\rho(\cdot|x)) \\ &\mathbb{E}_{x\sim\overline{\rho}}H(t,u_t^*,v_t^*,\theta_t^*) \ge \mathbb{E}_{x\sim\overline{\rho}}H(t,u_t^*,v_t^*,\theta), && \theta\in\Theta, t\in[0,T], \end{aligned}$$

where D denotes the usual Fréchet derivative. Note that by definition, we have $D_v H = -\mathrm{div}(fu)$ and $D_u H = f\cdot\nabla_x v$. Observe that the co-state v^* satisfies the (time-reversed) adjoint Liouville's equation with a specified terminal condition. The PMP for similar functional optimal control problems has been studied in, among others, Pogodaev (2016); Roy and Borz (2017), albeit without the expectation over initial density.

In summary, the advantage of this formulation is that we make no explicit reference to the training data or target functions and formulate the entire problem as a control problem on probability densities. Of course, in practice, to implement an MSA-like algorithm, the terminal condition of the co-state will depend on the target joint density, which we can only access through the sampled data. A rigorous analysis of this function space control formulation and its consequences will be explored in future work.

Appendix B. Proof of Lemma 2

First, observe that assumptions (A1)-(A2) in the main text implies that the second derivatives of f and Φ are bounded by K. Provided that P_t^θ is bounded, they also imply that the second derivatives of H with respect to x and p are bounded when evaluated on $X_t^\theta, P_t^\theta, \theta_t$. We first establish the boundedness of P_t^θ.

Lemma 6 *Assume that (A1)-(A2) hold. Then, there exists a constant $K'>0$ such that for any θ,*

$$\|P_t^\theta\| \le K',$$

for all $t\in[0,T]$.

Proof Using (7) and setting $\tau := T-t$, $\tilde{P}_\tau^\theta := P_{T-\tau}^\theta$ we get

$$\dot{\tilde{P}}_\tau^\theta = \tilde{P}_\tau^\theta\cdot\nabla_x f(t, X_{T-\tau}^\theta, T-\tau), \qquad \tilde{P}_0^\theta = -\nabla\Phi(X_T^\theta).$$

Using (A1)-(A2), we have $\|P_T^\theta\| = \|\nabla_x\Phi(X_T^\theta)\| \le K$ and $\|\nabla_x f(t,X_t^\theta,\theta_t)\|_2 \le K$. Hence,

$$\|\dot{\tilde{P}}_\tau^\theta\| \le K\|\tilde{P}_\tau^\theta\|,$$

and by Gronwall's inequality,

$$\|\tilde{P}_\tau^\theta\| \le Ke^{KT} =: K'.$$

This proves the claim since it holds for any τ. ∎

We now prove Lemma 2. The approach here is similar to that employed in Rozonoer (1959).

Proof [Proof of Lemma 2] From (6) and the definition of the Hamiltonian, we have for any $\theta\in\mathcal{U}$,

$$I(X^\theta,P^\theta,\theta) := \int_0^T P_t^\theta\cdot\dot{X}_t^\theta - H(t,X_t^\theta,P_t^\theta,\theta_t) - L(\theta_t)dt \equiv 0.$$

Denote $\delta X_t = X_t^\phi - X_t^\theta$ and $\delta P_t = P_t^\phi - P_t^\theta$, then we have

$$\begin{aligned} 0 \equiv& I(X^\phi,P^\phi,\phi) - I(X^\theta,P^\theta,\theta) \\ =& \int_0^T P_t^\theta\cdot\delta\dot{X}_t + \delta P_t\cdot\dot{X}_t^\theta + \delta P_t\cdot\delta\dot{X}_t dt \\ &- \int_0^T H(t,X_t^\phi,P_t^\phi,\phi_t) - H(t,X_t^\theta,P_t^\theta,\theta_t)dt \\ &- \int_0^T L(\phi_t) - L(\theta_t)dt. \end{aligned} \tag{17}$$

Now, by integration by parts

$$\int_0^T P_t^\theta\cdot\delta\dot{X}_t dt = P_t^\theta\cdot\delta X_t\Big|_0^T - \int_0^T \dot{P}_t^\theta\cdot\delta X_t dt, \tag{18}$$

$$\int_0^T \delta P_t\cdot\delta\dot{X}_t dt = \delta P_t\cdot\delta X_t\Big|_0^T - \int_0^T \delta\dot{P}_t\cdot\delta X_t dt. \tag{19}$$

Using (6), (7) and (18), we have

$$\begin{aligned} &\int_0^T P_t^\theta\cdot\delta\dot{X}_t + \delta P_t\cdot\dot{X}_t^\theta dt \\ =& P_t^\theta\cdot\delta X_t\Big|_0^T + \int_0^T \Big(f(t,X_t^\theta;\theta_t)\cdot\delta P + \nabla_x H(t,X_t^\theta,P_t^\theta,\theta_t)\cdot\delta X\Big)dt \\ =& P_t^\theta\cdot\delta X_t\Big|_0^T + \int_0^T \Big(\nabla_z H(t,Z_t^\theta,\theta_t)\cdot\delta Z\Big)dt. \end{aligned} \tag{20}$$

where in the last line we defined $Z^\theta := (X^\theta, P^\theta)$. Similarly, from (19) we get

$$\begin{aligned}\int_0^T \delta P_t \cdot \delta \dot{X}_t dt =& \frac{1}{2}\int_0^T \delta P_t \cdot \delta \dot{X}_t dt + \frac{1}{2}\int_0^T \delta P_t \cdot \delta \dot{X}_t dt \\ =& \frac{1}{2}\delta P_t \cdot \delta X_t\Big|_0^T \\ &+ \frac{1}{2}\int_0^T \left([\nabla_z H(t, Z_t^\phi, \phi_t) - \nabla_z H(t, Z_t^\theta, \theta_t)] \cdot \delta Z_t \right) dt \\ =& \frac{1}{2}\delta P_t \cdot \delta X_t\Big|_0^T \\ &+ \frac{1}{2}\int_0^T [\nabla_z H(t, Z_t^\theta, \phi_t) - \nabla_z H(t, Z_t^\theta, \theta_t)] \cdot \delta Z_t dt \\ &+ \frac{1}{2}\int_0^T \delta Z_t \cdot \nabla_z^2 H(t, Z_t^\theta + r_1(t)\delta Z_t, \phi_t) \cdot \delta Z_t dt. \end{aligned} \tag{21}$$

where we have used Taylor's theorem in the last step with $r_1(t) \in [0,1]$. We now rewrite the boundary terms. Since $\delta X_0 = 0$, we have

$$\begin{aligned}&(P_t^\theta + \frac{1}{2}\delta P_t) \cdot \delta X_t\Big|_0^T = (P_T^\theta + \frac{1}{2}\delta P_T) \cdot \delta X_T \\ =& -\nabla\Phi(X_T^\theta) \cdot \delta X_T - \frac{1}{2}(\nabla\Phi(X_T^\phi) - \nabla\Phi(X_T^\theta)) \cdot \delta X_T \\ =& -\nabla\Phi(X_T^\theta) \cdot \delta X_T - \frac{1}{2}\delta X_T \cdot \nabla^2\Phi(X_T^\theta + r_2\delta X_T) \cdot \delta X_T \\ =& -(\Phi(X_T^\phi) - \Phi(X_T^\theta)) - \frac{1}{2}\delta X_T \cdot (\nabla^2\Phi(X_T^\theta + r_2\delta X_T) + \nabla^2\Phi(X_T^\theta + r_3\delta X_T)) \cdot \delta X_T, \end{aligned} \tag{22}$$

for some $r_2, r_3 \in [0,1]$. Lastly, for each $t \in [0,T]$ we have

$$\begin{aligned}H(t, Z_t^\phi, \phi_t) - H(t, Z_t^\theta, \theta_t) =& H(t, Z_t^\theta, \phi_t) - H(t, Z_t^\theta, \theta_t) \\ &+ \nabla_z H(t, Z_t^\theta, \phi_t) \cdot \delta Z_t \\ &+ \frac{1}{2}\delta Z_t \cdot \nabla_z^2 H(t, Z_t^\theta + r_4(t)\delta Z_t, \phi_t) \cdot \delta Z_t, \end{aligned} \tag{23}$$

where $r_4(t) \in [0,1]$.

Substituting (20), (21), (22), (23) into (17), we obtain

$$\begin{aligned}&\left[\Phi(X_T^\phi) + \int_0^T L(\phi_t)\right] - \left[\Phi(X_T^\theta) + \int_0^T L(\theta_t)\right] \\ =& \frac{1}{2}\delta X_T \cdot (\nabla^2\Phi(X_T^\theta + r_2\delta X_T) + \nabla^2\Phi(X_T^\theta + r_3\delta X_T)) \cdot \delta X_T \\ &- \int_0^T \Delta H_{\phi,\theta}(t) dt \\ &+ \frac{1}{2}\int_0^T (\nabla_z H(t, Z_t^\theta, \phi_t) - \nabla_z H(t, Z_t^\theta, \theta_t)) \cdot \delta Z_t dt \\ &+ \frac{1}{2}\int_0^T \left(\delta Z_t \cdot [\nabla_z^2 H(t, Z_t^\theta + r_1(t)\delta Z_t, \phi_t) - \nabla_z^2 H(t, Z_t^\theta + r_4(t)\delta Z_t, \phi_t)] \cdot \delta Z_t \right) dt. \end{aligned} \tag{24}$$

The left hand side is simply $J(\phi) - J(\theta)$, and so it remains to estimate the right hand side terms. First, let us estimate δX and δP. By definition,

$$\delta \dot{X}_t = f(t, X_t^\phi, \phi_t) - f(t, X_t^\theta, \theta_t).$$

Integrating, we get

$$\delta X_t = \int_0^t f(t, X_s^\phi, \phi_s) - f(t, X_s^\theta, \theta_s) ds,$$

and so

$$\begin{aligned}\|\delta X_t\| \le& \int_0^t \|f(t, X_s^\phi, \phi_s) - f(t, X_s^\theta, \theta_s)\| ds \\ \le& \int_0^t \|f(t, X_s^\phi, \phi_s) - f(t, X_s^\theta, \phi_s)\| ds \\ &+ \int_0^t \|f(t, X_s^\theta, \phi_s) - f(t, X_s^\theta, \theta_s)\| ds \\ \le& \int_0^T \|f(t, X_s^\theta, \phi_s) - f(t, X_s^\theta, \theta_s)\| ds \\ &+ K\int_0^t \|\delta X_s\| dt. \end{aligned} \tag{25}$$

By Gronwall's inequality, we have

$$\|\delta X_t\| \le e^{KT}\int_0^T \|f(t, X_s^\theta, \phi_s) - f(t, X_s^\theta, \theta_s)\| ds. \tag{26}$$

To estimate δP, we use the same substitution as in Lemma 6 with $\tau = T - t$ and $\tilde{\cdot}_\tau = \cdot_{T-t}$. We get

$$\delta \tilde{P}_\tau = \delta \tilde{P}_0 + \int_0^\tau \nabla_x H(t, \tilde{X}_s^\phi, \tilde{P}_s^\phi, \tilde{\phi}_s) - \nabla_x H(t, \tilde{X}_s^\theta, \tilde{P}_s^\theta, \tilde{\theta}_s) ds,$$

and hence using Lemma 6 and assumptions (A1)-(A2),

$$\begin{aligned}\|\delta \tilde{P}_\tau\| \le& \|\delta \tilde{P}_0\| + \int_0^\tau \|\nabla_x H(t, \tilde{X}_s^\phi, \tilde{P}_s^\phi, \tilde{\phi}_s) - \nabla_x H(t, \tilde{X}_s^\theta, \tilde{P}_s^\theta, \tilde{\theta}_s)\| ds \\ \le& K\|\delta X_T\| + KK'\int_0^T \|\delta X_s\| ds + K\int_0^\tau \|\delta \tilde{P}_s\| ds \\ &+ \int_0^T \|\nabla_x H(t, X_s^\theta, P_s^\theta, \phi_s) - \nabla_x H(t, X_s^\theta, P_s^\theta, \theta_s)\| ds \\ \le& e^{KT}K(\|\delta X_T\| + K'\int_0^T \|\delta X_s\| ds) \\ &+ e^{KT}\int_0^T \|\nabla_x H(t, X_s^\theta, P_s^\theta, \phi_s) - \nabla_x H(t, X_s^\theta, P_s^\theta, \theta_s)\| ds. \end{aligned} \tag{27}$$

Using estimate (26), we obtain

$$\begin{aligned}\|\delta P_t\| \leq & e^{2KT}K(1+K'T)\int_0^T \|f(t,X_s^\theta,\phi_s)-f(t,X_s^\theta,\theta_s)\|ds \\ & + e^{KT}\int_0^T \|\nabla_x H(t,X_s^\theta,P_s^\theta,\phi_s)-\nabla_x H(t,X_s^\theta,P_s^\theta,\theta_s)\|ds. \end{aligned} \tag{28}$$

Now, we substitute estimates (26) and (28) into (24) and rename constants for simplicity. Note that by assumptions (A1)-(A2) and Lemma 6, all the second derivative terms are bounded element-wise by some constant K''. Hence, we have $|\delta Z_t \cdot A \cdot \delta Z_t| \leq K''\|\delta Z\|^2$ for each A being a second derivative matrix. Thus we obtain

$$\begin{aligned} J(\phi)-J(\theta) \leq & -\int_0^T \Delta H_{\phi,\theta}(t)dt \\ & + \frac{1}{2}K''\|\delta X_T\|^2 \\ & + K''\int_0^T (\|\delta X_t\|^2+\|\delta P_t\|^2)dt \\ & + \frac{1}{2}\int_0^T \|\delta X_t\|\|f(t,X_t^\theta,\phi_t)-f(t,X_t^\theta,\theta_t)\|dt \\ & + \frac{1}{2}\int_0^T \|\delta P_t\|\|\nabla_x H(t,X_t^\theta,P_t^\theta,\phi_t)-\nabla_x H(t,X_t^\theta,P_t^\theta,\theta_t)\|dt \\ \leq & -\int_0^T \Delta H_{\phi,\theta}(t)dt \\ & + C\left(\int_0^T \|f(t,X_t^\theta,\phi_t)-f(t,X_t^\theta,\theta_t)\|dt\right)^2 \\ & + C\left(\int_0^T \|\nabla_x H(t,X_t^\theta,P_t^\theta,\phi_t)-\nabla_x H(t,X_t^\theta,P_t^\theta,\theta_t)\|^2 dt\right)^2 \\ \leq & -\int_0^T \Delta H_{\phi,\theta}(t)dt \\ & + C\int_0^T \|f(t,X_t^\theta,\phi_t)-f(t,X_t^\theta,\theta_t)\|^2 dt \\ & + C\int_0^T \|\nabla_x H(t,X_t^\theta,P_t^\theta,\phi_t)-\nabla_x H(t,X_t^\theta,P_t^\theta,\theta_t)\|^2 dt. \end{aligned} \tag{29}$$

■

Remark 7 *For applications, the global Lipschitz condition (A2) w.r.t. x on f may be restrictive. Note that this can be replaced by a local Lipschitz condition if we can show that X_t, $t \in [0,T]$ is bounded for all $\theta \in \mathcal{U}$. This is true if the parameter space Θ is bounded, which we can safely assume in practice, as long as a suitable regularization is used that prevents the parameters from getting arbitrarily large. Alternatively, a projection step can be used to restrict the parameters to a bounded set. In either cases, this should not negatively affect the performance of the model.*

References

Vladimir V Aleksandrov. On the accumulation of perturbations in the linear systems with two coordinates. *Vestnik MGU*, 3, 1968.

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, pages 3981–3989, 2016.

Michael Athans and Peter L Falb. *Optimal control: an introduction to the theory and its applications*. Courier Corporation, 2013.

Atilim G Baydin, Barak A Pearlmutter, Alexey A Radul, and Jeffrey M Siskind. Automatic differentiation in machine learning: a survey. *arXiv preprint arXiv:1502.05767*, 2015.

Mokhtar S Bazaraa, Hanif D Sherali, and Chitharanjan M Shetty. *Nonlinear programming: theory and algorithms*. John Wiley & Sons, 2013.

Richard Bellman. *Dynamic programming*. Courier Corporation, 2013.

Yoshua Bengio. Learning deep architectures for AI. *Foundations and trends in Machine Learning*, 2(1):1–127, 2009.

Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena scientific Belmont, MA, 1995.

Dimitri P Bertsekas. *Nonlinear programming*. Athena scientific Belmont, 1999.

John T Betts. Survey of numerical methods for trajectory optimization. *Journal of Guidance control and dynamics*, 21(2):193–207, 1998.

Vladimir Grigor'evich Boltyanskii, Revaz Valer'yanovich Gamkrelidze, and Lev Semenovich Pontryagin. The theory of optimal processes. i. the maximum principle. Technical report, TRW SPACE TECHNOLOGY LABS LOS ANGELES CALIF, 1960.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

Alberto Bressan and Benedetto Piccoli. Introduction to mathematical control theory. *AIMS series on applied mathematics, Philadelphia*, 2007.

Arthur Earl Bryson. *Applied optimal control: optimization, estimation and control*. CRC Press, 1975.

Anatolii B Butkovsky. Necessary and sufficient optimality conditions for sampled-data control systems. *Avtomat. i Telemekh*, 24(8):1056–1064, 1963.

Bo Chang, Lili Meng, Eldad Haber, Lars Ruthotto, David Begert, and Elliot Holtham. Reversible architectures for arbitrarily deep residual neural networks. *arXiv preprint arXiv:1709.03698*, 2017.

Felix L Chernousko and Alexey A Lyubushin. Method of successive approximations for solution of optimal control problems. *Optimal Control Applications and Methods*, 3(2): 101–114, 1982.

Francis Clarke. The maximum principle in optimal control, then and now. *Control and Cybernetics*, 34(3):709, 2005.

Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in Neural Information Processing Systems*, pages 3123–3131, 2015.

Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to + 1 or - 1. *arXiv preprint arXiv:1602.02830*, 2016.

Wojciech M Czarnecki, Grzegorz Świrszcz, Max Jaderberg, Simon Osindero, Oriol Vinyals, and Koray Kavukcuoglu. Understanding synthetic gradients and decoupled neural interfaces. *arXiv preprint arXiv:1703.00522*, 2017.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul): 2121–2159, 2011.

Weinan E. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 5(1):1–11, 2017.

Thomas Frerix, Thomas Möllenhoff, Michael Moeller, and Daniel Cremers. Proximal backpropagation. *arXiv preprint arXiv:1706.04638*, 2017.

J Willard Gibbs. *Elementary principles in statistical mechanics*. Courier Corporation, 2014.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Eldad Haber and Lars Ruthotto. Stable architectures for deep neural networks. *arXiv preprint arXiv:1705.03341*, 2017.

Richard HR Hahnloser, Rahul Sarpeshkar, Misha A Mahowald, Rodney J Douglas, and H Sebastian Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947, 2000.

Hubert Halkin. A maximum principle of the pontryagin type for systems described by nonlinear difference equations. *SIAM Journal on control*, 4(1):90–111, 1966.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Magnus R Hestenes. Multiplier and gradient methods. *Journal of optimization theory and applications*, 4(5):303–320, 1969.

R Jackson and F Horn. On discrete analogues of pontryagin's maximum principle. *International Journal of Control*, 1(4):389–395, 1965.

Max Jaderberg, Wojciech M Czarnecki, Simon Osindero, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. Decoupled neural interfaces using synthetic gradients. *arXiv preprint arXiv:1608.05343*, 2016.

Robert I Jennrich. Asymptotic properties of non-linear least squares estimators. *The Annals of Mathematical Statistics*, 40(2):633–643, 1969.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.

Henry J Kelley. Gradient theory of optimal flight paths. *Ars Journal*, 30(10):947–954, 1960.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Ivan A Krylov and Felix L Chernousko. On the method of successive approximations for solution of optimal control problems. *J. Comp. Mathem. and Mathematical Physics*, 2 (6), 1962.

Yann LeCun. A theoretical framework for back-propagation. In *The Connectionist Models Summer School*, volume 1, pages 21–28, 1988.

Yann LeCun. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

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

Kwang Y Lee and Mohamed A El-Sharkawi. *Modern heuristic optimization techniques: theory and applications to power systems*, volume 39. John Wiley & Sons, 2008.

Qianxiao Li, Cheng Tai, and Weinan E. Stochastic modified equations and adaptive stochastic gradient algorithms. In *International Conference on Machine Learning*, pages 2101–2110, 2017.

Daniel Liberzon. *Calculus of variations and optimal control theory: a concise introduction*. Princeton University Press, 2012.

Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

Alexey A Lyubushin. Modifications of the method of successive approximations for solving optimal control problems. *USSR Computational Mathematics and Mathematical Physics*, 22(1):29–34, 1982.

Zbigniew Nahorski, Hans F Ravn, and René Victor Valqui Vidal. The discrete-time maximum principle: a survey and some new results. *International Journal of Control*, 40(3): 533–554, 1984.

Nikolay Pogodaev. Optimal control of continuity equations. *Nonlinear Differential Equations and Applications*, 23(2):21, 2016.

Lev S Pontryagin. *Mathematical theory of optimal processes*. CRC Press, 1987.

Anil V Rao. A survey of numerical methods for optimal control. *Advances in the Astronautical Sciences*, 135(1):497–528, 2009.

Hannes Risken. Fokker-planck equation. In *The Fokker-Planck Equation*, pages 63–95. Springer, 1996.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

Sanford M Roberts and Jerome S Shipman. Two-point boundary value problems: shooting methods. *SIAM Rev.*, 16(2):265266, 1972.

Souvik Roy and Alfio Borz. Numerical investigation of a class of liouville control problems. *J Sci Comput*, 73:178, 2017.

Lev I Rozonoer. The maximum principle of L.S. Pontryagin in optimal-system theory. *Automation and Remote Control*, 20(10):11, 1959.

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

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.

Gavin Taylor, Ryan Burmeister, Zheng Xu, Bharat Singh, Ankit Patel, and Tom Goldstein. Training neural networks without gradients: A scalable ADMM approach. In *International Conference on Machine Learning*, pages 2722–2731, 2016.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

Journal of Machine Learning Research 18 (2018) 1-43 Submitted 10/14; Revised 5/17; Published 5/18

Gradient Hard Thresholding Pursuit

Xiao-Tong Yuan XTYUAN@NUIST.EDU.CN
B-DAT Lab, Nanjing University of Information Science and Technology
Nanjing 210044, China

Ping Li PINGLI98@GMAIL.COM
Baidu Research USA
Bellevue, WA 98004, USA

Tong Zhang TONGZHANG@TONGZHANG-ML.ORG
Tencent AI Lab
Shenzhen 518057, China

Editor: Yoram Singer

Abstract

Hard Thresholding Pursuit (HTP) is an iterative greedy selection procedure for finding sparse solutions of underdetermined linear systems. This method has been shown to have strong theoretical guarantee and impressive numerical performance. In this article, we generalize HTP from compressed sensing to a generic problem setup of sparsity-constrained convex optimization. The proposed algorithm iterates between a standard gradient descent step and a hard-thresholding step with or without debiasing. We analyze the parameter estimation and sparsity recovery performance of the proposed method. Extensive numerical results confirm our theoretical predictions and demonstrate the superiority of our method to the state-of-the-art greedy selection methods in sparse linear regression, sparse logistic regression and sparse precision matrix estimation problems.[1]

Keywords: Hard Thresholding Pursuit, Sparsity Recovery, Greedy Selection

1. Introduction

In the past decade, high-dimensional data analysis has received broad research interest in data mining and scientific discovery, with many significant results obtained in theory, algorithm and application. The major driving force is the rapid development of data collection technologies in many application domains such as social networks, natural language processing, bioinformatics and computer vision. In these applications it is not unusual that data samples are represented with millions or even billions of features using which an underlying statistical learning model must be fit. In many circumstances, however, the number of collected samples is substantially smaller than the dimensionality of features, implying that consistent estimators cannot be hoped for unless additional assumptions are imposed on the model. One of the most popular prior assumptions is that the data exhibit low-dimensional structure, which can often be captured by imposing sparsity constraint on model parameter space. It is thus crucial to develop robust and efficient computational procedures for high-dimensional estimation with sparsity constraint.

1. A conference version of this work appeared in ICML 2014 (Yuan et al., 2014).

In this article, we consider the following generic sparsity-constrained loss minimization problem:

$$\min_{x\in\mathbb{R}^p} f(x), \quad \text{s.t. } \|x\|_0 \leq k, \tag{1}$$

where $f : \mathbb{R}^p \mapsto \mathbb{R}$ is a smooth convex loss function and $\|x\|_0$ denotes the number of nonzero entries in parameter vector x. Among others, several popular examples falling into this framework include: (i) Sparsity-constrained linear regression model (Tropp & Gilbert, 2007) where the residual error is used to measure data reconstruction error; (ii) Sparsity-constrained logistic regression model (Bahmani et al., 2013) where the sigmoid loss is used to measure prediction error; (iii) Sparsity-constrained graphical model learning (Jalali et al., 2011) where the likelihood of samples drawn from an underlying probabilistic model is used to measure data fidelity.

Due to the presence of cardinality constraint $\|x\|_0 \leq k$, problem (1) is generally NP-hard even for the quadratic loss function (Natarajan, 1995). Thus, one must instead seek approximate solutions. For the special case of (1) with least squares error loss in compressed sensing (Donoho, 2006), a number of low-complexity greedy pursuit methods have been studied including matching pursuit (MP) (Mallat & Zhang, 1993), orthogonal matching pursuit (OMP) (Pati et al., 1993), iterative hard thresholding (IHT) (Blumensath & Davies, 2009), compressed sampling matching pursuit (CoSaMP) (Needell & Tropp, 2009) and hard thresholding pursuit (HTP) (Foucart, 2011) to name a few. These algorithms successively select the position of nonzero entries and estimate their values via exploring the residual error from the previous iteration. Comparing to those first-order convex optimization methods developed for ℓ_1-regularized sparse learning (Beck & Teboulle, 2009; Langford et al., 2009; Agarwal et al., 2012), these greedy pursuit algorithms often exhibit more attractive computational efficiency and scalability in practice.

The least squares error used in compressed sensing, however, is not an appropriate measure of discrepancy in a variety of applications beyond signal processing. For example, in statistical machine learning the log-likelihood function is commonly used in logistic regression (Bishop, 2006) and graphical model learning (Jalali et al., 2011; Ravikumar et al., 2011). Thus, it is desirable to investigate theory and algorithms applicable to a broader class of sparse learning problems as formulated by (1). To this end, several forward selection algorithms have been proposed to select the nonzero entries in a sequential fashion (Kim & Kim, 2004; Shalev-Shwartz et al., 2010; Yuan & Yan, 2013; Jaggi, 2011). This category of methods dates back to the Frank-Wolfe method (Frank & Wolfe, 1956). In the meanwhile, the forward greedy selection method has been generalized to convex loss minimization over the linear hull of a collection of atoms (Tewari et al., 2011; Yuan & Yan, 2013). To make the greedy selection procedure more adaptive, Zhang (2008) proposed a forward-backward algorithm which takes backward steps adaptively whenever beneficial. Jalali et al. (2011) have applied this forward-backward selection method to learn the sparse structure of graphical model. Bahmani et al. (2013) proposed a gradient support pursuit method that generalizes CoSaMP from compressed sensing to the generic sparse minimization problem (1). Jain et al. (2014) presented and analyzed several HTP/IHT-style algorithms for high-dimensional sparse estimation. In the paper of Blumensath (2013), a nonlinear-IHT algorithm was investigated in the generic setting of sparsity-constrained loss minimization. Recently, the extensions of HTP/IHT-style methods to structured and stochastic sparse estimation have

been extensively studied in machine learning community (Jain et al., 2016; Li et al., 2016; Shen & Li, 2016; Liu et al., 2017; Nguyen et al., 2017).

1.1 Overview of Our Contribution

In this article, inspired by the success of Hard Thresholding Pursuit (HTP) (Foucart, 2011, 2012) in compressed sensing, we propose and analyze the Gradient Hard Thresholding Pursuit (GraHTP) method to encompass the sparse estimation problems arising from applications with general nonlinear models. At each iteration, GraHTP performs standard gradient descent followed by a hard thresholding operation which first selects the top k (in magnitude) entries of the resultant vector and then (optionally) conducts debiasing on the selected entries. We show that in various settings with or without assuming RIP-type conditions, GraHTP has strong theoretical guarantees analogous to HTP in terms of parameter estimation accuracy.

Apart from the accuracy of objective value and parameter estimation, in many applications such as compressed sensing and graphical models learning, one property of central importance for sparse estimation is the recovery of sparsity pattern, which corresponds to the set of indices of nonzero components of the model parameters. Once the sparsity pattern is recovered, computing the actual nonzero coefficients just boils down to solving a convex minimization problem over the supporting indices. For perfect measurements, the results obtained by Foucart (2011) show that under proper conditions HTP can exactly recover the underlying true model parameters. For noisy models, however, the sparsity recovery analysis is a crucial challenge remains unsolved for HTP-style methods. As a core contribution of this work, we provide a systematic sparsity recovery analysis for GraHTP. Since the output of GraHTP is always k-sparse, the parameter estimation error bounds established in this article roughly imply a sufficient condition for sparsity recovery: as long as the smallest (in magnitude) nonzero entry of a k-sparse target model is larger than the estimation error bound, exact recovery of such a target model can be guaranteed. With more insightful analysis, we further derive some refined sparsity recovery results for GraHTP and for the k-sparse minimizer of problem (1) as well. Some preliminary results on sparsity recovery of GraHTP have been presented in a prior work of ours (Yuan et al., 2016), which we have improved largely in this article.

Comparing to the prior analysis for HTP-style methods, the merits of our main results can be distilled to the following two aspects:

- **Parameter estimation accuracy analysis with/without RIP-type conditions.** Our parameter estimation accuracy analysis for GraHTP simultaneously covers the setting where the target solution is an arbitrary k-sparse solution for which the RIP-type conditions are required, and the setting where the target solution is certain $\bar{k}$-sparse solutions with $\bar{k} \ll k$ for which the RIP-type conditions can be waived;

- **Systematic sparsity recovery analysis.** We extensively investigate the sparsity recovery performance of GraHTP which is of great importance and practical vale in many sparse learning applications including compressed sensing and graphical models learning.

Results	Target Solution	RIP Cond. Free	Sparsity Recovery
(Foucart, 2011)	True k-sparse signal x	$\times$	$\times$
(Blumensath, 2013)	$x^\star = \arg\min_{\|x\|_0 \le k} f(x)$	$\times$	$\times$
(Jain et al., 2014)	$\bar{x} = \arg\min_{\|x\|_0 \le \bar{k}} f(x)$ for proper $\bar{k} \ll k$	$\surd$	$\times$
This Work	$\bar{x}$ with $\|\bar{x}\|_0 \le k$	$\times$ (for $\|\bar{x}\|_0 = k$), $\surd$ (for $\|\bar{x}\|_0 \ll k$)	$\surd$

Table 1: Comparison between the results obtained in this work and several representative prior results for HTP-style algorithms.

Table 1 summarizes a high level comparison between our results and several representative state-of-the-art results for HTP-style algorithms, in terms of target solution, dependence on RIP-type conditions, and sparsity recovery analysis.

We have applied GraHTP to sparse linear regression, sparse logistic regression and sparse precision matrix estimation problems, with its algorithm and/or theory substantialized for these models. Empirically we demonstrate that GraHTP is competitive to the state-of-the-art greedy selection methods in these sparse learning problems.

1.2 Notation

In the following, x is a vector, A is a matrix, and F is an index set. The following notations will be used in this article.

- $[x]_i$: the ith entry of vector x.
- x_F: the restriction of x on F, i.e., $[x_F]_i = [x]_i$ if $i \in F$, and $[x_F]_i = 0$ otherwise.
- x_k: the restriction of x on its top k (in modulus) entries.
- $\|x\| = \sqrt{x^\top x}$: the Euclidean norm of x.
- $\|x\|_1 = \sum_i |[x]_i|$: the ℓ_1-norm of x.
- $\|x\|_\infty = \max_i |[x]_i|$: the ℓ_∞-norm of x.
- $\|x\|_0$: the number of nonzero entries of x.
- $\mathrm{supp}(x)$: the index set of nonzero entries of x.
- $\mathrm{supp}(x, k)$: the index set of the top k (in modulus) entries of x.
- $x_{\min} = \min_{i \in \mathrm{supp}(x)} |[x]_i|$: the smallest absolute value of nonzero element of x.
- $[A]_{ij}$: the element on the ith row and jth column of matrix A.

- $\|A\| = \sup_{\|x\|\leq 1} \|Ax\|$: the spectral norm of matrix A.
- $|A|_\infty = \max_{i,j} |[A]_{ij}|$: the element-wise ℓ_∞-norm of A.
- $\mathrm{Tr}(A)$: the trace (sum of diagonal elements) of a square matrix A.
- A_F: the restriction of A on index set F.
- A^-: the restriction of a square matrix A on its off-diagonal entries.
- $\mathrm{vect}(A)$: (column wise) vectorization of a matrix A.
- $\lambda_{\max}(A,k) = \max_{\|x\|=1,\|x\|_0\leq k} x^\top Ax$: the largest k-sparse eigenvalue of a positive semi-definite matrix A.
- $\lambda_{\min}(A,k) = \min_{\|x\|=1,\|x\|_0\leq k} x^\top Ax$: the smallest k-sparse eigenvalue of a positive semi-definite matrix A.

1.3 Organization

This article proceeds as follows: We present in Section 2 the GraHTP algorithm. The parameter estimation error and exact sparsity recovery guarantees of GraHTP are respectively analyzed in Section 3 and Section 4. The implications of GraHTP in linear regression, logistic regression and Gaussian graphical model learning are discussed in Section 5. Monte-Carlo simulations and real data experimental results are presented in Section 6. We conclude this article in Section 7.

2. Algorithm

GraHTP is an iterative greedy selection procedure for approximately optimizing the nonconvex problem (1). A high level summary of GraHTP is described in the top panel of Algorithm 1. The procedure generates a sequence of intermediate k-sparse vectors $x^{(0)}, x^{(1)}, \ldots$ from an initial sparse approximation $x^{(0)}$ (typically $x^{(0)} = 0$). At the t-th iteration, the first step (**S1**), $\tilde{x}^{(t)} = x^{(t-1)} - \eta\nabla f(x^{(t-1)})$, computes the gradient descent at the point $x^{(t-1)}$ with step-size η. Then in the second step (**S2**), the k coordinates of the vector $\tilde{x}^{(t)}$ that have the largest magnitude are chosen as the support in which pursuing the minimization will be most effective. In the third step (**S3**), we find a vector with this support which minimizes the objective function, which becomes $x^{(t)}$. This last step, which is often referred to as *debiasing*, has been shown to improve the performance in other algorithms too (Yuan & Zhang, 2013; Bahmani et al., 2013). The iterations continue until the algorithm reaches certain terminating condition, e.g., the difference of objective value or model parameters between adjacent iterations converges. A more intuitive criterion is $F^{(t)} = F^{(t-1)}$ (see (**S2**) for the definition of $F^{(t)}$), since then $x^{(\tau)} = x^{(t)}$ for all $\tau \geq t$, although there is no guarantee that this should occur in general cases. It will be assumed throughout the article that the sparsity level k is known. In practice this integer parameter may be tuned via, for example, cross-validation in supervised learning tasks.

In the standard form of GraHTP, the debiasing step (**S3**) requires to minimize $f(x)$ over the supporting set $F^{(t)}$. If this step is judged too costly, we may consider instead a

fast variant of GraHTP, where the debiasing is replaced by a simple truncation operation $x^{(t)} = \tilde{x}^{(t)}_k$. This leads to the Fast GraHTP (FGraHTP) as described in the bottom panel of Algorithm 1, which can be understood as a projected gradient descent procedure for optimizing the nonconvex minimization problem (1). Up to the cost of truncation operation, its per-iteration computational overload is almost identical to that of the standard gradient descent procedure. The iteration procedure of FGraHTP is also known as the nonlinear-IHT algorithm (Blumensath, 2013). Comparing to that prior work, our analysis for FGraHTP is more comprehensive and the results are tighter especially in sparsity recovery analysis. While in this article we only study the FGraHTP outlined in Algorithm 1, we should mention that other fast variants of GraHTP can also be considered. For instance, to reduce the computational cost of the debiasing step (**S3**), we can take a restricted Newton step or a restricted gradient descent step to calculate $x^{(t)}$.

We close this section by pointing out that, in the special case where the squares error $f(x) = \frac{1}{2}\|y - Ax\|^2$ is the cost function, GraHTP reduces to HTP (Foucart, 2011). Specifically, the gradient descent step (**S1**) reduces to $\tilde{x}^{(t)} = x^{(t-1)} + \eta A^\top(y - Ax^{(t-1)})$ and the debiasing step (**S3**) reduces to the orthogonal projection $x^{(t)} = \arg\min\{\frac{1}{2}\|y-Ax\|^2, \mathrm{supp}(x) \subseteq F^{(t)}\}$. In the meanwhile, FGraHTP reduces to IHT (Blumensath & Davies, 2009), which is also known as Gradient Descent with Sparsification (Garg & Khandekar, 2009), of which the iteration is defined as $x^{(t)} = (x^{(t-1)} + \eta A^\top(y - Ax^{(t-1)}))_k$.

Algorithm 1: Gradient Hard Thresholding Pursuit (GraHTP).

Initialization: $x^{(0)}$ *with* $\|x^{(0)}\|_0 \leq k$ *(typically* $x^{(0)} = 0$*)*, $t = 1$.
Output: $x^{(t)}$.
repeat
  (**S1**) Compute $\tilde{x}^{(t)} = x^{(t-1)} - \eta\nabla f(x^{(t-1)})$;
  (**S2**) Let $F^{(t)} = \mathrm{supp}(\tilde{x}^{(t)}, k)$ be the indices of $\tilde{x}^{(t)}$ with the largest k absolute values;
  (**S3**) Compute $x^{(t)} = \arg\min\{f(x); \mathrm{supp}(x) \subseteq F^{(t)}\}$;
  $t = t + 1$;
until *halting condition holds*;

★ *Fast GraHTP* ★

repeat
  Compute $\tilde{x}^{(t)} = x^{(t-1)} - \eta\nabla f(x^{(t-1)})$;
  Compute $x^{(t)} = \tilde{x}^{(t)}_k$ as the truncation of $\tilde{x}^{(t)}$ with top k (in magnitude) entries preserved;
  $t = t + 1$;
until *halting condition holds*;

3. Parameter Estimation Analysis

In this section, we analyze the parameter estimation accuracy of GraHTP/FGraHTP. To simplify notation, we abbreviate $\nabla_F f = (\nabla f)_F$ and $\nabla_s f = (\nabla f)_s$. Our analysis relies on the conditions of Restricted Strong Convexity/Smoothness (RSC/RSS) which are conven-

tionally used in the analysis of greedy sparse optimization methods (Shalev-Shwartz et al., 2010; Bahmani et al., 2013; Jain et al., 2014).

Definition 1 (Restricted Strong Convexity/Smoothness) *For any integer $s > 0$, we say $f(x)$ is restricted m_s-strongly convex and M_s-smooth if there exist $m_s, M_s > 0$ such that*

$$\frac{m_s}{2}\|x-y\|^2 \leq f(x) - f(y) - \langle \nabla f(y), x-y\rangle \leq \frac{M_s}{2}\|x-y\|^2, \quad \forall \|x-y\|_0 \leq s. \tag{2}$$

The ratio number M_s/m_s, which measures the curvature of the loss function over sparse subspaces, will be referred to as *restricted strong condition number* in this article.

3.1 Main Results

The following theorem is our main result on the parameter estimation accuracy of GraHTP and FGraHTP with respect to arbitrary k-sparse target solutions. A proof of this theorem is provided in Appendix B.1.

Theorem 2 *Assume that f is M_{3k}-smooth and m_{3k}-strongly convex. Let $\bar{x}$ be an arbitrary k-sparse vector and $\rho = \sqrt{1 - 2\eta m_{3k} + \eta^2 M_{3k}^2}$.*

(a) Assume that $M_{3k}/m_{3k} < 2\sqrt{3}/3$ and the step-size η is chosen such that $\rho < 0.5$. Then GraHTP outputs $x^{(t)}$ satisfying

$$\|x^{(t)} - \bar{x}\| \leq \mu_1^t \|x^{(0)} - \bar{x}\| + \frac{2.83\eta\sqrt{k}}{1-2\rho}\|\nabla f(\bar{x})\|_\infty,$$

where $\mu_1 = \rho/(1-\rho) \in (0,1)$.

(b) Assume that $M_{3k}/m_{3k} < 1.26$ and the step-size η is chosen such that $\rho < 0.62$. Then FGraHTP outputs $x^{(t)}$ satisfying

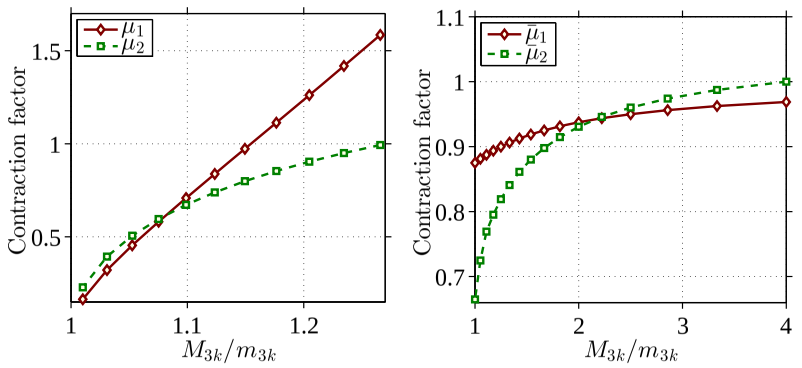
$$\|x^{(t)} - \bar{x}\| \leq \mu_2^t \|x^{(0)} - \bar{x}\| + \frac{2.81\eta\sqrt{k}}{1-1.62\rho}\|\nabla f(\bar{x})\|_\infty,$$

where $\mu_2 = 1.62\rho \in (0,1)$.

In the part (a) of Theorem 2, the contraction factor $\mu_1 < 1$ controls the convergence rate of GraHTP. The condition $\rho < 0.5$ requires the step-size to be selected according to

$$\frac{2m_{3k} - \sqrt{4m_{3k}^2 - 3M_{3k}^2}}{2M_{3k}^2} < \eta < \frac{2m_{3k} + \sqrt{4m_{3k}^2 - 3M_{3k}^2}}{2M_{3k}^2}, \tag{3}$$

from which we can see that $M_{3k}/m_{3k} < 2\sqrt{3}/3$ is a necessary condition to guarantee the existence of η such that $\rho < 0.5$ and $\mu_1 < 1$. The condition of $\rho < 0.5$ is analogous to the RIP condition for estimation from noisy measurements in compressed sensing (Candès et al., 2006; Needell & Tropp, 2009; Foucart, 2011). Indeed, in compressed sensing, GraHTP reduces to HTP which requires weaker RIP condition than prior compressed sensing algorithms. The condition in (3) also suggests that the value of η should be bounded from



(a) Theorem 2 (b) Theorem 5

Figure 1: Evolving curves of the contraction factors in Theorem 2 and Theorem 5.

above to guarantee convergence, and be bounded away from zero to avoid early stopping as well. Similarly, $M_{3k}/m_{3k} < 1.26$ in the part(b) is a necessary condition to guarantee the existence of η such that $\rho < 0.62$ and $\mu_2 < 1$. Figure 1(a) shows the evolving curves of contraction factors μ_1 and μ_2 as functions of M_{3k}/m_{3k} in the interval $[1, 1.26)$. It can be seen from this figure that $\mu_1 < \mu_2$ when $M_{3k}/m_{3k} \to 1$ and $\mu_1 > \mu_2$ for relatively larger M_{3k}/m_{3k}.

The non-vanishing terms in the error bounds of Theorem 2 indicate that the estimation errors of GraHTP and FGraHTP are controlled by the multiplier of $\sqrt{k}\|\nabla f(\bar{x})\|_\infty$. Particularly if the sparse vector $\bar{x}$ is sufficiently close to an unconstrained minimum of f, then the estimation error floor is negligible because $\|\nabla f(\bar{x})\|_\infty$ has small magnitude. The following corollary is a direct consequence of Theorem 2 which shows that exact support recovery is possible when $\bar{x}_{\min}$ is significantly larger than $\sqrt{k}\|\nabla f(\bar{x})\|_\infty$.

Corollary 3 *Assume the conditions in Theorem 2 hold.*

(a) Let $\bar{x}$ be an arbitrary k-sparse vector satisfying $\bar{x}_{\min} > \frac{5.66\eta\sqrt{k}}{1-2\rho}\|\nabla f(\bar{x})\|_\infty$. Then GraHTP will output $x^{(t)}$ satisfying $supp(x^{(t)}) = supp(\bar{x})$ after $t = \left\lceil \frac{1}{\mu_1} \ln\left(\frac{2\|x^{(0)} - \bar{x}\|}{\bar{x}_{\min}}\right)\right\rceil$ steps of iteration.

(b) Let $\bar{x}$ be an arbitrary k-sparse vector satisfying $\bar{x}_{\min} > \frac{5.62\eta\sqrt{k}}{1-1.62\rho}\|\nabla f(\bar{x})\|_\infty$. Then FGraHTP will output $x^{(t)}$ satisfying $supp(x^{(t)}) = supp(\bar{x})$ after $t = \left\lceil \frac{1}{\mu_2} \ln\left(\frac{2\|x^{(0)} - \bar{x}\|}{\bar{x}_{\min}}\right)\right\rceil$ steps of iteration.

Indeed, given the conditions in Corollary 3, for both GraHTP and FGraHTP we can show that $\|x^{(t)} - \bar{x}\| < \bar{x}_{\min}$ and thus $\text{supp}(x^{(t)}) = \text{supp}(\bar{x})$ must hold as $x^{(t)}$ and $\bar{x}$ are both k-sparse vectors.

Remark 4 *Corollary 3 shows that GraHTP/FGraHTP requires RIP-type conditions as in Theorem 2 to guarantee exact support recovery. As a comparison, the existing sparsity recovery results for ℓ_1-estimators (Wainwright, 2009; Li et al., 2015) are free of RIP-type conditions but instead relying on the irrepresentablility condition which is known to be stronger. For example, a case where the RIP-type condition holds while the irrepresentability condition does not was given by Van De Geer & Bühlmann (2009, Example 10.4).*

The RIP-type conditions assumed in Theorem 2 could still be restrictive in real-life high-dimensional statistical settings wherein pairs of variables can be arbitrarily correlated. In the following theorem, we further show that by properly relaxing sparsity levels, GraHTP and FGraHTP are able to accurately estimate parameters without assuming bounded restricted strong condition numbers. A proof of this theorem is deferred to Appendix B.2.

Theorem 5 *Let $\bar{x}$ be an arbitrary $\bar{k}$-sparse vector with $\bar{k} \leq k$. Assume that $s = 2k + \bar{k} < p$.*

(a) Assume that f is M_{2k}-smooth and m_{2k}-strongly convex. Assume the step-size $\eta < 1/M_{2k}$. If $k \geq \left(2 + \frac{4}{\eta^2 m_{2k}^2}\right)\bar{k}$, then GraHTP outputs $x^{(t)}$ satisfying

$$\|x^{(t)} - \bar{x}\| \leq \sqrt{\frac{2\bar{\mu}_1^t \bar{\triangle}^{(0)}}{m_{2k}}} + \frac{2.83\sqrt{k}\|\nabla f(\bar{x})\|_\infty}{m_{2k}},$$

where $\bar{\mu}_1 = 1 - \eta m_{2k}(1 - \eta M_{2k})/2$ and $\bar{\triangle}^{(0)} = \max\{f(x^{(0)}) - f(\bar{x}), 0\}$.

(b) Assume that f is M_s-smooth and m_s-strongly convex. Assume the step-size $\eta < 2m_s/M_s^2$ such that $\rho = \sqrt{1 - 2\eta m_s + \eta^2 M_s^2} < 1$. If $k > \rho\bar{k}/(1-\rho)^2$, then FGraHTP outputs $x^{(t)}$ satisfying

$$\|x^{(t)} - \bar{x}\| \leq \bar{\mu}_2^t \|x^{(0)} - \bar{x}\| + \frac{\gamma\eta\sqrt{s}}{1 - \bar{\mu}_2}\|\nabla f(\bar{x})\|_\infty,$$

where $\bar{\mu}_2 = \rho\gamma \in (0, 1)$ and $\gamma = \sqrt{1 + \left(\bar{k}/k + \sqrt{(4 + \bar{k}/k)\bar{k}/k}\right)/2}$.

Remark 6 *When using step-size $\eta = \frac{1}{2M_{2k}}$, the part(a) of Theorem 5 tells that GraHTP converges linearly towards an arbitrary $\bar{k}$-sparse vector $\bar{x}$ if the sparsity level is chosen as $k \geq \left(2 + \frac{16M_{2k}^2}{m_{2k}^2}\right)\bar{k}$. The estimation error is controlled by the multiplier of $\sqrt{k}\|\nabla f(\bar{x})\|_\infty$. Similarly, the part(b) of Theorem 5 establishes the convergence result of FGraHTP with proper relaxed $k \gg \bar{k}$. Note that the condition $k > \rho\bar{k}/(1-\rho)^2$ in part(b) actually enforces the contraction factor $\bar{\mu}_2 < 1$. Figure 1(b) shows the evolving curves of contraction factors $\bar{\mu}_1$ and $\bar{\mu}_2$ as functions of M_{3k}/m_{3k}, with the same target sparsity $\bar{k}$. We can see from this figure that $\bar{\mu}_2$ is superior to $\bar{\mu}_1$ when M_{3k}/m_{3k} is relatively small.*

The following corollary of Theorem 5 shows that GrHTP/FGraHTP with certain relaxed sparsity levels can guarantee $\text{supp}(\bar{x}) \subseteq \text{supp}(x^{(t)})$ without assuming RIP-type conditions.

Corollary 7 *Let $\bar{x}$ be an arbitrary $\bar{k}$-sparse vector with $\bar{k} \leq k$.*

(a) Under the conditions in Theorem 5(a), if $\bar{x}_{\min} > \frac{5.66\sqrt{k}}{m_{2k}}\|\nabla f(\bar{x})\|_\infty$, then GraHTP will output $x^{(t)}$ satisfying $\text{supp}(\bar{x}) \subseteq \text{supp}(x^{(t)})$ after $t = \left\lceil \frac{1}{\bar{\mu}_1} \ln\left(\frac{8\bar{\triangle}^{(0)}}{m_{2k}\bar{x}_{\min}^2}\right)\right\rceil$ steps of iteration.

(b) Under the conditions in Theorem 5(b), if $\bar{x}_{\min} > \frac{2\gamma\eta\sqrt{s}}{1-\bar{\mu}_2}\|\nabla f(\bar{x})\|_\infty$, then FGraHTP will output $x^{(t)}$ satisfying $\text{supp}(\bar{x}) \subseteq \text{supp}(x^{(t)})$ after $t = \left\lceil \frac{1}{\bar{\mu}_2} \ln\left(\frac{2\|x^{(0)} - \bar{x}\|}{\bar{x}_{\min}}\right)\right\rceil$ steps of iteration.

Indeed, the conditions in Corollary 7 imply $\|x^{(t)} - \bar{x}\| < \bar{x}_{\min}$ which leads to $\text{supp}(\bar{x}) \subseteq \text{supp}(x^{(t)})$. We note that the parameter estimation error bound derived by Jain et al. (2014, Theorem 3) implies a similar support recovery guarantee as in Corollary 7(a).

3.2 Comparison to Prior Results

Now we compare our method and parameter estimation error bounds to some prior relevant methods and results.

Our method versus nonlinear-IHT (Blumensath, 2013). As we remarked in Section 2 that FGraHTP is identical to the nonlinear-IHT method proposed by Blumensath (2013). The estimation error results of the two, however, are different: the error bound of nonlinear-IHT is relying on the objective value at the target solution; whereas ours in Theorem 2(b) is controlled by the infinity norm of gradient at the target solution.

Our method versus ℓ_1-norm ball constrained estimation (Agarwal et al., 2012). It is worthwhile to compare our ℓ_0-estimation results to those established by Agarwal et al. (2012, Theorem 1) for ℓ_1-norm ball constrained M-estimator (maximum likelihood type estimator). Let us consider $\bar{x}$ as the underlying $\bar{k}$-sparse nominal parameter in a statistical model. When using sparsity level $k = \bar{k}$, the $O(\sqrt{\bar{k}}\|\nabla f(\bar{x})\|_\infty)$ estimation error bound in Theorem 2, which is at the same order of statistical error, is essentially identical to the error bound derived by Agarwal et al. (2012, Theorem 1). Our analysis, however, requires a bounding assumption on the restricted strong condition number which is not required in their result. This can be interpreted as the price of using nonconvex sparsity constraint rather than its convex relaxation. By using properly relaxed sparsity level $k = O(\bar{k})$, we obtain similar estimation error bounds in Theorem 5 but without assuming bounded restricted strong condition number. In this case, at a slight sacrifice in sparsity level, our methods gain better dependence on restricted strong condition number than those for convex models. Concerning the efficiency of projection steps, the ℓ_0-projection used in FGraHTP is more efficient than the ℓ_1-projection required by those first-order convex minimization methods. The projection operation of GraHTP is more expensive as it requires an additional debiasing step right after ℓ_0-projection.

Our method versus GraSP (Bahmani et al., 2013). A similar estimation error bound as in Theorem 2(a) has been established for the GraSP method (Bahmani et al., 2013). At time instance t, GraSP first conducts debiasing over the union of the top k entries of $x^{(t-1)}$ and the top $2k$ entries of $\nabla f(x^{(t-1)})$, and then preserves the top k entries of the resultant vector, which becomes $x^{(t)}$. Our GraHTP is connected to GraSP in the sense that the k largest absolute elements after the gradient descent step will come from some combination of the largest elements in $x^{(t-1)}$ and the largest elements in the gradient $\nabla f(x^{(t-1)})$.

Although having similar convergence behavior, the per-iteration cost of GraHTP is cheaper than GraSP: at each iteration, GraSP needs to minimize the objective over a support of size at least $2k$ while that size for GraHTP is k. FGraHTP is even cheaper for iteration as it does not need any debiasing operation. We will compare the actual numerical performance of these methods in the experiment section.

Our results versus the results obtained by Jain et al. (2014). The RIP-condition-free estimation error bound in Theorem 5(a) has also been proved by Jain et al. (2014, Theorem 3) with relaxed sparsity levels. As pointed out in Remark 6, the contraction factor $\bar{\mu}_1$ derived in Theorem 5(a) is inferior to the rate $\bar{\mu}_2$ in Theorem 5(b) when the restricted strong condition number is relatively small. Moreover, from Figure 1(b) we can see that $\bar{\mu}_1$ is valued in a quite restrictive interval $(0.87, 1)$ while $\bar{\mu}_2$ can be varied in a much wider range of $(0.65, 1)$. Figure 1(a) shows that the contraction factors μ_1 and μ_2 derived in Theorem 2 can be widely valued in $(0, 1)$. The more favorable contraction factors in Theorem 5(b) and Theorem 2 are resulted from a more careful analysis of GraHTP/FGraHTP and using a tight hard-thresholding bound derived by Shen & Li (2016).

4. Sparsity Recovery Analysis

In this section, we further analyze the sparsity recovery performance of GraHTP. In Corollary 3 and Corollary 7, we have already established some general sparsity recovery results for GraHTP. Here we will provide a refined analysis without assuming bounded restricted strong condition number. Moreover, we will analyze the sparsity recovery behavior of the sparse estimator $x^\star = \arg\min_{\|x\|_0 \le k} f(x)$ which to our knowledge has not been addressed elsewhere in literature. The main results obtained in this section are highlighted in below:

- For GraHTP algorithm, we derive in Theorem 8 an improved RIP-condition-free result for exactly recovering the support of a target $\bar{k}$-sparse vector with $\bar{k} < k$.
- For the global k-sparse minimizer $x^\star$, we provide in Theorem 10 a set of sufficient conditions under which $x^\star$ is able to recover the support of a target sparse vector.

4.1 Sparsity Recovery of GraHTP

In the following theorem, we show that for proper $k > \bar{k}$, GraHTP is able to recover the support of certain target $\bar{k}$-sparse vector without assuming bounded restricted strong condition numbers. A proof of this theorem is given in Appendix C.1.

Theorem 8 *Assume that f is M_{2k}-smooth and m_{2k}-strongly convex. Let $\bar{x}$ be an arbitrary $\bar{k}$-sparse vector satisfying $k \ge \left(1 + \frac{16M_{2k}^2}{m_{2k}^2}\right)\bar{k}$. Set the step-size to be $\eta = \frac{1}{2M_{2k}}$. If $\bar{x}_{\min} > 2.3\sqrt{\frac{f(\bar{x}) - f(x^\star)}{m_{2k}}}$, then GraHTP will terminate and output $x^{(t)}$ satisfying $supp(x^{(t)}, \bar{k}) = supp(\bar{x})$ after at most*

$$t = \left\lceil \frac{2kM_{2k}}{m_{2k}} \ln \frac{\triangle^{(0)}}{\triangle^{-\star}} \right\rceil$$

steps of iteration, where $\triangle^{(0)} = f(x^{(0)}) - f(x^\star)$ and

$$\triangle^{-\star} = \min_{\|x\|_0 \le k, supp(x) \ne supp(x^\star), f(x) > f(x^\star)} f(x) - f(x^\star).$$

Results	Target Solution	RIP Condition	x-min Condition
Corollary 3(a)	Arbitrary k-sparse $\bar{x}$	Required	$\bar{x}_{\min} > \mathcal{O}\left(\frac{\sqrt{k}\|\nabla f(\bar{x})\|_\infty}{m_{2k}}\right)$
Corollary 7(a)	$\|\bar{x}\|_0 = \mathcal{O}\left((\frac{m_{2k}}{M_{2k}})^2 k\right)$	Free	$\bar{x}_{\min} > \mathcal{O}\left(\frac{\sqrt{k}\|\nabla f(\bar{x})\|_\infty}{m_{2k}}\right)$
Theorem 8	$\|\bar{x}\|_0 = \mathcal{O}\left((\frac{m_{2k}}{M_{2k}})^2 k\right)$	Free	$\bar{x}_{\min} > \mathcal{O}\left(\sqrt{\frac{f(\bar{x}) - f(x^\star)}{m_{2k}}}\right)$

Table 2: Comparison of Theorem 8 against Corollary 3 and Corollary 7.

Remark 9 *The main message conveyed by Theorem 8 is: If $\bar{k} = \mathcal{O}\left(\frac{m_{2k}^2}{M_{2k}^2} k\right)$ and the nonzero elements in $\bar{x}$ are significantly larger than the value $\sqrt{(f(\bar{x}) - f(x^\star))/m_{2k}}$, then GraHTP will output $x^{(t)}$ whose top $\bar{k}$ entries are exactly the supporting set of $\bar{x}$. The implication of this result is that in order to recover certain $\bar{k}$-sparse signals, one may run GraHTP with a properly relaxed sparsity level k until convergence and then preserve the top $\bar{k}$ entries of the k-sparse output as the final estimation.*

In Table 2, we summarize the sparsity recovery results established in Theorem 8, Corollary 3 and Corollary 7. We claim that the x-min condition in Theorem 8 is no stronger than those in Corollary 3 and Corollary 7. Indeed, when $\bar{x} \ne x^\star$, from the restricted strong-convexity of f and the fact $x^\top y \le \|x\|_\infty \|y\|_1$ we can derive the following inequality:

$$f(\bar{x}) - f(x^\star) \le \frac{\|\nabla f(\bar{x})\|_\infty^2 \|\bar{x} - x^\star\|_1^2}{2m_{2k}\|\bar{x} - x^\star\|^2}.$$

It can be verified that the factor $\bar{l} = \|\bar{x} - x^\star\|_1^2 / \|\bar{x} - x^\star\|^2$ is valued in the interval $[1, k + \bar{k}]$ if $\bar{x} \ne x^\star$. Since $k > \bar{k}$, we then always have $\sqrt{(f(\bar{x}) - f(x^\star))/m_{2k}} \le \sqrt{k}\|\nabla f(\bar{x})\|_\infty / m_{2k}$. The closer $\bar{l}$ is to 1, the weaker lower bound condition can be imposed on $\bar{x}_{\min}$ in Theorem 8. In the extreme case when $\bar{l} = 1$, the $\bar{x}_{\min}$ condition becomes $\bar{x}_{\min} > \mathcal{O}(\|\nabla f(\bar{x})\|_\infty / m_{2k})$ which is not dependent on factor $\sqrt{k}$ and thus is weaker than those in Corollary 3 and Corollary 7.

4.2 Sparsity Recovery of $x^\star$

Given a target solution $\bar{x}$, the following result gives some sufficient conditions under which the sparse estimator $x^\star$ is able to exactly recover the supporting set of $\bar{x}$. A proof of this result is provided in Appendix C.2.

Theorem 10 *Assume that f is M_{2k}-smooth and m_{2k}-strongly convex. Let $\bar{x}$ be an arbitrary $\bar{k}$-sparse vector with $\bar{k} \le k$. Then $supp(\bar{x}) = supp(x^\star, \bar{k})$ if either of the following two conditions holds:*

(1) $\bar{x}_{\min} > \frac{4.59\sqrt{k}}{m_{2k}} \|\nabla f(\bar{x})\|_\infty$;

(2) $k \ge \left(1 + \frac{4M_{2k}^2}{m_{2k}^2}\right)\bar{k}$ *and* $\bar{x}_{\min} > 2.3\sqrt{\frac{f(\bar{x}) - f(x^\star)}{m_{2k}}}$.

Remark 11 *Theorem 10 shows that when using sparsity level $k \geq \bar{k}$, the top $\bar{k}$ entries of the k-sparse global minimizer $x^\star$ is exactly the support of $\bar{x}$ if $\bar{x}_{\min}$ is significantly larger than $\sqrt{k}\|\nabla f(\bar{x})\|_\infty/m_{2k}$. By using a more relaxed sparsity level as in condition (2), the top $\bar{k}$ entries of $x^\star$ is exactly the support of $\bar{x}$ when $\bar{x}_{\min}$ is significantly larger than $\sqrt{(f(\bar{x}) - f(x^\star))/m_{2k}}$. Note that Theorem 10 is valid without imposing bounding assumptions on restricted strong condition number.*

We now compare the support recovery result in Theorem 10 for the ℓ_0-estimator (1) to those known for the following ℓ_1-regularized estimator:

$$\min_{x\in\mathbb{R}^p} f(x) + \lambda\|x\|_1, \tag{4}$$

where $f(x)$ is a convex loss function and λ is the regularization strength parameter. When the loss function is quadratic, a set of sufficient conditions were derived by Wainwright (2009) to guarantee exact sparsity recovery of Lasso-type estimators. For more general loss functions, a unified sparsity recovery analysis was presented in the paper of Li et al. (2015). We summarize in below a comparison between Theorem 10 and those sparsity recovery results for ℓ_1-regularized estimators (Li et al., 2015) with respect to several key conditions:

- **Local structured smoothness/convexity condition:** Theorem 10 only requires first-order local structured smoothness/convexity conditions (i.e., RSC/RSS) while the results obtained by Li et al. (2015, Theorem 5.1, Condition 1) rely on certain second-order and third-order local structured smoothness conditions.
- **Irrepresentablility condition:** Theorem 10 is free of the so called irrepresentablility condition which is typically required to guarantee the sparsistency of ℓ_1-regularized estimators (Li et al., 2015, Theorem 5.1, Condition 3).
- **x-min condition:** Comparing to the x-min condition derived by Li et al. (2015, Theorem 5.1, Condition 4) which is of order $\mathcal{O}(\sqrt{k}\|\nabla f(\bar{x})\|_\infty)$, the x-min condition (1) in Theorem 10 is comparable at the same order while the x-min condition (2) is sharper since $\sqrt{f(\bar{x}) - f(x^\star)/m_{2k}} \leq \sqrt{k}\|\nabla f(\bar{x})\|_\infty/m_{2k}$.

We comment that the above key differences also apply to the comparison between Theorem 8 for GraHTP and the sparsity recovery results for ℓ_1-regularized estimators. In Section 5.1, we will further specify our results to the setting of sparse linear regression and make a comparison against those sparsity recovery results for Lasso-type estimators (Wainwright, 2009).

5. Applications to Sparsity-Constrained M-estimation

We now specify GraHTP and its analysis to the M-estimation problem which is a popular formulation in statistical machine learning. Given a set of n independently drawn data samples $\{x^{(i)}\}_{i=1}^n$, the M-estimation problem is defined as to minimize the following empirical risk function averaged over the samples:

$$f(w) = \frac{1}{n}\sum_{i=1}^n \phi(x^{(i)} \mid w),$$

where ϕ is a loss function and w is a set of adjustable parameters. The sparsity-constrained M-estimation problem is then given by

$$\min_w f(w), \quad \text{subject to } \|w\|_0 \leq k. \tag{5}$$

In the subsections to follow, we will consider three instances of this model: linear regression, logistic regression and Gaussian precision matrix estimation.

5.1 Sparsity-constrained Linear Regression

Given a $\bar{k}$-sparse parameter vector $\bar{w}$, we assume the samples are generated according to the linear model $v^{(i)} = \bar{w}^\top u^{(i)} + \varepsilon^{(i)}$ where $\varepsilon^{(i)}$ are n i.i.d. sub-Gaussian random variables with parameter σ. The sparsity-constrained least squares regression model is then given by

$$\min_w f(w) = \frac{1}{2n}\sum_{i=1}^n \|v^{(i)} - w^\top u^{(i)}\|^2, \quad \text{subject to } \|w\|_0 \leq k. \tag{6}$$

In this case, GraHTP (and FGraHTP) reduces to the conventional HTP (and IHT) of which the parameter estimation performance has been extensively studied in compressed sensing (Foucart, 2011; Blumensath & Davies, 2009). Here we illustrate the sparsity recovery results we established in Section 4 and compare them against those for ℓ_1-estimators. Suppose $u^{(i)}$ are drawn from Gaussian distribution with covariance matrix $\Sigma \succ 0$. Then it holds with high probability that $f(w)$ has RSC constant $m_{2k} \geq \lambda_{\min}(\Sigma) - \mathcal{O}(\bar{k}\log p/n)$ and RSS constant $M_{2k} \leq \lambda_{\max}(\Sigma) + \mathcal{O}(\bar{k}\log p/n)$, and $\|\nabla f(\bar{w})\|_\infty = \mathcal{O}\left(\sigma\sqrt{\log p/n}\right)$. Assume that $k \geq \bar{k}$. We summarize in below the implications of our sparsity recovery results in sparse linear regression:

- Sparsity recovery of GraHTP. Corollary 3 shows that if $\bar{w}_{\min} > \mathcal{O}\left(\frac{\sigma\sqrt{k\log p/n}}{\lambda_{\min}(\Sigma)}\right)$ and $\frac{\lambda_{\max}(\Sigma)}{\lambda_{\min}(\Sigma)}$ is well upper bounded, then after sufficient iteration GraHTP and FGraHTP with $k = \bar{k}$ will guarantee support recovery $\mathrm{supp}(x^{(t)}) = \mathrm{supp}(\bar{x})$ with high probability. Corollary 7 indicates that when using certain relaxed sparsity level $k = \mathcal{O}\left(\frac{\lambda^2_{\max}(\Sigma)}{\lambda^2_{\min}(\Sigma)}\bar{k}\right)$, GraHTP and FGraHTP are able to guarantee $\mathrm{supp}(x^{(t)}) \supseteq \mathrm{supp}(\bar{x})$ without assuming bounded condition number. Since $f(\bar{x}) - f(x^\star) \leq \frac{\bar{l}\|\nabla f(\bar{x})\|^2_\infty}{2m_{2k}}$ where $\bar{l} = \|\bar{x} - x^\star\|_1^2/\|\bar{x} - x^\star\|^2 \in [1, k+\bar{k}]$, Theorem 8 implies that if $\bar{w}_{\min} > \mathcal{O}\left(\frac{\sigma\sqrt{\bar{l}\log p/n}}{\lambda_{\min}(\Sigma)}\right)$ and $k = \mathcal{O}\left(\frac{\lambda^2_{\max}(\Sigma)}{\lambda^2_{\min}(\Sigma)}\bar{k}\right)$, then after finite iteration GraHTP will guarantee $\mathrm{supp}(x^{(t)}, \bar{k}) = \mathrm{supp}(\bar{x})$ with high probability.
- Sparsity recovery of the least squares estimator (6). Let $w^\star$ be the global k-sparse minimizer of (6). Theorem 10 shows that $\mathrm{supp}(w^\star, \bar{k}) = \mathrm{supp}(\bar{w})$ holds with high probability if $\bar{w}_{\min} > \mathcal{O}\left(\frac{\sigma\sqrt{k\log p/n}}{\lambda_{\min}(\Sigma)}\right)$. To compare our sparsity recovery results for ℓ_0-estimators against those established by Wainwright (2009, Theorem 1) for Lasso-type estimators, the signal-noise-ratio condition of $\bar{w}_{\min} > \mathcal{O}\left(\frac{\sigma\sqrt{k\log p/n}}{\lambda_{\min}(\Sigma)}\right)$ is shared

in that paper. The key difference is that our analysis is valid without imposing the irrepresentablility condition on design matrix which is required in the sparsity recovery analysis of Lasso-type estimators.

5.2 Sparsity-constrained Logistic Regression

Logistic regression is one of the most popular models in statistical machine learning (Bishop, 2006). In this model the relation between the random feature vector $u \in \mathbb{R}^p$ and its associated random binary label $v \in \{-1, +1\}$ is determined by the conditional probability

$$\mathbb{P}(v|u; \bar{w}) = \frac{\exp(2v\bar{w}^\top u)}{1 + \exp(2v\bar{w}^\top u)}, \tag{7}$$

where $\bar{w} \in \mathbb{R}^p$ denotes parameter vector. Given a set of n independently drawn data samples $\{(u^{(i)}, v^{(i)})\}_{i=1}^n$, logistic regression learns the parameters so as to minimize the following logistic loss function:

$$l(w) := -\frac{1}{n} \log \prod_i \mathbb{P}(u^{(i)} \mid v^{(i)}; w) = \frac{1}{n} \sum_{i=1}^n \log(1 + \exp(-2v^{(i)} w^\top u^{(i)})),$$

which is known to be convex. Unfortunately, in high-dimensional setting, i.e., $n < p$, the problem can be underdetermined and thus its minimum is not unique. A conventional way to handle this issue is to impose ℓ_2-regularization to the logistic loss to avoid singularity. The ℓ_2-penalty, however, does not promote sparse solutions which are often desirable in high-dimensional learning tasks. The sparsity-constrained ℓ_2-regularized logistic regression is then given by

$$\min_w f(w) = l(w) + \frac{\lambda}{2}\|w\|^2, \quad \text{subject to } \|w\|_0 \le k, \tag{8}$$

where $\lambda > 0$ is the regularization strength parameter. Obviously $f(w)$ is λ-strongly convex. The cardinality constraint enforces the solution to be sparse.

Verifying restricted smoothness and strong convexity. Let $U = [u^{(1)}, ..., u^{(n)}] \in \mathbb{R}^{p\times n}$ be the design matrix and $\sigma(z) = 1/(1+\exp(-z))$ be the sigmoid function. In the case of ℓ_2-regularized logistic loss considered in this section we have $\nabla f(w) = Ua(w)/n + \lambda w$ in which the vector $a(w) \in \mathbb{R}^n$ is given by $[a(w)]_i = -2v^{(i)}(1-\sigma(2v^{(i)}w^\top u^{(i)}))$, and the Hessian $\nabla^2 f(w) = U\Lambda(w)U^\top/n + \lambda I$ where $\Lambda(w)$ is an $n \times n$ diagonal matrix whose diagonal entries $[\Lambda(w)]_{ii} = 4\sigma(2v_i w^\top u_i)(1 - \sigma(2v_i w^\top u_i))$. Given an integer s, recall that $\lambda_{\max}(A, s)$ denotes the largest s-sparse eigenvalue of a positive semi-definite matrix A and $\lambda_{\min}(A, s)$ denotes the smallest s-sparse eigenvalue of A. Assume that the algorithm is initialized with all-zero vector. Then it can be verified that $f(w)$ is $(\lambda_{\max}(UU^\top, s)+\lambda)$-smooth and $(\gamma_s+\lambda)$-strongly convex where $\gamma_s := \min_{f(w)\le f(0)} \lambda_{\min}(U\Lambda(w)U^\top, s)$.

Bounding the value of $\|\nabla f(\bar{w})\|_\infty$. We now bound the infinity norm $\|\nabla f(\bar{w})\|_\infty$ which controls the estimation error and sparisty recovery bounds of GraHTP/FGraHTP. In the following derivation, we assume that the joint density of the random vector $(u, v) \in \mathbb{R}^{p+1}$ is given by the following exponential family distribution:

$$\mathbb{P}(u, v; \bar{w}) = \exp\left(v\bar{w}^\top u + B(u) - A(\bar{w})\right), \tag{9}$$

where

$$A(\bar{w}) := \log \sum_{v=\{-1,1\}} \int_{\mathbb{R}^p} \exp\left(v\bar{w}^\top u + B(u)\right) du$$

is the log-partition function. The term $B(u)$ characterizes the marginal behavior of u. Obviously, the conditional distribution of v given u, $\mathbb{P}(v \mid u; \bar{w})$, is given by the Bernoulli distribution in (7). By doing some elementary manipulations (see, e.g., Wainwright & Jordan, 2008) we can obtain the following standard result which shows that the first derivative of the logistic log-likelihood $l(w)$ yields the cumulants of the random variables $v[u]_j$:

$$\frac{\partial l}{\partial [w]_j} = \frac{1}{n} \sum_{i=1}^n \left\{-v^{(i)}[u^{(i)}]_j + \mathbb{E}_v[v[u^{(i)}]_j \mid u^{(i)}]\right\}. \tag{10}$$

Here the expectation $\mathbb{E}_v[\cdot \mid u]$ is taken over the conditional distribution (7). We introduce the following sub-Gaussian condition on the random variate $v[u]_j$.

Assumption 1 *For all j, we assume that there exists constant $\sigma > 0$ such that for all ζ,*

$$\mathbb{E}[\exp(\zeta v[u]_j)] \le \exp\left(\sigma^2\zeta^2/2\right).$$

This assumption holds when $[u]_j$ are sub-Gaussian (e.g., Gaussian or bounded) random variables. The following result establishes the bound of $\|\nabla f(\bar{w})\|_\infty$.

Proposition 12 *If Assumption 1 holds, then with probability at least $1 - 4p^{-1}$,*

$$\|\nabla f(\bar{w})\|_\infty \le 4\sigma\sqrt{\ln p/n} + \lambda\|\bar{w}\|_\infty.$$

A proof of this result is provided in Appendix D.1. If we choose $\lambda = O(\sqrt{\ln p/n})$, then with overwhelming probability $\|\nabla f(\bar{w})\|_\infty$ vanishes at the rate of $O(\sqrt{\ln p/n})$. This bound is superior to the bound obtained by Bahmani et al. (2013, Section 4.2) which is not vanishing as sample size increases. Based on the above discussion, we can similarly specify our parameter estimation and sparsity recovery results to sparse logistic regression. Here we omit the detailed specification of results for the sake of redundance reducing.

5.3 Sparsity-constrained Gaussian Precision Matrix Estimation

As an important class of sparse learning problems for exploring the interrelationship among a large number of random variables, the sparse Gaussian precision (inverse covariance) matrix estimation problem has received significant interest in a variety of scientific and engineering domains, including computational biology, natural language processing and document analysis.

Let x be a p-variate random vector with zero-mean Gaussian distribution $\mathcal{N}(0, \bar{\Sigma})$. Its density is parameterized by the precision matrix $\bar{\Omega} = \bar{\Sigma}^{-1} \succ 0$ as

$$\phi(x; \bar{\Omega}) = \frac{1}{\sqrt{(2\pi)^p (\det \bar{\Omega})^{-1}}} \exp\left(-\frac{1}{2} x^\top \bar{\Omega} x\right).$$

It is well known that the conditional independence between the variables $[x]_i$ and $[x]_j$ given $\{[x]_k, k \ne i, j\}$ is equivalent to $[\bar{\Omega}]_{ij} = 0$. The conditional independence relations between

components of x, on the other hand, can be represented by a graph $\mathcal{G} = (V, E)$ in which the vertex set V has p elements corresponding to $[x]_1, ..., [x]_p$, and the edge set E consists of edges between node pairs $\{[x]_i, [x]_j\}$. The edge between $[x]_i$ and $[x]_j$ is excluded from E if and only if $[x]_i$ and $[x]_j$ are conditionally independent given other variables. This graphical model is known as Gaussian Markov random field (GMRF) (Edwards, 2000). Thus for multivariate Gaussian distribution, estimating the support of the precision matrix $\bar{\Omega}$ is equivalent to learning the structure of GMRF $\mathcal{G}$.

Given i.i.d. samples $\mathbb{X}_n = \{x^{(i)}\}_{i=1}^n$ drawn from $\mathcal{N}(0, \bar{\Sigma})$, the negative log-likelihood, up to a constant, can be written in terms of the precision matrix as

$$\mathcal{L}(\mathbb{X}_n; \bar{\Omega}) := -\log\det\bar{\Omega} + \langle\Sigma_n, \bar{\Omega}\rangle,$$

where Σ_n is the sample covariance matrix. We are interested in the problem of estimating a sparse precision $\bar{\Omega}$ with no more than a pre-specified number of off-diagonal nonzero entries. For this purpose, we consider the following cardinality constrained log-determinant program:

$$\min_{\Omega\succ 0} L(\Omega) := -\log\det\Omega + \langle\Sigma_n, \Omega\rangle, \quad \text{s.t. } \|\Omega^-\|_0 \le 2k, \tag{11}$$

where Ω^- is the restriction of Ω on the off-diagonal entries, $\|\Omega^-\|_0 = |\text{supp}(\Omega^-)|$ is the cardinality of the supporting set of Ω^- and the integer $k > 0$ controls the number of edges, i.e., $|E|$, in the graph.

Verifying restricted smoothness and strong convexity. It can be verified that the Hessian matrix of $L(\Omega)$ is given by $\nabla^2 L(\Omega) = \Omega^{-1} \otimes \Omega^{-1}$, where $\otimes$ denotes the Kronecker product operator. Suppose that $\|\Omega^-\|_0 \le s$ and $\alpha_s I \preceq \Omega \preceq \beta_s I$ for some $0 < \alpha_s \le \beta_s$. Due to the fact that the eigenvalues of Kronecker products of symmetric matrices are the products of the eigenvalues of their factors, it holds that $\beta_s^{-2} I \preceq \Omega^{-1} \otimes \Omega^{-1} \preceq \alpha_s^{-2} I$. Therefore we have $\beta_s^{-2} \le \|\nabla^2 L(\Omega)\| \le \alpha_s^{-2}$ which implies that $L(\Omega)$ is β_s^{-2}-strongly convex and α_s^{-2}-smooth. Inspired by this property, we consider applying GraHTP to the following variant of problem (11):

$$\min_{\alpha I\preceq\Omega\preceq\beta I} L(\Omega), \quad \text{s.t. } \|\Omega^-\|_0 \le 2k, \tag{12}$$

where $0 < \alpha \le \beta$ are two constants which respectively lower and upper bound the eigenvalues of the desired solution. To roughly estimate α and β, we employ a rule proposed by Lu (2009, Proposition 3.1) for the ℓ_1-regularized log-determinant program. Specifically, we set

$$\alpha = (\|\Sigma_n\|_2 + n\xi)^{-1}, \quad \beta = \xi^{-1}(n - \alpha\text{Tr}(\Sigma_n)),$$

where ξ is a small enough positive number (e.g., $\xi = 10^{-2}$ as used in our implementation).

Bounding the value of $|\nabla L(\bar{\Omega})|_\infty$. It is standard to know that $|\nabla L(\bar{\Omega})|_\infty = |\Sigma_n - \bar{\Sigma}|_\infty = \mathcal{O}(\sqrt{\log p/n})$ with probability at least $1 - c_0 p^{-c_1}$ for some positive constants c_0 and c_1 and sufficiently large n (see, e.g., Ravikumar et al., 2011, Lemma 1). Therefore, with overwhelming probability we have $|\nabla L(\bar{\Omega})|_\infty = \mathcal{O}(\sqrt{\log p/n})$ when n is sufficiently large.

A Modified GraHTP. Note that GraHTP is not directly applicable to the problem (12) due to the presence of the constraint $\alpha I \preceq \Omega \preceq \beta I$ in addition to the sparsity constraint. To address this issue, we need to accordingly modify the debiasing step (**S3**) of GraHTP to minimize $L(\Omega)$ over the constraints $\alpha I \preceq \Omega \preceq \beta I$ and $\text{supp}(\Omega) \subseteq F^{(t)}$:

$$\min_{\alpha I\preceq\Omega\preceq\beta I} L(\Omega), \quad \text{s.t. } \text{supp}(\Omega) \subseteq F^{(t)}. \tag{13}$$

Since this problem is convex, any off-the-shelf convex solver can be applied for optimization. In our implementation, we resort to the alternating direction method of multipliers (ADMM) (Boyd et al., 2010; Yuan, 2012) which has been observed to be efficient in our numerical practice. The implementation details of ADMM for solving the subproblem (13) are deferred to Appendix D.2. The modified GraHTP for sparse Gaussian precision matrix estimation is outlined in Algorithm 2.

Algorithm 2: A Modified GraHTP for Sparse Gaussian Precision Matrix Estimation.

Initialization: $\Omega^{(0)}$ *with* $\|(\Omega^{(0)})^-\|_0 \le 2k$ *and* $\alpha I \preceq \Omega^{(0)} \preceq \beta I$ *(typically* $\Omega^{(0)} = \alpha I$*)*, $t = 1$.
Output: $\Omega^{(t)}$.
repeat
- (**S1**) Compute $\tilde{\Omega}^{(t)} = \Omega^{(t-1)} - \eta\nabla L(\Omega^{(t-1)})$;
- (**S2**) Let $\tilde{F}^{(t)} = \text{supp}((\tilde{\Omega}^{(t)})^-, 2k)$ be the indices of $(\tilde{\Omega}^{(t)})^-$ with the largest $2k$ absolute values and $F^{(t)} = \tilde{F}^{(t)} \cup \{(1,1), ..., (p,p)\}$;
- (**S3**) Compute $\Omega^{(t)} = \arg\min\left\{L(\Omega); \alpha I \preceq \Omega \preceq \beta I, \text{supp}(\Omega) \subseteq F^{(t)}\right\}$;
- $t = t + 1$;

until *halting condition holds*;

6. Experimental Results

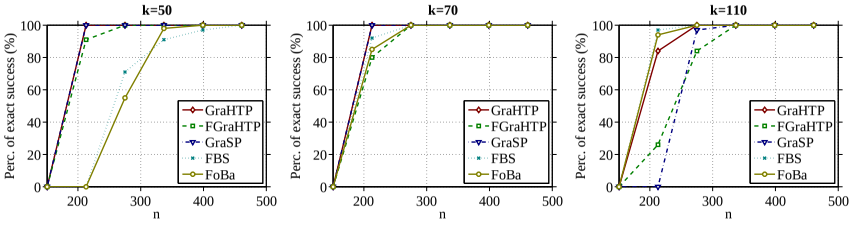
This section is devoted to illustrating the empirical performance of GraHTP/FGraHTP when applied to sparse learning tasks. Our algorithms are implemented in Matlab 7.12 running on a desktop with Intel Core i7 3.2G CPU and 16G RAM.

6.1 Sparsity-constrained Linear Regression

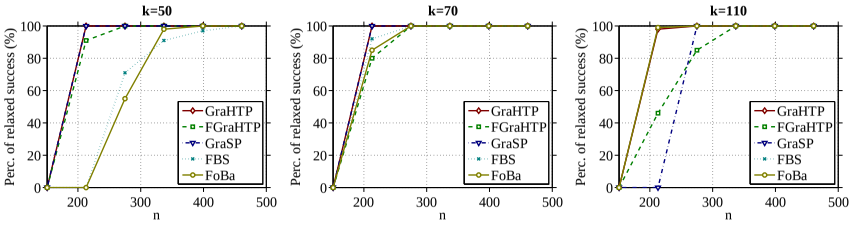
We conduct a group of Monte-Carlo simulation experiments on sparse linear regression model to verify the sparsity recovery results presented in Section 4.

Data generation. We consider a synthetic data model in which the sparse parameter $\bar{w}$ is a $p = 500$ dimensional vector that has $\bar{k} = 50$ nonzero entries drawn independently from a Gaussian distribution with significant mean. Each data sample u is a normally distributed dense vector. The responses are generated by $v = \bar{w}^\top u + \varepsilon$ where ε is a standard Gaussian noise. We allow the sample size n to be varying and for each n, we generate 100 random copies of data independently.

Baselines and evaluation metric. We test GraHTP and FGraHTP with varying sparsity level $k \ge \bar{k}$ and compare their performance with three state-of-the-art greedy selection methods: GraSP (Bahmani et al., 2013), FBS (Yuan & Yan, 2013) and FoBa (Zhang, 2008). As we have mentioned, GraSP is also a hard-thresholding-type method. This method



(a) Exact support recovery



(b) Relaxed support recovery

Figure 2: Sparse linear regression on simulated data: chance of success curves for support recovery under varying sample size and sparsity level.

simultaneously selects at each iteration k nonzero entries and update their values via exploring the top k entries in the previous iterate as well as the top $2k$ entries in the previous gradient. FBS is a forward-selection-type method which iteratively selects an atom from the dictionary and minimizes the objective function over the linear combinations of all the selected atoms. FoBa is an adaptive forward-backward greedy selection algorithm which allows elimination of selected variables when the objective value does not increase significantly. We use two metrics to measure the support recovery performance. We say a *relaxed support recovery* is successful if $\mathrm{supp}(\bar{w}) \subseteq \mathrm{supp}(w^{(t)})$ and an *exact support recovery* is successful if $\mathrm{supp}(\bar{w}) = \mathrm{supp}(w^{(t)}, \bar{k})$. We replicate the experiment over the 100 trials and record the percentage of relaxed success and percentage of exact success for each configuration of the pair (n, k).

Results. Figure 2 shows the percentage of exact (relaxed) success curves as functions of sample size n, under different sparsity levels $k \in \{50, 70, 110\}$. From these curves we can make the following observations:

- For each curve, the chance of success increases as sample size n increases. This is as expected because the larger sample size is, the easier the x-min conditions can be fulfilled so as to guarantee exact support recovery;
- GraHTP is superior to FGraHTP for sparsity recovery, especially when using sparsity level $k > \bar{k}$ and relatively small sample size. This indicates that the debiasing step conducted in GraHPT can significantly improve the accuracy of sparsity recovery, especially in noisy settings.
- The left panel of Figure 2 shows that when $k = \bar{k}$, GraHTP/FGraHTP and GraSP are comparable and they all significantly outperform FBS and FoBa, especially when the sample size is relatively small. This observation suggests that hard-thresholding-type methods are more accurate than forward and/or backward selection methods for sparsity recovery with exact sparsity level. The middle panel shows that for slightly increased sparsity level $k = 70$, GraHTP and GraSP still exhibit superior performance, while the performance gap among all the considered algorithms decreases. From the right panel we can see that for relatively large $k > \bar{k}$, FBS, Foba and GraHTP have much better performance than FGraHTP and GraSP.

From the above observations we conclude that GraHTP is able to achieve better trade-off between accuracy and stability of sparsity recovery than the other considered methods.

6.2 Sparsity-constrained Logistic Regression

We present in this subsection the experimental results on several synthetic and real-data sparse logistic regression tasks.

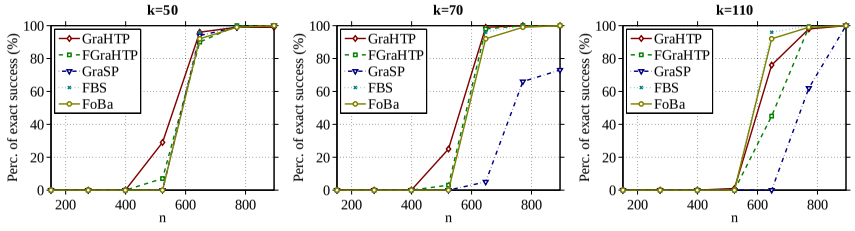
6.2.1 Monte-Carlo simulation

In this group of Monte-Carlo experiments, we use a simulated data to verify the sparsity recovery performance of GraHTP and FGraHTP on logistic regression model. The sparse parameter and design matrix are generated in an identical way to that of the linear regression model. The data labels, $v \in \{-1, 1\}$, are generated randomly according to the Bernoulli distribution $\mathbb{P}(v = 1|u; \bar{w}) = \exp(2\bar{w}^\top u)/(1 + \exp(2\bar{w}^\top u))$. The same experiment protocol as used in the previous linear regression setting applies here. Inspired by Theorem 8 and the discussion in Section 5.2, we set the step-size $\eta = \frac{1}{2M_{2k}}$ where $M_{2k} = \lambda_{\max}(UU^\top, 2k) + \lambda$. The sparse eigenvalue $\lambda_{\max}(UU^\top, 2k)$ can be computed using the truncated power method (Yuan & Zhang, 2013).

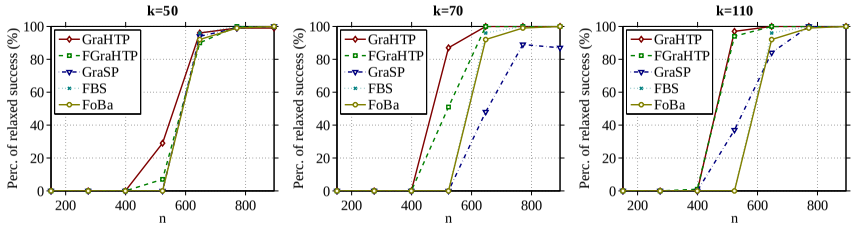
Results. For different sparsity levels $k \geq \bar{k}$, Figure 3 shows the chance of exact (relaxed) success curves as functions of sample size n. Again, from these curves we can observe that: 1) in a wide range of sparsity level, GraHP achieves better trade-off between accuracy and stability than the other considered sparsity recovery methods; and 2) GraHTP consistently outperforms FGraHTP in noisy settings when using $k > \bar{k}$.

6.2.2 Real data experiments

We further illustrate the performance of GraHTP/FGraHTP on real data for binary logistic regression. The data used for evaluation include two *dense* data sets gisette (Guyon et al., 2005) and breast cancer (Hess et al., 2006), and two *sparse* data sets rcv1.binary (Lewis et al., 2005) and news20.binary (Keerthi & DeCoste, 2005). Table 3 summaries the statistics of these data sets. For each data set, we test with sparsity parameters $k \in \{100, 200, ..., 1000\}$ and fix the regularization parameter $\lambda = 10^{-5}$. We initialize $w^{(0)} = 0$ and set the stopping criterion as $\|w^{(t)} - w^{(t-1)}\|/\|w^{(t-1)}\| \leq 10^{-4}$.



(a) Exact support recovery



(b) Relaxed support recovery

Figure 3: Sparse logistic regression on simulated data: chance of success curves for support recovery under varying sample size and sparsity level.

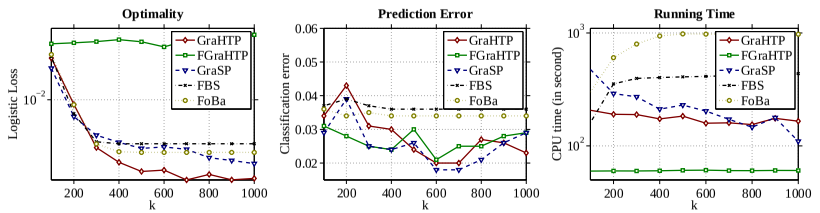
Datasets	Training Size	Testing Size	Dimensionality
gisette	6,000	1,000	5,000
breast cancer	54	79	22,283
rcv1.binary	20,242	20,000	47,236
news20.binary	10,000	9,996	1,355,191

Table 3: Statistics of data sets used in binary logistic regression experiment.

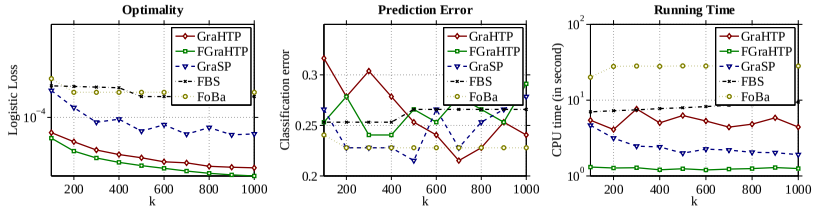
Results. The objective value, test classification error and CPU running time curves under varying sparsity level k are plot in Figure 4. From these curves we have the following observations:

- On optimality: GraHTP is superior to the other considered algorithms in most cases. FGraHTP is less optimal on gisette data, while it is comparable to the other algorithms on the other three data sets.

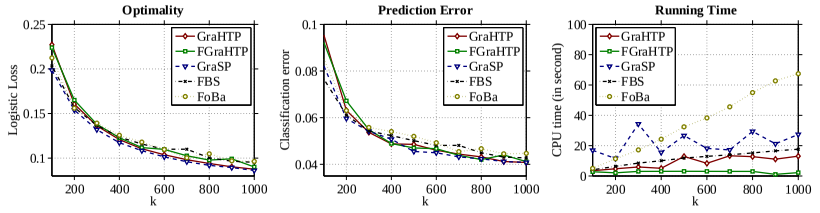
- On classification accuracy: GraHTP and GraSP are comparable to each other and they are slightly superior to the other algorithms in most cases; FGraHTP is average in classification accuracy in most cases.



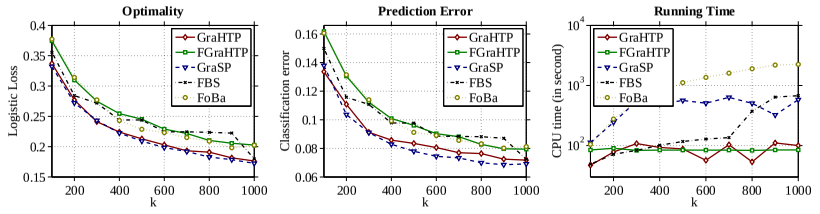
(a) gisette



(b) breast cancer

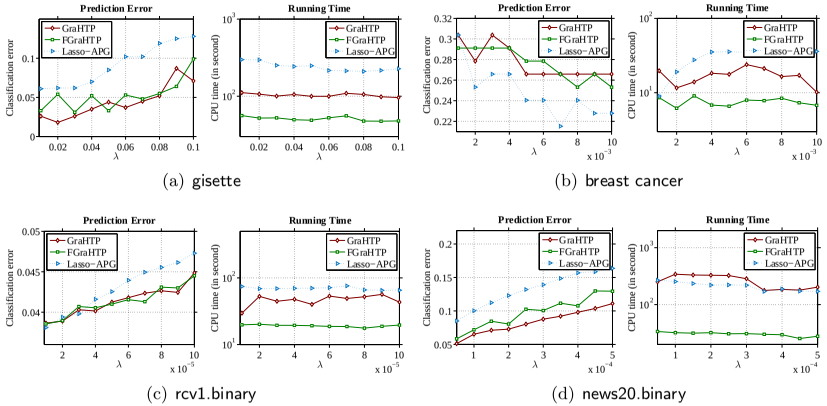


(c) rcv1.binary



(d) news20.binary

Figure 4: Sparse logistic regression on real data: objective value, classification error and CPU running time curves under varying sparsity level.



(a) gisette (b) breast cancer

(c) rcv1.binary (d) news20.binary

Figure 5: Sparse logistic regression on real data: comparison between GraHTP/FGraHTP and Lasso-type estimator in classification error and CPU running time.

- On execution time: FGraHTP is the most efficient one and GraHTP is the runner-up except on breast cancer. Particularly, as shown in Figure 4(d) that the computational advantage of FGraHTP/GraHTP over the other considered methods becomes significant on news20.binary which is relatively large in scale.

To summarize, GraHTP and FGraHTP are able to achieve desirable trade-off between accuracy and efficiency on the considered data sets.

Comparison against Lasso-type estimator. We have also conducted a set of experiments to compare GraHTP/FGraHTP against the Lasso-type estimator (4) for ℓ_1-regularized sparse learning. To make a fair comparison, we first solve the Lasso-type estimator (4) using an accelerated proximal gradient method (Beck & Teboulle, 2009), which we call Lasso-APG, and then run GraHTP with the sparsity level of the Lasso-APG solution. Figure 5 shows the test classification error and CPU running time curves under varying regularization parameter λ. We can observe from this group of results that: (1) GraHTP and FGraHTP outperform Lasso-APG in classification accuracy on three out of the four data sets in use; and (2) FGraHTP is the most efficient one on all the data sets and GraHTP is faster than Lasso-APG on three of the data sets. Based on these observations, we can conclude that GraHTP and FGraHTP tend to be more accurate and efficient than Lasso-type estimator when their output solutions are at the same sparsity level.

6.3 Sparsity-constrained Gaussian Precision Matrix Estimation

We further assess the performance of GraHTP/FGraHTP when applied to sparse precision matrix estimation.

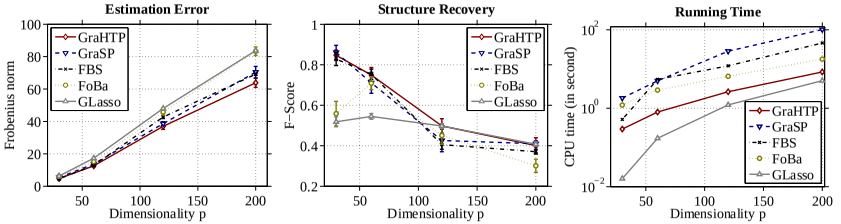


Figure 6: Sparse precision matrix estimation on simulated data: Matrix Frobenius norm loss, support recovery F-score and CPU running time curves under varying data dimensionality. The larger the F-score, the better the support recovery performance.

6.3.1 Monte-Carlo Simulation

Our simulation study employs the sparse precision matrix model $\bar{\Omega} = \Theta + \sigma I$ where each off-diagonal entry in Θ is generated independently and equals 1 with probability $P = 0.1$ or 0 with probability $1 - P = 0.9$. Θ has zeros on the diagonal, and σ is chosen so that the condition number of $\bar{\Omega}$ is p. Let $\bar{\Sigma} = \bar{\Omega}^{-1}$ be the covariance matrix. We generate a training sample of size $n = 100$ from $\mathcal{N}(0, \bar{\Sigma})$, and an independent sample of size 100 from the same distribution for tuning the parameter k. The numerical performance is evaluated with different values of $p \in \{30, 60, 120, 200\}$, replicated 100 times each.

We compare the modified GraHTP (as outlined in Algorithm 2) with GraSP, FBS and FoBa. To adopt GraSP to sparse precision matrix estimation, we modify the algorithm with a similar two-stage strategy as used in the modified GraHTP such that it can handle the eigenvalue bounding constraint in addition to the sparsity constraint. FBS and FoBa have already been applied to sparse precision matrix estimation problems in literature (Yuan & Yan, 2013; Jalali et al., 2011). Also, we compare GraHTP with Graphical Lasso (GLasso) which is one of the representative Lasso-type convex estimators for ℓ_1-penalized log-determinant program (Friedman et al., 2008). The quality of precision matrix estimation is measured by its distance to the truth in Frobenius norm and the support recovery F-score. The larger the F-score, the better the support recovery performance.

Figure 6 compares the matrix error in Frobenius norm, support recovery F-score and CPU running time achieved by each of the considered algorithms for different p. The results show that GraHTP performs favorably in terms of estimation error and support recovery accuracy. We note from the error bars in the curves that the standard error (in 100 replication) of GraHTP is relatively larger than Glasso. This is because GraHTP approximately solves a nonconvex problem via greedy selection at each iteration; the procedure is less stable than those convex solvers such as GLasso. Similar phenomenon of instability has also been observed for the other considered ℓ_0-estimators. The right panel of Figure 6 shows the computational time of the considered algorithms. We can see that GLasso is more efficient than the four greedy selection methods. Although inferior to GLasso, GraHTP is still computationally more attractive than the other considered greedy selection solvers.

Methods	Specificity	Sensitivity	MCC	CPU Time (sec.)
GraHTP	0.77 (0.11)	0.77 (0.19)	**0.49** (0.19)	1.92
GraSP	0.73 (0.10)	**0.78** (0.18)	0.45 (0.17)	4.06
FBS	0.78 (0.11)	0.74 (0.18)	0.48 (0.19)	8.73
FoBa	0.72 (0.11)	**0.78** (0.18)	0.44 (0.18)	6.73
GLasso	**0.81** (0.11)	0.64 (0.21)	0.45 (0.19)	**1.19**

Table 4: Sparse precision matrix estimation on breast cancer data: comparison of average (std) classification accuracy and average CPU running time over 100 replications.

6.3.2 Real Data

We consider the task of LDA (linear discriminant analysis) classification of tumors using the breast cancer data set. This data consists of 133 subjects, each of which is associated with 22,283 gene expression levels. Among these subjects, 34 are with pathological complete response (pCR) and 99 are with residual disease (RD). The pCR subjects are considered to have a high chance of cancer free survival in the long term. Based on the estimated precision matrix of the gene expression levels, we apply LDA to predict whether a subject can achieve the pCR state or the RD state.

Experiment protocol. In this experiment, we follow the same protocol as what was used in the paper of Cai et al. (2011). The data are randomly divided into the training and test sets. In each random division, 5 pCR subjects and 16 RD subjects are randomly selected to constitute the test data, and the remaining subjects form the training set with size $n = 112$. By using two-sample t test, $p = 113$ most significant genes are selected as covariates. Following the LDA framework, we assume that the normalized gene expression data are normally distributed as $\mathcal{N}(\mu_l, \bar{\Sigma})$, where the two classes are assumed to have the same covariance matrix, $\bar{\Sigma}$, but different means, μ_l, $l = 1$ for pCR state and $l = 2$ for RD state. Given a test data sample x, we calculate its LDA scores, $\delta_l(x) = x^\top \hat{\Omega}\hat{\mu}_l - \frac{1}{2}\hat{\mu}_l^\top \hat{\Omega}\hat{\mu}_l + \log \hat{\pi}_l$, $l = 1, 2$, using the precision matrix $\hat{\Omega}$ estimated by the considered methods. Here $\hat{\mu}_l = (1/n_l)\sum_{i\in \text{class}_l} x_i$ is the within-class mean in the training set and $\hat{\pi}_l = n_l/n$ is the proportion of class l subjects in the training set. The classification rule is $\hat{l}(x) = \arg\max_{l=1,2} \delta_l(x)$. Clearly, the classification performance is directly affected by the estimation quality of $\hat{\Omega}$. Hence, we assess the precision matrix estimation performance on the test data and compare GraHTP with GraSP, FBS, FoBa and GLasso. We use a 6-fold cross-validation on the training data for tuning the sparsity level parameter in ℓ_0-estimators and the regularization strength parameter in GLasso. We replicate the experiment 100 times.

Evaluation metric and results. To evaluate classification performance, we use the following defined specificity, sensitivity (or recall), and Mathews correlation coefficient (MCC) criteria as used by Cai et al. (2011):

$$\text{Specificity} = \frac{\text{TN}}{\text{TN}+\text{FP}}, \qquad \text{Sensitivity} = \frac{\text{TP}}{\text{TP}+\text{FN}},$$
$$\text{MCC} = \frac{\text{TP}\times\text{TN} - \text{FP}\times\text{FN}}{\sqrt{(\text{TP}+\text{FP})(\text{TP}+\text{FN})(\text{TN}+\text{FP})(\text{TN}+\text{FN})}},$$

where TP and TN stand for true positives (pCR) and true negatives (RD), respectively, and FP and FN stand for false positives/negatives, respectively. The larger the criterion value, the better the classification performance. Since one can adjust decision threshold in any specific algorithm to trade-off specificity and sensitivity (increase one while reduce the other), the MCC is more meaningful as a single performance metric. Table 4 lists the averages and standard deviations, in the parentheses, of the three classification criteria over 100 replications. It can be observed that GraHTP is quite competitive to the leading methods in all the three metrics. The average CPU running time of each considered method is listed in the rightmost column of Table 4.

7. Conclusion

In this article, we proposed GraHTP as a generalization of HTP from compressed sensing to the generic problem of sparsity-constrained loss minimization. The main idea is to force the gradient descent iteration to be sparse via hard thresholding. Theoretically, we proved that under mild conditions, GraHTP converges geometrically and its estimation error is controlled by the restricted norm of gradient at the target sparse solution. Under properly strengthened conditions, we further established the sparsity recovery performance of GraHTP which to our knowledge has not been systematically analyzed elsewhere in literature. Also, we have proposed and analyzed the FGraHTP algorithm as a fast variant of GraHTP without applying the debiasing operation after truncation. Empirically, we showed that GraHTP and FGraHTP are superior or competitive to the state-of-the-art greedy pursuit methods when applied to sparse learning problems including linear regression, logistic regression and precision matrix estimation. To conclude, simply combining gradient descent with hard thresholding leads to an accurate and computationally tractable procedure for solving sparsity-constrained loss minimization problems.

Acknowledgments

The authors would like to thank the anonymous referees for their constructive comments which are extremely helpful for improving this work. Xiao-Tong Yuan and Ping Li were partially supported by NSF-Bigdata-1419210, NSF-III-1360971, ONR-N00014-13-1-0764, and AFOSR-FA9550-13-1-0137. Xiao-Tong Yuan is also partially supported by NSFC-61522308 and Tencent AI Lab Rhino-Bird Joint Research Program (No.JR201801). Tong Zhang was supported by NSF-IIS-1407939 and NSF-IIS-1250985.

Appendix A. Technical Lemmas

We present in this appendix section a few technical lemmas to be used in the proofs of main results.

Lemma 13 *Let x be a k-sparse vector and $y = x - \eta\nabla f(x)$. If f is M_{2k}-smooth, then the following inequality holds:*

$$f(y_k) \le f(x) - \frac{1-\eta M_{2k}}{2\eta}\|y_k - x\|^2.$$

Proof Since f is M_{2k}-smooth, it follows that

$$\begin{aligned} f(y_k) - f(x) \le& \langle \nabla f(x), y_k - x\rangle + \frac{M_{2k}}{2}\|y_k - x\|^2 \\ \overset{\xi_1}{\le}& -\frac{1}{2\eta}\|y_k - x\|^2 + \frac{M_{2k}}{2}\|y_k - x\|^2 \\ =& -\frac{1-\eta M_{2k}}{2\eta}\|y_k - x\|^2, \end{aligned}$$

where "ξ_1" follows from the fact that y_k is the best k-support approximation to y such that

$$\|y_k - y\|^2 = \|y_k - x + \eta\nabla f(x)\|^2 \le \|x - x + \eta\nabla f(x)\|^2 = \|\eta\nabla f(x)\|^2,$$

which implies $2\eta\langle\nabla f(x), y_k - x\rangle \le -\|y_k - x\|^2$. ■

Lemma 14 *Assume that f is m_s-strongly convex. Then for any $\|x - x'\|_0 \le s$ it holds that*

$$\|x - x'\| \le \sqrt{\frac{2\max\{f(x) - f(x'), 0\}}{m_s}} + \frac{2\|\nabla_{F\cup F'} f(x')\|}{m_s},$$

where $F = supp(x)$ and $F' = supp(x')$.

Proof Since f is m_s-strongly convex, we have

$$\begin{aligned} f(x) \ge& f(x') + \langle\nabla f(x'), x - x'\rangle + \frac{m_s}{2}\|x - x'\|^2 \\ \ge& f(x') - \|\nabla_{F\cup F'} f(x')\|\|x - x'\| + \frac{m_s}{2}\|x - x'\|^2, \end{aligned}$$

where the second inequality follows from Cauchy-Schwarz inequality. From this above inequality we can see that if $f(x) \le f(x')$, then

$$\|x - x'\| \le \frac{2\|\nabla_{F\cup F'} f(x')\|}{m_s}.$$

If otherwise $f(x) > f(x')$, then we have

$$\begin{aligned} \|x - x'\| \le& \frac{\|\nabla_{F\cup F'} f(x')\| + \sqrt{\|\nabla_{F\cup F'} f(x')\|^2 + 2m_s(f(x) - f(x'))}}{m_s} \\ \le& \frac{2\|\nabla_{F\cup F'} f(x')\| + \sqrt{2m_s(f(x) - f(x'))}}{m_s}. \end{aligned}$$

By combining the above two cases we get the desired bound. ■

Lemma 15 *Assume that f is m_s-strongly convex and M_s-smooth. For any index set F with cardinality $|F| \le s$ and any x, y with $supp(x) \cup supp(y) \subseteq F$, if $\eta \in (0, 2m_s/M_s^2)$, then*

$$\|x - y - \eta\nabla_F f(x) + \eta\nabla_F f(y)\| \le \sqrt{1 - 2\eta m_s + \eta^2 M_s^2}\|x - y\|,$$

and $\sqrt{1 - 2\eta m_s + \eta^2 M_s^2} < 1$.

Proof By adding two copies of the inequality (2) with x and y interchanged and applying Theorem 2.1.5 in the textbook (Nesterov, 2004) on the supporting set F, we can show that

$$(x - y)^\top(\nabla f(x) - \nabla f(y)) \ge m_s\|x - y\|^2, \quad \|\nabla_F f(x) - \nabla_F f(y)\| \le M_s\|x - y\|.$$

Then for any $\eta > 0$ we have

$$\|x - y - \eta\nabla_F f(x) + \eta\nabla_F f(y)\|^2 \le (1 - 2\eta m_s + \eta^2 M_s^2)\|x - y\|^2.$$

It is clear that $1 - 2\eta m_s + \eta^2 M_s^2 \ge 1 - m_s^2/M_s^2 \ge 0$. The condition $\eta < 2m_s/M_s^2$ implies $\sqrt{1 - 2\eta m_s + \eta^2 M_s^2} < 1$. This proves the lemma. ■

Lemma 16 *Assume that f is M_s-smooth and m_s-strongly convex. Let F and F' be two index sets with cardinality $|F \cup F'| = s$. Let $x = \arg\min_{supp(y)\subseteq F} f(y)$ and $supp(x') \subseteq F'$. Then for any $\eta \in (0, 2m_s/M_s^2)$, the following two inequalities hold:*

$$\|(x - x')_F\| \le \frac{\rho\|x'_{F'\setminus F}\|}{1-\rho} + \frac{\eta\|\nabla_{F\cup F'} f(x')\|}{1-\rho}, \tag{14}$$

$$\|x - x'\| \le \frac{\|x'_{F'\setminus F}\|}{1-\rho} + \frac{\eta\|\nabla_{F\cup F'} f(x')\|}{1-\rho}, \tag{15}$$

where $\rho = \sqrt{1 - 2\eta m_s + \eta^2 M_s^2} < 1$.

Proof Since x is the minimum of $f(y)$ restricted over the supporting set F, we have $\langle\nabla f(x), z\rangle = 0$ whenever $\mathrm{supp}(z) \subseteq F$. Then

$$\begin{aligned} &\|(x - x')_F\|^2 \\ =& \langle x - x', (x - x')_F\rangle \\ =& \langle x - x' - \eta\nabla_{F\cup F'} f(x) + \eta\nabla_{F\cup F'} f(x'), (x - x')_F\rangle - \eta\langle\nabla_{F\cup F'} f(x'), (x - x')_F\rangle \\ \overset{\xi_1}{\le}& \sqrt{1 - 2\eta m_s + \eta^2 M_s^2}\|x - x'\|\|(x - x')_F\| + \eta\|\nabla_{F\cup F'} f(x')\|\|(x - x')_F\|, \end{aligned}$$

where "ξ_1" follows from Lemma 15. Let us abbreviate $\rho = \sqrt{1 - 2\eta m_s + \eta^2 M_s^2}$. After simplification, we have

$$\|(x - x')_F\| \le \rho\|x - x'\| + \eta\|\nabla_{F\cup F'} f(x')\|. \tag{16}$$

It follows that

$$\begin{aligned} \|x - x'\| \le& \|(x - x')_F\| + \|(x - x')_{F'\setminus F}\| \\ \le& \rho\|x - x'\| + \eta\|\nabla_{F\cup F'} f(x')\| + \|(x - x')_{F'\setminus F}\|. \end{aligned}$$

After rearrangement we obtain

$$\begin{aligned}\|x-x'\| \leq & \frac{\|(x-x')_{F'\setminus F}\|}{1-\rho}+\frac{\eta\|\nabla_{F\cup F'}f(x')\|}{1-\rho}\\ = & \frac{\|x'_{F'\setminus F}\|}{1-\rho}+\frac{\eta\|\nabla_{F\cup F'}f(x')\|}{1-\rho}.\end{aligned} \tag{17}$$

By combining (16) and (17) we get

$$\|(x-x')_F\| \leq \frac{\rho\|x'_{F'\setminus F}\|}{1-\rho}+\frac{\eta\|\nabla_{F\cup F'}f(x')\|}{1-\rho}.$$

This proves the desired bounds in this lemma. ■

The following lemma is established by Shen & Li (2016, Theorem 1) for bounding the estimation error of hard-thresholding operation. This result will be extensively used in our analysis.

Lemma 17 *Let $b\in\mathbb{R}^p$ be an arbitrary p-dimensional vector and $a\in\mathbb{R}^p$ be any k-sparse vector. Denote $\bar{k}=\|a\|_0\leq k$. Then, we have the following universal bound:*

$$\|b_k-a\|^2\leq\nu\|b-a\|^2,\quad \nu=1+\frac{\beta+\sqrt{(4+\beta)\beta}}{2},\quad \beta=\frac{\min\{\bar{k},p-k\}}{k-\bar{k}+\min\{\bar{k},p-k\}}.$$

Appendix B. Proofs of Main Theorems in Section 3

The technical proofs of main results in Section 3 are collected in this appendix section.

B.1 Proof of Theorem 2

Before proving Theorem 2, we first present two lemmas which are respectively key to the proof of part(a) and part(b) of Theorem 2.

Lemma 18 *Assume that f is M_{3k}-smooth and m_{3k}-strongly convex. Let $\bar{x}$ be an arbitrary k-sparse vector. Then at time instance t, for any $\eta\in(0,2m_{3k}/M_{3k}^2)$, GraHTP will output $x^{(t)}$ satisfying*

$$\|x^{(t)}-\bar{x}\|\leq\frac{\rho}{1-\rho}\|x^{(t-1)}-\bar{x}\|+\frac{2\eta\|\nabla_{2k}f(\bar{x})\|}{1-\rho},$$

where $\rho=\sqrt{1-2\eta m_{3k}+\eta^2M_{3k}^2}<1$.

Proof Denote $\bar{F}=\mathrm{supp}(\bar{x})$. Since $x^{(t)}$ is the minimum of $f(x)$ restricted over the supporting set $F^{(t)}$, it is directly known from the inequality (15) in Lemma 16 that

$$\|x^{(t)}-\bar{x}\|\leq\frac{\|(x^{(t)}-\bar{x})_{\bar{F}\setminus F^{(t)}}\|}{1-\rho}+\frac{\eta\|\nabla_{F^{(t)}}f(\bar{x})\|}{1-\rho}. \tag{18}$$

According to the definition of $F^{(t)}$,

$$\|(x^{(t-1)}-\eta\nabla f(x^{(t-1)}))_{\bar{F}}\|\leq\|(x^{(t-1)}-\eta\nabla f(x^{(t-1)}))_{F^{(t)}}\|.$$

By eliminating the contribution on $\bar{F}\cap F^{(t)}$ we get

$$\|(x^{(t-1)}-\eta\nabla f(x^{(t-1)}))_{\bar{F}\setminus F^{(t)}}\|\leq\|(x^{(t-1)}-\eta\nabla f(x^{(t-1)}))_{F^{(t)}\setminus\bar{F}}\|. \tag{19}$$

For the right-hand side, we can derive

$$\begin{aligned}&\|(x^{(t-1)}-\eta\nabla f(x^{(t-1)}))_{F^{(t)}\setminus\bar{F}}\|\\ \leq&\|(x^{(t-1)}-\bar{x}-\eta\nabla f(x^{(t-1)})+\eta\nabla f(\bar{x}))_{F^{(t)}\setminus\bar{F}}\|+\eta\|\nabla_{F^{(t)}\setminus\bar{F}}f(\bar{x})\|.\end{aligned} \tag{20}$$

As for the left-hand side, we can see that

$$\begin{aligned}&\|(x^{(t-1)}-\eta\nabla f(x^{(t-1)}))_{\bar{F}\setminus F^{(t)}}\|\\ \geq&\|(x^{(t-1)}-\bar{x}-\eta\nabla f(x^{(t-1)})+\eta\nabla f(\bar{x}))_{\bar{F}\setminus F^{(t)}}-(x^{(t)}-\bar{x})_{\bar{F}\setminus F^{(t)}}-\eta\nabla_{\bar{F}\setminus F^{(t)}}f(\bar{x})\|\\ \geq&\|(x^{(t)}-\bar{x})_{\bar{F}\setminus F^{(t)}}\|-\|(x^{(t-1)}-\bar{x}-\eta\nabla f(x^{(t-1)})+\eta\nabla f(\bar{x}))_{\bar{F}\setminus F^{(t)}}\|\\ &-\eta\|\nabla_{\bar{F}\setminus F^{(t)}}f(\bar{x})\|.\end{aligned} \tag{21}$$

Denote $\bar{F}\Delta F^{(t)}$ the symmetric difference of $\bar{F}$ and $F^{(t)}$ and let $F=\bar{F}\cup F^{(t)}\cup F^{(t-1)}$. It can be shown from (19), (20) and (21) that

$$\begin{aligned}&\|(x^{(t)}-\bar{x})_{\bar{F}\setminus F^{(t)}}\|\\ \leq&\|(x^{(t-1)}-\bar{x}-\eta\nabla f(x^{(t-1)})+\eta\nabla f(\bar{x}))_{\bar{F}\Delta F^{(t)}}\|+\eta\|\nabla_{\bar{F}\Delta F^{(t)}}f(\bar{x})\|\\ \leq&\|x^{(t-1)}-\bar{x}-\eta\nabla_F f(x^{(t-1)})+\eta\nabla_F f(\bar{x})\|+\eta\|\nabla_{F^{(t)}\Delta\bar{F}}f(\bar{x})\|\\ \overset{\xi_1}{\leq}&\rho\|x^{(t-1)}-\bar{x}\|+\eta\|\nabla_{F^{(t)}\Delta\bar{F}}f(\bar{x})\|,\end{aligned} \tag{22}$$

where "ξ_1" follows from Lemma 15. As a final step, combining (18) and (22) gives us

$$\begin{aligned}\|x^{(t)}-\bar{x}\|\leq&\frac{\rho}{1-\rho}\|x^{(t-1)}-\bar{x}\|+\frac{2\eta\|\nabla_{F^{(t)}\cup\bar{F}}f(\bar{x})\|}{1-\rho}\\ \leq&\frac{\rho}{1-\rho}\|x^{(t-1)}-\bar{x}\|+\frac{2\eta\|\nabla_{2k}f(\bar{x})\|}{1-\rho}.\end{aligned}$$

This completes the proof. ■

Lemma 19 *Let $\bar{x}$ be an arbitrary $\bar{k}$-sparse vector. Assume that $s=2k+\bar{k}\leq p$ and f is M_s-smooth and m_s-strongly convex. Then at time instance t, for any $\eta\in(0,2m_s/M_s^2)$, FGraHTP will output $x^{(t)}$ satisfying*

$$\|x^{(t)}-\bar{x}\|\leq\gamma\rho\|x^{(t-1)}-\bar{x}\|+\gamma\eta\|\nabla_s f(\bar{x})\|,$$

where $\rho=\sqrt{1-2\eta m_s+\eta^2M_s^2}<1$ and $\gamma=\sqrt{1+\left(\bar{k}/k+\sqrt{(4+\bar{k}/k)\bar{k}/k}\right)/2}$.

Proof Recall that $F^{(t)} = \text{supp}(x^{(t)})$ and $F = F^{(t-1)} \cup F^{(t)} \cup \text{supp}(\bar{x})$. Consider the following vector

$$y = x^{(t-1)} - \eta \nabla_F f(x^{(t-1)}).$$

By using triangular inequality,

$$\begin{aligned} \|y - \bar{x}\| &= \|x^{(t-1)} - \eta \nabla_F f(x^{(t-1)}) - \bar{x}\| \\ &\leq \|x^{(t-1)} - \bar{x} - \eta \nabla_F f(x^{(t-1)}) + \eta \nabla_F f(\bar{x})\| + \eta \|\nabla_F f(\bar{x})\| \\ &\leq \rho \|x^{(t-1)} - \bar{x}\| + \eta \|\nabla_s f(\bar{x})\|, \end{aligned}$$

where the last inequality follows from Lemma 15 and $\|\nabla_F f(\bar{x})\| \leq \|\nabla_s f(\bar{x})\|$. We note that $x^{(t)} = y_k$ in FGraHTP. Then, by invoking Lemma 17 we get

$$\|x^{(t)} - \bar{x}\| \leq \gamma \|y - \bar{x}\|,$$

where $\gamma = \sqrt{1 + \left(\bar{k}/k + \sqrt{(4 + \bar{k}/k)\bar{k}/k}\right)/2}$. It follows that

$$\|x^{(t)} - \bar{x}\| \leq \gamma\rho \|x^{(t-1)} - \bar{x}\| + \gamma\eta \|\nabla_s f(\bar{x})\|.$$

This proves the desired bound. ■

Equipped with Lemma 18 and Lemma 19, we can now prove Theorem 2 in a straightforward way.

Proof [of Theorem 2]

Part(a): Since $M_{3k}/m_{3k} < 2\sqrt{3}/3$, there exists $\eta \in (0, 2m_{3k}/M_{3k}^2)$ such that $\rho = \sqrt{1 - 2\eta m_{3k} + \eta^2 M_{3k}^2} < 0.5$ and thus $\rho/(1-\rho) < 1$. By recursively applying Lemma 18 and noting the fact $\|\nabla_s f(x)\| \leq \sqrt{s}\|\nabla f(x)\|_\infty$ we obtain the desired bound in this part.

Part(b): Note that $\gamma = 1.62$ when $\bar{k} = k$ in Lemma 19. Since $M_{3k}/m_{3k} < 1.26$, there exists $\eta \in (0, 2m_{3k}/M_{3k}^2)$ such that $\rho < 0.62$ and thus $1.62\rho < 1$. Then by recursively applying Lemma 19 with $\bar{k} = k$ we obtain the desired bound in this part. ■

B.2 Proof of Theorem 5

We need the following lemma to prove Theorem 5.

Lemma 20 *Assume that f is M_{2k}-smooth and m_{2k}-strongly convex. Assume the step-size $\eta < 1/M_{2k}$. Let $\bar{x}$ be an arbitrary $\bar{k}$-sparse vector with $k \geq \left(2 + \frac{4}{\eta^2 m_{2k}^2}\right)\bar{k}$. Then GraHTP outputs $x^{(t)}$ satisfying*

$$f(x^{(t)}) \leq f(\bar{x}) + (1 - \bar{\nu})^t \bar{\triangle}^{(0)},$$

where $\bar{\nu} = \eta m_{2k}(1 - \eta M_{2k})/2 \in (0, 0.125 m_{2k}/M_{2k})$ and $\bar{\triangle}^{(0)} = f(x^{(0)}) - f(\bar{x})$.

Proof From the definition of $\tilde{x}^{(t)}$ we know that the following inequality holds:

$$\|\tilde{x}^{(t)}_{F^{(t)}} - x^{(t-1)}\| \geq \eta \|\nabla_{F^{(t)} \setminus F^{(t-1)}} f(x^{(t-1)})\|. \tag{23}$$

From Lemma 13 we get

$$f(x^{(t)}) - f(x^{(t-1)}) \leq f(\tilde{x}^{(t)}_{F^{(t)}}) - f(x^{(t-1)}) \leq -\frac{1 - \eta M_{2k}}{2\eta} \|\tilde{x}^{(t)}_{F^{(t)}} - x^{(t-1)}\|^2. \tag{24}$$

Combining the above two inequalities (23) and (24) gives us

$$f(x^{(t)}) - f(x^{(t-1)}) \leq -\frac{(1 - \eta M_{2k})\eta}{2} \|\nabla_{F^{(t)} \setminus F^{(t-1)}} f(x^{(t-1)})\|^2. \tag{25}$$

Let $\bar{F} = \text{supp}(\bar{x})$. Under the conditions in the theorem, we claim

$$\|\nabla_{F^{(t)} \setminus F^{(t-1)}} f(x^{(t-1)})\|^2 \geq m_{2k} \left[f(x^{(t-1)}) - f(\bar{x}) \right]. \tag{26}$$

To prove this , let us distinguish the following two mutually complementary cases:

- Case I: $|F^{(t)} \setminus F^{(t-1)}| \geq \bar{k}$. In this case, we have $|F^{(t)} \setminus F^{(t-1)}| \geq |\bar{F} \setminus F^{(t-1)}|$. From the m_{2k}-strong convexity of f we have

$$\begin{aligned} &\frac{m_{2k}}{2} \|\bar{x} - x^{(t-1)}\|^2 \\ \leq & f(\bar{x}) - f(x^{(t-1)}) - (\bar{x} - x^{(t-1)})^\top \nabla f(x^{(t-1)}) \\ \overset{\xi_1}{\leq} & f(\bar{x}) - f(x^{(t-1)}) + \frac{m_{2k}}{2} \|\bar{x} - x^{(t-1)}\|^2 + \frac{1}{2m_{2k}} \|\nabla_{\bar{F} \setminus F^{(t-1)}} f(x^{(t-1)})\|^2, \end{aligned}$$

  where "ξ_1" follows from Cauchy-Schwartz inequality, a basic inequality $ma^2/2 + b^2/(2m) \geq ab$ for any $m > 0$, and $\nabla_{F^{(t-1)}} f(x^{(t-1)}) = 0$. This implies

$$\|\nabla_{\bar{F} \setminus F^{(t-1)}} f(x^{(t-1)})\|^2 \geq 2m_{2k} \left[f(x^{(t-1)}) - f(\bar{x}) \right]. \tag{27}$$

  Since $F^{(t)} \setminus F^{(t-1)}$ contains the top $|F^{(t)} \setminus F^{(t-1)}|$ (in magnitude) entries in $\nabla f(x^{(t-1)})$ and $|F^{(t)} \setminus F^{(t-1)}| \geq |\bar{F} \setminus F^{(t-1)}|$, it follows that

$$\|\nabla_{F^{(t)} \setminus F^{(t-1)}} f(x^{(t-1)})\|^2 \geq \|\nabla_{\bar{F} \setminus F^{(t-1)}} f(x^{(t-1)})\|^2 \geq 2m_{2k} \left[f(x^{(t-1)}) - f(\bar{x}) \right].$$

- Case II: $|F^{(t)} \setminus F^{(t-1)}| < \bar{k}$. In this case, from the step (**S2**) we know that each element of $\tilde{x}^{(t)}$ over $\bar{F} \setminus (F^{(t)} \cup F^{(t-1)})$ has smaller magnitude than that over $F^{(t)} \cap F^{(t-1)}$. This implies

$$\frac{\|\tilde{x}^{(t)}_{\bar{F} \setminus (F^{(t)} \cup F^{(t-1)})}\|^2}{|\bar{F} \setminus (F^{(t)} \cup F^{(t-1)})|} \leq \frac{\|\tilde{x}^{(t)}_{(F^{(t)} \cap F^{(t-1)}) \setminus \bar{F}}\|^2}{|(F^{(t)} \cap F^{(t-1)}) \setminus \bar{F}|}.$$

  Since $\tilde{x}^{(t)}_{\bar{F} \setminus (F^{(t)} \cup F^{(t-1)})} = -\eta \nabla_{\bar{F} \setminus (F^{(t)} \cup F^{(t-1)})} f(x^{(t-1)})$, $\tilde{x}^{(t)}_{(F^{(t)} \cap F^{(t-1)}) \setminus \bar{F}} = x^{(t-1)}_{(F^{(t)} \cap F^{(t-1)}) \setminus \bar{F}}$, $|\bar{F} \setminus (F^{(t)} \cup F^{(t-1)})| \leq \bar{k}$ and $|(F^{(t)} \cap F^{(t-1)}) \setminus \bar{F}| \geq k - 2\bar{k}$, we have

$$\eta^2 \|\nabla_{\bar{F} \setminus (F^{(t)} \cup F^{(t-1)})} f(x^{(t-1)})\|^2 \leq \frac{\bar{k}}{k - 2\bar{k}} \|x^{(t-1)}_{(F^{(t)} \cap F^{(t-1)}) \setminus \bar{F}}\|^2.$$

From the m_{2k}-strong convexity of f we have

$$\begin{aligned}
&\frac{m_{2k}}{2}\|\bar{x}-x^{(t-1)}\|^2\\
\leq& f(\bar{x})-f(x^{(t-1)})-(\bar{x}-x^{(t-1)})^\top\nabla f(x^{(t-1)})\\
\overset{\xi_1}{\leq}& f(\bar{x})-f(x^{(t-1)})+\frac{m_{2k}}{4}\|\bar{x}-x^{(t-1)}\|^2+\frac{1}{m_{2k}}\|\nabla_{\bar{F}\setminus F^{(t-1)}}f(x^{(t-1)})\|^2\\
\leq& f(\bar{x})-f(x^{(t-1)})+\frac{m_{2k}}{4}\|\bar{x}-x^{(t-1)}\|^2+\frac{1}{m_{2k}}\|\nabla_{\bar{F}\setminus(F^{(t)}\cup F^{(t-1)})}f(x^{(t-1)})\|^2\\
&+\frac{1}{m_{2k}}\|\nabla_{(F^{(t)}\setminus F^{(t-1)})\cap\bar{F}}f(x^{(t-1)})\|^2\\
\overset{\xi_2}{\leq}& f(\bar{x})-f(x^{(t-1)})+\frac{m_{2k}}{4}\|\bar{x}-x^{(t-1)}\|^2+\frac{\bar{k}}{\eta^2(k-2\bar{k})m_{2k}}\|x^{(t-1)}_{(F^{(t)}\cap F^{(t-1)})\setminus\bar{F}}\|^2\\
&+\frac{1}{m_{2k}}\|\nabla_{F^{(t)}\setminus F^{(t-1)}}f(x^{(t-1)})\|^2\\
\overset{\xi_3}{\leq}& f(\bar{x})-f(x^{(t-1)})+\frac{m_{2k}}{4}\|\bar{x}-x^{(t-1)}\|^2+\frac{\bar{k}}{\eta^2(k-2\bar{k})m_{2k}}\|\bar{x}-x^{(t-1)}\|^2\\
&+\frac{1}{m_{2k}}\|\nabla_{F^{(t)}\setminus F^{(t-1)}}f(x^{(t-1)})\|^2,
\end{aligned}$$

where "ξ_1" follows from Cauchy-Schwartz inequality, $ma^2/4+b^2/(m)\geq ab$ for any $m>0$, and $\nabla_{F^{(t-1)}}f(x^{(t-1)})=0$, "$\xi_2$" follows from the preceding inequality, and "ξ_3" is due to $\|x^{(t-1)}_{(F^{(t)}\cap F^{(t-1)})\setminus\bar{F}}\|\leq\|\bar{x}-x^{(t-1)}\|$. Since $k\geq\left(2+\frac{4}{\eta^2 m_{2k}^2}\right)\bar{k}$, the above inequality leads to

$$\|\nabla_{\bar{F}\setminus F^{(t-1)}}f(x^{(t-1)})\|^2\geq m_{2k}\left[f(x^{(t-1)})-f(\bar{x})\right].$$

Since $\eta<1/M_{2k}$, from (25) and (26) we get that

$$\begin{aligned}
f(x^{(t)})&\leq f(x^{(t-1)})-\frac{\eta m_{2k}(1-\eta M_{2k})}{2}\left[f(x^{(t-1)})-f(\bar{x})\right]\\
&=f(x^{(t-1)})-\bar{\nu}\left[f(x^{(t-1)})-f(\bar{x})\right].
\end{aligned}$$

Therefore, we get

$$f(x^{(t)})-f(\bar{x})\leq(1-\bar{\nu})(f(x^{(t-1)})-f(\bar{x})).$$

Since $m_{2k}\leq M_{2k}$ and $\eta\in(0,1/M_{2k})$, it can be verified that $\bar{\nu}\in(0,0.125m_{2k}/M_{2k})$. By recursively applying the above inequality we obtain the desired result. ■

We are now in the position to prove Theorem 5.

Proof [of Theorem 5] **Part(a):** Since $\eta<1/M_{2k}$, it is directly known from Lemma 13 that $\{f(x^{(t)})\}$ is monotonically decreasing. From Lemma 14 we know that the result holds when

$f(x^{(t)})\leq f(\bar{x})$. Therefore, we only need to consider the case when $f(x^{(t)})>f(\bar{x})$. In this case, from Lemma 14 and Lemma 20 we get

$$\begin{aligned}
\|x^{(t)}-\bar{x}\|\leq&\sqrt{\frac{2(f(x^{(t)})-f(\bar{x}))}{m_{2k}}}+\frac{2\|\nabla_{2k}f(\bar{x})\|}{m_{2k}}\\
\leq&\sqrt{\frac{2(1-\bar{\nu})^t\bar{\triangle}^{(0)}}{m_{2k}}}+\frac{2\sqrt{2\bar{k}}\|\nabla f(\bar{x})\|_\infty}{m_{2k}}.
\end{aligned}$$

This proves the result in part(a).

Part(b): From the condition $k\geq\rho\bar{k}/(1-\rho)^2$ we can verify that $\bar{\mu}_2=\rho\gamma<1$. Thus, the result can be directly proved by recursively applying Lemma 19. ■

Appendix C. Proofs of Main Theorems in Section 4

The technical proofs of main results in Section 4 are collected in this appendix section.

C.1 Proof of Theorem 8

Before commencing with the actual proof, we first present an overview of the proof procedure which consists of the following three key ingredients:

(a) We first prove that under the given conditions, GraHTP will not terminate (i.e., $F^{(t)}\neq F^{(t-1)}$) whenever $\mathrm{supp}(\bar{x})\nsubseteq\mathrm{supp}(x^{(t-1)})$.

(b) We then show that $\mathrm{supp}(\bar{x})=\mathrm{supp}(x^{(t)},\bar{k})$ when GraHTP terminates at $x^{(t)}$.

(c) Finally we show that the conditions in the theorem guarantee finite termination of GraHTP and analyze its iteration complexity before termination.

Proof [of Theorem 8]

We first show that $F^{(t)}\neq F^{(t-1)}$ whenever $\mathrm{supp}(\bar{x})\nsubseteq\mathrm{supp}(x^{(t-1)})$. To this end, let us assume $\mathrm{supp}(\bar{x})\nsubseteq\mathrm{supp}(x^{(t-1)})$. Recall $x^\star=\arg\min_{\|x\|_0\leq k}f(x)$. Then

$$\begin{aligned}
\bar{x}_{\min}+\|x^{(t-1)}_{F^{(t-1)}\setminus\bar{F}}\|\leq&\|\bar{x}-x^{(t-1)}\|\\
\overset{\xi_1}{\leq}&\sqrt{\frac{2\max\left\{f(\bar{x})-f(x^{(t-1)}),0\right\}}{m_{2k}}}+\frac{2\|\nabla_{\bar{F}\setminus F^{(t-1)}}f(x^{(t-1)})\|}{m_{2k}}\\
\overset{\xi_2}{\leq}&\sqrt{\frac{2(f(\bar{x})-f(x^\star))}{m_{2k}}}+\frac{2\|\nabla_{\bar{F}\setminus F^{(t-1)}}f(x^{(t-1)})\|}{m_{2k}},
\end{aligned}$$

where "ξ_1" follows from Lemma 14 and "ξ_2" is due to the fact of $f(x^{(t-1)})\geq f(x^\star)$. Since it is assumed $\bar{x}_{\min}>1.62\sqrt{\frac{2(f(\bar{x})-f(x^\star))}{m_{2k}}}$, the above inequality implies

$$\|x^{(t-1)}_{F^{(t-1)}\setminus\bar{F}}\|<\frac{2\|\nabla_{\bar{F}\setminus F^{(t-1)}}f(x^{(t-1)})\|}{m_{2k}},$$

which then gives us

$$\sqrt{k-\bar{k}}x_{\min}^{(t-1)} < \frac{2\sqrt{\bar{k}}}{m_{2k}}\|\nabla f(x^{(t-1)})\|_\infty.$$

Since $\eta = \frac{1}{2M_{2k}}$ and $k \geq \left(1+\frac{16M_{2k}^2}{m_{2k}^2}\right)\bar{k}$, we then have

$$\eta\|\nabla f(x^{(t-1)})\|_\infty > x_{\min}^{(t-1)}.$$

This means that at least the smallest nonzero entry of $x^{(t-1)}$ and the largest entry of $\nabla f(x^{(t-1)})$ can be swapped in step (**S2**) of Algorithm 1, and thus $F^{(t)} \neq F^{(t-1)}$. Therefore, when the algorithm terminates at time instance t, i.e., $F^{(t)} = F^{(t-1)}$, we must have $\mathrm{supp}(\bar{x}) \subseteq \mathrm{supp}(x^{(t)})$.

Next we show that $\mathrm{supp}(\bar{x}) = \mathrm{supp}(x^{(t)}, \bar{k})$ when GraHTP terminates at time instance t with $\mathrm{supp}(\bar{x}) \subseteq \mathrm{supp}(x^{(t)})$. Assume otherwise $\mathrm{supp}(\bar{x}) \neq \mathrm{supp}(x^{(t)}, \bar{k})$. Then

$$\begin{aligned}
\bar{x}_{\min} &\leq \|\bar{x} - x_{\bar{k}}^{(t)}\| \\
&\overset{\xi_1}{\leq} 1.62\|\bar{x} - x^{(t)}\| \\
&\overset{\xi_2}{\leq} 1.62\left(\sqrt{\frac{2(f(\bar{x}) - f(x^\star))}{m_{2k}}} + \frac{2\|\nabla_{\bar{F}\setminus F^{(t)}} f(x^{(t)})\|}{m_{2k}}\right) \\
&\overset{\xi_3}{\leq} 1.62\sqrt{\frac{2(f(\bar{x}) - f(x^\star))}{m_{2k}}},
\end{aligned}$$

where "ξ_1" is based on the truncation error bound given by Shen & Li (2016, Theorem 1), "ξ_2" follows from Lemma 14 and the fact of $f(x^{(t)}) \geq f(x^\star)$, and "$\xi_3$" is the consequence of $\bar{F} \subseteq F^{(t)}$. This above inequality contradicts the assumption on $\bar{x}_{\min}$. Therefore, it must hold that $\mathrm{supp}(\bar{x}) = \mathrm{supp}(x^{(t)}, \bar{k})$.

Now we claim that GraHTP is finite under the assumed conditions. Indeed, based on Lemma 13 it is easy to verify that when $\eta = \frac{1}{2M_{2k}}$, the sequence $\{f(x^{(t)})\}$ generated by Algorithm 1 is monotonically decreasing. Since the number of k-support index sets is finite, the sequence $\{f(x^{(t)})\}$ will be eventually periodic, and thus must be eventually a constant. Therefore we deduce that $\tilde{x}_k^{(t)} = x^{(t-1)}$, i.e., $F^{(t)} = F^{(t-1)}$, when t is sufficiently enough.

Finally, we estimate the iteration complexity bound before algorithm termination. Suppose that $F^{(t)} \neq F^{(t-1)}$ (otherwise GraHTP terminates at time instance t). From the step (**S3**) we know that $\nabla_{F^{(t-1)}} f(x^{(t-1)}) = 0$. By definition of $F^{(t)}$ we may decompose $F^{(t)} = G_1 \cup (F^{(t-1)} \setminus G_2)$ with $G_1 \subseteq \mathrm{supp}(\nabla f(x^{(t-1)}))$, $G_2 \subseteq F^{(t-1)}$ and $|G_1| = |G_2| = k' \leq k$. Here, G_1 contains the top k' (in magnitude) entries in $\nabla f(x^{(t-1)})$ while G_2 contains the bottom k' nonzero entries in $x^{(t-1)}$. Since $F^{(t)} \neq F^{(t-1)}$, we have $k' \geq 1$. From the step (**S2**) we know that

$$\|x_{G_2}^{(t-1)}\| < \eta\|\nabla_{G_1} f(x^{(t-1)})\|. \tag{28}$$

Let $F = F^{(t-1)} \cup \mathrm{supp}(x^\star)$. From the conditions in the theorem we have

$$\begin{aligned}
\frac{m_{2k}}{2}\|x^\star - x^{(t-1)}\|^2 &\leq f(x^\star) - f(x^{(t-1)}) - (x^\star - x^{(t-1)})^\top \nabla f(x^{(t-1)}) \\
&\leq f(x^\star) - f(x^{(t-1)}) + \frac{m_{2k}}{2}\|x^\star - x^{(t-1)}\|^2 + \frac{1}{2m_{2k}}\|\nabla_F f(x^{(t-1)})\|^2,
\end{aligned}$$

where the second inequality follows from Cauchy-Schwartz inequality and a basic inequality $ma^2/2 + b^2/(2m) \geq ab$ for any $m > 0$. This implies

$$\|\nabla_F f(x^{(t-1)})\|^2 \geq 2m_{2k}\left[f(x^{(t-1)}) - f(x^\star)\right].$$

Let $F^\star = \mathrm{supp}(x^\star)$ and $k'' = |F^\star \setminus F^{(t-1)}|$. Obviously, we have $k'' \leq k$. Based on the above arguments, it can be verified that

$$\begin{aligned}
k\|\nabla_{G_1} f(x^{(t-1)})\|^2 &\geq (k''/k')\|\nabla_{G_1} f(x^{(t-1)})\|^2 \\
&\geq \|\nabla_F f(x^{(t-1)})\|^2 \\
&\geq 2m_{2k}\left[f(x^{(t-1)}) - f(x^\star)\right].
\end{aligned} \tag{29}$$

Now let $y^{(t)} := x^{(t-1)} + \delta^{(t-1)}$ in which

$$\delta^{(t-1)} = -\eta\nabla_{G_1} f(x^{(t-1)}) - x_{G_2}^{(t-1)}.$$

From the steps (**S1**) and (**S3**) in Algorithm 1 we get

$$\begin{aligned}
f(x^{(t)}) &\leq f(y^{(t)}) \\
&\leq f(x^{(t-1)}) + \langle \nabla f(x^{(t-1)}), \Delta^{(t-1)}\rangle + \frac{M_{2k}}{2}\|\Delta^{(t-1)}\|^2 \\
&\leq f(x^{(t-1)}) + \frac{M_{2k}}{2}\|x_{G_2}^{(t-1)}\|^2 - \frac{2\eta - \eta^2 M_{2k}}{2}\|\nabla_{G_1} f(x^{(t-1)})\|^2 \\
&\overset{\xi_1}{\leq} f(x^{(t-1)}) - (\eta - \eta^2 M_{2k})\|\nabla_{G_1} f(x^{(t-1)})\|^2 \\
&\overset{\xi_2}{\leq} f(x^{(t-1)}) - \frac{m_{2k}}{2kM_{2k}}(f(x^{(t-1)}) - f(x^\star)),
\end{aligned}$$

where "ξ_1" follows from (28) and "ξ_2" uses (29) and $\eta = \frac{1}{M_{2k}}$ as well. Therefore, we get

$$f(x^{(t)}) - f(x^\star) \leq \left(1 - \frac{m_{2k}}{2kM_{2k}}\right)(f(x^{(t-1)}) - f(x^\star)).$$

Note that $f(x^{(t)}) \geq f(x^\star)$ for all $t \geq 0$. By recursively using the above inequality we get

$$f(x^{(t)}) - f(x^\star) \leq \left(1 - \frac{m_{2k}}{2kM_{2k}}\right)^t (f(x^{(0)}) - f(x^\star)).$$

Let us define the following quantity

$$\triangle^{-\star} = \min_{\|x\|_0 \leq k, \mathrm{supp}(x) \neq \mathrm{supp}(x^\star), f(x) > f(x^\star)} f(x) - f(x^\star).$$

Then $f(x^{(t)}) - f(x^\star) \leq \triangle^{-\star}$ when $t \geq \frac{2kM_{2k}}{m_{2k}} \ln \frac{\triangle^{(0)}}{\triangle^{-\star}}$ (note that $\triangle^{-\star} > 0$ by definition). After that, we have $f(x^{(t)}) = f(x^\star)$, i.e., $x^{(t)}$ is also a k-sparse minimizer. Then according to Lemma 21 we have $x_{\min}^{(t)} \geq \frac{\|\nabla f(x^{(t)})\|_\infty}{M_{2k}} > \eta\|\nabla f(x^{(t)})\|_\infty$, and thus the algorithm terminates at $x^{(t)}$. Based on the above arguments, we can conclude that GraHTP terminates after at most $t = \left\lceil \frac{2kM_{2k}}{m_{2k}} \ln \frac{\triangle^{(0)}}{\triangle^{-\star}} \right\rceil$ steps of iteration. This completes the proof. ■

C.2 Proof of Theorem 10

The following lemma gives a necessary condition on the k-sparse minimizer $x^\star$. A similar result was proved by Beck & Eldar (2013).

Lemma 21 *If f is M_{2k}-smooth, then the following inequality holds for the global k-sparse minimizer $x^\star = \arg\min_{\|x\|_0\leq k} f(x)$:*

$$x^\star_{\min} \geq \frac{\|\nabla f(x^\star)\|_\infty}{M_{2k}}.$$

Proof Assume otherwise that $\vartheta^\star := \frac{M_{2k}x^\star_{\min}}{\|\nabla f(x^\star)\|_\infty} < 1$. Let us consider $\tilde{x}^\star = x^\star - \eta\nabla f(x^\star)$ with any $\eta \in (\vartheta^\star/M_{2k}, 1/M_{2k})$. From Lemma 13 we get that

$$f(\tilde{x}^\star_k)) \leq f(x^\star) - \frac{1-\eta M_{2k}}{2\eta}\|\tilde{x}^\star_k - x^\star\|^2.$$

Since $\eta < \frac{1}{M_{2k}}$ and $x^\star_{\min} = \frac{\vartheta^\star\|\nabla f(x^\star)\|_\infty}{M_{2k}} < \eta\|\nabla f(x^\star)\|_\infty$, we have $\tilde{x}^\star_k \neq x^\star$ and thus it follows from the above inequality that $f(\tilde{x}^\star_k) < f(x^\star)$ which contradicts the optimality of $x^\star$. ■

Now we can prove the main result in Theorem 10.

Proof [of Theorem 10] We first show that $\mathrm{supp}(\bar{x}) = \mathrm{supp}(x^\star, \bar{k})$ if the condition (1) is satisfied. Assume otherwise $\mathrm{supp}(\bar{x}) \neq \mathrm{supp}(x^\star, \bar{k})$. From the optimality of $x^\star$ and $k \geq \bar{k}$ we have $f(x^\star) \leq f(\bar{x})$. By invoking Lemma 14 and the truncation error bound by Shen & Li (2016, Theorem 1) we get

$$\bar{x}_{\min} \leq \|x^\star_{\bar{k}} - \bar{x}\| \leq 1.62\|x^\star - \bar{x}\| \leq \frac{3.24\sqrt{2k}\|\nabla f(\bar{x})\|_\infty}{m_{2k}} < \frac{4.59\sqrt{k}\|\nabla f(\bar{x})\|_\infty}{m_{2k}},$$

which contradicts the condition.

Next we prove that $\mathrm{supp}(\bar{x}) = \mathrm{supp}(x^\star, \bar{k})$ if the condition (2) is satisfied. Let $\bar{F} = \mathrm{supp}(\bar{x})$ and $F^\star = \mathrm{supp}(x^\star)$. We first claim that $\bar{F} \subseteq F^\star$. Indeed, if otherwise $\bar{F} \nsubseteq F^\star$, then

$$\begin{aligned}\bar{x}_{\min} + \|x^\star_{F^\star\setminus\bar{F}}\| &\leq \|\bar{x} - x^\star\| \\ &\overset{\xi_1}{\leq} \sqrt{\frac{2\max\{f(\bar{x}) - f(x^\star), 0\}}{m_{2k}}} + \frac{2\|\nabla_{\bar{F}\setminus F^\star} f(x^\star)\|}{m_{2k}} \\ &= \sqrt{\frac{2(f(\bar{x}) - f(x^\star))}{m_{2k}}} + \frac{2\|\nabla_{\bar{F}\setminus F^\star} f(x^\star)\|}{m_{2k}},\end{aligned}$$

where "ξ_1" follows from Lemma 14. Since $\bar{x}_{\min} > 1.62\sqrt{\frac{2(f(\bar{x})-f(x^\star))}{m_{2k}}}$, the above inequality leads to

$$\sqrt{k-\bar{k}}x^\star_{\min} \leq \|x^\star_{F^\star\setminus\bar{F}}\| < \frac{2\|\nabla_{\bar{F}\setminus F^\star} f(x^\star)\|}{m_{2k}} \leq \frac{2\sqrt{k}\|\nabla f(x^\star)\|_\infty}{m_{2k}}.$$

Since $k \geq \left(1 + \frac{4M^2_{2k}}{m^2_{2k}}\right)\bar{k}$, we thus have $x^\star_{\min} < \frac{\|\nabla f(x^\star)\|_\infty}{M_{2k}}$. This contradicts Lemma 21. Therefore we must have $\bar{F} \subseteq F^\star$. Now let us assume $\mathrm{supp}(\bar{x}) \neq \mathrm{supp}(x^\star, \bar{k})$. Then

$$\begin{aligned}\bar{x}_{\min} &\leq \|\bar{x} - x^\star_{\bar{k}}\| \\ &\overset{\xi_1}{\leq} 1.62\|\bar{x} - x^\star\| \\ &\overset{\xi_2}{\leq} 1.62\left(\sqrt{\frac{2(f(\bar{x}) - f(x^\star))}{m_{2k}}} + \frac{2\|\nabla_{\bar{F}\setminus F^\star} f(x^\star)\|}{m_{2k}}\right) \\ &\overset{\xi_3}{=} 1.62\sqrt{\frac{2(f(\bar{x}) - f(x^\star))}{m_{2k}}} < 2.3\sqrt{\frac{f(\bar{x}) - f(x^\star)}{m_{2k}}},\end{aligned}$$

where "ξ_1" is based on the truncation error bound by Shen & Li (2016, Theorem 1), "ξ_2" follows from Lemma 14 and the fact of $f(\bar{x}) \geq f(x^\star)$, and "$\xi_3$" is the consequence of $\bar{F} \subseteq F^\star$. This above inequality contradicts the assumption on $\bar{x}_{\min}$. Therefore, it must hold that $\mathrm{supp}(\bar{x}) = \mathrm{supp}(x^\star, \bar{k})$. ■

Appendix D. Some Technical Details in Section 5

In this appendix section, we give the proof of Proposition 12 and present some implementation details of the proposed ADMM method for solving the subproblem (13).

D.1 Proof of Proposition 12

Proof It is straightforward to show that

$$\|\nabla f(\bar{w})\|_\infty \leq \|\nabla l(\bar{w})\|_\infty + \lambda\|\bar{w}\|_\infty. \tag{30}$$

We next bound the term $\|\nabla l(\bar{w})\|_\infty$. From (10) we have

$$\begin{aligned}\left|\frac{\partial l}{\partial[\bar{w}]_j}\right| &= \left|\frac{1}{n}\sum_{i=1}^n -v^{(i)}[u^{(i)}]_j + \mathbb{E}_v[v[u^{(i)}]_j \mid u^{(i)}]\right| \\ &\leq \left|\frac{1}{n}\sum_{i=1}^n v^{(i)}[u^{(i)}]_j - \mathbb{E}[v[u]_j]\right| + \left|\frac{1}{n}\sum_{i=1}^n \mathbb{E}_v[v[u^{(i)}]_j \mid u^{(i)}] - \mathbb{E}[v[u]_j]\right|,\end{aligned}$$

where $\mathbb{E}[\cdot]$ is taken over the distribution (9). Therefore, for any $\varepsilon > 0$,

$$\begin{aligned}\mathbb{P}\left(\left|\frac{\partial l}{\partial[\bar{w}]_j}\right| > \varepsilon\right) \leq& \mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^n v^{(i)}[u^{(i)}]_j - \mathbb{E}[v[u]_j]\right| > \frac{\varepsilon}{2}\right) \\ &+ \mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^n \mathbb{E}_v[v[u^{(i)}]_j \mid u^{(i)}] - \mathbb{E}[v[u]_j]\right| > \frac{\varepsilon}{2}\right) \\ \overset{\xi_1}{\leq}& 4\exp\left\{-\frac{n\varepsilon^2}{8\sigma^2}\right\},\end{aligned}$$

where "ξ_1" follows from the large deviation inequality of sub-Gaussian random variables which is standard (see, e.g., Vershynin, 2011). By the union bound we have

$$\mathbb{P}(\|\nabla l(\bar{w})\|_\infty > \varepsilon) \quad \leq \quad 4p\exp\left\{-\frac{n\varepsilon^2}{8\sigma^2}\right\}.$$

By choosing $\varepsilon = 4\sigma\sqrt{\ln p/n}$ in the above inequality we obtain that with probability at least $1 - 4p^{-1}$,

$$\|\nabla l(\bar{w})\|_\infty \leq 4\sigma\sqrt{\ln p/n}.$$

Combining the above bound with (30) yields the desired result. ■

D.2 The ADMM Method for Solving the Subproblem (13)

Now we present the algorithmic procedure of ADMM for solving the subproblem (13). By introducing an auxiliary variable $\Theta \in \mathbb{R}^{p\times p}$, this subproblem can be equivalently formulated as

$$\min_{\alpha I \preceq \Omega \preceq \beta I} L(\Omega), \quad \text{s.t. } \Omega = \Theta, \ \text{supp}(\Theta) \subseteq F. \tag{31}$$

Then, the augmented Lagrangian function of (31) is

$$J(\Omega, \Theta, \Gamma) := L(\Omega) - \langle \Gamma, \Omega - \Theta\rangle + \frac{\rho}{2}\|\Omega - \Theta\|_{Frob}^2,$$

where $\Gamma \in \mathbb{R}^{p\times p}$ is the multiplier of the linear constraint $\Omega = \Theta$ and $\rho > 0$ is the penalty strength parameter for the violation of the linear constraint. The ADMM method alternately solves the following problems to generate the new iterate:

$$\Omega^{(\tau)} = \underset{\alpha I \preceq \Omega \preceq \beta I}{\arg\min}\ J(\Omega, \Theta^{(\tau-1)}, \Gamma^{(\tau-1)}), \tag{32}$$

$$\Theta^{(\tau)} = \underset{\text{supp}(\Theta)\subseteq F}{\arg\min}\ J(\Omega^{(\tau)}, \Theta, \Gamma^{(\tau-1)}), \tag{33}$$

$$\Gamma^{(\tau)} = \Gamma^{(\tau-1)} - \rho(\Omega^{(\tau)} - \Theta^{(\tau)}).$$

Let us first consider the minimization problem (32) for updating $\Omega^{(\tau)}$. It is equivalent to the following minimization problem:

$$\Omega^{(\tau)} = \underset{\alpha I \preceq \Omega \preceq \beta I}{\arg\min}\ \frac{1}{2}\|\Omega - M\|_{Frob}^2 - \frac{1}{\rho}\log\det\Omega,$$

where

$$M = \Theta^{(\tau-1)} - \frac{1}{\rho}(\Sigma_n - \Gamma^{(\tau-1)}).$$

Let the eigenvalue decomposition of M be

$$M = V\Lambda V^\top, \quad \text{with } \Lambda = \text{diag}(\lambda_1, ..., \lambda_n).$$

It is easy to verify that the solution of problem (32) is given by

$$\Omega^{(\tau)} = V\tilde{\Lambda}V^\top, \quad \text{with } \tilde{\Lambda} = \text{diag}(\tilde{\lambda}_1, ..., \tilde{\lambda}_n),$$

where

$$\tilde{\lambda}_j = \min\left\{\beta, \max\left\{\alpha, \frac{\lambda_j + \sqrt{\lambda_j^2 + 4/\rho}}{2}\right\}\right\}.$$

Next, we consider the minimization problem (33) for updating $\Theta^{(\tau)}$. It is straightforward to see that the solution of problem (33) is given by

$$\Theta^{(\tau)} = \left[\Omega^{(\tau)} - \frac{1}{\rho}\Gamma^{(\tau-1)}\right]_F.$$

References

Agarwal, A., Negahban, S., and Wainwright, M. J. Fast global convergence of gradient methods for high-dimensional statistical recovery. *The Annals of Statistics*, 40(5):2452–2482, 2012.

Bahmani, S., Raj, B., and Boufounos, P. Greedy sparsity-constrained optimization. *Journal of Machine Learning Research*, 14:807–841, 2013.

Beck, A. and Eldar, Y. C. Sparsity constrained nonlinear optimization: optimality conditions and algorithms. *SIAM Journal on Optimization*, 23(3):1480–1509, 2013.

Beck, A. and Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2006.

Blumensath, T. Compressed sensing with nonlinear observations and related nonlinear optimization problems. *IEEE Transactions on Information Theory*, 59(6):3466–3474, 2013.

Blumensath, T. and Davies, M. E. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265–274, 2009.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010.

Cai, T., Liu, W., and Luo, X. A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607, 2011.

Candès, E. J., Romberg, J. K., and Tao, T. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8): 1207–1223, 2006.

Donoho, D. L. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4): 1289–1306, 2006.

Edwards, D. M. *Introduction to Graphical Modelling*. Springer Science & Business Media New York, 2000.

Foucart, S. Hard thresholding pursuit: an algorithm for compressive sensing. *SIAM Journal on Numerical Analysis*, 49(6):2543–2563, 2011.

Foucart, S. Sparse recovery algorithms: sufficient conditions in terms of restricted isometry constants. In *Approximation Theory XIII: San Antonio 2010*, pages 65–77, 2012.

Frank, M. and Wolfe, P. An algorithm for quadratic programming. *Naval Research Logistics (NRL)*, 3(1-2):95–110, 1956.

Friedman, J., Hastie, T., and Tibshirani, R. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Garg, R. and Khandekar, R. Gradient descent with sparsification: an iterative algorithm for sparse recovery with restricted isometry property. In *International Conference on Machine Learning (ICML)*, pages 337–344, 2009.

Guyon, I., Gunn, S., Hur A. B., and Dror, G. Result analysis of the NIPS 2003 feature selection challenge. In *Advances in Neural Information Processing Systems (NIPS)*, pages 545–552, 2005.

Hess, K. R., Anderson, K., Symmans, W. F., and *et al.* Pharmacogenomic predictor of snesitivity to preoperative chemotherapy with paclitaxel and fluorouracil, doxorubicin, and cyclophosphamide in breast cancer. *Journal of Clinical Oncology*, 24(26):4236–4244, 2006.

Jaggi, M. Sparse convex optimization methods for machine learning. Technical report, PhD thesis in Theoretical Computer Science, ETH Zurich, 2011.

Jain, P., Rao, N., and Dhillon, I. Structured sparse regression via greedy hard-thresholding. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1516–1524, 2016.

Jain, P., Tewari, A., and Kar, P. On iterative hard thresholding methods for high-dimensional m-estimation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 685–693, 2014.

Jalali, A., Johnson, C. C., and Ravikumar, P. K. On learning discrete graphical models using greedy methods. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1935–1943, 2011.

Keerthi, S. S. and DeCoste, D. A modified finite newton method for fast solution of large scale linear svms. *Journal of Machine Learning Research*, 6:341–361, 2005.

Kim, Y. and Kim, J. Gradient lasso for feature selection. In *International Conference on Machine Learning (ICML)*, pages 60–67, 2004.

Langford, J., Li, L., and Zhang, T. Sparse online learning via truncated gradient. *Journal of Machine Learning Research*, 10:777–801, 2009.

Lewis, D., Yang, Y., Rose, T., and Li, F. Rcv1: a new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

Li, X., Zhao, T., Arora, R., Liu, H., and Haupt, J. Stochastic variance reduced optimization for nonconvex sparse learning. In *International Conference on Machine Learning (ICML)*, pages 917–925, 2016.

Li, Y.-H., Scarlett, J., Ravikumar, P., and Cevher, V. Sparsistency of ℓ_1-regularized m-estimators. In *International Conference on Artifficial Intelligence and Statistics (AISTATS)*, pages 644–652, 2015.

Liu, B., Yuan, X.-T., Wang, L., Liu, Q., and Metaxas, D. N. Dual Iterative Hard Thresholding: From Non-convex Sparse Minimization to Non-smooth Concave Maximization. In *International Conference on Machine Learning (ICML)*, pages 2179–2187, 2017.

Lu, Z. Smooth optimization approach for sparse covariance selection. *SIAM Journal on Optimization*, 19(4):1807–1827, 2009.

Mallat, S. G. and Zhang, Z. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

Natarajan, B. K. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2):227–234, 1995.

Needell, D. and Tropp, J. A. Cosamp: iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.

Nesterov, Y. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2004.

Nguyen, N., Needell, D., and Woolf, T. Linear convergence of stochastic iterative greedy algorithms with sparse constraints. *IEEE Transactions on Information Theory*, 63(11): 6869–6895, 2017.

Pati, Y. C., Rezaiifar, R., and Krishnaprasad, P. S. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Annual Asilomar Conference on Signals, Systems, and Computers*, pages 40–44, 1993.

Ravikumar, P., Wainwright, M. J., Raskutti, G., and Yu, B. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

Shalev-Shwartz, S., Srebro, N., and Zhang, T. Trading accuracy for sparsity in optimization problems with sparsity constraints. *SIAM Journal on Optimization*, 20(6):2807–2832, 2010.

Shen, J. and Li, P. A Tight Bound of Hard Thresholding. *arXiv preprint arXiv:1605.01656*, 2016. URL `http://arxiv.org/pdf/1605.01656.pdf`.

Tewari, A., Ravikumar, P., and Dhillon, I. S. Greedy algorithms for structurally constrained high-dimensional problems. In *Advances in Neural Information Processing Systems (NIPS)*, pages 882–890, 2011.

Tropp, J. and Gilbert, A. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007.

Van De Geer, S. and Bühlmann, P. On the conditions used to prove oracle results for the Lasso. *Electronic Journal of Statistics*, 3:1360–1392, 2009.

Vershynin, R. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2011. URL `http://arxiv.org/pdf/1011.3027.pdf`.

Wainwright, M. J. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (Lasso). *IEEE Transactions on Information Theory*, 55(5):2183–2202, 2009.

Wainwright, M. J. and Jordan, M.I. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.

Yuan, X. M. Alternating direction method for covariance selection models. *Journal of Scientific Computing*, 51(2):261–273, 2012.

Yuan, X.-T., Li, P., and Zhang, T. Gradient hard thresholding pursuit for sparsity-constrained optimization. In *International Conference on Machine Learning (ICML)*, pages 127–135, 2014.

Yuan, X.-T., Li, P., and Zhang, T. Exact recovery of hard thresholding pursuit. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3558–3566, 2016.

Yuan, X.-T. and Yan, S. Forward basis selection for pursuing sparse representations over a dictionary. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12): 3025–3036, 2013.

Yuan, X.-T. and Zhang, T. Truncated power method for sparse eigenvalue problems. *Journal of Machine Learning Research*, 14:899–925, 2013.

Zhang, T. Adaptive forward-backward greedy algorithm for sparse learning with linear models. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1921–1928, 2008.

Journal of Machine Learning Research 18 (2018) 1-51 Submitted 12/15; Revised 4/17; Published 4/18

Risk-Constrained Reinforcement Learning with Percentile Risk Criteria

Yinlam Chow YINLAMCHOW@GOOGLE.COM
DeepMind
Mountain View, CA 94043, USA

Mohammad Ghavamzadeh GHAVAMZA@GOOGLE.COM
DeepMind
Mountain View, CA 94043, USA

Lucas Janson LJANSON@STANFORD.EDU
Department of Statistics
Stanford University
Stanford, CA 94305, USA

Marco Pavone PAVONE@STANFORD.EDU
Aeronautics and Astronautics
Stanford University
Stanford, CA 94305, USA

Editor: Jan Peters

Abstract

In many sequential decision-making problems one is interested in minimizing an expected cumulative cost while taking into account *risk*, i.e., increased awareness of events of small probability and high consequences. Accordingly, the objective of this paper is to present efficient reinforcement learning algorithms for risk-constrained Markov decision processes (MDPs), where risk is represented via a chance constraint or a constraint on the conditional value-at-risk (CVaR) of the cumulative cost. We collectively refer to such problems as percentile risk-constrained MDPs. Specifically, we first derive a formula for computing the gradient of the Lagrangian function for percentile risk-constrained MDPs. Then, we devise policy gradient and actor-critic algorithms that (1) estimate such gradient, (2) update the policy in the descent direction, and (3) update the Lagrange multiplier in the ascent direction. For these algorithms we prove convergence to locally optimal policies. Finally, we demonstrate the effectiveness of our algorithms in an optimal stopping problem and an online marketing application.

Keywords: Markov Decision Process, Reinforcement Learning, Conditional Value-at-Risk, Chance-Constrained Optimization, Policy Gradient Algorithms, Actor-Critic Algorithms

1. Introduction

The most widely-adopted optimization criterion for Markov decision processes (MDPs) is represented by the *risk-neutral* expectation of a cumulative cost. However, in many applications one is interested in taking into account risk, i.e., increased awareness of events of small probability and high consequences. Accordingly, in *risk-sensitive* MDPs the objective is to minimize a risk-sensitive criterion such as the expected exponential utility, a variance-related measure, or percentile performance. There are several risk metrics available in the literature, and constructing a "good" risk criterion in a manner that is both conceptually meaningful and computationally tractable remains a topic of current research.

Risk-Sensitive MDPs: One of the earliest risk metrics used for risk-sensitive MDPs is the exponential risk metric $(1/\gamma)\mathbb{E}\big[\exp(\gamma Z)\big]$, where Z represents the cumulative cost for a sequence of decisions (Howard and Matheson, 1972). In this setting, the degree of risk-aversion is controlled by the parameter γ, whose selection, however, is often challenging. This motivated the study of several different approaches. In Collins (1997), the authors considered the maximization of a strictly concave functional of the distribution of the terminal state. In Wu and Lin (1999); Boda et al. (2004); Filar et al. (1995), risk-sensitive MDPs are cast as the problem of maximizing percentile performance. Variance-related risk metrics are considered, e.g., in Sobel (1982); Filar et al. (1989). Other mean, variance, and probabilistic criteria for risk-sensitive MDPs are discussed in the survey (White, 1988).

Numerous alternative risk metrics have recently been proposed in the literature, usually with the goal of providing an "intuitive" notion of risk and/or to ensure computational tractability. *Value-at-risk* (VaR) and *conditional value-at-risk* (CVaR) represent two promising such alternatives. They both aim at quantifying costs that might be encountered in the tail of a cost distribution, but in different ways. Specifically, for continuous cost distributions, VaR_α measures risk as the maximum cost that might be incurred with respect to a given confidence level α. This risk metric is particularly useful when there is a well-defined failure state, e.g., a state that leads a robot to collide with an obstacle. A VaR_α constraint is often referred to as a chance (probability) constraint, especially in the engineering literature, and we will use this terminology in the remainder of the paper. In contrast, CVaR_α measures risk as the expected cost given that such cost is greater than or equal to VaR_α, and provides a number of theoretical and computational advantages. CVaR optimization was first developed by Rockafellar and Uryasev (Rockafellar and Uryasev, 2002, 2000) and its numerical effectiveness has been demonstrated in several portfolio optimization and option hedging problems. Risk-sensitive MDPs with a conditional value at risk metric were considered in Boda and Filar (2006); Ott (2010); Bäuerle and Ott (2011), and a mean-average-value-at-risk problem has been solved in Bäuerle and Mundt (2009) for minimizing risk in financial markets.

The aforementioned works focus on the derivation of exact solutions, and the ensuing algorithms are only applicable to relatively small problems. This has recently motivated the application of reinforcement learning (RL) methods to risk-sensitive MDPs. We will refer to such problems as risk-sensitive RL.

Risk-Sensitive RL: To address large-scale problems, it is natural to apply reinforcement learning (RL) techniques to risk-sensitive MDPs. Reinforcement learning (Bertsekas and Tsitsiklis, 1996; Sutton and Barto, 1998) can be viewed as a class of sampling-based methods for solving MDPs. Popular reinforcement learning techniques include policy gradient (Williams, 1992; Marbach, 1998; Baxter and Bartlett, 2001) and actor-critic methods (Sutton et al., 2000; Konda and Tsitsiklis, 2000; Peters et al., 2005; Borkar, 2005; Bhatnagar et al., 2009; Bhatnagar and Lakshmanan, 2012), whereby policies are parameterized in terms of a parameter vector and policy search is performed via gradient flow approaches. One effective way to estimate gradients in RL problems is by simultaneous perturbation stochastic approximation (SPSA) (Spall, 1992). Risk-sensitive RL with expected exponential utility has been considered in Borkar (2001, 2002). More recently, the works in Tamar et al. (2012); Prashanth and Ghavamzadeh (2013) present RL algorithms for several variance-related risk measures, the works in Morimura et al. (2010); Tamar et al. (2015); Petrik and Subramanian (2012) consider CVaR-based formulations, and the works in Tallec (2007); Shapiro et al. (2013) consider nested CVaR-based formulations.

Risk-Constrained RL and Paper Contributions: Despite the rather large literature on risk-sensitive MDPs and RL, *risk-constrained* formulations have largely gone unaddressed, with only a few exceptions, e.g., Chow and Pavone (2013); Borkar and Jain (2014). Yet constrained formulations naturally arise in several domains, including engineering, finance, and logistics, and provide a principled approach to address multi-objective problems. The objective of this paper is to fill this gap by devising policy gradient and actor-critic algorithms for risk-constrained MDPs, where risk is represented via a constraint on the conditional value-at-risk (CVaR) of the cumulative cost or as a chance constraint. Specifically, the contribution of this paper is fourfold.

1. We formulate two risk-constrained MDP problems. The first one involves a CVaR constraint and the second one involves a chance constraint. For the CVaR-constrained optimization problem, we consider both discrete and continuous cost distributions. By re-writing the problems using a Lagrangian formulation, we derive for both problems a Bellman optimality condition with respect to an augmented MDP whose state consists of two parts, with the first part capturing the state of the original MDP and the second part keeping track of the cumulative constraint cost.

2. We devise a trajectory-based policy gradient algorithm for both CVaR-constrained and chance-constrained MDPs. The key novelty of this algorithm lies in an unbiased gradient estimation procedure under Monte Carlo sampling. Using an ordinary differential equation (ODE) approach, we establish convergence of the algorithm to locally optimal policies.

3. Using the aforementioned Bellman optimality condition, we derive several actor-critic algorithms to optimize policy and value function approximation parameters in an online fashion. As for the trajectory-based policy gradient algorithm, we show that the proposed actor-critic algorithms converge to locally optimal solutions.

4. We demonstrate the effectiveness of our algorithms in an optimal stopping problem as well as in a realistic personalized advertisement recommendation (ad recommendation) problem (see Derfer et al. (2007) for more details). For the latter problem, we empirically show that our CVaR-constrained RL algorithms successfully guarantee that the worst-case revenue is lower-bounded by the pre-specified company yearly target.

The rest of the paper is structured as follows. In Section 2 we introduce our notation and rigorously state the problem we wish to address, namely risk-constrained RL. The next two sections provide various RL methods to approximately compute (locally) optimal policies for CVaR constrained MDPs. A trajectory-based policy gradient algorithm is presented in Section 3 and its convergence analysis is provided in Appendix A (Appendix A.1 provides the gradient estimates of the CVaR parameter, the policy parameter, and the Lagrange multiplier, and Appendix A.2 gives their convergence proofs). Actor-critic algorithms are presented in Section 4 and their convergence analysis is provided in Appendix B (Appendix B.1 derives the gradient of the Lagrange multiplier as a function of the state-action value function, Appendix B.2.1 analyzes the convergence of the critic, and Appendix B.2.2 provides the multi-timescale convergence results of the CVaR parameter, the policy parameter, and the Lagrange multiplier). Section 5 extends the above policy gradient and actor-critic methods to the chance-constrained case. Empirical evaluation of our algorithms is the subject of Section 6. Finally, we conclude the paper in Section 7, where we also provide directions for future work.

This paper generalizes earlier results by the authors presented in Chow and Ghavamzadeh (2014).

2. Preliminaries

We begin by defining some notation that is used throughout the paper, as well as defining the problem addressed herein and stating some basic assumptions.

2.1 Notation

We consider decision-making problems modeled as a finite MDP (an MDP with finite state and action spaces). A finite MDP is a tuple $(\mathcal{X}, \mathcal{A}, C, D, P, P_0)$ where $\mathcal{X} = \{1, \ldots, n, x_{\text{Tar}}\}$ and $\mathcal{A} = \{1, \ldots, m\}$ are the state and action spaces, x_{Tar} is a recurrent target state, and for a state x and an action a, $C(x, a)$ is a cost function with $|C(x, a)| \leq C_{\max}$, $D(x, a)$ is a constraint cost function with $|D(x, a)| \leq D_{\max}$ [1], $P(\cdot|x, a)$ is the transition probability distribution, and $P_0(\cdot)$ is the initial state distribution. For simplicity, in this paper we assume $P_0 = \mathbf{1}\{x = x^0\}$ for some given initial state $x^0 \in \{1, \ldots, n\}$. Generalizations to non-atomic initial state distributions are straightforward, for which the details are omitted for the sake of brevity. A *stationary policy* $\mu(\cdot|x)$ for an MDP is a probability distribution over actions, conditioned on the current state. In policy gradient methods, such policies are parameterized by a κ-dimensional vector θ, so the space of policies can be written as $\{\mu(\cdot|x; \theta), x \in \mathcal{X}, \theta \in \Theta \subseteq R^\kappa\}$. Since in this setting a policy μ is uniquely defined by its parameter vector θ, policy-dependent functions can be written as a function of μ or θ, and we use $\mu(\cdot|x; \theta)$ to denote the policy and θ to denote the dependency on the policy (parameter).

Given a fixed $\gamma \in (0, 1)$, we denote by $d^\mu_\gamma(x|x^0) = (1-\gamma)\sum_{k=0}^\infty \gamma^k \mathbb{P}(x_k = x|x_0 = x^0; \mu)$ and $\pi^\mu_\gamma(x, a|x^0) = d^\mu_\gamma(x|x^0)\mu(a|x)$, the γ-discounted occupation measure of state x and state-action pair (x, a) under policy μ, respectively. This occupation measure is a γ-discounted probability distribution for visiting each state and action pair, and it plays an important role in sampling states and actions from the real system in policy gradient and actor-critic algorithms, and in guaranteeing their convergence. Because the state and action spaces are finite, Theorem 3.1 in Altman (1999) shows that the occupation measure $d^\mu_\gamma(x|x^0)$ is a well-defined probability distribution. On the other hand, when $\gamma = 1$ the occupation measure of state x and state-action pair (x, a) under policy μ are respectively defined by $d^\mu(x|x^0) = \sum_{t=0}^\infty \mathbb{P}(x_t = x|x^0; \mu)$ and $\pi^\mu(x, a|x^0) = d^\mu(x|x^0)\mu(a|x)$. In this case the occupation measures characterize the total sums of visiting probabilities (although they are not in general probability distributions themselves) of state x and state-action pair (x, a). To study the well-posedness of the occupation measure, we define the following notion of a transient MDP.

Definition 1 *Define $\mathcal{X}' = \mathcal{X} \setminus \{x_{Tar}\} = \{1, \ldots, n\}$ as a state space of transient states. An MDP is said to be transient if,*

1. *$\sum_{k=0}^\infty \mathbb{P}(x_k = x|x^0, \mu) < \infty$ for every $x \in \mathcal{X}'$ and every stationary policy μ,*

2. *$P(x_{Tar}|x_{Tar}, a) = 1$ for every admissible control action $a \in \mathcal{A}$.*

Furthermore let $T_{\mu,x}$ be the first-hitting time of the target state x_{Tar} from an arbitrary initial state $x \in \mathcal{X}$ in the Markov chain induced by transition probability $P(\cdot|x, a)$ and policy μ. Although transience implies the first-hitting time is square integrable and finite almost surely, we will make the stronger assumption (which implies transience) on the uniform boundedness of the first-hitting time.

1. Without loss of generality, we set the cost function $C(x, a)$ and constraint cost function $D(x, a)$ to zero when $x = x_{\text{Tar}}$.

Assumption 2 *The first-hitting time $T_{\mu,x}$ is bounded almost surely over all stationary policies μ and all initial states $x \in \mathcal{X}$. We will refer to this upper bound as T, i.e., $T_{\mu,x} \leq T$ almost surely.*

The above assumption can be justified by the fact that sample trajectories collected in most reinforcement learning algorithms (including policy gradient and actor-critic methods) consist of bounded finite stopping time (also known as a time-out). Note that although a bounded stopping time would seem to conflict with the time-stationarity of the transition probabilities, this can be resolved by augmenting the state space with a time-counter state, analogous to the arguments given in Section 4.7 in Bertsekas (1995).

Finally, we define the constraint and cost functions. Let Z be a finite-mean ($\mathbb{E}[|Z|] < \infty$) random variable representing cost, with the cumulative distribution function $F_Z(z) = \mathbb{P}(Z \leq z)$ (e.g., one may think of Z as the total cost of an investment strategy μ). We define the *value-at-risk* at confidence level $\alpha \in (0,1)$ as

$$\text{VaR}_\alpha(Z) = \min\big\{z \mid F_Z(z) \geq \alpha\big\}.$$

Here the minimum is attained because F_Z is non-decreasing and right-continuous in z. When F_Z is continuous and strictly increasing, $\text{VaR}_\alpha(Z)$ is the unique z satisfying $F_Z(z) = \alpha$. As mentioned, we refer to a constraint on the VaR as a chance constraint.

Although VaR is a popular risk measure, it is not a *coherent* risk measure (Artzner et al., 1999) and does not quantify the costs that might be suffered beyond its value in the α-tail of the distribution (Rockafellar and Uryasev, 2000), Rockafellar and Uryasev (2002). In many *financial applications* such as portfolio optimization where the probability of undesirable events could be small but the cost incurred could still be significant, besides describing risk as the probability of incurring costs, it will be more interesting to study the cost in the tail of the risk distribution. In this case, an alternative measure that addresses most of the VaR's shortcomings is the *conditional value-at-risk*, defined as (Rockafellar and Uryasev, 2000)

$$\text{CVaR}_\alpha(Z) := \min_{\nu \in \mathbb{R}} \Big\{\nu + \frac{1}{1-\alpha}\mathbb{E}\big[(Z-\nu)^+\big]\Big\}, \tag{1}$$

where $(x)^+ = \max(x, 0)$ represents the positive part of x. While it might not be an immediate observation, it has been shown in Theorem 1 of Rockafellar and Uryasev (2000) that the CVaR of the loss random variable Z is equal to the average of the worst-case α-fraction of losses.

We define the parameter $\gamma \in (0,1]$ as the *discounting factor* for the cost and constraint cost functions. When $\gamma < 1$, we are aiming to solve the MDP problem with more focus on optimizing current costs over future costs. For a policy μ, we define the cost of a state x (state-action pair (x,a)) as the sum of (discounted) costs encountered by the decision-maker when it starts at state x (state-action pair (x,a)) and then follows policy μ, i.e.,

$$\mathcal{G}^\theta(x) = \sum_{k=0}^{T-1} \gamma^k C(x_k, a_k) \mid x_0 = x,\ \mu(\cdot|\cdot,\theta), \qquad \mathcal{J}^\theta(x) = \sum_{k=0}^{T-1} \gamma^k D(x_k, a_k) \mid x_0 = x,\ \mu(\cdot|\cdot,\theta),$$

and

$$\mathcal{G}^\theta(x,a) = \sum_{k=0}^{T-1} \gamma^k C(x_k, a_k) \mid x_0 = x,\ a_0 = a,\ \mu(\cdot|\cdot,\theta),$$
$$\mathcal{J}^\theta(x,a) = \sum_{k=0}^{T-1} \gamma^k D(x_k, a_k) \mid x_0 = x,\ a_0 = a,\ \mu(\cdot|\cdot,\theta).$$

The expected values of the random variables $\mathcal{G}^\theta(x)$ and $\mathcal{G}^\theta(x,a)$ are known as the value and action-value functions of policy μ, and are denoted by

$$V^\theta(x) = \mathbb{E}\big[\mathcal{G}^\theta(x)\big], \qquad Q^\theta(x,a) = \mathbb{E}\big[\mathcal{G}^\theta(x,a)\big].$$

2.2 Problem Statement

The goal for standard discounted MDPs is to find an optimal policy that solves

$$\theta^* = \operatorname*{argmin}_\theta V^\theta(x^0).$$

For *CVaR-constrained* optimization in MDPs, we consider the discounted cost optimization problem with $\gamma \in (0,1)$, i.e., for a given confidence level $\alpha \in (0,1)$ and cost tolerance $\beta \in \mathbb{R}$,

$$\min_\theta V^\theta(x^0) \qquad \text{subject to} \qquad \text{CVaR}_\alpha\big(\mathcal{J}^\theta(x^0)\big) \leq \beta. \tag{2}$$

Using the definition of $H_\alpha(Z,\nu)$, one can reformulate (2) as:

$$\min_{\theta,\nu} V^\theta(x^0) \qquad \text{subject to} \qquad H_\alpha\big(\mathcal{J}^\theta(x^0),\nu\big) \leq \beta, \tag{3}$$

where

$$H_\alpha(Z,\nu) := \nu + \frac{1}{1-\alpha}\mathbb{E}\big[(Z-\nu)^+\big].$$

The equivalence between problem (2) and problem (3) can be shown as follows. Let $\theta_2 \in \Theta$ be any arbitrary feasible policy parameter of problem (2). With θ_2, one can always construct $\nu_2 = \text{VaR}_\alpha(\mathcal{J}^{\theta_2}(x^0))$, such that (θ_2, ν_2) is feasible to problem (3). This in turn implies that the solution of (3) is less than the solution of (2). On the other hand, the following chain of inequalities holds for any $\nu \in \mathbb{R}$: $\text{CVaR}_\alpha\big(\mathcal{J}^\theta(x^0)\big) \leq H_\alpha\big(\mathcal{J}^\theta(x^0),\nu\big) \leq \beta$. This implies that the feasible set of θ in problem (3) is a subset of the feasible set of θ in problem (2), which further indicates that the solution of problem (2) is less than the solution of problem (3). By combining both arguments, one concludes the equivalence relation of these two problems.

It is shown in Rockafellar and Uryasev (2000) and Rockafellar and Uryasev (2002) that the optimal ν actually equals VaR_α, so we refer to this parameter as the VaR parameter. Here we choose to analyze the discounted-cost CVaR-constrained optimization problem, i.e., with $\gamma \in (0,1)$, as in many financial and marketing applications where CVaR constraints are used, it is more intuitive to put more emphasis on current costs rather than on future costs. The analysis can be easily generalized for the case where $\gamma = 1$.

For *chance-constrained* optimization in MDPs, we consider the stopping cost optimization problem with $\gamma = 1$, i.e., for a given confidence level $\beta \in (0,1)$ and cost tolerance $\alpha \in \mathbb{R}$,

$$\min_\theta V^\theta(x^0) \qquad \text{subject to} \qquad \mathbb{P}\big(\mathcal{J}^\theta(x^0) \geq \alpha\big) \leq \beta. \tag{4}$$

Here we choose $\gamma = 1$ because in many engineering applications, where chance constraints are used to ensure overall safety, there is no notion of discounting since future threats are often as important as the current one. Similarly, the analysis can be easily extended to the case where $\gamma \in (0,1)$.

There are a number of mild technical and notational assumptions which we will make throughout the paper, so we state them here:

Assumption 3 (Differentiability) *For any state-action pair (x,a), $\mu(a|x;\theta)$ is continuously differentiable in θ and $\nabla_\theta \mu(a|x;\theta)$ is a Lipschitz function in θ for every $a \in \mathcal{A}$ and $x \in \mathcal{X}$.*[2]

2. In actor-critic algorithms, the assumption on continuous differentiability holds for the augmented state Markovian policies $\mu(a|x,s;\theta)$.

Assumption 4 (Strict Feasibility) *There exists a transient policy $\mu(\cdot|x;\theta)$ such that*

$$H_\alpha\big(\mathcal{J}^\theta(x^0),\nu\big) < \beta$$

in the CVaR-constrained optimization problem, and $P\big(\mathcal{J}^\theta(x^0) \geq \alpha\big) < \beta$ in the chance-constrained problem.

In the remainder of the paper we first focus on studying stochastic approximation algorithms for the CVaR-constrained optimization problem (Sections 3 and 4) and then adapt the results to the chance-constrained optimization problem in Section 5. Our solution approach relies on a Lagrangian relaxation procedure, which is discussed next.

2.3 Lagrangian Approach and Reformulation

To solve (3), we employ a Lagrangian relaxation procedure (Chapter 3 of Bertsekas (1999)), which leads to the unconstrained problem:

$$\max_{\lambda\geq 0}\min_{\theta,\nu}\Big(L(\nu,\theta,\lambda) := V^\theta(x^0) + \lambda\Big(H_\alpha\big(\mathcal{J}^\theta(x^0),\nu\big) - \beta\Big)\Big), \tag{5}$$

where λ is the Lagrange multiplier. Notice that $L(\nu,\theta,\lambda)$ is a linear function in λ and $H_\alpha\big(\mathcal{J}^\theta(x^0),\nu\big)$ is a continuous function in ν. The saddle point theorem from Chapter 3 of Bertsekas (1999) states that a local saddle point $(\nu^*,\theta^*,\lambda^*)$ for the maximin optimization problem $\max_{\lambda\geq 0}\min_{\theta,\nu} L(\nu,\theta,\lambda)$ is indeed a locally optimal policy θ^* for the CVaR-constrained optimization problem. To further explore this connection, we first have the following definition of a saddle point:

Definition 5 *A local saddle point of $L(\nu,\theta,\lambda)$ is a point $(\nu^*,\theta^*,\lambda^*)$ such that for some $r>0$, $\forall(\theta,\nu)\in\Theta\times\Big[-\frac{D_{\max}}{1-\gamma},\frac{D_{\max}}{1-\gamma}\Big]\cap\mathcal{B}_{(\theta^*,\nu^*)}(r)$ and $\forall\lambda\geq 0$, we have*

$$L(\nu,\theta,\lambda^*) \geq L(\nu^*,\theta^*,\lambda^*) \geq L(\nu^*,\theta^*,\lambda), \tag{6}$$

where $\mathcal{B}_{(\theta^*,\nu^*)}(r)$ is a hyper-dimensional ball centered at (θ^*,ν^*) with radius $r>0$.

In Chapter 7 of Ott (2010) and in Bäuerle and Ott (2011) it is shown that there exists a *deterministic history-dependent* optimal policy for CVaR-constrained optimization. The important point is that this policy does not depend on the complete history, but only on the current time step k, current state of the system x_k, and accumulated discounted constraint cost $\sum_{i=0}^k \gamma^i D(x_k,a_k)$.

In the following two sections, we present a policy gradient (PG) algorithm (Section 3) and several actor-critic (AC) algorithms (Section 4) to optimize (5) (and hence find a locally optimal solution to problem (3)). While the PG algorithm updates its parameters after observing several trajectories, the AC algorithms are incremental and update their parameters at each time-step.

3. A Trajectory-based Policy Gradient Algorithm

In this section, we present a policy gradient algorithm to solve the optimization problem (5). The idea of the algorithm is to descend in (θ,ν) and ascend in λ using the gradients of $L(\nu,\theta,\lambda)$ w.r.t. θ, ν, and λ, i.e.,[3]

$$\nabla_\theta L(\nu,\theta,\lambda) = \nabla_\theta V^\theta(x^0) + \frac{\lambda}{(1-\alpha)}\nabla_\theta\mathbb{E}\Big[\big(\mathcal{J}^\theta(x^0)-\nu\big)^+\Big], \tag{7}$$

$$\partial_\nu L(\nu,\theta,\lambda) = \lambda\Bigg(1+\frac{1}{(1-\alpha)}\partial_\nu\mathbb{E}\Big[\big(\mathcal{J}^\theta(x^0)-\nu\big)^+\Big]\Bigg) \ni \lambda\Bigg(1-\frac{1}{(1-\alpha)}\mathbb{P}\big(\mathcal{J}^\theta(x^0)\geq\nu\big)\Bigg), \tag{8}$$

$$\nabla_\lambda L(\nu,\theta,\lambda) = \nu + \frac{1}{(1-\alpha)}\mathbb{E}\Big[\big(\mathcal{J}^\theta(x^0)-\nu\big)^+\Big] - \beta. \tag{9}$$

The unit of observation in this algorithm is a trajectory generated by following the current policy. At each iteration, the algorithm generates N trajectories by following the current policy, uses them to estimate the gradients in (7)–(9), and then uses these estimates to update the parameters ν,θ,λ.

Let $\xi=\{x_0,a_0,c_0,x_1,a_1,c_1,\ldots,x_{T-1},a_{T-1},c_{T-1},x_T\}$ be a trajectory generated by following the policy θ, where $x_T = x_{\text{Tar}}$ is the target state of the system. The cost, constraint cost, and probability of ξ are defined as $\mathcal{G}(\xi)=\sum_{k=0}^{T-1}\gamma^k C(x_k,a_k)$, $\mathcal{J}(\xi)=\sum_{k=0}^{T-1}\gamma^k D(x_k,a_k)$, and $\mathbb{P}_\theta(\xi)=P_0(x_0)\prod_{k=0}^{T-1}\mu(a_k|x_k;\theta)P(x_{k+1}|x_k,a_k)$, respectively. Based on the definition of $\mathbb{P}_\theta(\xi)$, one obtains $\nabla_\theta\log\mathbb{P}_\theta(\xi)=\sum_{k=0}^{T-1}\nabla_\theta\log\mu(a_k|x_k;\theta)$.

Algorithm 1 contains the pseudo-code of our proposed policy gradient algorithm. What appears inside the parentheses on the right-hand-side of the update equations are the estimates of the gradients of $L(\nu,\theta,\lambda)$ w.r.t. θ,ν,λ (estimates of (7)–(9)). Gradient estimates of the Lagrangian function can be found in Appendix A.1. In the algorithm, Γ_Θ is an operator that projects a vector $\theta\in\mathbb{R}^\kappa$ to the closest point in a compact and convex set $\Theta\subset\mathbb{R}^\kappa$, i.e., $\Gamma_\Theta(\theta)=\arg\min_{\hat{\theta}\in\Theta}\|\theta-\hat{\theta}\|_2^2$, $\Gamma_\mathcal{N}$ is a projection operator to $[-\frac{D_{\max}}{1-\gamma},\frac{D_{\max}}{1-\gamma}]$, i.e., $\Gamma_\mathcal{N}(\nu)=\arg\min_{\hat{\nu}\in[-\frac{D_{\max}}{1-\gamma},\frac{D_{\max}}{1-\gamma}]}\|\nu-\hat{\nu}\|_2^2$, and Γ_Λ is a projection operator to $[0,\lambda_{\max}]$, i.e., $\Gamma_\Lambda(\lambda)=\arg\min_{\hat{\lambda}\in[0,\lambda_{\max}]}\|\lambda-\hat{\lambda}\|_2^2$. These projection operators are necessary to ensure the convergence of the algorithm; see the end of Appendix A.2 for details. Next we introduce the following assumptions for the step-sizes of the policy gradient method in Algorithm 1.

Assumption 6 (Step Sizes for Policy Gradient) *The step size schedules $\{\zeta_1(k)\}$, $\{\zeta_2(k)\}$, and $\{\zeta_3(k)\}$ satisfy*

$$\sum_k\zeta_1(k)=\sum_k\zeta_2(k)=\sum_k\zeta_3(k)=\infty, \tag{10}$$

$$\sum_k\zeta_1(k)^2,\quad\sum_k\zeta_2(k)^2,\quad\sum_k\zeta_3(k)^2<\infty, \tag{11}$$

$$\zeta_1(k)=o\big(\zeta_2(k)\big),\quad\zeta_2(k)=o\big(\zeta_3(k)\big). \tag{12}$$

These step-size schedules satisfy the standard conditions for stochastic approximation algorithms, and ensure that the ν update is on the fastest time-scale $\{\zeta_3(k)\}$, the policy θ update is on the intermediate time-scale $\{\zeta_2(k)\}$, and the Lagrange multiplier λ update is on the slowest time-scale $\{\zeta_1(k)\}$. This results in a three time-scale stochastic approximation algorithm.

In the following theorem, we prove that our policy gradient algorithm converges to a locally optimal policy for the CVaR-constrained optimization problem.

3. The notation $\ni$ in (8) means that the right-most term is a member of the sub-gradient set $\partial_\nu L(\nu,\theta,\lambda)$.

Theorem 7 *Under Assumptions 2–6, the sequence of policy updates in Algorithm 1 converges almost surely to a locally optimal policy θ^* for the CVaR-constrained optimization problem as k goes to infinity.*

While we refer the reader to Appendix A.2 for the technical details of this proof, a high level overview of the proof technique is given as follows.

1. First we show that each update of the multi-time scale discrete stochastic approximation algorithm $(\nu_k, \theta_k, \lambda_k)$ converges almost surely, but at different speeds, to the stationary point $(\nu^*, \theta^*, \lambda^*)$ of the corresponding continuous time system.

2. Then by using Lyapunov analysis, we show that the continuous time system is locally asymptotically stable at the stationary point $(\nu^*, \theta^*, \lambda^*)$.

3. Since the Lyapunov function used in the above analysis is the Lagrangian function $L(\nu, \theta, \lambda)$, we finally conclude that the stationary point $(\nu^*, \theta^*, \lambda^*)$ is also a local saddle point, which by the saddle point theorem (see e.g., Chapter 3 of Bertsekas (1999)), implies that θ^* is a locally optimal solution of the CVaR-constrained MDP problem (the primal problem).

This convergence proof procedure is standard for stochastic approximation algorithms, see (Bhatnagar et al., 2009; Bhatnagar and Lakshmanan, 2012; Prashanth and Ghavamzadeh, 2013) for more details, and represents the structural backbone for the convergence analysis of the other policy gradient and actor-critic methods provided in this paper.

Notice that the difference in convergence speeds between θ_k, ν_k, and λ_k is due to the step-size schedules. Here ν converges faster than θ and θ converges faster than λ. This multi-time scale convergence property allows us to simplify the convergence analysis by assuming that θ and λ are fixed in ν's convergence analysis, assuming that ν converges to $\nu^*(\theta)$ and λ is fixed in θ's convergence analysis, and finally assuming that ν and θ have already converged to $\nu^*(\lambda)$ and $\theta^*(\lambda)$ in λ's convergence analysis. To illustrate this idea, consider the following two-time scale stochastic approximation algorithm for updating $(x_k, y_k) \in \mathbf{X} \times \mathbf{Y}$:

$$x_{k+1} = x_k + \zeta_1(k)\big(f(x_k, y_k) + M_{k+1}\big), \tag{13}$$

$$y_{k+1} = y_k + \zeta_2(k)\big(g(x_k, y_k) + N_{k+1}\big), \tag{14}$$

where $f(x_k, y_k)$ and $g(x_k, y_k)$ are Lipschitz continuous functions, M_{k+1}, N_{k+1} are square integrable Martingale differences w.r.t. the σ-fields $\sigma(x_i, y_i, M_i, i \leq k)$ and $\sigma(x_i, y_i, N_i, i \leq k)$, and $\zeta_1(k)$ and $\zeta_2(k)$ are non-summable, square summable step sizes. If $\zeta_2(k)$ converges to zero faster than $\zeta_1(k)$, then (13) is a faster recursion than (14) after some iteration k_0 (i.e., for $k \geq k_0$), which means (13) has uniformly larger increments than (14). Since (14) can be written as

$$y_{k+1} = y_k + \zeta_1(k)\Big(\frac{\zeta_2(k)}{\zeta_1(k)}\big(g(x_k, y_k) + N_{k+1}\big)\Big),$$

and by the fact that $\zeta_2(k)$ converges to zero faster than $\zeta_1(k)$, (13) and (14) can be viewed as noisy Euler discretizations of the ODEs $\dot{x} = f(x, y)$ and $\dot{y} = 0$. Note that one can consider the ODE $\dot{x} = f(x, y_0)$ in place of $\dot{x} = f(x, y)$, where y_0 is constant, because $\dot{y} = 0$. One can then show (see e.g., Theorem 2 in Chapter 6 of Borkar (2008)) the main two-timescale convergence result, i.e., under the above assumptions associated with (14), the sequence (x_k, y_k) converges to $\big(\mu(y^\star), y^\star\big)$ as $i \to \infty$, with probability one, where $\mu(y_0)$ is a locally asymptotically stable equilibrium of the ODE $\dot{x} = f(x, y_0)$, μ is a Lipschitz continuous function, and $y^\star$ is a locally asymptotically stable equilibrium of the ODE $\dot{y} = g\big(\mu(y), y\big)$.

Algorithm 1 Trajectory-based Policy Gradient Algorithm for CVaR MDP

Input: parameterized policy $\mu(\cdot|\cdot;\theta)$, confidence level α, and cost tolerance β
Initialization: policy $\theta = \theta_0$, VaR parameter $\nu = \nu_0$, and the Lagrangian parameter $\lambda = \lambda_0$
while TRUE **do**
  for $k = 0, 1, 2, \ldots$ **do**
    Generate N trajectories $\{\xi_{j,k}\}_{j=1}^N$ by starting at $x_0 = x^0$ and following the current policy θ_k.

$$\nu \textbf{ Update:} \quad \nu_{k+1} = \Gamma_{\mathcal{N}}\Big[\nu_k - \zeta_3(k)\Big(\lambda_k - \frac{\lambda_k}{(1-\alpha)N}\sum_{j=1}^N \mathbf{1}\big\{\mathcal{J}(\xi_{j,k}) \geq \nu_k\big\}\Big)\Big]$$

$$\theta \textbf{ Update:} \quad \theta_{k+1} = \Gamma_\Theta\Big[\theta_k - \zeta_2(k)\Big(\frac{1}{N}\sum_{j=1}^N \nabla_\theta \log \mathbb{P}_\theta(\xi_{j,k})|_{\theta=\theta_k}\mathcal{G}(\xi_{j,k}) + \frac{\lambda_k}{(1-\alpha)N}\sum_{j=1}^N \nabla_\theta \log \mathbb{P}_\theta(\xi_{j,k})|_{\theta=\theta_k}\big(\mathcal{J}(\xi_{j,k}) - \nu_k\big)\mathbf{1}\big\{\mathcal{J}(\xi_{j,k}) \geq \nu_k\big\}\Big)\Big]$$

$$\lambda \textbf{ Update:} \quad \lambda_{k+1} = \Gamma_\Lambda\Big[\lambda_k + \zeta_1(k)\Big(\nu_k - \beta + \frac{1}{(1-\alpha)N}\sum_{j=1}^N \big(\mathcal{J}(\xi_{j,k}) - \nu_k\big)\mathbf{1}\big\{\mathcal{J}(\xi_{j,k}) \geq \nu_k\big\}\Big)\Big]$$

  end for
  if $\{\lambda_k\}$ converges to $\lambda_{\max}$, i.e., $|\lambda_{i^*} - \lambda_{\max}| \leq \epsilon$ for some tolerance parameter $\epsilon > 0$ **then**
    Set $\lambda_{\max} \leftarrow 2\lambda_{\max}$.
  else
    return parameters ν, θ, λ and **break**
  end if
end while

4. Actor-Critic Algorithms

As mentioned in Section 3, the unit of observation in our policy gradient algorithm (Algorithm 1) is a system trajectory. This may result in high variance for the gradient estimates, especially when the length of the trajectories is long. To address this issue, in this section we propose two actor-critic algorithms that approximate some quantities in the gradient estimates by linear combinations of basis functions and update the parameters (linear coefficients) incrementally (after each state-action transition). We present two actor-critic algorithms for optimizing (5). These algorithms are based on the gradient estimates of Sections 4.1-4.3. While the first algorithm (SPSA-based) is fully incremental and updates all the parameters θ, ν, λ at each time-step, the second one updates θ at each time-step and updates ν and λ only at the end of each trajectory, thus is regarded as a semi-trajectory-based method. Algorithm 2 contains the pseudo-code of these algorithms. The projection operators Γ_Θ, $\Gamma_\mathcal{N}$, and Γ_Λ are defined as in Section 3 and are necessary to ensure the convergence of the algorithms. At each step of our actor critic algorithms (steps indexed by k in Algorithm 1 and in Algorithm 2) there are two parts:

- **Inner loop (critic update)**: For a fixed policy (given as $\theta \in \Theta$), take action $a_k \sim \mu(\cdot|x_k, s_k; \theta_k)$, observe the cost $c(x_k, a_k)$, the constraint cost $d(x_k, a_k)$, and the next state (x_{k+1}, s_{k+1}). Us-

ing the method of temporal differences (TD) from Chapter 6 of Sutton and Barto (1998), estimate the value function $V^\theta(x,s)$.

- **Outer loop (actor update)**: Estimate the gradient of $V^\theta(x,s)$ for policy parameter θ, and hence the gradient of the Lagrangian $L(\nu,\theta,\lambda)$, using the unbiased sampling based point estimator for gradients with respect to θ and λ and either: (1) using the SPSA method (20) to obtain an incremental estimator for gradient with respect to ν or (2) only calculating the gradient estimator with respect to ν at the end of the trajectory (see (23) for more details). Update the policy parameter $\theta \in \Theta$ in the descent direction, the VaR approximation $\nu \in \mathcal{N}$ in the descent direction, and the Lagrange multiplier $\lambda \in \Lambda$ in the ascent direction on specific timescales that ensure convergence to locally optimal solutions.

Next, we introduce the following assumptions for the step-sizes of the actor-critic method in Algorithm 2.

Assumption 8 (Step Sizes) *The step size schedules $\{\zeta_1(k)\}$, $\{\zeta_2(k)\}$, $\{\zeta_3(k)\}$, and $\{\zeta_4(k)\}$ satisfy*

$$\sum_k \zeta_1(k) = \sum_k \zeta_2(k) = \sum_k \zeta_3(k) = \sum_k \zeta_4(k) = \infty, \tag{15}$$

$$\sum_k \zeta_1(k)^2, \quad \sum_k \zeta_2(k)^2, \quad \sum_k \zeta_3(k)^2, \quad \sum_k \zeta_4(k)^2 < \infty, \tag{16}$$

$$\zeta_1(k) = o\big(\zeta_2(k)\big), \quad \zeta_2(k) = o\big(\zeta_3(k)\big), \quad \zeta_3(k) = o\big(\zeta_4(k)\big). \tag{17}$$

Furthermore, the SPSA step size $\{\Delta_k\}$ in the actor-critic algorithm satisfies $\Delta_k \to 0$ as $k \to \infty$ and $\sum_k (\zeta_2(k)/\Delta_k)^2 < \infty$.

These step-size schedules satisfy the standard conditions for stochastic approximation algorithms, and ensure that the critic update is on the fastest time-scale $\big\{\zeta_4(k)\big\}$, the policy and VaR parameter updates are on the intermediate time-scale, with the ν-update $\big\{\zeta_3(k)\big\}$ being faster than the θ-update $\big\{\zeta_2(k)\big\}$, and finally the Lagrange multiplier update is on the slowest time-scale $\big\{\zeta_1(k)\big\}$. This results in four time-scale stochastic approximation algorithms.

4.1 Gradient w.r.t. the Policy Parameters θ

The gradient of the objective function w.r.t. the policy θ in (7) may be rewritten as

$$\nabla_\theta L(\nu,\theta,\lambda) = \nabla_\theta \left(\mathbb{E}\big[\mathcal{G}^\theta(x^0)\big] + \frac{\lambda}{(1-\alpha)} \mathbb{E}\Big[\big(\mathcal{J}^\theta(x^0) - \nu\big)^+\Big] \right). \tag{24}$$

Given the original MDP $\mathcal{M} = (\mathcal{X}, \mathcal{A}, C, D, P, P_0)$ and the parameter λ, we define the augmented MDP $\bar{\mathcal{M}} = (\bar{\mathcal{X}}, \bar{\mathcal{A}}, \bar{C}_\lambda, \bar{P}, \bar{P}_0)$ as $\bar{\mathcal{X}} = \mathcal{X} \times \mathcal{S}$, $\bar{\mathcal{A}} = \mathcal{A}$, $\bar{P}_0(x,s) = P_0(x)\mathbf{1}\{s_0 = s\}$, and

$$\bar{C}_\lambda(x,s,a) = \begin{cases} \lambda(-s)^+/(1-\alpha) & \text{if } x = x_{\text{Tar}}, \\ C(x,a) & \text{otherwise}, \end{cases}$$

$$\bar{P}(x',s'|x,s,a) = \begin{cases} P(x'|x,a)\mathbf{1}\{s' = \big(s - D(x,a)\big)/\gamma\} & \text{if } x \in \mathcal{X}', \\ \mathbf{1}\{x' = x_{\text{Tar}}, s' = 0\} & \text{if } x = x_{\text{Tar}}, \end{cases}$$

where x_{Tar} is the target state of the original MDP $\mathcal{M}$, $\mathcal{S}$ and s_0 are respectively the finite state space and the initial state of the s part of the state in the augmented MDP $\bar{\mathcal{M}}$. Furthermore, we denote by

Algorithm 2 Actor-Critic Algorithms for CVaR MDP

Input: Parameterized policy $\mu(\cdot|\cdot;\theta)$ and value function feature vector $\phi(\cdot)$ (both over the augmented MDP $\bar{\mathcal{M}}$), confidence level α, and cost tolerance β
Initialization: policy $\theta = \theta_0$; VaR parameter $\nu = \nu_0$; Lagrangian parameter $\lambda = \lambda_0$; value function weight vector $v = v_0$; initial condition $(x_0, s_0) = (x^0, \nu)$
while TRUE **do**
  // (1) SPSA-based Algorithm:
  for $k = 0, 1, 2, \ldots$ **do**
    Draw action $a_k \sim \mu(\cdot|x_k, s_k; \theta_k)$; Observe cost $\bar{C}_{\lambda_k}(x_k, s_k, a_k)$;
    Observe next state $(x_{k+1}, s_{k+1}) \sim \bar{P}(\cdot|x_k, s_k, a_k)$; *// note that* $s_{k+1} = (s_k - D(x_k, a_k))/\gamma$
    // AC Algorithm:

$$\textbf{TD Error:} \quad \delta_k(v_k) = \bar{C}_{\lambda_k}(x_k, s_k, a_k) + \gamma v_k^\top \phi(x_{k+1}, s_{k+1}) - v_k^\top \phi(x_k, s_k) \tag{18}$$

$$\textbf{Critic Update:} \quad v_{k+1} = v_k + \zeta_4(k)\delta_k(v_k)\phi(x_k, s_k) \tag{19}$$

$$\nu \textbf{ Update:} \quad \nu_{k+1} = \Gamma_{\mathcal{N}}\left(\nu_k - \zeta_3(k)\Big(\lambda_k + \frac{v_k^\top\big[\phi\big(x^0, \nu_k + \Delta_k\big) - \phi\big(x^0, \nu_k - \Delta_k\big)\big]}{2\Delta_k}\Big)\right) \tag{20}$$

$$\theta \textbf{ Update:} \quad \theta_{k+1} = \Gamma_\Theta\Big(\theta_k - \frac{\zeta_2(k)}{1-\gamma}\nabla_\theta \log \mu(a_k|x_k, s_k; \theta) \cdot \delta_k(v_k)\Big) \tag{21}$$

$$\lambda \textbf{ Update:} \quad \lambda_{k+1} = \Gamma_\Lambda\Big(\lambda_k + \zeta_1(k)\big(\nu_k - \beta + \frac{1}{(1-\alpha)(1-\gamma)}\mathbf{1}\{x_k = x_{\text{Tar}}\}(-s_k)^+\big)\Big) \tag{22}$$

    if $x_k = x_{\text{Tar}}$ (reach a target state), **then** set $(x_{k+1}, s_{k+1}) = (x^0, \nu_{k+1})$
  end for
  // (2) Semi Trajectory-based Algorithm:
  Initialize $t = 0$
  for $k = 0, 1, 2, \ldots$ **do**
    Draw action $a_k \sim \mu(\cdot|x_k, s_k; \theta_k)$, observe cost $\bar{C}_{\lambda_k}(x_k, s_k, a_k)$, and next state $(x_{k+1}, s_{k+1}) \sim \bar{P}(\cdot|x_k, s_k, a_k)$; Update $(\delta_k, v_k, \theta_k, \lambda_k)$ using Eqs. (18), (19), (21), and (22)
    if $x_k = x_{\text{Tar}}$ **then**
      Update ν as

$$\nu \textbf{ Update:} \quad \nu_{k+1} = \Gamma_{\mathcal{N}}\left(\nu_k - \zeta_3(k)\Big(\lambda_k - \frac{\lambda_k}{1-\alpha}\mathbf{1}\big\{x_k = x_{\text{Tar}}, s_k \le 0\big\}\Big)\right) \tag{23}$$

      Set $(x_{k+1}, s_{k+1}) = (x^0, \nu_{k+1})$ and $t = 0$
    else
      $t \leftarrow t + 1$
    end if
  end for
  if $\{\lambda_k\}$ converges to $\lambda_{\max}$, i.e., $|\lambda_{i^*} - \lambda_{\max}| \le \epsilon$ for some tolerance parameter $\epsilon > 0$ **then**
    Set $\lambda_{\max} \leftarrow 2\lambda_{\max}$.
  else
    return parameters $v, w, \nu, \theta, \lambda$, and **break**
  end if
end while

s_{Tar} the s part of the state in $\bar{\mathcal{M}}$ when a policy θ reaches a target state x_{Tar} (which we assume occurs before an upper-bound T number of steps), i.e.,

$$s_{\text{Tar}} = \frac{1}{\gamma^T}\left(\nu - \sum_{k=0}^{T-1}\gamma^k D(x_k, a_k)\right),$$

such that the initial state is given by $s_0 = \nu$. We will now use $n = |\bar{\mathcal{X}}|$ to indicate the size of the *augmented* state space $\bar{\mathcal{X}}$ instead of the size of the original state space $\mathcal{X}$. It can be later seen that the augmented state s in the MDP $\bar{\mathcal{M}}$ keeps track of the cumulative CVaR constraint cost. Similar to the analysis in Bäuerle and Ott (2011), the major motivation of introducing the aforementioned augmented MDP $\bar{\mathcal{M}}$ is that, by utilizing the augmented state $s \in \mathcal{S}$ that monitors the running constraint cost and thus the feasibility region of the original CVaR constrained MDP, one is able to define a Bellman operator on $\bar{\mathcal{M}}$ (whose exact definition can be found in Theorem 10), whose fixed point solution is equal to the solution of the original CVaR Lagrangian problem. Therefore by combining these properties, this reformulation allows one to transform the CVaR Lagrangian problem to a standard MDP problem.

We define a class of parameterized stochastic policies $\{\mu(\cdot|x, s; \theta), (x, s) \in \bar{\mathcal{X}}, \theta \in \Theta \subseteq R^{\kappa_1}\}$ for this augmented MDP. Recall that $\mathcal{G}^\theta(x)$ is the discounted cumulative cost and $\mathcal{J}^\theta(x)$ is the discounted cumulative constraint cost. Therefore, the total (discounted) cost of a trajectory can be written as

$$\sum_{k=0}^{T}\gamma^k \bar{C}_\lambda(x_k, s_k, a_k) \mid x_0 = x, s_0 = s, \mu \;=\; \mathcal{G}^\theta(x) + \frac{\lambda}{(1-\alpha)}\big(\mathcal{J}^\theta(x) - s\big)^+. \tag{25}$$

From (25), it is clear that the quantity in the parenthesis of (24) is the value function of the policy θ at state (x^0, ν) in the augmented MDP $\bar{\mathcal{M}}$, i.e., $V^\theta(x^0, \nu)$. Thus, it is easy to show that[4]

$$\nabla_\theta L(\nu, \theta, \lambda) = \nabla_\theta V^\theta(x^0, \nu) = \frac{1}{1-\gamma}\sum_{x,s,a}\pi_\gamma^\theta(x, s, a|x^0, \nu)\,\nabla \log \mu(a|x, s; \theta)\,Q^\theta(x, s, a),^5 \tag{26}$$

where π_γ^θ is the discounted occupation measure (defined in Section 2) and Q^θ is the action-value function of policy θ in the augmented MDP $\bar{\mathcal{M}}$. We can show that $\frac{1}{1-\gamma}\nabla \log \mu(a_k|x_k, s_k; \theta)\cdot\delta_k$ is an unbiased estimate of $\nabla_\theta L(\nu, \theta, \lambda)$, where

$$\delta_k = \bar{C}_\lambda(x_k, s_k, a_k) + \gamma\widehat{V}(x_{k+1}, s_{k+1}) - \widehat{V}(x_k, s_k)$$

is the temporal-difference (TD) error in the MDP $\bar{\mathcal{M}}$ from (18), and $\widehat{V}$ is an unbiased estimator of V^θ (see e.g., Bhatnagar et al. (2009)). In our actor-critic algorithms, the critic uses linear approximation for the value function $V^\theta(x, s) \approx v^\top\phi(x, s) = \tilde{V}^{\theta,v}(x, s)$, where the feature vector $\phi(\cdot)$ belongs to a low-dimensional space $\mathbb{R}^{\kappa_1}$ with dimension κ_1. The linear approximation $\tilde{V}^{\theta,v}$ belongs to a low-dimensional subspace $S_V = \{\Phi v | v \in \mathbb{R}^{\kappa_1}\}$, where Φ is the $n \times \kappa_1$ matrix whose

4. Note that the second equality in Equation (26) is the result of the policy gradient theorem (Sutton et al., 2000; Peters et al., 2005).
5. Notice that the state and action spaces of the original MDP are finite, and there is only a finite number of outcomes in the transition of s (due to the assumption of a bounded first hitting time). Therefore the augmented state s belongs to a finite state space as well.

rows are the transposed feature vectors $\phi^\top(\cdot)$. To ensure that the set of feature vectors forms a well-posed linear approximation to the value function, we impose the following assumption on the basis functions.

Assumption 9 (Independent Basis Functions) *The basis functions $\{\phi^{(i)}\}_{i=1}^{\kappa_1}$ are linearly independent. In particular, $\kappa_1 \leq n$ and Φ is full column rank. Moreover, for every $v \in \mathbb{R}^{\kappa_1}$, $\Phi v \neq e$, where e is the n-dimensional vector with all entries equal to one.*

The following theorem shows that the critic update v_k converges almost surely to v^*, the minimizer of the Bellman residual. Details of the proof can be found in Appendix B.2.

Theorem 10 *Define $v^* \in \arg\min_v \|B_\theta[\Phi v] - \Phi v\|^2_{d_\gamma^\theta}$ as the minimizer to the Bellman residual, where the Bellman operator is given by*

$$B_\theta[V](x, s) = \sum_a \mu(a|x, s; \theta)\left\{\bar{C}_\lambda(x, s, a) + \sum_{x', s'}\gamma\bar{P}(x', s'|x, s, a)V(x', s')\right\}$$

and $\tilde{V}^*(x, s) = (v^*)^\top\phi(x, s)$ is the projected Bellman fixed point of $V^\theta(x, s)$, i.e., $\tilde{V}^*(x, s) = \Pi B_\theta[\tilde{V}^*](x, s)$. Suppose the γ-occupation measure π_γ^θ is used to generate samples of (x_k, s_k, a_k) for any $k \in \{0, 1, \ldots, \}$. Then under Assumptions 8–9, the v-update in the actor-critic algorithm converges to v^* almost surely.

4.2 Gradient w.r.t. the Lagrangian Parameter λ

We may rewrite the gradient of the objective function w.r.t. the Lagrangian parameters λ in (9) as

$$\nabla_\lambda L(\nu, \theta, \lambda) = \nu - \beta + \nabla_\lambda\left(\mathbb{E}\big[\mathcal{G}^\theta(x^0)\big] + \frac{\lambda}{(1-\alpha)}\mathbb{E}\Big[\big(\mathcal{J}^\theta(x^0) - \nu\big)^+\Big]\right) \overset{\text{(a)}}{=} \nu - \beta + \nabla_\lambda V^\theta(x^0, \nu). \tag{27}$$

Similar to Section 4.1, equality (a) comes from the fact that the quantity in parenthesis in (27) is $V^\theta(x^0, \nu)$, the value function of the policy θ at state (x^0, ν) in the augmented MDP $\bar{\mathcal{M}}$. Note that the dependence of $V^\theta(x^0, \nu)$ on λ comes from the definition of the cost function $\bar{C}_\lambda$ in $\bar{\mathcal{M}}$. We now derive an expression for $\nabla_\lambda V^\theta(x^0, \nu)$, which in turn will give us an expression for $\nabla_\lambda L(\nu, \theta, \lambda)$.

Lemma 11 *The gradient of $V^\theta(x^0, \nu)$ w.r.t. the Lagrangian parameter λ may be written as*

$$\nabla_\lambda V^\theta(x^0, \nu) = \frac{1}{1-\gamma}\sum_{x,s,a}\pi_\gamma^\theta(x, s, a|x^0, \nu)\frac{1}{(1-\alpha)}\mathbf{1}\{x = x_{Tar}\}(-s)^+. \tag{28}$$

Proof. See Appendix B.1. ■

From Lemma 11 and (27), it is easy to see that $\nu - \beta + \frac{1}{(1-\gamma)(1-\alpha)}\mathbf{1}\{x = x_{\text{Tar}}\}(-s)^+$ is an unbiased estimate of $\nabla_\lambda L(\nu, \theta, \lambda)$. An issue with this estimator is that its value is fixed to $\nu_k - \beta$ all along a trajectory, and only changes at the end to $\nu_k - \beta + \frac{1}{(1-\gamma)(1-\alpha)}(-s_{\text{Tar}})^+$. This may affect the incremental nature of our actor-critic algorithm. To address this issue, Chow and Ghavamzadeh (2014) previously proposed a different approach to estimate the gradients w.r.t. θ and λ which involves another value function approximation to the constraint. However this approach is less desirable in many practical applications as it increases the approximation error and impedes the speed of convergence.

Another important issue is that the above estimator is unbiased only if the samples are generated from the distribution $\pi_\gamma^\theta(\cdot|x^0,\nu)$. If we just follow the policy θ, then we may use $\nu_k - \beta + \frac{\gamma^k}{(1-\alpha)}\mathbf{1}\{x_k = x_{\text{Tar}}\}(-s_k)^+$ as an estimate for $\nabla_\lambda L(\nu,\theta,\lambda)$. Note that this is an issue for all discounted actor-critic algorithms: their (likelihood ratio based) estimate for the gradient is unbiased only if the samples are generated from π_γ^θ, and not when we simply follow the policy. This might also be the reason why, to the best of our knowledge, no rigorous convergence analysis can be found in the literature for (likelihood ratio based) discounted actor-critic algorithms under the sampling distribution.[6]

4.3 Sub-Gradient w.r.t. the VaR Parameter ν

We may rewrite the sub-gradient of our objective function w.r.t. the VaR parameter ν in (8) as

$$\partial_\nu L(\nu,\theta,\lambda) \ni \lambda\Big(1 - \frac{1}{(1-\alpha)}\mathbb{P}\Big(\sum_{k=0}^{\infty}\gamma^k D(x_k,a_k) \geq \nu \mid x_0 = x^0;\theta\Big)\Big). \tag{29}$$

From the definition of the augmented MDP $\bar{\mathcal{M}}$, the probability in (29) may be written as $\mathbb{P}(s_{\text{Tar}} \leq 0 \mid x_0 = x^0, s_0 = \nu;\theta)$, where s_{Tar} is the s part of the state in $\bar{\mathcal{M}}$ when we reach a target state, i.e., $x = x_{\text{Tar}}$ (see Section 4.1). Thus, we may rewrite (29) as

$$\partial_\nu L(\nu,\theta,\lambda) \ni \lambda\Big(1 - \frac{1}{(1-\alpha)}\mathbb{P}\big(s_{\text{Tar}} \leq 0 \mid x_0 = x^0, s_0 = \nu;\theta\big)\Big). \tag{30}$$

From (30), it is easy to see that $\lambda - \lambda\mathbf{1}\{s_{\text{Tar}} \leq 0\}/(1-\alpha)$ is an unbiased estimate of the sub-gradient of $L(\nu,\theta,\lambda)$ w.r.t. ν. An issue with this (unbiased) estimator is that it can only be applied at the end of a trajectory (i.e., when we reach the target state x_{Tar}), and thus, using it prevents us from having a fully incremental algorithm. In fact, this is the estimator that we use in our *semi-trajectory-based* actor-critic algorithm.

One approach to estimate this sub-gradient incrementally is to use the *simultaneous perturbation stochastic approximation* (SPSA) method (Chapter 5 of Bhatnagar et al. (2013)). The idea of SPSA is to estimate the sub-gradient $g(\nu) \in \partial_\nu L(\nu,\theta,\lambda)$ using two values of g at $\nu^- = \nu - \Delta$ and $\nu^+ = \nu + \Delta$, where $\Delta > 0$ is a positive perturbation (see Chapter 5 of Bhatnagar et al. (2013) or Prashanth and Ghavamzadeh (2013) for the detailed description of Δ).[7] In order to see how SPSA can help us to estimate our sub-gradient incrementally, note that

$$\partial_\nu L(\nu,\theta,\lambda) = \lambda + \partial_\nu\left(\mathbb{E}\big[\mathcal{J}^\theta(x^0)\big] + \frac{\lambda}{(1-\alpha)}\mathbb{E}\Big[\big(\mathcal{J}^\theta(x^0) - \nu\big)^+\Big]\right) \overset{\text{(a)}}{=} \lambda + \partial_\nu V^\theta(x^0,\nu). \tag{31}$$

Similar to Sections 4.1–4.3, equality (a) comes from the fact that the quantity in parenthesis in (31) is $V^\theta(x^0,\nu)$, the value function of the policy θ at state (x^0,ν) in the augmented MDP $\bar{\mathcal{M}}$. Since the critic uses a linear approximation for the value function, i.e., $V^\theta(x,s) \approx v^\top\phi(x,s)$, in our actor-critic algorithms (see Section 4.1 and Algorithm 2), the SPSA estimate of the sub-gradient would be of the form $g(\nu) \approx \lambda + v^\top\big[\phi(x^0,\nu^+) - \phi(x^0,\nu^-)\big]/2\Delta$.

6. Note that the discounted actor-critic algorithm with convergence proof in (Bhatnagar, 2010) is based on SPSA.
7. SPSA-based gradient estimate was first proposed in Spall (1992) and has been widely used in various settings, especially those involving a high-dimensional parameter. The SPSA estimate described above is two-sided. It can also be implemented single-sided, where we use the values of the function at ν and ν^+. We refer the readers to Chapter 5 of Bhatnagar et al. (2013) for more details on SPSA and to Prashanth and Ghavamzadeh (2013) for its application to learning in mean-variance risk-sensitive MDPs.

4.4 Convergence of Actor-Critic Methods

In this section, we will prove that the actor-critic algorithms converge to a locally optimal policy for the CVaR-constrained optimization problem. Define

$$\epsilon_\theta(v_k) = \|B_\theta[\Phi v_k] - \Phi v_k\|_\infty$$

as the residual of the value function approximation at step k, induced by policy $\mu(\cdot|\cdot,\cdot;\theta)$. By the triangle inequality and fixed point theorem $B_\theta[V^*] = V^*$, it can be easily seen that $\|V^* - \Phi v_k\|_\infty \leq \epsilon_\theta(v_k) + \|B_\theta[\Phi v_k] - B_\theta[V^*]\|_\infty \leq \epsilon_\theta(v_k) + \gamma\|\Phi v_k - V^*\|_\infty$. The last inequality follows from the contraction property of the Bellman operator. Thus, one concludes that $\|V^* - \Phi v_k\|_\infty \leq \epsilon_\theta(v_k)/(1-\gamma)$. Now, we state the main theorem for the convergence of actor-critic methods.

Theorem 12 *Suppose $\epsilon_{\theta_k}(v_k) \to 0$ and the γ-occupation measure π_γ^θ is used to generate samples of (x_k, s_k, a_k) for any $k \in \{0,1,\ldots\}$. For the SPSA-based algorithms, suppose the feature vector satisfies the technical Assumption 21 (provided in Appendix B.2.2) and suppose the SPSA step-size satisfies the condition $\epsilon_{\theta_k}(v_k) = o(\Delta_k)$, i.e., $\epsilon_{\theta_k}(v_k)/\Delta_k \to 0$. Then under Assumptions 2–4 and 8–9, the sequence of policy updates in Algorithm 2 converges almost surely to a locally optimal policy for the CVaR-constrained optimization problem.*

Details of the proof can be found in Appendix B.2.

5. Extension to Chance-Constrained Optimization of MDPs

In many applications, in particular in engineering (see, for example, Ono et al. (2015)), *chance constraints* are imposed to ensure mission success with high probability. Accordingly, in this section we extend the analysis of CVaR-constrained MDPs to chance-constrained MDPs (i.e., (4)). As for CVaR-constrained MDPs, we employ a Lagrangian relaxation procedure (Chapter 3 of Bertsekas (1999)) to convert a chance-constrained optimization problem into the following unconstrained problem:

$$\max_\lambda \min_{\theta,\alpha}\Big(L(\theta,\lambda) := \mathcal{G}^\theta(x^0) + \lambda\Big(\mathbb{P}\big(\mathcal{J}^\theta(x^0) \geq \alpha\big) - \beta\Big)\Big), \tag{32}$$

where λ is the Lagrange multiplier. Recall Assumption 4 which assumed strict feasibility, i.e., there exists a transient policy $\mu(\cdot|x;\theta)$ such that $\mathbb{P}\big(\mathcal{J}^\theta(x^0) \geq \alpha\big) < \beta$. This is needed to guarantee the existence of a local saddle point.

5.1 Policy Gradient Method

In this section we propose a policy gradient method for chance-constrained MDPs (similar to Algorithm 1). Since we do not need to estimate the ν-parameter in chance-constrained optimization, the corresponding policy gradient algorithm can be simplified and at each inner loop of Algorithm 1 we only perform the following updates at the end of each trajectory:

θ **Update:** $\theta_{k+1} = \Gamma_\Theta\Big[\theta_k - \frac{\zeta_2(k)}{N}\Big(\sum_{j=1}^{N}\nabla_\theta \log\mathbb{P}(\xi_{j,k})\mathcal{G}(\xi_{j,k}) + \lambda_k\nabla_\theta\log\mathbb{P}(\xi_{j,k})\mathbf{1}\big\{\mathcal{J}(\xi_{j,k}) \geq \alpha\big\}\Big)\Big]$

λ **Update:** $\lambda_{k+1} = \Gamma_\Lambda\Big[\lambda_k + \zeta_1(k)\Big(-\beta + \frac{1}{N}\sum_{j=1}^{N}\mathbf{1}\big\{\mathcal{J}(\xi_{j,k}) \geq \alpha\big\}\Big)\Big]$

Considering the multi-time-scale step-size rules in Assumption 6, the θ update is on the fast time-scale $\{\zeta_2(k)\}$ and the Lagrange multiplier λ update is on the slow time-scale $\{\zeta_1(k)\}$. This results in a two time-scale stochastic approximation algorithm. In the following theorem, we prove that our policy gradient algorithm converges to a locally optimal policy for the chance-constrained problem.

Theorem 13 *Under Assumptions 2–6, the sequence of policy updates in Algorithm 1 converges to a locally optimal policy θ^* for the chance-constrained optimization problem almost surely.*

Proof. [Sketch] By taking the gradient of $L(\theta,\lambda)$ w.r.t. θ, we have

$$\nabla_\theta L(\theta,\lambda) = \nabla_\theta \mathcal{G}^\theta(x^0) + \lambda \nabla_\theta \mathbb{P}\big(\mathcal{J}^\theta(x^0) \geq \alpha\big) = \sum_\xi \nabla_\theta \mathbb{P}_\theta(\xi)\mathcal{G}(\xi) + \lambda \sum_\xi \nabla_\theta \mathbb{P}_\theta(\xi)\mathbf{1}\{\mathcal{J}(\xi) \geq \alpha\}.$$

On the other hand, the gradient of $L(\theta,\lambda)$ w.r.t. λ is given by

$$\nabla_\lambda L(\theta,\lambda) = \mathbb{P}\big(\mathcal{J}^\theta(x^0) \geq \alpha\big) - \beta.$$

One can easily verify that the θ and λ updates are therefore unbiased estimates of $\nabla_\theta L(\theta,\lambda)$ and $\nabla_\lambda L(\theta,\lambda)$, respectively. Then the rest of the proof follows analogously from the convergence proof of Algorithm 1 in steps 2 and 3 of Theorem 7. ■

5.2 Actor-Critic Method

In this section, we present an actor-critic algorithm for the chance-constrained optimization. Given the original MDP $\mathcal{M} = (\mathcal{X}, \mathcal{A}, C, D, P, P_0)$ and parameter λ, we define the augmented MDP $\bar{\mathcal{M}} = (\bar{\mathcal{X}}, \bar{\mathcal{A}}, \bar{C}_\lambda, \bar{P}, \bar{P}_0)$ as in the CVaR counterpart, except that $\bar{P}_0(x,s) = P_0(x)\mathbf{1}\{s = \alpha\}$ and

$$\bar{C}_\lambda(x,s,a) = \begin{cases} \lambda \mathbf{1}\{s \leq 0\} & \text{if } x = x_{\text{Tar}}, \\ C(x,a) & \text{otherwise.} \end{cases}$$

Thus, the total cost of a trajectory can be written as

$$\sum_{k=0}^{T} \bar{C}_\lambda(x_k, s_k, a_k) \mid x_0 = x, s_0 = \beta,\ \mu = \mathcal{G}^\theta(x) + \lambda \mathbb{P}(\mathcal{J}^\theta(x) \geq \beta). \tag{33}$$

Unlike the actor-critic algorithms for CVaR-constrained optimization, here the value function approximation parameter v, policy θ, and Lagrange multiplier estimate λ are updated episodically, i.e., after each episode ends by time T when $(x_k, s_k) = (x_{\text{Tar}}, s_{\text{Tar}})$[8], as follows:

$$\textbf{Critic Update:} \quad v_{k+1} = v_k + \zeta_3(k) \sum_{h=0}^{T} \phi(x_h, s_h)\delta_h(v_k) \tag{34}$$

$$\textbf{Actor Updates:} \quad \theta_{k+1} = \Gamma_\Theta\Big(\theta_k - \zeta_2(k) \sum_{h=0}^{T} \nabla_\theta \log \mu(a_h|x_h, s_h; \theta)|_{\theta=\theta_k} \cdot \delta_h(v_k)\Big) \tag{35}$$

$$\lambda_{k+1} = \Gamma_\Lambda\Big(\lambda_k + \zeta_1(k)\big(-\beta + \mathbf{1}\{s_{\text{Tar}} \leq 0\}\big)\Big) \tag{36}$$

From analogous analysis as for the CVaR actor-critic method, the following theorem shows that the critic update v_k converges almost surely to v^*.

8. Note that s_{Tar} is the state of s_t when x_t hits the (recurrent) target state x_{Tar}.

Theorem 14 *Let $v^* \in \arg\min_v \|B_\theta[\Phi v] - \Phi v\|^2_{d^\theta}$ be a minimizer of the Bellman residual, where the undiscounted Bellman operator at every $(x,s) \in \bar{\mathcal{X}}'$ is given by*

$$B_\theta[V](x,s) = \sum_{a\in\mathcal{A}} \mu(a|x,s;\theta)\Big\{\bar{C}_\lambda(x,s,a) + \sum_{(x',s')\in\bar{\mathcal{X}}'} \bar{P}(x',s'|x,s,a)V(x',s')\Big\}$$

and $\tilde{V}^*(x,s) = \phi^\top(x,s)v^*$ is the projected Bellman fixed point of $V^\theta(x,s)$, i.e., $\tilde{V}^*(x,s) = \Pi B_\theta[\tilde{V}^*](x,s)$ for $(x,s) \in \bar{\mathcal{X}}'$. Then under Assumptions 8–9, the v-update in the actor-critic algorithm converges to v^* almost surely.

Proof. [Sketch] The proof of this theorem follows the same steps as those in the proof of Theorem 10, except replacing the γ-occupation measure d^θ_γ with the occupation measure d^θ (the total visiting probability). Similar analysis can also be found in the proof of Theorem 10 in Tamar and Mannor (2013). Under Assumption 2, the occupation measure of any transient states $x \in \mathcal{X}'$ (starting at an arbitrary initial transient state $x_0 \in \mathcal{X}'$) can be written as $d^\mu(x|x^0) = \sum_{t=0}^{T_{\mu,x}} \mathbb{P}(x_t = x|x^0;\mu)$ when $\gamma = 1$. This further implies the total visiting probabilities are bounded as follows: $d^\mu(x|x^0) \leq T_{\mu,x}$ and $\pi^\mu(x,a|x^0) \leq T_{\mu,x}$ for any $x, x_0 \in \mathcal{X}'$. Therefore, when the sequence of states $\{(x_h, s_h)\}_{h=0}^T$ is sampled by the h-step transition distribution $\mathbb{P}(x_h, s_h \mid x^0, s^0, \theta)$, $\forall h \leq T$, the unbiased estimators of

$$A := \sum_{(y,s')\in\bar{\mathcal{X}}', a'\in\mathcal{A}} \pi^\theta(y,s',a'|x,s)\phi(y,s')\Big(\phi^\top(y,s') - \sum_{(z,s'')\in\bar{\mathcal{X}}'} \bar{P}(z,s''|y,s',a)\phi^\top(z,s'')\Big)$$

and

$$b := \sum_{(y,s')\in\bar{\mathcal{X}}', a'\in\mathcal{A}} \pi^\theta(y,s',a'|x,s)\phi(y,s')\bar{C}_\lambda(y,s',a')$$

are given by $\sum_{h=0}^T \phi(x_h,s_h)(\phi^\top(x_h,s_h) - \phi^\top(x_{h+1},s_{h+1}))$ and $\sum_{h=0}^T \phi(x_h,s_h)\bar{C}_\lambda(x_h,s_h,a_h)$, respectively. Note that in this theorem, we directly use the results from Theorem 7.1 in Bertsekas (1995) to show that every eigenvalue of matrix A has positive real part, instead of using the technical result in Lemma 20. ■

Recall that $\epsilon_\theta(v_k) = \|B_\theta[\Phi v_k] - \Phi v_k\|_\infty$ is the residual of the value function approximation at step k induced by policy $\mu(\cdot|\cdot,\cdot;\theta)$. By the triangle inequality and fixed-point theorem of stochastic stopping problems, i.e., $B_\theta[V^*] = V^*$ from Theorem 3.1 in Bertsekas (1995), it can be easily seen that $\|V^* - \Phi v_k\|_\infty \leq \epsilon_\theta(v_k) + \|B_\theta[\Phi v_k] - B_\theta[V^*]\|_\infty \leq \epsilon_\theta(v_k) + \kappa\|\Phi v_k - V^*\|_\infty$ for some $\kappa \in (0,1)$. Similar to the actor-critic algorithm for CVaR-constrained optimization, the last inequality also follows from the contraction mapping property of B_θ from Theorem 3.2 in Bertsekas (1995). Now, we state the main theorem for the convergence of the actor-critic method.

Theorem 15 *Under Assumptions 2–9, if $\epsilon_{\theta_k}(v_k) \to 0$, then the sequence of policy updates converges almost surely to a locally optimal policy θ^* for the chance-constrained optimization problem.*

Proof. [Sketch] From Theorem 14, the critic update converges to the minimizer of the Bellman residual. Since the critic update converges on the fastest scale, as in the proof of Theorem 12, one can replace v_k by $v^*(\theta_k)$ in the convergence proof of the actor update. Furthermore, by sampling the sequence of states $\{(x_h, s_h)\}_{h=0}^T$ with the h-step transition distribution $\mathbb{P}(x_h, s_h \mid x^0, s^0, \theta)$, $\forall h \leq T$, the unbiased estimator of the gradient of the linear approximation to the Lagrangian function is given by

$$\nabla_\theta \tilde{L}^v(\theta,\lambda) := \sum_{(x,s)\in\bar{\mathcal{X}}', a\in\mathcal{A}} \pi^\theta(x,s,a|x_0 = x^0, s_0 = \nu)\nabla_\theta \log \mu(a|x,s;\theta)\tilde{A}^{\theta,v}(x,s,a),$$

where $\tilde{Q}^{\theta,v}(x,s,a) - v^\top \phi(x,s)$ is given by $\sum_{h=0}^{T} \nabla_\theta \log \mu(a_h|x_h,s_h;\theta)|_{\theta=\theta_k} \cdot \delta_h(v^*)$ and the unbiased estimator of $\nabla_\lambda L(\theta,\lambda) = -\beta + \mathbb{P}(s_{\text{Tar}} \leq 0)$ is given by $-\beta + \mathbf{1}\{s_{\text{Tar}} \leq 0\}$. Analogous to equation (77) in the proof of Theorem 24, by convexity of quadratic functions, we have for any value function approximation v,

$$\sum_{(y,s')\in\bar{\mathcal{X}}',a'\in\mathcal{A}} \pi^\theta(y,s',a'|x,s)(A_\theta(y,s',a') - \tilde{A}^v_\theta(y,s',a')) \leq 2T\frac{\epsilon_\theta(v)}{1-\kappa},$$

which further implies that $\nabla_\theta L(\theta,\lambda) - \nabla_\theta \tilde{L}^v(\theta,\lambda) \to 0$ when $\epsilon_\theta(v) \to 0$ at $v = v^*(\theta_k)$. The rest of the proof follows identical arguments as in steps 3 to 5 of the proof of Theorem 12. ■

6. Examples

In this section we illustrate the effectiveness of our risk-constrained policy gradient and actor-critic algorithms by testing them on an American option stopping problem and on a long-term personalized advertisement-recommendation (ad-recommendation) problem.

6.1 The Optimal Stopping Problem

We consider an optimal stopping problem of purchasing certain types of goods, in which the state at each time step $k \leq T$ consists of a purchase cost c_k and time k, i.e., $x = (c_k, k)$, where T is the deterministic upper bound of the random stopping time. The purchase cost sequence $\{c_k\}_{k=0}^T$ is randomly generated by a Markov chain with two modes. Specifically, due to future market uncertainties, at time k the random purchase cost at the next time step c_{k+1} either grows by a factor $f_u > 1$, i.e., $c_{k+1} = f_u c_k$, with probability p, or drops by a factor $f_d < 1$, i.e., $c_{k+1} = f_d c_k$, with probability $1-p$. Here f_u and f_d are constants that represent the rates of appreciation (due to anticipated shortage of supplies from vendors) and depreciation (due to reduction of demands in the market) respectively. The agent (buyer) should decide either to accept the present cost ($u_k = 1$) or wait ($u_k = 0$). If he/she accepts the cost or when the system terminates at time $k = T$, the purchase cost is set at $\max(K, c_k)$, where K is the maximum cost threshold. Otherwise, to account for a steady rate of inflation, at each time step the buyer receives an extra cost of p_h that is independent to the purchase cost. Moreover, there is a discount factor $\gamma \in (0,1)$ to account for the increase in the buyer's affordability. Note that if we change cost to reward and minimization to maximization, this is exactly the American option pricing problem, a standard testbed to evaluate risk-sensitive algorithms (e.g., see Tamar et al. (2012)). Since the state space size n is exponential in T, finding an exact solution via dynamic programming (DP) quickly becomes infeasible, and thus the problem requires approximation and sampling techniques.

The optimal stopping problem can be reformulated as follows

$$\min_\theta \mathbb{E}\left[\mathcal{G}^\theta(x^0)\right] \qquad \text{subject to} \qquad \text{CVaR}_\alpha\left(\mathcal{G}^\theta(x^0)\right) \leq \beta \quad \text{or} \quad \mathbb{P}\left(\mathcal{G}^\theta(x^0) \geq \beta\right) \leq 1-\alpha^9, \tag{37}$$

where the discounted cost function is given by

$$\mathcal{G}^\theta(x) = \sum_{k=0}^{T} \gamma^k \left(\mathbf{1}\{u_k = 1\}\max(K, c_k) + \mathbf{1}\{u_k = 0\}p_h\right) \mid x_0 = x,\ \mu.$$

9. To ensure that the notation is consistent between the CVaR and chance constraints, in the chance constraint definition the confidence level is denoted by α and the tolerance threshold of $\mathcal{G}^\theta(x^0)$ is denoted by β.

We set the parameters of the MDP as follows: $x_0 = [1;0]$, $p_h = 0.1$, $T = 20$, $K = 5$, $\gamma = 0.95$, $f_u = 2$, $f_d = 0.5$, and $p = 0.65$. The confidence level and constraint threshold are given by $\alpha = 0.95$ and $\beta = 3$. The number of sample trajectories N is set to $500,000$ and the parameter bounds are $\lambda_{\max} = 5,000$ and $\Theta = [-20,20]^{\kappa_1}$, where the dimension of the basis functions is $\kappa_1 = 1024$. We implement the standard Gaussian radial basis functions (RBFs) as feature functions and search over the class of Boltzmann policies $\left\{\theta : \theta = \{\theta_{x,a}\}_{x\in\mathcal{X},a\in\mathcal{A}},\ \mu_\theta(a|x) = \frac{\exp(\theta_{x,a}^\top x)}{\sum_{a\in\mathcal{A}}\exp(\theta_{x,a}^\top x)}\right\}$.

We consider the following trajectory-based algorithms:

1. **PG:** This is a policy gradient algorithm that minimizes the expected discounted cost function without considering any risk criteria.

2. **PG-CVaR/PG-CC:** These are the CVaR/chance-constrained simulated trajectory-based policy gradient algorithms given in Section 3.

The experiments for each algorithm comprise the following two phases:

1. **Tuning phase:** We run the algorithm and update the policy until (ν,θ,λ) converges.

2. **Converged run:** Having obtained a converged policy θ^* in the tuning phase, in the converged run phase, we perform a Monte Carlo simulation of $10,000$ trajectories and report the results as averages over these trials.

We also consider the following incremental algorithms:

1. **AC:** This is an actor-critic algorithm that minimizes the expected discounted cost function without considering any risk criteria. This is similar to Algorithm 1 in Bhatnagar (2010).

2. **AC-CVaR/AC-CC:** These are the CVaR/chance-constrained semi-trajectory actor-critic algorithms given in Section 4.

3. **AC-CVaR-SPSA:** This is the CVaR-constrained SPSA actor-critic algorithm given in Section 4.

Similar to the trajectory-based algorithms, we use RBF features for $[x;s]$ and consider the family of augmented state Boltzmann policies. Similarly, the experiments comprise two phases: 1) the tuning phase, where the set of parameters (v,ν,θ,λ) is obtained after the algorithm converges, and 2) the converged run, where the policy is simulated with $10,000$ trajectories.

We compare the performance of PG-CVaR and PG-CC (given in Algorithm 1), and AC-CVaR-SPSA, AC-CVaR, and AC-CC (given in Algorithm 2), with PG and AC, their risk-neutral counterparts. Figures 1 and 2 show the distribution of the discounted cumulative cost $\mathcal{G}^\theta(x^0)$ for the policy θ learned by each of these algorithms. The results indicate that the risk-constrained algorithms yield a higher expected cost, but less worst-case variability, compared to the risk-neutral methods. More precisely, the cost distributions of the risk-constrained algorithms have lower right-tail (worst-case) distribution than their risk-neutral counterparts. Table 1 summarizes the performance of these algorithms. The numbers reiterate what we concluded from Figures 1 and 2.

Notice that while the risk averse policy satisfies the CVaR constraint, it is not tight (i.e., the constraint is not matched). In fact this is a problem of local optimality, and other experiments in the literature (for example see the numerical results in Prashanth and Ghavamzadeh (2013) and in

Bhatnagar and Lakshmanan (2012)) have the same problem of producing solutions which obey the constraints but not tightly. However, since both the expectation and the CVaR risk metric are sub-additive and convex, one can always construct a policy that is a linear combination of the risk neutral optimal policy and the risk averse policy such that it matches the constraint threshold and has a lower cost compared to the risk averse policy.

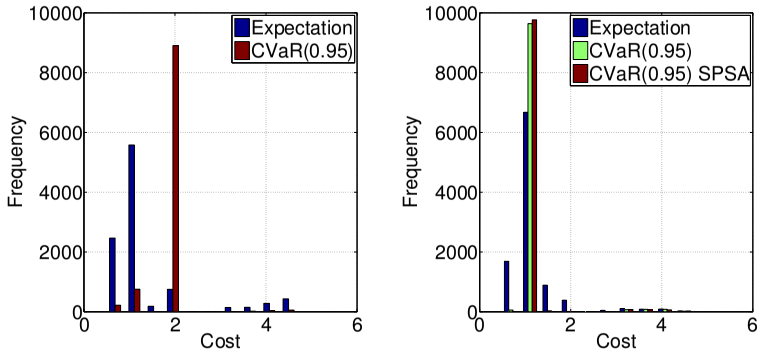


Figure 1: Cost distributions for the policies learned by the CVaR-constrained and risk-neutral policy gradient and actor-critic algorithms. The left figure corresponds to the PG methods and the right figure corresponds to the AC algorithms.

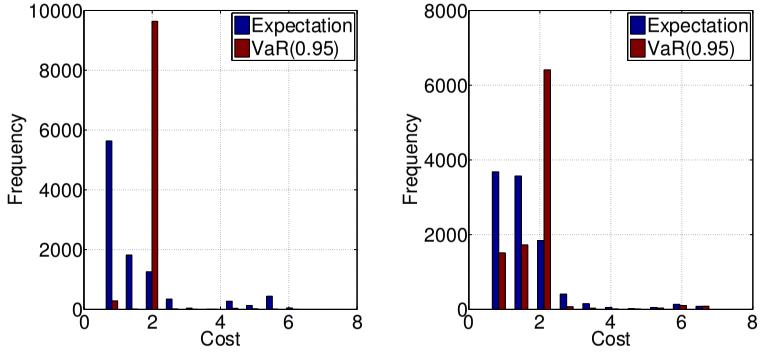


Figure 2: Cost distributions for the policies learned by the chance-constrained and risk-neutral policy gradient and actor-critic algorithms. The left figure corresponds to the PG methods and the right figure corresponds to the AC algorithms.

	$\mathbb{E}(\mathcal{G}^\theta(x^0))$	$\sigma(\mathcal{G}^\theta(x^0))$	$\text{CVaR}(\mathcal{G}^\theta(x^0))$	$\text{VaR}(\mathcal{G}^\theta(x^0))$
PG	1.177	1.065	4.464	4.005
PG-CVaR	1.997	0.060	2.000	2.000
PG-CC	1.994	0.121	2.058	2.000
AC	1.113	0.607	3.331	3.220
AC-CVaR-SPSA	1.326	0.322	2.145	1.283
AC-CVaR	1.343	0.346	2.208	1.290
AC-CC	1.817	0.753	4.006	2.300

Table 1: Performance comparison of the policies learned by the risk-constrained and risk-neutral algorithms. In this table $\sigma(\mathcal{G}^\theta(x^0))$ stands for the standard deviation of the total cost.

6.2 A Personalized Ad-Recommendation System

Many companies such as banks and retailers use user-specific targeting of advertisements to attract more customers and increase their revenue. When a user requests a webpage that contains a box for an advertisement, the system should decide which advertisement (among those in the current campaign) to show to this particular user based on a vector containing all her features, often collected by a cookie. Our goal here is to generate a strategy that for each user of the website selects an ad that when it is presented to her has the highest probability to be clicked on. These days, almost all the industrial personalized ad recommendation systems use supervised learning or contextual bandits algorithms. These methods are based on the i.i.d. assumption of the visits (to the website) and do not discriminate between a visit and a visitor, i.e., each visit is considered as a new visitor that has been sampled i.i.d. from the population of the visitors. As a result, these algorithms are myopic and do not try to optimize for the long-term performance. Despite their success, these methods seem to be insufficient as users establish longer-term relationship with the websites they visit, i.e., the ad recommendation systems should deal with more and more returning visitors. The increase in returning visitors violates (more) the main assumption underlying the supervised learning and bandit algorithms, i.e., there is no difference between a visit and a visitor, and thus, shows the need for a new class of solutions.

The reinforcement learning (RL) algorithms that have been designed to optimize the long-term performance of the system (expected sum of rewards/costs) seem to be suitable candidates for ad recommendation systems (Shani et al., 2002). The nature of these algorithms allows them to take into account all the available knowledge about the user at the current visit, and then selects an offer to maximize the total number of times she will click over multiple visits, also known as the user's life-time value (LTV). Unlike myopic approaches, RL algorithms differentiate between a visit and a visitor, and consider all the visits of a user (in chronological order) as a trajectory generated by her. In this approach, while the visitors are i.i.d. samples from the population of the users, their visits are not. This long-term approach to the ad recommendation problem allows us to make decisions that are not usually possible with myopic techniques, such as to propose an offer to a user that might be a loss to the company in the short term, but has the effect that makes the user engaged with the website/company and brings her back to spend more money in the future.

For our second case study, we use an Adobe personalized ad-recommendation (Theocharous and Hallak, 2013) simulator that has been trained based on real data captured with permission from the website of a Fortune 50 company that receives hundreds of visitors per day. The simulator produces a vector of 31 real-valued features that provide a compressed representation of all of the

available information about a user. The advertisements are clustered into four high-level classes that the agent must select between. After the agent selects an advertisement, the user either clicks (reward of $+1$) or does not click (reward of 0) and the feature vector describing the user is updated. In this case, we test our algorithm by maximizing the customers' life-time value in 15 time steps subject to a bounded tail risk.

Instead of using the cost-minimization framework from the problem statement in Section 2.2, by defining the return random variable (under a fixed policy θ) $\mathcal{R}^\theta(x^0)$ as the (discounted) total number of clicks along a user's trajectory, here we formulate the personalized ad-recommendation problem as a return maximization problem where the tail risk corresponds to the worst case return distribution:

$$\max_\theta \mathbb{E}\left[\mathcal{R}^\theta(x^0)\right] \qquad \text{subject to} \qquad \text{CVaR}_{1-\alpha}\big(-\mathcal{R}^\theta(x^0)\big) \leq \beta. \tag{38}$$

We set the parameters of the MDP as $T = 15$ and $\gamma = 0.98$, the confidence level and constraint threshold as $\alpha = 0.05$ and $\beta = -0.12$, the number of sample trajectories N to $1,000,000$, and the parameter bounds as $\lambda_{\max} = 5,000$ and $\Theta = [-60, 60]^{\kappa_1}$, where the dimension of the basis functions is $\kappa_1 = 4096$. Similar to the optimal stopping problem, we implement both the trajectory based algorithm (PG, PG-CVaR) and the actor-critic algorithms (AC, AC-CVaR) for risk-neutral and risk sensitive optimal control. Here we used the 3rd order Fourier basis with cross-products in Konidaris et al. (2011) as features and search over the family of Boltzmann policies. We compared the performance of PG-CVaR and AC-CVaR, our risk-constrained policy gradient (Algorithm 1) and actor-critic (Algorithms 2) algorithms, with their risk-neutral counterparts (PG and AC). Figure 3 shows the distribution of the discounted cumulative return $\mathcal{R}^\theta(x^0)$ for the policy θ learned by each of these algorithms. The results indicate that the risk-constrained algorithms yield a lower expected reward, but have higher left tail (worst-case) reward distributions. Table 2 summarizes the findings of this experiment.

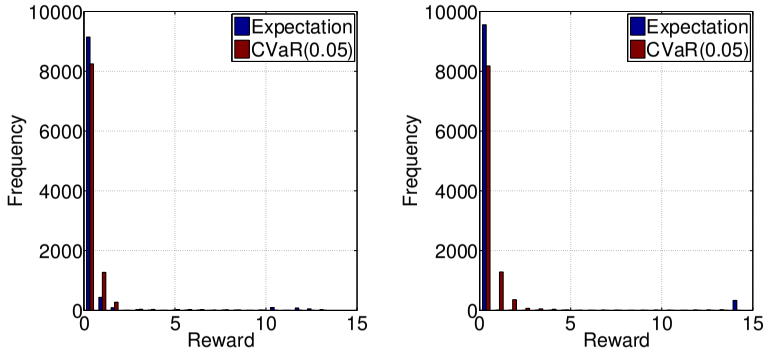


Figure 3: Reward distributions for the policies learned by the CVaR-constrained and risk-neutral policy gradient and actor-critic algorithms. The left figure corresponds to the PG methods and the right figure corresponds to the AC algorithms.

	$\mathbb{E}\big(\mathcal{R}^\theta(x^0)\big)$	$\sigma\big(\mathcal{R}^\theta(x^0)\big)$	$\text{CVaR}\big(\mathcal{R}^\theta(x^0)\big)$	$\text{VaR}\big(\mathcal{R}^\theta(x^0)\big)$
PG	0.396	1.898	0.037	1.000
PG-CVaR	0.287	0.914	0.126	1.795
AC	0.581	2.778	0	0
AC-CVaR	0.253	0.634	0.137	1.890

Table 2: Performance comparison of the policies learned by the CVaR-constrained and risk-neutral algorithms. In this table $\sigma\big(\mathcal{R}^\theta(x^0)\big)$ stands for the standard deviation of the total reward.

7. Conclusions and Future Work

We proposed novel policy gradient and actor-critic algorithms for CVaR-constrained and chance-constrained optimization in MDPs, and proved their convergence. Using an optimal stopping problem and a personalized ad-recommendation problem, we showed that our algorithms resulted in policies whose cost distributions have lower right-tail compared to their risk-neutral counterparts. This is important for a risk-averse decision-maker, especially if the right-tail contains catastrophic costs. Future work includes: 1) Providing convergence proofs for our AC algorithms when the samples are generated by following the policy and not from its discounted occupation measure , 2) Using importance sampling methods (Bardou et al., 2009; Tamar et al., 2015) to improve gradient estimates in the right-tail of the cost distribution (worst-case events that are observed with low probability), and 3) Applying the algorithms presented in this paper to a variety of applications ranging from operations research to robotics and finance.

Acknowledgments

We would like to thank Csaba Szepesvari for his comments that helped us with the derivation of the algorithms, Georgios Theocharous for sharing his ad-recommendation simulator with us, and Philip Thomas for helping us with the experiments with the simulator. We would also like to thank the reviewers for their very helpful comments and suggestions, which helped us to significantly improve the paper. Y-L. Chow is partially supported by The Croucher Foundation doctoral scholarship. L. Janson was partially supported by NIH training grant T32GM096982. M. Pavone was partially supported by the Office of Naval Research, Science of Autonomy Program, under Contract N00014-15-1-2673.

Appendix A. Convergence of Policy Gradient Methods

A.1 Computing the Gradients

i) $\nabla_\theta L(\nu, \theta, \lambda)$: Gradient of $L(\nu, \theta, \lambda)$ w.r.t. θ By expanding the expectations in the definition of the objective function $L(\nu, \theta, \lambda)$ in (5), we obtain

$$L(\nu, \theta, \lambda) = \sum_\xi \mathbb{P}_\theta(\xi)\mathcal{G}(\xi) + \lambda\nu + \frac{\lambda}{1-\alpha}\sum_\xi \mathbb{P}_\theta(\xi)\big(\mathcal{J}(\xi) - \nu\big)^+ - \lambda\beta.$$

By taking the gradient with respect to θ, we have

$$\nabla_\theta L(\nu, \theta, \lambda) = \sum_\xi \nabla_\theta \mathbb{P}_\theta(\xi)\mathcal{G}(\xi) + \frac{\lambda}{1-\alpha}\sum_\xi \nabla_\theta \mathbb{P}_\theta(\xi)\big(\mathcal{J}(\xi) - \nu\big)^+.$$

This gradient can be rewritten as

$$\nabla_\theta L(\nu,\theta,\lambda) = \sum_{\xi:\mathbb{P}_\theta(\xi)\neq 0} \mathbb{P}_\theta(\xi)\cdot\nabla_\theta \log \mathbb{P}_\theta(\xi)\left(\mathcal{G}(\xi)+\frac{\lambda}{1-\alpha}\big(\mathcal{J}(\xi)-\nu\big)\mathbf{1}\big\{\mathcal{J}(\xi)\geq\nu\big\}\right), \quad (39)$$

where in the case of $\mathbb{P}_\theta(\xi)\neq 0$, the term $\nabla_\theta \log \mathbb{P}_\theta(\xi)$ is given by:

$$\begin{aligned}\nabla_\theta \log \mathbb{P}_\theta(\xi) =& \nabla_\theta\left\{\sum_{k=0}^{T-1}\log P(x_{k+1}|x_k,a_k)+\log\mu(a_k|x_k;\theta)+\log\mathbf{1}\{x_0=x^0\}\right\}\\ =&\sum_{k=0}^{T-1}\nabla_\theta\log\mu(a_k|x_k;\theta)\\ =&\sum_{k=0}^{T-1}\frac{1}{\mu(a_k|x_k;\theta)}\nabla_\theta\mu(a_k|x_k;\theta).\end{aligned}$$

ii) $\partial_\nu L(\nu,\theta,\lambda)$: Sub-differential of $L(\nu,\theta,\lambda)$ w.r.t. ν From the definition of $L(\nu,\theta,\lambda)$, we can easily see that $L(\nu,\theta,\lambda)$ is a convex function in ν for any fixed $\theta\in\Theta$. Note that for every fixed ν and any ν', we have

$$\big(\mathcal{J}(\xi)-\nu'\big)^+-\big(\mathcal{J}(\xi)-\nu\big)^+\geq g\cdot(\nu'-\nu),$$

where g is any element in the set of sub-derivatives:

$$g\in\partial_\nu\big(\mathcal{J}(\xi)-\nu\big)^+ := \begin{cases}-1 & \text{if } \nu<\mathcal{J}(\xi),\\ -q:q\in[0,1] & \text{if } \nu=\mathcal{J}(\xi),\\ 0 & \text{otherwise.}\end{cases}$$

Since $L(\nu,\theta,\lambda)$ is finite-valued for any $\nu\in\mathbb{R}$, by the additive rule of sub-derivatives, we have

$$\partial_\nu L(\nu,\theta,\lambda)=\left\{-\frac{\lambda}{1-\alpha}\sum_\xi\mathbb{P}_\theta(\xi)\mathbf{1}\big\{\mathcal{J}(\xi)>\nu\big\}-\frac{\lambda q}{1-\alpha}\sum_\xi\mathbb{P}_\theta(\xi)\mathbf{1}\big\{\mathcal{J}(\xi)=\nu\big\}+\lambda\mid q\in[0,1]\right\}. \quad (40)$$

In particular for $q=1$, we may write the sub-gradient of $L(\nu,\theta,\lambda)$ w.r.t. ν as

$$\partial_\nu L(\nu,\theta,\lambda)|_{q=0}=\lambda-\frac{\lambda}{1-\alpha}\sum_\xi\mathbb{P}_\theta(\xi)\cdot\mathbf{1}\big\{\mathcal{J}(\xi)\geq\nu\big\}$$

or

$$\lambda-\frac{\lambda}{1-\alpha}\sum_\xi\mathbb{P}_\theta(\xi)\cdot\mathbf{1}\big\{\mathcal{J}(\xi)\geq\nu\big\}\in\partial_\nu L(\nu,\theta,\lambda).$$

iii) $\nabla_\lambda L(\nu,\theta,\lambda)$: Gradient of $L(\nu,\theta,\lambda)$ w.r.t. λ Since $L(\nu,\theta,\lambda)$ is a linear function in λ, one can express the gradient of $L(\nu,\theta,\lambda)$ w.r.t. λ as follows:

$$\nabla_\lambda L(\nu,\theta,\lambda)=\nu-\beta+\frac{1}{1-\alpha}\sum_\xi\mathbb{P}_\theta(\xi)\cdot\big(\mathcal{J}(\xi)-\nu\big)\mathbf{1}\big\{\mathcal{J}(\xi)\geq\nu\big\}. \quad (41)$$

A.2 Proof of Convergence of the Policy Gradient Algorithm

In this section, we prove the convergence of the policy gradient algorithm (Algorithm 1). Before going through the details of the convergence proof, a high level overview of the proof technique is given as follows.

1. First, by convergence properties of multi-time scale discrete stochastic approximation algorithms, we show that each update $(\nu_k,\theta_k,\lambda_k)$ converges almost surely to a stationary point $(\nu^*,\theta^*,\lambda^*)$ of the corresponding continuous time system. In particular, by adopting the step-size rules defined in Assumption 6, we show that the convergence rate of ν is fastest, followed by the convergence rate of θ, while the convergence rate of λ is the slowest among the set of parameters.

2. By using Lyapunov analysis, we show that the continuous time system is locally asymptotically stable at the stationary point $(\nu^*,\theta^*,\lambda^*)$.

3. Since the Lyapunov function used in the above analysis is the Lagrangian function $L(\nu,\theta,\lambda)$, we conclude that the stationary point $(\nu^*,\theta^*,\lambda^*)$ is a local saddle point. Finally by the local saddle point theorem, we deduce that θ^* is a locally optimal solution for the CVaR-constrained MDP problem.

This convergence proof procedure is standard for stochastic approximation algorithms, see (Bhatnagar et al., 2009; Bhatnagar and Lakshmanan, 2012) for further references.

Since ν converges on the faster timescale than θ and λ, the ν-update can be rewritten by assuming (θ,λ) as invariant quantities, i.e.,

$$\nu_{k+1}=\Gamma_{\mathcal{N}}\left[\nu_k-\zeta_3(k)\left(\lambda-\frac{\lambda}{(1-\alpha)N}\sum_{j=1}^N\mathbf{1}\big\{\mathcal{J}(\xi_{j,k})\geq\nu_k\big\}\right)\right]. \quad (42)$$

Consider the continuous time dynamics of ν defined using differential inclusion

$$\dot\nu\in\Upsilon_\nu\left[-g(\nu)\right],\qquad \forall g(\nu)\in\partial_\nu L(\nu,\theta,\lambda), \quad (43)$$

where

$$\Upsilon_\nu[K(\nu)]:=\lim_{0<\eta\to 0}\frac{\Gamma_{\mathcal{N}}(\nu+\eta K(\nu))-\Gamma_{\mathcal{N}}(\nu)}{\eta}.$$

Here $\Upsilon_\nu[K(\nu)]$ is the left directional derivative of the function $\Gamma_{\mathcal{N}}(\nu)$ in the direction of $K(\nu)$. By using the left directional derivative $\Upsilon_\nu\left[-g(\nu)\right]$ in the sub-gradient descent algorithm for ν, the gradient will point in the descent direction along the boundary of ν whenever the ν-update hits its boundary.

Furthermore, since ν converges on a faster timescale than θ, and λ is on the slowest time-scale, the θ-update can be rewritten using the converged $\nu^*(\theta)$, assuming λ as an invariant quantity, i.e.,

$$\begin{aligned}\theta_{k+1}=&\Gamma_\Theta\Bigg[\theta_k-\zeta_2(k)\Bigg(\frac{1}{N}\sum_{j=1}^N\nabla_\theta\log\mathbb{P}_\theta(\xi_{j,k})|_{\theta=\theta_k}\mathcal{G}(\xi_{j,k})\\ &+\frac{\lambda}{(1-\alpha)N}\sum_{j=1}^N\nabla_\theta\log\mathbb{P}_\theta(\xi_{j,k})|_{\theta=\theta_k}\big(\mathcal{J}(\xi_{j,k})-\nu\big)\mathbf{1}\big\{\mathcal{J}(\xi_{j,k})\geq\nu^*(\theta_k)\big\}\Bigg)\Bigg].\end{aligned}$$

Consider the continuous time dynamics of $\theta \in \Theta$:

$$\dot{\theta} = \Upsilon_\theta \left[-\nabla_\theta L(\nu, \theta, \lambda)\right]|_{\nu=\nu^*(\theta)}, \tag{44}$$

where

$$\Upsilon_\theta[K(\theta)] := \lim_{0<\eta\to 0} \frac{\Gamma_\Theta(\theta + \eta K(\theta)) - \Gamma_\Theta(\theta)}{\eta}.$$

Similar to the analysis of ν, $\Upsilon_\theta[K(\theta)]$ is the left directional derivative of the function $\Gamma_\Theta(\theta)$ in the direction of $K(\theta)$. By using the left directional derivative $\Upsilon_\theta\left[-\nabla_\theta L(\nu, \theta, \lambda)\right]$ in the gradient descent algorithm for θ, the gradient will point in the descent direction along the boundary of Θ whenever the θ-update hits its boundary.

Finally, since the λ-update converges in the slowest time-scale, the λ-update can be rewritten using the converged $\theta^*(\lambda)$ and $\nu^*(\lambda)$, i.e.,

$$\lambda_{k+1} = \Gamma_\Lambda\left(\lambda_k + \zeta_1(k)\left(\nu^*(\lambda_k) + \frac{1}{1-\alpha}\frac{1}{N}\sum_{j=1}^{N}\left(\mathcal{J}(\xi_{j,k}) - \nu^*(\lambda_k)\right)^+ - \beta\right)\right). \tag{45}$$

Consider the continuous time system

$$\dot{\lambda}(t) = \Upsilon_\lambda\left[\nabla_\lambda L(\nu, \theta, \lambda)\Big|_{\theta=\theta^*(\lambda),\nu=\nu^*(\lambda)}\right], \qquad \lambda(t) \geq 0, \tag{46}$$

where

$$\Upsilon_\lambda[K(\lambda)] := \lim_{0<\eta\to 0} \frac{\Gamma_\Lambda\big(\lambda + \eta K(\lambda)\big) - \Gamma_\Lambda(\lambda)}{\eta}.$$

Again, similar to the analysis of (ν, θ), $\Upsilon_\lambda[K(\lambda)]$ is the left directional derivative of the function $\Gamma_\Lambda(\lambda)$ in the direction of $K(\lambda)$. By using the left directional derivative $\Upsilon_\lambda\left[\nabla_\lambda L(\nu, \theta, \lambda)\right]$ in the gradient ascent algorithm for λ, the gradient will point in the ascent direction along the boundary of $[0, \lambda_{\max}]$ whenever the λ-update hits its boundary.

Define

$$L^*(\lambda) = L(\nu^*(\lambda), \theta^*(\lambda), \lambda),$$

for $\lambda \geq 0$ where $(\theta^*(\lambda), \nu^*(\lambda)) \in \Theta \times [-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma}]$ is a local minimum of $L(\nu, \theta, \lambda)$ for fixed $\lambda \geq 0$, i.e., $L(\nu, \theta, \lambda) \geq L(\nu^*(\lambda), \theta^*(\lambda), \lambda)$ for any $(\theta, \nu) \in \Theta \times [-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma}] \cap \mathcal{B}_{(\theta^*(\lambda),\nu^*(\lambda))}(r)$ for some $r > 0$.

Next, we want to show that the ODE (46) is actually a gradient ascent of the Lagrangian function using the envelope theorem from mathematical economics (Milgrom and Segal, 2002). The envelope theorem describes sufficient conditions for the derivative of L^* with respect to λ to equal the partial derivative of the objective function L with respect to λ, holding (θ, ν) at its local optimum $(\theta, \nu) = (\theta^*(\lambda), \nu^*(\lambda))$. We will show that $\nabla_\lambda L^*(\lambda)$ coincides with $\nabla_\lambda L(\nu, \theta, \lambda)|_{\theta=\theta^*(\lambda),\nu=\nu^*(\lambda)}$ as follows.

Theorem 16 *The value function L^* is absolutely continuous. Furthermore,*

$$L^*(\lambda) = L^*(0) + \int_0^\lambda \nabla_{\lambda'} L(\nu, \theta, \lambda')\Big|_{\theta=\theta^*(s),\nu=\nu^*(s),\lambda'=s} ds, \;\; \lambda \geq 0. \tag{47}$$

Proof. The proof follows from analogous arguments to Lemma 4.3 in Borkar (2005). From the definition of L^*, observe that for any $\lambda', \lambda'' \geq 0$ with $\lambda' < \lambda''$,

$$\begin{aligned}
|L^*(\lambda'') - L^*(\lambda')| &\leq \sup_{\theta\in\Theta,\nu\in[-\frac{D_{\max}}{1-\gamma},\frac{D_{\max}}{1-\gamma}]} |L(\nu, \theta, \lambda'') - L(\nu, \theta, \lambda')| \\
&= \sup_{\theta\in\Theta,\nu\in[-\frac{D_{\max}}{1-\gamma},\frac{D_{\max}}{1-\gamma}]} \left|\int_{\lambda'}^{\lambda''} \nabla_\lambda L(\nu, \theta, s) ds\right| \\
&\leq \int_{\lambda'}^{\lambda''} \sup_{\theta\in\Theta,\nu\in[\frac{-D_{\max}}{1-\gamma},\frac{D_{\max}}{1-\gamma}]} |\nabla_\lambda L(\nu, \theta, s)|\, ds \leq \frac{3D_{\max}}{(1-\alpha)(1-\gamma)}(\lambda'' - \lambda').
\end{aligned}$$

This implies that L^* is absolutely continuous. Therefore, L^* is continuous everywhere and differentiable almost everywhere.

By the Milgrom–Segal envelope theorem in mathematical economics (Theorem 1 of Milgrom and Segal (2002)), one concludes that the derivative of $L^*(\lambda)$ coincides with the derivative of $L(\nu, \theta, \lambda)$ at the point of differentiability λ and $\theta = \theta^*(\lambda)$, $\nu = \nu^*(\lambda)$. Also since L^* is absolutely continuous, the limit of $(L^*(\lambda) - L^*(\lambda'))/(\lambda - \lambda')$ at $\lambda \uparrow \lambda'$ (or $\lambda \downarrow \lambda'$) coincides with the lower/upper directional derivatives if λ' is a point of non-differentiability. Thus, there is only a countable number of non-differentiable points in L^* and the set of non-differentiable points of L^* has measure zero. Therefore, expression (47) holds and one concludes that $\nabla_\lambda L^*(\lambda)$ coincides with $\nabla_\lambda L(\nu, \theta, \lambda)|_{\theta=\theta^*(\lambda),\nu=\nu^*(\lambda)}$. ■

Before getting into the main result, we have the following technical proposition whose proof directly follows from the definition of $\log \mathbb{P}_\theta(\xi)$ and Assumption 3 that $\nabla_\theta \mu(a_k|x_k; \theta)$ is Lipschitz in θ.

Proposition 17 *$\nabla_\theta L(\nu, \theta, \lambda)$ is Lipschitz in θ.*

Proof. Recall that

$$\nabla_\theta L(\nu, \theta, \lambda) = \sum_\xi \mathbb{P}_\theta(\xi) \cdot \nabla_\theta \log \mathbb{P}_\theta(\xi) \left(\mathcal{G}(\xi) + \frac{\lambda}{1-\alpha}\big(\mathcal{J}(\xi) - \nu\big)\mathbf{1}\big\{\mathcal{J}(\xi) \geq \nu\big\}\right)$$

and $\nabla_\theta \log \mathbb{P}_\theta(\xi) = \sum_{k=0}^{T-1} \nabla_\theta \mu(a_k|x_k; \theta)/\mu(a_k|x_k; \theta)$ whenever $\mu(a_k|x_k; \theta) \in (0, 1]$. Now Assumption (A1) implies that $\nabla_\theta \mu(a_k|x_k; \theta)$ is a Lipschitz function in θ for any $a \in \mathcal{A}$ and $k \in \{0, \ldots, T-1\}$ and $\mu(a_k|x_k; \theta)$ is differentiable in θ. Therefore, by recalling that

$$\mathbb{P}_\theta(\xi) = \prod_{k=0}^{T-1} P(x_{k+1}|x_k, a_k)\mu(a_k|x_k; \theta)\mathbf{1}\{x_0 = x^0\}$$

and by combining these arguments and noting that the sum of products of Lipschitz functions is Lipschitz, one concludes that $\nabla_\theta L(\nu, \theta, \lambda)$ is Lipschitz in θ. ■

Remark 18 *The fact that $\nabla_\theta L(\nu, \theta, \lambda)$ is Lipschitz in θ implies that*

$$\|\nabla_\theta L(\nu, \theta, \lambda)\|^2 \leq 2(\|\nabla_\theta L(\nu, \theta_0, \lambda)\| + \|\theta_0\|)^2 + 2\|\theta\|^2$$

which further implies that

$$\|\nabla_\theta L(\nu, \theta, \lambda)\|^2 \leq K_1(1 + \|\theta\|^2).$$

for $K_1 = 2\max(1, (\|\nabla_\theta L(\nu, \theta_0, \lambda)\| + \|\theta_0\|)^2) > 0$. *Similarly, the fact that* $\nabla_\theta \log \mathbb{P}_\theta(\xi)$ *is Lipschitz implies that*

$$\|\nabla_\theta \log \mathbb{P}_\theta(\xi)\|^2 \leq K_2(\xi)(1 + \|\theta\|^2)$$

for a positive random variable $K_2(\xi)$. *Furthermore, since* $T < \infty$ *w.p.* 1, $\mu(a_k|x_k;\theta) \in (0, 1]$ *and* $\nabla_\theta \mu(a_k|x_k;\theta)$ *is Lipschitz for any* $k < T$, $K_2(\xi) < \infty$ *w.p.* 1.

Remark 19 *For any given* $\theta \in \Theta$, $\lambda \geq 0$, *and* $g(\nu) \in \partial_\nu L(\nu, \theta, \lambda)$, *we have*

$$|g(\nu)| \leq 3\lambda(1 + |\nu|)/(1 - \alpha). \tag{48}$$

To see this, recall that the set of $g(\nu) \in \partial_\nu L(\nu, \theta, \lambda)$ *can be parameterized by* $q \in [0, 1]$ *as*

$$g(\nu; q) = -\frac{\lambda}{(1-\alpha)} \sum_\xi \mathbb{P}_\theta(\xi)\mathbf{1}\left\{\mathcal{J}(\xi) > \nu\right\} - \frac{\lambda q}{1-\alpha} \sum_\xi \mathbb{P}_\theta(\xi)\mathbf{1}\left\{\mathcal{J}(\xi) = \nu\right\} + \lambda.$$

It is obvious that $|\mathbf{1}\left\{\mathcal{J}(\xi) = \nu\right\}|, |\mathbf{1}\left\{\mathcal{J}(\xi) > \nu\right\}| \leq 1 + |\nu|$. *Thus,* $\left|\sum_\xi \mathbb{P}_\theta(\xi)\mathbf{1}\left\{\mathcal{J}(\xi) > \nu\right\}\right| \leq \sup_\xi |\mathbf{1}\left\{\mathcal{J}(\xi) > \nu\right\}| \leq 1 + |\nu|$, *and* $\left|\sum_\xi \mathbb{P}_\theta(\xi)\mathbf{1}\left\{\mathcal{J}(\xi) = \nu\right\}\right| \leq 1 + |\nu|$. *Recalling that* $0 < (1 - q), (1 - \alpha) < 1$, *these arguments imply the claim of* (48).

We are now in a position to prove the convergence analysis of Theorem 7.

Proof. [Proof of Theorem 7] We split the proof into the following four steps:

Step 1 (Convergence of ν-update) Since ν converges on a faster time scale than θ and λ, according to Lemma 1 in Chapter 6 of Borkar (2008), one can analyze the convergence properties of ν in the following update rule for arbitrary quantities of θ and λ (i.e., here we have $\theta = \theta_k$ and $\lambda = \lambda_k$):

$$\nu_{k+1} = \Gamma_{\mathcal{N}}\left(\nu_k + \zeta_3(k)\left(\frac{\lambda}{(1-\alpha)N}\sum_{j=1}^N \mathbf{1}\big\{\mathcal{J}(\xi_{j,k}) \geq \nu_k\big\} - \lambda + \delta\nu_{k+1}\right)\right), \tag{49}$$

and the Martingale difference term with respect to ν is given by

$$\delta\nu_{k+1} = \frac{\lambda}{1-\alpha}\left(-\frac{1}{N}\sum_{j=1}^N \mathbf{1}\big\{\mathcal{J}(\xi_{j,k}) \geq \nu_k\big\} + \sum_\xi \mathbb{P}_\theta(\xi)\mathbf{1}\{\mathcal{J}(\xi) \geq \nu_k\}\right). \tag{50}$$

First, one can show that $\delta\nu_{k+1}$ is square integrable, i.e.,

$$\mathbb{E}[\|\delta\nu_{k+1}\|^2 \mid F_{\nu,k}] \leq 4\left(\frac{\lambda_{\max}}{1-\alpha}\right)^2$$

where $\mathcal{F}_{\nu,k} = \sigma\big(\nu_m,\, \delta\nu_m,\, m \leq k\big)$ is the filtration of ν_k generated by different independent trajectories.

Second, since the history trajectories are generated based on the sampling probability mass function $\mathbb{P}_\theta(\xi)$, expression (40) implies that $\mathbb{E}\left[\delta\nu_{k+1} \mid \mathcal{F}_{\nu,k}\right] = 0$. Therefore, the ν-update is a stochastic approximation of the ODE (43) with a Martingale difference error term, i.e.,

$$\frac{\lambda}{1-\alpha}\sum_\xi \mathbb{P}_\theta(\xi)\mathbf{1}\{\mathcal{J}(\xi) \geq \nu_k\} - \lambda \in -\partial_\nu L(\nu, \theta, \lambda)|_{\nu=\nu_k}.$$

Then one can invoke Corollary 4 in Chapter 5 of Borkar (2008) (stochastic approximation theory for non-differentiable systems) to show that the sequence $\{\nu_k\}$, $\nu_k \in [-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma}]$ converges almost surely to a fixed point $\nu^* \in [-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma}]$ of the differential inclusion (43), where

$$\nu^* \in N_c := \left\{\nu \in \left[-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma}\right] : \Upsilon_\nu[-g(\nu)] = 0,\, g(\nu) \in \partial_\nu L(\nu, \theta, \lambda)\right\}.$$

To justify the assumptions of this corollary, 1) from Remark 19, the Lipschitz property is satisfied, i.e., $\sup_{g(\nu)\in\partial_\nu L(\nu,\theta,\lambda)} |g(\nu)| \leq 3\lambda(1+|\nu|)/(1-\alpha)$, 2) $[-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma}]$ and $\partial_\nu L(\nu, \theta, \lambda)$ are convex compact sets by definition, which implies $\{(\nu, g(\nu)) \mid g(\nu) \in \partial_\nu L(\nu, \theta, \lambda)\}$ is a closed set, and further implies $\partial_\nu L(\nu, \theta, \lambda)$ is an upper semi-continuous set valued mapping, 3) the step-size rule follows from Assumption 6, 4) the Martingale difference assumption follows from (50), and 5) $\nu_k \in [-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma}], \forall i$ implies that $\sup_k \|\nu_k\| < \infty$ almost surely.

Consider the ODE for $\nu \in \mathbb{R}$ in (43), we define the set-valued derivative of L as follows:

$$D_t L(\nu, \theta, \lambda) = \big\{g(\nu)\Upsilon_\nu\big[-g(\nu)\big] \mid \forall g(\nu) \in \partial_\nu L(\nu, \theta, \lambda)\big\}.$$

One can conclude that

$$\max_{g(\nu)} D_t L(\nu, \theta, \lambda) = \max\big\{g(\nu)\Upsilon_\nu\big[-g(\nu)\big] \mid g(\nu) \in \partial_\nu L(\nu, \theta, \lambda)\big\}.$$

We now show that $\max_{g(\nu)} D_t L(\nu, \theta, \lambda) \leq 0$ and this quantity is non-zero if $\Upsilon_\nu\big[-g(\nu)\big] \neq 0$ for every $g(\nu) \in \partial_\nu L(\nu, \theta, \lambda)$ by considering three cases. To distinguish the latter two cases, we need to define,

$$\mathcal{J}(\nu) := \left\{g(\nu) \in \partial L_\nu(\nu, \theta, \lambda) \,\middle|\, \forall \eta_0 > 0,\, \exists \eta \in (0, \eta_0] \text{ such that } \theta - \eta g(\nu) \notin \left[-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma}\right]\right\}.$$

Case 1: $\nu \in (-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma})$.
For every $g(\nu) \in \partial_\nu L(\nu, \theta, \lambda)$, there exists a sufficiently small $\eta_0 > 0$ such that $\nu - \eta_0 g(\nu) \in [-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma}]$ and

$$\Gamma_{\mathcal{N}}\big(\theta - \eta_0 g(\nu)\big) - \theta = -\eta_0 g(\nu).$$

Therefore, the definition of $\Upsilon_\theta[-g(\nu)]$ implies

$$\max_{g(\nu)} D_t L(\nu, \theta, \lambda) = \max\big\{-g^2(\nu) \mid g(\nu) \in \partial_\nu L(\nu, \theta, \lambda)\big\} \leq 0. \tag{51}$$

The maximum is attained because $\partial_\nu L(\nu, \theta, \lambda)$ is a convex compact set and $g(\nu)\Upsilon_\nu\big[-g(\nu)\big]$ is a continuous function. At the same time, we have $\max_{g(\nu)} D_t L(\nu, \theta, \lambda) < 0$ whenever $0 \notin \partial_\nu L(\nu, \theta, \lambda)$.
Case 2: $\nu \in \{-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma}\}$ *and* $\mathcal{J}(\nu)$ *is empty.*
The condition $\nu - \eta g(\nu) \in [-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma}]$ implies that

$$\Upsilon_\nu\big[-g(\nu)\big] = -g(\nu).$$

Then we obtain

$$\max_{g(\nu)} D_t L(\nu, \theta, \lambda) = \max\big\{-g^2(\nu) \mid g(\nu) \in \partial_\nu L(\nu, \theta, \lambda)\big\} \leq 0. \tag{52}$$

Furthermore, we have $\max_{g(\nu)} D_t L(\nu,\theta,\lambda) < 0$ whenever $0 \notin \partial_\nu L(\nu,\theta,\lambda)$.

Case 3: $\nu \in \{-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma}\}$ *and* $\mathcal{J}(\nu)$ *is nonempty*.

First, consider any $g(\nu) \in \mathcal{J}(\nu)$. For any $\eta > 0$, define $\nu_\eta := \nu - \eta g(\nu)$. The above condition implies that when $0 < \eta \to 0$, $\Gamma_{\mathcal{N}}[\nu_\eta]$ is the projection of ν_η to the tangent space of $[-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma}]$. For any element $\hat{\nu} \in [-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma}]$, since the set $\{\nu \in [-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma}] : \|\nu - \nu_\eta\|_2 \leq \|\hat{\nu} - \nu_\eta\|_2\}$ is compact, the projection of ν_η on $[-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma}]$ exists. Furthermore, since $f(\nu) := \frac{1}{2}(\nu - \nu_\eta)^2$ is a strongly convex function and $\nabla f(\nu) = \nu - \nu_\eta$, by the first order optimality condition, one obtains

$$\nabla f(\nu_\eta^*)(\nu - \nu_\eta^*) = (\nu_\eta^* - \nu_\eta)(\nu - \nu_\eta^*) \geq 0, \quad \forall \nu \in \left[-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma}\right]$$

where ν_η^* is the unique projection of ν_η (the projection is unique because $f(\nu)$ is strongly convex and $[-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma}]$ is a convex compact set). Since the projection (minimizer) is unique, the above equality holds if and only if $\nu = \nu_\eta^*$.

Therefore, for any $\nu \in [-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma}]$ and $\eta > 0$,

$$g(\nu)\Upsilon_\nu\big[-g(\nu)\big] = g(\nu)\left(\lim_{0<\eta\to 0}\frac{\nu_\eta^* - \nu}{\eta}\right)$$

$$= \left(\lim_{0<\eta\to 0}\frac{\nu - \nu_\eta}{\eta}\right)\left(\lim_{0<\eta\to 0}\frac{\nu_\eta^* - \nu}{\eta}\right) = \lim_{0<\eta\to 0}\frac{-\|\nu_\eta^* - \nu\|^2}{\eta^2} + \lim_{0<\eta\to 0}\big(\nu_\eta^* - \nu_\eta\big)\left(\frac{\nu_\eta^* - \nu}{\eta^2}\right) \leq 0.$$

Second, for any $g(\nu) \in \partial_\nu L(\nu,\theta,\lambda) \cap \mathcal{J}(\nu)^c$, one obtains $\nu - \eta g(\nu) \in [-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma}]$, for any $\eta \in (0, \eta_0]$ and some $\eta_0 > 0$. In this case, the arguments follow from case 2 and the following expression holds: $\Upsilon_\nu\big[-g(\nu)\big] = -g(\nu)$.

Combining these arguments, one concludes that

$$\max_{g(\nu)} D_t L(\nu,\theta,\lambda)$$
$$\leq \max\big\{\max\big\{g(\nu)\,\Upsilon_\nu\big[-g(\nu)\big] \mid g(\nu) \in \mathcal{J}(\nu)\big\}, \max\big\{-g^2(\nu) \mid g(\nu) \in \partial_\nu L(\nu,\theta,\lambda) \cap \mathcal{J}(\nu)^c\big\}\big\} \leq 0. \tag{53}$$

This quantity is non-zero whenever $0 \notin \{g(\nu)\,\Upsilon_\nu\big[-g(\nu)\big] \mid \forall g(\nu) \in \partial_\nu L(\nu,\theta,\lambda)\}$ (this is because, for any $g(\nu) \in \partial_\nu L(\nu,\theta,\lambda) \cap \mathcal{J}(\nu)^c$, one obtains $g(\nu)\,\Upsilon_\nu\big[-g(\nu)\big] = -g(\nu)^2$). Thus, by similar arguments one may conclude that $\max_{g(\nu)} D_t L(\nu,\theta,\lambda) \leq 0$ and it is non-zero if $\Upsilon_\nu\big[-g(\nu)\big] \neq 0$ for every $g(\nu) \in \partial_\nu L(\nu,\theta,\lambda)$.

Now for any given θ and λ, define the following Lyapunov function

$$\mathcal{L}_{\theta,\lambda}(\nu) = L(\nu,\theta,\lambda) - L(\nu^*,\theta,\lambda)$$

where ν^* is a minimum point (for any given (θ,λ), L is a convex function in ν). Then $\mathcal{L}_{\theta,\lambda}(\nu)$ is a positive definite function, i.e., $\mathcal{L}_{\theta,\lambda}(\nu) \geq 0$. On the other hand, by the definition of a minimum point, one easily obtains $0 \in \{g(\nu^*)\,\Upsilon_\nu\big[-g(\nu^*)\big]|_{\nu=\nu^*} \mid \forall g(\nu^*) \in \partial_\nu L(\nu,\theta,\lambda)|_{\nu=\nu^*}\}$ which means that ν^* is also a stationary point, i.e., $\nu^* \in N_c$.

Note that $\max_{g(\nu)} D_t \mathcal{L}_{\theta,\lambda}(\nu) = \max_{g(\nu)} D_t L(\nu,\theta,\lambda) \leq 0$ and this quantity is non-zero if $\Upsilon_\nu\big[-g(\nu)\big] \neq 0$ for every $g(\nu) \in \partial_\nu L(\nu,\theta,\lambda)$. Therefore, by the Lyapunov theory for asymptotically stable differential inclusions (see Theorem 3.10 and Corollary 3.11 in Benaim et al. (2006), where the Lyapunov function $\mathcal{L}_{\theta,\lambda}(\nu)$ satisfies Hypothesis 3.1 and the property in (53) is equivalent

to Hypothesis 3.9 in the reference), the above arguments imply that with any initial condition $\nu(0)$, the state trajectory $\nu(t)$ of (43) converges to ν^*, i.e., $L(\nu^*,\theta,\lambda) \leq L(\nu(t),\theta,\lambda) \leq L(\nu(0),\theta,\lambda)$ for any $t \geq 0$.

As stated earlier, the sequence $\{\nu_k\}$, $\nu_k \in [-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma}]$ constitutes a stochastic approximation to the differential inclusion (43), and thus converges almost surely its solution (Borkar, 2008), which further converges almost surely to $\nu^* \in N_c$. Also, it can be easily seen that N_c is a closed subset of the compact set $[-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma}]$, and therefore a compact set itself.

Step 2 (Convergence of θ-update) Since θ converges on a faster time scale than λ and ν converges faster than θ, according to Lemma 1 in Chapter 6 of Borkar (2008) one can prove convergence of the θ update for any arbitrary λ (i.e., $\lambda = \lambda_k$). Furthermore, in the θ-update, we have that $\|\nu_k - \nu^*(\theta_k)\| \to 0$ almost surely. By the continuity condition of $\nabla_\theta L(\nu,\theta,\lambda)$, this also implies $\|\nabla_\theta L(\nu,\theta,\lambda)|_{\theta=\theta_k,\nu=\nu_k} - \nabla_\theta L(\nu,\theta,\lambda)|_{\theta=\theta_k,\nu=\nu^*(\theta_k)}\| \to 0$. Therefore, the θ-update can be rewritten as a stochastic approximation, i.e.,

$$\theta_{k+1} = \Gamma_\Theta\left(\theta_k + \zeta_2(k)\Big(-\nabla_\theta L(\nu,\theta,\lambda)|_{\theta=\theta_k,\nu=\nu^*(\theta_k)} + \delta\theta_{k+1}\Big)\right), \tag{54}$$

where

$$\begin{aligned}\delta\theta_{k+1} =& \nabla_\theta L(\nu,\theta,\lambda)|_{\theta=\theta_k,\nu=\nu^*(\theta_k)} - \frac{1}{N}\sum_{j=1}^N \nabla_\theta \log \mathbb{P}_\theta(\xi_{j,k})\,|_{\theta=\theta_k}\,\mathcal{G}(\xi_{j,k})\\ &- \frac{\lambda}{(1-\alpha)N}\sum_{j=1}^N \nabla_\theta \log \mathbb{P}_\theta(\xi_{j,k})|_{\theta=\theta_k}\big(\mathcal{J}(\xi_{j,k}) - \nu^*(\theta_k)\big)\mathbf{1}\big\{\mathcal{J}(\xi_{j,k}) \geq \nu^*(\theta_k)\big\}\\ &+ \frac{\lambda}{(1-\alpha)N}\sum_{j=1}^N \nabla_\theta \log \mathbb{P}_\theta(\xi_{j,k})|_{\theta=\theta_k}\big(\nu^*(\theta_k) - \nu_k\big)\mathbf{1}\big\{\mathcal{J}(\xi_{j,k}) \geq \nu^*(\theta_k)\big\}\\ &+ \frac{\lambda}{(1-\alpha)N}\sum_{j=1}^N \nabla_\theta \log \mathbb{P}_\theta(\xi_{j,k})|_{\theta=\theta_k}\big(\mathcal{J}(\xi_{j,k}) - \nu_k\big)\big(\mathbf{1}\big\{\mathcal{J}(\xi_{j,k}) \geq \nu_k\big\} - \mathbf{1}\big\{\mathcal{J}(\xi_{j,k}) \geq \nu^*(\theta_k)\big\}\big).\end{aligned} \tag{55}$$

First, we consider the last two components in (61). Recall that $\|\nu_k - \nu^*(\theta_k)\| \to 0$ almost surely. Furthermore by noticing that $\nabla_\theta \log \mathbb{P}_\theta(\xi_{j,k})$ is Lipschitz in θ, θ lies in a compact set Θ, both $\mathcal{J}(\xi_{j,k})$ and ν_k are bounded, and $\nu, \nu^*(\theta_k)$ lie in a compact set $\mathcal{N}$, one immediately concludes that as $i \to \infty$,

$$\begin{aligned}&\big(\nu^*(\theta_k) - \nu_k\big)\mathbf{1}\big\{\mathcal{J}(\xi_{j,k}) \geq \nu^*(\theta_k)\big\} \to 0, \quad \text{almost surely}\\ &\big(\mathcal{J}(\xi_{j,k}) - \nu_k\big)\big(\mathbf{1}\big\{\mathcal{J}(\xi_{j,k}) \geq \nu_k\big\} - \mathbf{1}\big\{\mathcal{J}(\xi_{j,k}) \geq \nu^*(\theta_k)\big\}\big) \to 0, \quad \text{almost surely}\end{aligned} \tag{56}$$

Second, one can show that $\delta\theta_{k+1}$ is square integrable, i.e., $\mathbb{E}[\|\delta\theta_{k+1}\|^2 \mid F_{\theta,k}] \leq K_k(1 + \|\theta_k\|^2)$ for some $K_k > 0$, where $\mathcal{F}_{\theta,k} = \sigma\big(\theta_m, \delta\theta_m, m \leq k\big)$ is the filtration of θ_k generated by different

independent trajectories. To see this, notice that

$$
\begin{aligned}
&\|\delta\theta_{k+1}\|^2 \\
\leq& 2\left(\nabla_\theta L(\nu,\theta,\lambda)|_{\theta=\theta_k,\nu=\nu^*(\theta_k)}\right)^2 + \frac{2}{N^2}\left(\frac{C_{\max}}{1-\gamma}+\frac{2\lambda D_{\max}}{(1-\alpha)(1-\gamma)}\right)^2\left(\sum_{j=1}^N \nabla_\theta \log \mathbb{P}_\theta(\xi_{j,k})\ |_{\theta=\theta_k}\right)^2 \\
\leq& 2K_{1,k}(1+\|\theta_k\|^2) + \frac{2^N}{N^2}\left(\frac{C_{\max}}{1-\gamma}+\frac{2\lambda_{\max} D_{\max}}{(1-\alpha)(1-\gamma)}\right)^2\left(\sum_{j=1}^N \|\nabla_\theta \log \mathbb{P}_\theta(\xi_{j,k})\ |_{\theta=\theta_k}\|^2\right) \\
\leq& 2K_{1,k}(1+\|\theta_k\|^2) + \frac{2^N}{N^2}\left(\frac{C_{\max}}{1-\gamma}+\frac{2\lambda_{\max} D_{\max}}{(1-\alpha)(1-\gamma)}\right)^2\left(\sum_{j=1}^N K_2(\xi_{j,k})(1+\|\theta_k\|^2)\right) \\
\leq& 2\left(K_{1,k}+\frac{2^{N-1}}{N}\left(\frac{C_{\max}}{1-\gamma}+\frac{2\lambda_{\max} D_{\max}}{(1-\alpha)(1-\gamma)}\right)^2 \max_{1\leq j\leq N} K_2(\xi_{j,k})\right)(1+\|\theta_k\|^2).
\end{aligned}
$$

The Lipschitz upper bounds are due to the results in Remark 18. Since $K_2(\xi_{j,k}) < \infty$ w.p. 1, there exists $K_{2,k} < \infty$ such that $\max_{1\leq j\leq N} K_2(\xi_{j,k}) \leq K_{2,k}$. By combining these results, one concludes that $\mathbb{E}[\|\delta\theta_{k+1}\|^2 \mid F_{\theta,k}] \leq K_k(1+\|\theta_k\|^2)$ where

$$
K_k = 2\left(K_{1,k}+\frac{2^{N-1}K_{2,k}}{N}\left(\frac{C_{\max}}{1-\gamma}+\frac{2\lambda_{\max} D_{\max}}{(1-\alpha)(1-\gamma)}\right)^2\right) < \infty.
$$

Third, since the history trajectories are generated based on the sampling probability mass function $\mathbb{P}_{\theta_k}(\xi)$, expression (39) implies that $\mathbb{E}\left[\delta\theta_{k+1} \mid \mathcal{F}_{\theta,k}\right] = 0$. Therefore, the θ-update is a stochastic approximation of the ODE (44) with a Martingale difference error term. In addition, from the convergence analysis of the ν-update, $\nu^*(\theta)$ is an asymptotically stable equilibrium point for the sequence $\{\nu_k\}$. From (40), $\partial_\nu L(\nu,\theta,\lambda)$ is a Lipschitz set-valued mapping in θ (since $\mathbb{P}_\theta(\xi)$ is Lipschitz in θ), and thus it can be easily seen that $\nu^*(\theta)$ is a Lipschitz continuous mapping of θ.

Now consider the continuous time dynamics for $\theta \in \Theta$, given in (44). We may write

$$
\left.\frac{dL(\nu,\theta,\lambda)}{dt}\right|_{\nu=\nu^*(\theta)} = \left(\nabla_\theta L(\nu,\theta,\lambda)|_{\nu=\nu^*(\theta)}\right)^\top \Upsilon_\theta\left[-\nabla_\theta L(\nu,\theta,\lambda)|_{\nu=\nu^*(\theta)}\right]. \tag{57}
$$

By considering the following cases, we now show that $dL(\nu,\theta,\lambda)/dt|_{\nu=\nu^*(\theta)} \leq 0$ and this quantity is non-zero whenever $\left\|\Upsilon_\theta\left[-\nabla_\theta L(\nu,\theta,\lambda)|_{\nu=\nu^*(\theta)}\right]\right\| \neq 0$.

Case 1: When $\theta \in \Theta^\circ = \Theta \setminus \partial\Theta$.
Since Θ° is the interior of the set Θ and Θ is a convex compact set, there exists a sufficiently small $\eta_0 > 0$ such that $\theta - \eta_0\nabla_\theta L(\nu,\theta,\lambda)|_{\nu=\nu^*(\theta)} \in \Theta$ and

$$
\Gamma_\Theta\left(\theta - \eta_0\nabla_\theta L(\nu,\theta,\lambda)|_{\nu=\nu^*(\theta)}\right) - \theta = -\eta_0\nabla_\theta L(\nu,\theta,\lambda)|_{\nu=\nu^*(\theta)}.
$$

Therefore, the definition of $\Upsilon_\theta\left[-\nabla_\theta L(\nu,\theta,\lambda)|_{\nu=\nu^*(\theta)}\right]$ implies

$$
\left.\frac{dL(\nu,\theta,\lambda)}{dt}\right|_{\nu=\nu^*(\theta)} = -\left\|\nabla_\theta L(\nu,\theta,\lambda)|_{\nu=\nu^*(\theta)}\right\|^2 \leq 0. \tag{58}
$$

At the same time, we have $dL(\nu,\theta,\lambda)/dt|_{\nu=\nu^*(\theta)} < 0$ whenever $\|\nabla_\theta L(\nu,\theta,\lambda)|_{\nu=\nu^*(\theta)}\| \neq 0$.

Case 2: When $\theta \in \partial\Theta$ and $\theta - \eta\nabla_\theta L(\nu,\theta,\lambda)|_{\nu=\nu^*(\theta)} \in \Theta$ for any $\eta \in (0,\eta_0]$ and some $\eta_0 > 0$.
The condition $\theta - \eta\nabla_\theta L(\nu,\theta,\lambda)|_{\nu=\nu^*(\theta)} \in \Theta$ implies that

$$
\Upsilon_\theta\left[-\nabla_\theta L(\nu,\theta,\lambda)|_{\nu=\nu^*(\theta)}\right] = -\nabla_\theta L(\nu,\theta,\lambda)|_{\nu=\nu^*(\theta)}.
$$

Then we obtain

$$
\left.\frac{dL(\nu,\theta,\lambda)}{dt}\right|_{\nu=\nu^*(\theta)} = -\left\|\nabla_\theta L(\nu,\theta,\lambda)|_{\nu=\nu^*(\theta)}\right\|^2 \leq 0. \tag{59}
$$

Furthermore, $dL(\nu,\theta,\lambda)/dt|_{\nu=\nu^*(\theta)} < 0$ when $\|\nabla_\theta L(\nu,\theta,\lambda)|_{\nu=\nu^*(\theta)}\| \neq 0$.

Case 3: When $\theta \in \partial\Theta$ and $\theta - \eta\nabla_\theta L(\nu,\theta,\lambda)|_{\nu=\nu^*(\theta)} \notin \Theta$ for some $\eta \in (0,\eta_0]$ and any $\eta_0 > 0$.
For any $\eta > 0$, define $\theta_\eta := \theta - \eta\nabla_\theta L(\nu,\theta,\lambda)|_{\nu=\nu^*(\theta)}$. The above condition implies that when $0 < \eta \to 0$, $\Gamma_\Theta[\theta_\eta]$ is the projection of θ_η to the tangent space of Θ. For any element $\hat{\theta} \in \Theta$, since the set $\{\theta \in \Theta : \|\theta - \theta_\eta\|_2 \leq \|\hat{\theta} - \theta_\eta\|_2\}$ is compact, the projection of θ_η on Θ exists. Furthermore, since $f(\theta) := \frac{1}{2}\|\theta - \theta_\eta\|_2^2$ is a strongly convex function and $\nabla f(\theta) = \theta - \theta_\eta$, by the first order optimality condition, one obtains

$$
\nabla f(\theta_\eta^*)^\top(\theta - \theta_\eta^*) = (\theta_\eta^* - \theta_\eta)^\top(\theta - \theta_\eta^*) \geq 0, \quad \forall\theta \in \Theta,
$$

where θ_η^* is the unique projection of θ_η (the projection is unique because $f(\theta)$ is strongly convex and Θ is a convex compact set). Since the projection (minimizer) is unique, the above equality holds if and only if $\theta = \theta_\eta^*$.

Therefore, for any $\theta \in \Theta$ and $\eta > 0$,

$$
\begin{aligned}
&\left(\nabla_\theta L(\nu,\theta,\lambda)|_{\nu=\nu^*(\theta)}\right)^\top \Upsilon_\theta\left[-\nabla_\theta L(\nu,\theta,\lambda)|_{\nu=\nu^*(\theta)}\right] = \left(\nabla_\theta L(\nu,\theta,\lambda)|_{\nu=\nu^*(\theta)}\right)^\top \left(\lim_{0<\eta\to 0}\frac{\theta_\eta^* - \theta}{\eta}\right) \\
=& \left(\lim_{0<\eta\to 0}\frac{\theta - \theta_\eta}{\eta}\right)^\top \left(\lim_{0<\eta\to 0}\frac{\theta_\eta^* - \theta}{\eta}\right) = \lim_{0<\eta\to 0}\frac{-\|\theta_\eta^* - \theta\|^2}{\eta^2} + \lim_{0<\eta\to 0}\left(\theta_\eta^* - \theta_\eta\right)^\top\left(\frac{\theta_\eta^* - \theta}{\eta^2}\right) \leq 0.
\end{aligned}
$$

By combining these arguments, one concludes that $dL(\nu,\theta,\lambda)/dt|_{\nu=\nu^*(\theta)} \leq 0$ and this quantity is non-zero whenever $\left\|\Upsilon_\theta\left[-\nabla_\theta L(\nu,\theta,\lambda)|_{\nu=\nu^*(\theta)}\right]\right\| \neq 0$.

Now, for any given λ, define the Lyapunov function

$$
\mathcal{L}_\lambda(\theta) = L(\nu^*(\theta),\theta,\lambda) - L(\nu^*(\theta^*),\theta^*,\lambda),
$$

where θ^* is a local minimum point. Then there exists a ball centered at θ^* with radius r such that for any $\theta \in \mathcal{B}_{\theta^*}(r)$, $\mathcal{L}_\lambda(\theta)$ is a locally positive definite function, i.e., $\mathcal{L}_\lambda(\theta) \geq 0$. On the other hand, by the definition of a local minimum point, one obtains $\Upsilon_\theta[-\nabla_\theta L(\theta^*,\nu,\lambda)|_{\nu=\nu^*(\theta^*)}]|_{\theta=\theta^*} = 0$ which means that θ^* is a stationary point, i.e., $\theta^* \in \Theta_c$.

Note that $d\mathcal{L}_\lambda(\theta(t))/dt = dL(\theta(t),\nu^*(\theta(t)),\lambda)/dt \leq 0$ and the time-derivative is non-zero whenever $\left\|\Upsilon_\theta\left[-\nabla_\theta L(\nu,\theta,\lambda)|_{\nu=\nu^*(\theta)}\right]\right\| \neq 0$. Therefore, by the Lyapunov theory for asymptotically stable systems from Chapter 4 of Khalil and Grizzle (2002), the above arguments imply that with any initial condition $\theta(0) \in \mathcal{B}_{\theta^*}(r)$, the state trajectory $\theta(t)$ of (44) converges to θ^*, i.e., $L(\theta^*,\nu^*(\theta^*),\lambda) \leq L(\theta(t),\nu^*(\theta(t)),\lambda) \leq L(\theta(0),\nu^*(\theta(0)),\lambda)$ for any $t \geq 0$.

Based on the above properties and noting that 1) from Proposition 17, $\nabla_\theta L(\nu,\theta,\lambda)$ is a Lipschitz function in θ, 2) the step-size rule follows from Assumption 6, 3) expression (61) implies that $\delta\theta_{k+1}$ is a square integrable Martingale difference, and 4) $\theta_k \in \Theta$, $\forall i$ implies that $\sup_k \|\theta_k\| < \infty$ almost surely, one can invoke Theorem 2 in Chapter 6 of Borkar (2008) (multi-time scale stochastic approximation theory) to show that the sequence $\{\theta_k\},\ \theta_k \in \Theta$ converges almost surely to the solution of the ODE (44), which further converges almost surely to $\theta^* \in \Theta$.

Step 3 (Local Minimum) Now, we want to show that the sequence $\{\theta_k, \nu_k\}$ converges to a local minimum of $L(\nu,\theta,\lambda)$ for any fixed λ. Recall that $\{\theta_k,\nu_k\}$ converges to $(\theta^*,\nu^*) := (\theta^*,\nu^*(\theta^*))$. Previous arguments on the (ν,θ)-convergence imply that with any initial condition $(\theta(0),\nu(0))$, the state trajectories $\theta(t)$ and $\nu(t)$ of (43) and (44) converge to the set of stationary points (θ^*,ν^*) in the positive invariant set $\Theta_c \times N_c$ and $L(\theta^*,\nu^*,\lambda) \leq L(\theta(t),\nu^*(\theta(t)),\lambda) \leq L(\theta(0),\nu^*(\theta(0)),\lambda) \leq L(\theta(0),\nu(t),\lambda) \leq L(\theta(0),\nu(0),\lambda)$ for any $t \geq 0$.

By contradiction, suppose (θ^*,ν^*) is not a local minimum. Then there exists $(\bar\theta,\bar\nu) \in \Theta \times [-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma}] \cap \mathcal{B}_{(\theta^*,\nu^*)}(r)$ such that

$$L(\bar\theta,\bar\nu,\lambda) = \min_{(\theta,\nu)\in\Theta\times[-\frac{D_{\max}}{1-\gamma},\frac{D_{\max}}{1-\gamma}]\cap\mathcal{B}_{(\theta^*,\nu^*)}(r)} L(\nu,\theta,\lambda).$$

The minimum is attained by the Weierstrass extreme value theorem. By putting $\theta(0) = \bar\theta$, the above arguments imply that

$$L(\bar\theta,\bar\nu,\lambda) = \min_{(\theta,\nu)\in\Theta\times[-\frac{D_{\max}}{1-\gamma},\frac{D_{\max}}{1-\gamma}]\cap\mathcal{B}_{(\theta^*,\nu^*)}(r)} L(\nu,\theta,\lambda) < L(\theta^*,\nu^*,\lambda) \leq L(\bar\theta,\bar\nu,\lambda)$$

which is a contradiction. Therefore, the stationary point (θ^*,ν^*) is a local minimum of $L(\nu,\theta,\lambda)$ as well.

Step 4 (Convergence of λ-update) Since the λ-update converges in the slowest time scale, according to previous analysis, we have that $\|\theta_k - \theta^*(\nu^*(\lambda_k),\lambda_k)\| \to 0$, $\|\nu_k - \nu^*(\lambda_k)\| \to 0$ almost surely. By continuity of $\nabla_\lambda L(\nu,\theta,\lambda)$, we also have the following:

$$\left\| \nabla_\lambda L(\nu,\theta,\lambda)\Big|_{\theta=\theta^*(\lambda_k),\nu=\nu^*(\lambda_k),\lambda=\lambda_k} - \nabla_\lambda L(\nu,\theta,\lambda)\Big|_{\theta=\theta_k,\nu=\nu_k,\lambda=\lambda_k} \right\| \to 0, \quad \text{almost surely.}$$

Therefore, the λ-update rule can be re-written as follows:

$$\lambda_{k+1} = \Gamma_\Lambda\left(\lambda_k + \zeta_1(k)\left(\nabla_\lambda L(\nu,\theta,\lambda)\Big|_{\theta=\theta^*(\lambda_k),\nu=\nu^*(\lambda_k),\lambda=\lambda_k} + \delta\lambda_{k+1}\right)\right) \tag{60}$$

where

$$\begin{aligned}\delta\lambda_{k+1} = &-\nabla_\lambda L(\nu,\theta,\lambda)\Big|_{\theta=\theta^*(\lambda),\nu=\nu^*(\lambda),\lambda=\lambda_k} + \left(\nu^*(\lambda_k) + \frac{1}{1-\alpha}\frac{1}{N}\sum_{j=1}^N \left(\mathcal{J}(\xi_{j,k}) - \nu^*(\lambda_k)\right)^+ - \beta\right)\\ &+(\nu_k - \nu^*(\lambda_k)) + \frac{1}{1-\alpha}\frac{1}{N}\sum_{j=1}^N\left(\left(\mathcal{J}(\xi_{j,k}) - \nu_k\right)^+ - \left(\mathcal{J}(\xi_{j,k}) - \nu^*(\lambda_k)\right)^+\right).\end{aligned} \tag{61}$$

From the fact that $\|\theta_k - \theta^*(\nu^*(\lambda_k),\lambda_k)\| \to 0$ almost surely as $i \to \infty$, one can conclude that the last component of the above expression vanishes, i.e., both $\|\nu_k - \nu^*(\lambda_k)\| \to 0$ and $\|\left(\mathcal{J}(\xi_{j,k}) - \nu_k\right)^+ - \left(\mathcal{J}(\xi_{j,k}) - \nu^*(\lambda_k)\right)^+\| \to 0$ almost surely. Moreover, from (41), we see that $\nabla_\lambda L(\nu,\theta,\lambda)$ is a constant function of λ. Similar to the θ-update, one can easily show that $\delta\lambda_{k+1}$ is square integrable, i.e.,

$$\mathbb{E}[\|\delta\lambda_{k+1}\|^2 \mid \mathcal{F}_{\lambda,k}] \leq 2\left(\beta + \frac{3D_{\max}}{(1-\gamma)(1-\alpha)}\right)^2,$$

where $\mathcal{F}_{\lambda,k} = \sigma\left(\lambda_m,\ \delta\lambda_m,\ m \leq k\right)$ is the filtration of λ generated by different independent trajectories. Furthermore, expression (41) implies that $\mathbb{E}\left[\delta\lambda_{k+1} \mid \mathcal{F}_{\lambda,k}\right] = 0$. Therefore, the λ-update is a stochastic approximation of the ODE (46) with a Martingale difference error term. In addition, from the convergence analysis of the (θ,ν)-update, $(\theta^*(\lambda),\nu^*(\lambda))$ is an asymptotically stable equilibrium point for the sequence $\{\theta_k,\nu_k\}$. From (39), $\nabla_\theta L(\nu,\theta,\lambda)$ is a linear mapping in λ, and $(\theta^*(\lambda),\nu^*(\lambda))$ is a Lipschitz continuous mapping of λ.

Consider the ODE for $\lambda \in [0,\lambda_{\max}]$ in (46). Analogous to the arguments for the θ-update, we can write

$$\frac{d(-L(\nu,\theta,\lambda))}{dt}\Big|_{\theta=\theta^*(\lambda),\nu=\nu^*(\lambda)} = -\nabla_\lambda L(\nu,\theta,\lambda)\Big|_{\theta=\theta^*(\lambda),\nu=\nu^*(\lambda)} \Upsilon_\lambda\left[\nabla_\lambda L(\nu,\theta,\lambda)\Big|_{\theta=\theta^*(\lambda),\nu=\nu^*(\lambda)}\right],$$

and show that $-dL(\nu,\theta,\lambda)/dt|_{\theta=\theta^*(\lambda),\nu=\nu^*(\lambda)} \leq 0$. This quantity is non-zero whenever

$$\left\|\Upsilon_\lambda\left[dL(\nu,\theta,\lambda)/d\lambda|_{\theta=\theta^*(\lambda),\nu=\nu^*(\lambda)}\right]\right\| \neq 0.$$

Consider the Lyapunov function

$$\mathcal{L}(\lambda) = -L(\theta^*(\lambda),\nu^*(\lambda),\lambda) + L(\theta^*(\lambda^*),\nu^*(\lambda^*),\lambda^*)$$

where λ^* is a local maximum point. Then there exists a ball centered at λ^* with radius r such that for any $\lambda \in \mathcal{B}_{\lambda^*}(r)$, $\mathcal{L}(\lambda)$ is a locally positive definite function, i.e., $\mathcal{L}(\lambda) \geq 0$. On the other hand, by the definition of a local maximum point, one obtains

$$\Upsilon_\lambda\left[dL(\nu,\theta,\lambda)/d\lambda|_{\theta=\theta^*(\lambda),\nu=\nu^*(\lambda),\lambda=\lambda^*}\right]|_{\lambda=\lambda^*} = 0$$

which means that λ^* is also a stationary point, i.e., $\lambda^* \in \Lambda_c$. Since

$$\frac{d\mathcal{L}(\lambda(t))}{dt} = -\frac{dL(\theta^*(\lambda(t)),\nu^*(\lambda(t)),\lambda(t))}{dt} \leq 0$$

and the time-derivative is non-zero whenever $\left\|\Upsilon_\lambda[\nabla_\lambda L(\nu,\theta,\lambda)\ |_{\nu=\nu^*(\lambda),\theta=\theta^*(\lambda)}]\right\| \neq 0$, the Lyapunov theory for asymptotically stable systems implies that $\lambda(t)$ converges to λ^*.

Given the above results and noting that the step size rule is selected according to Assumption 6, one can apply the multi-time scale stochastic approximation theory (Theorem 2 in Chapter 6 of Borkar (2008)) to show that the sequence $\{\lambda_k\}$ converges almost surely to the solution of the ODE (46), which further converges almost surely to $\lambda^* \in [0,\lambda_{\max}]$. Since $[0,\lambda_{\max}]$ is a compact set, following the same lines of arguments and recalling the envelope theorem (Theorem 16) for local optima, one further concludes that λ^* is a local maximum of $L(\theta^*(\lambda),\nu^*(\lambda),\lambda) = L^*(\lambda)$.

Step 5 (Local Saddle Point) By letting $\theta^* = \theta^*\big(\nu^*(\lambda^*), \lambda^*\big)$ and $\nu^* = \nu^*(\lambda^*)$, we will show that $(\theta^*, \nu^*, \lambda^*)$ is a local saddle point of the Lagrangian function $L(\nu, \theta, \lambda)$ if $\lambda^* \in [0, \lambda_{\max})$, and thus by the local saddle point theorem, θ^* is a locally optimal solution for the CVaR-constrained optimization.

Suppose the sequence $\{\lambda_k\}$ generated from (60) converges to a stationary point $\lambda^* \in [0, \lambda_{\max})$. Since step 3 implies that (θ^*, ν^*) is a local minimum of $L(\nu, \theta, \lambda^*)$ over the feasible set $(\theta, \nu) \in \Theta \times [-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma}]$, there exists a $r > 0$ such that

$$L(\theta^*, \nu^*, \lambda^*) \leq L(\nu, \theta, \lambda^*), \quad \forall (\theta, \nu) \in \Theta \times \left[-\frac{D_{\max}}{1-\gamma}, \frac{D_{\max}}{1-\gamma}\right] \cap \mathcal{B}_{(\theta^*, \nu^*)}(r).$$

In order to complete the proof, we must show

$$\nu^* + \frac{1}{1-\alpha}\mathbb{E}\left[\left(\mathcal{J}^{\theta^*}(x^0) - \nu^*\right)^+\right] \leq \beta, \tag{62}$$

and

$$\lambda^*\left(\nu^* + \frac{1}{1-\alpha}\mathbb{E}\left[\left(\mathcal{J}^{\theta^*}(x^0) - \nu^*\right)^+\right] - \beta\right) = 0. \tag{63}$$

These two equations imply

$$\begin{aligned} L(\theta^*, \nu^*, \lambda^*) =& V^{\theta^*}(x^0) + \lambda^*\left(\nu^* + \frac{1}{1-\alpha}\mathbb{E}\left[\left(\mathcal{J}^{\theta^*}(x^0) - \nu^*\right)^+\right] - \beta\right) \\ =& V^{\theta^*}(x^0) \\ \geq& V^{\theta^*}(x^0) + \lambda\left(\nu^* + \frac{1}{1-\alpha}\mathbb{E}\left[\left(\mathcal{J}^{\theta^*}(x^0) - \nu^*\right)^+\right] - \beta\right) = L(\theta^*, \nu^*, \lambda), \end{aligned}$$

which further implies that $(\theta^*, \nu^*, \lambda^*)$ is a saddle point of $L(\nu, \theta, \lambda)$. We now show that (62) and (63) hold.

Recall that

$$\Upsilon_\lambda\left[\nabla_\lambda L(\nu, \theta, \lambda)|_{\theta=\theta^*(\lambda), \nu=\nu^*(\lambda), \lambda=\lambda^*}\right]|_{\lambda=\lambda^*} = 0.$$

We show (62) by contradiction. Suppose

$$\nu^* + \frac{1}{1-\alpha}\mathbb{E}\left[\left(\mathcal{J}^{\theta^*}(x^0) - \nu^*\right)^+\right] > \beta.$$

This implies that for $\lambda^* \in [0, \lambda_{\max})$, we have

$$\Gamma_\Lambda\left(\lambda^* - \eta\left(\beta - \left(\nu^* + \frac{1}{1-\alpha}\mathbb{E}\left[\left(\mathcal{J}^{\theta^*}(x^0) - \nu^*\right)^+\right]\right)\right)\right) = \lambda^* - \eta\left(\beta - \left(\nu^* + \frac{1}{1-\alpha}\mathbb{E}\left[\left(\mathcal{J}^{\theta^*}(x^0) - \nu^*\right)^+\right]\right)\right)$$

for any $\eta \in (0, \eta_{\max}]$, for some sufficiently small $\eta_{\max} > 0$. Therefore,

$$\Upsilon_\lambda\left[\nabla_\lambda L(\nu, \theta, \lambda)\Big|_{\theta=\theta^*(\lambda), \nu=\nu^*(\lambda), \lambda=\lambda^*}\right]\Big|_{\lambda=\lambda^*} = \nu^* + \frac{1}{1-\alpha}\mathbb{E}\left[\left(\mathcal{J}^{\theta^*}(x^0) - \nu^*\right)^+\right] - \beta > 0.$$

This is in contradiction with the fact that $\Upsilon_\lambda\left[\nabla_\lambda L(\nu, \theta, \lambda)|_{\theta=\theta^*(\lambda), \nu=\nu^*(\lambda), \lambda=\lambda^*}\right]|_{\lambda=\lambda^*} = 0$. Therefore, (62) holds.

To show that (63) holds, we only need to show that $\lambda^* = 0$ if

$$\nu^* + \frac{1}{1-\alpha}\mathbb{E}\left[\left(\mathcal{J}^{\theta^*}(x^0) - \nu^*\right)^+\right] < \beta.$$

Suppose $\lambda^* \in (0, \lambda_{\max})$, then there exists a sufficiently small $\eta_0 > 0$ such that

$$\begin{aligned} &\frac{1}{\eta_0}\left(\Gamma_\Lambda\left(\lambda^* - \eta_0\left(\beta - \left(\nu^* + \frac{1}{1-\alpha}\mathbb{E}\left[\left(\mathcal{J}^{\theta^*}(x^0) - \nu^*\right)^+\right]\right)\right)\right) - \Gamma_\Lambda(\lambda^*)\right) \\ =& \nu^* + \frac{1}{1-\alpha}\mathbb{E}\left[\left(\mathcal{J}^{\theta^*}(x^0) - \nu^*\right)^+\right] - \beta < 0. \end{aligned}$$

This again contradicts the assumption $\Upsilon_\lambda\left[\nabla_\lambda L(\nu, \theta, \lambda)|_{\theta=\theta^*(\lambda), \nu=\nu^*(\lambda), \lambda=\lambda^*}\right]|_{\lambda=\lambda^*} = 0$. Therefore (63) holds.

When $\lambda^* = \lambda_{\max}$ and $\nu^* + \frac{1}{1-\alpha}\mathbb{E}\left[\left(\mathcal{J}^{\theta^*}(x^0) - \nu^*\right)^+\right] > \beta$,

$$\Gamma_\Lambda\left(\lambda^* - \eta\left(\beta - \left(\nu^* + \frac{1}{1-\alpha}\mathbb{E}\left[\left(\mathcal{J}^{\theta^*}(x^0) - \nu^*\right)^+\right]\right)\right)\right) = \lambda_{\max}$$

for any $\eta > 0$ and

$$\Upsilon_\lambda\left[\nabla_\lambda L(\nu, \theta, \lambda)|_{\theta=\theta^*(\lambda), \nu=\nu^*(\lambda), \lambda=\lambda^*}\right]|_{\lambda=\lambda^*} = 0.$$

In this case one cannot guarantee feasibility using the above analysis, and $(\theta^*, \nu^*, \lambda^*)$ is not a local saddle point. Such a λ^* is referred to as a spurious fixed point (see e.g., Chapter 8 of Kushner and Yin (1997)). Notice that λ^* is bounded (otherwise we can conclude that the problem is infeasible), so that by incrementally increasing $\lambda_{\max}$ in Algorithm 1, we can always prevent ourselves from obtaining a spurious fixed point solution.

Combining the above arguments, we finally conclude that $(\theta^*, \nu^*, \lambda^*)$ is a local saddle point of $L(\nu, \theta, \lambda)$. Then by the saddle point theorem, θ^* is a locally optimal policy for the CVaR-constrained optimization problem. ■

Appendix B. Convergence of Actor-Critic Algorithms

Recall from Assumption 6 that the SPSA step size $\{\Delta_k\}$ satisfies $\Delta_k \to 0$ as $k \to \infty$ and $\sum_k (\zeta_2(k)/\Delta_k)^2 < \infty$.

B.1 Gradient with Respect to λ (Proof of Lemma 11)

By taking the gradient of $V^\theta(x^0,\nu)$ w.r.t. λ (recall that both V and Q depend on λ through the cost function $\bar{C}$ of the augmented MDP $\bar{\mathcal{M}}$), we obtain

$$\begin{aligned}
\nabla_\lambda V^\theta(x^0,\nu) &= \sum_{a\in\bar{\mathcal{A}}} \mu(a|x^0,\nu;\theta)\nabla_\lambda Q^\theta(x^0,\nu,a)\\
&= \sum_{a\in\bar{\mathcal{A}}} \mu(a|x^0,\nu;\theta)\nabla_\lambda\Big[\bar{C}(x^0,\nu,a)+\sum_{(x',s')\in\bar{\mathcal{X}}}\gamma\bar{P}(x',s'|x^0,\nu,a)V^\theta(x',s')\Big]\\
&= \underbrace{\sum_{a}\mu(a|x^0,\nu;\theta)\nabla_\lambda\bar{C}(x^0,\nu,a)}_{h(x^0,\nu)}+\gamma\sum_{a,x',s'}\mu(a|x^0,\nu;\theta)\bar{P}(x',s'|x^0,\nu,a)\nabla_\lambda V^\theta(x',s')\\
&= h(x^0,\nu)+\gamma\sum_{a,x',s'}\mu(a|x^0,\nu;\theta)\bar{P}(x',s'|x^0,\nu,a)\nabla_\lambda V^\theta(x',s') \qquad (64)\\
&= h(x^0,\nu)+\gamma\sum_{a,x',s'}\mu(a|x^0,\nu;\theta)\bar{P}(x',s'|x^0,\nu,a)\Big[h(x',s')\\
&\qquad +\gamma\sum_{a',x'',s''}\mu(a'|x',s';\theta)\bar{P}(x'',s''|x',s',a')\nabla_\lambda V^\theta(x'',s'')\Big].
\end{aligned}$$

By unrolling the last equation using the definition of $\nabla_\lambda V^\theta(x,s)$ from (64), we obtain

$$\begin{aligned}
\nabla_\lambda V^\theta(x^0,\nu) &= \sum_{k=0}^{\infty}\gamma^k\sum_{x,s}\mathbb{P}(x_k=x,s_k=s\mid x_0=x^0,s_0=\nu;\theta)h(x,s)\\
&= \frac{1}{1-\gamma}\sum_{x,s}d_\gamma^\theta(x,s|x^0,\nu)h(x,s)=\frac{1}{1-\gamma}\sum_{x,s,a}d_\gamma^\theta(x,s|x^0,\nu)\mu(a|x,s)\nabla_\lambda\bar{C}(x,s,a)\\
&= \frac{1}{1-\gamma}\sum_{x,s,a}\pi_\gamma^\theta(x,s,a|x^0,\nu)\nabla_\lambda\bar{C}(x,s,a)\\
&= \frac{1}{1-\gamma}\sum_{x,s,a}\pi_\gamma^\theta(x,s,a|x^0,\nu)\frac{1}{1-\alpha}\mathbf{1}\{x=x_{\text{Tar}}\}(-s)^+.
\end{aligned}$$

This completes the proof.

B.2 Proof of Convergence of the Actor-Critic Algorithms

Before going through the details of the convergence proof, a high level overview of the proof technique is given as follows.

1. By utilizing temporal difference techniques, we show the critic update converges (in the fastest time scale) almost surely to a fixed point solution v^* of the projected form of Bellman's equation, which is defined on the augmented MDP $\bar{\mathcal{M}}$.

2. Similar to the analysis of the policy gradient algorithm, by convergence properties of multi-time scale discrete stochastic approximation algorithms, we show that each update $(\nu_k,\theta_k,\lambda_k)$ converges almost surely to a stationary point $(\nu^*,\theta^*,\lambda^*)$ of the corresponding continuous time system. In particular, by adopting the step-size rules defined in Assumption 8, we show that the convergence rate of v is fastest, followed by the convergence rate of ν and the convergence rate of θ, while the convergence rate of λ is the slowest among the set of parameters. Different from the policy gradient algorithm, the parameters of the actor-critic algorithm are updated incrementally. To adjust for this difference in the convergence analysis, modifications to the gradient estimate of ν are introduced, either via the SPSA method or the semi-trajectory method, to ensure the gradient estimates are unbiased. Following from the arguments of Lyapunov analysis, we prove that the continuous time system is locally asymptotically stable at the stationary point $(\nu^*,\theta^*,\lambda^*)$.

3. Following the same line of arguments from the proof of the policy gradient algorithm, we conclude that the stationary point $(v^*,\nu^*,\theta^*,\lambda^*)$ is a local saddle point. Finally, by the the local saddle point theorem, we deduce that θ^* is a locally optimal solution for the CVaR-constrained MDP problem.

This convergence proof procedure is rather standard for stochastic approximation algorithms, see (Bhatnagar et al., 2009; Bhatnagar and Lakshmanan, 2012) for further references.

B.2.1 PROOF OF THEOREM 10: CRITIC UPDATE (v-UPDATE)

By the step size conditions, one notices that $\{v_k\}$ converges on a faster time scale than $\{\nu_k\}$, $\{\theta_k\}$, and $\{\lambda_k\}$. According to Lemma 1 in Chapter 6 of Borkar (2008), one can take (ν,θ,λ) in the v-update as arbitrarily fixed quantities (in this case we have $(\nu,\theta,\lambda)=(\nu_k,\theta_k,\lambda_k)$). Thus the critic update can be re-written as follows:

$$v_{k+1}=v_k+\zeta_4(k)\phi(x_k,s_k)\delta_k(v_k), \qquad (65)$$

where the scalar

$$\delta_k(v_k)=-v_k^\top\phi(x_k,s_k)+\gamma v_k^\top\phi(x_{k+1},s_{k+1})+\bar{C}_\lambda(x_k,s_k,a_k)$$

is the temporal difference (TD) from (18). Define

$$A:=\sum_{y,a',s'}\pi_\gamma^\theta(y,s',a'|x,s)\phi(y,s')\left(\phi^\top(y,s')-\gamma\sum_{z,s''}\bar{P}(z,s''|y,s',a)\phi^\top(z,s'')\right), \qquad (66)$$

and

$$b:=\sum_{y,a',s'}\pi_\gamma^\theta(y,s',a'|x,s)\phi(y,s')\bar{C}_\lambda(y,s',a'). \qquad (67)$$

It is easy to see that the critic update v_k in (65) can be re-written as the following stochastic approximation scheme:

$$v_{k+1}=v_k+\zeta_4(k)(b-Av_k+\delta A_{k+1}), \qquad (68)$$

where the noise term δA_{k+1} is a square integrable Martingale difference, i.e., $\mathbb{E}[\delta A_{k+1}\mid\mathcal{F}_k]=0$ if the γ-occupation measure π_γ^θ is used to generate samples of $(x_k,s_k,a_k)$with $\mathcal{F}_k$ being the filtration generated by different independent trajectories. By writing

$$\delta A_{k+1}=-(b-Av_k)+\phi(x_k,s_k)\delta_k(v_k)$$

and noting $\mathbb{E}_{\pi_\gamma^\theta}[\phi(x_k,s_k)\delta_k(v_k) \mid \mathcal{F}_k] = -Av_k + b$, one can easily verify that the stochastic approximation scheme in (68) is equivalent to the critic iterates in (65) and δA_{k+1} is a Martingale difference, i.e., $\mathbb{E}_{\pi_\gamma^\theta}[\delta A_{k+1} \mid \mathcal{F}_k] = 0$. Let

$$h(v) := -Av + b.$$

Before getting into the convergence analysis, we present a technical lemma whose proof can be found in Lemma 6.10 of Bertsekas and Tsitsiklis (1996).

Lemma 20 *Every eigenvalue of the matrix A has positive real part.*

We now turn to the analysis of the critic iteration. Note that the following properties hold for the critic update scheme in (65): 1) $h(v)$ is Lipschitz, 2) the step size satisfies the properties in Assumption 8, 3) the noise term δA_{k+1} is a square integrable Martingale difference, 4) the function $h_c(v) := h(cv)/c$, $c \geq 1$ converges uniformly to a continuous function $h_\infty(v)$ for any v in a compact set, i.e., $h_c(v) \to h_\infty(v)$ as $c \to \infty$, and 5) the ordinary differential equation (ODE) $\dot{v} = h_\infty(v)$ has the origin as its unique locally asymptotically stable equilibrium. The fourth property can be easily verified from the fact that the magnitude of b is finite and $h_\infty(v) = -Av$. The fifth property follows directly from the facts that $h_\infty(v) = -Av$ and all eigenvalues of A have positive real parts.

By Theorem 3.1 in Borkar (2008), these five properties imply:

The critic iterates $\{v_k\}$ are bounded almost surely, i.e., $\sup_k \|v_k\| < \infty$ almost surely.

The convergence of the critic iterates in (65) can be related to the asymptotic behavior of the ODE

$$\dot{v} = h(v) = b - Av. \tag{69}$$

Specifically, Theorem 2 in Chapter 2 of Borkar (2008) and the above conditions imply $v_k \to v^*$ with probability 1, where the limit v^* depends on (ν,θ,λ) and is the unique solution satisfying $h(v^*) = 0$, i.e., $Av^* = b$. Therefore, the critic iterates converge to the unique fixed point v^* almost surely, as $k \to \infty$.

B.2.2 PROOF OF THEOREM 12

Step 1 (Convergence of v-update) The proof of convergence for the critic parameter follows directly from Theorem 10.

Step 2 (Convergence of SPSA Based ν-update) In this section, we analyze the ν-update for the incremental actor-critic method. This update is based on the SPSA perturbation method. The idea of this method is to estimate the sub-gradient $g(\nu) \in \partial_\nu L(\nu,\theta,\lambda)$ using two simulated value functions corresponding to $\nu^- = \nu - \Delta$ and $\nu^+ = \nu + \Delta$. Here $\Delta \geq 0$ is a positive random perturbation that vanishes asymptotically. The SPSA-based estimate for a sub-gradient $g(\nu) \in \partial_\nu L(\nu,\theta,\lambda)$ is given by

$$g(\nu) \approx \lambda + \frac{1}{2\Delta}\left(\phi^\top\left(x^0,\nu+\Delta\right) - \phi^\top\left(x^0,\nu-\Delta\right)\right)v.$$

First, we consider the following assumption on the feature functions in order to prove that the SPSA approximation is asymptotically unbiased.

Assumption 21 *For any $v \in \mathbb{R}^{\kappa_1}$, the feature functions satisfy the following conditions*

$$|\phi^\top\left(x^0,\nu+\Delta\right)v - \phi^\top\left(x^0,\nu-\Delta\right)v| \leq K_1(v)(1+\Delta).$$

Furthermore, the Lipschitz constants are uniformly bounded, i.e., $\sup_{v\in\mathbb{R}^{\kappa_1}} K_1^2(v) < \infty$.

This assumption is mild as the expected utility objective function implies that $L(\nu,\theta,\lambda)$ is Lipschitz in ν, and $\phi_V^\top\left(x^0,\nu\right)v$ is just a linear function approximation of $V^\theta(x^0,\nu)$.

Next, we turn to the convergence analysis of the sub-gradient estimation and ν-update. Since ν converges faster than θ and λ. Consider the ν-update in (20):

$$\nu_{k+1} = \Gamma_{\mathcal{N}}\left(\nu_k - \zeta_3(k)\left(\lambda + \frac{1}{2\Delta_k}\left(\phi^\top\left(x^0,\nu_k+\Delta_k\right) - \phi^\top\left(x^0,\nu_k-\Delta_k\right)\right)v_k\right)\right), \tag{70}$$

where according to Lemma 1 in Chapter 6 of Borkar (2008), (θ_k,λ_k) in this expression are viewed as constant quantities. Since v converges faster than ν, Lemma 1 in Chapter 6 of Borkar (2008) also implies $\|v_k - v^*(\nu_k)\| \to 0$ almost surely, where $v^*(\nu)$ is the converged critic parameter. Together with the above assumption that the feature function is bounded, one can rewrite the ν-update in (20) as follows:

$$\nu_{k+1} = \Gamma_{\mathcal{N}}\left(\nu_k - \zeta_3(k)\left(\lambda + \frac{1}{2\Delta_k}\left(\phi^\top\left(x^0,\nu_k+\Delta_k\right) - \phi^\top\left(x^0,\nu_k-\Delta_k\right)\right)v^*(\nu_k) + \epsilon_k\right)\right), \tag{71}$$

where

$$\epsilon_k = \frac{1}{2\Delta_k}\left(\phi^\top\left(x^0,\nu_k+\Delta_k\right) - \phi^\top\left(x^0,\nu_k-\Delta_k\right)\right)(v_k - v^*(\nu_k)) \to 0, \quad \text{almost surely.}$$

Equipped with this intermediate result, we establish the bias and convergence of the stochastic sub-gradient estimate. Let

$$\overline{g}(\nu_k) \in \arg\max\left\{g : g \in \partial_\nu L(\nu,\theta,\lambda)|_{\nu=\nu_k}\right\},$$

and

$$\begin{aligned}
\Lambda_{1,k+1} =& \left(\frac{\left(\phi^\top\left(x^0,\nu_k+\Delta_k\right) - \phi^\top\left(x^0,\nu_k-\Delta_k\right)\right)v^*(\nu_k)}{2\Delta_k} - E_M(k)\right),\\
\Lambda_{2,k} =& \lambda_k + E_M^L(k) - \overline{g}(\nu_k),\\
\Lambda_{3,k} =& E_M(k) - E_M^L(k),
\end{aligned}$$

where

$$\begin{aligned}
E_M(k) :=& \mathbb{E}\left[\frac{1}{2\Delta_k}\left(\phi^\top\left(x^0,\nu_k+\Delta_k\right) - \phi^\top\left(x^0,\nu_k-\Delta_k\right)\right)v^*(\nu_k) \mid \Delta_k\right],\\
E_M^L(k) :=& \mathbb{E}\left[\frac{1}{2\Delta_k}\left(V^\theta\left(x^0,\nu_k+\Delta_k\right) - V^\theta\left(x^0,\nu_k-\Delta_k\right)\right) \mid \Delta_k\right].
\end{aligned}$$

Note that (71) is equivalent to

$$\nu_{k+1} = \Gamma_{\mathcal{N}}\left(\nu_k - \zeta_3(k)\left(\overline{g}(\nu_k) + \Lambda_{1,k+1} + \Lambda_{2,k} + \Lambda_{3,k}\right)\right). \tag{72}$$

First, it is clear that $\Lambda_{1,k+1}$ is a Martingale difference as $\mathbb{E}[\Lambda_{1,k+1} \mid \mathcal{F}_k] = 0$, which implies that

$$M_{k+1} = \sum_{j=0}^{k} \zeta_3(j)\Lambda_{1,j+1}$$

is a Martingale w.r.t. the filtration $\mathcal{F}_k$. By the Martingale convergence theorem, we can show that if $\sup_{k\geq 0}\mathbb{E}[M_k^2] < \infty$, when $k \to \infty$, M_k converges almost surely and $\zeta_3(k)\Lambda_{1,k+1} \to 0$ almost surely. To show that $\sup_{k\geq 0}\mathbb{E}[M_k^2] < \infty$, for any $t \geq 0$ one observes that

$$\begin{aligned}\mathbb{E}[M_{k+1}^2] &= \sum_{j=0}^{k} (\zeta_3(j))^2\, \mathbb{E}[\mathbb{E}[\Lambda_{1,j+1}^2 \mid \Delta_j]] \\ &\leq 2\sum_{j=0}^{k} \mathbb{E}\Bigg[\left(\frac{\zeta_3(j)}{2\Delta_j}\right)^2 \Bigg\{\mathbb{E}\left[\left(\left(\phi^\top\left(x^0, \nu_j + \Delta_j\right) - \phi^\top\left(x^0, \nu_j - \Delta_j\right)\right)v^*(\nu_j)\right)^2 \mid \Delta_j\right] \\ &\qquad + \mathbb{E}\left[\left(\phi^\top\left(x^0, \nu_j + \Delta_j\right) - \phi^\top\left(x^0, \nu_j - \Delta_j\right)\right)v^*(\nu_j) \mid \Delta_j\right]^2\Bigg\}\Bigg].\end{aligned}$$

Now based on Assumption 21, the above expression implies

$$\mathbb{E}[M_{k+1}^2] \leq 2\sum_{j=0}^{k} \mathbb{E}\left[\left(\frac{\zeta_3(j)}{2\Delta_j}\right)^2 2K_1^2(1+\Delta_j)^2\right].$$

Combining the above results with the step size conditions, there exists $K = 4K_1^2 > 0$ such that

$$\sup_{k\geq 0}\mathbb{E}[M_{k+1}^2] \leq K\sum_{j=0}^{\infty} \mathbb{E}\left[\left(\frac{\zeta_3(j)}{2\Delta_j}\right)^2\right] + (\zeta_3(j))^2 < \infty.$$

Second, by the Min Common/Max Crossing theorem in Chapter 5 of Bertsekas (2009), one can show that $\partial_\nu L(\nu,\theta,\lambda)|_{\nu=\nu_k}$ is a non-empty, convex, and compact set. Therefore, by duality of directional directives and sub-differentials, i.e.,

$$\max\{g : g \in \partial_\nu L(\nu,\theta,\lambda)|_{\nu=\nu_k}\} = \lim_{\xi\downarrow 0}\frac{L(\nu_k+\xi,\theta,\lambda) - L(\nu_k-\xi,\theta,\lambda)}{2\xi},$$

one concludes that for $\lambda_k = \lambda$ (we can treat λ_k as a constant because it converges on a slower time scale than ν_k),

$$\lambda + E_M^L(k) = \overline{g}(\nu_k) + O(\Delta_k),$$

almost surely. This further implies that

$$\Lambda_{2,k} = O(\Delta_k), \quad \text{i.e.,} \quad \Lambda_{2,k} \to 0 \quad \text{as} \quad k \to \infty,$$

almost surely.

Third, since $d_\gamma^\theta(x^0, \nu | x^0, \nu) = 1$, from the definition of $\epsilon_\theta(v^*(\nu_k))$,

$$|\Lambda_{3,k}| \leq 2\epsilon_\theta(v^*(\nu_k))/\Delta_k.$$

As t goes to infinity, $\epsilon_\theta(v^*(\nu_k))/\Delta_k \to 0$ by assumption and $\Lambda_{3,k} \to 0$.

Finally, since $\zeta_3(k)\Lambda_{1,k+1} \to 0$, $\Lambda_{2,k} \to 0$, and $\Lambda_{3,k} \to 0$ almost surely, the ν-update in (72) is a noisy sub-gradient descent update with vanishing disturbance bias. Thus, the ν-update in (20) can be viewed as an Euler discretization of an element of the following differential inclusion,

$$\dot{\nu} \in \Upsilon_\nu\left[-g(\nu)\right], \qquad \forall g(\nu) \in \partial_\nu L(\nu,\theta,\lambda), \tag{73}$$

and the ν-convergence analysis is analogous to Step 1 of the proof of Theorem 7.

Step 2′ (Convergence of Semi-trajectory ν-update) Since ν converges on a faster timescale than θ and λ, according to Lemma 1 in Chapter 6 of Borkar (2008), the convergence property of ν in (23) can be shown for any arbitrarily fixed pair of (θ,λ) (in this case we have $(\theta,\lambda) = (\theta_k,\lambda_k)$), i.e.,

$$\nu_{k+1} = \Gamma_{\mathcal{N}}\left(\nu_k - \zeta_3(k)\left(\lambda - \frac{\lambda}{1-\alpha}\left(\mathbb{P}\left(s_{\text{Tar}} \leq 0 \mid x_0 = x^0, s_0 = \nu_k, \theta\right) + \delta\nu_{M,k+1}\right)\right)\right), \tag{74}$$

where

$$\delta\nu_{M,k+1} = -\mathbb{P}\left(s_{\text{Tar}} \leq 0 \mid x_0 = x^0, s_0 = \nu_k, \mu\right) + \mathbf{1}\left\{x_k = x_{\text{Tar}}, s_k \leq 0\right\} \tag{75}$$

is a square integrable stochastic term, specifically,

$$\mathbb{E}[(\delta\nu_{M,k+1})^2 \mid \mathcal{F}_{\nu,k}] \leq 2,$$

where $\mathcal{F}_{\nu,k} = \sigma(\nu_m,\, \delta\nu_m,\, m \leq k)$ is the filtration generated by ν. Since $\mathbb{E}\left[\delta\nu_{M,k+1} \mid \mathcal{F}_{\nu,k}\right] = 0$, $\delta\nu_{M,k+1}$ is a Martingale difference and the ν-update in (74) is a stochastic approximation of an element of the differential inclusion

$$\frac{\lambda}{1-\alpha}\mathbb{P}\left(s_{\text{Tar}} \leq 0 \mid x_0 = x^0, s_0 = \nu_k, \theta\right) - \lambda \in -\partial_\nu L(\nu,\theta,\lambda)|_{\nu=\nu_k}.$$

Thus, the ν-update in (23) can be viewed as an Euler discretization of the differential inclusion in (73), and the ν-convergence analysis is analogous to Step 1 of the proof of Theorem 7.

Step 3 (Convergence of θ-update) We first analyze the actor update (θ-update). Since θ converges on a faster time scale than λ, according to Lemma 1 in Chapter 6 of Borkar (2008), one can take λ in the θ-update as a fixed quantity (i.e., here we have that $\lambda = \lambda_k$). Furthermore, since v and ν converge on a faster scale than θ, one also have $\|v_k - v^*(\theta_k)\| \to 0$, $\|\nu_k - \nu^*(\theta_k)\| \to 0$ almost surely, and since convergence almost surely of the ν sequence implies convergence in distribution, we have $\|\pi_\gamma^{\theta_k}(x', s', a' | x_0 = x^0, s_0 = \nu_k) - \pi_\gamma^{\theta_k}(x', s', a' | x_0 = x^0, s_0 = \nu^*(\theta_k))\| \to 0$. In the following analysis, we assume that the initial state $x^0 \in \mathcal{X}$ is given. Consider the θ-update in (21)

$$\theta_{k+1} = \Gamma_\Theta\left(\theta_k - \zeta_2(k)\left(\nabla_\theta \log\mu(a_k|x_k,s_k;\theta)|_{\theta=\theta_k}\frac{\delta_k(v_k)}{1-\gamma}\right)\right). \tag{76}$$

Utilizing the above convergence properties, (76) can be rewritten as follows:

$$\theta_{k+1} = \Gamma_\Theta\left(\theta_k - \zeta_2(k)\left(\nabla_\theta \log\mu(a_k|x_k,s_k;\theta)|_{\theta=\theta_k}\left(\frac{\delta_k(v^*(\theta_k))}{1-\gamma} + \epsilon_k\right)\right)\right),$$

where we showed in the convergence analysis of the ν sequence that

$$\epsilon_k = \frac{\delta_k(v_k)}{1-\gamma} - \frac{\delta_k(v^*(\theta_k))}{1-\gamma} \to 0, \quad \text{almost surely.}$$

Consider the case in which the value function for a fixed policy θ (i.e., $\theta = \theta_k$) is approximated by a learned function approximation, $\phi^\top(x,s)v^*(\theta_k)$. If the approximation is sufficiently good, we might hope to use it in place of $V^\theta(x,s)$ and still point roughly in the direction of the true gradient. Recall the temporal difference error (random variable) for a given pair $(x_k, s_k) \in \mathcal{X} \times \mathbb{R}$:

$$\delta_k(v) = -v^\top \phi(x_k, s_k) + \gamma v^\top \phi(x_{k+1}, s_{k+1}) + \bar{C}_\lambda(x_k, s_k, a_k).$$

Define the v-dependent approximated advantage function

$$\tilde{A}^{\theta,v}(x,s,a) := \tilde{Q}^{\theta,v}(x,s,a) - v^\top \phi(x,s),$$

where

$$\tilde{Q}^{\theta,v}(x,s,a) = \gamma \sum_{x',s'} \bar{P}(x',s'|x,s,a) v^\top \phi(x',s') + \bar{C}_\lambda(x,s,a).$$

The following lemma, whose proof follows from the proof of Lemma 3 in Bhatnagar et al. (2009), shows that $\delta_k(v)$ is an unbiased estimator of $\tilde{A}^{\theta,v}$.

Lemma 22 *For any given policy μ and $v \in \mathbb{R}^{\kappa_1}$, we have*

$$\tilde{A}^{\theta,v}(x,s,a) = \mathbb{E}[\delta_k(v) \mid x_k = x, s_k = s, a_k = a].$$

Define

$$\nabla_\theta \tilde{L}_v(\nu,\theta,\lambda) := \frac{1}{1-\gamma} \sum_{x,a,s} \pi_\gamma^\theta(x,s,a|x_0 = x^0, s_0 = \nu) \nabla_\theta \log \mu(a|x,s;\theta) \tilde{A}^{\theta,v}(x,s,a)$$

as the linear function approximation of $\nabla_\theta \tilde{L}(\nu,\theta,\lambda)$. Similar to Proposition 17, we present the following technical lemma on the Lipschitz property of $\nabla_\theta \tilde{L}_v(\nu,\theta,\lambda)$.

Proposition 23 *$\nabla_\theta \tilde{L}_v(\nu,\theta,\lambda)$ is a Lipschitz function in θ.*

Proof. Consider the feature vector v. Recall that the feature vector satisfies the linear equation $Av = b$, where A and b are given by (66) and (67), respectively. From Lemma 1 in Bhatnagar and Lakshmanan (2012), by exploiting the inverse of A using Cramer's rule, one may show that v is continuously differentiable in θ. Now consider the γ-occupation measure π_γ^θ. By applying Theorem 2 in Altman et al. (2004) (or Theorem 3.1 in Shardlow and Stuart (2000)), it can be seen that the occupation measure π_γ^θ of the process (x_k, s_k) is continuously differentiable in θ. Recall from Assumption 3 in Section 2.2 that $\nabla_\theta \mu(a_k|x_k,s_k;\theta)$ is a Lipschitz function in θ for any $a \in \mathcal{A}$ and $k \in \{0,\ldots,T-1\}$, and $\mu(a_k|x_k,s_k;\theta)$ is differentiable in θ. By combining these arguments and noting that the sum of products of Lipschitz functions is Lipschitz, one concludes that $\nabla_\theta \tilde{L}_v(\nu,\theta,\lambda)$ is Lipschitz in θ. ■

We turn to the convergence proof of θ.

Theorem 24 *The sequence of θ-updates in* (21) *converges almost surely to an equilibrium point $\widehat{\theta}^*$ that satisfies $\Upsilon_\theta\left[-\nabla_\theta \tilde{L}_{v^*(\theta)}(\nu^*(\theta),\theta,\lambda)\right] = 0$, for a given $\lambda \in [0,\lambda_{\max}]$. Furthermore, if the function approximation error $\epsilon_\theta(v_k)$ vanishes as the feature vector v_k converges to v^*, then the sequence of θ-updates converges to θ^* almost surely, where θ^* is a local minimum point of $L(\nu^*(\theta),\theta,\lambda)$ for a given $\lambda \in [0,\lambda_{\max}]$.*

Proof. We will mainly focus on proving the convergence of $\theta_k \to \theta^*$ (second part of the theorem). Since we just showed in Proposition 23 that $\nabla_\theta \tilde{L}_{v^*(\theta)}(\nu^*(\theta),\theta,\lambda)$ is Lipschitz in θ, the convergence proof of $\theta_k \to \widehat{\theta}^*$ (first part of the theorem) follows from identical arguments.

Note that the θ-update in (76) can be rewritten as:

$$\theta_{k+1} = \Gamma_\Theta\left(\theta_k + \zeta_2(k)\left(-\nabla_\theta L(\nu,\theta,\lambda)|_{\nu=\nu^*(\theta),\theta=\theta_k} + \delta\theta_{k+1} + \delta\theta_\epsilon\right)\right),$$

where

$$\delta\theta_{k+1} = \sum_{x',a',s'} \pi_\gamma^{\theta_k}(x',s',a'|x_0 = x^0, s_0 = \nu^*(\theta_k)) \nabla_\theta \log \mu(a'|x',s';\theta)|_{\theta=\theta_k} \frac{\tilde{A}^{\theta_k,v^*(\theta_k)}(x',s',a')}{1-\gamma}$$
$$- \nabla_\theta \log \mu(a_k|x_k,s_k;\theta)|_{\theta=\theta_k} \frac{\delta_k(v^*(\theta_k))}{1-\gamma}.$$

and

$$\delta\theta_\epsilon = \sum_{x',a',s'} \pi_\gamma^{\theta_k}(x',s',a'|x_0 = x^0, s_0 = \nu^*(\theta_k)) \cdot$$
$$\frac{\nabla_\theta \log \mu(a'|x',s';\theta)|_{\theta=\theta_k}}{1-\gamma} (A^{\theta_k}(x',s',a') - \tilde{A}^{\theta_k,v^*(\theta_k)}(x',s',a'))$$

First, one can show that $\delta\theta_{k+1}$ is square integrable, specifically,

$$\begin{aligned}
&\mathbb{E}[\|\delta\theta_{k+1}\|^2 \mid \mathcal{F}_{\theta,k}] \\
&\leq \frac{2}{1-\gamma} \|\nabla_\theta \log \mu(u|x,s;\theta)|_{\theta=\theta_k}\, \mathbf{1}\{\mu(u|x,s;\theta_k) > 0\}\, \|_\infty^2 \left(\|\tilde{A}^{\theta_k,v^*(\theta_k)}(x,s,a)\|_\infty^2 + |\delta_k(v^*(\theta_k))|^2\right) \\
&\leq \frac{2}{1-\gamma} \cdot \frac{\|\nabla_\theta \mu(u|x,s;\theta)|_{\theta=\theta_k}\|_\infty^2}{\min\{\mu(u|x,s;\theta_k) \mid \mu(u|x,s;\theta_k) > 0\}^2} \left(\|\tilde{A}^{\theta_k,v^*(\theta_k)}(x,s,a)\|_\infty^2 + |\delta_k(v^*(\theta_k))|^2\right) \\
&\leq 64 \frac{K^2\|\theta_k\|^2}{1-\gamma} \left(\max_{x,s,a} |\bar{C}_\lambda(x,s,a)|^2 + 2 \max_{x,s} \|\phi(x,s)\|^2 \sup_k \|v_k\|^2\right) \\
&\leq 64 \frac{K^2\|\theta_k\|^2}{1-\gamma} \left(\left|\max\left\{C_{\max}, \frac{2\lambda D_{\max}}{\gamma^T(1-\alpha)(1-\gamma)}\right\}\right|^2 + 2 \max_{x,s} \|\phi(x,s)\|^2 \sup_k \|v_k\|^2\right),
\end{aligned}$$

for some Lipschitz constant K, where the indicator function in the second line can be explained by the fact that $\pi_\gamma^{\theta_k}(x,s,u) = 0$ whenever $\mu(u \mid x,s;\theta_k) = 0$ and because the expectation is taken with respect to $\pi_\gamma^{\theta_k}$. The third inequality uses Assumption 3 and the fact that μ takes on finitely-many values (and thus its nonzero values are bounded away from zero). Finally, $\sup_k \|v_k\| < \infty$ follows from the Lyapunov analysis in the critic update.

Second, note that

$$\delta\theta_\epsilon \leq \frac{(1+\gamma)\|\psi_{\theta_k}\|_\infty}{(1-\gamma)^2} \epsilon_{\theta_k}(v^*(\theta_k)), \tag{77}$$

where $\psi_\theta(x,s,a) = \nabla_\theta \log \mu(a|x,s;\theta)$ is the "compatible feature." The last inequality is due to the fact that since π_γ^θ is a probability measure, convexity of quadratic functions implies

$$
\begin{aligned}
&\sum_{x',a',s'} \pi_\gamma^\theta(x',s',a'|x_0=x^0,s_0=\nu^*(\theta))(A^\theta(x',s',a') - \tilde{A}^{\theta,v}(x',s',a')) \\
\leq &\sum_{x',a',s'} \pi_\gamma^\theta(x',s',a'|x_0=x^0,s_0=\nu^*(\theta))(Q^\theta(x',s',a') - \tilde{Q}^{\theta,v}(x',s',a')) \\
&+ \sum_{x',s'} d_\gamma^\theta(x',s'|x_0=x^0,s_0=\nu^*(\theta))(V^\theta(x',s') - \widetilde{V}^{\theta,v}(x',s')) \\
= &\gamma \sum_{x',a',s'} \pi_\gamma^\theta(x',s',a'|x_0=x^0,s_0=\nu^*(\theta)) \sum_{x'',s''} \bar{P}(x'',s''|x',s',a')(V^\theta(x'',s'') - \phi^\top(x'',s'')v) \\
&+ \sqrt{\sum_{x',s'} d_\gamma^\theta(x',s'|x_0=x^0,s_0=\nu^*(\theta))(V^\theta(x',s') - \widetilde{V}^{\theta,v}(x',s'))^2} \\
\leq &\gamma \sqrt{\sum_{x',a',s'} \pi_\gamma^\theta(x',s',a'|x_0=x^0,s_0=\nu^*(\theta)) \sum_{x'',s''} \bar{P}(x'',s''|x',s',a')(V^\theta(x'',s'') - \phi^\top(x'',s'')v)^2} \\
&+ \frac{\epsilon_\theta(v)}{1-\gamma} \\
\leq &\sqrt{\sum_{x'',s''} \left(d_\gamma^\theta(x'',s''|x^0,\nu^*(\theta)) - (1-\gamma)1\{x^0=x'',\nu=s''\}\right)(V^\theta(x'',s'') - \phi^\top(x'',s'')v)^2} + \frac{\epsilon_\theta(v)}{1-\gamma} \\
\leq &\left(\frac{1+\gamma}{1-\gamma}\right)\epsilon_\theta(v).
\end{aligned}
$$

Then by Lemma 22, if the γ-occupation measure π_γ^θ is used to generate samples (x_k, s_k, a_k), one obtains $\mathbb{E}\left[\delta\theta_{k+1} \mid \mathcal{F}_{\theta,k}\right] = 0$, where $\mathcal{F}_{\theta,k} = \sigma(\theta_m,\, \delta\theta_m,\, m \leq k)$ is the filtration generated by different independent trajectories. On the other hand, $|\delta\theta_\epsilon| \to 0$ as $\epsilon_{\theta_k}(v^*(\theta_k)) \to 0$. Therefore, the θ-update in (76) is a stochastic approximation of the continuous system $\theta(t)$, described by the ODE

$$\dot{\theta} = \Upsilon_\theta\left[-\nabla_\theta L(\nu,\theta,\lambda)|_{\nu=\nu^*(\theta)}\right],$$

with an error term that is a sum of a vanishing bias and a Martingale difference. Thus, the convergence analysis of θ follows analogously from Step 2 in the proof of Theorem 7, i.e., the sequence of θ-updates in (21) converges to θ^* almost surely, where θ^* is the equilibrium point of the continuous system θ satisfying

$$\Upsilon_\theta\left[-\nabla_\theta L(\nu,\theta,\lambda)|_{\nu=\nu^*(\theta)}\right] = 0. \tag{78}$$

■

Step 4 (Local Minimum) The proof that (θ^*,ν^*) is a local minimum follows directly from the arguments in Step 3 in the proof of Theorem 7.

Step 5 (λ-update and Convergence to Local Saddle Point) Note that the λ-update converges on the slowest time scale, according to previous analysis, we have that $\|\theta_k - \theta^*(\lambda_k)\| \to 0$, $\|\nu_k - \nu^*(\lambda_k)\| \to 0$ almost surely. By continuity of $\nabla_\lambda L(\nu,\theta,\lambda)$, we also have the following:

$$\left\| \nabla_\lambda L(\nu,\theta,\lambda)\Big|_{\theta=\theta^*(\lambda_k),\nu=\nu^*(\lambda_k),\lambda=\lambda_k} - \nabla_\lambda L(\nu,\theta,\lambda)\Big|_{\theta=\theta_k,\nu=\nu_k,\lambda=\lambda_k} \right\| \to 0.$$

Thus, (20) may be rewritten as

$$\lambda_{k+1} = \Gamma_\Lambda\left(\lambda_k + \zeta_1(k)\left(\nabla_\lambda L(\nu,\theta,\lambda)\Big|_{\theta=\theta^*(\lambda),\nu=\nu^*(\lambda),\lambda=\lambda_k} + \delta\lambda_{k+1}\right)\right), \tag{79}$$

where

$$
\begin{aligned}
\delta\lambda_{k+1} = &-\nabla_\lambda L(\nu,\theta,\lambda)\Big|_{\theta=\theta^*(\lambda),\nu=\nu^*(\lambda),\lambda=\lambda_k} + \\
&\left(\underbrace{(\nu_k - \nu^*(\lambda_k))}_{\|\nu_k-\nu^*(\lambda_k)\|\to 0} + \nu^*(\lambda_k) + \frac{(-s_k)^+}{(1-\alpha)(1-\gamma)}\mathbf{1}\{x_k = x_{\text{Tar}}\} - \beta\right).
\end{aligned} \tag{80}
$$

From (41), $\nabla_\lambda L(\nu,\theta,\lambda)$ does not depend on λ. Similar to the θ-update, one can easily show that $\delta\lambda_{k+1}$ is square integrable, specifically,

$$\mathbb{E}[\|\delta\lambda_{k+1}\|^2 \mid \mathcal{F}_{\lambda,k}] \leq 8\left(\beta^2 + \left(\frac{D_{\max}}{1-\gamma}\right)^2 + \left(\frac{2D_{\max}}{(1-\gamma)^2(1-\alpha)}\right)^2\right),$$

where $\mathcal{F}_{\lambda,k} = \sigma\big(\lambda_m,\, \delta\lambda_m,\, m \leq k\big)$ is the filtration of λ generated by different independent trajectories. Similar to the θ-update, using the γ-occupation measure π_γ^θ, one obtains $\mathbb{E}\left[\delta\lambda_{k+1} \mid \mathcal{F}_{\lambda,k}\right] = 0$. As above, the λ-update is a stochastic approximation for the continuous system $\lambda(t)$ described by the ODE

$$\dot{\lambda} = \Upsilon_\lambda\left[\nabla_\lambda L(\nu,\theta,\lambda)\Big|_{\theta=\theta^*(\lambda),\nu=\nu^*(\lambda)}\right],$$

with an error term that is a Martingale difference. Then the λ-convergence and the analysis of local optima follow from analogous arguments in Steps 4 and 5 in the proof of Theorem 7.

References

E. Altman. *Constrained Markov decision processes*, volume 7. CRC Press, 1999.

E. Altman, K. Avrachenkov, and R. Núñez-Queija. Perturbation analysis for denumerable Markov chains with application to queueing models. *Advances in Applied Probability*, pages 839–853, 2004.

P. Artzner, F. Delbaen, J. Eber, and D. Heath. Coherent measures of risk. *Journal of Mathematical Finance*, 9(3):203–228, 1999.

O. Bardou, N. Frikha, and G. Pagès. Computing VaR and CVaR using stochastic approximation and adaptive unconstrained importance sampling. *Monte Carlo Methods and Applications*, 15(3):173–210, 2009.

N. Bäuerle and A. Mundt. Dynamic mean-risk optimization in a binomial model. *Mathematical Methods of Operations Research*, 70(2):219–239, 2009.

N. Bäuerle and J. Ott. Markov decision processes with average-value-at-risk criteria. *Mathematical Methods of Operations Research*, 74(3):361–379, 2011.

J. Baxter and P. Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001.

M. Benaim, J. Hofbauer, and S. Sorin. Stochastic approximations and differential inclusions, Part II: Applications. *Mathematics of Operations Research*, 31(4):673–695, 2006.

D. Bertsekas. *Dynamic programming and optimal control*. Athena Scientific, 1995.

D. Bertsekas. *Nonlinear programming*. Athena Scientific, 1999.

D. Bertsekas. Min common/max crossing duality: A geometric view of conjugacy in convex optimization. *Lab. for Information and Decision Systems, MIT, Tech. Rep. Report LIDS-P-2796*, 2009.

D. Bertsekas and J. Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, 1996.

S. Bhatnagar. An actor-critic algorithm with function approximation for discounted cost constrained Markov decision processes. *Systems & Control Letters*, 59(12):760–766, 2010.

S. Bhatnagar and K. Lakshmanan. An online actor-critic algorithm with function approximation for constrained Markov decision processes. *Journal of Optimization Theory and Applications*, 153(3):688–708, 2012.

S. Bhatnagar, R. Sutton, M. Ghavamzadeh, and M. Lee. Natural actor-critic algorithms. *Automatica*, 45(11): 2471–2482, 2009.

S. Bhatnagar, H. Prasad, and L. Prashanth. *Stochastic recursive algorithms for optimization*, volume 434. Springer, 2013.

K. Boda and J. Filar. Time consistent dynamic risk measures. *Mathematical Methods of Operations Research*, 63(1):169–186, 2006.

K. Boda, J. Filar, Y. Lin, and L. Spanjers. Stochastic target hitting time and the problem of early retirement. *Automatic Control, IEEE Transactions on*, 49(3):409–419, 2004.

V. Borkar. A sensitivity formula for the risk-sensitive cost and the actor-critic algorithm. *Systems & Control Letters*, 44:339–346, 2001.

V. Borkar. Q-learning for risk-sensitive control. *Mathematics of Operations Research*, 27:294–311, 2002.

V. Borkar. An actor-critic algorithm for constrained Markov decision processes. *Systems & Control Letters*, 54(3):207–213, 2005.

V. Borkar. *Stochastic approximation: a dynamical systems viewpoint*. Cambridge University Press, 2008.

V. Borkar and R. Jain. Risk-constrained Markov decision processes. *IEEE Transaction on Automatic Control*, 2014.

Y. Chow and M. Ghavamzadeh. Algorithms for CVaR optimization in MDPs. In *Advances in Neural Information Processing Systems*, pages 3509–3517, 2014.

Y. Chow and M. Pavone. Stochastic Optimal Control with Dynamic, Time-Consistent Risk Constraints. In *American Control Conference*, pages 390–395, Washington, DC, June 2013. doi: 10.1109/ACC.2013.6579868. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6579868`.

E. Collins. Using Markov decision processes to optimize a nonlinear functional of the final distribution, with manufacturing applications. In *Stochastic Modelling in Innovative Manufacturing*, pages 30–45. Springer, 1997.

B. Derfer, N. Goodyear, K. Hung, C. Matthews, G. Paoni, K. Rollins, R. Rose, M. Seaman, and J. Wiles. Online marketing platform, August 17 2007. US Patent App. 11/893,765.

J. Filar, L. Kallenberg, and H. Lee. Variance-penalized Markov decision processes. *Mathematics of Operations Research*, 14(1):147–161, 1989.

J. Filar, D. Krass, and K. Ross. Percentile performance criteria for limiting average Markov decision processes. *IEEE Transaction of Automatic Control*, 40(1):2–10, 1995.

R. Howard and J. Matheson. Risk sensitive Markov decision processes. *Management Science*, 18(7):356–369, 1972.

H. Khalil and J. Grizzle. *Nonlinear systems*, volume 3. Prentice hall Upper Saddle River, 2002.

V. Konda and J. Tsitsiklis. Actor-Critic algorithms. In *Proceedings of Advances in Neural Information Processing Systems 12*, pages 1008–1014, 2000.

G. Konidaris, S. Osentoski, and P. Thomas. Value function approximation in reinforcement learning using the Fourier basis. In *AAAI*, 2011.

H. Kushner and G. Yin. *Stochastic approximation algorithms and applications*. Springer, 1997.

P. Marbach. *Simulated-Based Methods for Markov Decision Processes*. PhD thesis, Massachusetts Institute of Technology, 1998.

P. Milgrom and I. Segal. Envelope theorems for arbitrary choice sets. *Econometrica*, 70(2):583–601, 2002.

T. Morimura, M. Sugiyama, M. Kashima, H. Hachiya, and T. Tanaka. Nonparametric return distribution approximation for reinforcement learning. In *Proceedings of the 27th International Conference on Machine Learning*, pages 799–806, 2010.

M. Ono, M. Pavone, Y. Kuwata, and J. Balaram. Chance-constrained dynamic programming with application to risk-aware robotic space exploration. *Autonomous Robots*, 39(4):555–571, 2015.

J. Ott. *A Markov decision model for a surveillance application and risk-sensitive Markov decision processes*. PhD thesis, Karlsruhe Institute of Technology, 2010.

J. Peters, S. Vijayakumar, and S. Schaal. Natural actor-critic. In *Proceedings of the Sixteenth European Conference on Machine Learning*, pages 280–291, 2005.

M. Petrik and D. Subramanian. An approximate solution method for large risk-averse Markov decision processes. In *Proceedings of the 28th International Conference on Uncertainty in Artificial Intelligence*, 2012.

L. Prashanth and M. Ghavamzadeh. Actor-critic algorithms for risk-sensitive MDPs. In *Proceedings of Advances in Neural Information Processing Systems 26*, pages 252–260, 2013.

R. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–42, 2000.

R. Rockafellar and S. Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking and Finance*, 26(7):1443 – 1471, 2002.

G. Shani, R. Brafman, and D. Heckerman. An MDP-based recommender system. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 453–460. Morgan Kaufmann Publishers Inc., 2002.

A. Shapiro, W. Tekaya, J. da Costa, and M. Soares. Risk neutral and risk averse stochastic dual dynamic programming method. *European journal of operational research*, 224(2):375–391, 2013.

T. Shardlow and A. Stuart. A perturbation theory for ergodic Markov chains and application to numerical approximations. *SIAM journal on numerical analysis*, 37(4):1120–1137, 2000.

M. Sobel. The variance of discounted Markov decision processes. *Applied Probability*, pages 794–802, 1982.

J. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, 1992.

R. Sutton and A. Barto. *Introduction to reinforcement learning*. MIT Press, 1998.

R. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of Advances in Neural Information Processing Systems 12*, pages 1057–1063, 2000.

Y. Le Tallec. *Robust, risk-sensitive, and data-driven control of Markov decision processes*. PhD thesis, Massachusetts Institute of Technology, 2007.

A. Tamar and S. Mannor. Variance adjusted actor critic algorithms. *arXiv preprint arXiv:1310.3697*, 2013.

A. Tamar, D. Di Castro, and S. Mannor. Policy gradients with variance related risk criteria. In *Proceedings of the Twenty-Ninth International Conference on Machine Learning*, pages 387–396, 2012.

A. Tamar, Y. Glassner, and S. Mannor. Policy gradients beyond expectations: Conditional value-at-risk. In *AAAI*, 2015.

G. Theocharous and A. Hallak. Lifetime value marketing using reinforcement learning. *RLDM 2013*, page 19, 2013.

D. White. Mean, variance, and probabilistic criteria in finite Markov decision processes: A review. *Journal of Optimization Theory and Applications*, 56(1):1–29, 1988.

R. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

C. Wu and Y. Lin. Minimizing risk models in Markov decision processes with policies depending on target values. *Journal of Mathematical Analysis and Applications*, 231(1):47–67, 1999.

Journal of Machine Learning Research 18 (2018) 1-56 Submitted 3/16; Revised 7/17; Published 4/18

Local Identifiability of ℓ_1-minimization Dictionary Learning: a Sufficient and Almost Necessary Condition

Siqi Wu SIQI@STAT.BERKELEY.EDU

Bin Yu * BINYU@BERKELEY.EDU

Department of Statistics
University of California
Berkeley, CA 94720-1776, USA

Editor: Hui Zou

Abstract

We study the theoretical properties of learning a dictionary from N signals $\mathbf{x}_i \in \mathbb{R}^K$ for $i = 1, \ldots, N$ via ℓ_1-minimization. We assume that $\mathbf{x}_i$'s are *i.i.d.* random linear combinations of the K columns from a complete (i.e., square and invertible) reference dictionary $\mathbf{D}_0 \in \mathbb{R}^{K\times K}$. Here, the random linear coefficients are generated from either the s-sparse Gaussian model or the Bernoulli-Gaussian model. First, for the population case, we establish a sufficient and almost necessary condition for the reference dictionary $\mathbf{D}_0$ to be locally identifiable, i.e., a strict local minimum of the expected ℓ_1-norm objective function. Our condition covers both sparse and dense cases of the random linear coefficients and significantly improves the sufficient condition by Gribonval and Schnass (2010). In addition, we show that for a complete μ-coherent reference dictionary, i.e., a dictionary with absolute pairwise column inner-product at most $\mu \in [0, 1)$, local identifiability holds even when the random linear coefficient vector has up to $O(\mu^{-2})$ nonzero entries. Moreover, our local identifiability results also translate to the finite sample case with high probability provided that the number of signals N scales as $O(K \log K)$.

Keywords: dictionary learning, ℓ_1-minimization, local minimum, non-convex optimization, sparse decomposition

1. Introduction

Expressing signals as sparse linear combinations of a dictionary basis has enjoyed great success in applications ranging from image denoising to audio compression. Given a known dictionary matrix $\mathbf{D} \in \mathbb{R}^{d\times K}$ with K columns or atoms, one popular method to recover the sparse coefficients $\boldsymbol{\alpha} \in \mathbb{R}^K$ of a signal $\mathbf{x} \in \mathbb{R}^d$ is through solving the convex ℓ_1-minimization problem

$$\text{minimize } \|\boldsymbol{\alpha}\|_1 \text{ subject to } \mathbf{x} = \mathbf{D}\boldsymbol{\alpha}.$$

This approach, known as *basis pursuit* (Chen et al., 1998), along with many of its variants, has been studied extensively in statistics and signal processing communities. See e.g., Donoho and Elad (2003); Fuchs (2004); Candes and Tao (2005).

For certain data types such as natural image patches, predefined dictionaries like the wavelets (Mallat, 2008) are usually available. However, for a less-known data type, a new dictionary has to be designed to effectively represent the data. Dictionary learning, or sparse coding, learns adaptively a dictionary from a set of training signals such that they have sparse representations under this dictionary (Olshausen and Field, 1997). One formulation of dictionary learning involves solving a non-convex ℓ_1-minimization problem (Zibulevsky et al., 2001; Plumbley, 2007; Gribonval and Schnass, 2010; Geng et al., 2011). Concretely, define

$$l(\mathbf{x}, \mathbf{D}) = \min_{\boldsymbol{\alpha}\in\mathbb{R}^K} \{\|\boldsymbol{\alpha}\|_1, \text{ subject to } \mathbf{x} = \mathbf{D}\boldsymbol{\alpha}\}. \tag{1}$$

We learn a dictionary from the N signals $\mathbf{x}_i \in \mathbb{R}^d$ for $i = 1, \ldots, N$ by solving

$$\min_{\mathbf{D}\in\mathcal{D}} \; L_N(\mathbf{D}) = \min_{\mathbf{D}\in\mathcal{D}} \frac{1}{N}\sum_{i=1}^{N} l(\mathbf{x}_i, \mathbf{D}). \tag{2}$$

Here, $\mathcal{D} \subset \mathbb{R}^{d\times K}$ is a constraint set for candidate dictionaries. In many signal processing tasks, learning an adaptive dictionary via the optimization problem (2) and its variants is empirically demonstrated to have superior performance over fixed standard dictionaries (Elad and Aharon, 2006; Peyré, 2009; Grosse et al., 2012). For a review of dictionary learning algorithms and applications, see Elad (2010); Rubinstein et al. (2010); Mairal et al. (2014).

Apart from the empirical success of many dictionary learning formulations, recently there is a growing body of work on the theory of dictionary learning. One line of research treats the problem of *dictionary identifiability*: if the signals are generated using a dictionary $\mathbf{D}_0$ referred to as the *reference dictionary*, under what conditions can we recover $\mathbf{D}_0$ by solving the dictionary learning problem? Being able to identify the reference dictionary is important when there is a need to interpret the learned dictionary, see for example Wu et al. (2016). Let $\boldsymbol{\alpha}_i \in \mathbb{R}^K$ for $i = 1, \ldots, N$ be some random vectors. A popular signal generation model assumes that a signal vector can be expressed as a linear combination of the columns of the reference dictionary: $\mathbf{x}_i \approx \mathbf{D}_0\boldsymbol{\alpha}_i$ (Gribonval and Schnass, 2010; Geng et al., 2011; Gribonval et al., 2015). In this paper, we will study the problem of *local identifiability* of ℓ_1-minimization dictionary learning (2) under this generating model.

Local identifiability. A reference dictionary $\mathbf{D}_0$ is said to be *locally identifiable* with respect to an objective function $L(\mathbf{D})$ if $\mathbf{D}_0$ is one of the strict local minima of L. The pioneer work of Gribonval and Schnass (2010) (referred to as GS henceforth) analyzed the ℓ_1-minimization problem (2) for noiseless signals ($\mathbf{x}_i = \mathbf{D}_0\boldsymbol{\alpha}_i$) and complete ($K = d$ and full rank) dictionaries. Under the sparse Bernoulli-Gaussian model for the linear coefficients $\boldsymbol{\alpha}_i$'s, they showed that for a sufficiently incoherent reference dictionary $\mathbf{D}_0$, $N = O(K \log K)$ samples can guarantee local identifiability with respect to $L_N(\mathbf{D})$ in (2) with high probability. Still in the noiseless setting, Geng et al. (2011) extended the analysis to over-complete ($K > d$) dictionaries. More recently under the noisy linear generative model with possible outliers, Gribonval et al. (2015) established local identifiability results for (2) with $l(\mathbf{x}, \mathbf{D})$ replaced by the LASSO objective function of Tibshirani (1996). Other related works on local identifiability include Schnass (2014) and Schnass (2015), who gave respectively sufficient conditions for the local correctness of the K-SVD (Aharon et al., 2006b) algorithm and a maximum response formulation of dictionary learning.

∗. Also in the Department of Electrical Engineering & Computer Science.

Contributions. There has not been much work on necessary conditions for local dictionary identifiability. Numerical experiments demonstrate that local identifiability undergoes a phase transition (Figure 1; see also Figure 3 of GS). The bound implied by the sufficient condition in GS falls well below the empirical phase boundary. Thus, even though theoretical results for the more general scenarios are available, we adopt the noiseless signals and complete dictionary setting of GS in order to find better local identifiability conditions. We summarize our main contributions below:

- For the population case where $N = \infty$, we establish a sufficient and almost necessary condition for local identifiability under both the s-sparse Gaussian and the Bernoulli-Gaussian models. For the Bernoulli-Gaussian model, the phase boundary implied by our condition significantly improves the GS bound and agrees well with the empirical phase boundary (Figure 1).

- We provide lower and upper bounds to approximate the quantities in our local identifiability condition, as it generally requires to solve a series of second-order cone programs to compute those quantities.

- As a consequence, we show that a μ-coherent reference dictionary—a dictionary with absolute pairwise column inner-product at most $\mu \in [0, 1)$—is locally identifiable for sparsity level, measured by the average number of nonzeros in the random linear coefficient vectors, up to the order $O(\mu^{-2})$. Moreover, if the sparsity level is greater than $O(\mu^{-2})$, the reference dictionary is generally not locally identifiable. In comparison, the sufficient condition by GS demands the number of dictionary atoms $K = O(\mu^{-2})$, which is a much more stringent requirement. For over-complete dictionaries, Geng et al. (2011) requires the sparsity level to be of the order $O(\mu^{-1})$. It should also be noted that Schnass (2015) establishes the bound $O(\mu^{-2})$ for *approximate* local identifiability under a new response maximization formulation of dictionary learning. To the best of our knowledge, our result is the first in showing that $O(\mu^{-2})$ is achievable and optimal for *exact* local recovery under the ℓ_1-minimization criterion.

- We also extend our identifiability results to the finite sample case. We show that for a fixed sparsity level, we need $N = O(K \log K)$ *i.i.d.* signals to determine whether or not the reference dictionary can be identified locally. This sample requirement is the same as GS's and is the best known sample requirement among all previous studies on local identifiability.

Other related works. Apart from analyzing the local minima of dictionary learning, another line of research aims at designing provable algorithms for recovering the reference dictionary. Georgiev et al. (2005) and Aharon et al. (2006a) proposed combinatorial algorithms and gave deterministic conditions for dictionary recovery which require sample size N to be exponentially large in the number of dictionary atoms K. Spielman et al. (2012) established exact global recovery results for complete dictionaries through efficient convex programs. Agarwal et al. (2014c) and Arora et al. (2014) proposed clustering-based methods to estimate the reference dictionary in the over-complete setting. Agarwal et al. (2014a) and Arora et al. (2015) provided theoretical guarantees for their alternating minimization algorithms. Sun et al. (2017) proposed a non-convex optimization algorithm that provably

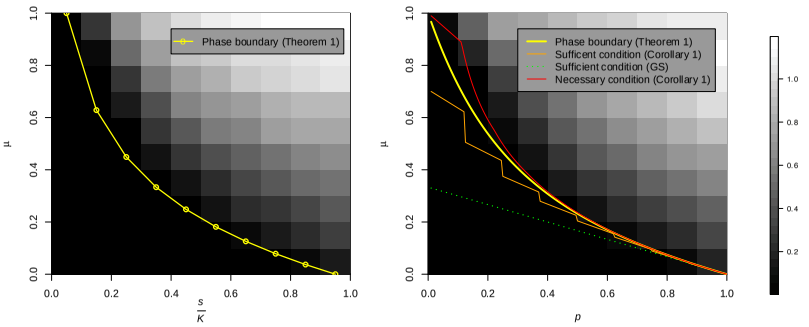


Figure 1: Local recovery errors for the s-sparse Gaussian model (Left) and the Bernoulli(p)-Gaussian model (Right). Under the s-sparse Gaussian model, the parameter $s \in \{1, \ldots, K\}$ is the number of nonzeros in each linear coefficient vector. Under the Bernoulli(p)-Gaussian model, $p \in (0, 1]$ is the probability of an entry of the linear coefficient vector being nonzero. The data are generated with the reference dictionary $\mathbf{D}_0 \in \mathbb{R}^{10\times 10}$ (i.e., $K = 10$) satisfying $\mathbf{D}_0^T\mathbf{D}_0 = \mu\mathbf{1}\mathbf{1}^T + (1-\mu)\mathbf{I}$ for $\mu \in [0, 1)$, see Example 5 for details. For each $(\mu, \frac{s}{K})$ or (μ, p) tuple, ten batches of $N = 2000$ signals $\{\mathbf{x}_i\}_{i=1}^{2000}$ are generated according to the noiseless linear model $\mathbf{x}_i = \mathbf{D}_0\boldsymbol{\alpha}_i$, with $\{\boldsymbol{\alpha}_i\}_{i=1}^{2000}$ drawn *i.i.d.* from the s-sparse Gaussian model or *i.i.d.* from the Bernoulli(p)-Gaussian model. For each batch, the dictionary is estimated through an alternating minimization algorithm in the `SPAMS` package (Mairal et al., 2010), with initial dictionary set to be $\mathbf{D}_0$. The grayscale intensity in the figure corresponds to the Frobenius error of the difference between the estimated dictionary and the reference dictionary $\mathbf{D}_0$, averaged for the ten batches. The "phase boundary" curve corresponds to the theoretical boundary that separates the region of local identifiability (below the curve) and the region of local non-identifiability (above the curve) according to Theorem 1 of this paper. The "Sufficient condition (Corollary 1)" and "Necessary condition (Corollary 1)" curves are the lower and upper bounds given by Corollary 1 to approximate the exact phase boundary. Finally, the "Sufficient condition (GS)" curve corresponds to the lower bound by GS. Note that for the s-sparse Gaussian model, the "Sufficient condition (Corollary 1)" and "Necessary condition (Corollary 1)" curves coincide with the phase boundary. See also Appendix Figures B.1 and B.2 for additional simulation results with $K = 20$ and $K = 50$.

recovers a complete reference dictionary for sparsity level up to $O(K)$. While in this paper we do not provide an algorithm, our identifiability conditions suggest theoretical limits of dictionary recovery for all algorithms attempting to solve the optimization problem (2). In particular, in the regime where the reference dictionary is not identifiable, no algorithm can simultaneously solve (2) and recover the ground truth reference dictionary.

Other related works include generalization bounds for signal reconstruction errors under the learned dictionary (Maurer and Pontil, 2010; Vainsencher et al., 2011; Mehta and Gray, 2013; Gribonval et al., 2013), dictionary identifiability through combinatorial matrix theory (Hillar and Sommer, 2015), as well as algorithms and theories for the closely related independent component analysis (Comon, 1994; Arora et al., 2012b) and nonnegative matrix factorization (Arora et al., 2012a; Recht et al., 2012).

The rest of the paper is organized as follows: In Section 2, we give basic assumptions and describe the two probabilistic models for signal generation. Section 3 develops sufficient and almost necessary local identifiability conditions under both models for the population problem, and establishes lower and upper approximating bounds. In Section 4, we will present local identifiability results for the finite sample problem. Detailed proofs for the theoretical results can be found in the Appendix.

2. Preliminaries

In this section, we will introduce notations and basic assumptions for our analysis.

2.1 Notations

For a positive integer m, define $[\![m]\!]$ to be the set of the first m positive integers, $\{1, \ldots, m\}$. The notation $\mathbf{x}[i]$ denotes the i-th entry of the vector $\mathbf{x} \in \mathbb{R}^m$. For a non-empty index set $S \subset [\![m]\!]$, we denote by $|S|$ the set cardinality and $\mathbf{x}[S] \in \mathbb{R}^{|S|}$ the sub-vector indexed by S. We define $\mathbf{x}[-j] := (\mathbf{x}[1], \ldots, \mathbf{x}[j-1], \mathbf{x}[j+1], \ldots, \mathbf{x}[m]) \in \mathbb{R}^{m-1}$ to be the vector $\mathbf{x}$ without its j-th entry.

For a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, we denote by $\mathbf{A}[i, j]$ its (i, j)-th entry. For non-empty sets $S \subset [\![m]\!]$ and $T \subset [\![n]\!]$, denote by $\mathbf{A}[S, T]$ the sub-matrix of $\mathbf{A}$ with the rows indexed by S and columns indexed by T. Denote by $\mathbf{A}[i,]$ and $\mathbf{A}[, j]$ the i-th row and the j-th column of $\mathbf{A}$ respectively. Similar to the vector case, the notation $\mathbf{A}[-i, j] \in \mathbb{R}^{m-1}$ denotes the j-th column of $\mathbf{A}$ without its i-th entry.

For $p \geq 1$, the ℓ_p-norm of a vector $\mathbf{x} \in \mathbb{R}^m$ is defined as $\|\mathbf{x}\|_p = (\sum_{i=1}^m |\mathbf{x}[i]|^p)^{1/p}$, with the convention that $\|\mathbf{x}\|_0 = |\{i : \mathbf{x}[i] \neq 0\}|$ and $\|\mathbf{x}\|_\infty = \max_i |\mathbf{x}[i]|$. For any norm $\|.\|$ on $\mathbb{R}^m$, the dual norm of $\|.\|$ is defined as $\|\mathbf{x}\|^* = \sup_{\mathbf{y} \neq 0} \frac{\mathbf{x}^T \mathbf{y}}{\|\mathbf{y}\|}$.

For two sequences of real numbers $\{a_n\}_{n=1}^\infty$ and $\{b_n\}_{n=1}^\infty$, we denote by $a_n = O(b_n)$ if there is a constant $C > 0$ such that $a_n \leq C b_n$ for all $n \geq 1$. For $a \in \mathbb{R}$, denote by $\lfloor a \rfloor$ the integer part of a and $\lceil a \rceil$ the smallest integer greater than or equal to a. Throughout this paper, we shall agree that $\frac{0}{0} = 0$.

2.2 Basic Assumptions

We denote by $\mathcal{D} \subset \mathbb{R}^{d \times K}$ the constraint set of dictionaries for the optimization problem (2). In this paper, we consider square dictionaries, i.e., $d = K \geq 2$. As in GS, we choose $\mathcal{D}$ to

be the *oblique manifold* (Absil et al., 2008):

$$\mathcal{D} = \left\{ \mathbf{D} \in \mathbb{R}^{K \times K} : \ \|\mathbf{D}[, k]\|_2 = 1 \text{ for all } k = 1, \ldots, K \right\}.$$

We also call a dictionary column $\mathbf{D}[, k]$ an *atom* of the dictionary. Denote by $\mathbf{D}_0 \in \mathcal{D}$ the *reference dictionary*, i.e., the ground truth dictionary that generates the signals. With these notations, we now give a formal definition for local identifiability:

Definition 1 *(Local identifiability) Let $L(\mathbf{D}) : \mathcal{D} \to \mathbb{R}$ be an objective function. We say that the reference dictionary $\mathbf{D}_0$ is locally identifiable with respect to $L(\mathbf{D})$ if $\mathbf{D}_0$ is a strict local minimum of $L(\mathbf{D})$.*

Sign-permutation ambiguity. As noted by previous works GS and Geng et al. (2011), the ℓ_1-norm objective function $L(\mathbf{D}) = L_N(\mathbf{D})$ of (2) has an intrinsic sign-permutation ambiguity. Let $\mathbf{D}' = \mathbf{D}\mathbf{P}\mathbf{\Lambda}$ for some permutation matrix $\mathbf{P}$ and diagonal matrix $\mathbf{\Lambda}$ with ± 1 diagonal entries. It is easy to see that $\mathbf{D}'$ and $\mathbf{D}$ have the same objective value. Thus, the objective function $L_N(\mathbf{D})$ has at least $2^K K!$ local minima. We can only recover $\mathbf{D}_0$ up to column permutation and sign changes.

Note that if the dictionary atoms are linearly dependent, the effective dimension is strictly less than K and the problem essentially becomes over-complete. Since dealing with over-complete dictionaries is beyond the scope of this paper, we make the following assumption:

Assumption I *(Complete dictionaries). The reference dictionary $\mathbf{D}_0 \in \mathcal{D} \subset \mathbb{R}^{K \times K}$ is full rank.*

Let $\mathbf{M}_0 = \mathbf{D}_0^T \mathbf{D}_0$ be the *dictionary atom collinearity matrix* containing the inner-products between dictionary atoms. Since each dictionary atom has unit ℓ_2-norm, $\mathbf{M}_0[i, i] = 1$ for all $i \in [\![K]\!]$. In addition, as $\mathbf{D}_0$ is full rank, $\mathbf{M}_0$ is positive definite and $|\mathbf{M}_0[i, j]| < 1$ for all $i \neq j$.

We assume that a signal is generated as a random linear combination of the dictionary atoms. In this paper, we consider the following two probabilistic models for the random linear coefficients:

Probabilistic models for sparse coefficients. Denote by $\mathbf{z} \in \mathbb{R}^K$ a random vector from the K-dimensional standard normal distribution.

Model 1—$SG(s)$**.** Let $\mathbf{S}$ be a size-s subset uniformly drawn from all size-s subsets of $[\![K]\!]$. Define $\boldsymbol{\xi} \in \{0, 1\}^K$ by setting $\boldsymbol{\xi}[j] = I\{j \in \mathbf{S}\}$ for $j \in [\![K]\!]$, where $I\{.\}$ is the indicator function. Let $\boldsymbol{\alpha} \in \mathbb{R}^K$ be such that $\boldsymbol{\alpha}[j] = \boldsymbol{\xi}[j]\mathbf{z}[j]$. Then we say $\boldsymbol{\alpha}$ is drawn from the *s-sparse Gaussian model*, or $SG(s)$.

Model 2—$BG(p)$**.** For $j \in [\![K]\!]$, let $\boldsymbol{\xi}[j]$'s be *i.i.d.* Bernoulli random variable with success probability $p \in (0, 1]$. Let $\boldsymbol{\alpha} \in \mathbb{R}^K$ be such that $\boldsymbol{\alpha}[j] = \boldsymbol{\xi}[j]\mathbf{z}[j]$. Then we say $\boldsymbol{\alpha}$ is drawn from the *Bernoulli(p)-Gaussian model*, or $BG(p)$.

With the above two models, we state the following assumption for random signal generation:

Assumption II *(Signal generation). For $i \in [\![N]\!]$, let $\boldsymbol{\alpha}_i \in \mathbb{R}^K$ be either i.i.d. s-sparse Gaussian vectors or i.i.d. Bernoulli(p)-Gaussian vectors. The signals $\mathbf{x}_i \in \mathbb{R}^K$ are generated according to the noiseless linear model:*

$$\mathbf{x}_i = \mathbf{D}_0 \boldsymbol{\alpha}_i.$$

Remarks:
(1) The above two models and their variants were studied in a number of prior theoretical works, including Gribonval and Schnass (2010); Geng et al. (2011); Agarwal et al. (2014b); Sun et al. (2017).
(2) By construction, a random vector generated from the s-sparse model has exactly s nonzero entries. The data points $\mathbf{x}_i$'s therefore lie within the union of the linear spans of s dictionary atoms (Figure 2 Left). The Bernoulli(p)-Gaussian model, on the other hand, allows the random coefficient vector to have any number of nonzero entries ranging from 0 to K with a mean of pK. As a result, some data points, called "non-sparse outliers" in GS, can be outside of any sparse linear span of the dictionary atoms (Figure 2 Right and Figure 1 of GS). We refer readers to the remarks following Example 5 in Section 3 for a discussion of the effect of non-sparse outliers on local identifiability.
(3) Gribonval et al. (2015) assumed a more general distribution for the sparse coefficients. While our local identifiability results can potentially be extended to their model, such an extension would require significantly more complicated notations and make the corresponding results less interpretable. For the sake of accessibility and interpretability, we focus only on the two probabilistic models above.

In this paper, we study the problem of dictionary identifiability with respect to the population objective function $\mathbb{E}\, L_N(\mathbf{D})$ (Section 3) and the finite sample objective function $L_N(\mathbf{D})$ (Section 4). In order to analyze these objective functions, it is convenient to define the following "group LASSO"-type norms:

Definition 2 *Let $m \geq 1$ be an integer and $\mathbf{w} \in \mathbb{R}^m$.*

1. *For $k \in [\![m]\!]$, define*

$$|||\mathbf{w}|||_k = \frac{\sum_{|S|=k} \|\mathbf{w}[S]\|_2}{\binom{m-1}{k-1}}.$$

2. *For $p \in (0,1)$, define*

$$|||\mathbf{w}|||_p = \sum_{k=0}^{m-1} \texttt{pbinom}(k; m-1, p) |||\mathbf{w}|||_{k+1},$$

where `pbinom` *is the probability mass function of the binomial distribution:*

$$\texttt{pbinom}(k; n, p) = \binom{n}{k} p^k (1-p)^{n-k}.$$

Remarks:
(1) Note that the above norms $|||\mathbf{w}|||_k$ and $|||\mathbf{w}|||_p$ are in fact the expected values of $|\mathbf{w}^T \boldsymbol{\alpha}|$

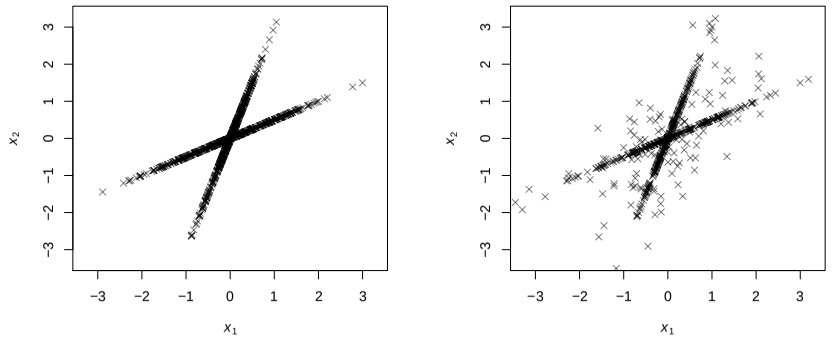


Figure 2: Data generation for $K = 2$. Left: the s-sparse Gaussian model with $s = 1$; Right: the Bernoulli(p)-Gaussian model with $p = 0.2$. The inner product between the two dictionary atoms is 0.7. A sample of $N = 1000$ data points are generated for both models. For the s-sparse model, all data points are perfectly aligned with the two lines corresponding to the two dictionary atoms. For the Bernoulli(p)-Gaussian model, a number of data points fall outside the two lines. According to our Theorem 1 and 3, despite those outliers and the high collinearity between the two atoms, the reference dictionary is still locally identifiable for $N = \infty$ and with high probability for finite samples.

with the random vector $\boldsymbol{\alpha}$ drawn from $SG(s)$ and $BG(p)$ models respectively. For invertible $\mathbf{D} \in \mathcal{D}$, it can be shown that the objective function for one signal $\mathbf{x} = \mathbf{D}_0 \boldsymbol{\alpha}$ is

$$l(\mathbf{x}, \mathbf{D}) = \|\mathbf{H}\boldsymbol{\alpha}\|_1 = \sum_{j=1}^{K} |\mathbf{H}[j,]\boldsymbol{\alpha}|,$$

where $\mathbf{H} = \mathbf{D}^{-1}\mathbf{D}_0$. Thus, taking the expectation of the objective function with respect to $\mathbf{x}$, we end up with a quantity involving either $\sum_{j=1}^K |||\mathbf{H}[j,]|||_s$ or $\sum_{j=1}^K |||\mathbf{H}[j,]|||_p$. This is the motivation of defining these norms.
(2) In particular, $|||\mathbf{w}|||_1 = \|\mathbf{w}\|_1$ and $|||\mathbf{w}|||_m = \|\mathbf{w}\|_2$.
(3) The norms defined above are special cases of the group LASSO penalty by Yuan and Lin (2006). For $|||\mathbf{w}|||_k$, the summation covers all size-k subsets of $[\![m]\!]$. The normalization factor is the number of times $\mathbf{w}[i]$ appears in the numerator. Thus, $|||\mathbf{w}|||_k$ is essentially the average of the ℓ_2-norms of all size-k sub-vectors of $\mathbf{w}$. On the other hand, $|||\mathbf{w}|||_p$ is a weighted average of $|||\mathbf{w}|||_k$'s with binomial probabilities.

3. Population Analysis

In this section, we establish local identifiability results for the case where infinitely many signals are observed. Denote by $\mathbb{E}\, l(\mathbf{x}_1, \mathbf{D})$ the expectation of the objective function $l(\mathbf{x}_1, \mathbf{D})$ of (1) with respect to the random signal $\mathbf{x}_1$. By the strong law of large numbers, as the number of signals N tends to infinity, the empirical objective function $L_N(\mathbf{D}) = \frac{1}{N}\sum_{i=1}^{N} l(\mathbf{x}_i, \mathbf{D})$ converges almost surely to its population mean $\mathbb{E}\, l(\mathbf{x}_1, \mathbf{D})$ for each fixed $\mathbf{D} \in \mathcal{D}$. Therefore the population version of the optimization problem (2) is

$$\min_{\mathbf{D}\in\mathcal{D}} \mathbb{E}\, l(\mathbf{x}_1, \mathbf{D}) \tag{3}$$

Since by assumption the reference dictionary $\mathbf{D}_0$ is full rank, we only need to work with $\mathbf{D} \in \mathcal{D}$ that is also full rank. Indeed, if the linear span of columns $\mathrm{span}(\mathbf{D}) \neq \mathbb{R}^K$, then $\mathbf{D}_0\boldsymbol{\alpha}_1 \notin \mathrm{span}(\mathbf{D})$ with nonzero probability. Thus $\mathbf{D}$ is infeasible with nonzero probability and so $\mathbb{E}\, l(\mathbf{x}_1, \mathbf{D}) = +\infty$. For a full rank dictionary $\mathbf{D}$, the following lemma gives the closed-form expression for the expected objective function $\mathbb{E}\, l(\mathbf{x}_1, \mathbf{D})$.

Lemma 1 *(Closed-form objective functions) Let $\mathbf{D}$ be a full rank dictionary in $\mathcal{D}$ and $\mathbf{x}_1 = \mathbf{D}_0\boldsymbol{\alpha}_1$, where $\boldsymbol{\alpha}_1 \in \mathbb{R}^K$ is a random vector. For notational convenience, let $\mathbf{H} = \mathbf{D}^{-1}\mathbf{D}_0$.*

1. *If $\boldsymbol{\alpha}_1$ is generated according to the $SG(s)$ model with $s \in [\![K-1]\!]$,*

$$L_{SG(s)}(\mathbf{D}) \coloneqq \mathbb{E}\, l(\mathbf{x}_1, \mathbf{D}) = \sqrt{\frac{2}{\pi}}\frac{s}{K}\sum_{j=1}^{K} |\!|\!| \mathbf{H}[j,] |\!|\!|_s. \tag{4}$$

2. *If $\boldsymbol{\alpha}_1$ is generated according to the $BG(p)$ model with $p \in (0,1)$,*

$$L_{BG(p)}(\mathbf{D}) \coloneqq \mathbb{E}\, l(\mathbf{x}_1, \mathbf{D}) = \sqrt{\frac{2}{\pi}}p\sum_{j=1}^{K} |\!|\!| \mathbf{H}[j,] |\!|\!|_p. \tag{5}$$

For the non-sparse cases where $s = K$ and $p = 1$, we have

$$L_{SG(s)}(\mathbf{D}) = L_{BG(p)}(\mathbf{D}) = \sqrt{\frac{2}{\pi}}\sum_{j=1}^{K} \|\mathbf{H}[j,]\|_2.$$

Remarks: It can be seen from the above closed-form expressions that the two models are closely related. First of all, it is natural to identify p with $\frac{s}{K}$, the fraction of expected number of nonzero entries in $\boldsymbol{\alpha}_1$. Next, by definition, $|\!|\!|.|\!|\!|_p$ is a binomial average of $|\!|\!|.|\!|\!|_k$. Therefore, the Bernoulli-Gaussian objective function $L_{BG(p)}(\mathbf{D})$ can be treated as a binomial average of the s-sparse objective function $L_{SG(s)}(\mathbf{D})$.

By analyzing the above closed-form expressions of the ℓ_1-norm objective function, we establish the following sufficient and almost necessary conditions for population local identifiability:

Theorem 1 *(Population local identifiability) Recall that $\mathbf{M}_0 = \mathbf{D}_0^T\mathbf{D}_0$ and $\mathbf{M}_0[-j,j]$ denotes the j-th column of the off-diagonal part of $\mathbf{M}_0$. Let $|\!|\!|.|\!|\!|_s^*$ and $|\!|\!|.|\!|\!|_p^*$ be the dual norm of $|\!|\!|.|\!|\!|_s$ and $|\!|\!|.|\!|\!|_p$ respectively.*

1. *($SG(s)$ models) For $K \geq 2$ and $s \in [\![K-1]\!]$, if*

$$\max_{j\in[\![K]\!]} |\!|\!| \mathbf{M}_0[-j,j] |\!|\!|_s^* < 1 - \frac{s-1}{K-1}.$$

   then $\mathbf{D}_0$ is locally identifiable with respect to $L_{SG(s)}$.

2. *($BG(p)$ models) For $K \geq 2$ and $p \in (0,1)$, if*

$$\max_{j\in[\![K]\!]} |\!|\!| \mathbf{M}_0[-j,j] |\!|\!|_p^* < 1 - p.$$

   then $\mathbf{D}_0$ is locally identifiable with respect to $L_{BG(p)}$.

Moreover, the above conditions are almost necessary in the sense that if the reversed strict inequalities hold, then $\mathbf{D}_0$ is not locally identifiable.

On the other hand, if $s = K$ or $p = 1$, then $\mathbf{D}_0$ is not locally identifiable with respect to $L_{SG(s)}$ or $L_{BG(p)}$.

Proof sketch. Let $\{\mathbf{D}_t\}_{t\in\mathbb{R}}$ be a collection of dictionaries $\mathbf{D}_t \in \mathcal{D}$ indexed by $t \in \mathbb{R}$ and $L(\mathbf{D}) = \mathbb{E}\, l(\mathbf{x}_1, \mathbf{D})$ be the population objective function. The reference dictionary $\mathbf{D}_0$ is a strict local minimum of $L(\mathbf{D})$ on the manifold $\mathcal{D}$ if and only if the following statement holds: for any $\{\mathbf{D}_t\}_{t\in\mathbb{R}}$ that is a smooth function of t with non-vanishing derivative at $t = 0$, $L(\mathbf{D}_t)$ has a strict local minimum at $t = 0$. For a fixed $\{\mathbf{D}_t\}_{t\in\mathbb{R}}$, to ensure that $L(\mathbf{D}_t)$ achieves a strict local minimum at $t = 0$, it suffices to have the following one-sided derivative inequalities:

$$\lim_{t\downarrow 0^+} \frac{L(\mathbf{D}_t) - L(\mathbf{D}_0)}{t} > 0 \text{ and } \lim_{t\uparrow 0^-} \frac{L(\mathbf{D}_t) - L(\mathbf{D}_0)}{t} < 0.$$

It can be shown that the above inequalities are equivalent to:

$$\max_{j\in[\![K]\!]} \left|\mathbf{M}_0[-j,j]^T\mathbf{w}\right| < \begin{cases} 1 - \frac{s-1}{K-1} & \text{for } SG(s) \\ 1-p & \text{for } BG(p) \end{cases}$$

where $\mathbf{w} \in \mathbb{R}^{K-1}$ is a unit vector under norm $|\!|\!|.|\!|\!|_s$ or $|\!|\!|.|\!|\!|_p$ and corresponds to the direction in which $\mathbf{D}_t$ approaches $\mathbf{D}_0$ as t tends to zero. Since $t = 0$ has to be a strict local minimum for all smooth $\{\mathbf{D}_t\}_{t\in\mathbb{R}}$ or approaching directions, by taking the supremum over all such unit vectors the LHS of the above inequality becomes the dual norm of $|\!|\!|.|\!|\!|_s$ or $|\!|\!|.|\!|\!|_p$. On the other hand, $\mathbf{D}_0$ is not a local minimum if $\lim_{t\downarrow 0^+}(L(\mathbf{D}_t) - L(\mathbf{D}_0))/t < 0$ or $\lim_{t\uparrow 0^-}(L(\mathbf{D}_t) - L(\mathbf{D}_0))/t > 0$ for some $\{\mathbf{D}_t\}_{t\in\mathbb{R}}$. Thus the sufficient condition is also almost necessary. We refer readers to Section A.1.2 for the detailed proof.

Local identifiability phase boundary. Theorem 1 indicates that population local identifiability undergoes a phase transition. The following equations

$$\max_{j\in[\![K]\!]} |\!|\!| \mathbf{M}_0[-j,j] |\!|\!|_s^* = 1 - \frac{s-1}{K-1} \text{ and } \max_{j\in[\![K]\!]} |\!|\!| \mathbf{M}_0[-j,j] |\!|\!|_p^* = 1 - p \tag{6}$$

define the phase boundaries which separate the regions of local identifiability and non-identifiability under respective models. It is unclear whether $\mathbf{D}_0$ is locally identifiable on the phase boundary. If either equality in (6) holds, the directional derivative of the objective function at $\mathbf{D}_0$ become zero in certain directions. Hence analyzing local identifiability in this case requires higher order derivative computations that quickly become complicated.

Collinearity and sparsity. These are the two factors that determine local identifiability. Intuitively, for $\mathbf{D}_0$ to be locally identifiable, neither can the atoms of $\mathbf{D}_0$ be too linearly dependent, nor can the random linear coefficients be too dense. For the s-sparse Gaussian model, the quantity $\max_{j\in[\![K]\!]} |||\mathbf{M}_0[-j,j]|||_s^*$ measures the size of the off-diagonal entries of $\mathbf{M}_0$ and hence the collinearity of the dictionary atoms. In addition, that quantity depends on the sparsity parameter s. By Lemma 7 in the Appendix, $\max_{j\in[\![K]\!]} |||\mathbf{M}_0[-j,j]|||_s^*$ is increasing with respect to s. Similar conclusion holds for the Bernoulli-Gaussian model. Therefore, sparser linear coefficients will lead to less restrictive requirement on dictionary atom collinearity. See, for example, the phase boundaries in Figure 1.

Next, we will present a few examples to gain more intuition for the local identifiability conditions.

Example 1 *(1-sparse Gaussian model) A full rank $\mathbf{D}_0$ is always locally identifiable at the population level under a 1-sparse Gaussian model. Indeed, by Corollary 7 in the Appendix, $|||\mathbf{M}_0[-j,j]|||_1^* = \max_{i\neq j} |\mathbf{M}_0[i,j]| < 1$ for all $j \in [\![K]\!]$. Thus, a full rank dictionary $\mathbf{D}_0$ always satisfies the sufficient condition.*

Example 2 *($(K-1)$-sparse Gaussian model) For $j \in [\![K]\!]$, by Corollary 7,*

$$|||\mathbf{M}_0[-j,j]|||_{K-1}^* = \|\mathbf{M}_0[-j,j]\|_2.$$

Therefore the phase boundary under the $(K-1)$-sparse model is

$$\max_{j\in[\![K]\!]} \|\mathbf{M}_0[-j,j]\|_2 = \frac{1}{K-1}.$$

Example 3 *(Orthogonal dictionaries) If $\mathbf{M}_0 = \mathbf{I}$, then*

$$\max_{j\in[\![K]\!]} |||\mathbf{M}_0[-j,j]|||_s^* = \max_{j\in[\![K]\!]} |||\mathbf{M}_0[-j,j]|||_p^* = 0.$$

Therefore orthogonal dictionaries are always locally identifiable if $s < K$ or $p < 1$.

Example 4 *(Minimally dependent dictionary atoms) Let $\mu \in (-1,1)$. Consider a dictionary atom collinearity matrix $\mathbf{M}_0$ such that $\mathbf{M}_0[1,2] = \mathbf{M}_0[2,1] = \mu$ and $\mathbf{M}_0[i,j] = 0$ for all other $i \neq j$. By Corollary 8,*

$$\max_{j\in[\![K]\!]} |||\mathbf{M}_0[-j,j]|||_s^* = \max_{j\in[\![K]\!]} |||\mathbf{M}_0[-j,j]|||_p^* = |\mu|.$$

Thus the phase boundaries under respective models are

$$|\mu| = 1 - \frac{s-1}{K-1} \text{ and } |\mu| = 1 - p.$$

Notice that for the Bernoulli-Gaussian model with $K = 2$, the phase boundary agrees with the empirical phase boundary in Figure 3 of GS.

Example 5 *(Constant inner-product dictionaries) Let $\mathbf{M}_0 = \mu \mathbf{1}\mathbf{1}^T + (1-\mu)\mathbf{I}$, i.e., $\mathbf{D}_0[,i]^T\mathbf{D}_0[,j] = \mu$ for $1 \leq i < j \leq K$. Note that $\mathbf{M}_0$ is positive definite if and only if $\mu \in (-\frac{1}{K-1}, 1)$. By Corollary 9, we have*

$$|||\mathbf{M}_0[-j,j]|||_s^* = \sqrt{s}|\mu|.$$

Thus for the s-sparse model, the phase boundary is

$$\sqrt{s}|\mu| = 1 - \frac{s-1}{K-1}.$$

Similarly for the Bernoulli(p)-Gaussian model, we have

$$|||\mathbf{M}_0[-j,j]|||_p^* = |\mu|p(K-1)\Big(\sum_{k=0}^{K-1} \texttt{pbinom}(k, K-1, p)\sqrt{k}\Big)^{-1}.$$

Thus the phase boundary is

$$|\mu| = \frac{1-p}{p(K-1)} \sum_{k=0}^{K-1} \texttt{pbinom}(k, K-1, p)\sqrt{k}.$$

Figures 3 shows the phase boundaries for different dictionary sizes under the two models. As K increases, the phase boundary moves toward the lower left of the region. This observation indicates that recovering the reference dictionary locally becomes increasingly difficult for larger dictionary size. See also Appendix Figures B.1 and B.2 for simulation results with larger K's.

The effect of non-sparse outliers. Example 5 demonstrates how the presence of non-sparse outliers in the Bernoulli-Gaussian model (Figure 2 Right) affects the requirements for local identifiability. Set $p = \frac{s}{K}$ in order to have the same level of sparsity with the $SG(s)$ model. Applying Jensen's inequality, one can show that

$$\frac{1-p}{p(K-1)} \sum_{k=0}^{K-1} \texttt{pbinom}(k, K-1, p)\sqrt{k} < \frac{1}{\sqrt{s}}(1 - \frac{s-1}{K-1}),$$

which indicates that the phase boundary of the s-sparse models is always above that of the Bernoulli-Gaussian model with the same level of sparsity. The gap between the two phase boundaries is the extra cost in terms of the collinearity parameter μ for locally recovering the dictionary in the presence of non-sparse outliers. One extreme example is the case where $s = 1$ and correspondingly $p = \frac{1}{K}$. By Example 1, under a 1-sparse model the reference dictionary $\mathbf{D}_0$ is always locally identifiable if $|\mu| < 1$. But for the $BG(\frac{1}{K})$ model, by the remarks under Corollary 1, $\mathbf{D}_0$ is not locally identifiable if $|\mu| > 1 - \frac{1}{K}$. Hence, the requirement for μ in the presence of outliers is at least $\frac{1}{K}$ more stringent than that in the case of no outliers.

However, such a difference diminishes as the number of dictionary atoms K increases. Indeed, by Lemma 2, one can establish the following lower bound for the phase boundary under the $BG(p)$ model

$$\frac{1-p}{p(K-1)} \sum_{k=0}^{K-1} \texttt{pbinom}(k, K-1, p)\sqrt{k} \geq \frac{1-p}{\sqrt{p(K-1)+1}} \approx \frac{1}{\sqrt{s}}(1 - \frac{s-1}{K-1}),$$

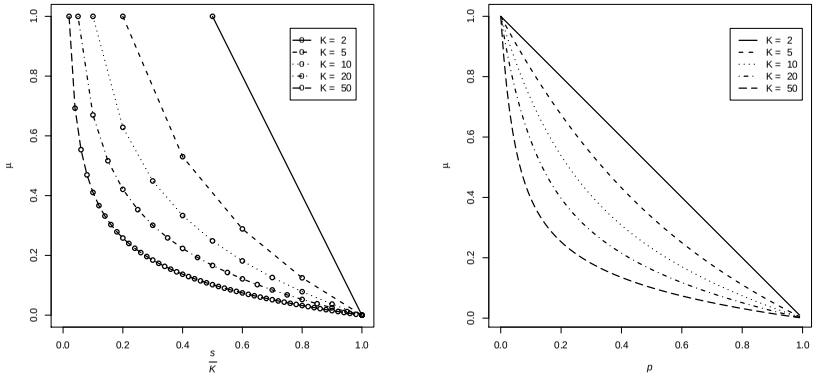


Figure 3: Local identifiability phase boundaries for constant inner-product dictionaries, under Left: the s-sparse Gaussian model; Right: the Bernoulli(p)-Gaussian model. For each model, phase boundaries for different dictionary sizes K are shown. Note that $\frac{s}{K} \in \{\frac{1}{K}, \frac{2}{K}, \ldots, 1\}$ and $p \in (0, 1]$. The area under the curves is the region where the reference dictionaries are locally identifiable at the population level. Due to symmetry, we only plot the portion of the phase boundaries for $\mu > 0$.

for fixed sparsity level $p = \frac{s}{K}$ and large K.

In general, the dual norms $|||.|||_s^*$ and $|||.|||_p^*$ have no closed-form expressions. According to Corollary 6 in the Appendix, computing those quantities involves solving a second order cone problem (SOCP) with exponentially many constraints. The following Lemma 2, on the other hand, gives computationally inexpensive approximation bounds.

Definition 3 *(Hyper-geometric distribution related quantities) Let m be a positive integer and $d, k \in \{0\} \cup [\![m]\!]$. Denote by $L_m(d, k)$ the hypergeometric random variable with parameter m, d and k, i.e., the number of 1's after drawing without replacement k elements from d 1's and $m - d$ 0's. Now for each $d \in \{0\} \cup [\![m]\!]$, define the function $\tau_m(d, .)$ with domain on $[0, m]$ as follows: set $\tau_m(d, 0) = 0$. For $a \in (k - 1, k]$ where $k \in [\![m]\!]$, define*

$$\tau_m(d, a) = \mathbb{E}\sqrt{L_m(d, k-1)} + (\mathbb{E}\sqrt{L_m(d, k)} - \mathbb{E}\sqrt{L_m(d, k-1)})(a - (k-1)).$$

Lemma 2 *(Lower and upper bounds for $|||.|||_s^*$ and $|||.|||_p^*$) Let m be a positive integer and $\mathbf{z} \in \mathbb{R}^m$.*

1. *For $s \in [\![m]\!]$,*

$$\max\left(\|\mathbf{z}\|_\infty, \sqrt{\frac{s}{m}} \max_{T \subset [\![m]\!]} \frac{\|\mathbf{z}[T]\|_1}{\sqrt{|T|}}\right) \le \frac{s}{m} \max_{T \subset [\![m]\!]} \frac{\|\mathbf{z}[T]\|_1}{\tau_m(|T|, s)} \le |||\mathbf{z}|||_s^* \le \max_{S \subset [\![m]\!], |S| = s} \|\mathbf{z}[S]\|_2.$$

2. *For $p \in (0, 1)$,*

$$\max\left(\|\mathbf{z}\|_\infty, \sqrt{p} \max_{T \subset [\![m]\!]} \frac{\|\mathbf{z}[T]\|_1}{\sqrt{|T|}}\right) \le p \max_{T \subset [\![m]\!]} \frac{\|\mathbf{z}[T]\|_1}{\tau_m(|T|, pm)} \le |||\mathbf{z}|||_p^* \le \max_{S \subset [\![m]\!], |S| = k} \|\mathbf{z}[S]\|_2.$$

where $k = \lceil p(m-1) + 1 \rceil$.

Remarks:
(1) We refer readers to Lemma 10 and 11 for the detailed version of the above results.
(2) Since we agree that $\frac{0}{0} = 0$, the case where $T = \emptyset$ does not affect taking the maximum of all subsets.
(3) Consider a sparse vector $\mathbf{z} = (z, 0, \ldots, 0)^T \in \mathbb{R}^m$. By Corollary 8,

$$|||\mathbf{z}|||_s^* = |||\mathbf{z}|||_p^* = |z| = \|\mathbf{z}\|_\infty = \max_{S \subset [\![m]\!], |S| = 1} \|\mathbf{z}[S]\|_2.$$

So the all the bounds are achievable by a sparse vector.
(4) Now consider a dense vector $\mathbf{z} = (z, \ldots, z)^T \in \mathbb{R}^m$. By Corollary 9,

$$|||\mathbf{z}|||_s^* = \sqrt{s}|z| = \sqrt{\frac{s}{m}} \max_{T \subset [\![m]\!]} \frac{\|\mathbf{z}[T]\|_1}{\sqrt{|T|}} = \max_{S \subset [\![m]\!], |S| = s} \|\mathbf{z}[S]\|_2.$$

Thus the bounds for $|||\mathbf{z}|||_s^*$ can also be achieved by a dense vector. Similarly, by the upper-bound for $|||\mathbf{z}|||_p^*$,

$$|||\mathbf{z}|||_p^* \le \sqrt{pm + 1}|z|.$$

On the other hand,

$$|||\mathbf{z}|||_p^* \ge \sqrt{p} \max_{T \subset [\![m]\!]} \frac{\|\mathbf{z}\|_1}{\sqrt{|T|}} = \sqrt{p}|z| \max_{T \subset [\![m]\!]} \sqrt{|T|} = \sqrt{pm}|z|.$$

Thus both bounds for $|||\mathbf{z}|||_p^*$ are basically the same for large pm.
(5) **Computation.** To compute the lower and upper bounds efficiently, we first sort the elements in $|\mathbf{z}|$ in descending order. Without loss of generality, we can assume that $|\mathbf{z}[1]| \ge |\mathbf{z}[2]| \ge \ldots \ge |\mathbf{z}[m]|$. Thus the upper-bound quantity becomes

$$\max_{S \subset [\![m]\!], |S| = k} \|\mathbf{z}[S]\|_2 = (\sum_{i=1}^k \mathbf{z}[i]^2)^{1/2}.$$

For the lower-bound quantities, note that

$$\max_{T \subset [\![m]\!]} \frac{\|\mathbf{z}[T]\|_1}{\tau_m(|T|, k)} = \max_{d \in [\![m]\!]} \max_{T \subset [\![m]\!], |T| = d} \frac{\|\mathbf{z}[T]\|_1}{\tau_m(d, k)} = \max_{d \in [\![m]\!]} \frac{\sum_{i=1}^d |\mathbf{z}[i]|}{\tau_m(d, k)}.$$

Thus, the major computation burden now is $\tau_m(d, k) = \mathbb{E}\sqrt{L_m(d, k)}$, for all $d \in [\![m]\!]$. We do not know a closed-form formula for $\mathbb{E}\sqrt{L_m(d, k)}$ except for $d = 1$ or $d = m$. In practice, we compute $\mathbb{E}\sqrt{L_m(d, k)}$ using its definition formula. On an OS X laptop with 1.8 GHz Intel Core i7 processor and 4GB of memory, the function `dhyper` in the statistics software

R can compute $\mathbb{E}\sqrt{L_{2000}(d,1000)}$ for all $d \in [\![2000]\!]$ within 0.635 second. Note that the number of dictionary atoms in most applications is typically smaller than 2000.

When m is too large, the LHS lower bounds can be used. Note that

$$\max_{T\subset[\![m]\!]} \frac{\|\mathbf{z}[T]\|_1}{\sqrt{|T|}} = \max_{d\in[\![m]\!]} \frac{\sum_{i=1}^d |\mathbf{z}[i]|}{\sqrt{d}},$$

which can be computed easily.

For notational simplicity, we will define the following quantities:

Definition 4 *For $a \in (0,K)$, define*

$$\nu_a(\mathbf{M}_0) = \max_{1\le j\le K} \max_{S\subset[\![K]\!], j\notin S} \frac{\|\mathbf{M}_0[S,j]\|_1}{\tau_{K-1}(|S|,a)}.$$

Definition 5 *(Cumulative coherence) For $k \in [\![K-1]\!]$, define the k-th cumulative coherence of a reference dictionary $\mathbf{D}_0$ as*

$$\mu_k(\mathbf{M}_0) = \max_{1\le j\le K} \max_{S\subset[\![K]\!], |S|=k, j\notin S} \|\mathbf{M}_0[S,j]\|_2.$$

Remarks: The above quantity is actually the ℓ_2 analog of the ℓ_1 k-th cumulative coherence defined in Gribonval et al. (2015). Also, notice that $\mu_1(\mathbf{M}_0) = \max_{l\neq j} |\mathbf{M}_0[l,j]|$ which is the plain mutual coherence of the reference dictionary.

With the above definitions and as a direct consequence of Lemma 2, we obtain a sufficient condition and a necessary condition for population local identifiability:

Corollary 1 *Under the notations of Theorem 1, we have*

1. *Let $K \ge 2$ and $s \in [\![K-1]\!]$.*
   - *If $\mu_s(\mathbf{M}_0) < 1 - \frac{s-1}{K-1}$, then $\mathbf{D}_0$ is locally identifiable with respect to $L_{SG(s)}$;*
   - *If $\frac{s}{K-1}\nu_s(\mathbf{M}_0) > 1 - \frac{s-1}{K-1}$, then $\mathbf{D}_0$ is not locally identifiable with respect to $L_{SG(s)}$.*
2. *Let $K \ge 2$ and $p \in (0,1)$.*
   - *If $\mu_k(\mathbf{M}_0) < 1-p$, where $k = \lceil p(K-2)+1\rceil$, then $\mathbf{D}_0$ is locally identifiable with respect to $L_{BG(p)}$;*
   - *If $p\nu_k(\mathbf{M}_0) > 1-p$, where $k = p(K-1)$, then $\mathbf{D}_0$ is not locally identifiable with respect to $L_{BG(p)}$.*

Remarks:
(1) In particular, by Lemma 2, if $\mu_1(\mathbf{M}_0) > 1 - \frac{s-1}{K-1}$ or $\mu_1(\mathbf{M}_0) > 1-p$, then $\mathbf{D}_0$ is not locally identifiable.
(2) We can also replace $\frac{s}{K-1}\nu_s(\mathbf{M}_0)$ or $p\nu_k(\mathbf{M}_0)$ by the corresponding lower bound quantities in Lemma 2 which are easier to compute but give weaker necessary conditions.

Comparison with GS. Corollary 1 enables us to compare our local identifiability condition directly with that of GS. For the Bernoulli(p)-Gaussian model, the population version of the sufficient condition for local identifiability by GS is:

$$\mu_{K-1}(\mathbf{M}_0) = \max_{1\le j\le K} \|\mathbf{M}_0[-j,j]\|_2 < 1-p. \tag{7}$$

Note that $\mu_{K-1}(\mathbf{M}_0) \ge \mu_k(\mathbf{M}_0)$ for $k \le K-1$.

Thus, our local identifiability result implies that of GS. Moreover, the quantity $\|\mathbf{M}_0[-j,j]\|_2$ in inequality (7) computes the ℓ_2-norm of the entire $\mathbf{M}_0[-j,j]$ vector and is independent of the sparsity parameter p. On the other hand, in our sufficient condition, $\max_{|S|=k, j\notin S} \|\mathbf{M}_0[S,j]\|_2$ computes the largest ℓ_2-norm of all size-k sub-vectors of $\mathbf{M}_0[-j,j]$. Since $k = \lceil p(K-2)+1\rceil$ is essentially pK, when the random linear coefficients are sparse, the sufficient bound by GS is much more conservative compared to ours.

More concretely, let us consider constant inner-product dictionaries with parameter $\mu > 0$ as in Example 5. The sufficient conditions by GS and by our Corollary 1 are respectively

$$\sqrt{K}\mu \le 1-p \text{ and } \sqrt{pK+1}\mu \le 1-p,$$

showing that the sufficient condition by GS is much more conservative for small value of p. See Figure 1, Appendix Figures B.1 and B.2 for a graphical comparison of the bounds for $K = 10, 20$ and 50.

Local identifiability for sparsity level $O(\mu^{-2})$. For notational convenience, let $\mu = \mu_1(\mathbf{M}_0)$ be the mutual coherence of the reference dictionary. For the s-sparse model, by Lemma 2, $\mu_s(\mathbf{M}_0) \le \sqrt{s}\mu$. Thus the first part of the corollary implies a simpler sufficient condition:

$$\sqrt{s}\mu < 1 - \frac{s-1}{K-1}.$$

From the above inequality, it can be seen that if $1 - \frac{s-1}{K-1} > \delta$ for some $\delta > 0$, the reference dictionary is locally identifiable for sparsity level s up to the order $O(\mu^{-2})$.

Similarly for the Bernoulli(p)-Gaussian model, since for $k = \lceil p(K-2)+1\rceil$,

$$\mu_k(\mathbf{M}_0) \le \sqrt{pK+1}\mu,$$

we have the following sufficient condition for local identifiability:

$$\sqrt{pK+1}\mu \le 1-p.$$

As before, if $1-p > \delta$ for some $\delta > 0$, the reference dictionary is locally identifiable for sparsity level pK up to the order $O(\mu^{-2})$. On the other hand, the same arguments for the condition by GS leads to $K = O(\mu^{-2})$, which, does not take advantage of sparsity.

In addition, by Example 5 and Remark (4) under Lemma 2, we also know that the sparsity requirement $O(\mu^{-2})$ cannot be improved in general.

Our result seems to be the first to demonstrate that $O(\mu^{-2})$ is the optimal order of sparsity level for exact local recovery of a reference dictionary. For a predefined over-complete dictionary, classical results such as Donoho and Elad (2003) and Fuchs (2004) show that basis pursuit recovers an s-sparse linear coefficient vector with sparsity level s

up to the order $O(\mu^{-1})$. For over-complete dictionary learning, Geng et al. (2011) showed that exact local recovery is also possible for s-sparse model with s up to $O(\mu^{-1})$. While our results are only for complete dictionaries, we conjecture that $O(\mu^{-2})$ is also the optimal order of sparsity level for over-complete dictionaries. In fact, Schnass (2015) proved that the response maximization criterion—an alternative formulation of dictionary learning—can approximately recover the over-complete reference dictionary locally with sparsity level s up to $O(\mu^{-2})$. It will be of interest to investigate whether the same sparsity requirement hold for the ℓ_1-minimization dictionary learning (2) in the case of exact local recovery and over-complete dictionaries.

A note on global identifiability. With the notations in Definition 1, we say that the reference dictionary $\mathbf{D}_0$ is *globally identifiable* with respect to $L(\mathbf{D})$ if (1) $\mathbf{D}_0$ is a global minimum of $L(\mathbf{D})$, and (2) for any other dictionary $\mathbf{D}'$ that cannot be transformed to $\mathbf{D}_0$ under any column permutation and sign changes, $L(\mathbf{D}_0) < L(\mathbf{D}')$. It is easy to see that global identifiability implies local identifiability, and so any necessary conditions for local identifiability are also necessary for global identifiability. Sufficient conditions for the global case are much harder to find. However, for dictionaries with orthogonal columns, we can show the following:

Corollary 2 *Suppose that the reference dictionary $\mathbf{D}_0$ is orthogonal, i.e., $\mathbf{M}_0 = \mathbf{I}$.*

1. *$\mathbf{D}_0$ is globally identifiable with respect to $L_{SG(s)}$ if and only if $\frac{s}{K} < 1$.*
2. *$\mathbf{D}_0$ is globally identifiable with respect to $L_{BG(p)}$ if and only if $p < 1$.*

The above result indicates that for orthogonal dictionaries, we have global identifiability as long as the linear coefficients $\boldsymbol{\alpha}_i$'s are not entirely dense. Note that these conditions are exactly the same as in the local identifiability case, see Example 3. Naturally, it is of greater interest to derive global identifiability condition for non-orthogonal dictionaries. Simulation results in Figure 3 of GS demonstrate that for $K = 2$, global identifiability seems to share the same phase transition boundary with local identifiability, i.e., $\mu + p = 1$. Furthermore, the surface plot of the objective function in Figure 2 of GS shows no other spurious local minima. To establish global identifiability for non-orthogonal dictionaries, it is unavoidable to characterize global optima of the non-convex objective function. An on-going work of Y. Wang and the authors (Wang et al.) might help shed light on this challenging problem.

4. Finite Sample Analysis

In this section, we will present finite sample results for local dictionary identifiability. For notational convenience, we first define the following quantities:

$$\mathcal{P}_1(\epsilon, N; \mu, K) = 2\exp\left(-\frac{N\epsilon^2}{108K\mu}\right),$$
$$\mathcal{P}_2(\epsilon, N; p, K) = 2\exp\left(-p\frac{N\epsilon^2}{18p^2K + 9\sqrt{2pK}}\right),$$
$$\mathcal{P}_3(\epsilon, N; p, K) = 3\left(\frac{24}{\epsilon p} + 1\right)^K \exp\left(-p\frac{N\epsilon^2}{360}\right).$$

Recall that $\mathbf{M}_0 = \mathbf{D}_0^T\mathbf{D}_0$ and $\mu_1(\mathbf{M}_0)$ is the mutual coherence of the reference dictionary $\mathbf{D}_0$. The following two theorems give local identifiability conditions under the s-sparse Gaussian model and the Bernoulli-Gaussian model:

Theorem 2 *(Finite sample local identifiability for $SG(s)$) Let $\boldsymbol{\alpha}_i \in \mathbb{R}^K$, $i \in [\![N]\!]$, be i.i.d. $SG(s)$ random vectors with $s \in [\![K-1]\!]$. The signals $\mathbf{x}_i$'s are generated as $\mathbf{x}_i = \mathbf{D}_0\boldsymbol{\alpha}_i$. Assume $0 < \epsilon \le \frac{1}{2}$.*

1. *If*
$$\max_{j\in[\![K]\!]} |||\mathbf{M}_0[-j, j]|||_s^* \le 1 - \frac{s-1}{K-1} - \sqrt{\frac{\pi}{2}}\epsilon,$$
then $\mathbf{D}_0$ is locally identifiable with respect to $L_N(\mathbf{D})$ with probability exceeding
$$1 - K^2\left(\mathcal{P}_1(\epsilon, N; \mu_1(\mathbf{M}_0), K) + \mathcal{P}_2(\epsilon, N; \frac{s}{K}, K) + \mathcal{P}_3(\epsilon, N; \frac{s}{K}, K)\right).$$
2. *If*
$$\max_{j\in[\![K]\!]} |||\mathbf{M}_0[-j, j]|||_s^* \ge 1 - \frac{s-1}{K-1} + \sqrt{\frac{\pi}{2}}\epsilon,$$
then $\mathbf{D}_0$ is not locally identifiable with respect to $L_N(\mathbf{D})$ with probability exceeding
$$1 - K\left(\mathcal{P}_1(\epsilon, N; \mu_1(\mathbf{M}_0), K) + \mathcal{P}_2(\epsilon, N; \frac{s}{K}, K) + \mathcal{P}_3(\epsilon, N; \frac{s}{K}, K)\right).$$

Theorem 3 *(Finite sample local identifiability for $BG(p)$) Let $\boldsymbol{\alpha}_i \in \mathbb{R}^K$, $i \in [\![N]\!]$, be i.i.d. $BG(p)$ random vectors with $p \in (0, 1)$. The signals $\mathbf{x}_i$'s are generated as $\mathbf{x}_i = \mathbf{D}_0\boldsymbol{\alpha}_i$. Let $K_p = K + 2p^{-1}$ and assume $0 < \epsilon \le \frac{1}{2}$.*

1. *If*
$$\max_{j\in[\![K]\!]} |||\mathbf{M}_0[-j, j]|||_p^* \le 1 - p - \sqrt{\frac{\pi}{2}}\epsilon,$$
then $\mathbf{D}_0$ is locally identifiable with respect to $L_N(\mathbf{D})$ with probability exceeding
$$1 - K^2\left(\mathcal{P}_1(\epsilon, N; \mu_1(\mathbf{M}_0), K_p) + \mathcal{P}_2(\epsilon, N; p, K_p) + \mathcal{P}_3(\epsilon, N; p, K)\right).$$
2. *If*
$$\max_{j\in[\![K]\!]} |||\mathbf{M}_0[-j, j]|||_p^* \ge 1 - p + \sqrt{\frac{\pi}{2}}\epsilon,$$
then $\mathbf{D}_0$ is not locally identifiable with respect to $L_N(\mathbf{D})$ with probability exceeding
$$1 - K\left(\mathcal{P}_1(\epsilon, N; \mu_1(\mathbf{M}_0), K_p) + \mathcal{P}_2(\epsilon, N; p, K_p) + \mathcal{P}_3(\epsilon, N; p, K)\right).$$

Remarks: The conditions for finite sample local identifiability are essentially identical to their population counterparts. The main difference is an margin of $\sqrt{\frac{\pi}{2}}\epsilon$ on the RHS of the inequalities. Such a margin appears as a result of our proof techniques: we show that the derivative of L_N is within $O(\epsilon)$ of its expectation and then apply local identifiability results for the population case.

Sample size requirement. The theorems indicate that if the number of signals is a multiple of the following quantity,

$$\text{For } SG(s)\text{: } \frac{1}{\epsilon^2}\max\left\{\mu_1(\mathbf{M}_0)K\log K,\ s\log K,\ \frac{K}{s}K\log\left(\frac{K}{\epsilon s}\right)\right\}$$

$$\text{For } BG(p)\text{: } \frac{1}{\epsilon^2}\max\left\{\mu_1(\mathbf{M}_0)K\log K,\ pK\log K,\ \frac{1}{p}K\log\left(\frac{1}{\epsilon p}\right)\right\}$$

then with high probability we can determine the conditions for local identifiability. Thus, in the worst case, the sample size requirements for the two models are respectively

$$O(\frac{K\log K}{\frac{s}{K}}) \text{ and } O(\frac{K\log K}{p}).$$

Our sample size requirement is similar to that of GS, who shows that $O(\frac{K\log K}{p(1-p)})$ signals is enough for locally recovering an incoherent reference dictionary. Our result indicates the $1-p$ factor in their denominator is not necessary.

The following two corollaries are the finite sample counterparts of Corollary 1.

Corollary 3 *Under the same assumptions of Theorem 2,*

1. *(Sufficient condition for $SG(s)$) If*

$$\mu_s(\mathbf{M}_0) \le 1 - \frac{s-1}{K-1} - \sqrt{\frac{\pi}{2}}\epsilon,$$

then $\mathbf{D}_0$ is locally identifiable with respect to $L_N(\mathbf{D})$, with the same probability bound in the first part of Theorem 2.

2. *(Necessary condition for $SG(s)$) If*

$$\frac{s}{K-1}\nu_s(\mathbf{M}_0) \ge 1 - \frac{s-1}{K-1} + \sqrt{\frac{\pi}{2}}\epsilon,$$

then $\mathbf{D}_0$ is not locally identifiable with respect to $L_N(\mathbf{D})$, with the same probability bound in the second part of Theorem 2.

Corollary 4 *Under the same assumptions of Theorem 3,*

1. *(Sufficient condition for $BG(p)$) Let $k = \lceil p(K-1)+1\rceil$. If*

$$\mu_k(\mathbf{M}_0) \le 1 - p - \sqrt{\frac{\pi}{2}}\epsilon,$$

then $\mathbf{D}_0$ is locally identifiable with respect to $L_N(\mathbf{D})$, with the same probability bound in the first part of Theorem 3.

2. *(Necessary condition for $BG(p)$) Let $k = p(K-1)$. If*

$$p\nu_k(\mathbf{M}_0) \ge 1 - p + \sqrt{\frac{\pi}{2}}\epsilon,$$

then $\mathbf{D}_0$ is not locally identifiable with respect to $L_N(\mathbf{D})$, with the same probability bound in the second part of Theorem 3.

Remarks: As before, denote by $\mu \in [0,1)$ the coherence of the reference dictionary. The above two corollaries indicate that the reference dictionary is locally identifiable with high probability for sparsity level s or pK up to the order $O(\mu^{-2})$.

Proof sketch for Theorem 2 and 3. Similar to the population case, by taking one-sided derivatives of $L_N(\mathbf{D}_t)$ with respect to t at $t=0$ for all smooth $\{\mathbf{D}_t\}_{t\in\mathbb{R}}$, we can derive a sufficient and almost necessary algebraic condition for the reference dictionary $\mathbf{D}_0$ to be a strict local minimum of $L_N(\mathbf{D})$. Using the concentration inequalities in Lemma 3—5, we show that the stochastic terms in the algebraic condition are close to their expectations with high probability. The population results for local identifiability can then be applied. The proofs for the two signal generation models are conceptually the same after establishing Lemma 8 to relate the $|||.|||_p^*$ norm to the $|||.|||_s^*$ norm. The detailed proof can be found in Section A.2.

Comparison with the proof by GS. The key difference between our analysis and that of GS is that we use an alternative but equivalent formulation of dictionary learning. Instead of (2), GS studied the following problem:

$$\min_{\mathbf{D}\in\mathcal{D},\boldsymbol{\alpha}_i} \frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{\alpha}_i\|_1 \qquad (8)$$
$$\text{subject to } \mathbf{x}_i = \mathbf{D}\boldsymbol{\alpha}_i \text{ for all } i \in [\![N]\!].$$

Note that the above formulation optimizes jointly over $\mathbf{D}$ and $\boldsymbol{\alpha}_i$ for $i \in [\![N]\!]$, as opposed to optimizing with respect to the only parameter $\mathbf{D}$ in our case. For complete dictionaries, this formulation is equivalent to the formulation in (2) in the sense that $\hat{\mathbf{D}}$ is a local minimum of (2) if and only if $(\hat{\mathbf{D}}, \hat{\mathbf{D}}^{-1}[\mathbf{x}_1,\ldots,\mathbf{x}_N])$ is a local minimum of (8), see Remark 3.1 of GS. The number of parameters to be estimated in (8) is $O(K^2+KN)$, compared to $O(K^2)$ free parameters in (2). The growing number of parameters make the GS formulation less tractable to analyze under a signal generation model.

GS did not directly study the population case. They first obtained an algebraic condition for local identifiability that is sufficient and almost necessary. However, the condition is convoluted and hard to interpret due to its direct dependence on the signals $\mathbf{x}_i$'s. In order to determine the number of signals required for successful local recovery, they further investigated their condition under the Bernoulli-Gaussian model. During the probabilistic analysis, the sharp algebraic condition was weakened, resulting in a sufficient condition that is far from being necessary.

In contrast, we start with probabilistic generative models. The number of parameters remains constant as N increases. This allows us to study the population problem directly

and to apply concentration inequalities for the finite sample problem. Therefore, studying the optimization problem (2) instead of (8) is the key to establishing an interpretable sufficient and almost necessary local identifiability condition.

5. Conclusions and Future Work

We have established sufficient and almost necessary conditions for local dictionary identifiability under both the s-sparse Gaussian model and the Bernoulli-Gaussian model in the case of noiseless signals and complete dictionaries. For finite samples with a fixed sparsity level, we have shown that as long as the number of $i.i.d.$ signals scales as $O(K \log K)$, with high probability we can determine the local identifiability conditions of a reference dictionary.

There are several directions for future research. In this paper, we focused mainly on the local behaviors of the ℓ_1-norm objective function. As we previously discussed, investigating global identifiability conditions is a natural next step. Simulations in GS suggest a close connection between local and global identifiability. To understand this problem further, we need to characterize global optima of the non-convex objective function. In an on-going work of Y. Wang and the authors, we are developing promising techniques to analyze global properties of ℓ_1-minimization dictionary learning in the non-orthogonal case.

Moreover, one can extend our results to a wider class of sub-Gaussian distributions other than the standard Gaussian distribution considered in this paper. We foresee little technical difficulty for this extension. However, it should be noted that the quantities involved in our local identifiability conditions, i.e., the $|||.|||_s^*$ and $|||.|||_p^*$ norms, are consequences of the standard Gaussian assumption. Under a different distribution, it can be even more computationally challenging to verify the resulting local identifiability conditions.

Finally, it would be also desirable to improve the sufficient condition by Geng et al. (2011) and Gribonval et al. (2015) for over-complete dictionaries and noisy signals. One interesting implication of our results is that local recovery is possible for sparsity level up to the order $O(\mu^{-2})$ for a μ-coherent reference dictionary. We conjecture the same sparsity requirement for the over-complete and/or noisy signal cases. In either scenario, the closed-form expression for the objective function is no longer available. A full characterization of local dictionary identifiability demands novel techniques for analyzing the local behaviors of the objective function.

Acknowledgments

This research is supported in part by the Citadel Fellowship at the Department of Statistics of UCB, NHGRI grant U01HG007031, NSF grants DMS-1107000, CDS&E-MSS 1228246, DMS-1160319 (FRG), ARO grant W911NF-11-1-0114, AFOSR grant FA9550-14-1-0016,and the Center for Science of Information (CSoI), a US NSF Science and Technology Center, under grant agreement CCF-0939370. The authors would like to thank Sivaraman Balakrishnan and Yu Wang for helpful comments on the manuscript.

Appendix A. Proofs

Let $L(\mathbf{D})$ be the dictionary learning objective function and $\{\mathbf{D}_t\}_{t\in\mathbb{R}}$ be the collection of dictionaries $\mathbf{D}_t \in \mathcal{D}$ parameterized by $t \in \mathbb{R}$. By definition, $\{\mathbf{D}_t\}_{t\in\mathbb{R}}$ passes through the reference dictionary $\mathbf{D}_0$ at $t = 0$. Similar to Gribonval and Schnass (2010), to ensure that $\mathbf{D}_0$ is a strict local minimum of $L(\mathbf{D})$, it suffices to have

$$\lim_{t\downarrow 0^+} \frac{L(\mathbf{D}_t) - L(\mathbf{D}_0)}{t} > 0 \text{ and } \lim_{t\uparrow 0^-} \frac{L(\mathbf{D}_t) - L(\mathbf{D}_0)}{t} < 0,$$

for all $\{\mathbf{D}_t\}_{t\in\mathbb{R}}$ that is a smooth function of t with rate of approaching $\mathbf{D}_0$ bounded away from zero. On the other hand, if either of the above strict inequalities holds in the reversed direction for some smooth $\{\mathbf{D}_t\}_{t\in\mathbb{R}}$, then $\mathbf{D}_0$ is not a local minimum of $L(\mathbf{D})$.

Since $\mathbf{D}_0$ is full rank by assumption, the minimum eigenvalue of $\mathbf{M}_0 = \mathbf{D}_0^T\mathbf{D}_0$ is strictly greater than zero. By continuity of the minimum eigenvalue of $\mathbf{D}_t^T\mathbf{D}_t$ (see e.g., the Bauer-Fike Theorem), $\mathbf{D}_t$ should also be full rank when t is sufficiently small. Thus without loss of generality we can only consider full rank dictionaries $\mathbf{D}_t$. For any full rank $\mathbf{D} \in \mathcal{D}$, there is an invertible matrix $\mathbf{A} \in \mathbb{R}^{K\times K}$ such that $\mathbf{D} = \mathbf{D}_0\mathbf{A}$. For any $k \in [\![K]\!]$, by the constraint $\|\mathbf{D}[,k]\|_2 = 1$, $\mathbf{A}[,k]^T\mathbf{M}_0\mathbf{A}[,k] = 1$. Define the set for all such $\mathbf{A}$'s as

$$\mathcal{A} = \{\mathbf{A} \in \mathbb{R}^{K\times K} : \mathbf{A} \text{ is invertible and } \mathbf{A}[,k]^T\mathbf{M}_0\mathbf{A}[,k] = 1 \text{ for all } k \in [\![K]\!]\}. \quad (9)$$

It follows immediately that the set $\{\mathbf{D}_0\mathbf{A} : \mathbf{A} \in \mathcal{A}\}$ is the collection of $\mathbf{D} \in \mathcal{D}$ such that $\mathbf{D}$ is full rank. Thus, to ensure that $\mathbf{D}_0$ is a strict local minimum of $L(\mathbf{D})$, it suffices to show

$$\Delta^+(L, \{\mathbf{A}_t\}_t) := \lim_{t\downarrow 0^+} \frac{L(\mathbf{D}_0\mathbf{A}_t) - L(\mathbf{D}_0)}{t} > 0, \quad (10)$$

$$\Delta^-(L, \{\mathbf{A}_t\}_t) := \lim_{t\uparrow 0^-} \frac{L(\mathbf{D}_0\mathbf{A}_t) - L(\mathbf{D}_0)}{t} < 0, \quad (11)$$

for all smooth functions $\{\mathbf{A}_t\}_{t\in\mathbb{R}}$ with $\mathbf{A}_t \in \mathcal{A}$, $\mathbf{A}_0 = \mathbf{I}$ and nonzero derivative at $t = 0$. In addition, to demonstrate that $\mathbf{D}_0$ is not a local minimum of $L(\mathbf{D})$, it suffices to have (10) or (11) to hold in the reversed direction for some $\{\mathbf{A}_t\}_t$ with the aforementioned properties. We will be using this characterization of local minimum to prove local identifiability results for both the population case and the finite sample case.

A.1 Proofs of the Population Results

A.1.1 Proof of Lemma 1

Proof Since $\mathbb{E}||\mathbf{H}\boldsymbol{\alpha}_1||_1 = \sum_{j=1}^K \mathbb{E}|\mathbf{H}[j,]\boldsymbol{\alpha}_1|$, it suffices to compute $\mathbb{E}|\mathbf{H}[j,]\boldsymbol{\alpha}_1|$. Let S be any nonempty subset of $[\![K]\!]$. Recall that the random variable $\mathbf{S}_1 \subset [\![K]\!]$ denotes the support of random coefficient $\boldsymbol{\alpha}_1$. Conditioning on the event $\{\mathbf{S}_1 = S\}$, the random variable $\mathbf{H}[j,]\boldsymbol{\alpha}_1$ follows a normal distribution with mean 0 and standard deviation $\|\mathbf{H}[j,S]\|_2$. Hence

$$\mathbb{E}\,|\mathbf{H}[j,]\boldsymbol{\alpha}_1| = \mathbb{E}[\mathbb{E}[|\mathbf{H}[j,]\boldsymbol{\alpha}_1|\,|\mathbf{S}_1]] = \sqrt{\frac{2}{\pi}}\mathbb{E}\|\mathbf{H}[j,\mathbf{S}_1]\|_2.$$

(1) Under the s-sparse Gaussian model, $\mathbb{P}(\mathbf{S}_1 = S) = \binom{K}{s}^{-1}$ for any $|S| = s$. Thus we have

$$\mathbb{E}\|\mathbf{H}[j,\mathbf{S}_1]\|_2 = \binom{K}{s}^{-1} \sum_{S:|S|=s} \|\mathbf{H}[j,S]\|_2 = \frac{s}{K}|||\mathbf{H}[j,]|||_s.$$

Hence the objective function for the s-sparse Gaussian model is

$$L_{SG(s)}(\mathbf{D}) = \sum_{j=1}^{K} \mathbb{E}\,|\mathbf{H}[j,]\boldsymbol{\alpha}_1| = \sqrt{\frac{2}{\pi}}\frac{s}{K}\sum_{j=1}^{K}|||\mathbf{H}[j,]|||_s.$$

In particular, for $s = K$, $|||\mathbf{H}[j,]|||_K = \|\mathbf{H}[j,]\|_2$ and so

$$L_{SG(s)}(\mathbf{D}) = \sqrt{\frac{2}{\pi}}\sum_{j=1}^{K}\|\mathbf{H}[j,]\|_2.$$

(2) Under the Bernoulli(p)-Gaussian model, $\mathbb{P}(\mathbf{S}_1 = S) = p^{|S|}(1-p)^{K-|S|}$. So we have

$$\begin{aligned}\mathbb{E}[\|\mathbf{H}[j,\mathbf{S}_1]\|_2] &= \sum_{k=1}^{K}\sum_{S:|S|=k} p^k(1-p)^{K-k}\|\mathbf{H}[j,S]\|_2 \\ &= p\sum_{k=0}^{K-1}\texttt{pbinom}(k;K-1,p)|||\mathbf{H}[j,]|||_{k+1}.\end{aligned}$$

Therefore for $p \in (0,1)$, the objective function under the Bernoulli-Gaussian model is

$$L_{BG(p)}(\mathbf{D}) = \sum_{j=1}^{K}\mathbb{E}\,|\mathbf{H}[j,]\boldsymbol{\alpha}_1| = \sqrt{\frac{2}{\pi}}p\sum_{j=1}^{K}|||\mathbf{H}[j,]|||_p.$$

Finally, if $p = 1$, we have

$$L_{BG(p)}(\mathbf{D}) = \sqrt{\frac{2}{\pi}}\sum_{j=1}^{K}\|\mathbf{H}[j,]\|_2.$$

■

A.1.2 Proof of Theorem 1

Proof (1) Let us first consider the s-sparse Gaussian model. By (10) and (11), to ensure that $\mathbf{D}_0$ is a local minimum of $L_{SG(s)}(\mathbf{D})$, it suffices to show

$$\Delta^+(L_{SG(s)},\{\mathbf{A}_t\}_t) > 0 \text{ and } \Delta^-(L_{SG(s)},\{\mathbf{A}_t\}_t) < 0, \tag{12}$$

for all smooth functions $\{\mathbf{A}_t\}_t$ with $\mathbf{A}_t \in \mathcal{A}$, $\mathbf{A}_0 = \mathbf{I}$ and nonzero derivative at $t = 0$. Note that by Lemma 1,

$$\Delta^+(L_{SG(s)},\{\mathbf{A}_t\}_t) = \sqrt{\frac{2}{\pi}}\frac{s}{K}\sum_{j=1}^{K}\lim_{t\downarrow 0^+}\frac{1}{t}\left(\left|\left|\left|\mathbf{A}_t^{-1}[j,]\right|\right|\right|_s - |||\mathbf{I}[j,]|||_s\right). \tag{13}$$

For each $j \in [\![K]\!]$, we have

$$\binom{K-1}{s-1}\left|\left|\left|\mathbf{A}_t^{-1}[j,]\right|\right|\right|_s = \sum_{S:|S|=s,j\in S}\|\mathbf{A}_t^{-1}[j,S]\|_2 + \sum_{S:|S|=s,j\notin S}\|\mathbf{A}_t^{-1}[j,S]\|_2 \tag{14}$$

Denote by $\dot{\mathbf{A}}_0 \in \mathbb{R}^{K\times K}$ the derivative of $\{\mathbf{A}_t\}_t$ at $t = 0$. Since $\mathbf{A}_t \in \mathcal{A}$ for all $t \in \mathbb{R}$, it can be shown that

$$\mathbf{M}_0[,k]^T\dot{\mathbf{A}}_0[,k] = 0 \quad \text{for all } k \in [\![K]\!]. \tag{15}$$

By (15), we have

$$\dot{\mathbf{A}}_0[j,j] = -\sum_{i\neq j}\mathbf{M}_0[i,j]\dot{\mathbf{A}}_0[i,j] \text{ for all } j \in [\![K]\!]. \tag{16}$$

Now notice that

$$\left.\frac{d\mathbf{A}_t^{-1}}{dt}\right|_{t=0} = -\mathbf{A}_0^{-1}\dot{\mathbf{A}}_0\mathbf{A}_0^{-1} = -\dot{\mathbf{A}}_0. \tag{17}$$

Combining the above equality with Lemma 14 and 15, we have

$$\lim_{t\downarrow 0^+}\frac{1}{t}\left(\|\mathbf{A}_t^{-1}[j,S]\|_2 - \|\mathbf{I}[j,S]\|_2\right) = \begin{cases} -\dot{\mathbf{A}}_0[j,j] & \text{if } j \in S \\ \|\dot{\mathbf{A}}_0[j,S]\|_2 & \text{if } j \notin S\end{cases}$$

Therefore

$$\lim_{t\downarrow 0^+}\frac{1}{t}\left(\left|\left|\left|\mathbf{A}_t^{-1}[j,]\right|\right|\right|_s - |||\mathbf{I}[j,]|||_s\right) = -\dot{\mathbf{A}}_0[j,j] + \binom{K-1}{s-1}^{-1}\sum_{S:|S|=s,j\notin S}\|\dot{\mathbf{A}}_0[j,S]\|_2. \tag{18}$$

Combining (13), (14), (16) and (18), we have

$$\begin{aligned}\sqrt{\frac{\pi}{2}}\frac{K}{s}\Delta^+(L_{SG(s)},\{\mathbf{A}_t\}_t) &= -\sum_{j=1}^{K}\dot{\mathbf{A}}_0[j,j] + \binom{K-1}{s-1}^{-1}\sum_j\sum_{S:|S|=s,j\notin S}||\dot{\mathbf{A}}_0[j,S]||_2 \\ &= \sum_{j=1}^{K}\Big(\sum_{i\neq j}\mathbf{M}_0[i,j]\dot{\mathbf{A}}_0[j,i] + \binom{K-1}{s-1}^{-1}\sum_{S:|S|=s,j\notin S}\|\dot{\mathbf{A}}_0[j,S]\|_2\Big).\end{aligned}$$

Similarly, one can show

$$\sqrt{\frac{\pi}{2}}\frac{K}{s}\Delta^-(L_{SG(s)},\{\mathbf{A}_t\}_t) = \sum_{j=1}^{K}\Big(\sum_{i\neq j}\mathbf{M}_0[i,j]\dot{\mathbf{A}}_0[j,i] - \binom{K-1}{s-1}^{-1}\sum_{S:|S|=s,j\notin S}\|\dot{\mathbf{A}}_0[j,S]\|_2\Big).$$

Thus for $s \in [\![K-1]\!]$, to establish (12) it suffices to require for each $j \in [\![K]\!]$,

$$\Big|\sum_{i\neq j}\mathbf{M}_0[i,j]\dot{\mathbf{A}}_0[j,i]\Big| < \frac{K-s}{K-1}\binom{K-2}{s-1}^{-1}\sum_{S:|S|=s,j\notin S}\|\dot{\mathbf{A}}_0[j,S]\|_2 = \frac{K-s}{K-1}\left|\left|\left|\dot{\mathbf{A}}_0[j,-j]\right|\right|\right|_s. \tag{19}$$

for any $\dot{\mathbf{A}}_0$ such that $\dot{\mathbf{A}}_0[j,-j] \neq 0$. Since $\dot{\mathbf{A}}_0[j,i]$ is a free variable for $i \neq j$, (19) is equivalent to

$$\left|\mathbf{M}_0[-j,j]^T\mathbf{w}\right| < \frac{K-s}{K-1},$$

for all $\mathbf{w} \in \mathbb{R}^{K-1}$ such that $\|\mathbf{w}\|_s = 1$. Thus by the definition of the dual norm, it suffices to have

$$\|\mathbf{M}_0[-j,j]\|_s^* = \sup_{\|\mathbf{w}\|_s=1} \left|\mathbf{M}_0[-j,j]^T\mathbf{w}\right| < \frac{K-s}{K-1}.$$

Therefore, the condition

$$\max_{1\leq j\leq K} \|\mathbf{M}_0[-j,j]\|_s^* < \frac{K-s}{K-1} = 1 - \frac{s-1}{K-1}. \tag{20}$$

is sufficient for $\mathbf{D}_0$ to be locally identifiable with respect to the objective function $L_{SG(s)}$.

Similarly, one can check that if the reversed strict inequality in (20) holds, $\mathbf{D}_0$ is not a local minimum of $L_{SG(s)}(\mathbf{D})$. Thus we complete the proof for the s-sparse model.

(2) Now consider the Bernoulli(p)-Gaussian model for $p \in (0,1)$. First of all, note that we have

$$\begin{aligned}
&\sqrt{\frac{\pi}{2}}\frac{1}{p}\Delta^{\pm}(L_{BG(p)}, \{\mathbf{A}_t\}_t) = \sum_{j=1}^{K} \lim_{t\to 0^{\pm}} \frac{1}{t}\left(\|\|\mathbf{A}_t^{-1}[j,]\|\|_p - \|\|\mathbf{I}[j,]\|\|_p\right)\\
=&\sum_{j=1}^{K}\left(\sum_{i\neq j}\mathbf{M}_0[i,j]\dot{\mathbf{A}}_0[j,i] \pm (1-p)\sum_{k=0}^{K-2} p^k(1-p)^{K-2-k}\sum_{S:|S|=k+1, j\notin S}\|\dot{\mathbf{A}}_0[j,S]\|_2\right)\\
=&\sum_{j=1}^{K}\left(\dot{\mathbf{A}}_0[j,-j]^T\mathbf{M}_0[-j,j] \pm (1-p)\sum_{k=0}^{K-2}\texttt{pbinom}(k;K-2,p)\left\|\left|\dot{\mathbf{A}}_0[j,-j]\right|\right\|_{k+1}\right)\\
=&\sum_{j=1}^{K}\left(\dot{\mathbf{A}}_0[j,-j]^T\mathbf{M}_0[-j,j] \pm (1-p)\left\|\left|\dot{\mathbf{A}}_0[j,-j]\right|\right\|_p\right).
\end{aligned}$$

Thus, similar to the s-sparse Gaussian case, it can be shown that a sufficient condition for local identifiability is

$$\left|\mathbf{M}_0[-j,j]^T\mathbf{w}\right| < 1-p,$$

for all $j \in [\![K]\!]$ and all $\mathbf{w} \in \mathbb{R}^{K-1}$ such that $\|\mathbf{w}\|_p = 1$. The above condition is equivalent to

$$\max_{1\leq j\leq K} \|\mathbf{M}_0[-j,j]\|_p^* < 1-p.$$

The rest of the proof can be proceeded as in the case of the s-sparse Gaussian model.

(3) Let us consider the non-sparse case where $s = K$ or $p = 1$. In this case, since the objective functions are the same under both models (see Theorem 1), we only need to consider the s-sparse Gaussian model. If $s = K$, the RHS quantity in Inequality (19) is zero. Thus, the reference dictionary is not locally identifiable if

$$\left|\mathbf{M}_0[-j,j]^T\mathbf{w}\right| > 0,$$

for some $j \in [\![K]\!]$ and $\mathbf{w} \in \mathbb{R}^{K-1}$. Thus, if $\mathbf{M}_0$ is not the identity matrix, or equivalently, if the reference dictionary $\mathbf{D}_0$ is not orthogonal, $\mathbf{D}_0$ is not locally identifiable.

Next, let us deal with the case where $\mathbf{D}_0$ is orthogonal. Let $\mathbf{D} \in \mathcal{D}$ be a full rank dictionary and $\mathbf{W} = \mathbf{D}^{-1}$. Since $\mathbf{D}_0$ is orthogonal, $\|\mathbf{W}[j,]\mathbf{D}_0\|_2 = \|\mathbf{W}[j,]\|_2$. By the fact that $\mathbf{WD} = \mathbf{I}$ and $\|\mathbf{D}[,j]\|_2 = 1$, we have $1 = \mathbf{W}[j,]\mathbf{D}[,j] \leq \|\mathbf{W}[j,]\|_2\|\mathbf{D}[,j]\|_2 = \|\mathbf{W}[j,]\|_2$, where the equality holds if and only if $\mathbf{W}[j,]^T = \pm\mathbf{D}[,j]$.

Under the K-sparse Gaussian model,

$$L_{SG(K)}(\mathbf{D}) = \sqrt{\frac{2}{\pi}}\sum_{j=1}^{K}\|\mathbf{W}[j,]\mathbf{D}_0\|_2 = \sqrt{\frac{2}{\pi}}\sum_{j=1}^{K}\|\mathbf{W}[j,]\|_2 \geq \sqrt{\frac{2}{\pi}}K = L_{SG(K)}(\mathbf{D}_0),$$

where the equality holds for any $\mathbf{D}$ such that $\mathbf{D}^T\mathbf{D} = \mathbf{I}$. Thus, $L_{SG(K)}(\mathbf{D}_0) = L_{SG(K)}(\mathbf{D}_0\mathbf{U})$ for any orthogonal matrix $\mathbf{U} \in \mathbb{R}^{K\times K}$, i.e., the objective function remains the same as we rotate $\mathbf{D}_0$. Therefore, $\mathbf{D}_0$ is not a strict local minimum of $L_{SG(K)}$.

In conclusion, $\mathbf{D}_0$ is not locally identifiable when $s = K$ or $p = 1$. ■

A.1.3 Proof of Corollary 2

Proof Let $\mathbf{D} \in \mathcal{D}$ be a full rank dictionary and $\mathbf{W} = \mathbf{D}^{-1}$. Under the K-sparse Gaussian model, $\mathbf{D}_0$ is not locally identifiable by Theorem 1. Hence it is not a strict local minimum of $L_{SG(K)}(\mathbf{D})$ and cannot be a strict global minimum.

Now suppose $s < K$, by Lemma 1 and 7,

$$L_{SG(s)}(\mathbf{D}) = \sqrt{\frac{2}{\pi}}\frac{s}{K}\sum_{j=1}^{K}\|\|\mathbf{W}[j,]\mathbf{D}_0\|\|_s \geq \sqrt{\frac{2}{\pi}}\frac{s}{K}\sum_{j=1}^{K}\|\mathbf{W}[j,]\mathbf{D}_0\|_2 \geq \sqrt{\frac{2}{\pi}}s = L_{SG(s)}(\mathbf{D}_0).$$

So $\mathbf{D}_0$ is a global minimum. Next we will show $\mathbf{D}_0$ is the only strict global minimum up to column permutation and sign changes. In the above formula, for the first equality to hold, we have $\|\|\mathbf{W}[j,]\mathbf{D}_0\|\|_s = \|\mathbf{W}[j,]\mathbf{D}_0\|_2$ for all j, implying that the vector $\mathbf{W}[j,]\mathbf{D}_0$ has at most one nonzero entry, see Lemma 7. For the second inequality to hold, $\mathbf{W}[j,]^T = \pm\mathbf{D}[,j]$, see arguments in Part (3) of the Theorem 1 proof. Combining these two conditions, $\|\mathbf{W}[j,]\mathbf{D}_0\|_2 = \|\mathbf{D}[j,]\|_2 = 1$ and so the nonzero entry can only be ± 1. Therefore, $\mathbf{WD}_0$ is an identity matrix after proper column permutation and sign changes.

The proof for the Bernoulli-Gaussian model is similar and hence omitted. ■

A.2 Proofs of the Finite Sample Results: Theorem 2 and Theorem 3

Proof We will first recall the signal generation procedure in Section 2. Let $\mathbf{z}$ be a K-dimensional standard Gaussian vector, and $\boldsymbol{\xi} \in \{0,1\}^K$ be either an s-sparse random vector or a Bernoulli random vector with probability p. Let $\mathbf{z}_1,\ldots,\mathbf{z}_N$ and $\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_N$ be identical and independent copies of $\mathbf{z}$ and $\boldsymbol{\xi}$ respectively. For each $i \in [\![N]\!]$ and $j \in [\![K]\!]$, define

$\boldsymbol{\alpha}_i[j] = \mathbf{z}_i[j]\boldsymbol{\xi}_i[j]$. For $S \subset [\![K]\!]$, define

$$\chi_i(S) = \begin{cases} 1 & \text{if } \boldsymbol{\xi}_i[k] = 1 \text{ for all } k \in S \text{ and } \boldsymbol{\xi}_i[k] = 0 \text{ for all } k \in S^c, \\ 0 & \text{otherwise.} \end{cases}$$

As in the population case, in the following analysis we will work with full rank dictionaries. First of all, notice that

$$l(\mathbf{D}, \mathbf{x}_i) = \|\mathbf{D}^{-1}\mathbf{x}_i\|_1 = \|\mathbf{D}^{-1}\mathbf{D}_0\boldsymbol{\alpha}_i\|_1 = \sum_{j=1}^{K} \left|\mathbf{A}^{-1}[j,]\boldsymbol{\alpha}_i\right| = \sum_{j=1}^{K}\sum_{k=1}^{K}\left(\sum_{S:|S|=k} |\mathbf{A}^{-1}[j,S]\mathbf{z}_i[S]|\chi_i(S)\right).$$

Next, we have

$$\begin{aligned}\Delta^+(l(.,\mathbf{x}_i),\{\mathbf{A}_t\}_t) &= \lim_{t\downarrow 0^+}\frac{1}{t}\left(l(\mathbf{D}_0\mathbf{A}_t,\mathbf{x}_i) - l(\mathbf{D}_0,\mathbf{x}_i)\right) \\ &= \sum_{j=1}^{K}\Bigg(-\sum_{k=1}^{K}\sum_{S:j\in S,|S|=k}\dot{\mathbf{A}}_0[j,j]|\mathbf{z}_i[j]|\chi_i(S) \\ &\quad - \mathbf{sgn}(\mathbf{z}_i[j])\sum_{k=2}^{K}\sum_{S:j\in S,|S|=k}\sum_{l\in S,l\neq j}\dot{\mathbf{A}}_0[j,l]\mathbf{z}_i[l]\chi_i(S) \\ &\quad + \sum_{k=1}^{K-1}\sum_{S:j\notin S,|S|=k}|\dot{\mathbf{A}}_0[j,S]\mathbf{z}_i[S]|\chi_i(S)\Bigg). \end{aligned} \tag{21}$$

Here $\mathbf{sgn}(x)$ is the sign function of $x \in \mathbb{R}$ such that $\mathbf{sgn}(x) = 1$ for $x > 0$, $\mathbf{sgn}(x) = -1$ for $x < 0$ and $\mathbf{sgn}(x) = 0$ for $x = 0$. By (16), the first term in (21) can be rearranged as follows

$$\begin{aligned}-\sum_{j=1}^{K}|\mathbf{z}_i[j]|\sum_{k=1}^{K}\sum_{S:j\in S,|S|=k}\dot{\mathbf{A}}_0[j,j]\chi_i(S) &= \sum_{j=1}^{K}|\mathbf{z}_i[j]|\sum_{k=1}^{K}\sum_{S:j\in S,|S|=k}\sum_{l\neq j}\mathbf{M}_0[l,j]\dot{\mathbf{A}}_0[l,j]\chi_i(S) \\ &= \sum_{j=1}^{K}\sum_{l\neq j}\mathbf{M}_0[j,l]\dot{\mathbf{A}}_0[j,l]\left(|\mathbf{z}_i[l]|\sum_{k=1}^{K}\sum_{S:l\in S,|S|=k}\chi_i(S)\right).\end{aligned}$$

The second term in (21) can be rewritten as

$$-\sum_{j=1}^{K}\mathbf{sgn}(\mathbf{z}_i[j])\times\sum_{l\neq j}(\dot{\mathbf{A}}_0[j,l]\mathbf{z}_i[l])\times\sum_{k=2}^{K}\sum_{S:\{j,l\}\in S,|S|=k}\chi_i(S).$$

For $j, l \in [\![K]\!]$ such that $j \neq l$, define the following quantities

$$\mathbf{F}_i[l,j] = \mathbf{M}_0[j,l]|\mathbf{z}_i[l]|\sum_{k=1}^{K}\sum_{S:l\in S,|S|=k}\chi_i(S), \tag{22}$$

$$\mathbf{G}_i[l,j] = \mathbf{sgn}(\mathbf{z}_i[j])\mathbf{z}_i[l]\sum_{k=2}^{K}\sum_{S:\{j,l\}\in S,|S|=s}\chi_i(S), \tag{23}$$

whereas $\mathbf{F}[j,j] = \mathbf{G}[j,j] = 0$. For each $j \in [\![K]\!]$, also define

$$\mathbf{t}_i[j](\mathbf{w}) = \sum_{k=1}^{K-1}\sum_{S:j\notin S,|S|=k}|\mathbf{w}[S]^T\mathbf{z}_i[S]|\chi_i(S). \tag{24}$$

Let $\bar{\mathbf{F}}$, $\bar{\mathbf{G}}$ and $\bar{\mathbf{t}}$ be the sample average of $\mathbf{F}_i$, $\mathbf{G}_i$ and $\mathbf{t}_i$ respectively. With the definitions (22)—(24), we have

$$\begin{aligned}\Delta^+(L_N,\{\mathbf{A}_t\}_t) &= \frac{1}{N}\sum_{i=1}^{N}\Delta^+(l(.,\mathbf{x}_i),\{\mathbf{A}_t\}_t) \\ &= \sum_{j=1}^{K}\frac{1}{N}\sum_{i=1}^{N}\left(\dot{\mathbf{A}}_0[j,]\mathbf{F}_i[,j] + \dot{\mathbf{A}}_0[j,]\mathbf{G}_i[,j] + \mathbf{t}_i[j](\dot{\mathbf{A}}_0[j,])\right) \\ &= \sum_{j=1}^{K}\left(\dot{\mathbf{A}}_0[j,]\bar{\mathbf{F}}[,j] - \dot{\mathbf{A}}_0[j,]\bar{\mathbf{G}}[,j] + \bar{\mathbf{t}}[j](\dot{\mathbf{A}}_0[j,])\right)\end{aligned}$$

On the other hand,

$$\Delta^-(L_N,\{\mathbf{A}_t\}_t) = \sum_{j=1}^{K}\left(\dot{\mathbf{A}}_0[j,]\bar{\mathbf{F}}[,j] - \dot{\mathbf{A}}_0[j,]\bar{\mathbf{G}}[,j] - \bar{\mathbf{t}}[j](\dot{\mathbf{A}}_0[j,])\right).$$

Now for $j \in [\![K]\!]$, $s \in [\![K-1]\!]$ and $p \in (0,1)$, define

$$\mathcal{E}_j(s) = \{\mathbf{w} \in \mathbb{R}^K, \|\mathbf{w}[-j]\|_s = 1, \mathbf{w}[j] = 0\},$$

$$\mathcal{F}_j(p) = \{\mathbf{w} \in \mathbb{R}^K, \|\mathbf{w}[-j]\|_p = 1, \mathbf{w}[j] = 0\}.$$

Thus to ensure that $\mathbf{D}_0$ is a local minimum, it suffices to have for each $j \in [\![K]\!]$,

$$H_j(\mathbf{w}) \coloneqq \left|\mathbf{w}^T\bar{\mathbf{F}}[,j] - \mathbf{w}^T\bar{\mathbf{G}}[,j]\right| - \bar{\mathbf{t}}[j](\mathbf{w}) < 0,$$

for all $\mathbf{w} \in \mathcal{E}_j(s)$ for the s-sparse Gaussian model or all $\mathbf{w} \in \mathcal{F}_j(p)$ for the Bernoulli(p)-Gaussian model.

(1) For the s-sparse Gaussian model, let $j \in [\![K]\!]$ and define

$$h_j(\mathbf{w}) = \sqrt{\frac{2}{\pi}}\frac{s}{K}\left(\left|\mathbf{w}^T\mathbf{M}_0[,j]\right| - \frac{K-s}{K-1}\right),$$

which can be thought of as the expected value of $H_j(\mathbf{w})$. Note that by triangle inequality,

$$\begin{aligned}&\sup_{\mathbf{w}\in\mathcal{E}_j(s)}|H_j(\mathbf{w}) - h_j(\mathbf{w})| \\ \leq &\sup_{\mathbf{w}\in\mathcal{E}_j(s)}\left|\mathbf{w}^T\left(\bar{\mathbf{F}}[,j] - \sqrt{\frac{2}{\pi}}\frac{s}{K}\mathbf{M}_0[,j]\right)\right| + \sup_{\mathbf{w}\in\mathcal{E}_j(s)}\left|\mathbf{w}^T\bar{\mathbf{G}}[,j]\right| + \sup_{\mathbf{w}\in\mathcal{E}_j(s)}\left|\bar{\mathbf{t}}[j](\mathbf{w}) - \sqrt{\frac{2}{\pi}}\frac{s}{K}\frac{K-s}{K-1}\right| \\ = &\left\|\left\|\bar{\mathbf{F}}[-j,j] - \sqrt{\frac{2}{\pi}}\frac{s}{K}\mathbf{M}_0[-j,j]\right\|\right\|_s^* + \left\|\left\|\bar{\mathbf{G}}[-j,j]\right\|\right\|_s^* + \sup_{\mathbf{w}\in\mathcal{E}_j(s)}\left|\bar{\mathbf{t}}[j](\mathbf{w}) - \sqrt{\frac{2}{\pi}}\frac{s}{K}\frac{K-s}{K-1}\right|.\end{aligned} \tag{25}$$

Thus, $\sup_{\mathbf{w}\in\mathcal{E}_j(s)}|H_j(\mathbf{w})-h_j(\mathbf{w})|>\frac{s}{K}\epsilon$ implies at least one of the three terms on the RHS is greater than $\frac{s}{K}\frac{\epsilon}{3}$. Using a union bound and by Lemma 3—5, we have

$$\begin{aligned}\mathbb{P}\left\{\sup_{\mathbf{w}\in\mathcal{E}_j(s)}|H_j(\mathbf{w})-h_j(\mathbf{w})|>\frac{s}{K}\epsilon\right\}&\le 2K\exp\left(-\frac{N\epsilon^2}{108K\|\mathbf{M}_0[-j,j]\|_\infty}\right)\\&\quad+2K\exp\left(-\frac{s}{K}\frac{N\epsilon^2}{18(s/K)s+9\sqrt{2s}}\right)\\&\quad+3\left(\frac{24K}{\epsilon s}+1\right)^K\exp\left(-\frac{s}{K}\frac{N\epsilon^2}{360}\right).\end{aligned}\tag{26}$$

It is easy to see that the event $\left\{\sup_{\mathbf{w}\in\mathcal{E}_j(s)}|H_j(\mathbf{w})-h_j(\mathbf{w})|\le\frac{s}{K}\epsilon\right\}$ implies

$$\sup_{\mathbf{w}\in\mathcal{E}_j(s)}h_j(\mathbf{w})-\frac{s}{K}\epsilon\le\sup_{\mathbf{w}\in\mathcal{E}_j(s)}H_j(\mathbf{w})\le\sup_{\mathbf{w}\in\mathcal{E}_j(s)}h_j(\mathbf{w})+\frac{s}{K}\epsilon.\tag{27}$$

On the other hand,

$$\sup_{\mathbf{w}\in\mathcal{E}_j(s)}h_j(\mathbf{w})=\sqrt{\frac{2}{\pi}}\frac{s}{K}\left(|||\mathbf{M}_0[-j,j]|||_s^*-\frac{K-s}{K-1}\right).$$

Thus, if $|||\mathbf{M}_0[-j,j]|||_s^*<\frac{K-s}{K-1}-\sqrt{\frac{\pi}{2}}\epsilon$, $\sup_{\mathbf{w}\in\mathcal{E}_j(s)}H_j(\mathbf{w})<0$ except with probability at most the bound in (26). To ensure $\mathbf{D}_0$ to be a local minimum, it suffices to have $\sup_{\mathbf{w}\in\mathcal{E}_j(s)}H_j(\mathbf{w})<0$ for all $j\in[\![K]\!]$. Thus, if $|||\mathbf{M}_0[-j,j]|||_s^*<\frac{K-s}{K-1}-\sqrt{\frac{\pi}{2}}\epsilon$ for all $j\in[\![K]\!]$, we have

$$\begin{aligned}\mathbb{P}\{\mathbf{D}_0\text{ is locally identifiable}\}&\ge\mathbb{P}\left\{\max_j\sup_{\mathbf{w}\in\mathcal{E}_j(s)}H_j(\mathbf{w})<0\right\}\\&\ge1-\mathbb{P}\left\{\max_j\sup_{\mathbf{w}\in\mathcal{E}_j(s)}H_j(\mathbf{w})\ge0\right\}\\&\ge1-\sum_{j=1}^K\mathbb{P}\left\{\sup_{\mathbf{w}\in\mathcal{E}_j(s)}H_j(\mathbf{w})\ge0\right\}\\&\ge1-\sum_{j=1}^K\mathbb{P}\left\{\sup_{\mathbf{w}\in\mathcal{E}_j(s)}|H_j(\mathbf{w})-h_j(\mathbf{w})|>\frac{s}{K}\epsilon\right\}\\&\ge1-2K^2\exp\left(-\frac{N\epsilon^2}{108K\max_{l\neq j}|\mathbf{M}_0[l,j]|}\right)\\&\quad-2K^2\exp\left(-\frac{s}{K}\frac{N\epsilon^2}{18(s/K)s+9\sqrt{2s}}\right)\\&\quad-3K\left(\frac{24K}{\epsilon s}+1\right)^K\exp\left(-\frac{s}{K}\frac{N\epsilon^2}{360}\right).\end{aligned}$$

On the other hand, to ensure $\mathbf{D}_0$ is not locally identifiable with high probability, it suffices to have $|||\mathbf{M}_0[-j,j]|||_s^*>\frac{K-s}{K-1}+\sqrt{\frac{\pi}{2}}\epsilon$ for some $j\in[\![K]\!]$. Indeed, under that condition, the

LHS inequality in (27) implies $\sup_{\mathbf{w}\in\mathcal{E}_j(s)}H_j(\mathbf{w})>0$. Therefore

$$\begin{aligned}\mathbb{P}\{\mathbf{D}_0\text{ is not locally identifiable}\}&\ge\mathbb{P}\left\{\sup_{\mathbf{w}\in\mathcal{E}_j(s)}H_j(\mathbf{w})>0\right\}\\&\ge1-\mathbb{P}\left\{\sup_{\mathbf{w}\in\mathcal{E}_j(s)}H_j(\mathbf{w})\le0\right\}\\&\ge1-\mathbb{P}\left\{\sup_{\mathbf{w}\in\mathcal{E}_j(s)}|H_j(\mathbf{w})-h_j(\mathbf{w})|>\frac{s}{K}\epsilon\right\}\\&\ge1-2K\exp\left(-\frac{N\epsilon^2}{108K\|\mathbf{M}_0[-j,j]\|_\infty}\right)\\&\quad-2K\exp\left(-\frac{s}{K}\frac{N\epsilon^2}{18(s/K)s+9\sqrt{2s}}\right)\\&\quad-3\left(\frac{24K}{\epsilon s}+1\right)^K\exp\left(-\frac{s}{K}\frac{N\epsilon^2}{360}\right).\end{aligned}$$

(2) For the Bernoulli(p)-Gaussian model, define

$$\nu_j(\mathbf{w})=\sqrt{\frac{2}{\pi}}p\left(\left|\mathbf{w}^T\mathbf{M}_0[,j]\right|-(1-p)\right).$$

Similar to (25), by triangle inequality,

$$\begin{aligned}&\sup_{\mathbf{w}\in\mathcal{F}_j(p)}|H_j(\mathbf{w})-\nu_j(\mathbf{w})|\\&\le\left|\left|\left|\bar{\mathbf{F}}[-j,j]-\sqrt{\frac{2}{\pi}}p\mathbf{M}_0[-j,j]\right|\right|\right|_p^*+|||\bar{\mathbf{G}}[-j,j]|||_p^*+\sup_{\mathbf{w}\in\mathcal{F}_j(p)}\left|\bar{\mathbf{t}}[j](\mathbf{w})-\sqrt{\frac{2}{\pi}}p(1-p)\right|.\end{aligned}$$

Then the analysis can be carried out in a similar manner using the parallel version of the concentration inequalities, i.e., Part 2 of Lemma 3—5. ■

A.3 Concentration Inequalities

We will make frequent use of the following version of Bernstein's inequality. The proof of the inequality can be found in, e.g., Chapter 14 of Bühlmann and van de Geer (2011).

Theorem 4 *(Bernstein's inequality) Let $Y_1,\ldots,Y_N$ be independent random variables that satisfy the moment condition*

$$\mathbb{E}Y_i^m\le\frac{1}{2}\times V\times m!\times B^{m-2},$$

for integers $m\ge2$. Then

$$\mathbb{P}\left\{\frac{1}{N}\left|\sum_{i=1}^N Y_i-\mathbb{E}Y_i\right|>\epsilon\right\}\le2\exp\left(-\frac{N\epsilon^2}{2V+2B\epsilon}\right).$$

Lemma 3 *(Uniform concentration of $\bar{\mathbf{F}}[-j,j]$) For $i \in [\![N]\!]$, let $\mathbf{F}_i \in \mathbb{R}^{K\times K}$ be defined as in (22) and $\bar{\mathbf{F}} = (1/N)\sum_{i=1}^N \mathbf{F}_i$.*

1. *Under the s-sparse Gaussian model with $s \in [\![K-1]\!]$,*

$$\mathbb{P}\left\{\left\|\bar{\mathbf{F}}[-j,j] - \sqrt{\frac{2}{\pi}}\frac{s}{K}\mathbf{M}_0[-j,j]\right\|_s^* > \frac{s}{K}\epsilon\right\} \le 2K\exp\left(-\frac{N\epsilon^2}{12K\|\mathbf{M}_0[-j,j]\|_\infty}\right),$$

for $0 < \epsilon \le 1$.

2. *Under the Bernoulli-Gaussian model with parameter $p \in (0,1)$,*

$$\mathbb{P}\left\{\left\|\bar{\mathbf{F}}[-j,j] - \sqrt{\frac{2}{\pi}}p\mathbf{M}_0[-j,j]\right\|_p^* > p\epsilon\right\} \le 2K\exp\left(-\frac{N\epsilon^2}{12(K+2p^{-1})\|\mathbf{M}_0[-j,j]\|_\infty}\right),$$

for $0 < \epsilon \le 1$.

In particular, if $\|\mathbf{M}_0[-j,j]\|_\infty = 0$, then the RHS bound is trivially zero.

Proof (1) First of all, we will prove the inequality for the s-sparse model. Notice that by Lemma 2, we have

$$\begin{aligned}\left\|\bar{\mathbf{F}}[-j,j] - \sqrt{\frac{2}{\pi}}\frac{s}{K}\mathbf{M}_0[-j,j]\right\|_s^* &\le \max_{|S|=s, j\notin S}\|\bar{\mathbf{F}}[S,j] - \sqrt{\frac{2}{\pi}}\frac{s}{K}\mathbf{M}_0[S,j]\|_2\\ &\le \sqrt{s}\max_{l\neq j}|\bar{\mathbf{F}}[l,j] - \sqrt{\frac{2}{\pi}}\frac{s}{K}\mathbf{M}_0[l,j]|.\end{aligned}$$

For convenience, define

$$\mathbf{v}_i[l] = |\mathbf{z}_i[l]|\sum_{k=1}^K\sum_{|S|=k, l\in S}\chi_i(S) - \sqrt{\frac{2}{\pi}}\frac{s}{K}.$$

for $i \in [\![N]\!]$ and $l \in [\![K]\!]$. Note that $\sum_{k=1}^K\sum_{l\in S,|S|=k}\chi_i(S) = 1$ with probability $\binom{K}{s}^{-1}\binom{K-1}{s-1} = \frac{s}{K}$. Thus

$$\mathbb{E}\left(\sum_{k=1}^K\sum_{|S|=k,l\in S}\chi_i(S)\right)^m = \frac{s}{K}.$$

For $m \ge 1$, by Jensen's inequality $|\frac{a+b}{2}|^m \le \frac{1}{2}(|a|^m + |b|^m)$ and $\mathbb{E}|Z|^m \ge (\mathbb{E}|Z|)^m = \left(\frac{2}{\pi}\right)^{\frac{m}{2}}$, where Z is a standard Gaussian variable. In addition, $\mathbb{E}|Z|^m \le (m-1)!! \le 2^{-\frac{m}{2}}m!$. Hence

$$\begin{aligned}\mathbb{E}|\mathbf{v}_i[l]|^m &\le 2^{m-1}\left(\mathbb{E}|\mathbf{z}_i[l]|^m + \left(\frac{2}{\pi}\right)^{\frac{m}{2}}\left(\frac{s}{K}\right)^m\right)\\ &\le 2\times\mathbb{E}|Z|^m\times 2^{m-1}\\ &\le 2\times\left(\frac{1}{2}\right)^{\frac{m}{2}}m!\times 2^{m-1}\\ &= \frac{1}{2}\times\frac{4s}{K}\times m!\times(\sqrt{2})^{m-2}.\end{aligned}$$

Thus by Bernstein's inequality, we have

$$\mathbb{P}\left\{\left|\frac{1}{N}\sum_{i=1}^N\mathbf{v}_i[l]\right| > \epsilon\right\} \le 2\exp\left(-\frac{N\epsilon^2}{2(4\frac{s}{K}+\sqrt{2}\epsilon)}\right).$$

Therefore,

$$\begin{aligned}\mathbb{P}\left\{\left|\mathbf{M}_0[j,l]\frac{1}{N}\sum_{i=1}^N\mathbf{v}_i[l]\right| > \frac{s}{K}\epsilon\right\} &\le 2\exp\left(-\frac{s}{K}\frac{N\epsilon^2}{2(4\mathbf{M}_0[j,l]^2 + \sqrt{2}\,|\mathbf{M}_0[j,l]|\,\epsilon)}\right)\\ &\le 2\exp\left(-\frac{s}{K}\frac{N\epsilon^2}{2\,|\mathbf{M}_0[j,l]|\,(4+\sqrt{2}\epsilon)}\right)\\ &\le 2\exp\left(-\frac{s}{K}\frac{N\epsilon^2}{12\,|\mathbf{M}_0[j,l]|}\right).\end{aligned}$$

for $\epsilon \le 1$. Notice that if $\mathbf{M}_0[j,l] = 0$ the LHS probability is trivially zero. Using a union bound, we have

$$\begin{aligned}\mathbb{P}\left\{\|\bar{\mathbf{F}}[-j,j] - \sqrt{\frac{2}{\pi}}\frac{s}{K}\mathbf{M}_0[-j,j]\|_\infty > \frac{s}{K}\epsilon\right\} &= \mathbb{P}\left\{\max_{l\neq j}|\mathbf{M}_0[j,l]\frac{1}{N}\sum_{i=1}^N\mathbf{v}_i[l]| > \epsilon\right\}\\ &\le 2K\exp\left(-\frac{s}{K}\frac{N\epsilon^2}{12\|\mathbf{M}_0[-j,j]\|_\infty}\right).\end{aligned}$$

Therefore

$$\begin{aligned}\mathbb{P}\left\{\left\|\bar{\mathbf{F}}[-j,j] - \sqrt{\frac{2}{\pi}}\frac{s}{K}\mathbf{M}_0[-j,j]\right\|_s^* > \frac{s}{K}\epsilon\right\} &\le \mathbb{P}\left\{\sqrt{s}\|\bar{\mathbf{F}}[,j] - \sqrt{\frac{2}{\pi}}\frac{s}{K}\mathbf{M}_0[,j]\|_\infty > \frac{s}{K}\epsilon\right\}\\ &\le 2K\exp\left(-\frac{N\epsilon^2}{12K\|\mathbf{M}_0[-j,j]\|_\infty}\right).\end{aligned}$$

(2) Now let us consider the Bernoulli-Gaussian model. Notice that by Lemma 8, for $\frac{s-1}{K-1} \ge p$, we have

$$\begin{aligned}\left\|\bar{\mathbf{F}}[-j,j] - \sqrt{\frac{2}{\pi}}p\mathbf{M}_0[-j,j]\right\|_p^* &\le \left\|\bar{\mathbf{F}}[-j,j] - \sqrt{\frac{2}{\pi}}p\mathbf{M}_0[-j,j]\right\|_s^*\\ &\le \sqrt{s}\left\|\bar{\mathbf{F}}[-j,j] - \sqrt{\frac{2}{\pi}}p\mathbf{M}_0[-j,j]\right\|_\infty.\end{aligned}$$

Now let $s = \lceil pK - p + 1\rceil \le pK + 2$. For $i \in [\![N]\!]$ and $l \in [\![K]\!]$, define

$$\mathbf{u}_i[l] = |\mathbf{z}_i[l]|\sum_{k=1}^K\sum_{|S|=s,l\in S}\chi_i(S) - \sqrt{\frac{2}{\pi}}p.$$

Note that the event $\left\{\sum_{k=1}^{K}\sum_{|S|=k,l\in S}\chi_i(S)=1\right\}$ is the same as the event that $\{\boldsymbol{\alpha}_i[l]=1\}$, which, happens with probability p. Thus

$$\mathbb{E}\left(\sum_{k=1}^{K}\sum_{|S|=k,l\in S}\chi_i(S)\right)^m = p.$$

Similar to the case of s-sparse model,

$$\mathbb{E}\,|\mathbf{u}_i[l]|^m \le \frac{1}{2}\times 4p\times m!\times\left(\sqrt{2}\right)^{m-2}.$$

By Bernstein's inequality, we have

$$\mathbb{P}\left\{\left|\frac{1}{N}\sum_{i=1}^{N}\mathbf{u}_i[l]\right|>\epsilon\right\}\le 2\exp\left(-\frac{N\epsilon^2}{2(4p+\sqrt{2}\epsilon)}\right).$$

Therefore

$$\begin{aligned}\mathbb{P}\left\{\left\|\left\|\bar{\mathbf{F}}[-j,j]-\sqrt{\frac{2}{\pi}}p\mathbf{M}_0[-j,j]\right\|\right\|_p^*>p\epsilon\right\}&\le\mathbb{P}\left\{\sqrt{s}\|\bar{\mathbf{F}}[,j]-\sqrt{\frac{2}{\pi}}p\mathbf{M}_0[,j]\|_\infty>p\epsilon\right\}\\&\le 2K\exp\left(-\frac{p}{s}\frac{N\epsilon^2}{2\|\mathbf{M}_0[-j,j]\|_\infty(4+\sqrt{2}\epsilon)}\right)\\&\le 2K\exp\left(-\frac{N\epsilon^2}{12(K+2p^{-1})\|\mathbf{M}_0[-j,j]\|_\infty}\right),\end{aligned}$$

for $\epsilon\le 1$. ■

Lemma 4 *(Uniform concentration of $\bar{\mathbf{G}}[-j,j]$) For $i\in[\![N]\!]$, let $\mathbf{G}_i\in\mathbb{R}^{K\times K}$ be defined as in (23) and $\bar{\mathbf{G}}=(1/N)\sum_{i=1}^{N}\mathbf{G}_i$.*

1. *Under the s-sparse Gaussian model with $s\in[\![K-1]\!]$,*

$$\mathbb{P}\left\{\left|\left|\left|\bar{\mathbf{G}}[-j,j]\right|\right|\right|_s^*>\frac{s}{K}\epsilon\right\}\le 2K\exp\left(-\frac{s}{K}\frac{N\epsilon^2}{2(s/K)s+\sqrt{2s}}\right),$$

   for $0<\epsilon\le 1$.

2. *Under the Bernoulli-Gaussian model with parameter $p\in(0,1)$,*

$$\mathbb{P}\left\{\left|\left|\left|\bar{\mathbf{G}}[-j,j]\right|\right|\right|_p^*>p\epsilon\right\}\le 2K\exp\left(-p\frac{N\epsilon^2}{p(pK+2)+\sqrt{2(pK+2)}}\right),$$

   for $0<\epsilon\le 1$.

Proof The proof is highly similar to that of Lemma 3 and so we will omit some common steps.
(1) We first prove the concentration inequality for the s-sparse model. Notice that

$$\left|\left|\left|\bar{\mathbf{G}}[-j,j]\right|\right|\right|_s^*\le\sqrt{s}\max_{l\neq j}|\bar{\mathbf{G}}[l,j]|.$$

In addition,

$$\begin{aligned}\mathbb{E}\left(\sum_{k=2}^{K}\sum_{\{j,l\}\in S,|S|=k}\chi_i(S)\right)^m&=\mathbb{E}\left(\sum_{k=2}^{K}\sum_{|S|=k,\{j,l\}\in S}\chi_i(S)\right)\\&=\binom{K}{s}^{-1}\binom{K-2}{s-2}=\frac{s(s-1)}{K(K-1)}\le(\frac{s}{K})^2.\end{aligned}$$

Thus

$$\mathbb{E}|\mathbf{G}_i[l,j]|^m\le 2^{-m/2}m!\times(\frac{s}{K})^2=\frac{1}{2}\times(\frac{s}{K})^2\times m!\times(\frac{1}{\sqrt{2}})^{m-2}.$$

By Bernstein inequality:

$$\mathbb{P}\left\{\left|\frac{1}{N}\sum_{i=1}^{N}\mathbf{G}_i[l,j]\right|>\epsilon\right\}\le 2\exp\left(-\frac{N\epsilon^2}{2(s/K)^2+\sqrt{2}\epsilon)}\right).$$

Thus we have

$$\begin{aligned}\mathbb{P}\left\{\left|\left|\left|\bar{\mathbf{G}}[-j,j]\right|\right|\right|_s^*>\frac{s}{K}\epsilon\right\}&\le\mathbb{P}\left\{\sqrt{s}\max_{l\neq j}|\bar{\mathbf{G}}[l,j]|>\frac{s}{K}\epsilon\right\}\\&\le 2K\exp\left(-\frac{(s/K)^2N(\epsilon^2/s)}{2(s/K)^2+\sqrt{2}(s/K)(\epsilon/\sqrt{s}))}\right)\\&\le 2K\exp\left(-\frac{s}{K}\frac{N\epsilon^2}{2(s/K)s+\sqrt{2s}\epsilon}\right)\\&\le 2K\exp\left(-\frac{s}{K}\frac{N\epsilon^2}{2(s/K)s+\sqrt{2s}}\right),\end{aligned}$$

for $\epsilon\le 1$.
(2) For Bernoulli-Gaussian model, notice that

$$\left|\left|\left|\bar{\mathbf{G}}[-j,j]\right|\right|\right|_p^*\le\left|\left|\left|\bar{\mathbf{G}}[-j,j]\right|\right|\right|_s^*\le\sqrt{s}\max_{l\neq j}|\bar{\mathbf{G}}[l,j]|,$$

for $s=\lceil pK-p+1\rceil\le pK+2$. Also,

$$\mathbb{E}|\mathbf{G}_i[l,j]|^m\le 2^{-m/2}m!\times p^2=\frac{1}{2}\times p^2\times m!\times(\frac{1}{\sqrt{2}})^{m-2}.$$

Thus

$$\begin{aligned}\mathbb{P}\left\{\left|\left|\left|\bar{\mathbf{G}}[-j,j]\right|\right|\right|_s^*>p\epsilon\right\}&\le\mathbb{P}\left\{\sqrt{s}\max_{l\neq j}|\bar{\mathbf{G}}[l,j]|>p\epsilon\right\}\\&\le 2K\exp(-p\frac{N(\epsilon^2)}{2ps+\sqrt{2s}})\\&\le 2K\exp\left(-p\frac{N\epsilon^2}{p(pK+2)+\sqrt{2(pK+2)}}\right),\end{aligned}$$

for $\epsilon \le 1$. ■

Lemma 5 *(Uniform concentration of $\bar{\mathbf{t}}[j](\mathbf{w})$) For $i \in [\![N]\!]$, let $\mathbf{t}_i$ be the function from $\mathbb{R}^K$ to $\mathbb{R}^K$ defined as in (24) and $\bar{\mathbf{t}} = (1/N)\sum_{i=1}^N \mathbf{t}_i$. Recall that for $j \in [\![K]\!]$, $s \in [\![K-1]\!]$ and $p \in (0,1)$,*

$$\mathcal{E}_j(s) = \{\mathbf{w} \in \mathbb{R}^K, |||\mathbf{w}[-j]|||_s = 1, \mathbf{w}[j] = 0\},$$

$$\mathcal{F}_j(p) = \{\mathbf{w} \in \mathbb{R}^K, |||\mathbf{w}[-j]|||_p = 1, \mathbf{w}[j] = 0\}.$$

1. *Under the s-sparse Gaussian model with $s \in [\![K-1]\!]$,*

$$\mathbb{P}\left\{\sup_{w\in\mathcal{E}_j(s)}\left|\bar{\mathbf{t}}[j](\mathbf{w}) - \sqrt{\frac{2}{\pi}}\frac{s}{K}\frac{K-s}{K-1}\right| > \frac{s}{K}\epsilon\right\} \le 3\left(\frac{8K}{\epsilon s}+1\right)^K \exp\left(-\frac{s}{K}\frac{N\epsilon^2}{40}\right),$$

   for $0 < \epsilon \le \frac{1}{2}$.

2. *Under the Bernoulli-Gaussian model with parameter $p \in (0,1)$,*

$$\mathbb{P}\left\{\sup_{w\in\mathcal{F}_j(p)}\left|\bar{\mathbf{t}}[j](\mathbf{w}) - \sqrt{\frac{2}{\pi}}p(1-p)\right| > p\epsilon\right\} \le 3\left(\frac{8}{\epsilon p}+1\right)^K \exp\left(-p\frac{N\epsilon^2}{40}\right),$$

   for $0 < \epsilon \le \frac{1}{2}$.

Proof (1) Under the s-sparse model, we have

$$\begin{aligned}\mathbb{E}|\mathbf{t}_i[j](\mathbf{w})|^m &= \mathbb{E}\left(\sum_{|S|=s, j\notin S}|\mathbf{w}[S]^T\mathbf{z}_i[S]|\chi_i(S)\right)^m \\ &= \sum_{|S|=s,j\notin S}\mathbb{E}|\mathbf{w}[S]^T\mathbf{z}_i[S]|^m\mathbb{E}\chi_i(S) \\ &= \binom{K}{s}^{-1}\sum_{|S|=s,j\notin S}\mathbb{E}|\mathbf{w}[S]^T\mathbf{z}_i[S]|^m.\end{aligned}$$

Notice that we have used the facts that the events $\chi_i(S)$'s are mutually exclusive and that $\mathbf{z}_i[S]$ and $\chi_i(S)$ are independent. Since the random variable $\mathbf{w}[S]^T\mathbf{z}_i[S]$ has distribution $N(0, \|\mathbf{w}[S]\|_2)$, $\mathbb{E}|\mathbf{w}[S]^T\mathbf{z}_i[S]|^m = \|\mathbf{w}[S]\|_2^m\mathbb{E}|Z|^m \le 2^{-\frac{m}{2}}m!$. Therefore

$$\mathbb{E}|\mathbf{t}_i[j](\mathbf{w})|^m \le 2^{-\frac{m}{2}}m!\binom{K}{s}^{-1}\sum_{j\notin S,|S|=s}\|\mathbf{w}[S]\|_2^m.$$

Note that by Lemma 7, $|||\mathbf{w}[-j]|||_s \ge \|\mathbf{w}[-j]\|_2 \ge \|\mathbf{w}[S]\|_2$ for all S such that $j \notin S$. For $\mathbf{w} \in \mathcal{E}_j(s)$, $|||\mathbf{w}|||_s = 1$ and so $\|\mathbf{w}_S\|_2 \le 1$, which, further implies that $\|\mathbf{w}[S]\|_2^m \le \|\mathbf{w}[S]\|_2$.

Thus we have

$$\begin{aligned}\mathbb{E}|\mathbf{t}_i[j](\mathbf{w})|^m &\le 2^{-\frac{m}{2}}m!\binom{K}{s}^{-1}\sum_{j\notin S,|S|=s}\|\mathbf{w}[S]\|_2 \\ &\le 2^{-\frac{m}{2}}m!\frac{s(K-s)}{K(K-1)}|||\mathbf{w}[-j]|||_s \\ &= 2^{-\frac{m}{2}}m!\frac{s(K-s)}{K(K-1)}\end{aligned}$$

For a fixed j, define

$$U_i(\mathbf{w}) = \mathbf{t}_i[j](\mathbf{w}) - \sqrt{\frac{2}{\pi}}\frac{s}{K}\frac{K-s}{K-1}.$$

Notice that $\mathbb{E}U_i(\mathbf{w}) = 0$. In addition,

$$\mathbb{E}|U_i(\mathbf{w})|^m \le 2^m\mathbb{E}|\mathbf{t}_i[j](\mathbf{w})|^m \le \frac{1}{2}\times 4\frac{s}{K}\frac{K-s}{K-1}\times m!\times(\sqrt{2})^{m-2}.$$

By Bernstein's inequality

$$\mathbb{P}\left\{\frac{1}{N}\left|\sum_{i=1}^N U_i(\mathbf{w})\right| > \frac{s}{K}\epsilon\right\} \le 2\exp\left(-\frac{s}{K}\frac{N\epsilon^2}{2(4\frac{K-s}{K-1}+\sqrt{2}\epsilon)}\right) \le 2\exp\left(-\frac{s}{K}\frac{N\epsilon^2}{10}\right),$$

for $0 < \epsilon \le 1/2$. Now let $\{\mathbf{w}_i\}$ be an δ-cover of $\mathcal{E}_j(s)$. Since $\mathcal{E}_j(s)$ is contained in the unit ball $\{\mathbf{w} \in \mathbb{R}^{K-1} : \|\mathbf{w}\|_2 \le 1\}$, there exists a cover such that $|\{\mathbf{w}_l\}| \le \left(\frac{2}{\delta}+1\right)^{K-1}$. For any $\mathbf{w}, \mathbf{w}' \in \mathcal{E}_j(s)$, we have

$$\left|U_i(\mathbf{w}) - U_i(\mathbf{w}')\right| \le \sum_{j\notin S,|S|=s}\left|(\mathbf{w}[S]-\mathbf{w}'[S])^T\mathbf{z}_i[S]\right|\chi_i(S).$$

Let Z be a standard Gaussian variable. We have

$$\begin{aligned}\mathbb{P}\left\{\sum_{|S|=s,j\notin S}\left|\mathbf{w}[S]^T\mathbf{z}_i[S]\right|\chi_i(S) > \epsilon\right\} &= \binom{K-1}{s}^{-1}\sum_{|S|=s,j\notin S}\mathbb{P}\left\{\left|\mathbf{w}[S]^T\mathbf{z}_i[S]\right| > \epsilon\right\} \\ &= \binom{K-1}{s}^{-1}\sum_{|S|=s,j\notin S}\mathbb{P}\left\{\|\mathbf{w}[S]\|_2|Z| > \epsilon\right\} \\ &\le \mathbb{P}\left\{\|\mathbf{w}\|_2|Z| > \epsilon\right\}.\end{aligned}$$

Let Z_i, $i = 1, \ldots, N$, be *i.i.d.* standard Gaussian variables. By the one-sided Bernstein's inequality,

$$\mathbb{P}\left\{\frac{1}{N}\sum_{i=1}^N|Z_i| \ge 2\right\} \le \exp\left(-\frac{N(2-\sqrt{2/\pi})^2}{2(4+\sqrt{2}(2-\sqrt{2/\pi}))}\right) \le \exp\left(-\frac{N}{8}\right).$$

Now let $\delta = \frac{s}{K}\frac{\epsilon}{4}$. Thus

$$\begin{aligned}\mathbb{P}\left\{\sup_{\|\mathbf{w}-\mathbf{w}'\|_2\le\delta}\left|\frac{1}{N}\sum_{i=1}^{N}(U_i(\mathbf{w})-U_i(\mathbf{w}'))\right|>\frac{s}{K}\frac{\epsilon}{2}\right\} &\le \mathbb{P}\left\{\sup_{\|\mathbf{w}'-\mathbf{w}\|_2\le\delta}\frac{1}{N}\sum_{i=1}^{N}\left|U_i(\mathbf{w})-U_i(\mathbf{w}')\right|>\frac{s}{K}\frac{\epsilon}{2}\right\}\\ &\le \mathbb{P}\left\{\sup_{\|\mathbf{w}'-\mathbf{w}\|_2\le\delta}\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{w}-\mathbf{w}'\|_2|Z_i|>\frac{s}{K}\frac{\epsilon}{2}\right\}\\ &\le \mathbb{P}\left\{\delta\frac{1}{N}\sum_{i=1}^{N}|Z_i|>\frac{s}{K}\frac{\epsilon}{2}\right\}\le\mathbb{P}\left\{\frac{1}{N}\sum_{i=1}^{N}|Z_i|>2\right\}\\ &\le \exp\left(-\frac{N}{8}\right).\end{aligned}$$

By triangle inequality

$$\sup_{\|\mathbf{w}'-\mathbf{w}\|_2\le\delta}\left|\frac{1}{N}\sum_{i=1}^{N}U_i(\mathbf{w}')\right|\le\sup_{\|\mathbf{w}'-\mathbf{w}\|_2\le\delta}\left|\frac{1}{N}\sum_{i=1}^{N}(U_i(\mathbf{w})-U_i(\mathbf{w}'))\right|+\left|\frac{1}{N}\sum_{i=1}^{N}U_i(\mathbf{w})\right|.$$

Using a union bound, we have

$$\begin{aligned}\mathbb{P}\{\sup_{\|\mathbf{w}'-\mathbf{w}\|_2\le\delta}\left|\frac{1}{N}\sum_{i=1}^{N}U_i(\mathbf{w}')\right|>\frac{s}{K}\epsilon\} &\le \mathbb{P}\left\{\sup_{\|\mathbf{w}-\mathbf{w}'\|_2\le\delta}\left|\frac{1}{N}\sum_{i=1}^{N}(U_i(\mathbf{w})-U_i(\mathbf{w}'))\right|>\frac{s}{K}\frac{\epsilon}{2}\right\}\\ &\quad+\mathbb{P}\left\{\left|\frac{1}{N}\sum_{i=1}^{N}U_i(\mathbf{w})\right|>\frac{s}{K}\frac{\epsilon}{2}\right\}\\ &\le \exp\left(-\frac{N}{8}\right)+2\exp\left(-\frac{s}{K}\frac{N\epsilon^2}{40}\right)\\ &\le 3\exp\left(-\frac{s}{K}\frac{N\epsilon^2}{40}\right),\end{aligned}$$

for $0<\epsilon\le 1$. Now apply union bound again,

$$\begin{aligned}\mathbb{P}\left\{\sup_{\mathbf{w}\in\mathcal{E}_j(s)}\frac{1}{N}\left|\sum_{i=1}^{N}U_i(\mathbf{w})\right|>\frac{s}{K}\epsilon\right\} &\le \mathbb{P}\left\{\max_{l}\sup_{||\mathbf{w}-\mathbf{w}_l||_2\le\delta}\frac{1}{N}\left|\sum_{i=1}^{N}U_i(\mathbf{w})\right|>\frac{s}{K}\epsilon\right\}\\ &\le 3\left(\frac{8K}{\epsilon s}+1\right)^{K}\exp\left(-\frac{s}{K}\frac{N\epsilon^2}{40}\right).\end{aligned}$$

(2) For $\mathbf{w}\in\mathcal{F}_j(p)$, under the Bernoulli-Gaussian model,

$$\begin{aligned}\mathbb{E}\left|\mathbf{t}_i[j](\mathbf{w})\right|^m &= \mathbb{E}|Z|^m\sum_{k=1}^{K-1}\sum_{|S|=k,j\notin S}\|\mathbf{w}[S]\|_2^m\times p^k(1-p)^{K-k}\\ &\le \mathbb{E}|Z|^m p\sum_{k=1}^{K-1}\sum_{|S|=k,j\notin S}\|\mathbf{w}[S]\|_2\times p^{k-1}(1-p)^{K-k}\\ &= \mathbb{E}|Z|^m p(1-p)\sum_{k=0}^{K-2}\sum_{|S|=k+1,j\notin S}\|\mathbf{w}[S]\|_2\times p^k(1-p)^{K-2-k}\\ &= \mathbb{E}|Z|^m p(1-p)|||\mathbf{w}[-j]|||_p=\mathbb{E}|Z|^m p(1-p)\\ &\le 2^{-m/2}m!p(1-p).\end{aligned}$$

Notice that we have used the fact that $\|\mathbf{w}[S]\|_2\le\|\mathbf{w}[-j]\|_2\le|||\mathbf{w}[-j]|||_p=1$ for all S such that $j\notin S$. For each fixed $\mathbf{w}$, define

$$V_i(\mathbf{w})=\mathbf{t}_i[j](\mathbf{w})-\sqrt{\frac{2}{\pi}}(1-p)p.$$

Now we have

$$\mathbb{E}|V_i(\mathbf{w})|^m\le 2^m\mathbb{E}|\mathbf{t}_i[j](\mathbf{w})|^m\le\frac{1}{2}\times 4p(1-p)\times m!\times(\sqrt{2})^{m-2}.$$

The remaining parts of the proof can be proceeded exactly as in the case of the s-sparse model, noticing that we only need to replace $\frac{s}{K}$ by p, and $\frac{K-s}{K-1}$ by $1-p$.

■

A.4 Dual Analysis of $|||.|||_s$ and $|||.|||_p$

In this section, we will characterize the dual norms $|||.|||_s^*$ and $|||.|||_p^*$ by second order cone programs (SOCP). The characterization is helpful for deriving bounds for these special norms in the next section.

Lemma 6 *For $i\in[\![M]\!]$, let $\mathbf{A}_i$ be an $k_i\times K$ with rank k_i. For $\mathbf{z}\in\mathbb{R}^K$, define*

$$\|\mathbf{z}\|_{\mathbf{A}}=\sum_{i=1}^{M}\|\mathbf{A}_i\mathbf{z}\|_2.$$

Then the dual norm of $\|.\|_{\mathbf{A}}$ is

$$\|\mathbf{v}\|_{\mathbf{A}}^*=\inf\left\{\max_i\|\mathbf{y}_i\|_2,\mathbf{y}_i\in\mathbb{R}^{k_i},\sum_{i=1}^{M}\mathbf{A}_i^T\mathbf{y}_i=\mathbf{v}\right\}.$$

Proof

$$\|\mathbf{v}\|_{\mathbf{A}}^* = \sup_{\mathbf{z}\neq 0} \frac{\mathbf{v}^T\mathbf{z}}{\|\mathbf{z}\|_{\mathbf{A}}} = \sup\left\{\mathbf{v}^T\mathbf{z} : \|\mathbf{z}\|_{\mathbf{A}} \leq 1\right\}.$$

Introducing Lagrange multiplier $\lambda \geq 0$ for the inequality constraint, the above problem is equivalent to the following

$$\begin{aligned}\|\mathbf{v}\|_{\mathbf{A}}^* &= \sup_{\mathbf{z}}\left\{\inf_{\lambda\geq 0}\left\{\mathbf{v}^T\mathbf{z} + \lambda(1 - \|\mathbf{z}\|_{\mathbf{A}})\right\}\right\} \\ &= \sup_{\mathbf{z}}\left\{\inf_{\lambda\geq 0}\left\{\mathbf{v}^T\mathbf{z} + \lambda(1 - \sum_{i=1}^{M}\|\mathbf{A}_i\mathbf{z}\|_2)\right\}\right\}.\end{aligned}$$

The dual problem is

$$d = \inf_{\lambda\geq 0}\left\{\sup_{\mathbf{z}}\left\{\mathbf{v}^T\mathbf{z} + \lambda(1 - \sum_{i=1}^{M}\|\mathbf{A}_i\mathbf{z}\|_2)\right\}\right\}.$$

Notice that $\|\mathbf{A}_i\mathbf{z}\|_2 = \sup\{\mathbf{z}^T\mathbf{A}_i^T\mathbf{u}_i : \|\mathbf{u}_i\|_2 \leq 1\}$. Hence

$$d = \inf_{\lambda\geq 0}\left\{\lambda + \sup_{\mathbf{z},\mathbf{u}}\left\{\mathbf{z}^T(\mathbf{v} - \lambda\sum_{i=1}^{M}\mathbf{A}_i^T\mathbf{u}_i) : \|\mathbf{u}_i\|_2 \leq 1\right\}\right\}.$$

Since the vector $\mathbf{z}$ can be arbitrary, in order to have a finite value, we must have $\lambda\sum_{i=1}^{M}\mathbf{A}_i^T\mathbf{u}_i = \mathbf{v}$. Now let $\mathbf{y}_i = \lambda\mathbf{u}_i$, the problem becomes

$$d = \inf_{\lambda\geq 0}\left\{\lambda : \sum_{i=1}^{M}\mathbf{A}_i^T\mathbf{y}_i = \mathbf{v}, \|\mathbf{y}_i\|_2 \leq \lambda\right\}.$$

The above problem is exactly equivalent to

$$\inf\left\{\max_i \|\mathbf{y}_i\|_2, \mathbf{y}_i \in \mathbb{R}^{k_i}, \sum_{i=1}^{M}\mathbf{A}_i^T\mathbf{y}_i = \mathbf{v}\right\}.$$

Finally, notice that the original problem is convex and strictly feasible. Thus Slater's condition holds and the duality gap is zero. Hence

$$||\mathbf{v}||_{\mathbf{A}}^* = \inf\left\{\max_i \|\mathbf{y}_i\|_2, \mathbf{y}_i \in \mathbb{R}^{k_i}, \sum_{i=1}^{M}\mathbf{A}_i^T\mathbf{y}_i = \mathbf{v}\right\}.$$

■

The following corollary gives an alternative characterization of $|||.|||_s$ and $|||.|||_p$:

Corollary 5 *Denote by $\mathbf{y}_S \in \mathbb{R}^{|S|}$ a variable vector indexed by the set S (as opposed to being a sub-vector of $\mathbf{y}$). For $\mathbf{z} \in \mathbb{R}^m$, we have*

$$|||\mathbf{z}|||_s^* = \inf\left\{\max_{|S|=s}\|\mathbf{y}_S\|_2 : \mathbf{y}_S \in \mathbb{R}^s, \sum_{|S|=s}\mathbf{E}_S^T\mathbf{y}_S = \mathbf{z}\right\},$$

and

$$|||\mathbf{z}|||_p^* = \inf\left\{\max_S \|\mathbf{y}_S\|_2 : \mathbf{y}_S \in \mathbb{R}^{|S|}, \sum_{k=0}^{m-1}\texttt{pbinom}(k; m-1, p)\sum_{|S|=k+1}\mathbf{E}_S^T\mathbf{y}_S = \mathbf{z}\right\},$$

where $\mathbf{E}_S = \mathbf{I}[S,]/\binom{m-1}{|S|-1}$ and $\mathbf{I} \in \mathbb{R}^{m\times m}$ is the identity matrix.

Proof This is simply a direct application of Lemma 6. ■

Corollary 6 *The dual norms $|||.|||_s^*$ and $|||.|||_p^*$ can be computed via a Second Order Cone Program (SOCP).*

Proof Introducing additional variable $t \geq 0$, the problem of computing $|||\mathbf{z}|||_s^*$ is equivalent to the following formulation

$$\begin{aligned}\inf_{t,\mathbf{y}_S} \quad & t \;\; s.t. \;\; \|\mathbf{y}_S\|_2 \leq t \text{ for all } S \text{ such that } |S| = s \\ & \text{and } \sum_{|S|=s}\mathbf{E}_S^T\mathbf{y}_S = \mathbf{z}.\end{aligned}$$

Notice that the above program is already in the standard form of SOCP. The case of $|||.|||_p^*$ can be handled in a similar manner. ■

A.5 Inequalities of $|||.|||_s$ and $|||.|||_p$ and Their Duals

As demonstrated in the last section, it is in general expensive to compute $|||.|||_s^*$ and $|||.|||_p^*$. In this section, we will derive sharp and easy-to-compute lower and upper bounds to approximate these quantities.

Lemma 7 *(Monotonicity of $|||\mathbf{z}|||_s$ and $|||\mathbf{z}|||_p$) Let $\mathbf{z} \in \mathbb{R}^m$. $|||\mathbf{z}|||_1 = \|\mathbf{z}\|_1$ and $|||\mathbf{z}|||_m = \|\mathbf{z}\|_2$. For $1 \leq l < k \leq m$, we have $|||\mathbf{z}|||_l \geq |||\mathbf{z}|||_k$; similarly for $0 < p < q < 1$, $|||\mathbf{z}|||_p \geq |||\mathbf{z}|||_q$. Furthermore, the equalities hold iff the vector $\mathbf{z}$ contains at most one non-zero entry.*

Proof By definition, we have

$$|||\mathbf{w}|||_1 = \frac{\sum_{|S|=1}\|\mathbf{w}[S]\|_2}{\binom{m-1}{1-1}} = \|\mathbf{w}\|_1.$$

Similarly,

$$\|\mathbf{w}\|_m = \frac{\sum_{|S|=m} \|\mathbf{w}[S]\|_2}{\binom{m-1}{m-1}} = \|\mathbf{w}\|_2.$$

For $1 \le k \le m-1$, let S' be a subset of $[\![m]\!]$ such that $|S'| = k+1$. By triangle inequality

$$\sum_{|S|=k, S\subset S'} \|\mathbf{z}[S]\|_2 \ge k\|\mathbf{z}[S']\|_2,$$

where the equality holds iff $\|\mathbf{z}[S']\|_0 \le 1$. Thus

$$\sum_{|S'|=k+1} \sum_{|S|=k, S\subset S'} \|\mathbf{z}[S]\|_2 \ge k \sum_{|S'|=k+1} \|\mathbf{z}[S']\|_2,$$

and the equality holds iff $\|\mathbf{z}\|_0 \le 1$. Notice that the LHS of the above inequality is simply $(m-k)\sum_{|S|=k} \|\mathbf{z}[S]\|_2$. Therefore

$$\|\mathbf{z}\|_k = \binom{m-1}{k-1}^{-1} \sum_{|S|=k} \|\mathbf{z}[S]\|_2 \ge \binom{m-1}{k}^{-1} \sum_{|S|=k+1} \|\mathbf{z}[S]\|_2 = \|\mathbf{z}\|_{k+1},$$

and so the inequality holds.

For $\|.\|_p$, let Y be a random variable that follows the binomial distribution with parameters $m-1$ and p. Observe that $\|\mathbf{z}\|_p = \mathbb{E}\|\mathbf{z}\|_{Y+1}$, where the expectation is taken with respect to Y. If $\|\mathbf{z}\|_0 > 1$, $\|\mathbf{z}\|_k$ is strictly decreasing in k by the first part. Hence, $\|\mathbf{z}\|_p$ as a function of p is also strictly decreasing on $(0,1)$. Indeed, it can be shown that

$$\frac{d}{dp}\|\mathbf{z}\|_p = \sum_{k=0}^{m-1} \texttt{pbinom}(k; K-1, p)\left(\|\mathbf{z}\|_{k+1} - \|\mathbf{z}\|_k\right) < 0.$$

If $\|\mathbf{z}\|_0 \le 1$, then $\|\mathbf{z}\|_1 = \|\mathbf{z}\|_m$ and so $\frac{d}{dp}\|\mathbf{z}\|_p = 0$. Therefore $\|\mathbf{z}\|_p = \|\mathbf{z}\|_1$ is a constant in p. On the other hand, if $\|\mathbf{z}\|_p = \|\mathbf{z}\|_q$ for $0 < p < q < 1$, by the fact that $\frac{d}{dp}\|\mathbf{z}\|_p \le 0$, we must have $\frac{d}{dp}\|\mathbf{z}\|_p = 0$ and so $\|\mathbf{z}\|_{k-1} = \|\mathbf{z}\|_k$ for all $k \in [\![m]\!]$. Thus $\|\mathbf{z}\|_0 \le 1$. ■

Corollary 7 *(Monotonicity of $\|\mathbf{z}\|_s^*$ and $\|\mathbf{z}\|_p^*$) Let $\mathbf{z} \in \mathbb{R}^m$. $\|\mathbf{z}\|_1^* = \|\mathbf{z}\|_\infty$ and $\|\mathbf{z}\|_m^* = \|\mathbf{z}\|_2$. For $1 \le i < j \le m$, we have $\|\mathbf{z}\|_i^* \le \|\mathbf{z}\|_j^*$; similarly for $0 < p < q < 1$, $\|\mathbf{z}\|_p^* \le \|\mathbf{z}\|_q^*$. Furthermore, the equalities hold iff the vector $\mathbf{z}$ contains at most one non-zero entry.*

Proof This is a direct consequence of Lemma 7 and the dual norm definition $\|\mathbf{z}\|^* = \sup_{\mathbf{y}\neq 0} \frac{\mathbf{z}^T\mathbf{y}}{\|\mathbf{y}\|}$. ■

Lemma 8 *Let $p \in (0,1)$ and $k = \lceil (m-1)p + 1 \rceil$. For any $\mathbf{z} \in \mathbb{R}^m$, we have*

1. $\|\mathbf{z}\|_p \ge \|\mathbf{z}\|_k$.

2. $\|\mathbf{z}\|_p^* \le \|\mathbf{z}\|_k^*$.

Proof Define the function f with domain on $[1, m]$ as follows: let $f(1) = \|\mathbf{z}\|_1 = \|\mathbf{z}\|_1$; for $i \in [\![m-1]\!]$ and $a \in (i, i+1]$, define

$$f(a) = \|\mathbf{z}\|_i + (\|\mathbf{z}\|_{i+1} - \|\mathbf{z}\|_i)(a - i).$$

It is clear that f is piecewise linear by construction. In addition, by Lemma 12, f is also convex. Notice that $\|\mathbf{z}\|_p = \mathbb{E}\|\mathbf{z}\|_{Y+1} = \mathbb{E}f(Y+1)$, where Y is a random variable from the binomial distribution with parameters $m-1$ and p. By Jensen's inequality,

$$\mathbb{E}f(Y+1) \ge f(\mathbb{E}Y + 1) = f((m-1)p + 1).$$

Thus by Lemma 7, $\|\mathbf{z}\|_p \ge \|\mathbf{z}\|_k$ for all $k \ge (m-1)p + 1$. So the first part follows.

To upperbound $\|\mathbf{z}\|_p^*$, notice that if $k \ge (m-1)p + 1$,

$$\|\mathbf{z}\|_p^* = \sup_{\mathbf{w}\neq 0} \frac{\mathbf{w}^T\mathbf{z}}{\|\mathbf{w}\|_p} \le \sup_{\mathbf{w}\neq 0} \frac{\mathbf{w}^T\mathbf{z}}{\|\mathbf{w}\|_k} = \|\mathbf{z}\|_k^*.$$

■

For the following lemmas, the quantities $\tau_m(d,a)$ and $L_m(d,k)$ are defined as in Definition 3.

Lemma 9 *(Approximating $\tau_m(d,a)$) For $d \in [\![m]\!]$ and $a \in (0, m]$,*

$$\tau_m(d,a) \le \sqrt{\frac{da}{m}}.$$

Proof For $k \in [\![m]\!]$, by Jensen's inequality,

$$\mathbb{E}\sqrt{L_m(d,k)} \le \sqrt{\mathbb{E}L_m(d,k)} = \sqrt{\frac{dk}{m}}.$$

Note that the last equality follows from the expectation of a hypergeometric random variable. Now suppose $a \in (k-1, k]$. By the above inequality and apply Jensen's inequality one more time, we have

$$\begin{aligned}\tau_m(d,a) &= (k-a)\mathbb{E}\sqrt{L_m(d,k-1)} + (1-(k-a))\mathbb{E}\sqrt{L_m(d,k)} \\ &\le (k-a)\sqrt{\frac{d(k-1)}{m}} + (1-(k-a))\sqrt{\frac{dk}{m}} = \sqrt{\frac{da}{m}}.\end{aligned}$$

■

Lemma 10 *(Lower bounds for $\|\mathbf{z}\|_s^*$ and $\|\mathbf{z}\|_p^*$) Let $\mathbf{z} \in \mathbb{R}^m$. We have*

1. For $s \in [\![m]\!]$,

$$|||\mathbf{z}|||_s^* \geq \frac{s}{m} \max_{T \subset [\![m]\!]} \frac{\|\mathbf{z}[T]\|_1}{\tau_m(|T|, s)} \geq \max\left(\|\mathbf{z}\|_\infty, \sqrt{\frac{s}{m}} \max_{T \subset [\![m]\!]} \frac{\|\mathbf{z}[T]\|_1}{\sqrt{|T|}} \right).$$

2. For $p \in (0, 1)$,

$$\begin{aligned} |||\mathbf{z}|||_p^* &\geq p \max_{T \subset [\![m]\!]} \left\{ \Big(\sum_{k=0}^{m} \mathtt{pbinom}(k, m, p) \tau_m(|T|, k) \Big)^{-1} \|\mathbf{z}[T]\|_1 \right\} \\ &\geq p \max_{T \subset [\![m]\!]} \frac{\|\mathbf{z}[T]\|_1}{\tau_m(|T|, pm)} = \max\left(\|\mathbf{z}\|_\infty, \sqrt{p} \max_{T \subset [\![m]\!]} \frac{\|\mathbf{z}[T]\|_1}{\sqrt{|T|}} \right). \end{aligned}$$

Proof (1) Note that by definition,

$$|||\mathbf{z}|||_s^* = \sup_{\mathbf{w}} \frac{\mathbf{z}^T \mathbf{w}}{|||\mathbf{w}|||_s}$$

Let $d \in [\![m]\!]$ and $T \subset [\![m]\!]$ such that $|T| = d$. Define $\mathbf{w} \in \mathbb{R}^m$ such that $\mathbf{w}[i] = 1$ for $i \in T$ and $\mathbf{w}[i] = 0$ for $i \in T^c$. We have:

$$\begin{aligned} |||\mathbf{w}|||_s &= \binom{m-1}{s-1}^{-1} \sum_{|S|=s} \|\mathbf{w}[S]\|_2 = \binom{m-1}{s-1}^{-1} \sum_{l=\max(0,s+d-m)}^{\min(s,d)} \sum_{|S|=s, |S\cap T|=l} \|\mathbf{w}[S]\|_2 \\ &= \binom{m-1}{s-1}^{-1} \sum_{l=\max(0,s+d-m)}^{\min(s,d)} \sum_{|S|=s, |S\cap T|=l} \sqrt{l} \\ &= \binom{m-1}{s-1}^{-1} \sum_{l=\max(0,s+d-m)}^{\min(s,d)} \binom{d}{l} \binom{m-d}{s-l} \sqrt{l} \\ &= \frac{m}{s} \mathbb{E} \sqrt{L_m(d, s)} = \frac{m}{s} \tau_m(d, s). \end{aligned}$$

Thus for all $d \in [\![m]\!]$ and any subset T such that $|T| = d$, we have shown

$$|||\mathbf{z}|||_s^* \geq \frac{s}{m} \frac{\|\mathbf{z}[T]\|_1}{\tau_m(d, s)}.$$

Note that if $d = 1$, $\mathbb{E}\sqrt{L_m(d, s)} = \frac{s}{m}$. Therefore

$$\frac{s}{m} \frac{\|\mathbf{z}[T]\|_1}{\tau_m(d, s)} \geq \|\mathbf{z}\|_\infty,$$

Moreover, by Lemma 9,

$$\tau_m(d, s) \leq \sqrt{\frac{ds}{m}}.$$

Hence we have

$$\frac{s}{m} \frac{\|\mathbf{z}[T]\|_1}{\tau_m(d, s)} \geq \sqrt{\frac{s}{m}} \frac{\|\mathbf{z}[T]\|_1}{\sqrt{d}},$$

and the first part of the claim follows.
(2) For the same $\mathbf{w} \in \mathbb{R}^m$ defined previously,

$$\begin{aligned} |||\mathbf{w}|||_p &= \sum_{k=0}^{m-1} \mathtt{pbinom}(k, m-1, p) |||\mathbf{w}|||_{k+1} \\ &= m \sum_{k=0}^{m-1} \mathtt{pbinom}(k, m-1, p) \frac{\tau_m(d, k+1)}{k+1} \\ &= m \sum_{k=0}^{m-1} \binom{m-1}{k} p^k (1-p)^{m-k-1} \frac{1}{k+1} \tau_m(d, k+1) \\ &= \frac{1}{p} \sum_{k=0}^{m-1} \binom{m}{k+1} p^{k+1} (1-p)^{m-(k+1)} \tau_m(d, k+1) \\ &= \frac{1}{p} \sum_{k=0}^{m} \binom{m}{k} p^k (1-p)^{m-k} \tau_m(d, k). \end{aligned}$$

Thus for all $d \in [\![m]\!]$ and any subset T such that $|T| = d$, we have shown

$$|||\mathbf{z}|||_p^* \geq p \Big(\sum_{k=0}^{m} \mathtt{pbinom}(k, m, p) \tau_m(d, k) \Big)^{-1} \|\mathbf{z}[T]\|_1.$$

Next, we will show

$$\sum_{k=0}^{m} \mathtt{pbinom}(k, m, p) \tau_m(d, k) \leq \tau_m(d, pm).$$

To this end, let us first notice that the LHS quantity is a binomial average of $\tau_m(d, k)$ with respect to k. By construction, $\tau_m(d, .)$ is piecewise linear. Furthermore, $\tau_m(d, .)$ is also concave by Lemma 13. Now let Y be a random variable having the binomial distribution with parameters m and p. By Jensen's inequality,

$$\sum_{k=0}^{m} \mathtt{pbinom}(k, m, p) \tau_m(d, k) = \mathbb{E} \tau_m(d, Y) \leq \tau_m(d, \mathbb{E}Y) = \tau_m(d, mp).$$

In particular, if $d = 1$, it is easy to see that $\tau_m(d, mp) = p$. So

$$p \left(\max_{T \subset [\![m]\!], |T|=1} \Big(\sum_{k=0}^{m} \mathtt{pbinom}(k, m, p) \tau_m(|T|, k) \Big)^{-1} \|\mathbf{z}[T]\|_1 \right) \geq \|\mathbf{z}\|_\infty.$$

On the other hand, by Lemma 9,

$$\tau_m(d, pm) \leq \sqrt{\frac{d}{m}} \sqrt{pm} = \sqrt{pd}.$$

Therefore

$$p\Big(\sum_{k=0}^{m}\texttt{pbinom}(k,m,p)\tau_m(d,k)\Big)^{-1}\|\mathbf{z}[T]\|_1 \geq \sqrt{p}\frac{\|\mathbf{z}[T]\|_1}{\sqrt{d}},$$

and the proof is complete. ■

Lemma 11 *(Upper bounds for $\|\mathbf{z}\|_s^*$ and $\|\mathbf{z}\|_p^*$) Let $\mathbf{z}\in\mathbb{R}^m$.*

1. *For $s\in[\![m]\!]$,*
$$\|\mathbf{z}\|_s^* \leq \max_{|S|=s}\|\mathbf{z}[S]\|_2.$$
2. *For $p\in(0,1)$,*
$$\|\mathbf{z}\|_p^* \leq \max_{|S|=k}\|\mathbf{z}[S]\|_2,$$
where $k=\lceil p(m-1)+1\rceil$.

Proof To establish the upper bound, we will use the equivalent formulation of $\|\|.\|\|_s^*$ in Corollary 5. For $S\subset[\![m]\!]$ of size s, as in Corollary 5, let $\mathbf{E}_S=\mathbf{I}[S,]/\binom{m-1}{s-1}$ where $\mathbf{I}\in\mathbb{R}^{m\times m}$ is the identity matrix. If we set $\mathbf{y}_S=\mathbf{z}[S]$, then $\sum_{|S|=s}\mathbf{E}_S^T\mathbf{y}_S=\mathbf{z}$ and so $\{\mathbf{y}_S\}$ is feasible. Therefore

$$\|\mathbf{z}\|_s^* \leq \max_{|S|=s}\|\mathbf{z}[S]\|_2.$$

The upperbound of $\|\mathbf{z}\|_p^*$ follows from the inequality $\|\mathbf{z}\|_p^*\leq\|\mathbf{z}\|_k^*$ for $k=\lceil p(m-1)+1\rceil$ by the second part of Lemma 8. ■

Corollary 8 *(1-sparse vectors) Let $\mathbf{z}=(z,0,\ldots,0)^T\in\mathbb{R}^m$. We have*

$$\|\mathbf{z}\|_s^*=\|\mathbf{z}\|_p^*=|z|.$$

Proof These are direct consequences of Lemma 10 and Lemma 11. ■

Corollary 9 *(All-constant vectors) Let $\mathbf{z}\in\mathbb{R}^m$ be such that $\mathbf{z}[i]=z$ for all $i\in[\![m]\!]$. We have*

1. $\|\mathbf{z}\|_s^*=\sqrt{s}|z|$.
2. $\|\mathbf{z}\|_p^*=mp\Big/\Big(\sum_{k=0}^m\texttt{pbinom}(k,m,p)\sqrt{k}\Big)|z|$.

Proof First of all, note that $L_m(m,k)=k$ and $\mathbb{E}\sqrt{L_m(m,k)}=\sqrt{k}$. Thus by Lemma 10 and 11, we have

$$\|\mathbf{z}\|_s^*=\sqrt{s}|z|.$$

So the first part of the claim is verified. Next, by Lemma 10,

$$\|\mathbf{z}\|_p^*\geq mp\Big(\sum_{k=0}^m\texttt{pbinom}(k,m,p)\sqrt{k}\Big)^{-1}|z|.$$

On the other hand, for S such that $|S|=s$, we can define

$$\mathbf{y}_S=\frac{mp}{\sqrt{s}}\Big(\sum_{k=0}^m\texttt{pbinom}(k,m,p)\sqrt{k}\Big)^{-1}(z,\ldots,z)^T\in\mathbb{R}^s,$$

For notation simplicity, let $c=\frac{1}{mp}\Big(\sum_{k=0}^m\texttt{pbinom}(k,m,p)\sqrt{k}\Big)$. As in Corollary 5, for $S\subset[\![m]\!]$, let $\mathbf{E}_S=\mathbf{I}[S,]/\binom{m-1}{|S|-1}$. For $i\in[\![m]\!]$, we have

$$\begin{aligned}\sum_{k=0}^{m-1}\texttt{pbinom}(k;m-1,p)\sum_{|S|=k+1}(\mathbf{E}_S^T\mathbf{y}_S)[i] &= c^{-1}\sum_{k=0}^{m-1}\texttt{pbinom}(k;m-1,p)\frac{1}{\sqrt{k+1}}\\ &= c^{-1}\frac{z}{mp}\sum_{k=0}^{m}\texttt{pbinom}(k;m,p)\sqrt{k}=z.\end{aligned}$$

Thus by Corollary 5,

$$\|\mathbf{z}\|_p^*\leq\max_S\|\mathbf{y}_S\|_2=mp\Big(\sum_{k=0}^m\texttt{pbinom}(k,m,p)\sqrt{k}\Big)^{-1}|z|,$$

and the proof is complete. ■

Lemma 12 *(Convexity of $\|\|\mathbf{z}\|\|_k$) Let $\mathbf{z}\in\mathbb{R}^m$, where $m\geq3$. For $k\in[\![m-2]\!]$, we have the following inequality*

$$\|\mathbf{z}\|_k+\|\mathbf{z}\|_{k+2}\geq2\|\mathbf{z}\|_{k+1}. \tag{28}$$

Proof We will first show that the claim is true for $k=m-2$. Notice that in this case $\|\mathbf{z}\|_{k+2}=\|\mathbf{z}\|_m=\|\mathbf{z}\|_2$. If $\|\mathbf{z}\|_2=0$, the inequality (28) is trivially true. Now suppose $\|\mathbf{z}\|_2>0$, dividing both sides of the inequality by $\|\mathbf{z}\|_2$, we have

$$\binom{m-1}{m-3}^{-1}\sum_{|S|=m-2}\frac{\|\mathbf{z}[S]\|_2}{\|\mathbf{z}\|_2}+1\geq2\binom{m-1}{m-2}^{-1}\sum_{|S|=m-1}\frac{\|\mathbf{z}[S]\|_2}{\|\mathbf{z}\|_2}.$$

Now let $\mathbf{x}=(x_1,\ldots,x_m)^T\in\mathbb{R}^m$ be such that $x_i=\mathbf{z}[i]^2/\|\mathbf{z}\|_2^2$. It suffices to show

$$\sum_{|S|=m-2}\Big(\sum_{i\in S}x_i\Big)^{1/2}+\frac{(m-1)(m-2)}{2}\geq(m-2)\sum_{i=1}^m\sqrt{1-x_i}, \tag{29}$$

for all $\mathbf{x} \geq 0$ entry-wise such that $\sum_i x_i = 1$. We will now prove the above inequality by induction on m. First of all, notice that for the base case where $m = 3$, we need to show:

$$\sqrt{x_1} + \sqrt{x_2} + \sqrt{x_3} + 1 \geq \sqrt{1 - x_1} + \sqrt{1 - x_2} + \sqrt{1 - x_3},$$

with the constraints $x_i \geq 0$ and $x_1 + x_2 + x_3 = 1$. For fixed x_3, let

$$f(x_1) = \sqrt{x_1} + \sqrt{1 - x_1 - x_3} + \sqrt{x_3} + 1 - \sqrt{x_1 + x_3} - \sqrt{1 - x_1} - \sqrt{1 - x_3}.$$

We will show that $f(x_1)$ is minimized at $x_1 = 0$ or $x_1 = 1 - x_3$. Suppose now $x_1 > 0$. Taking derivative with respect to x_1:

$$f'(x_1) = \frac{1}{2}\Big(\frac{1}{\sqrt{x_1}} - \frac{1}{\sqrt{1 - x_1 - x_3}} - \frac{1}{\sqrt{x_1 + x_3}} + \frac{1}{\sqrt{1 - x_1}}\Big).$$

Let $l(x_1) = \frac{1}{\sqrt{x_1}} - \frac{1}{\sqrt{x_1 + x_3}}$. Note that $f'(x_1) = \frac{1}{2}l(x_1) - \frac{1}{2}l(1 - x_3 - x_1)$. Now we have

$$l'(x_1) = \frac{1}{2}(x_1 + x_3)^{-3/2} - \frac{1}{2}{x_1}^{-3/2}.$$

So $l(x_1)$ is decreasing on $(0, 1 - x_3)$ and by symmetry the function $l(1 - x_3 - x_1)$ is increasing on $(0, 1 - x_3)$. On the other hand, since $\lim_{x_1 \downarrow 0^+} l(x_1) = +\infty$ and $\lim_{x_1 \downarrow 0^+} l(1 - x_3 - x_1) = -\infty$, we know that $f'(x_1) > 0$ on $(0, \frac{1-x_3}{2})$ and < 0 on $(\frac{1-x_3}{2}, 1 - x_3)$. Thus, the minimum of f can only be attained at the boundaries, i.e., $x_1 = 0$ or $x_1 = 1 - x_3$. In either case we have

$$\begin{aligned}&\sqrt{x_1} + \sqrt{x_2} + \sqrt{x_3} + 1 - \sqrt{1 - x_1} - \sqrt{1 - x_2} - \sqrt{1 - x_3}\\ \geq &\sqrt{x_2} + \sqrt{x_3} - \sqrt{1 - x_2} - \sqrt{1 - x_3} = 0,\end{aligned}$$

as $x_2 + x_3 = 1$. So we establish (29) for $m = 3$.

Suppose (29) is also true for $m = n - 1$. For $m = n$, similar to the $m = 3$ case, for fixed $x_3, \ldots, x_n$, define

$$f(x_1) = \sum_{|S|=n-2} \Big(\sum_{i \in S} x_i\Big)^{1/2} + \frac{(n-1)(n-2)}{2} - (n-2)\sum_{i=1}^{n}\sqrt{1 - x_i},$$

subject to $x_i \geq 0$ and $\sum_i x_i = 1$. Again, we will show f attains its minimum at either $x_1 = 0$ or $x_1 = 1 - \sum_{i=3}^n x_i$. Notice that

$$\begin{aligned}\sum_{|S|=n-2} \Big(\sum_{j \in S} x_j\Big)^{1/2} =& \sum_{|S|=n-3,1,2\notin S} \Big(x_1 + \sum_{j \in S} x_j\Big)^{1/2} + \sum_{|S|=n-4,1,2\notin S} \Big(x_1 + x_2 + \sum_{j \in S} x_j\Big)^{1/2}\\ &+ \sum_{|S|=n-3,1,2\notin S} \Big(x_2 + \sum_{j \in S} x_j\Big)^{1/2} + \Big(\sum_{j=3}^{n} x_j\Big)^{1/2}\\ =&\sum_{i=3}^{n}\Big(x_1 + \sum_{j=3}^{n} x_j - x_i\Big)^{1/2} + \sum_{3 \leq i < j \leq n} (1 - x_i - x_j)^{1/2}\\ &+ \sum_{i=3}^{n}(1 - x_1 - x_i)^{1/2} + \Big(\sum_{j=3}^{n} x_j\Big)^{1/2}.\end{aligned}$$

In addition,

$$\sum_{i=1}^{n}(1 - x_i)^{1/2} = (1 - x_1)^{1/2} + \Big(x_1 + \sum_{j=3}^{n} x_j\Big)^{1/2} + \sum_{i=3}^{n}(1 - x_i)^{1/2}.$$

Taking derivative with respect to x_1,

$$\begin{aligned}f'(x_1) =& \frac{1}{2}\Big(\sum_{i=3}^{n}\Big(x_1 + \sum_{j=3}^{n} x_j - x_i\Big)^{-1/2} - \sum_{i=3}^{n}(1 - x_1 - x_i)^{-1/2}\\ &+ (n-2)(1 - x_1)^{-1/2} - (n-2)\Big(x_1 + \sum_{i=3}^{n} x_i\Big)^{-1/2}\Big).\end{aligned}$$

Now let

$$l(x_1) = \sum_{i=3}^{n}\Big(x_1 + \sum_{j=3}^{n} x_j - x_i\Big)^{-1/2} - (n-2)\Big(x_1 + \sum_{j=3}^{n} x_j\Big)^{-1/2}.$$

So $2f'(x_1) = l(x_1) - l(1 - \sum_{i=3}^n x_i - x_1)$. Again

$$l'(x_1) = -\frac{1}{2}\sum_{i=3}^{n}\Big(x_1 + \sum_{j=3}^{n} x_j - x_i\Big)^{-3/2} + \frac{n-2}{2}\Big(x_1 + \sum_{j=3}^{n} x_j\Big)^{-3/2}.$$

It is easy to see that $l'(x_1) < 0$ and so $l(x_1)$ is decreasing on $(0, 1 - \sum_{i=3}^n x_i - x_1)$. On the other hand $\lim_{x_1 \downarrow 0^+} l(x_1) = +\infty$. By symmetry $f'(x_1) > 0$ on $\big(0, \frac{1}{2}(1 - \sum_{i=3}^n x_i - x_1)\big)$ and < 0 on $\big(0, \frac{1}{2}(1 - \sum_{i=3}^n x_i - x_1)\big)$. So f attains its minimum at $x_1 = 0$ or $x_1 = 1 - \sum_{i=3}^n x_i$. Hence we have

$$\begin{aligned}&\sum_{|S|=n-2} \Big(\sum_{i \in S} x_i\Big)^{1/2} + \frac{(n-1)(n-2)}{2} - (n-2)\sum_{i=1}^{n}(1 - x_i)^{1/2}\\ \geq&\Big(\sum_{|S|=n-3,1\notin S} + \sum_{|S|=n-2,1\notin S}\Big)\Big(\sum_{j \in S} x_j\Big)^{1/2} + \frac{(n-2)(n-3)}{2} - (n-2)\sum_{i=2}^{n}(1 - x_i)^{1/2}. \quad (30)\end{aligned}$$

By the induction assumption that (29) holds when $m = n - 1$, we have

$$\sum_{|S|=n-3,1\notin S} \Big(\sum_{j \in S} x_j\Big)^{1/2} + \frac{(n-2)(n-3)}{2} \geq (n-3)\sum_{i=2}^{n}(1 - x_i)^{1/2}.$$

Thus (30) is greater than or equal to

$$\begin{aligned}&-\frac{(n-2)(n-3)}{2} + \sum_{|S|=n-2,1\notin S} \Big(\sum_{j \in S} x_j\Big)^{1/2} + \frac{(n-1)(n-2)}{2} - (n-2) - \sum_{i=2}^{n}(1 - x_i)^{1/2}\\ =&\sum_{|S|=n-2,1\notin S} \Big(\sum_{j \in S} x_j\Big)^{1/2} - \sum_{i=2}^{n}(1 - x_i)^{1/2} = \sum_{i=2}^{n}(1 - x_i)^{1/2} - \sum_{i=2}^{n}(1 - x_i)^{1/2} = 0.\end{aligned}$$

Thus we have verified the claim that (29) and hence (28) holds for $k = m-2$ for all $m \geq 3$. To establish the case for general $1 \leq k \leq m-2$, we again perform induction on the (m,k)-tuple. Note that the base case $m = 3$ and $k = 1$ has been previously proved. Suppose (28) holds for $m = n-1$ and $1 \leq k \leq n-3$. Now consider $m = n$ and $1 \leq k < n-2$. Notice that

$$\begin{aligned}
|||\mathbf{z}|||_k =& \frac{1}{n-k}\binom{n-1}{k-1}^{-1} \sum_{|T|=n-1} \sum_{|S|=k, S\subset T} \|\mathbf{z}[S]\|_2 \\
=& (n-1)\binom{n-2}{k-1}^{-1} \sum_{|T|=n-1} \sum_{|S|=k, S\subset T} \|\mathbf{z}[S]\|_2 \\
=& (n-1) \sum_{|T|=n-1} |||\mathbf{z}[T]|||_k.
\end{aligned}$$

By the induction assumption, for all T such that $|T| = n-1$, we have:

$$|||\mathbf{z}[T]|||_k + |||\mathbf{z}[T]|||_{k+2} \geq 2|||\mathbf{z}[T]|||_{k+1}.$$

Therefore

$$|||\mathbf{z}|||_k + |||\mathbf{z}|||_{k+2} - 2|||\mathbf{z}|||_{k+1} = (n-1) \sum_{|T|=n-1} (|||\mathbf{z}[T]|||_k + |||\mathbf{z}[T]|||_{k+2} - 2|||\mathbf{z}[T]|||_{k+1}) \geq 0.$$

Thus the claim also holds for $m = n$ and $1 \leq k < n-2$, completing the proof. ■

A.6 Miscellaneous

Lemma 13 *(Concavity of $\mathbb{E}\sqrt{L_m(d,k)}$) Let $d \in [\![m]\!]$. For $k \in [\![m-2]\!]$, we have*

$$\mathbb{E}\sqrt{L_m(d,k)} + \mathbb{E}\sqrt{L_m(d,k+2)} \leq 2\mathbb{E}\sqrt{L_m(d,k+1)}. \tag{31}$$

where the geometric random variable $L_m(d,k)$ is defined as in Definition 3.

Proof Suppose we are now sampling without replacement from a pool of numbers with d 1's and $m-d$ 0's. For $i \in [\![m]\!]$, denote by $X_i \in \{0,1\}$ the i-th outcome. It is easy to see that $L_m(d,k)$ and $\sum_{i=1}^k X_i$ have the same distribution. To show (31), it suffices to prove the following conditional expectation inequality:

$$\sqrt{L_m(d,k)} + \mathbb{E}[\sqrt{L_m(d,k+2)} \mid L_m(d,k)] \leq 2\mathbb{E}[\sqrt{L_m(d,k+1)} \mid L_m(d,k)]$$

Note that the above inequality follows if for all $0 \leq a \leq \min(d,k)$:

$$\sqrt{a} + \mathbb{E}\sqrt{a + X_{k+1} + X_{k+2}} \leq 2\mathbb{E}\sqrt{a + X_{k+1}}$$

It is easy to see that

$$\begin{aligned}
\mathbb{E}\sqrt{a+X_{k+1}} =& \frac{d-a}{m-k}\sqrt{a+1} + (1 - \frac{d-a}{m-k})\sqrt{a}. \\
\mathbb{E}\sqrt{a+X_{k+1}+X_{k+2}} =& \frac{d-a}{m-k} \times \frac{d-a-1}{m-k-1}\sqrt{a+2} + 2 \times \frac{d-a}{m-k} \times \frac{m-k-(d-a)}{m-k-1}\sqrt{a+1} \\
&+ \frac{m-k-(d-a)}{m-k} \times \frac{m-k-(d-a)-1}{m-k-1}\sqrt{a}.
\end{aligned}$$

By elementary algebra, it can be shown that

$$\begin{aligned}
&2\mathbb{E}\sqrt{a+X_{k+1}} - \sqrt{a} - \mathbb{E}\sqrt{a+X_{k+1}+X_{k+2}} \\
=& \frac{d-a}{m-k} \times \frac{d-a-1}{m-k-1} \times (2\sqrt{a+1} - \sqrt{a+2} - \sqrt{a}) \geq 0,
\end{aligned}$$

The inequality follows since $f(x) = \sqrt{x}$ is a concave function. Thus the proof is complete. ■

Lemma 14 *Let $\mathbf{x}(t) = (x_1(t), \ldots, x_m(t))^T \in \mathbb{R}^m$ be an m-dimensional function on $[0,\epsilon)$ such that: (1) $x_1(0) = 1$ and for all $i \geq 2$, $x_i(0) = 0$; (2) The derivative $\dot{x}_i(t)$ exists and is bounded for all $t \in (0,\epsilon)$. We have*

$$\lim_{t\downarrow 0^+} \frac{\|\mathbf{x}(t)\|_2 - \|\mathbf{x}(0)\|_2}{t} = \lim_{t\downarrow 0^+} \dot{x}_1(t).$$

Proof

$$\begin{aligned}
\lim_{t\downarrow 0^+} \frac{\|\mathbf{x}(t)\|_2 - \|\mathbf{x}(0)\|_2}{t} &= \lim_{t\downarrow 0^+} \frac{(\sum_{i=1}^m x_i^2(t))^{1/2} - 1}{t} \\
&= \lim_{t\downarrow 0^+} \frac{\sum_{i=1}^m x_i^2(t) - 1}{t} \left((\sum_{i=1}^m x_i^2(t))^{1/2} + 1 \right)^{-1} \\
&= \frac{1}{2} \lim_{t\downarrow 0^+} \frac{\sum_{i=1}^m x_i^2(t) - 1}{t} \\
&= \frac{1}{2} \left(\lim_{t\downarrow 0^+} \frac{x_1^2(t) - 1}{t} + \sum_{i=2}^m \lim_{t\downarrow 0^+} \frac{x_i^2(t)}{t} \right) \\
&= \frac{1}{2} \left(\lim_{t\downarrow 0^+} \frac{x_1(t) - 1}{t}(x_1(t) + 1) + \sum_{i=2}^m \lim_{t\downarrow 0^+} \frac{x_i^2(t)}{t} \right) \\
&= \lim_{t\downarrow 0^+} \frac{x_1(t) - 1}{t} + \frac{1}{2} \sum_{i=2}^m \lim_{t\downarrow 0^+} \frac{x_i^2(t)}{t}.
\end{aligned}$$

By mean value theorem, for each $t \in (0,\epsilon)$, there exists $\delta_t \in (0,t)$ such that $x_1(t) - 1 = \dot{x}_1(\delta_t)t$. Thus the first term simply becomes $\lim_{t\downarrow 0^+} \dot{x}_1(t)$. By the same argument, for each $i \in \{2, \ldots, m\}$, $x_i(t) = \dot{x}_i(\delta_t)t$ for some $\delta_t \in (0,t)$. Since $\dot{x}_i(t)$ is bounded, we have

$$\lim_{t\downarrow 0^+} \frac{x_i^2(t)}{t} = \lim_{t\downarrow 0^+} \dot{x}_i(\delta_t)^2 t = 0.$$

Therefore the claim is verified. ■

Lemma 15 *Let $\mathbf{x}(t) = (x_1(t), \ldots, x_m(t))^T \in \mathbb{R}^m$ be an m-dimensional function on $[0,\epsilon)$ such that: (1) $x_i(0) = 0$ for all $i = 1, \ldots, m$; (2) The derivative $\dot{x}_i(t)$ exists for all $t \in (0,\epsilon)$. We have*

$$\lim_{t\downarrow 0^+} \frac{\|\mathbf{x}(t)\|_2}{t} = \| \lim_{t\downarrow 0^+} \dot{\mathbf{x}}(t)\|_2.$$

Proof

$$\lim_{t\downarrow 0^+}\frac{\|\mathbf{x}(t)\|_2}{t}=\lim_{t\downarrow 0^+}\left(\sum_{i=1}^m(\frac{x_i(t)}{t})^2\right)^{1/2}=\left(\sum_{i=1}^m(\lim_{t\downarrow 0^+}\frac{x_i(t)}{t})^2\right)^{1/2}=\|\lim_{t\downarrow 0^+}\dot{\mathbf{x}}(t)\|_2.$$

■

Lemma 16 *Let* $\mathbf{a}=(a_1,\ldots,a_m)^T\in\mathbb{R}^m$ *where* $a_1\neq 0$ *and* $\mathbf{x}(t)=(x_1(t),\ldots,x_m(t))^T\in\mathbb{R}^m$ *be an* m*-dimensional function on* $[0,\epsilon)$ *such that: (1)* $x_1(0)=1$ *and for all* $i\geq 2$, $x_i(0)=0$*; (2) The derivative* $\dot{x}_i(t)$ *exists and is bounded for all* $t\in(0,\epsilon)$*. We have*

$$\lim_{t\downarrow 0^+}\frac{|\mathbf{a}^T\mathbf{x}(t)|-|a_1|}{t}=|a_1|\lim_{t\downarrow 0^+}\dot{x}_1(t)+\boldsymbol{sgn}(a_1)\sum_{i=2}^m a_i\lim_{t\downarrow 0^+}\dot{x}_i(t).$$

Proof Without loss of generality, assume $a_1>0$. Since $x_1(0)=1$ and for all $i\geq 2$, $x_i(0)=0$, by continuity, for sufficiently small t, we have

$$\frac{|\mathbf{a}^T\mathbf{x}(t)|-|a_1|}{t}=\frac{|a_1x_1(t)+\sum_{i=2}^m a_ix_i(t)|-a_1}{t}=\frac{a_1x_1(t)-a_1+\sum_{i=2}^m a_ix_i(t)}{t}.$$

Therefore, by the same argument in the proof of Lemma 14,

$$\begin{aligned}\lim_{t\downarrow 0^+}\frac{|\mathbf{a}^T\mathbf{x}(t)|-|a_1|}{t}&=\lim_{t\downarrow 0^+}\frac{a_1x_1(t)-a_1}{t}+\lim_{t\downarrow 0^+}\sum_{i=2}^m\frac{a_ix_i(t)}{t}\\&=a_1\lim_{t\downarrow 0^+}\dot{x}_1(t)+\sum_{i=2}^m a_i\lim_{t\downarrow 0^+}\dot{x}_i(t).\end{aligned}$$

■

Lemma 17 *Let* $\mathbf{a}=(a_1,\ldots,a_m)^T\in\mathbb{R}^m$ *and* $\mathbf{x}(t)=(x_1(t),\ldots,x_m(t))^T\in\mathbb{R}^m$ *be an* m*-dimensional function on* $[0,\epsilon)$ *such that: (1)* $x_i(0)=0$ *for all* $i=1,\ldots,m$*; (2) The derivative* $\dot{x}_i(t)$ *exists for all* $t\in(0,\epsilon)$*. We have*

$$\lim_{t\downarrow 0^+}\frac{|\mathbf{a}^T\mathbf{x}(t)|}{t}=\left|\sum_{i=1}^m a_i\lim_{t\downarrow 0^+}\dot{x}_i(t)\right|.$$

Proof The proof is similar to that of Lemma 15. ■

Appendix B. Additional Simulations

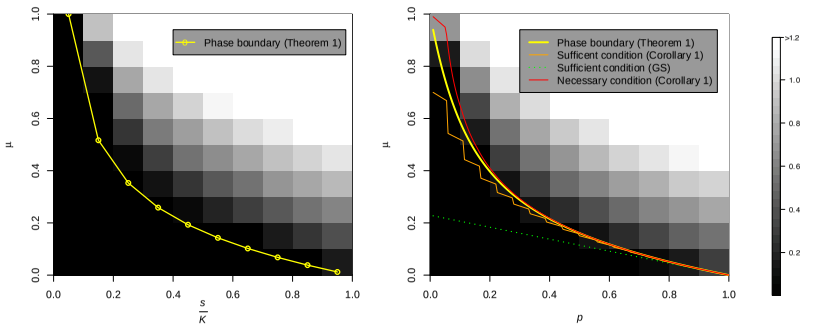


Figure B.1: Local recovery errors for the s-sparse Gaussian model (Left) and the Bernoulli(p)-Gaussian model (Right), with the number of dictionary atoms $K=20$. See Figure 1 for simulation details.

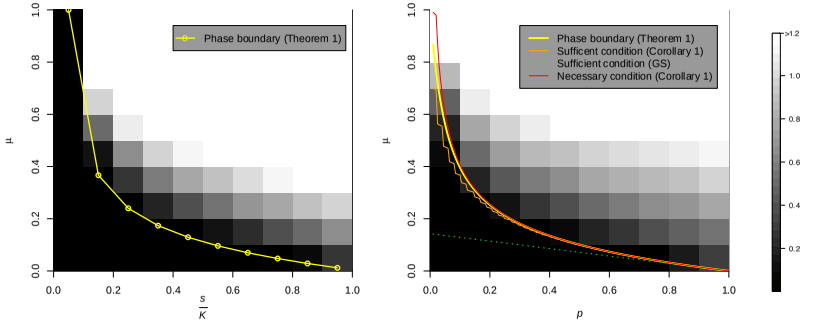


Figure B.2: Local recovery errors for the s-sparse Gaussian model (Left) and the Bernoulli(p)-Gaussian model (Right), with the number of dictionary atoms $K=50$. See Figure 1 for simulation details.

References

P.A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

Alekh Agarwal, Animashree Anandkumar, Prateek Jain, and Praneeth Netrapalli. Learning sparsely used overcomplete dictionaries via alternating minimization. *arXiv preprint arXiv:1310.7991v2*, 2014a.

Alekh Agarwal, Animashree Anandkumar, Prateek Jain, Praneeth Netrapalli, and Rashish Tandon. Learning sparsely used overcomplete dictionaries. In *Conference on Learning Theory*, pages 123–137, 2014b.

Alekh Agarwal, Animashree Anandkumar, and Praneeth Netrapalli. Exact recovery of sparsely used overcomplete dictionaries. *arXiv preprint arXiv:1309.1952*, 2014c.

Michal Aharon, Michael Elad, and Alfred M. Bruckstein. On the uniqueness of overcomplete dictionaries, and a practical way to retrieve them. *Linear Algebra and its Applications*, 416(1):48–67, 2006a.

Michal Aharon, Michael Elad, and Alfred M. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006b.

Sanjeev Arora, Rong Ge, Ravindran Kannan, and Ankur Moitra. Computing a nonnegative matrix factorization–provably. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, pages 145–162, 2012a.

Sanjeev Arora, Rong Ge, Ankur Moitra, and Sushant Sachdeva. Provable ICA with unknown gaussian noise, with implications for gaussian mixtures and autoencoders. In *Advances in Neural Information Processing Systems*, pages 2375–2383, 2012b.

Sanjeev Arora, Rong Ge, and Ankur Moitra. New algorithms for learning incoherent and overcomplete dictionaries. In *Conference on Learning Theory*, pages 779–806, 2014.

Sanjeev Arora, Rong Ge, Tengyu Ma, and Ankur Moitra. Simple, efficient, and neural algorithms for sparse coding. In *Conference on Learning Theory*, pages 113–149, 2015.

Peter Bühlmann and Sara van de Geer. *Statistics for High-Dimensional Data*. Springer, 2011.

Emanuel Candes and Terrence Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

Pierre Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3): 287–314, 1994.

David L. Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization. *Proceedings of the National Academy of Sciences of the United States of America*, 100(5):2197–2202, 2003.

Michael Elad. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer Science & Business Media, 2010.

Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006.

J-J Fuchs. On sparse representations in arbitrary redundant bases. *IEEE Transactions on Information Theory*, 50(6):1341–1344, 2004.

Quan Geng, Huan Wang, and John Wright. On the local correctness of ℓ^1-minimization for dictionary learning. *arXiv preprint arXiv:1101.5672*, 2011.

Pando Georgiev, Fabian Theis, and Andrzej Cichocki. Sparse component analysis and blind source separation of underdetermined mixtures. *IEEE Transactions on Neural Networks*, 16(4):992–996, 2005.

Rémi Gribonval and Karin Schnass. Dictionary identification—sparse matrix-factorisation via ℓ_1-minimisation. *IEEE Transactions on Information Theory*, 56(7):3523–3539, 2010.

Rémi Gribonval, Rodolphe Jenatton, Francis Bach, Martin Kleinsteuber, and Matthias Seibert. Sample complexity of dictionary learning and other matrix factorizations. *arXiv preprint arXiv:1312.3790*, 2013.

Rémi Gribonval, Rodolphe Jenatton, and Francis Bach. Sparse and spurious: dictionary learning with noise and outliers. *IEEE Transactions on Information Theory*, 61(11): 6298–6319, 2015.

Roger Grosse, Rajat Raina, Helen Kwong, and Andrew Ng. Shift-invariance sparse coding for audio classification. *arXiv preprint arXiv:1206.5241*, 2012.

Christopher Hillar and Friedrich T. Sommer. When can dictionary learning uniquely recover sparse data from subsamples? *IEEE Transactions on Information Theory*, 61(11):6290–6297, 2015.

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010.

Julien Mairal, Francis Bach, and Jean Ponce. Sparse modeling for image and vision processing. *Foundations and Trends in Computer Graphics and Vision*, 8(2-3):85–283, 2014.

Stephane Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 3rd edition, 2008.

Andreas Maurer and Massimiliano Pontil. k-dimensional coding schemes in Hilbert spaces. *IEEE Transactions on Information Theory*, 56(11):5839–5846, 2010.

Nishant A. Mehta and Alexander G. Gray. Sparsity-based generalization bounds for predictive sparse coding. In *Proceedings of the 30th International Conference on Machine Learning*, pages 36–44, 2013.

Bruno Olshausen and David Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37:3311–3325, 1997.

Gabriel Peyré. Sparse modeling of textures. *Journal of Mathematical Imaging and Vision*, 34(1):17–31, 2009.

Mark D. Plumbley. Dictionary learning for ℓ_1-exact sparse coding. In *Independent Component Analysis and Signal Separation*, pages 406–413. Springer, 2007.

Ben Recht, Christopher Re, Joel Tropp, and Victor Bittorf. Factoring nonnegative matrices with linear programs. In *Advances in Neural Information Processing Systems*, pages 1214–1222, 2012.

Ron Rubinstein, Alfred M. Bruckstein, and Michael Elad. Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6):1045–1057, 2010.

Karin Schnass. On the identifiability of overcomplete dictionaries via the minimisation principle underlying K-SVD. *Applied and Computational Harmonic Analysis*, 37(3):464–491, 2014.

Karin Schnass. Local identification for overcomplete dictionaries. *arXiv preprint arXiv:1401.6354v1*, 2015.

Daniel Spielman, Huan Wang, and John Wright. Exact recovery of sparsely-used dictionaries. *arXiv preprint arXiv:1206.5882*, 2012.

Ju Sun, Qing Qu, and John Wright. Complete dictionary recovery over the sphere I: Overview and the geometric picture. *IEEE Transactions on Information Theory*, 63(2): 853–884, 2017.

Robert Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288, 1996.

Daniel Vainsencher, Shie Mannor, and Alfred M. Bruckstein. The sample complexity of dictionary learning. *Journal of Machine Learning Research*, 12:3259–3281, 2011.

Yu Wang, Siqi Wu, and Bin Yu. Global identifiability of complete dictionary learning through ℓ_1-minimization. *Manuscript*.

Siqi Wu, Antony Joseph, Ann S. Hammonds, Susan E. Celniker, Bin Yu, and Erwin Frise. Stability-driven nonnegative matrix factorization to interpret spatial gene expression and build local gene networks. *Proceedings of the National Academy of Sciences of the United States of America*, 113(16):4290–4295, 2016.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, 68:49–57, 2006.

Michael Zibulevsky, Barak Pearlmutter, et al. Blind source separation by sparse decomposition in a signal dictionary. *Neural Computation*, 13(4):863–882, 2001.

Journal of Machine Learning Research 18 (2018) 1-32 Submitted 2/17; Revised 3/18; Published 4/18

In Search of Coherence and Consensus: Measuring the Interpretability of Statistical Topics

Fred Morstatter fredmors@isi.edu
University of Southern California
Information Sciences Institute
4676 Admiralty Way Ste. 1001
Marina Del Rey, CA 90292
Huan Liu huan.liu@asu.edu
Arizona State University
699 S. Mill Ave
Tempe, AZ 85283

Editor: David Blei

Abstract

Topic modeling is an important tool in natural language processing. Topic models provide two forms of output. The first is a predictive model. This type of model has the ability to predict unseen documents (e.g., their categories). When topic models are used in this way, there are ample measures to assess their performance. The second output of these models is the topics themselves. Topics are lists of keywords that describe the top words pertaining to each topic. Often, these lists of keywords are presented to a human subject who then assesses the meaning of the topic, which is ultimately subjective. One of the fundamental problems of topic models lies in assessing the quality of the topics from the perspective of human interpretability. Naturally, human subjects need to be employed to evaluate interpretability of a topic. Lately, crowdsourcing approaches are widely used to serve the role of human subjects in evaluation. In this work we study measures of interpretability and propose to measure topic interpretability from two perspectives: *topic coherence* and *topic consensus*. We start with an existing measure for topic coherence—model precision. It evaluates coherence of a topic by introducing an intruded word and measuring how well a human subject or a crowdsourcing approach could identify the intruded word: if it is easy to identify, the topic is coherent. We then investigate how we can measure coherence comprehensively by examining dimensions of topic coherence. For the second perspective of topic interpretability, we suggest topic consensus that measures how well the results of a crowdsourcing approach matches those given categories of topics. Good topics should lead to good categories, thus, high topic consensus. Therefore, if there is low topic consensus in terms of categories, topics could be of low interpretability. We then further discuss how topic coherence and topic consensus assess different aspects of topic interpretability and hope that this work can pave way for comprehensive measures of topic interpretability.

1. Introduction

Understanding natural language is one of the cornerstones of artificial intelligence research. Researchers can usually rely on text to give them signal for their specific problem. Text has been used to greatly aid artificial intelligence tasks such as opinion mining (Pang and Lee, 2008; Tumasjan et al., 2010), and user home location detection (Mahmud et al., 2012), and to find users in crisis situations (Morstatter et al., 2014). Topic modeling is one of the prominent text analysis techniques. Formally, topics are probability distributions over the words, but usually researchers treat them as lists of the most probable keywords. This process is akin to organizing newspaper articles by the "section" in which they appear, and simultaneously ranking words for that section. Topic models have been widely used for many tasks in social media research, such as using text to discover topics of discussion in crisis scenarios (Kireyev et al., 2009), event detection and analysis (Hu et al., 2012), and finding a Twitter user's home location (Eisenstein et al., 2010).

Topic modeling describes a family of approaches which work toward the above task. Latent Dirichlet Allocation (Blei et al., 2003), commonly known as LDA, is one example of a very popular topic model. LDA, and models like it, are used from two perspectives. The first is as a predictive model. When LDA is used in this way, the application is clear: there are many existing measures to assess the predictive performance. The other main use of LDA is to describe the dataset. The topics it learns are read by humans for them to get a better picture of the underlying themes in the dataset. When used this way, the topic distributions are manually inspected in many studies to show that some underlying pattern exists in the corpus. The meaning behind these topics is often interpreted by the individual who builds the model, and topics are often given a title or name to reflect their understanding of the underlying meaning of the topics. One key concern with topic models lies with how well human beings can actually understand the topics, or the problem of *topic interpretability*. It may be true that when presented with a group of words a human subject will always be able to assign some meaning. In this work, we question if we can tap on a group of human subjects or crowdsourcing in search of measures for topic interpretability. We focus on assessing the interpretability of topics from two perspectives: **coherence**, and **consensus**. Coherence measures how semantically close the top words of a topic are. Consensus measures how well the results of a crowdsourcing approach or generated by a group of human subjects match those given categories of topics.

The main contributions of this work are:

- We elaborate the need for measuring *topic coherence*, a novel dimension for measuring the semantic quality of topics. Based on this dimension, we propose "Model Precision Choose Two", a measure to comprehensively estimate how well a topic's top words are related to each other;
- We propose a *topic consensus* measure that estimates how well a statistical topic represents an underlying category of text in the corpus; and
- We demonstrate how these measures complement the existing framework and show how the results of these measures can help to further discover interpretable topics by a topic model.

2. Related Work

Topic modeling has been widely accepted in many communities such as machine learning, NLP, and social sciences (Ramage et al., 2009b). More recently topic modeling has been

widely applied to social media data. In the context of disaster-related tweets, Kireyev et al. (2009) tries to find disaster-related tweets, modeling two types of topics: informational and emotional. Joseph et al. (2012) studies the relation between users' posts and their geolocation. Several works (Yin et al., 2011; Hong et al., 2012; Pozdnoukhov and Kaiser, 2011) focus on identifying topics in geographical Twitter datasets, looking for topics that pertain to things such as local concerts and periods of mass unrest. Topic modeling was also used to find indication of bias in Twitter streams (Morstatter et al., 2013). Topic models exist have been developed to meet the unique needs of web data (Lin et al., 2014).

With such a wide acceptance, it is important that the topics produced by topic models are evaluated. Approaches to evaluating topic models follow two main avenues: evaluating the predictive performance of the model, and evaluating its interpretability. While we only focus on the latter in this work, we will cover the related work in the area of assessing the predictability before moving on to discuss the evaluation of interpretability.

2.1 Evaluating the Predictive Power of Topic Models

In the most general case, topic models are run over large corpora of data that do not contain a "ground truth" definition of the topics in the text. Because of this, we cannot apply supervised machine learning measures such as accuracy, precision, and recall to the task. Instead, the most often used measure for the predictive performance of topic models is "perplexity" (Jelinek et al., 1977; Jurafsky and Martin, 2000), which measures how well the topics match a set of held-out documents (Blei et al., 2003; Griffiths and Steyvers, 2004; Kawamae, 2016; Asuncion et al., 2009). Perplexity is defined as:

$$perp(q, x) = 2^{-\frac{1}{|x|}\sum_{i=1}^{|x|} log_2 q(x_i)}, \tag{1}$$

where q is the model we are testing, and x is the set of held-out documents. The intuition is that we are measuring how perplexed, or surprised, the model is. If the documents in x have a high probability of occurring, then the summation in the exponent will have a greater value, and thus the overall perplexity score will have a lower value.

Some specialized topic models can leverage ground truth labels. One such case is "Labeled LDA" (Ramage et al., 2009a), which is a different approach that takes labels into account when building the model. When evaluating its performance, traditional measures for supervised machine learning are applied such as F_1 (Frakes and Baeza-Yates, 1992), which is calculated as:

$$F_1 = \frac{2\pi\rho}{\pi + \rho}, \tag{2}$$

where $\pi = \frac{TP}{TP+FP}$ is the precision and $\rho = \frac{TP}{TP+FN}$ is the recall. Obtaining "true positives", "false positives", "true negatives", and "false negatives" requires data with ground truth, which is why we could not apply it when no labels are available.

2.2 Evaluating the Interpretability of Topic Models

Interpretability is largely a human issue. People generally read topics in order to assign meaning to them. Because of this, it is natural that humans would read topics in order to evaluate their interpretability. This issue has been addressed largely by two schools of

thought. The first school of thought is ad-hoc, where researchers manually read topics in order to judge their quality. In the second, researchers take a more principled approach, employing measures that can judge the quality of topics in a more automated manner. In the subsequent subsections, we provide more details regarding each approach and the related work that employs it.

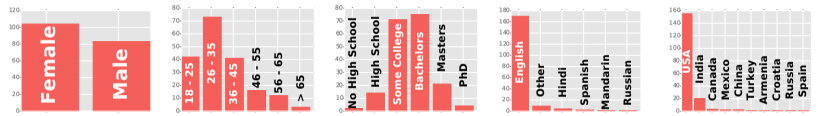
2.2.1 Eyeballing

The most common approach to assessing the quality of topics is the "eyeballing" approach, where topics are inspected carefully and manually assigned a label. After the topics are read, a "title" is assigned to each one based upon the top keywords in the text. In Grimmer and Stewart (2013), the authors manually label topics based upon the top words in the topic. These manual topic labels are supplemented with automatic labeling approaches such as Aletras and Stevenson (2014); Lau et al. (2011); Maiya et al. (2013); Mao et al. (2012), however the final call is made by a human. In other work, topic models were verified on scientific corpora to show that the topics that were produced by the model made sense (Blei et al., 2003; Griffiths and Steyvers, 2004). By displaying the top words from the topics to the reader, they make the case that the topics they find are of high quality. In the context of disaster-related tweets, Kireyev et al. (2009) try to find disaster-related tweets, labeling two types of topics: informational and emotional. This was done by interpreting a visualization of the topic clusters and manually assigning meaning to the topic groups. Other forms of visualization have been proposed to identify interpretable topics, such as that proposed in Le and Lauw (2016) where the authors show topics in a low-dimensional embedding. Another visualization approach is proposed in Sievert and Shirley (2014), where the authors show topics, their top words, and the size of the topic to the user to help them differentiate interpretable topics. Similarly, Hu et al. (2014) created an approach to iteratively add constraints to generate better topics.

The manual labeling of topics goes beyond mere text. For example, Schmidt (2012) uses LDA to cluster 1820's ship voyages. By treating trips as documents and nightly latitude/longitude checkins as words, the authors generate topics based upon these trips. Using manual inspection of topics, the authors are able to label topics as "trading" and "whaling" topics, amongst others. Other works focus on identifying topics in geographical Twitter datasets, looking for topics that pertain to things such as local concerts and periods of mass unrest (Yin et al., 2011; Hong et al., 2012).

While the manual inspection of topics is often used for topic labeling, it can also be used for topic filtering. In Kumar et al. (2013), the authors employ subject-matter experts to label the topics for them. In this case the authors had the labelers mark the topics as "relevant" to their study, or "not relevant". Ultimately those topics that were not deemed relevant were removed.

This method of evaluation, while common, has the issue that it is ad-hoc. This is a major problem in topic assessment as this evaluation can be subjective, sometimes coming down to just one researcher who assigns definitions to the topics learned from the model. To mitigate this issue, researchers have investigated imposing principled measures for topic interpretability. Additionally, this has implications for reproducibility, as a different researcher may have a different interpretation of the top words.



(a) Sex (b) Age (c) Education (d) First Language (e) Origin Country

Figure 1: Demographic breakdown of the Turkers who participated.

2.2.2 Principled Evaluation of Topic Interpretability

This method of assessing topic quality employs formal approaches and measures to assess the interpretability of statistical topics, generally through crowdsourcing. While aggregating the results of crowdsourced tasks is challenging, some work has focused on leveraging crowdsourcing to assess human interpretability (Zhou et al., 2014). Chang et al. (2009) proposed the first such framework for topic models in which a hybrid approach was employed. This approach focuses on crowdsourcing in order to assess topic interpretability. The results of the crowd are then aggregated through different measures to give a "score", which is a numerical value indicating the topic's interpretability. While not truly automated, it provides a reproducible framework that can be used by researchers to perform this assessment. In their paper, the authors focus on two main validation schemes for topic models: "Word Intrusion" which studies the top words within a topic by discovering how well participants can identify a word that does not belong. They also introduce "Topic Intrusion", which studies how well the topic probabilities for a document match a human's understanding of this document by showing three highly-probably *topics*, and one improbable topic, and asking the worker to select the "intruder".

Chang's influential paper provided the groundwork for a principled study of topic interpretability. Subsequent work aimed to provide automated approaches to replace the Turkers. Lau et al. (Lau et al., 2014) went about this by building heuristics to guess the actions of the workers. They provide an algorithm that will guess which answer a crowdsourced worker will choose when presented in an Human Intelligence Task (HIT). Other investigations into this measure include Roder et al. (Röder et al., 2015), who automatically explore the space of possible weights applied to existing interpretability measures and aggregation functions to find the measure that best approximates model precision.

2.3 Crowdsourcing Approaches to Evaluation

The crowdsourced experiments carried out in this work were performed using Amazon's Mechanical Turk[1] platform. Mechanical Turk is a crowdsourcing platform that allows requesters to coordinate Human Intelligence Tasks (HITs) to be solved by Turkers. The formulation of each HIT will be described in the corresponding section for each experiment. In all cases, each HIT was solved 8 times to overcome issues that arise from using non-expert annotators (Snow et al., 2008).

1. http://www.mturk.com

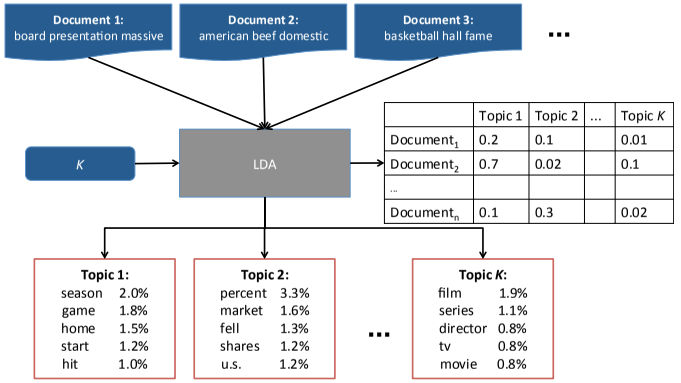


Figure 2: An overview of the LDA process. Each document is presented as a bag of words along with K, the number of topics the LDA operator wishes to discover. Two outputs are provided: the topics (on the bottom), and the document-topic associations (on the right).

In this paper we experiment with three measures that rely on crowdsourcing platforms like Amazon's Mechanical Turk. In order to ensure the reliability, as well as to account for noise in the results, it is important to have a basic understanding of the userbase solving the HITs we create. Prior to solving any HITs, we require the Turker to fill out a demographic survey. The demographic survey consists of five questions about the Turker's background: their sex, age, first language, country of origin, and highest level of education achieved. The demographic makeup of our Turkers can be seen in Figure 1. Figures 1(d) and 1(e) reveal a strong skew towards American Turkers who speak English as their first language. This could be partly attributed to a recent change in the Mechanical Turk terms of service that requires Turkers to provide their Social Security Number[2] in order to solve HITs on the site. Regardless, this allows us to go forward knowing that the participants are largely English speakers, and we cannot attribute poor performance in our analysis to a poor grasp of English.

This exercise in understanding the demographic makeup of our Turkers is done to give us a sense of the expected demographic makeup in future studies. We do not delete any Turkers who are non-native speakers. Instead, we investigate a Turkers' ability to solve our HITs based upon their performance at the "sanity" questions, the easiest HITs to solve in our set. These would be "control questions" to differentiate genuine workers (Liu et al., 2013). Initially, we planned to delete users who missed over 25% of these questions. Fortunately, no user fell below this threshold, and consequently no user was deleted from our study.

2. https://www.mturk.com/mturk/help?helpPage=worker#tax_no_have_tin

3. Topic Modeling

Topic modeling refers to a family of models that seek to discover abstract "topics" that occur within a corpus (Blei, 2012). Different approaches have been proposed for this task. For example, one of the first topic models was PLSA (Hofmann, 1999). LDA (Blei et al., 2003) built on this by adding a Dirichlet prior to the document-topic distribution. Many approaches have also been proposed for this task such as Hierarchical Dirichlet Processes (Teh et al., 2006), Correlated Topic Models (Blei and Lafferty, 2006), and Pachinko Allocation (Li and McCallum, 2006). While these are more recent than LDA, LDA is the most widely used topic modeling approach.

3.1 Latent Dirichlet Allocation

The goal of topic modeling is to learn "topics" from a large corpus of text. In LDA, each topic is a probability distribution over the entire vocabulary of the corpus. While each topic contains every word, the probabilities assigned to the word vary by topic. Furthermore, the model learns an association for each document over each of the topics. In other words, each document is described as a probability distribution over all of the topics in the corpus.

Formally, LDA takes two inputs:[3] A bag-of-words corpus containing d documents and a vocabulary of size v, and a scalar value K, which indicates the number of topics that the model will learn. LDA then outputs a model, m. The model, m, consists of two matrices:

1. A $Topic \times Vocabulary$ matrix, $\mathbf{T}^m \in \mathbb{R}^{K \times v}$. This is the matrix of topics that are learned by the model. $\mathbf{T}^m_{i,j}$ is the association of word j with topic i.

2. A $Document \times Topic$ matrix, $\mathbf{D}^m \in \mathbb{R}^{d \times K}$, with entry $\mathbf{D}^m_{i,j}$ representing the probability that document i is generated by topic j.

This is the fundamental input and output of the model, and can be seen in Figure 2. The model can be trained either through expectation maximization (Blei et al., 2003), or through Gibbs sampling (Griffiths and Steyvers, 2004). In this work we use the Mallet toolkit (McCallum, 2002), which uses the latter strategy to learn the parameters.

Beyond the notation, Figure 2 more clearly articulates the two schools of thought outlined in Section 2. When a researcher employs the "eyeballing" approach described in Section 2.2.1, what they are doing is having a human read the top words in the topic (depicted at the bottom of the figure) and divining a meaning for the topic. For example, a researcher may read the top words of "Topic 1" in Figure 2 and call it a "sports topic". While this ultimately may be an appropriate guess, there is no existing measure to determine the how well the definition fits the topic.

Principled evaluation, the second school of thought outlined in Section 2.2.2, consists of applying a standard framework to the practice of assessing topic quality. In (Chang et al., 2009), the authors propose a hybrid framework to assess topic quality, while (Mimno et al., 2011) propose a solution to do it automatically. The thrust of this work is to extend the existing principled framework in order to assess new dimensions of topic quality.

Going forward we will reproduce "Model Precision", one of the measures introduced in (Chang et al., 2009). Next, we will provide two new measures which can extend the existing framework and show how the insights they provide compare to the existing solution. The measures we propose are also hybrid, meaning that they depend on crowdsourcing to obtain their results. We note that while useful, this has major implications for reproducibility. These experiments require ample time and money in order to run, making it intractable for many researchers. Thus, we experiment with automated measures that can replace the effort performed by the crowdsourced workers.

3. For the sake of simplicity, we do not consider hyperparameters.

3.2 Data

We generate topics from LDA using two datasets. First, we use a dataset of scientific abstracts from the European Research Council. The text in these documents is of high quality, written by scientists who wish to argue their case in order to secure funding for their research. It is possible that topical misunderstanding can stem from a lack of understanding of the crowdsourced workers, who are not necessarily trained to understand scientific text. To complement this dataset, we use a large corpus of news articles curated by Yahoo News. We introduce both of these datasets in the subsequent sections.

3.2.1 Scientific Abstracts

The first text corpus focused upon in this study consists of 4,351 abstracts of accepted research proposals to the European Research Council.[4] In the first 7 years of its existence, the European Research Council (ERC) has funded approximately 4,500 projects, 4,351 of which are used in this study. Abstracts are limited to 2,000 characters, and when a researcher submits an abstract, they are required to select one of the three scientific domains their research fits into: Life Sciences (LS), Physical Sciences (PE), or Social Sciences and Humanities (SH). These labels will be used in the crowdsourced measure we propose later.

Mapping scientific research areas has become of growing interest to scientists, policymakers, funding agencies and industry. Traditional bibliometric data analyses such as cocitation analysis supply us with basic tools to map research fields. However, Social Sciences and Humanities (SH) are especially difficult to map and to survey, since the fields and disciplines are embedded in diverse and often diverging epistemic cultures. Some are specifically bound to local contexts, languages and terminologies, and the SH domain lacks coherent referencing bodies or citation indices, and dictionaries (Mayer and Pfeffer, 2014). Furthermore, SH terminology is often hard to identify as it resembles everyday speech. Innovative semantic technologies such as topic modeling promise alternative approaches to mapping SH, but the basic question here is: how interpretable are they and how can their results be evaluated in a systematic way? This raises further questions into the interpretability of the LDA topics we study in this paper.

The abstracts used in this research were accepted between 2007–2013, written in English, and classified by the authors[5] into one of the three main ERC domains. The first column of Table 1 shows some statistics of the corpus. The aim of each abstract is to provide a clear understanding of the objectives and methods to achieve them. Abstracts are also used to find reviewers or match authors to panels.

4. `http://erc.europa.eu/projects-and-results/erc-funded-projects`
5. The authors of the respective abstract, not the authors of this work.

Table 1: Properties of the European Research Council accepted abstracts and Yahoo News corpora. The values for each category represent the number of documents in the category.

Property	ERC Data	Yahoo News Data
Documents	4,351	258,919
Tokens	649,651	6,888,693
Types	10,016	214,957
Category 1	*LS:* 1,573	*S:* 88,934
Category 2	*PE:* 1,964	*B:* 90,159
Category 3	*SH:* 814	*E:* 79,826

3.2.2 News Data

The second text corpus used in this work consists of 258,919 news articles indexed by Yahoo News.[6] Yahoo News maintains a corpus of every news article published on its site between 2015-02-01 and 2015-06-03.[7] Similar to newspapers, these articles are tagged with a "section", which corresponds to a categorization. In this study, we select articles from three such categories: Sports (S), Business (B), and Entertainment (E).

The rationale for choosing this dataset is that discovering the topics of a corpus is very similar to automatically discovering the "sections" of a newspaper. When individuals read the top words of a topic, they often assign meanings to the topics, which could correspond to the types of categories seen in newspapers (Grimmer and Stewart, 2013). We employ this corpus because it can give us exactly this mapping, and, in the case of one of our measures, tells us exactly how well these topic understandings map onto the true distribution of the topic. Another important distinction of this dataset is that it consists of text that is meant to be read by everyone.

All of the articles in this corpus were written between February 1st, 2015 and June 3rd, 2015, in English, and classified by Yahoo into one of the three aforementioned categories. The summary statistics of our corpus can be seen in the second column of Table 1.

3.3 Extracting Topics from Text

We apply LDA to extract topics from the text. We run LDA on each dataset four times, with $K = 10, 25, 50, 100$, yielding a total of 185 ERC topics, and 185 Yahoo News topics. All LDA runs were carried out using the Mallet toolkit (McCallum, 2002). Before running LDA, we stripped the case of all words and removed stopwords according to the MySQL stopwords list.[8] Tokenization was performed using Mallet's preprocessing framework, using a special regular expression which preserves punctuation within words.[9] This allows for URLs, contractions, and possessives to be preserved. In all experiments, we fix the hyperparameter values to $\alpha = 5.0$ and $\beta = 0.01$.

6. `https://www.yahoo.com/news/`
7. `https://webscope.sandbox.yahoo.com/catalog.php?datatype=r&did=75`
8. `http://www.ranks.nl/stopwords`
9. We provided the regular expression '[\p{L}\p{P}]*\p{L}' to the "token-regex" argument in Mallet's import-file module.

Table 2: The LDA models generated for this study. These indicate the different values of "m" used throughout the experiments, *e.g.* in Equation 3.

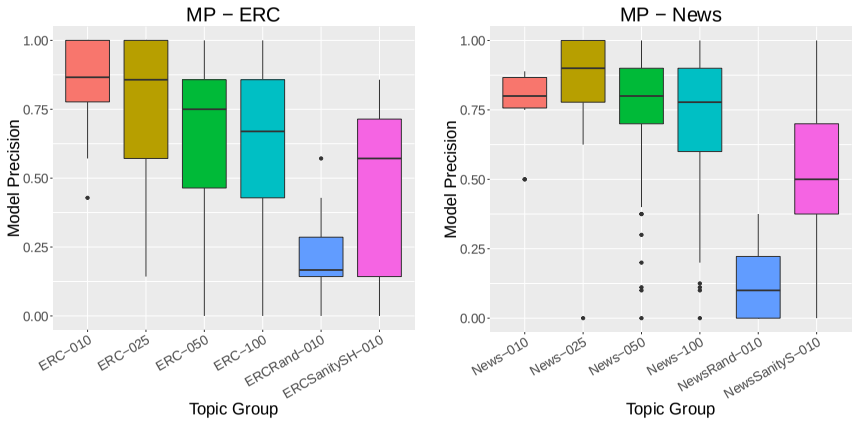
Name	Dataset	Strategy	Topics
ERC-010	ERC	LDA	10
ERC-025	ERC	LDA	25
ERC-050	ERC	LDA	50
ERC-100	ERC	LDA	100
ERCRand-010	ERC	Random	10
ERCSanitySH-010	ERC	Manual	10
News-010	News	LDA	10
News-025	News	LDA	25
News-050	News	LDA	50
News-100	News	LDA	100
NewsRand-010	News	Random	10
NewsSanityS-010	News	Manual	10

In addition to the LDA runs, we extract two additional topic groups from each corpus. This is done with the intent of giving some controls to understand the bounds of the topic measures. The first is a set of *random topics*. To generate these topics, we weight the words by their frequency in the corpus and randomly draw words from this distribution. These topics have a roughly equal mixture of the three categories in the corpus they represent.

To complement our random topics, we also create a set of "sanity" topics. These topics are handpicked to be a clear representation of one of the categories. In the ERC corpus, we calculate each word's probability of occurring in a SH, LS, or PE document. We then select words that occur most ($> 95\%$ of the time) in the SH category. Furthermore, we ensure that these words occur in fewer than $< 5\%$ of the documents from the other two categories. The intention behind these topics is that they provide a strictly pure representation of the SH category, and should provide as a useful sanity check of the Turker's labeling abilities. The same process is repeated with the Yahoo News data, by selecting topics similarly skewed towards the S category. These topics lie in contrast to the random topics in that they are strongly skewed to represent a single group.

In both cases of random and sanity topics, these topics are unlike traditional LDA topics, containing only a *set* of 5 words. We use both of these auxiliary topic sets for validation of the results obtained using the LDA topics. Table 2 shows an overview of all of the topic sets generated for this study.

These topic sets provide examples of how topic models can be applied to extract topics from real world data. We next investigate how to measure their interpretability from two perspectives: coherence, and consensus.



(a) ERC Data (b) Yahoo News Data

Figure 3: Results of "Model Precision" task on the topic sets from the two datasets. The horizontal bar represents the median, and the dots represent outliers.

4. Topic Coherence

One definition of coherence is "the quality or state of systematic or logical connection or consistency".[10] When a topic is a set of words, its coherence is about the relationships among the words. Since each individual can have their opinion of coherence, a crowdsourcing approach is used to obtain a group's feedback. This transforms an individual's opinion to a distribution of collective opinions. We first introduce Chang et al.'s ingenious measure of coherence.

4.1 Model Precision — A Measure of Coherence

Model Precision, introduced by (Chang et al., 2009), is a widely used measure of the "coherence" of an individual topic. It measures the distinctness of a randomly-inserted word into the top five words of a topic. The intuition is that if humans are consistently able to identify the randomly-inserted word, then the topic is more coherent because the intruded word is clearly distinct from the other 5 words. On the other hand, if the humans cannot consistently choose this randomly-inserted word, then the top 5 words of the topic are likely not coherent. This is because the humans are conflating the definition of the top words in the topic with a word that is far away.

For each topic, we show the Turker the top 5 most probable words from the topic's probability distribution along with one of the *least* probable words in the distribution. We call this word the "intruded" word. To prevent rare words from being selected as the

10. http://www.merriam-webster.com/dictionary/coherence

intruder, we ensure that it is also among the top 5 words from another topic. We then ask the Turker to select the word that they think is the intruded word. Model precision is the number of times a Turker was able to guess the intruded word divided by the number of times the HIT was solved, formally:

$$MP_k^m = \frac{1}{|\mathbf{s_k^m}|} \sum_{i=1}^{|\mathbf{s_k^m}|} \mathbb{1}(g_k^m = \mathbf{s}_{k,i}^m), \tag{3}$$

where MP_k^m is the model precision of the k-th topic from model m, g_k^m is the ground-truth intruded word for topic k, $\mathbf{s}_k^m$ is the vector of choices made by Turkers on topic k, and $\mathbb{1}(\cdot)$ is the indicator function which yields 1 if $g_k^m = \mathbf{s}_{k,i}^m$, and 0 otherwise.

The results of this measure on both of our datasets are shown in Figure 3. These figures show boxplots which depict the performance of the topics against this measure. Following traditional boxplot visualization techniques (Wickham and Stryjewski, 2011), the dark horizontal line in the middle of the box is the median, and the lower stem, lower box, upper box, and upper steam each account for 25% of the data. Dots represent topics determined to be outliers by a rule.[11]

We expect that the results would put the interpretability of the LDA topics somewhere in between the random and sanity topics. They should perform better than the random topics, as these are designed to be uninterpretable. Furthermore, our LDA topics should underperform when compared with the sanity topics, which are designed specifically to be highly interpretable. In fact, the results only partially back up this intuition. While the random topics do in fact perform very poorly, the LDA topics actually outperform the sanity topics. This is likely due to the homogeneity of the topics and the words from their vocabulary. When *all* of the choices are "SH" words in the ERC corpus, or "S" words in the News corpus, it is quite likely that the human workers will be confused.

While comparing the distributions of the results may shed light on the broad strokes, the outliers may also provide insight about the results of the experiments. For example, in both $K = 10$ cases, we see one outlier, indicating one underperforming topic. In the random topic set on the ERC data, we see one topic *over*performing, doing much better than the rest of the topics and even scoring near the median of the sanity topics. The four HITs are shown in Table 3. These results illuminate the issue of how these topics became outliers: the intruding word. In the bad topic modeling topics (rows 1, 2, and 3), the topics perform badly because they are not coherent. The words spread over many different concepts, and it is difficult to decipher the meaning. On the other hand the random topic which should have done badly (row 4) actually outperforms the rest of the topic set. Turkers coalesced around "can", and "seem" as the intruded words. Since these are randomly-generated topics, any word among the 6 could be considered "intruded", but only one answer is selected as correct from the setup of our HITs. It just so happened that "can" was the selected word, leading this topic to have an uncharacteristically high score.

These results may also give some inclination about how to use this measure. In fact, there are any number of ways to use these results. For example, if we set $K = 10$, we will get 10 topics from the dataset. On large corpora like those used here, we should expect these 10 topics to be very generic, encompassing major themes of the text. With $K = 100$ topics,

11. Outliers are determined using the "1.5 rule" (Frigge et al., 1989).

Table 3: Outlier topics from Model Precision.

Row	Model	Top 5 Words	Intruded Word
1	ERC-010	design, use, based, develop, can	accompany
2	News-010	news, state, network, april, day	bigeast.com
3	News-025	family, actress, life, -year-old, star	megan's
4	ERCRand-010	maps, resource, visual, manifestation, seem	can

we may also get generic topics, and additionally we are more likely to get fine-grained topics that discuss more specific issues. For example, if in the $K = 10$ case we get one "politics" topic, we may expect to get topics pertaining to specific candidates, issues, and localities in the $K = 100$ case. Interpretability measures can help to shed light on which topics are understandable by humans, and may give an indication of which topic model, or which topics within a model are the most interpretable.

In the following, we first examine what is not assessed by model precision in terms of coherence, and suggest the need for a new measure that can complement model precision for comprehensive measure of coherence.

4.2 A Missing Dimension of Model Precision

Model Precision works by asking the user to choose the word that does not fit within the rest of the set. We are measuring the top words in the topic by comparing them to an outlier. While this method is very useful for this task, it does not measure the coherence *within* the top words for the topic. This is because a good topic should have top words that are semantically close to each other, an aspect of topic quality which is not accounted for by Model Precision.

A diagram illustrating this phenomenon is shown in Figure 4. In Figure 4(a), we see a coherent topic. This topic is coherent because all 5 of the top words are close together, while the intruded word is far away. In Figure 4(b) we see a topic that is less coherent because the fifth word lies at a distance from the first four. In both cases, Model Precision gives us the intruder word in the topic, as seen in Figures 4(c), and 4(d). While this is the desired performance of Model Precision, it leaves us with no understanding of the coherence of the top words of the topic. Results are masked by the outlier, and do not give information about the intra-cluster distance, or coherence of the topic.

In light of this, we look for a way to separate topics not just by their distance from an outlier, but also by the distance within the top words in the topic. The next section of this paper investigates a method which can measure not just the intruder word, but also the coherence of the top words in the topic. In this way we separate topics such as those shown in Figure 4 based on the coherence of their top words.

4.3 Model Precision Choose Two — Another Dimension of Coherence

In this section we propose a new measure for the coherence of the top words of a topic. This experiment sets up the task as before: we select the top five words from a topic, and

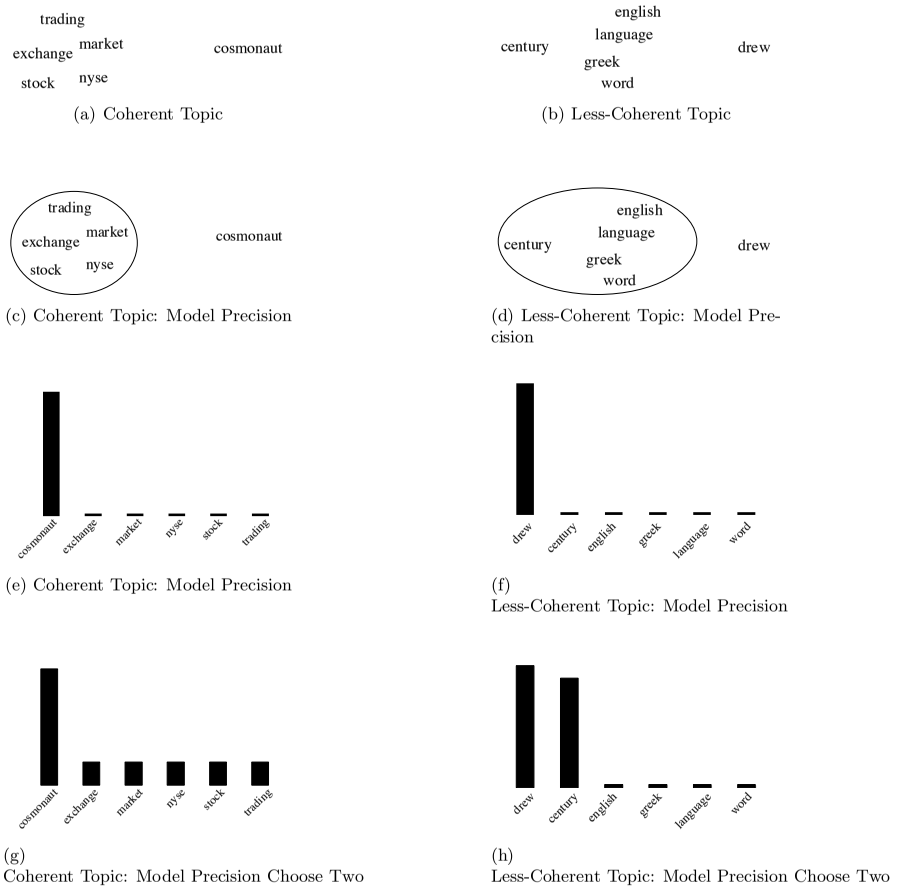


Figure 4: Comparison between Model Precision, and Model Precision Choose Two for two real topics from the Yahoo News corpus. In Figures 4(g) and 4(h), the height of the bars represents the number of times the word was selected in the crowdsourced experiments. Model Precision Choose Two can distinguish the less-coherent topic.

inject one low-probability word. The key difference is that we ask the Turker to select *two* intruded words among the six.

The intuition behind this experiment is that the Turkers' first choice will be the intruded word, just as in Model Precision. However, their second choice is what makes the topic's

quality clear. In a coherent topic the Turkers won't be able to distinguish a second word as all of the words will seem similar. A graphical representation of this phenomenon is shown in Figure 4(g). In the case of an incoherent, a strong "second-place" contender will emerge as the Turkers identify a 2nd intruder word, as in Figure 4(h).

4.3.1 Experimental Setup

To perform this experiment, we inject *one* low-probability word for each topic, and we ask the Turkers to select *two* words that do not fit within the group. We show the six words to the Turker in random order with the following prompt:

> You will be shown six words. Four words belong together, and two of them do not. Choose two words that do **not** belong in the group.

Coherent topics will cause the Turkers' responses regarding the second intruded word to be unpredictable. Thus, our measure of the goodness of the topic should be the predictability of the Turkers' second choice. We propose a new measure called "Model Precision Choose Two" to measure this. Model Precision Choose Two (MPCT) measures this spread as the peakedness of the probability distribution. We define $MPCT_k^m$ for topic k on model m as:

$$MPCT_k^m = H\big(p_{turk}(\mathbf{w}_{k,1}^m), ..., p_{turk}(\mathbf{w}_{k,5}^m)\big), \tag{4}$$

where $H(\cdot)$ is the Shannon entropy (Lin, 1991), $\mathbf{w}_k^m$ is the vector of the top words in topic k generated by model m, and $p_{turk}(\mathbf{w}_{k,i}^m)$ is the probability that a Turker selects $\mathbf{w}_{k,i}^m$. This measures the strength of the second-place candidate, with higher values indicating a smoother, more even distribution, and lower values indicating Turkers gravitation towards a second word.

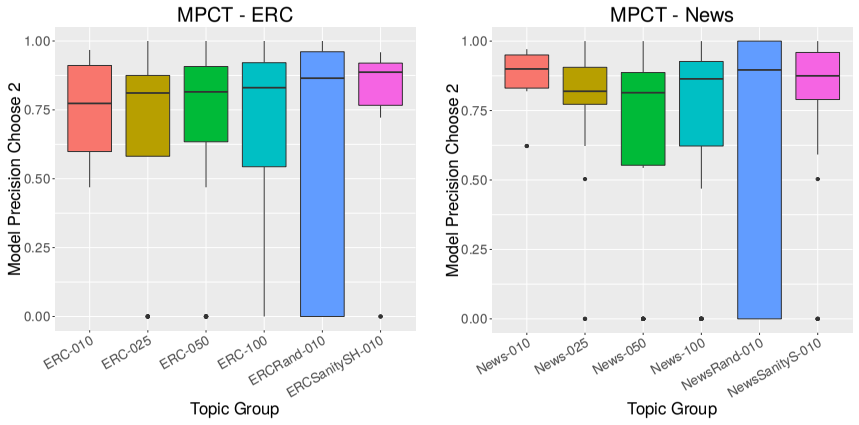
The intuition behind choosing entropy is that it will measure the unpredictability in the Turker selections. That is, if the Turkers are confused about which second word to choose, then their answers will be scattered amongst the remaining five words. As a result, the entropy will be *high*. Conversely, if the second word is obvious, the Turkers will begin to congregate around that second choice, meaning that their answers will be focused. As a result, the entropy will be *low*. Because entropy is able to measure the confusion of the Turkers responses about the second word, we use it directly in the design of our measure.

4.3.2 Results and Discussion of Model Precision Choose Two

The results of this experiment on both corpora are shown in Figure 5. The box plots show the distribution of the results across all of the topics in each topic group. These results illustrate the differences between the two datasets in terms of both performance as well as the appropriate value of K, the number of topics, to maximize the performance. In the ERC data a larger value of K consistently improved our results, while with the News dataset we achieved better results with a larger K.

The results of the MPCT experiments showed how to compute this measure on a topic set. While this new measure elicits another dimension of the topics, their coherence, they alone to not provide the whole picture of what makes a good topic.

Evidence of this is shown in both corpora in Figure 5. In both cases, the "sanity" topics do well, achieving among the best results. However, the median score from the "random"



(a) ERC Data (b) Yahoo News Data

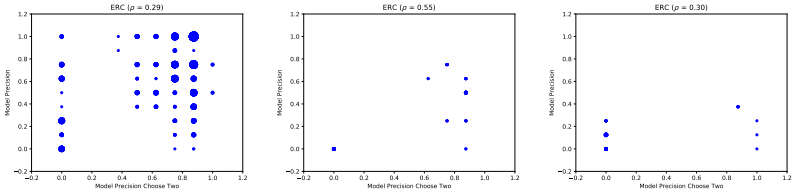
Figure 5: Results of "Model Precision Choose Two" on the topic sets from the two datasets. The figures are presented as before, with the horizontal bar indicating the median, and the dots representing outliers.

topics is also very high. It is easy to see why this occurs: in both very good *and* in very bad topic groups, the Turker has a difficult time choosing any second choice intruder, which drives the MPCT score up. Because of this, we need to combine these results with model precision in order to get a good understanding of the topic quality.

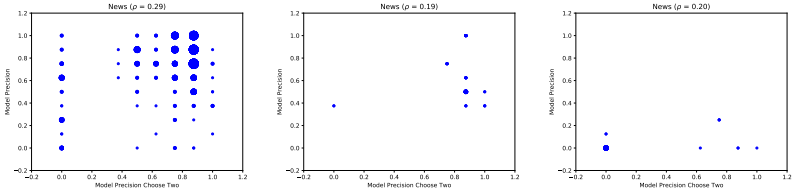
To better compare the results, we compare a case-by-case basis in Figure 6. Both yield similar results which show interesting properties about the dataset. First, Figures 6(c) and 6(f) confirm our hypothesis that random topics can have a high MPCT. However, these lousy topics all have a low MP score. Curiously, though, we see in Figures 6(a) and 6(d) show another interesting pattern: that it is possible for topics to have a high MP and a low MPCT.

To better demonstrate what these values mean for a topic, we show several topics alongside their MP and MPCT scores in Table 4. We show topics of varying quality from the perspective of both measures we introduce. By reading the topics, we can make several observations. First, row 1 and 2 have both high MP and high MPCT. Contrast these topics with rows 3 and 4 which have a high MP score, yet a low MPCT score. When we look at the intruded word in rows 1–4, they all seem equally "distant", however in rows 3–4, a second word also emerges to the Turkers.[12] In rows 5–8, all of the topics perform badly according to MP, but 5–6 are good according to MPCT, while 7–8 are bad according to both measures. After reading the topics in rows 5–8, it is difficult to understand the story. The results of this table tell us two things: 1) a high MP score is a necessary but not sufficient

12. In row 3, "large" was the word the turkers rallied around as the second choice. In row 4, it was "fans".



(a) ERC-* (b) ERCSanitySH-010 (c) ERCRand-010



(d) News-* (e) NewsSanityS-010 (f) NewsRand-010

Figure 6: Scatter plots of each topic's Model Precision Choose Two (x-axis), and Model Precision (y-axis) for each corpus. Above the plot is the correlation (ρ) of the values for all of the topics in the plot. The radius of the circle indicates the number of topics that received the score. "-*" means that all LDA-generated topic groups were considered for this scatter plot.

Table 4: Topics with varying MP and MPCT scores from models built on the two corpora in this work.

Row	Model	Top 5 Words	Intruded Word	MP Score	MPCT Score
1	ERC-100	production, plants, provide, food, plant	suppressor	1.00	0.99
2	News-100	number, system, transactions, card, money	flees	1.00	0.97
3	ERC-50	methods, data, information, analysis, large	diesel	1.00	0.00
4	News-25	series, fans, season, show, episode	leveon	1.00	0.00
5	ERC-100	nuclear, fundamental, water, understanding, surface	modularity	0.13	0.92
6	News-100	film, khan, ians, actor, bollywood	debonair	0.30	1.00
7	ERC-50	mechanisms, pathways, involved, molecular, role	specialized	0.00	0.00
8	News-100	injury, left, list, return, surgery	tests-results	0.00	0.25

condition to identify interpretable topics, and 2) by measuring coherence with MPCT we can identify quality topics better than with MP alone. A confusion matrix showing the differences between MP and MPCT are shown in Table 5.

Model Precision Choose Two is a new measure that rounds out the measurement of topic coherence. Coherence is one aspect of topic interpretability, however, we introduced two aspects of topic interpretability. We next investigate the interpretability from the second perspective: consensus.

Table 5: Qualitative assessment of the difference between low and high values. When MP is low, a topic is not interpretable regardless of the value of MPCT; when MP is high, a topic is interpretable when MPCT is high, otherwise it has limited interpretability. This is because high MP alone cannot reveal how interpretable a topic is as the topic is differentiable from an outlier term.

		MP	
		Low	**High**
MPCT	**High**	Uninterpretable	Interpretable
	Low	Uninterpretable	Uninterpretability Masked by Outliers

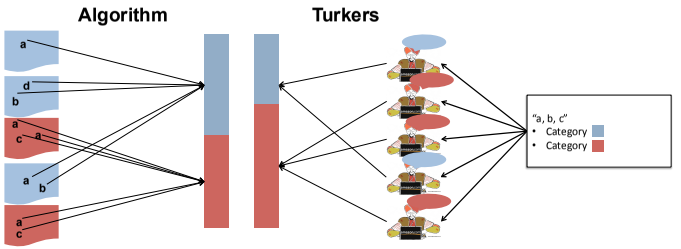


Figure 7: An overview of the setup of the topic consensus framework. The left stacked bar comes from the categories of the documents in which tokens appear. The right stacked bar comes from the aggregate of the Turker answers.

5. Topic Consensus

Understanding the underlying distribution of concepts in a topic is important. Since LDA is a mixture model, we can expect some of the topics identified by this model to contain a mixture of different topics. This phenomenon has been documented in the literature as "chimera" topics (Schmidt, 2012). Being able to identify these topics effectively is important, but we currently do not know how well crowdsourced workers will be able to identify them. In this section we propose a measure, "topic consensus," which measures how well the mixture of the documents in the LDA topic matches the mixture of labels from the workers.

Unlike topic coherence, topic consensus measures how well the results of a mixture of labels given by a group of human subjects match those given categories of topics, or human labels of topics. The "eyeballing" approach to topic interpretability, as discussed in Section 2.2.1 is the process of manually reading a topic and assigning it a title. These are explicit

titles such as "Environment", and "Judiciary" categories from congressional records (Grimmer and Stewart, 2013), or "disaster-related" categories from social media (Kireyev et al., 2009). Each title requires a human being to manually read these topics and to assign these category labels. Presumably, good topics should lead to good categories, thus, high topic consensus. Therefore, if there is low topic consensus in terms of categories, topics could be of low interpretability.

5.1 A New Measure for Topic Consensus

We discuss how well the topics from topic modeling conform to the natural categories underlying the text. Explicit topic categories are often present in many corpora, such as newspaper articles, due to their manual categorization by "sections" or categories. For example, our ERC abstracts which explicitly label each abstract with an ERC category, and we measure this conformity to the underlying topic distribution by leveraging the ground-truth topic category labels available as the ERC categories when the abstract is submitted.

The Turkers' answers reveal the labels that are assigned to topics. By showing the categories to the Turkers as multiple choice questions we can see the category label that a human would assign to the topic. Ideally, we would additionally ask the Turkers to assign confidence scores to their labels to better understand their labeling strategy and to get a better distribution of the category labels. However, since humans are bad at answering questions about themselves (i.e. their own internal confidence) (Bernard and Ryan, 2009), we instead ask many Turkers the same question about the same topic and aggregate the responses. By aggregating the category assignment of Turkers, we can obtain a distribution based on their understanding of the topic.

To understand how well the statistical topics mimic the underlying topics, we show the Turkers the top 20 words of a statistical topic and ask them to choose which of the three categories from that corpus the topic describes. For example, for a given ERC topic, we show the top 20 words along with options for "Life Sciences", "Physical Sciences", or "Social Sciences". We also provide a fourth option, "No Topic Matched", in case that any of the three categories do not make sense to the Turker. This is depicted in the right half of Figure 7, where Turkers are shown the HIT including top 3 words with 2 categories, and their answers are aggregated to make the distribution of Turker answers. In the figure, instead of 20 top words, top 3 words "a", "b", and "c" are chosen to be presented to Turkers.

To compute topic consensus, we compare the distribution of the Turkers' responses for that topic with the distribution of the topic over the ERC categories. To perform this analysis, we construct an LDA Topic × Category matrix $\mathbf{R}$, where $\mathbf{R}_{i,c}$ indicates topic i's probability of occurring in ERC category c. This can be seen in the lefthand side of Figure 7, where the tokens that are labeled with the topic are aggregated based upon the category label of the document they appear in to form the category distribution.

The structure of each row of $\mathbf{R}$ is dependent on the type of topic group it comes from. We construct the row of $\mathbf{R}$, which correspond to the automated distributions, as follows for each topic group:

- **ERC-* / News-*** — The $\mathbf{R}_i$ row vector for an ERC topic is created by taking the sum of the columns of the $\mathbf{D}$ matrix, as defined in Section 3.1. This sum is taken for each row (document) of $\mathbf{D}$ labeled with the corresponding ERC category. This is defined as:

$$\mathbf{R}_{i,c} = \frac{\sum_{j \in M_c} \mathbf{D}_{j,i}}{\sum \mathbf{D}_{*,i}}, \tag{5}$$

  where M_c is the set of documents containing the label corresponding to the column of $\mathbf{R}$, e.g., "SH", "LS", or "PE". This gives us an understanding of the category makeup for each LDA topic.

- **ERCSanitySH-010 / NewsSanityS-010** — The $\mathbf{R}_i$ row vector for an SH topic contains a 1 for the sanity category and a 0 for the others. This is due to the way the topics are generated, they contain purely words from that topic.

- **ERCRand-010 / NewsRand-010** — Turkers should not be able to read any definition from a random topic as it consists of random words from the vocabulary. Thus, the row vector for each topic in this set is a 1 for the "N/A" category, and a 0 for the other categories.

Using the responses from the Turkers, we build a separate $Topic \times Category$ matrix, $\mathbf{R^{AMT}}$ where $\mathbf{R^{AMT}}_{i,j}$ represents the Turkers' probability of choosing category j when presented with topic i. In this way, $\mathbf{R^{AMT}}$ is the representation of $\mathbf{R}$ obtained from the Turkers' responses. A row in $\mathbf{R^{AMT}}$ indicates the distribution over categories for a given LDA topic from the Turker's responses.

The consensus between the responses from the crowdsourced workers and the data is defined as:

$$\text{consensus}_t^m = 1 - \frac{JS(\mathbf{R^{AMT}}_t || \mathbf{R}_t)}{log_2(|c|+1)}, \tag{6}$$

where $|c|$ is the number of categories in the datasets (both the ERC and News datasets have $|c| = 3$ categories). We add 1 to account for the presence of the N/A answer. JS is the Jensen-Shannon divergence (Lin, 1991) between the two distributions $JS(\mathbf{R^{AMT}}_t || \mathbf{R}_t)$, defined as:

$$JS(\mathbf{R^{AMT}}_t || \mathbf{R}_t) = \frac{K(\mathbf{R^{AMT}}_t || M) + K(\mathbf{R}_t || M)}{2}, \tag{7}$$

where K is Kullback-Leibler divergence (Joyce, 2011), and $M = \frac{1}{2}(\mathbf{R^{AMT}}_t + \mathbf{R}_t)$. Jensen-Shannon is a natural choice as the rows of $\mathbf{R}$ and $\mathbf{R^{AMT}}$ are probability distributions over the 3 ERC or News categories and Jensen-Shannon is a measure of the similarity of two distributions. Jensen Shannon is bounded from $[0, log_2(|c|+1)]$; we divide by the upper bound to yield a number from $[0, 1]$. Finally, Jensen Shannon is a measure of *divergence*, meaning that a lower score means that the distributions are more aligned. Thus, we subtract the Jensen Shannon divergence from 1 in order to stay consistent with a consensus measure, where greater consensus means a better topic.

5.2 Topic Consensus Results

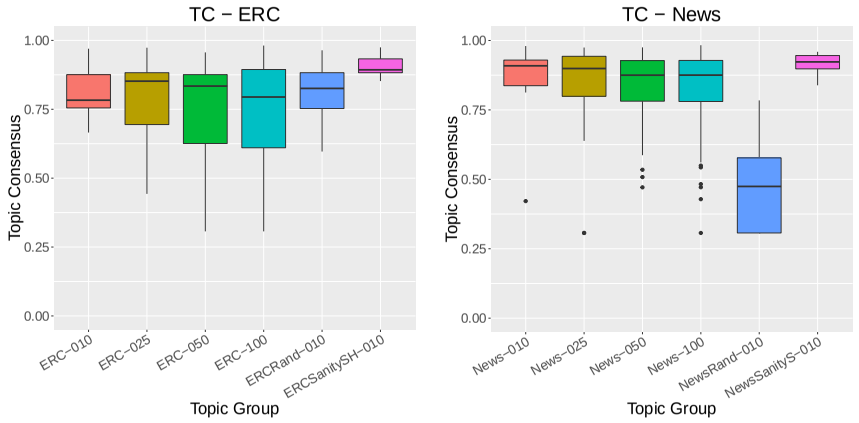
The results of the topic consensus experiment are shown in Figure 8. The results of both datasets indicate that the random topics perform worse than and the sanity topics perform better than the LDA topics. The results make sense: Sanity topics should have the highest

Table 6: Confusion matrix of ground truth ERC category assignments of topics against the category assignments made by the Turkers, taken from the outer product of the respective probability distributions. Rows are from the turkers, and columns are from LDA. In the ERC topics, we see that the Turkers are generally able to identify SH and LS topics, but overall fail to identify PE topics. The Turkers perform well when shown random topics, giving most of these topics a "not applicable" label.

	ERC-*				**ERCRand-010**				**ERCSanitySH-010**			
	Ground Truth											
	LS	SH	PE	NA	LS	SH	PE	NA	LS	SH	PE	NA
AMT Classification												
LS	29.69	5.21	13.39	0	0	0	0	0.59	0	0.09	0	0
SH	7.96	8.12	10.36	0	0	0	0	3.09	0	9.66	0	0
PE	16.62	8.14	36.21	0	0	0	0	0.61	0	0.11	0	0
NA	16.41	9.38	23.48	0	0	0	0	5.71	0	0.14	0	0
	News-*				**NewsRand-010**				**NewsSanityS-010**			
	Ground Truth											
	S	B	E	NA	S	B	E	NA	S	B	E	NA
AMT Classification												
S	35.31	7.74	8.36	0.00	0	0	0	4.26	9.84	0	0	0
B	9.88	50.00	8.61	0.00	0	0	0	1.96	0.04	0	0	0
E	10.00	7.12	21.67	0.00	0	0	0	2.60	0.08	0	0	0
NA	9.10	9.81	7.40	0.00	0	0	0	1.17	0.04	0	0	0

consensus and random topics the lowest consensus, and the performance of the other three topic models are in between.

To further investigate these answers we show a "confusion matrix" that compares the Turkers' responses with the ground truth in Table 6. Each cell in the matrix is the aggregation of the Turker's responses with the max of all the Turkers taken as the result. This is obtained from the sum of the outer product of the probability distributions from both the turkers (rows), and LDA (columns). The results for ERC-* and News-* topic sets show that most of the topics are understandable. In the case of the "PE" class in the ERC-* distribution, $\frac{52}{52+60+8+2} = 43\%$ of the topics are misclassified as NA. This could be because of the highly technical language in the physical sciences topics which causes users to select "NA". On the other hand the "SH" topics in the ERC corpus and the "B" topics in the News corpus are perfectly, and nearly-perfectly understood respectively. The sanity topics in both corpora achieve perfect scores. Finally, the random topics are vastly different between the two datasets. The random topics in the ERC corpus achieved 80% accuracy, however in the News domain the accuracy is 10%, with most of the misinterpretations leading the Turkers to categorize random topics as sports. We conjecture that the misclassification of sports may be a byproduct of the specific language of sports documents. For example, once a Turker sees a word like "football" in a random topic, they may be inclined to think that the topic is a sports topic.



(a) ERC (b) News

Figure 8: Topic consensus scores across all models across both corpora. Higher scores are better. On the left we see the form that is shown to the workers. On the right we see that the random topics perform worse than any of the ERC topics, and the SH topics perform the best.

5.3 Using Topic Interpretability Measures

We have employed three measures to identify interpretable topics from topic models thus far: Model Precision, Model Precision Choose Two, and Topic Consensus. At this point we will step back and discuss how to use these measures. There are two clear ways to use these measures. The first is for model selection, and the second is for topic selection.

Model selection is the process of choosing one model out of many (Kohavi et al., 1995). When selecting models for interpretability, we may choose the one that performs best. In the case of the News data, the model constructed with $K = 10$ performs best according to Topic Consensus, and the Model Precision Choose Two measures. Therefore, this model is the best one to choose from the perspective of interpretability.

The other strategy is topic selection. It may be that we are only using topic modeling to describe our dataset, and that we do not need the entire model in order to proceed. In this case, we can use the results of our topic interpretability measures to select a subset of topics that have high interpretability. This is favorable in cases where we need more topics. For example, in the case of Topic Consensus on the ERC data (Figure 8 (a)), we see that the median of $K = 25$ outperforms that of $K = 100$. However, the top quartile of $K = 25$ is worse than $K = 100$. This means that 25 of the 100 topics generated by the $K = 100$ model are, on average, better than the top 6 topics generated by the $K = 25$ model. Thus, if we are looking for topics which can help us to better understand our dataset, we may

decide to use topic selection with a larger K and use the interpretability measures proposed in this work to separate interpretable topics from less-interpretable ones.

6. Automating Measures of Topic Interpretability

In search of measures for topic interpretability, we resort to crowdsourcing approaches to measuring coherence and topic consensus. A natural question is whether we can replace crowdsourcing with automated measures. Crowdsourcing approaches could be costly, or not scalable without funds to recruit a sufficient numbers of Turkers. Furthermore, their results are hard to reproduce without good resources. In short, they present challenges in terms of scalability and reproducibility. The search for automated measures could replace Turkers and make empirical comparison available, saving researchers both time and cost of performing crowdsourced experiments. In this section, therefore, we investigate measures that can be used to automatically measure interpretable topics in terms of coherence and consensus. Mimno et al. (2011) are among researchers who first try to automate evaluation measures. They generated topics from medical literature and hired physicians, who are subject matter experts in their fields, to rate the quality of topics. They then proposed automated measures, Topic Size and Semantic Coherence, to replace these experts for reproducibility. Newman et al. (Newman et al., 2010) proposed a measure based on Pointwise Mutual Information. Aletras and Stevenson (2013) measured the quality of topics by inspecting the vector similarity between them. Morstatter et al. (2015) found that the peakedness of the distribution can be used to find interpretable topics. In the following, we introduce these measures, examine their correlations with the three measures of topic interpretability in this work, and then investigate if any of these automated measures can be used to accurately predict measures of topic coherence and topic consensus without the aid of the Turkers.

6.1 Automatic Topic Interpretability Measures

We introduce methods used for automatically assessing the quality of topics. Eight measures are given below:

i. **Topic Size (TS):** LDA soft assigns documents to topics by hard assigning the tokens within the document to a topic. At the end of training, each token in the corpus will have a topic assigned to it. "Topic Size" is the count of the number of tokens in the input corpus that are assigned to the topic after training. This was used in Mimno et al. (2011) as a possible measure for topic quality. The hypothesis behind this measure is that a larger topic (with more tokens) will represent more of the corpus, and thus will reveal a larger portion of the information within it.

ii. **Semantic Coherence (SC):** Also introduced by Mimno et al. (2011), this measures the probability of top words co-occuring within documents in the corpus:

$$SC(\mathbf{w}) = \sum_{j=2}^{|\mathbf{w}|} \sum_{k=1}^{j-1} log \frac{D(\mathbf{w}_j, \mathbf{w}_k) + 1}{D(\mathbf{w}_k)}, \tag{8}$$

where $\mathbf{w}$ is a vector of the top words in the topic sorted in descending order, and D is the number of documents containing all of the words provided as arguments. This measure is computed on the top 20 words of the topic.

iii. **Semantic Coherence Significance (SCS):** We adapt the **SC** measure above to understand the significance of the top words in the topic when compared to a random set of words. To calculate this measure we select 100 groups of words at random, following the topic's word distribution. We then recompute the Semantic Coherence measure for each of the random topics, obtaining a vector, $\mathbf{d}$, of topic coherence scores. We calculate the mean, $\bar{\mathbf{d}}$, and standard deviation, $std(\mathbf{d})$. Significance is defined as:

$$SCS(\mathbf{w}) = \frac{SC(\mathbf{w}) - \bar{\mathbf{d}}}{std(\mathbf{d})}. \tag{9}$$

iv. **Normalized Pointwise Mutual Information (NPMI):** Introduced by Bouma (2009), this metric measures the probability that two random variables coincide. This measure was used to estimate the performance of Model Precision in Lau et al. (2014), where the authors adapted it to measure the coincidence of the top $|\mathbf{w}|$ words. In this paper, two variables "coinciding" is the probability that they will co-occur in a document. The authors named this version OC-Auto-NPMI, formally:

$$\text{OC-Auto-NPMI}(\mathbf{w}) = \sum_{j=2}^{|\mathbf{w}|} \sum_{k=1}^{j-1} \frac{log \frac{P_D(\mathbf{w}_j, \mathbf{w}_k)}{P_D(\mathbf{w}_j) P_D(\mathbf{w}_k)}}{-log(P_D(\mathbf{w}_j, \mathbf{w}_k))}, \tag{10}$$

where $P_D(\cdot) = D(\cdot)/|N|$, where $|N|$ is the number of documents in the corpus. P_D measures the probability that a document in the corpus contains the words given to $D(\cdot)$. Going forward, we will refer to this measure as NPMI.

v. **Category-Distribution HHI (Cat-HHI):** The Herfindahl-Hirschman Index (Hirschman, 1945), or HHI, is a measure proposed to find the amount of competition in a market. This is calculated by measuring the market share of each firm in the market, formally:

$$HHI = \frac{(H - 1/N)}{(1 - 1/N)}, \tag{11}$$

Where $H = \sum_i s_i^2$, N is the number of firms in the market, and s_i is the market share of firm i, as a percentage. HHI ranges from 0 to 1, with 1 being a perfect monopoly (no competition), and 0 being an evenly split market. In this way we measure how focused the market is on a particular firm.

By treating the corpus categories as firms, and the market share distribution as $\mathbf{R}_i$, we can calculate how focused each topic is around a particular category.

vi. **Topic-Probability HHI (TP-HHI):** Using the same formulation as ERC-HHI, this time we treat every *word* in the vocabulary as a firm, and $\mathbf{T}_i^m$ as the probability distribution. In other words, the topic's probability distribution is the market, and TP-HHI measures the market's focus any word, or group of words. This measure varies from other significance measures, such as AlSumait et al. (2009), in that it focuses

purely on the peakedness of the distribution. This measures whether the focus of a topic is around a handful of words, or whether it is evenly spread across the entire vocabulary used to train the model.

vii. **No. of Word Senses:** The total number of word senses, according to WordNet, of the top five words in the topic. This varies slightly from the measure proposed in (Chang et al., 2009), where the authors also consider the intruded word. Because the intruded word is generally far away, we exclude it from our calculation.

viii. **Avg. Pairwise JCD:** The Jiang-Conrath (Jiang and Conrath, 1997) distance (JCD) is a measure of *semantic similarity*, or coherence, that considers the lowest common subsumer according to WordNet. Here we compute the average JCD of all $\binom{5}{2} = 10$ pairs of the top five words of the topic. This approach was introduced by (Chang et al., 2009), however we modify it slightly to only consider the topic's top five words.

ix. **Lesk Similarity:** The Lesk Algorithm (Banerjee and Pedersen, 2002) for word sense disambiguation uses WordNet to identify the most appropriate synset for an ambiguous word given a context. Following the use of this algorithm in (Newman et al., 2010), we adopt this technique in our automated measures. To turn the synset returned by this algorithm into a similarity measure, we evaluate the path similarity from the synset returned by this algorithm and all of the synsets in the context. The "context" is defined as the top 5 words in the topic, and the "ambiguous word" is the intruded word. When applying this to consensus, we use the same intruded word as in the MP and MPCT case.

6.2 Correlation with Crowdsourced Measures of Topic Interpretability

To see how well these automatic topic measures compare with the crowdsourced topic measures from the "Topic Interpretability" section, we calculate the Spearman's ρ (Spearman, 1904) rank correlation coefficient between the crowdsourced measure and the automatic measure. The correlations between each pair of crowdsourced and automatic measure are shown in Table 7. In this table, we present the Spearman's ρ, as well as an indication of the significance level for the following hypothesis test:

H_0 : The two sets of data are uncorrelated.

Instances where the hypothesis is rejected at the $\alpha = 0.05$ significance level are shown in Table 7. We see that the measure most correlated with Model Precision and Model Precision Choose Two (MPCT) is Avg. Pairwise JCD, meaning that documents whose words have a higher semantic similarity are more likely to achieve higher Model Precision (MP) values. We see that higher (better) values of Model Precision are accompanied by higher average JCD values. Table 7 shows the measure most correlated with Topic Consensus (TC) is TP-HHI. These correlations lead us to the next question: can we predict each of the three measures (MP, MPCT, and TC) of topic interpretability based on their correlations?

6.3 Predicting Crowdsourced Values of Topic Interpretability Measures

While correlation can be used to find the quality of a measure, we further ask if it is feasible to predict the *actual* value of the topics' crowdsourced measures. At first it may seem that

Table 7: Spearman's ρ between measures requiring Mechanical Turk and automated measures. Instances where we reject the null hypothesis at the significance level of $\alpha = 0.05$ are denoted with *, and instances where we reject the null hypothesis at the significance level of $\alpha = 10^{-4}$ are denoted with **.

	ERC-*			**News-***		
	MP	MPCT	TC	MP	MPCT	TC
TS	0.152*	-0.709**	0.532**	-0.585**	-0.645**	0.688**
SC	0.359**	-0.165*	0.584**	-0.443**	**-0.885****	0.163*
SCS	-0.074	-0.582**	0.788**	-0.337**	-0.410**	0.501**
NPMI	-0.562**	0.067	0.774**	0.189*	-0.203*	0.674**
Cat-HHI	0.103	-0.652**	0.478**	-0.223*	-0.416	0.588
TP-HHI	-0.471**	-0.057	**0.885****	-0.001	-0.318*	**0.913****
Senses	-0.111	-0.267*	0.854**	-0.055	0.022	0.674**
JCD	**1.000****	**0.750****	0.022	**0.685****	0.805**	0.349**
Lesk	0.050	-0.034	0.189*	0.533**	0.503**	0.442**

Table 8: Errors of the predictive algorithm trained on all of the automated measures. Results are presented as RMSE $\pm$ the standard deviation. These results indicate that Topic Consensus is the most predictable.

	Model Precision	Model Precision Choose Two	Topic Consensus
ERC-*	0.280 ± 0.039	0.352 ± 0.064	0.131 ± 0.021
News-*	0.239 ± 0.049	0.307 ± 0.075	0.114 ± 0.034

the most correlated measure may be the most predictive, however, due to nonlinear patterns within the data this may not be the case. In this section, we investigate if we can predict the true scores of topic interpretability by using the above eight automated measures.

We train a linear regression model to predict the true value of the crowdsourced measures using the automated measures. We build three models: one where the dependent variable is "Model Precision" (MP), another where it is "Model Precision Choose Two" (MPCT), and the other where the dependent variable is "Topic Coherence" (TC). In all three cases, the independent variables are all of the automated measures introduced previously. We use 10-fold cross validation and report both the mean and standard deviation of the performance of the models across all 10 runs in Table 8. These results indicate that it is easiest to predict the raw scores of Topic Consensus, and that it is the hardest to predict those of Model Precision Choose Two. An advantage of this prediction approach is that it allows for inclusion of new automated measures in order to increase the predictive capability of the prediction model.

7. Conclusion and Future Work

Statistical topic models are a key component for machine learning, NLP, and the social sciences. In this work we investigate different measures for the interpretability of the topics generated by these approaches. We view topic interpretability from two perspectives: *topic coherence* and *topic consensus*. Coherence measures how semantically close the top words of a topic are. Consensus measures the agreement between the mixture of labels assigned by the humans and the mixture of labels from the documents assigned to the topic. We investigate what is needed for comprehensive measure of each perspective, understand how different these measures for topic coherence and consensus are, and show their experimental results using real-world datasets.

For topic coherence, we study Model Precision (MP) and propose Model Precision Choose Two (MPCT). The two measures complement each other to better measure the semantic closeness of topic words. MP works by making sure that the topic words should be far away from the intruded word. Additionally, MPCT complements MP by addressing the closeness of the 5 words within the topic. We compare MPCT with MP and show how these two measures can work in tandem to identify interpretable topics. One natural question that can arise from the MPCT setup is why only two words are requested, when in fact any number of words could be asked for. While this is theoretically attractive, the practical implications for the Turkers can present a challenge. By asking them to select one intruder and one additional word, we present a small, viable workload for the Turkers. Finding more strategies to effectively measure coherence is an area for future work.

For topic consensus, we assess how well the workers' aggregate understanding of the topic matches the aggregate of the categories provided by the corpus. In both corpora, each document is tagged with a category. When we train a topic model, each topic will ultimately be a mixture of these categories. This method reveals how well the topics generated can be understood in relation to the underlying categories in the corpus. One natural application of this measure is to identify chimera topics, *i.e.* those topics that are split between two concepts or categories. Furthermore, by inspecting the Turker's results on known topics, we can see the bounds of this measure on extremely good, and extremely bad topics.

All three measures of topic interpretability rely on crowdsourcing platforms, which motivates a need to automate them in order to improve reproducibility and scalability. Recreating these results with human workers requires significant investments of time to recruit the workers, as well as funds to pay them. We investigate how to estimate these crowdsourced topic measures without the need of crowdsourcing tools such as Mechanical Turk. We find some automated measures that are highly correlated with crowdsourced measures (MP, MPCT, and TC), allowing researchers to reproduce these topic quality measures at scale. In addition, we propose to construct a prediction model for predicting MP, MPCT, and TC based on the automated measures. This prediction model can take advantage of new automated measures to improve their predictive power.

Topic interpretability is a challenge to the problem of topic quality assessment. This work makes extensive efforts to solidify its measurement via topic coherence and topic consensus and demonstrates that both coherence and consensus can help understand topic interpretability. More research remains to be done. One direction for future work is to catalogue the performance of all crowdsourced measures across more and different datasets

to provide a benchmark for different types of data. This is because different types of data intuitively require corresponding crowdsourced measures. Model Precision Choose Two was designed to measure an additional dimension of topic interpretability, however increasing the number of words the turker should choose may lead to more dimensions of interpretability. Future work also includes investigating different models, e.g., those that take K out of the equation, as is the case with nonparametric topic models such as Hierarchical Dirichlet Processes (Teh et al., 2006). New NLP models offer new semantic properties, such as word embeddings which attempt to embed words with similar semantic meanings closer together. Designing measures for these properties is an important area for future work. Additionally, there is more work to be done in how the setup changes the output of these measures. For example, in the "topic consensus" results, we noted that words like "football" may draw a Turker to think that a topic is a sports topic more strongly than words like "field", or "goal". Challenges such as this provide exciting research directions in this area.

References

Nikolaos Aletras and Mark Stevenson. Evaluating Topic Coherence using Distributional Semantics. In *IWCS*, pages 13–22, 2013.

Nikolaos Aletras and Mark Stevenson. Labelling topics using unsupervised graph-based methods. In *ACL*, pages 631–636, 2014.

Loulwah AlSumait, Daniel Barbará, James Gentle, and Carlotta Domeniconi. Topic significance ranking of lda generative models. *Machine Learning and Knowledge Discovery in Databases*, pages 67–82, 2009.

Arthur Asuncion, Max Welling, Padhraic Smyth, and Yee Whye Teh. On smoothing and inference for topic models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 27–34, 2009.

Satanjeev Banerjee and Ted Pedersen. An adapted lesk algorithm for word sense disambiguation using wordnet. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 136–145, 2002.

H Russell Bernard and Gery W Ryan. *Analyzing qualitative data: Systematic approaches*. SAGE publications, 2009.

D M Blei, A Y Ng, and M I Jordan. Latent Dirichlet Allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

David Blei and John Lafferty. Correlated topic models. *Advances in neural information processing systems*, 18:147, 2006.

David M Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

Gerlof Bouma. Normalized (Pointwise) Mutual Information in Collocation Extraction. *Proceedings of GSCL*, pages 31–40, 2009.

Jonathan Chang, Jordan L. Boyd-Graber, Sean Gerrish, Chong Wang, and David M Blei. Reading Tea Leaves: How Humans Interpret Topic Models. In *NIPS*, volume 22, pages 288–296, 2009.

Jacob Eisenstein, Brendan O'Connor, Noah A. Smith, and Eric P. Xing. A Latent Variable Model for Geographic Lexical Variation. In *EMNLP*, pages 1277–1287, 2010. URL `http://dl.acm.org/citation.cfm?id=1870658.1870782`.

William B Frakes and Ricardo Baeza-Yates. Information retrieval: data structures and algorithms. 1992.

Michael Frigge, David C Hoaglin, and Boris Iglewicz. Some implementations of the boxplot. *The American Statistician*, 43(1):50–54, 1989.

Thomas L Griffiths and Mark Steyvers. Finding Scientific Topics. *PNAS*, 101(Suppl 1): 5228–5235, 2004.

Justin Grimmer and Brandon M Stewart. Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis*, page mps028, 2013.

A. O. Hirschman. *National Power and the Structure of Foreign Trade*. University of California Press, Berkeley, CA, 1945.

Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, 1999.

Liangjie Hong, Amr Ahmed, Siva Gurumurthy, Alexander J. Smola, and Kostas Tsioutsiouliklis. Discovering Geographical Topics in the Twitter Stream. In *WWW*, pages 769–778, 2012. ISBN 978-1-4503-1229-5. doi: 10.1145/2187836.2187940. URL `http://doi.acm.org/10.1145/2187836.2187940`.

Yuening Hu, Jordan Boyd-Graber, Brianna Satinoff, and Alison Smith. Interactive topic modeling. *Machine learning*, 95(3):423–469, 2014.

Yuheng Hu, Ajita John, Fei Wang, and Subbarao Kambhampati. ET-LDA: Joint Topic Modeling for Aligning Events and their Twitter Feedback. In *AAAI*, volume 12, pages 59–65, 2012.

Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. Perplexitya measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63, 1977.

Jay J Jiang and David W Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*, 1997.

Kenneth Joseph, Chun How Tan, and Kathleen M. Carley. Beyond "Local", "Categories" and "Friends": Clustering foursquare Users with Latent "Topics". In *UbiComp*, pages 919–926, 2012. ISBN 978-1-4503-1224-0. doi: 10.1145/2370216.2370422. URL `http://doi.acm.org/10.1145/2370216.2370422`.

James M Joyce. Kullback-Leibler Divergence. In *International Encyclopedia of Statistical Science*, pages 720–722. Springer, 2011.

Daniel Jurafsky and James H Martin. Speech and language processing. *Pearson*, 2000.

Noriaki Kawamae. N-gram over context. In *Proceedings of the 25th International Conference on World Wide Web*, pages 1045–1055, 2016.

Kirill Kireyev, Leysia Palen, and Kenneth M. Anderson. Applications of Topics Models to Analysis of Disaster-Related Twitter Data. In *NIPS Workshop on Applications for Topic Models: Text and Beyond*, volume 1, 2009.

Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, volume 14, pages 1137–1145, 1995.

Shamanth Kumar, Fred Morstatter, Reza Zafarani, and Huan Liu. Whom should i follow?: identifying relevant users during crises. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, HT '13, pages 139–147, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1967-6. doi: 10.1145/2481492.2481507. URL `http://doi.acm.org/10.1145/2481492.2481507`.

Jey Han Lau, Karl Grieser, David Newman, and Timothy Baldwin. Automatic labelling of topic models. In *ACL*, pages 1536–1545, 2011.

Jey Han Lau, David Newman, and Timothy Baldwin. Machine Reading Tea Leaves: Automatically Evaluating Topic Coherence and Topic Model Quality. In *Proceedings of the European Chapter of the Association for Computational Linguistics*, pages 530–539, 2014.

Tuan Le and Hady W Lauw. Semantic visualization with neighborhood graph regularization. *Journal of Artificial Intelligence Research*, 55:1091–1133, 2016.

Wei Li and Andrew McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *Proceedings of the 23rd international conference on Machine learning*, pages 577–584. ACM, 2006.

Jianhua Lin. Divergence Measures Based on the Shannon Entropy. *Information Theory, IEEE Transactions on*, 37(1):145–151, Jan 1991. ISSN 0018-9448. doi: 10.1109/18.61115.

Tianyi Lin, Wentao Tian, Qiaozhu Mei, and Hong Cheng. The dual-sparse topic model: mining focused topics and focused terms in short text. In *Proceedings of the 23rd international conference on World wide web*, pages 539–550, 2014.

Qiang Liu, Alexander T Ihler, and Mark Steyvers. Scoring workers in crowdsourcing: How many control questions are enough? In *Advances in Neural Information Processing Systems*, pages 1914–1922, 2013.

Jalal Mahmud, Jeffrey Nichols, and Clemens Drews. Where Is This Tweet From? Inferring Home Locations of Twitter Users. In *ICWSM*, pages 511–514, 2012.

Arun S Maiya, John P Thompson, Francisco Loaiza-Lemos, and Robert M Rolfe. Exploratory analysis of highly heterogeneous document collections. In *KDD*, pages 1375–1383, 2013.

Xian-Ling Mao, Zhao-Yan Ming, Zheng-Jun Zha, Tat-Seng Chua, Hongfei Yan, and Xiaoming Li. Automatic labeling hierarchical topics. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2383–2386, 2012.

K Mayer and J Pfeffer. Mapping Social Sciences and Humanities. *Horizons for Social Sciences and Humanities*, 09 2014.

Andrew Kachites McCallum. MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu, 2002.

David Mimno, Hanna M. Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *EMNLP*, pages 262–272, 2011. ISBN 978-1-937284-11-4. URL `http://dl.acm.org/citation.cfm?id=2145432.2145462`.

Fred Morstatter, Jürgen Pfeffer, Huan Liu, and Kathleen M. Carley. Is the Sample Good Enough? Comparing Data from Twitter's Streaming API with Twitter's Firehose. In *ICWSM*, pages 400–408, 2013.

Fred Morstatter, Nichola Lubold, Heather Pon-Barry, J"urgen Pfeffer, and Huan Liu. Finding Eyewitness Tweets During Crises. In *ACL 2014 Workshop on Language Technologies and Computational Social Science*, 2014.

Fred Morstatter, Jürgen Pfeffer, Katja Mayer, and Huan Liu. Text, topics, and turkers: A consensus measure for statistical topics. In *Hypertext & Social Media*, pages 123–131. ACM, 2015.

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *ACL*, pages 100–108, 2010.

Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

Alexei Pozdnoukhov and Christian Kaiser. Space-time dynamics of topics in streaming text. In *Proc. of the 3rd ACM SIGSPATIAL Int'l Workshop on Location-Based Social Networks*, LBSN '11, pages 1–8, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-1033-8. doi: 10.1145/2063212.2063223. URL `http://doi.acm.org/10.1145/2063212.2063223`.

Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D Manning. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *EMNLP*, pages 248–256, 2009a.

Daniel Ramage, Evan Rosen, Jason Chuang, Christopher D Manning, and Daniel A McFarland. Topic modeling for the social sciences. In *NIPS 2009 Workshop on Applications for Topic Models: Text and Beyond*, volume 5, 2009b.

Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the Space of Topic Coherence Measures. In *WSDM*, pages 399–408, 2015.

Benjamin M Schmidt. Words Alone: Dismantling Topic Models in the Humanities. *Journal of Digital Humanities*, 2(1):49–65, 2012.

Carson Sievert and Kenneth E Shirley. LDAvis: A method for visualizing and interpreting topics. In *Proceedings of the workshop on interactive language learning, visualization, and interfaces*, pages 63–70, 2014.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. Cheap and Fast—But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks. In *EMNLP*, pages 254–263, 2008.

Charles Spearman. The Proof and Measurement of Association between Two Things. *The American Journal of Psychology*, 15(1):72–101, 1904.

Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101(476), 2006.

A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. In *ICWSM*, pages 178–185, 2010.

Hadley Wickham and Lisa Stryjewski. 40 years of boxplots. *American Statistician*, 2011.

Zhijun Yin, Liangliang Cao, Jiawei Han, Chengxiang Zhai, and Thomas Huang. Geographical Topic Discovery and Comparison. In *WWW*, pages 247–256, 2011.

Dengyong Zhou, Qiang Liu, John C Platt, and Christopher Meek. Aggregating ordinal labels from crowds by minimax conditional entropy. In *ICML*, volume 14, pages 262–270, 2014.

Journal of Machine Learning Research 18 (2018) 1-60 Submitted 3/17; Revised 2/18; Published 4/18

On the Behavior of Intrinsically High-Dimensional Spaces: Distances, Direct and Reverse Nearest Neighbors, and Hubness

Fabrizio Angiulli fabrizio.angiulli@unical.it
DIMES – Dept. of Computer, Modeling, Electronics, and Systems Engineering
University of Calabria
87036 Rende (CS), Italy

Editor: Sanjiv Kumar

Abstract

Over the years, different characterizations of the curse of dimensionality have been provided, usually stating the conditions under which, in the limit of the infinite dimensionality, distances become indistinguishable. However, these characterizations almost never address the form of associated distributions in the finite, although high-dimensional, case. This work aims to contribute in this respect by investigating the distribution of distances, and of direct and reverse nearest neighbors, in intrinsically high-dimensional spaces. Indeed, we derive a closed form for the distribution of distances from a given point, for the expected distance from a given point to its kth nearest neighbor, and for the expected size of the approximate set of neighbors of a given point in finite high-dimensional spaces. Additionally, the hubness problem is considered, which is related to the form of the function N_k representing the number of points that have a given point as one of their k nearest neighbors, which is also called the number of k-occurrences. Despite the extensive use of this function, the precise characterization of its form is a longstanding problem. We derive a closed form for the number of k-occurrences associated with a given point in finite high-dimensional spaces, together with the associated limiting probability distribution. By investigating the relationships with the hubness phenomenon emerging in network science, we find that the distribution of node (in-)degrees of some real-life, large-scale networks has connections with the distribution of k-occurrences described herein.

Keywords: high-dimensional data, distance concentration, distribution of distances, nearest neighbors, reverse nearest neighbors, hubness

1. Introduction

Although the size and the dimensionality of collected data are steadily growing, traditional techniques usually slow down exponentially with the number of attributes to be considered and are often overcome by linear scans of the whole data. In particular, the term *curse of dimensionality* (Bellmann, 1961), is used to refer to difficulties arising whenever high-dimensional data has to be taken into account.

One of the main aspects of this curse is known as *distance concentration* (Demartines, 1994), which is the tendency for distances to become almost indiscernible in high-dimensional spaces. This phenomenon may greatly affect the quality and performances of machine learning, data mining, and information-retrieval techniques. This effect results because almost all these techniques rely on the concept of distance, or dissimilarity, among data items in order to retrieve or analyze information. However, whereas low-dimensional spaces show good agreement between geometric proximity and the notion of similarity, as dimensionality increases, different counterintuitive phenomena arise that may be harmful to traditional techniques.

Over time, different characterizations of the curse of dimensionality and related phenomena have been provided (Demartines, 1994; Beyer et al., 1999; Aggarwal et al., 2001; Hinneburg et al., 2000; François et al., 2007). These characterizations usually state conditions under which, according to the limits of infinite dimensionality, distances become indistinguishable. However, almost never do these conditions address the form of associated distributions in finite, albeit high-dimensional, cases.

This work aims to contribute in this area by investigating the distribution of distances and of some related measures in intrinsically high-dimensional data. In particular, the analysis is conducted by applying the central limit theorem to the Euclidean distance random variable to approximate the distance probability distribution between pairs of random vectors, between a random vector and realizations of a random vector, and to obtain the expected distance from a given point to its kth nearest neighbor. It is then shown that an understanding of these distributions can be exploited to gain knowledge of the behavior of high-dimensional spaces, specifically the number of approximate nearest neighbors and the number of reverse nearest neighbors that are also investigated herein.

Nearest neighbors are transversal to many disciplines (Preparata and Shamos, 1985; Dasarathy, 1990; Beyer et al., 1999; Duda et al., 2000; Chávez et al., 2001; Shakhnarovich et al., 2006). In order to try to overcome the difficulty of answering nearest neighbor queries in high-dimensional spaces (Weber et al., 1998; Beyer et al., 1999; Pestov, 2000; Giannella, 2009; Kabán, 2012), the concept of the ϵ-approximate nearest neighbor (Indyk and Motwani, 1998; Arya et al., 1998) has been introduced. The ϵ-neighborhood of a query point is the set of points located at a distance not greater than $(1+\epsilon)$ times the distance separating the query from its true nearest neighbor.

Related to the notion of the ϵ-approximate nearest neighbor is the notion of neighborhood or query instability (Beyer et al., 1999): a query is said to be unstable if the ϵ-neighborhood of the query point consists of most of the data points. Although asymptotic results, such as that reported by Beyer et al. (1999), tell what happens when dimensionality is taken to infinity, nothing is said about the dimensionality at which the nearest neighbors become unstable. Pursuant to this scenario, this paper derives a closed form for the expected size of the ϵ-neighborhood in finite high-dimensional spaces, an expression that is then exploited to determine the critical dimensionality. Also, to quantify the difficulty of (approximate) nearest neighbor search, He et al. (2012) introduced the concept of relative contrast, a measure of separability of the nearest neighbor of the query point from the rest of the data, and provided an estimate which is applicable for finite dimensions. By leveraging the results concerning distance distributions, this paper derives a more accurate estimate for the relative contrast measure.

The number N_k of reverse nearest neighbors, also called the number of k-occurrences or the reverse nearest neighbor count, is the number of data points for which a given point is among their k nearest neighbors. Reverse nearest neighbors are of interest both in the database, information retrieval, and computational geometry literatures (Korn and

Muthukrishnan, 2000; Singh et al., 2003; Tao et al., 2007; Cheong et al., 2011; Yang et al., 2015), with uses having been proposed in the data mining and machine learning fields (Williams et al., 2002; Hautamäki et al., 2004; Radovanovic et al., 2009, 2010; Tomasev et al., 2014; Radovanovic et al., 2015; Tomasev and Buza, 2015), beyond being the objects of study in applied probability and mathematical psychology (Newman et al., 1983; Maloney, 1983; Tversky et al., 1983; Newman and Rinott, 1985; Yao and Simons, 1996).

Despite the usefulness and the extensive use of this construct, the precise characterization of the form of the function N_k both in the finite and infinite dimensional cases is a longstanding problem. What is already known is that for the infinite limit of size and dimension, N_k must converge in its distribution to zero; however, this result and its interpretations seem to be insufficient to characterize its observed behavior in finite samples and dimensions. Consequently, this paper derives a closed form of the number of k-occurrences associated with a given point in finite high-dimensional spaces, together with a generalized closed form of the associated limiting probability distribution that encompasses previous results and provides interpretability of its behavior and of the related hubness phenomenon.

The results, which are first illustrated for independent and identically distributed data, are subsequently extended to independent non-identically distributed data satisfying certain conditions, and then, connections with non-independent real data are depicted. Finally, it is discussed how to potentially extend the approach to Minkowski's metrics and, more generally, to distances satisfying certain conditions of spatial centrality.

Because hubness is a phenomenon of primary importance in network science, we also investigate if the findings relative to the distribution of the reverse nearest neighbors and the emergence of hubs in intrinsically high-dimensional contexts is connected to an analogous phenomenon occurring in the context of networks. The investigation reveals that for some real-life large-scale networks, the distribution of the incoming node degrees is connected to the herein-derived distribution of the infinite-dimensional k-occurrences function, which models the number of reverse k-nearest neighbors in an arbitrarily large feature space of independent dimensions. Hence, the provided distribution also appears to be suitable for modelling node-degree distributions in complex real networks.

The current study can be leveraged in several ways and in different contexts, such as in direct and reverse nearest neighbor searches, density estimation, anomaly and novelty detection, density-based clustering, and network analysis, among others. With regard to its possible applications, we can highlight approximations of measures related to distance distributions, worst-case scenarios for data analysis and retrieval techniques, design strategies that try to mitigate the curse of dimensionality, and models of complex networks. We refer to the concluding section for a more extensive discussion.

The rest of the work is organized as follows. Section 2 discusses related work concerning the concentration of distances, intrinsic dimensionality, and the number of k-occurrences and the associated hubness phenomenon. Section 3 presents the notation used to provide results. Section 4 introduces the main results of the paper. Section 5 discusses relationships between the study of the hubness phenomena occurring in high-dimensional spaces with the analogous phenomena observed in real-life, large-scale, complex networks. Section 6 concludes the work. Finally, the Appendix contains the proofs that are not reported within the main text.

2. Related Work

As already noted, the term *curse of dimensionality* is used to refer to difficulties arising when high-dimensional data must be taken into account, and one of the main aspects of this curse is *distance concentration*. In this regard, Demartines (1994) has shown that the expectation of the Euclidean norm of independent and identically distributed (i.i.d.) random vectors increases as the square root of the dimension, whereas its variance tends toward a constant and, hence, does not depend on the dimensionality. Specifically:

Theorem 1 (Demartines, 1994, cf. Theorem 2.1) *Let $\boldsymbol{X}_d$ be an i.i.d. d-dimensional random vector with common cdf F_X. Then*

$$\mathbf{E}[\|\boldsymbol{X}_d\|_2] = \sqrt{ad-b} + O(1/d) \quad \text{and} \quad \sigma^2(\|\boldsymbol{X}_d\|_2) = b + O(1/\sqrt{d}),$$

where a and b are constants depending on the central moments of F_X up to the fourth order but do not depend on the dimensionality d.

Demartines noticed that, because the Euclidean distance corresponds to the norm of the difference of two vectors, the distance between the i.i.d. random vectors must also exhibit the same behavior. This insightful result explains why high-dimensional vectors appear to be distributed around the surface of a sphere of radius $\mathbf{E}[\|\boldsymbol{X}_d\|]$ and why, because they seem to be normalized, the distances between pairs of high-dimensional random vectors tend to be similar.

The distance concentration phenomenon is usually characterized in the literature by means of a ratio between some measure related to the spread and some measure related to the magnitude of the norm, sometimes presented as the distance from a point located in the origin of the space. In particular, the conclusion is that there is a concentration of distances when the above ratio converges to 0 as the dimensionality tends to infinity.

Some authors have studied the concentration phenomenon by representing a data set as a set of n d-dimensional i.i.d. random vectors $\boldsymbol{X}_d^{(j)}$ $(1 \leq j \leq n)$ with not-necessarily common pdfs $f_{X^{(j)}}$. Specifically, the *contrast* is defined as the difference between the largest and the smallest observed norm, or rather the distance from a query point located at the origin, whereas the *relative contrast* is defined as

$$RC_d = \frac{\max_j \|\boldsymbol{X}_d^{(j)}\|_p - \min_j \|\boldsymbol{X}_d^{(j)}\|_p}{\min_j \|\boldsymbol{X}_d^{(j)}\|_p},$$

where $\|\cdot\|_p$ denotes the p-norm $\|\boldsymbol{x}_d\|_p = \left(\sum_{i=1}^d |x_i|^p\right)^{1/p}$, is the contrast normalized with respect to the smallest norm/distance.

Theorem 2 (Adapted from Beyer et al., 1999, cf. Theorem 1) *Let $\boldsymbol{X}_d^{(j)}$ $(1 \leq j \leq n)$ be n d-dimensional random vectors with common cdfs. If*

$$\lim_{d\to\infty} \sigma^2\left(\frac{\|\boldsymbol{X}_d^{(j)}\|_p}{\mathbf{E}[\|\boldsymbol{X}_d^{(j)}\|_p]}\right) = 0, \text{ then, for any } \epsilon > 0, \lim_{d\to\infty} Pr\left[RC_d \leq \epsilon\right] = 1.$$

If the hypothesis is verified, that is, if the variance of the ratio between the norm of the vectors and their expected value vanishes as the dimensionality goes to infinity, then the relative contrast also becomes smaller and smaller, and all the vectors seem to be located at approximatively the same distance from the reference vector. That is, given a query point $\boldsymbol{Q}_d$, the distance from the nearest and the furthest neighbor become negligible:

$$\lim_{d\to\infty} Pr\left[\max_j \|\boldsymbol{Q}_d - \boldsymbol{X}_d^{(j)}\|_p \leq (1+\epsilon)\min_j \|\boldsymbol{Q}_d - \boldsymbol{X}_d^{(j)}\|_p\right] = 1.$$

In (Beyer et al., 1999), it is shown that i.i.d. random vectors satisfy the above condition.

Other authors have provided characterizations of the concentration phenomenon by providing upper and lower bounds to the relative contrast in the cases of Minkowski and fractional norms (Hinneburg et al., 2000; Aggarwal et al., 2001).

Subsequently, (François et al., 2007) posed the following problem: is the concentration phenomenon a side effect of the *Empty Space Phenomenon* (Bellmann, 1961), just because we consider a finite number of points in a bounded portion of a high-dimensional space? To explore this problem, they studied the concentration phenomenon by taking the same perspective as Demartines, i.e., to refer to a distribution rather than to a finite set of points. The *relative variance*

$$RV_d = \frac{\sigma(\|\boldsymbol{X}_d\|_p)}{\mathbf{E}[\|\boldsymbol{X}_d\|_p]}$$

is a measure of concentration for distributions, corresponding to the ratio between the standard deviation and the expected value of the norm.

Theorem 3 (Adapted from François et al., 2007, cf. Theorem 5) *Let $\boldsymbol{X}_d$ be an i.i.d. d-dimensional random vector. Then*

$$\lim_{d\to\infty} RV_d = 0.$$

From the above result, they conclude that the concentration of the norms in high-dimensional spaces is an intrinsic property of the norms and not a side effect of the finite sample size or of the Empty Space Phenomenon. Because it does not depend on the sample size, this can be regarded as an extension of Demartines' results to all p-norms.

As a consequence of the distance concentration, the separation between the nearest neighbor and the farthest neighbor of a given point tend to become increasingly indistinct as the dimensionality increases.

Related to the analysis of i.i.d. data is the concept of intrinsic dimensionality. Although variables used to identify each datum could not be statistically independent, ultimately, the intrinsic dimensionality of the data is identified as the minimum number of variables needed to represent the data itself (van der Maaten et al., 2009). This corresponds in linear spaces to the number of linearly independent vectors needed to describe each point. As a matter of fact, an extensively used notion of intrinsic dimensionality, the *correlation dimension* (Grassberger and Procaccia, 1983), is based on identifying the dimensionality D at which the Euclidean space is homeomorphic to the manifold containing the support of the data:

$$D = \lim_{\delta\to 0} \frac{\ln F_{\text{dst}}(\delta)}{\ln \delta},$$

where F_{dst} denotes the cumulative distribution function of pairwise distances, which formalizes the notion that in the limit of small length-scales ($\delta \to 0$) upon which the manifold the data lie, the manifold is homeomorphic to the Euclidean space of dimension D.

And, indeed, (Demartines, 1994) mentions that if random vector components are not independent, the concentration phenomenon is still present provided that the actual number D of "degrees of freedom" is sufficiently large. Thus, results derived for i.i.d. data continue to be valid provided that the dimensionality d is replaced with D. Moreover, (Beyer et al., 1999) provided different examples of data presenting concentration, all of which share with the i.i.d. case a sparse correlation structure. (Durrant and Kabán, 2009) noted that it is difficult to identify meaningful workloads that do not exhibit concentration, and showed that for the family of linear latent variable models, a class of data distributions having non-i.i.d. dimensions, the Euclidean distance will not become concentrated as long as the number of relevant dimensions grows no more slowly than the overall data dimensions do. This also confirms that weakly dependent data lead to concentration; however, they also noted that the condition to avoid concentration is not often met in practice.

Another aspect of the curse of dimensionality problem, closely related to the distance concentration and the nearest neighbor relationship, is the so called *hubness* phenomenon. This phenomenon has been previously observed in several applications (Doddington et al., 1998; Singh et al., 2003; Aucouturier and Pachet, 2007), has recently undergone to direct investigation (Radovanovic et al., 2009, 2010; Low et al., 2013), and has been subjected to several different proposed techniques for overcoming the phenomenon (Radovanovic et al., 2015; Tomasev, 2015).

Specifically, consider the number $\mathrm{N}_k(\boldsymbol{x}_d)$ of observed points that have $\boldsymbol{x}_d$ among their k nearest neighbors. N_k is also called *k-occurrences* or the *reverse k-nearest neighbor count*.

It is known that in low dimensional spaces, the distribution of N_k complies with the binomial one and, in particular, for uniformly distributed data in low dimensions, it can be modeled as node in-degrees in the k-nearest neighbor graph, which follows the Erdős-Rényi random graph model, with a binomial degree distribution (Erdős and Rényi, 1959). However, it has been observed that as dimensionality increases, the distribution of N_k becomes skewed to the right, resulting in the emergence of *hubs*, which are points whose reverse k-nearest neighbor counts tend to be meaningfully larger than that associated with any other point.

The distribution of N_k has been explicitly studied in the applied probability and mathematical psychology communities (Newman et al., 1983; Maloney, 1983; Newman and Rinott, 1985; Tversky and Hutchinson, 1986; Yao and Simons, 1996). Almost all the results provided concern a Poisson process that spreads the vectors uniformly over $\mathbb{R}^d$, leading to the conclusion that the limiting distribution of N_k converges to the Poisson distribution with mean k. The case of continuous distributions with i.i.d. components has been considered in (Newman et al., 1983; Newman and Rinott, 1985), where the expression for the infinite-dimensional distribution of N_1 is characterized as follows.

Theorem 4 (Newman et al., 1983, cf. Theorem 7) *Let $\{\boldsymbol{X}_d^{(0)}, \boldsymbol{X}_d^{(1)}, \ldots, \boldsymbol{X}_d^{(n)}\}$ be i.i.d. random vectors with a common continuous cdf having a finite fourth moment. Let $N_1^{n,d}$ denote the number of elements from $\{\boldsymbol{X}_d^{(1)}, \ldots, \boldsymbol{X}_d^{(n)}\}$ whose nearest neighbor with*

respect to the Euclidean distance is $\boldsymbol{X}_d^{(0)}$. *Then*

$$\lim_{n\to\infty}\lim_{d\to\infty} \mathrm{N}_1^{n,d} \xrightarrow{D} 0 \qquad \textit{and} \qquad \lim_{n\to\infty}\lim_{d\to\infty} \sigma^2(\mathrm{N}_1^{n,d}) = \infty.$$

The interpretation of the above result due to (Tversky et al., 1983) is that if the number of dimensions is large relative to the number of points, a large portion of points will have reverse nearest neighbor count equaling zero, whereas a small fraction (i.e., the hubs) will score large counts.

In order to provide an explanation for hubness, (Radovanovic et al., 2010) noticed that it is expected for points that are closer to the mean of the data distribution to be closer, on average, to all other points. However, empirical evidence indicates that this tendency is amplified by high-dimensionality, making points that reside in the proximity of the datas mean become closer to all other points than their low-dimensional analogues are. This tendency causes high-dimensional points that are closer to the mean to have increased probability of being selected as k-nearest neighbors by other points, even for small values of k.

In order to formalize the above evidence in finite-dimensional spaces, the authors considered the simplified setting of normally distributed i.i.d. d-dimensional random vectors, for which the distribution of Euclidean distances, which are calculated as the square root of the sum of squares of i.i.d. normal variables, corresponds to a chi distribution with d degrees of freedom (Johnson et al., 1994) and the random variable $\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|$, representing the distance from a fixed point $\boldsymbol{x}_d$ to the rest of the data, follows the noncentral chi distribution with d degrees of freedom and noncentrality parameter $\lambda = \|\boldsymbol{x}_d\|$ (Oberto and Pennecchi, 2006).

Theorem 5 (Radovanovic et al., 2010, cf. Theorem 1) *Let* $\lambda_{d,1} = \mu_{\chi(d)} + c_1\sigma_{\chi(d)}$ *and* $\lambda_{d,2} = \mu_{\chi(d)} + c_2\sigma_{\chi(d)}$, *where* $d \in \mathbb{N}^+$, $c_1 < c_2 \leq 0$, *and* $\mu_{\chi(d)}$ *and* $\sigma_{\chi(d)}$ *are the mean and standard deviation of the chi distribution with d degrees of freedom, respectively. Define* $\Delta\mu_d(\boldsymbol{x}_{d,1}, \boldsymbol{x}_{d,2}) = \mu_{\chi(d,\lambda_{d,2})} - \mu_{\chi(d,\lambda_{d,1})}$, *where* $\mu_{\chi(d,\lambda)}$ *is the mean of the noncentral chi distribution with d degrees of freedom and noncentrality parameter* λ. *Then, there exists* $d_0 \in \mathbb{N}$ *such that for every* $d > d_0$, $\Delta\mu_d(\lambda_{d,1}, \lambda_{d,2}) > 0$, *and* $\Delta\mu_{d+2}(\lambda_{d+2,1}, \lambda_{d+2,2}) > \Delta\mu_d(\lambda_{d,1}, \lambda_{d,2})$.

Intuitively, $\lambda_{d,1}$ and $\lambda_{d,2}$ represent two d-dimensional points whose norms are located at c_1 and c_2, resp., standard deviations from the expected value of the norm in the dimensionality d, and for which $\Delta\mu_d(\lambda_{d,1}, \lambda_{d,2})$ is the distance between the expected value of the associated distribution of distances.

As stated by authors, the implication of the above theorem is that hubness is an inherent property of data distributions in high-dimensional space, rather than an artifact of other factors, such as finite sample size. However, Theorem 5 only formalizes the tendency of the difference between the means of the two distance distributions to increase with the dimensionality, and the proof is specific for Gaussian data. No model to predict the number of k-occurrences or to infer the form of the underlying distribution is provided, and the characterization of the distribution probability of N_k remains an open problem.

3. Notation

In the rest of this section, upper case letters, such as X, Y, Z, ..., denote random variables (r.v.) taking values in $\mathbb{R}$. f_X (F_X, resp.) denotes the probability density function (pdf) (probability distribution function (cdf), resp.) associated with X.

Boldface uppercase letters with d as a subscript, such as $\boldsymbol{X}_d$, $\boldsymbol{Y}_d$, $\boldsymbol{Z}_d$, ..., denote d-dimensional random vectors taking values in $\mathbb{R}^d$. The components X_i ($1 \leq i \leq d$) of a random vector $\boldsymbol{X}_d = (X_1, X_2, \ldots, X_d)$ are random variables having pdfs $f_{X_i} = f_i$ (cdf $F_{X_i} = F_i$). A d-dimensional random vector is said to be independent and identically distributed (i.i.d. for short) if its random variables are independent and have common pdf $f_X = f_{X_i}$ (cdf $F_X = F_{X_i}$).

Boldface lowercase letters with d as a subscript, such as $\boldsymbol{x}_d$, $\boldsymbol{y}_d$, $\boldsymbol{z}_d$, ..., denote a specific d-dimensional vector taking value in $\mathbb{R}^d$. The components of a vector $\boldsymbol{x}_d = (x_1, x_2, \ldots, x_d)$, denoted as x_i ($1 \leq i \leq d$), are real scalar values.

Given a random variable X, w.l.o.g. and for simplicity of treatment, sometimes it is assumed that the expected value μ (or μ_X) of f_X is $\mu = 0$. If that is not the case, to satisfy the assumption, it suffices to replace during the analysis the original random variable X with the novel random variable $\hat{X} = X - \mu_X$. Thus, $\hat{\boldsymbol{X}}_d$ denotes the random vector $\boldsymbol{X}_d - \mu_X$.

σ_X or $\sigma(X)$ (or σ alone, whenever X is clear from the context) is the standard deviation of the random variable X. By μ_k ($\hat{\mu}_k$, resp.), or $\mu_{X,k}$ ($\hat{\mu}_{X,k}$, resp.) whenever X is not clear from the context, it is denoted the *k-th moment* (*k-th central moment*, resp.) ($k > 0$) $\mu_k = \mathbf{E}[X^k]$ ($\hat{\mu}_k = \mathbf{E}[(X - \mu_X)^k]$, resp.) of the random variable X, where $\mathbf{E}[X]$ is the expectation of X. Clearly, when $\mu = \mu_1 = 0$, μ_k coincides with $\hat{\mu}_k$ and $\mu_2 = \sigma^2$.

Moments of a pdf f (cdf F, resp.) are those associated with a random variable X having pdf $f_X = f$ (cdf $F_X = f$, resp.). The moments of an i.i.d. random vector $\boldsymbol{X}_d$ are those associated with its cdf F_X.

It is said that a distribution function has *finite* (*central*) *moment* μ_k, if there exists $0 \leq \mu_{top} < \infty$ such that $|\mu_k| \leq \mu_{top}$.

Whenever moments are employed during the proofs, we always assume that they exist and are finite. Moreover, if the random variable associated with a moment employed in a proof is not explicitly stated, we assume that the moment is relative to the common cdf of the random vector(s) occurring in the distribution reported in the statement of the theorem.

Moreover, it is sometimes considered the case that $\mu_3 = 0$, a condition that is referred to as null *skewness*. It is known that odd central moments, provided they exist, are null if the pdf of X is symmetric with respect to the mean (with examples of distributions having null μ_3 value being the Uniform and Normal distributions).

The notation $\mathcal{N}(\mu, \sigma^2)$ represents the Normal distribution function with mean μ and variance σ^2. By Φ (ϕ, resp.) one denotes the cdf (pdf, resp.) of the standard normal distribution, whereas by Φ_X (ϕ_X, resp.) one denotes the cdf (pdf, resp.) of the normal distribution with mean μ_X and variance σ_X^2.

Let X represent a univariate random variable that is defined in terms of a real-valued function of one or more d-dimensional random vectors. For example, X could be defined as $\|\boldsymbol{X}_d\|^2$. The notation $X \simeq \mathcal{N}(\mu_X, \sigma_X^2)$ is shorthand to denote the fact that, as $d \to \infty$, the distribution $\widehat{F}_X$ of the standard score $\frac{X - \mu_X}{\sigma_X}$ of X converges to the normal distribution

$\mathcal{N}(0,1)$. Thus, for large values of d, $\mathcal{N}(\mu_X, \sigma_X^2)$ approximates the distribution probability F_X of X, and $Pr[X \leq \delta] \approx \Phi\left(\frac{\delta - \mu_X}{\sigma_X}\right)$.

In the following, $\|\cdot\|$ denotes the L_2 norm, i.e., $\|\boldsymbol{x}_d\| = \sqrt{\sum_{i=1}^{d} x_i^2}$. Moreover, $\text{dist}(P,Q)$ denotes the Euclidean distance $\|P - Q\|$ between (random) vector P and (random) vector Q.

Let $x \in \mathbb{R}$, and let X be a random variable. Then

$$z_{x,X} = \frac{x - \mu_X}{\sigma_X}$$

represents the value x standardized with respect to the mean and the standard deviation of X. For a d-dimensional vector $\boldsymbol{x}_d$, which is the realization of a d-dimensional i.i.d. random vector $\boldsymbol{Y}_d$, the notation $z_{\boldsymbol{x}_d}$ is used as shorthand for $z_{\|\boldsymbol{x}_d\|^2, \|\boldsymbol{Y}_d\|^2}$, i.e.,

$$z_{\boldsymbol{x}_d} = z_{\|\boldsymbol{x}_d\|^2, \|\boldsymbol{Y}_d\|^2} = \frac{\|\boldsymbol{x}_d\|^2 - \mu_{\|\boldsymbol{Y}_d\|^2}}{\sigma_{\|\boldsymbol{Y}_d\|^2}}.$$

Results in the following are derived by considering distributions. However, these results can be applied to a finite set of points by taking into account large samples. In order to deal with a finite set of points, $\{\boldsymbol{Y}_d\}_n$ denotes a set of n random vectors $\{\boldsymbol{Y}_d^{(1)}, \ldots, \boldsymbol{Y}_d^{(n)}\}$, each one distributed as $\boldsymbol{Y}_d$.

Now we recall the Lyapunov Central Limit Theorem (CLT) condition. Consider the sequence $W_1, W_2, W_3, \ldots$ of independent, but not identically distributed, random variables, and let $V_d = \sum_{i=1}^{d} W_i$. By the Lyapunov CLT condition (Ash and Doléans-Dade, 1999), if for some $\delta > 0$ it holds that

$$\lim_{d \to \infty} \frac{1}{s_d^{2+\delta}} \sum_{i=1}^{d} \mathbf{E}\left[|W_i - \mathbf{E}[W_i]|^{2+\delta}\right] = 0, \text{ where } s_d^2 = \sum_{i=1}^{d} \sigma_{W_i}^2, \tag{1}$$

then, as d goes to infinity,

$$\frac{U_d - \mathbf{E}[U_d]}{\sigma(U_d)} = \frac{\sum_{i=1}^{d} W_i - \sum_{i=1}^{d} \mathbf{E}[W_i]}{\sqrt{\sum_{i=1}^{d} \sigma_{W_i}^2}} \to \mathcal{N}(0,1),$$

i.e., the standard score $(V_d - \mathbf{E}[V_d])/\sigma(V_d)$ converges in distribution to a standard normal random variable.

In the following, when a statement involves a d-dimensional vector $\boldsymbol{x}_d$, we will usually assume that $\boldsymbol{x}_d$ is the realization of a specific d-dimensional random vector $\boldsymbol{X}_d$. Moreover, we will say that a result involving the realization $\boldsymbol{x}_d$ of a random vector $\boldsymbol{X}_d$ holds *with high probability* if the statement is true for all the realizations of $\boldsymbol{X}_d$ except for a subset which becomes increasingly less probable as the dimensionality d increases.

Technically, the assumption that $\boldsymbol{x}_d$ is a realization of a random vector $\boldsymbol{X}_d$ is leveraged to attain a proof of convergence in probability. This also means that when we simultaneously account for all the realizations of a random vector $\boldsymbol{X}_d$ (by integrating on all vectors $\boldsymbol{x}_d$ such that $f_X(\boldsymbol{x}_d) > 0$), the existence of such a negligible set does not affect the final result.

4. Results

This section presents the results of the work concerning distribution of distances, nearest neighbors, and reverse nearest neighbors.

Specifically, Section 4.1, concerning the distribution of distances between intrinsically high-dimensional data, derives the expressions for the distance distribution between pairs of random vectors and between a realization of a random vector and a random vector, and analyzes the error associated with expressions.

Section 4.2 takes into account the distribution of distances from nearest neighbors, derives the expected size of the ϵ-neighborhood in high-dimensional spaces, and leverages it to characterize neighborhood instability. The section also derives a novel estimate of the relative contrast measure.

Section 4.3 addresses the problem of determining the number of k-occurrences and determines the closed form of its limiting distribution, showing that it encompasses previous results and provides interpretability of the associated hubness phenomenon.

Section 4.4 generalizes the results derived for the i.i.d. case to independent non-identically distributed data, depicting connections with the behavior in real data.

Section 4.5 discusses relationship to other distances, including Minkowski's metrics and, in general, distances satisfying certain conditions of spatial centrality.

The first three sections deal with i.i.d. random vectors. In these sections, the synthetic data sets considered consist of data generated from a uniform distribution in $[-0.5, +0.5]$, a standard normal distribution, and an exponential distribution with mean 1.

For the proofs that are not reported within the main text, the reader is referred to the Appendix.

4.1 On the Distribution of Distances for i.i.d. Data

First of all, the probability that two d-dimensional i.i.d. random vectors lie at a distance not greater than δ from one another is considered. The expression of Theorem 6 results from the fact that the distribution of the random variable $\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|^2$ converges towards the normal distribution for large dimensionalities.

Theorem 6 *Let $\boldsymbol{X}_d$ and $\boldsymbol{Y}_d$ be two d-dimensional i.i.d. random vectors with common cdf F. Then, for large values of d,*

$$Pr\left[\text{dist}(\boldsymbol{X}_d, \boldsymbol{Y}_d) \leq \delta\right] \approx \Phi\left(\frac{\delta^2 - 2d(\mu_2 - \mu^2)}{\sqrt{2d\Big(\mu_4 + \mu_2^2 + 2\mu\big(\mu(2\mu_2 - \mu^2) - 2\mu_3\big)\Big)}}\right).$$

Proof of Theorem 6. The statement follows from the property shown in Lemma 7.

Lemma 7 $\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|^2 \simeq \mathcal{N}\left(2d(\mu_2 - \mu^2),\ 2d\left(\mu_4 + \mu_2^2 + 2\mu\big(\mu(2\mu_2 - \mu^2) - 2\mu_3\big)\right)\right)$.

Proof of Lemma 7. The squared norm $\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|^2$ can be written as $\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|^2 = \|\boldsymbol{X}_d\|^2 + \|\boldsymbol{Y}_d\|^2 - 2\langle \boldsymbol{X}_d, \boldsymbol{Y}_d \rangle$, where $\|\boldsymbol{X}_d\|^2 \equiv \|\boldsymbol{Y}_d\|^2$, and $\langle \boldsymbol{X}_d, \boldsymbol{Y}_d \rangle$ are the following random

variables

$$\|\boldsymbol{Y}_d\|^2 = \sum_{i=1}^{d} Y_i^2 \quad \text{and} \quad \langle \boldsymbol{X}_d, \boldsymbol{Y}_d \rangle = \sum_{i=1}^{d} X_i Y_i.$$

The proof proceeds by showing that, as $d \to \infty$, $\|\boldsymbol{X}_d\|^2$, $\|\boldsymbol{Y}_d\|^2$, and $\langle \boldsymbol{X}_d, \boldsymbol{Y}_d \rangle$ are both normally distributed and jointly normally distributed and by determining their covariance, which is accounted for in Propositions 8, 9, 10, and 11, as reported in the following.

Proposition 8 $\|\boldsymbol{Y}_d\|^2 \simeq \mathcal{N}\left(d\mu_2, d(\mu_4 - \mu_2^2)\right)$.

Proof of Proposition 8. Consider the random variable

$$\|\boldsymbol{Y}_d\|^2 = \sum_{i=1}^{d} Y_i^2 = \sum_{i=1}^{d} W_i,$$

where $W_i = Y_i^2$ is a novel random variable. Then, $\mu_W = \mathbf{E}[W_i] = \mathbf{E}[Y_i^2] = \mu_2$, and $\sigma_W^2 = \mathbf{E}[W_i^2] - \mathbf{E}[W_i]^2 = \mathbf{E}[Y_i^4] - \mu_2^2 = \mu_4 - \mu_2^2$.

Consider the sequence $W_1, W_2, W_3, \ldots$ of i.i.d. random variables. By the Central Limit Theorem (CLT for short) (Ash and Doléans-Dade, 1999), the standard score of W_i is such that, as $d \to \infty$,

$$\frac{\sum_{i=1}^{d} W_i - d\mu_W}{\sqrt{d\sigma_W}} = \frac{\sum_{i=1}^{d} Y_i^2 - d\mu_2}{\sqrt{d(\mu_4 - \mu_2^2)}} \to \mathcal{N}(0,1),$$

from which the result follows. ■

Proposition 9 $\langle \boldsymbol{X}_d, \boldsymbol{Y}_d \rangle \simeq \mathcal{N}\big(d\mu^2, d(\mu_2^2 - \mu^4)\big)$.

Proof of Proposition 9. Because $\langle \boldsymbol{X}_d, \boldsymbol{Y}_d \rangle = \sum_{i=1}^{d} X_i Y_i = \sum_{i=1}^{d} W_i$ is the sum of a sequence $W_1, W_2, W_3, \ldots$ of i.i.d. random variables with mean $\mathbf{E}[W_i] = \mathbf{E}[X_i Y_i] = \mathbf{E}[X_i]\mathbf{E}[Y_i] = \mu^2$ and variance $\sigma^2[W_i] = \mathbf{E}[W_i^2] - \mathbf{E}[W_i]^2 = \mathbf{E}[X_i^2 Y_i^2] - (\mu^2)^2 = \mathbf{E}[X_i^2]\mathbf{E}[Y_i^2] - \mu^4 = \mu_2^2 - \mu^4$, from the CLT the result follows. ■

Proposition 10 *As $d \to \infty$, $\|\boldsymbol{X}_d\|^2$, $\|\boldsymbol{Y}_d\|^2$ and $\langle \boldsymbol{X}_d, \boldsymbol{Y}_d \rangle$ are jointly normally distributed.*

Proof of Proposition 10. The statement holds provided that all linear combinations $W = a\|\boldsymbol{X}_d\|^2 + b\|\boldsymbol{Y}_d\|^2 + c\langle \boldsymbol{X}_d, \boldsymbol{Y}_d \rangle$ are normal. Notice that

$$W = a\left(\sum_{i=1}^{d} X_i^2\right) + b\left(\sum_{i=1}^{d} Y_i^2\right) + c\left(\sum_{i=1}^{d} X_i Y_i\right) = \sum_{i=1}^{d} \left(aX_i^2 + bY_i^2 + cX_i Y_i\right) = \sum_{i=1}^{d} W_i,$$

where $W_i = aX_i^2 + bY_i^2 + cX_i Y_i$ is a novel random variable. Because $W_1, W_2, W_3, \ldots$ is a sequence of i.i.d. random variables, the result follows from the CLT. ■

Proposition 11

$$cov\big(\|\boldsymbol{Y}_d\|^2, \langle \boldsymbol{X}_d, \boldsymbol{Y}_d \rangle\big) = d\mu(\mu_3 - \mu_2\mu)$$
$$\big(\textit{and } cov\big(\|\boldsymbol{X}_d\|^2, \langle \boldsymbol{X}_d, \boldsymbol{Y}_d \rangle\big) = d\mu(\mu_3 - \mu_2\mu), \textit{ for symmetry}\big).$$

Proof of Proposition 11. See the appendix. ■

Proof of Lemma 7 (continued). Because the random variables $\|\boldsymbol{X}_d\|^2$, $\|\boldsymbol{Y}_d\|^2$, and $\langle \boldsymbol{X}_d, \boldsymbol{Y}_d \rangle$ are jointly normally distributed (see Proposition 10), their linear combination $\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|^2 = \|\boldsymbol{X}_d\|^2 + \|\boldsymbol{Y}_d\|^2 - 2\langle \boldsymbol{X}_d, \boldsymbol{Y}_d \rangle$ is normally distributed with mean $\mu_{\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|^2} = \mu_{\|\boldsymbol{X}_d\|^2} + \mu_{\|\boldsymbol{Y}_d\|^2} - 2\mu_{\langle \boldsymbol{X}_d, \boldsymbol{Y}_d \rangle} = 2d(\mu_2 - \mu^2)$, and variance

$$\begin{aligned}
\sigma^2_{\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|^2} &= 2\sigma^2_{\|\boldsymbol{Y}_d\|^2} + (-2)^2\sigma^2_{\langle \boldsymbol{X}_d, \boldsymbol{Y}_d \rangle} + 4(-2)cov(\|\boldsymbol{Y}_d\|^2, \langle \boldsymbol{X}_d, \boldsymbol{Y}_d \rangle) = \\
&= 2d(\mu_4 - \mu_2^2) + 4d(\mu_2^2 - \mu^4) - 8d\mu(\mu_3 - \mu_2\mu) = \\
&= 2d\Big(\mu_4 + \mu_2^2 + 2\mu\big(\mu(2\mu_2 - \mu^2) - 2\mu_3\big)\Big).
\end{aligned}$$

■

Proof of Theorem 6 (continued). To conclude the proof: $Pr\left[\text{dist}(\boldsymbol{X}_d, \boldsymbol{Y}_d) \leq \delta\right] = Pr\left[\text{dist}(\boldsymbol{X}_d, \boldsymbol{Y}_d)^2 \leq \delta^2\right] = Pr\left[\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|^2 \leq \delta^2\right] \approx \Phi_{\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|^2}(\delta^2)$. ■

Note that, if $\boldsymbol{X}_d$ and $\boldsymbol{Y}_d$ have a common pdf with null mean ($\mu = 0$), $\|\boldsymbol{Y}_d\|^2$ ($\|\boldsymbol{X}_d\|^2$ equivalently) and $\langle \boldsymbol{X}_d, \boldsymbol{Y}_d \rangle$ are uncorrelated, and being jointly normal distributed, they are also independent. In such a case, the parameters of the distribution can be expressed in the following simplified form.

Corollary 12 *Let $\boldsymbol{X}_d$ and $\boldsymbol{Y}_d$ be two d-dimensional i.i.d. random vectors with common cdf F_X having mean μ. Then*

$$\|\hat{\boldsymbol{X}}_d - \hat{\boldsymbol{Y}}_d\|^2 \simeq \mathcal{N}\left(2d\hat{\mu}_2,\ 2d(\hat{\mu}_4 + \hat{\mu}_2^2)\right),$$

where $\hat{\boldsymbol{X}}_d = \boldsymbol{X}_d - \mu$ ($\hat{\boldsymbol{Y}}_d = \boldsymbol{Y}_d - \mu$, resp.) and $\hat{\mu}_k = \mathbf{E}[(X - \mu)^k]$ ($k > 0$) are the central moments of f_X (the moments of $f_{\hat{X}}$, resp.).

Proof of Corollary 12. Immediate from Theorem 7. ■

The notability of the above expression also stems from the following fact.

Proposition 13 $Pr[\text{dist}(\boldsymbol{X}_d, \boldsymbol{Y}_d) \leq \delta] = Pr[\text{dist}(\hat{\boldsymbol{X}}_d, \hat{\boldsymbol{Y}}_d) \leq \delta]$.

Proof of Proposition 13. Distances are not affected by translation. ■

Until now, it has been assumed that $\boldsymbol{X}_d$ and $\boldsymbol{Y}_d$ have a common cdf. The following expression takes into account the case of different cdfs.

Corollary 14 *Let $\boldsymbol{X}_d$ and $\boldsymbol{Y}_d$ be two d-dimensional i.i.d. random vectors with cdfs F_X and F_Y, respectively. Then $\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|^2 \simeq \mathcal{N}(\mu_{X,Y},\ \sigma^2_{X,Y})$, where*

$$\begin{aligned}\mu_{X,Y} &= d(\mu_{X,2} + \mu_{Y,2} - 2\mu_X\mu_Y), \text{ and}\\ \sigma^2_{X,Y} &= d\big((\mu_{X,4} - \mu^2_{X,2}) + (\mu_{Y,4} - \mu^2_{Y,2}) + 4\mu_{X,2}\mu_{Y,2} + 4\mu_X\mu_Y\big(\mu_{X,2} + \mu_{Y,2} - \mu_X\mu_Y\big) + \\ &\quad -4\mu_X\mu_{Y,3} - 4\mu_Y\mu_{X,3}\big).\end{aligned}$$

Proof of Corollary 14. The expression can be obtained by following the same line of reasoning of Theorem 7. ■

To characterize more precisely distance distributions, it is of interest to consider the case in which one of the two vectors is held fixed. With this aim, the following Theorem 15 concerns the probability that a given d-dimensional vector $\boldsymbol{x}_d$ and the realization of a d-dimensional i.i.d. random vector $\boldsymbol{Y}_d$ lie at a distance not greater than δ from one another. The result holds under the condition that $\boldsymbol{x}_d$ itself is the realization of a d-dimensional i.i.d. random vector $\boldsymbol{X}_d$, with the cdf F_X of $\boldsymbol{X}_d$ not necessarily being identical to the cdf F_Y of $\boldsymbol{Y}_d$.

Formally, Theorem 15 holds with high probability because it relies on a proof of convergence in probability exploited in Proposition 17. Although not all the realizations may comply with the condition of Proposition 17 (e.g., consider the case $x_i = c^i$ with $c \neq 1$), it holds anyway for almost all the realizations, except for a set of vanishing measure.

Theorem 15 *Let $\boldsymbol{x}_d$ denote a realization of a d-dimensional i.i.d. random vector $\boldsymbol{X}_d$, and let $\boldsymbol{Y}_d$ be a d-dimensional i.i.d. random vector. Then, for large values of d, with high probability*

$$Pr\left[\text{dist}(\boldsymbol{x}_d, \boldsymbol{Y}_d) \leq \delta\right] \approx \Phi\left(\frac{\delta^2 - \|\boldsymbol{x}_d\|^2 - d\mu_2 + 2\mu\sum_{i=1}^d x_i}{\sqrt{d(\mu_4 - \mu_2^2) + 4(\mu_2 - \mu^2)\|\boldsymbol{x}_d\|^2 - 4(\mu_3 - \mu\mu_2)\sum_{i=1}^d x_i}}\right),$$

where moments are relative to the random vector $\boldsymbol{Y}_d$.

Proof of Theorem 15. The proof relies on the result of Lemma 16 considering the distribution of $\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2$.

Lemma 16 *With high probability*

$$\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2 \simeq \mathcal{N}\left(\|\boldsymbol{x}_d\|^2 + d\mu_2 - 2\mu\sum_{i=1}^d x_i,\ d(\mu_4 - \mu_2^2) + 4(\mu_2 - \mu^2)\|\boldsymbol{x}_d\|^2 - 4(\mu_3 - \mu\mu_2)\sum_{i=1}^d x_i\right).$$

Proof of Lemma 16. Consider the equality $\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2 = \|\boldsymbol{x}_d\|^2 + \|\boldsymbol{Y}_d\|^2 - 2\langle\boldsymbol{x}_d, \boldsymbol{Y}_d\rangle$. The proof proceeds by studying the distribution of $\langle\boldsymbol{x}_d, \boldsymbol{Y}_d\rangle$ (see Proposition 18), by showing that $\|\boldsymbol{Y}_d\|^2$ and $\langle\boldsymbol{x}_d, \boldsymbol{Y}_d\rangle$ are jointly normally distributed (see Proposition 19), and by determining their covariance (see Proposition 20). However, a technical result that is leveraged in the sequel of the proof is first needed; this is presented in Proposition 17.

Proposition 17 *Let $\boldsymbol{X}_d$ be a d-dimensional i.i.d. random vector having cdf F_X. Moreover, let p and q be positive integers, and $\beta_0, \beta_1, \ldots, \beta_p$, $\alpha_0, \alpha_1, \ldots, \alpha_q$ be real coefficients such that $\beta_p \neq 0$ and $\alpha_q \neq 0$. Then, for any $\epsilon > 0$,*

$$\lim_{d\to\infty} Pr\left[\left|\frac{\sum_{i=1}^d\left(\sum_{j=0}^p \beta_j X_i^j\right)}{\left(\sum_{i=1}^d\left(\sum_{j=0}^q \alpha_j X_i^j\right)\right)^2}\right| \geq \epsilon\right] = 0.$$

Proof of Proposition 17. See the appendix. ■

Proposition 18 *With high probability $\langle\boldsymbol{x}_d, \boldsymbol{Y}_d\rangle \simeq \mathcal{N}\big(\mu\sum_{i=1}^d x_i,\ (\mu_2 - \mu^2)\|\boldsymbol{x}_d\|^2\big)$.*

Proof of Proposition 18. Consider the random variable $\langle\boldsymbol{x}_d, \boldsymbol{Y}_d\rangle$:

$$\langle\boldsymbol{x}_d, \boldsymbol{Y}_d\rangle = \sum_{i=1}^d x_i Y_i = \sum_{i=1}^d W_i,$$

where $W_i = x_i Y_i$ is a novel random variable. Then, $\mu_{W_i} = \mathbf{E}[W_i] = \mathbf{E}[x_i Y_i] = x_i\mathbf{E}[Y_i] = x_i\mu$, and $\sigma^2_{W_i} = \mathbf{E}[W_i^2] - \mathbf{E}[W_i]^2 = \mathbf{E}[x_i^2 Y_i^2] - x_i^2\mu^2 = x_i^2\mu_2 - x_i^2\mu^2 = (\mu_2 - \mu^2)x_i^2$.

Consider the sequence $W_1, W_2, W_3, \ldots$ of independent, but not identically distributed, random variables. If the Lyapunov CLT condition reported in Equation (1) holds, the standard score $(\langle\boldsymbol{x}_d, \boldsymbol{Y}_d\rangle - \mu_{\langle\boldsymbol{x}_d,\boldsymbol{Y}_d\rangle})/\sigma_{\langle\boldsymbol{x}_d,\boldsymbol{Y}_d\rangle}$ converges in distribution to a standard normal random variable as d goes to infinity, i.e.,

$$\frac{\langle\boldsymbol{x}_d, \boldsymbol{Y}_d\rangle - \mu_{\langle\boldsymbol{x}_d,\boldsymbol{Y}_d\rangle}}{\sigma_{\langle\boldsymbol{x}_d,\boldsymbol{Y}_d\rangle}} = \frac{\sum_{i=1}^d W_i - \sum_{i=1}^d \mathbf{E}[W_i]}{\sum_{i=1}^d \sigma^2_{W_i}} = \frac{\sum_{i=1}^d x_i Y_i - \mu\sum_{i=1}^d x_i}{\sqrt{\mu_2 - \mu^2}\|\boldsymbol{x}_d\|} \to \mathcal{N}(0,1).$$

Thus, consider the Lyapunov condition for $\delta = 2$:

$$\lim_{d\to\infty} \frac{\sum_{i=1}^d \mathbf{E}\left[|W_i - \mathbf{E}[W_i]|^{2+\delta}\right]}{s_d^{2+\delta}}\bigg|_{\delta=2} = \lim_{d\to\infty} \frac{\sum_{i=1}^d \mathbf{E}\left[|x_i(Y_i - \mu)|^4\right]}{(\mu_2 - \mu^2)^2\|\boldsymbol{x}_d\|^4} =$$
$$= \frac{\mu_4 + \mu(6\mu\mu_2 - 4\mu_3 - 3\mu^3)}{(\mu_2 - \mu^2)^2} \cdot \lim_{d\to\infty} \frac{\sum_{i=1}^d x_i^4}{\left(\sum_{i=1}^d x_i^2\right)^2} = 0.$$

The above limit converges in probability for the r.v. $\boldsymbol{X}_d$ by Proposition 17. ■

Proposition 19 *As $d \to \infty$, with high probability $\|\boldsymbol{Y}_d\|^2$ and $\langle\boldsymbol{x}_d, \boldsymbol{Y}_d\rangle$ are jointly normally distributed.*

Proof of Proposition 19. See the appendix. ■

Proposition 20 *cov* $\left(\|\boldsymbol{Y}_d\|^2, \langle \boldsymbol{x}_d, \boldsymbol{Y}_d \rangle\right) = (\mu_3 - \mu\mu_2) \sum_{i=1}^{d} x_i$.

Proof of Proposition 20. See the appendix. ■

Proof of Lemma 16 (continued). To conclude the proof of Lemma 16, because the random variables $\|\boldsymbol{Y}_d\|^2$, and $\langle \boldsymbol{x}_d, \boldsymbol{Y}_d \rangle$ are jointly normally distributed, then the random variable $\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2$ is normally distributed with mean

$$\mu_{\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2} = \mu_{\|\boldsymbol{x}_d\|^2} + \mu_{\|\boldsymbol{Y}_d\|^2} - 2\mu_{\langle \boldsymbol{x}_d, \boldsymbol{Y}_d \rangle} = \|\boldsymbol{x}_d\|^2 + d\mu_2 - 2\mu \sum_{i=1}^{d} x_i,$$

and variance

$$\begin{aligned} \sigma^2_{\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2} &= \sigma^2_{\|\boldsymbol{Y}_d\|^2} + (-2)^2 \sigma^2_{\langle \boldsymbol{x}_d, \boldsymbol{Y}_d \rangle} + 2(-2)\ cov(\|\boldsymbol{Y}_d\|^2, \langle \boldsymbol{x}_d, \boldsymbol{Y}_d \rangle). = \\ &= d(\mu_4 - \mu_2^2) + 4(\mu_2 - \mu^2)\|\boldsymbol{x}_d\|^2 - 4(\mu_3 - \mu\mu_2)\left(\sum_{i=1}^{d} x_i\right). \end{aligned}$$

■

Proof of Theorem 15 (continued). To conclude the proof: $Pr\left[\text{dist}(\boldsymbol{x}_d, \boldsymbol{Y}_d) \leq \delta\right] = Pr\left[\text{dist}(\boldsymbol{x}_d, \boldsymbol{Y}_d)^2 \leq \delta^2\right] = Pr\left[\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2 \leq \delta^2\right] = \Phi_{\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2}(\delta^2)$. ■

For distributions having null means, the above expressions can be simplified.

Corollary 21 *Let $\boldsymbol{x}_d$ denote a realization of a d-dimensional i.i.d. random vector $\boldsymbol{X}_d$, and let $\boldsymbol{Y}_d$ be a d-dimensional i.i.d. random vector with cdf F_Y having null mean $\mu_Y = 0$. Then, with high probability*

$$\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2 \simeq \mathcal{N}\left(\|\boldsymbol{x}_d\|^2 + d\mu_2,\ d(\mu_4 - \mu_2^2) + 4\mu_2\|\boldsymbol{x}_d\|^2 - 4\mu_3 \sum_{i=1}^{d} x_i\right),$$

where the moments are relative to the random vector $\boldsymbol{Y}_d$.

Proof of Corollary 21. The result follows by substituting $\mu = \mu_Y = 0$ in the right-hand side of the statement of Lemma 16. ■

In order to quantify the error associated with the approximations of Theorem 6 and Theorem 15, the Kolmogorov-Smirnov statistic D_n is employed here as an error measure. This statistic is usually used for comparing a theoretical cumulative distribution function F to a given empirical distribution function G_n for n observations, and it is defined as

$$D_n(G_n, F) = \sup_{\delta \in \mathbb{R}} |G_n(\delta) - F(\delta)|.$$

In our case, given an empirical distribution function $G_{d,n}$ for n observations and a theoretical distribution function F_d, both related to the dimensionality parameter d, we define the error $err_d(G_{d,n}, F_d)$ as $D_n(G_{d,n}, F_d)$.

As for the approximation of Theorem 6, $F_{\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|^2}(\delta) = \Phi\left(\frac{\delta - \mathbf{E}[\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|^2]}{\sigma(\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|^2)}\right)$ is employed as theoretical cdf F_d, whereas $F^{emp}_{\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|^2,n}(\delta)$, denoting the empirical distribution of the squared distances, is employed as the empirical cdf $G_{d,n}$, and the error measured is $e_d = err_d\left(F^{emp}_{\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|^2,n}, F_{\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|^2}\right)$.

As the approximation of Theorem 15, given the realization $\boldsymbol{x}_d$ of $\boldsymbol{X}_d$, $F_{\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2}(\delta) = \Phi\left(\frac{\delta - \mathbf{E}[\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2]}{\sigma(\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2)}\right)$ is employed as a theoretical cdf, whereas $F^{emp}_{\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2,n}(\delta)$ denotes the empirical cdf. Specifically, we considered three points $\boldsymbol{p}_d^{(i)}$ $(1 \leq i \leq 3)$ as instances of $\boldsymbol{x}_d$. Each point $\boldsymbol{p}_d^{(i)}$ lies k_i (with $k_1 = 0$, $k_2 = 1$, and $k_3 = 5$) standard deviations $\sigma_{\|\boldsymbol{X}_d\|^2}$ away from the mean $\mu_{\|\boldsymbol{X}_d\|^2}$ of the squared norm of $\boldsymbol{X}_d$, i.e., each point $\boldsymbol{p}_d^{(i)}$ is such that $z_{\|\boldsymbol{p}_d^{(i)}\|^2, \|\boldsymbol{X}_d\|^2} = k_i$ (in particular, the generic coordinate of $\boldsymbol{p}_d^{(i)}$ has value $\left((\mu_{\|\boldsymbol{X}_d\|^2} + k_i \cdot \sigma_{\|\boldsymbol{X}_d\|^2})/d\right)^{1/2}$). The error measured for each point is $e_d^{(i)} = err_d\left(F^{emp}_{\|\boldsymbol{p}_d^{(i)} - \boldsymbol{Y}_d\|^2,n}, F_{\|\boldsymbol{p}_d^{(i)} - \boldsymbol{Y}_d\|^2}\right)$.

The empirical cdf $F^{emp}_{\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|^2,n}$ has been obtained by generating n pairs $(\boldsymbol{x}_d^{(j)}, \boldsymbol{y}_d^{(j)})$ $(1 \leq j \leq n)$ of realizations of the random vectors $\boldsymbol{X}_d$ and $\boldsymbol{Y}_d$, respectively, and then by computing $F^{emp}_{\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|^2,n}(\delta) = \frac{1}{n}\sum_{j=1}^{n} I_{[0,\delta]}\left(\text{dist}(\boldsymbol{x}_d^{(j)}, \boldsymbol{y}_d^{(j)})\right)$, where I_S denotes the indicator function (with S representing a generic set), such that $I_S(v) = 1$, if $v \in S$, and $I_S(v) = 0$, otherwise. The empirical cdf $F^{emp}_{\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2,n}$, is obtained by generating n realizations $\boldsymbol{y}_d^{(j)}$ $(1 \leq j \leq n)$ of the random vector $\boldsymbol{Y}_d$, and then by computing $F^{emp}_{\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2,n}(\delta) = \frac{1}{n}\sum_{j=1}^{n} I_{[0,\delta]}\left(\text{dist}(\boldsymbol{x}_d, \boldsymbol{y}_d^{(j)})\right)$.

We note that, for any distance threshold $\delta \geq 0$, the value err_d represents an upper bound to the error committed when the theoretical cdf of Theorem 6 (Theorem 15, resp.) is used to estimate the probability $Pr[\|\boldsymbol{X}_d - \boldsymbol{Y}_d\| \leq \delta]$ ($Pr[\|\boldsymbol{x}_d - \boldsymbol{Y}_d\| \leq \delta]$, resp.).

Figure 1 shows the above defined errors e_d, $e_d^{(1)}$, $e_d^{(2)}$, and $e_d^{(3)}$ (red curves), for distributions $F_X = F_Y$, uniform in $[-0.5, +0.5]$ (Fig. 1a), standard normal (Fig. 1b), and exponential with $\lambda = 1$ (Fig. 1c), respectively.

Before commenting on the results, it must be pointed out that the error err_d depends on the size n of the sample employed to build the empirical distribution. Thus, first we discuss the behavior for unbounded sample sizes n, and then take into account the effect of finite sample sizes. In order to simulate an unbounded sample size, the curves in the figures have been obtained for a very large sample size $n > 1.5 \cdot 10^8$.

From Figures 1a, 1b, and 1c it can be seen that the error err_d decreases with the dimensionality. The trend of the error curves is more regular for the uniform and normal distribution than for the exponential distribution, probably due to the skewness of the exponential distribution. The error e_d associated with the cdf $F_{\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|^2}$ is greater than the errors $e_d^{(i)}$ associated with the cdf $F_{\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2}$. Intuitively, this can be explained since the degree of uncertainty is reduced if one of the two random vectors is replaced by a fixed point. In general, it holds that $e_d^{(1)} > e_d^{(2)} > e_d^{(3)}$, thus indicating that uncertainty

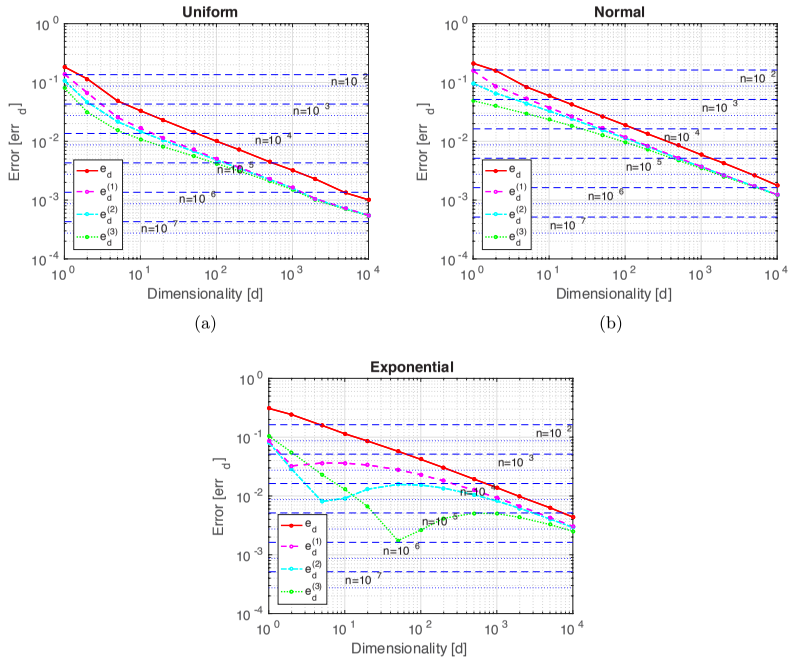


Figure 1: [Best viewed in color.] Empirical evaluation of the approximation errors of Th. 6 and Th. 15, for dimensionalities $d \in [10^0, 10^4]$ and sample sizes $n \in [10^2, 10^7]$. Error e_d (red solid line) is associated with the expression of Th. 6, whereas errors $e_d^{(1)}$ (magenta dashed line), $e_d^{(2)}$ (cyan dash-dotted line), and $e_d^{(3)}$ (green dotted line) are associated with the expression of Th. 15, for three different points whose squared norm standard scores are 0, 1, and 5, respectively. Horizontal blue lines take into account the sample size n: the dotted line is the expected error for different n values under the hypothesis that the distance distribution is indeed normal; the dashed line is the value under which the hypothesis that the sample is generated according the theoretical distribution can be accepted at the 95% confidence level.

increases towards the most largely populated regions of the space. Moreover, the larger the dimensionality d, the closer the errors $e_d^{(j)}$ to $e_d^{(1)}$.

As anticipated above, the error err_d depends on the size n of the sample employed to build the empirical distribution. Specifically, differently from the case of unbounded n values, for which the error decreases with the dimensionality, for any fixed sample size n, there exists a dimensionality d such that the error converges around a value $\overline{e}_n$. Such a value $\overline{e}_n$ corresponds to the error $D_n(\Phi_n^{emp}, \Phi)$ between the empirical cdf Φ_n^{emp} associated with a random sample of n elements of a (standard) normal distribution and the theoretical cdf Φ of a (standard) normal distribution.

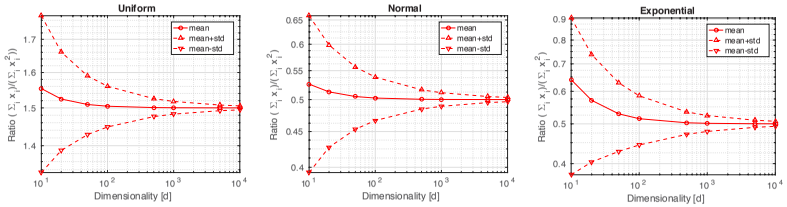
Let K be a random variable having a Kolmogorov distribution. According to the Kolmogorov-Smirnov test, the null hypothesis that the sample of n observations having empirical distribution G_n comes from the hypothesized distribution F is rejected at level $\alpha \in (0,1)$ if the statistic $\sqrt{n} \cdot D_n(G_n, F)$ is greater than the value K_α, where K_α is such that $Pr[K \leq K_\alpha] = 1 - \alpha$. It follows from the above that if, for a certain α and sample size n, it holds that err_d is smaller than $e_n^\alpha = K_\alpha \cdot n^{-1/2}$, then the hypothesis that the sample complies with the theoretical distribution can be accepted at the $1-\alpha$ confidence level, e.g., for $\alpha = 0.05$, the value $K_{0.05}$ is 1.3581. Moreover, the expected value $\overline{e}_n$ of $D_n(\Phi_n^{emp}, \Phi)$ approximately corresponds to $K_{\overline{\alpha}} \cdot n^{-1/2}$ with $\overline{\alpha} = 0.44$, i.e., to $\overline{e}_n \approx 0.8673 \cdot n^{-1/2}$.

Horizontal (blue) lines in Figures 1a, 1b, and 1c take into account the effect of the sample size n. Each pair of dashed and dotted lines is associated with a different value of $n \in \{10^2, 10^3, \ldots, 10^7\}$. Dashed lines are associated with the errors $e_n^{0.05}$, whereas dotted lines are associated with the errors $\overline{e}_n$. Let n be the actual sample size, and let d^* be the dimensionality such that the value e_{d^*} of the particular curve e_d is equal to $\overline{e}_n$ (dotted horizontal curve). Then, for $d \geq d^*$, the expected value of e_d tends to $\overline{e}_n$. Thus, for $d < d^*$, the curve of e_d is similar to the one reported in the figure, whereas for $d > d^*$, the curve of e_d tends to be horizontal, with a value close to $\overline{e}_n$. Moreover, if $e_d \leq e_n^{0.05}$ (dashed horizontal curve), then the hypothesis that the sample complies with the hypothesized distribution can be accepted at the 95% confidence level. Informally speaking, this means that in the latter case, the distribution hypothesized in Theorems 6 and 15 is indiscernible from the underlying distribution generating the observed inter-point distances.

In summary, as previously pointed out, because err_d depends on the worst-case threshold value δ, it is an upper bound to the error committed when estimating probabilities by leveraging the results previously presented. The analysis with unbounded sample size highlights that the worst-case error always decreases with the dimensionality. Moreover, let the effective error be defined as the difference between the observed error and the error expected when the data are generated according to the hypothesized distribution. The analysis of finite sample sizes highlights that, in practice, the effective error can become null.

For the distributions F_Y having both a null mean and null skewness ($\mu_3 = 0$), it follows from Propositions 19 and 20 that the random variables $\|\boldsymbol{Y}_d\|^2$ and $\langle \boldsymbol{x}_d, \boldsymbol{Y}_d \rangle$ are independent.

Moreover, the distribution defined in Corollary 21, in Theorem 15 and in Lemma 16, depend only on the squared norm $\|\boldsymbol{x}_d\|^2$, whereas the actual value of $\boldsymbol{x}_d$ does not matter. However, it can be shown that the same property holds also for skewed distributions, since the term $(\sum_i x_i)$ is related to $\|\boldsymbol{x}_d\|^2$, as accounted for in the subsequent result.



(a) (b) (c)

Figure 2: Empirical validation of Proposition 22 on different distributions: (a) uniform ($\mu_1/\mu_2 = 1.5$), (b) normal ($\mu_1/\mu_2 = 0.5$), and (c) exponential ($\mu_1/\mu_2 = 0.5$). The red solid curve represents the expected value μ_W of the ratio $W = (\sum_{i=1}^{d} X_i)/\|\boldsymbol{X}_d\|^2$, whereas the red dashed curves represent the values $\mu_W + \sigma_W$ and $\mu_W - \sigma_W$, measured for $n = 20{,}000$ points and $d \in [10^1, 10^4]$.

Proposition 22 *Let $\boldsymbol{x}_d$ denote a realization of a d-dimensional i.i.d. random vector $\boldsymbol{X}_d$ with cdf F_X. Then, for large values of d, with high probability*

$$\frac{\sum_{i=1}^{d} x_i}{\|\boldsymbol{x}_d\|^2} \to \frac{\mu_X}{\mu_{X,2}}.$$

Proof of Proposition 22. See the appendix. ■

Thus, the term $(\sum_i x_i)$ can be approximated by $\frac{\mu_X}{\mu_{X,2}}\|\boldsymbol{x}_d\|^2$.

Notice that the above result also states that for random vectors $\boldsymbol{X}_d$ having null mean, the term $(\sum_i x_i)$ becomes negligible with respect to $\|\boldsymbol{x}_d\|^2$ and, hence, that it can be ignored in the expression reported in Corollary 21, thus removing the dependence from the skewness of the distribution F_Y.

To empirically validate Proposition 22, the mean and the standard deviation of the ratio $W = (\sum_{i=1}^{d} X_i)/\|\boldsymbol{X}_d\|^2$ have been measured on distributions having non-null mean $\mu \neq 0$. Figure 2 reports the result of the experiment for $d \in [10, 10^4]$ and $n = 20{,}000$. Specifically, a uniform distribution with mean $\mu_1 = 0.5$ ($\mu_2 = 0.333$, $\mu_3 = 0.25$, and $\mu_4 = 0.2$) and ratio $\mu_1/\mu_2 = 1.5$ (Fig. 2a), a normal distribution with mean $\mu_1 = 1$ ($\mu_2 = 2$, $\mu_3 = 4$, and $\mu_4 = 10$) and ratio $\mu_1/\mu_2 = 0.5$ (Fig. 2b), and an exponential distribution with mean $\mu_1 = 1$ ($\mu_2 = 2$, $\mu_3 = 6$, and $\mu_4 = 24$) and ratio $\mu_1/\mu_2 = 0.5$ (Fig. 2c), were considered. It can be seen that the expected value $\mathbf{E}[W]$ of the ratio W rapidly converges to the limiting value μ_1/μ_2 and also that the standard deviation $\sigma(W)$ of the ratio W decreases with the dimensionality. Moreover, in all cases, the trend agrees with the prediction of Proposition 22, according to which it holds that $|\mathbf{E}[W] - \mu_1/\mu_2| = O(d^{-1})$ and $\sigma(W) = O(d^{-1/2})$.

4.2 On the Distribution of Nearest Neighbors for i.i.d. Data

Given a real number $\varrho \in [0, 1]$, a d-dimensional vector $\boldsymbol{x}_d$ and a d-dimensional random vector $\boldsymbol{Y}_d$, $\text{distnn}_\varrho(\boldsymbol{x}_d, \boldsymbol{Y}_d)$ denotes the radius of the smallest neighborhood centered in $\boldsymbol{x}_d$ containing at least the ϱ fraction of the realizations of $\boldsymbol{Y}_d$. Moreover, $nn_\varrho(\boldsymbol{x}_d, \boldsymbol{Y}_d)$, or $nn_\varrho(\boldsymbol{x}_d)$ whenever $\boldsymbol{Y}_d$ is clear from the context, also called *ϱ-th nearest neighbor* of $\boldsymbol{x}_d$ w.r.t. $\boldsymbol{Y}_d$, denotes an element of the set $\{\boldsymbol{y}_d \in \mathbb{R}^d \mid f_Y(\boldsymbol{y}_d) > 0 \text{ and } \text{dist}(\boldsymbol{x}_d, \boldsymbol{y}_d) = \text{distnn}_\varrho(\boldsymbol{x}_d, \boldsymbol{Y}_d)\}$.[1] $\text{NN}_\varrho(\boldsymbol{x}_d, \boldsymbol{Y}_d)$, or $\text{NN}_\varrho(\boldsymbol{x}_d)$ whenever $\boldsymbol{Y}_d$ is clear from the context, denotes the set of points $\{\boldsymbol{y}_d \in \mathbb{R}^d \mid f_Y(\boldsymbol{y}_d) > 0 \text{ and } \text{dist}(\boldsymbol{x}_d, \boldsymbol{y}_d) \leq \text{dist}(\boldsymbol{x}_d, nn_\varrho(\boldsymbol{x}_d, \boldsymbol{Y}_d))\}$.

In order to deal with finite sets of n points $\{\boldsymbol{Y}_d\}_n$, the integer parameter $k = \varrho n$ ($k \in \{1, \ldots, n\}$) has to be employed in place of ϱ. Thus, given a positive integer k, $\text{distnn}_k(\boldsymbol{x}_d, \{\boldsymbol{Y}_d\}_n)$ represents the radius of the smallest neighborhood centered in $\boldsymbol{x}_d$ containing at least k points of $\{\boldsymbol{Y}_d\}_n$. Moreover, $nn_k(\boldsymbol{x}_d, \{\boldsymbol{Y}_d\}_n)$ or $nn_k(\boldsymbol{x}_d)$, also called *k-th nearest neighbor* of $\boldsymbol{x}_d$ in $\{\boldsymbol{Y}_d\}_n$, denotes an element of the set $\{\boldsymbol{y}_d \in \{\boldsymbol{Y}_d\}_n \mid \text{dist}(\boldsymbol{x}_d, \boldsymbol{y}_d) = \text{distnn}_k(\boldsymbol{x}_d, \{\boldsymbol{Y}_d\}_n)\}$.[2] $\text{NN}_k(\boldsymbol{x}_d, \{\boldsymbol{Y}_d\}_n)$, or $\text{NN}_k(\boldsymbol{x}_d, \boldsymbol{Y}_d)$, denotes the set of points $\{\boldsymbol{y}_d \in \{\boldsymbol{Y}_d\}_n \mid \text{dist}(\boldsymbol{x}_d, \boldsymbol{y}_d) \leq \text{dist}(\boldsymbol{x}_d, nn_\varrho(\boldsymbol{x}_d, \{\boldsymbol{Y}_d\}_n))\}$.

In the rest of the work, given a d-dimensional i.i.d. random vector $\boldsymbol{X}_d$ with cdf F_X, representing the distribution of the query points, and a d-dimensional i.i.d. random vector $\boldsymbol{Y}_d$ with cdf F_Y, representing the distribution of the data points, we assume w.l.o.g. that F_Y has null mean μ_Y. Indeed, if it is not the case, it is sufficient to replace them with the random vectors $\boldsymbol{X}'_d = \boldsymbol{X}_d - \mu_Y$ and $\boldsymbol{Y}'_d = \hat{\boldsymbol{Y}}_d = \boldsymbol{Y}_d - \mu_Y$ such that $\mu_{Y'} = 0$. Moreover, a realization $\boldsymbol{x}_d$ of $\boldsymbol{X}_d$ can be replaced with $\boldsymbol{x}'_d = \boldsymbol{x}_d - \mu_Y$.

The following result considers the distance separating a vector from its ϱ-th nearest neighbor w.r.t. a d-dimensional i.i.d. random vector.

Lemma 23 *Let $\boldsymbol{x}_d$ denote a realization of a d-dimensional i.i.d. random vector $\boldsymbol{X}_d$ having cdf F_X. Consider the ϱ-nearest neighbor $nn_\varrho(\boldsymbol{x}_d, \boldsymbol{Y}_d)$ of $\boldsymbol{x}_d$ w.r.t. a d-dimensional i.i.d. random vector $\boldsymbol{Y}_d$ with cdf F_Y. Assume, w.l.o.g., that F_Y has null mean $\mu_Y = 0$. Then, for large values of d, with high probability*

$$\text{dist}(\boldsymbol{x}_d, nn_\varrho(\boldsymbol{x}_d, \boldsymbol{Y}_d)) \approx \sqrt{\|\boldsymbol{x}_d\|^2 + d\mu_2 + \Phi^{-1}(\varrho)\sqrt{d(\mu_4 - \mu_2^2) + 4\mu_2\|\boldsymbol{x}_d\|^2 - 4\mu_3 \sum_{i=1}^{d} x_i}}.$$

Proof of Lemma 23. By definition, $nn_\varrho(\boldsymbol{x}_d, \boldsymbol{Y}_d)$ is such that

$$Pr\left[\text{dist}(\boldsymbol{x}_d, \boldsymbol{Y}_d) \leq \text{dist}(\boldsymbol{x}_d, nn_\varrho(\boldsymbol{x}_d, \boldsymbol{Y}_d))\right] = \varrho.$$

By Corollary 21,

$$Pr\left[\text{dist}(\boldsymbol{x}_d, \boldsymbol{Y}_d) \leq \text{dist}(\boldsymbol{x}_d, nn_\varrho(\boldsymbol{x}_d, \boldsymbol{Y}_d))\right] \approx \Phi\left(\frac{\text{dist}(\boldsymbol{x}_d, nn_\varrho(\boldsymbol{x}_d, \boldsymbol{Y}_d))^2 - \mathbf{E}[\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2]}{\sigma(\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2)}\right).$$

1. Because our interest is only in the fact that $nn_\varrho(\boldsymbol{x}_d, \boldsymbol{Y}_d)$ satisfies the property $\text{dist}(\boldsymbol{x}_d, nn_\varrho(\boldsymbol{x}_d, \boldsymbol{Y}_d)) = \text{distnn}_\varrho(\boldsymbol{x}_d, \boldsymbol{Y}_d)$, it can be assumed that $nn_\varrho(\boldsymbol{x}_d)$ is randomly selected from the above set.
2. Because our interest is only in the fact that $nn_k(\boldsymbol{x}_d, \boldsymbol{Y}_d)$ satisfies the property $\text{dist}(\boldsymbol{x}_d, nn_k(\boldsymbol{x}_d, \{\boldsymbol{Y}_d\}_n)) = \text{distnn}_k(\boldsymbol{x}_d, \{\boldsymbol{Y}_d\}_n)$, it can be assumed that $nn_k(\boldsymbol{x}_d)$ is randomly selected from the above set.

Hence, $\text{dist}(\boldsymbol{x}_d, nn_\varrho(\boldsymbol{x}_d, \boldsymbol{Y}_d))^2 \approx \mathbf{E}[\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2] + \Phi^{-1}(\varrho)\,\sigma(\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2)$. ■

It has been already pointed out that if F_Y has null skewness ($\mu_3 = 0$), if $F_X = F_Y$, or if F_X has null mean $\mu_X = 0$, the term $4\mu_3(\sum_i x_i)$ can be disregarded.

Due to the difficulty of answering nearest neighbor queries in high-dimensional spaces, different authors have proposed to consider approximate nearest neighbor queries (Indyk and Motwani, 1998; Arya et al., 1998), returning an ϵ-approximate nearest neighbor instead of the exact nearest neighbor: given point $\boldsymbol{x}_d$ and $\epsilon \geq 0$, a point $\boldsymbol{y}_d$ is an ϵ-approximate nearest neighbor of $\boldsymbol{x}_d$ if it holds that $\text{dist}(\boldsymbol{x}_d, \boldsymbol{y}_d) \leq (1+\epsilon)\text{dist}(\boldsymbol{x}_d, nn_1(\boldsymbol{x}_d))$.

Beyer et al. (1999) called a nearest neighbor query *unstable* for a given $\epsilon \geq 0$, if the distance from the query point to most data points is less than $(1+\epsilon)$ times the distance from the query point to its nearest neighbor. Moreover, Beyer et al. (1999) have shown that in many situations, for any fixed $\epsilon > 0$, as dimensionality rises, the probability that a query is unstable converges to 1 (see Theorem 2).

Instability is undesirable because the points that fall in the enlarged query region, also called ϵ-neighborhood, are valid answers to the approximate nearest neighbor problem. Thus, the larger the expected number of data points falling within the ϵ-neighborhoods of the query points, the smaller the meaningfulness of the approximate query scenario.

Definition 24 *Let $\text{NN}^\epsilon_\varrho(\boldsymbol{x}_d, \boldsymbol{Y}_d)$ denote the set of the ϵ-approximate ϱ-nearest neighbors of $\boldsymbol{x}_d$, also called ϵ-neighborhood, that are the realizations $\boldsymbol{y}_d$ of $\boldsymbol{Y}_d$ whose distance from $\boldsymbol{x}_d$ is within $(1+\epsilon)$ times the distance separating $\boldsymbol{x}_d$ from its ϱ-th nearest neighbor $nn_\varrho(\boldsymbol{x}_d, \boldsymbol{Y}_d)$, i.e.,*

$$\text{NN}^\epsilon_\varrho(\boldsymbol{x}_d, \boldsymbol{Y}_d) = \{\boldsymbol{y}_d \in \mathbb{R}^d \mid f_Y(\boldsymbol{y}_d) > 0 \text{ and } \text{dist}(\boldsymbol{x}_d, \boldsymbol{y}_d) \leq (1+\epsilon)\,\text{dist}(\boldsymbol{x}_d, nn_\varrho(\boldsymbol{x}_d))\}.$$

In order to quantify the meaningfulness of ϵ-approximate queries, it is sensible to compute the expected size of the ϵ-neighborhoods associated with query points with respect to the data population, which is the task pursued in the following.

Theorem 25 *Let $\epsilon \geq 0$, let $\boldsymbol{X}_d$ be a d-dimensional i.i.d. random vector with cdf F_X, representing the distribution of the query points, and let $\boldsymbol{Y}_d$ be a d-dimensional i.i.d. random vector with cdf F_Y (not necessarily identical to F_X), representing the distribution of the data points. Assume, w.l.o.g., that F_Y has null mean $\mu_Y = 0$. Then, for large values of d,*

$$\mathbf{E}[|\text{NN}^\epsilon_k(\boldsymbol{X}_d, \{\boldsymbol{Y}_d\}_n)|] \approx$$
$$n\Phi\left(\frac{(\epsilon^2+2\epsilon)\,d(\mu_{X,2}+\mu_{Y,2}) + (1+\epsilon)^2\,\Phi^{-1}(\frac{k}{n})\sqrt{d\left(\mu_{Y,4} - \mu^2_{Y,2} + 4\mu_{Y,2}\mu_{X,2} - 4\mu_{Y,3}\mu_X\right)}}{\sqrt{d\left(\mu_{Y,4} - \mu^2_{Y,2} + 4\mu_{Y,2}\mu_{X,2} - 4\mu_{Y,3}\mu_X\right) + (\epsilon^2+2\epsilon)^2\,d(\mu_{X,4} - \mu^2_{X,2})}}\right).$$

Proof of Theorem 25. Consider the probability (exploiting Corollary 21, Proposition 22 and Lemmas 16 and 23)

$$\begin{aligned}
& Pr[\text{dist}(\boldsymbol{x}_d, \boldsymbol{Y}_d) \leq (1+\epsilon)\,\text{dist}(\boldsymbol{x}_d, nn_\varrho(\boldsymbol{x}_d, \boldsymbol{Y}_d))] = \\
= \;& Pr[\text{dist}(\boldsymbol{x}_d, \boldsymbol{Y}_d)^2 \leq (1+\epsilon)^2\text{dist}(\boldsymbol{x}_d, nn_\varrho(\boldsymbol{x}_d, \boldsymbol{Y}_d))^2] \approx \\
\approx \;& \Phi\left(\frac{(1+\epsilon)^2\left(\mathbf{E}[\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2] + \Phi^{-1}(\varrho)\,\sigma(\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2)\right) - \mathbf{E}[\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2]}{\sigma(\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2)}\right) = \\
= \;& \Phi\left(\frac{(\epsilon^2+2\epsilon)\,\mathbf{E}[\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2] + (1+\epsilon)^2\,\Phi^{-1}(\varrho)\,\sigma(\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2)}{\sigma(\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2)}\right) = \\
= \;& \Phi\left(\frac{(\epsilon^2+2\epsilon)(\|\boldsymbol{x}_d\|^2 + d\mu_{Y,2}) + (1+\epsilon)^2\,\Phi^{-1}(\varrho)\sqrt{d(\mu_{Y,4} - \mu^2_{Y,2}) + 4\mu_{Y,2}\|\boldsymbol{x}_d\|^2 - 4\mu_{Y,3}\frac{\mu_X}{\mu_{X,2}}\|\boldsymbol{x}_d\|^2}}{\sqrt{d(\mu_{Y,4} - \mu^2_{Y,2}) + 4\mu_{Y,2}\|\boldsymbol{x}_d\|^2 - 4\mu_{Y,3}\frac{\mu_X}{\mu_{X,2}}\|\boldsymbol{x}_d\|^2}}\right).
\end{aligned}$$

By taking into account the standard score of $\boldsymbol{x}_d$

$$\|\boldsymbol{x}_d\|^2 = z_{\boldsymbol{x}_d}\sigma_{\|\boldsymbol{X}_d\|^2} + \mu_{\|\boldsymbol{X}_d\|^2} = z_{\boldsymbol{x}_d}\sqrt{d(\mu_{X,4} - \mu^2_{X,2})} + d\mu_{X,2},$$

and by considering that for α, β, and z finite (note that $\phi(z)$ is practically negligible for $|z| \geq 5$) and d growing, $\sqrt{\alpha d + z\sqrt{\beta d}} \approx \sqrt{\alpha d}$, then the above probability can be approximated with $\Phi(a^{d,\epsilon,\varrho}_{X,Y} + b^{d,\epsilon,\varrho}_{X,Y} z_{\boldsymbol{x}_d})$, where

$$\begin{aligned}
a^{d,\epsilon,\varrho}_{X,Y} &= \frac{(\epsilon^2+2\epsilon)\,(d\mu_{X,2} + d\mu_{Y,2}) + (1+\epsilon)^2\,\Phi^{-1}(\varrho)\sqrt{d(\mu_{Y,4} - \mu^2_{Y,2}) + 4d\mu_{Y,2}\mu_{X,2} - 4d\mu_{Y,3}\mu_X}}{\sqrt{d(\mu_{Y,4} - \mu^2_{Y,2}) + 4d\mu_{Y,2}\mu_{X,2} - 4d\mu_{Y,3}\mu_X}}, \\
b^{d,\epsilon}_{X,Y} &= \frac{(\epsilon^2+2\epsilon)\sqrt{d(\mu_{X,4} - \mu^2_{X,2})}}{\sqrt{d(\mu_{Y,4} - \mu^2_{Y,2}) + 4d\mu_{Y,2}\mu_{X,2} - 4d\mu_{Y,3}\mu_X}}.
\end{aligned}$$

Consider now the expected value

$$\begin{aligned}
\mathbf{E}[|\text{NN}^\epsilon_\varrho(\boldsymbol{X}_d)|] &= \int_{\mathbb{R}^d} Pr[\text{dist}(\boldsymbol{x}_d, \boldsymbol{Y}_d) \leq (1+\epsilon)\,\text{dist}(\boldsymbol{x}_d, nn_\varrho(\boldsymbol{x}_d, \boldsymbol{Y}_d))] \cdot \Pr[\boldsymbol{X}_d = \boldsymbol{x}_d]\;\mathrm{d}\boldsymbol{x}_d = \\
&= \int_{z_{d,min}}^{z_{d,max}} \Phi\left(a^{d,\epsilon,\varrho}_{X,Y} + b^{d,\epsilon}_{X,Y}\,z_{\boldsymbol{x}_d}\right)\,\phi(z_{\boldsymbol{x}_d})\;\mathrm{d}z_{\boldsymbol{x}_d}.
\end{aligned}$$

The statement then follows by leveraging the following equation (Owen, 1980)

$$\int_{-\infty}^{+\infty} \Phi(a+bz)\,\phi(z)\;\mathrm{d}x = \Phi\left(\frac{a}{\sqrt{1+b^2}}\right), \tag{2}$$

taking the limits of integration to infinity. Note that the extra domain of integration considered is associated with a negligible probability because $z_{d,min} = (\mu_{\|\boldsymbol{X}_d\|^2} - \inf\|\boldsymbol{X}_d\|^2)/\sigma_{\|\boldsymbol{X}_d\|^2}$ and $z_{d,max} = (\sup\|\boldsymbol{X}_d\|^2 - \mu_{\|\boldsymbol{X}_d\|^2})/\sigma_{\|\boldsymbol{X}_d\|^2}$, are such that both $\phi(z_{d,\min})$ and $\phi(z_{d,\max})$ rapidly approach zero. ■

In order to validate the above result, the expected value $\mathbf{E}[|\text{NN}^\epsilon_k(\boldsymbol{X}_d, \{\boldsymbol{Y}_d\}_n)|]$ is empirically estimated for different values of k, d and $\epsilon \in [0, 0.5]$, by exploiting sets of $n = 10{,}000$

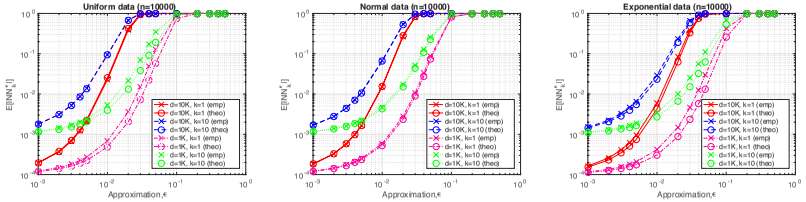


Figure 3: [Best viewed in color.] Comparison between the empirically estimated (x-marked curves) and the predicted by means of Th. 25 (o-marked curves) expected sizes of the ϵ-neighborhood, for $n = 10{,}000$, $d = 1{,}000$ and $k = 1$ (magenta dash-dotted line), $d = 1{,}000$ and $k = 10$ (green dotted line), $d = 10{,}000$ and $k = 1$ (red solid line), and $d = 10{,}000$ and $k = 10$ (blue dashed line).

realizations of the random vector $\boldsymbol{Y}_d$. Results are averaged by considering ten different sets. In the experiment, it is assumed that $F_X = F_Y$ and that each point of the set is used in turn as a query point; thus, the size of the ϵ-neighborhood may vary between k and $n-1$.

Figure 3 reports the results of this experiment for uniform, normal, and exponential i.i.d. data. The value $\mathbf{E}[|\mathrm{NN}_k^\epsilon(\boldsymbol{X}_d, \{\boldsymbol{Y}_d\}_n)|]$ empirically estimated as described above is compared with the value predicted by means of Theorem 25. The curves for the number of neighbors $k \in \{1, 10\}$ and the dimensionalities $d \in \{1{,}000, 10{,}000\}$ are reported. The curves confirm that the prediction follows the trend of the empirical evidence with the error vanishing as the dimensionality increases.

As already stated by Beyer et al. (1999), Theorem 2 only tells us what happens when we take the dimensionality to infinity, but nothing is said about the dimensionality at which do we anticipate nearest neighbors to become unstable, and the issue must be addressed through empirical studies.

The above dimensionality, called the *critical dimensionality*, can, however, be obtained as follows. Let $\theta \in [0, 1]$ represent a fraction of the data elements; the critical dimensionality $d^*_{\varrho,\epsilon,\theta}$ for the parameters ϱ and ϵ at the threshold level θ, also called selectivity, is such that

$$d^*_{\varrho,\epsilon,\theta} = \min\{d \in \mathbb{N}^+ : \mathbf{E}[|\mathrm{NN}^\epsilon_\varrho(\boldsymbol{X}_d, \boldsymbol{Y}_d)|] \geq \theta\},$$

i.e., the dimensionality at which the expected size of the ϵ-neighborhood contains at least the θ fraction of the data points.

Figure 4 reports the critical dimensionality for ϵ varying in $[0.001, 1]$, thresholds $\theta \in \{0.01, 0.1, 0.5\}$, $n = 10{,}000$ and $k = 1$ (i.e. $\varrho = k/(n-1) \approx 0.0001$), obtained by exploiting the expression reported in Theorem 25. For example, for $\theta = 0.1$, the plot says that for dimensionalities below the bottom curve, ϵ-neighborhoods contain on the average 10% of the points (one hundred points for $n = 10{,}000$). Note that analogous predictions can be obtained in a very similar way for any other combination of the parameters ϱ, θ, and ϵ, and distribution function F.

Figure 4 also report the values of the critical dimensionality estimated empirically (dashed lines). The plots highlight that the predicted critical dimensionality tends to the

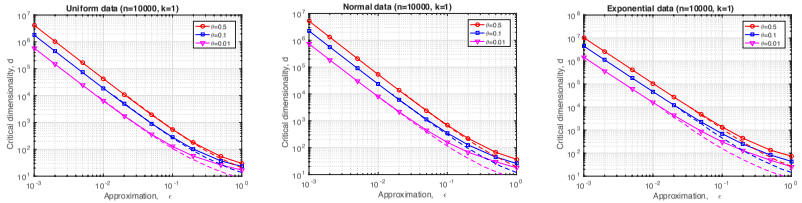


Figure 4: [Best viewed in color.] Critical dimensionality for $\epsilon \in [10^3, 10^0]$, $n = 10{,}000$, $k = 1$, and $\theta = 0.01$ (red solid line), $\theta = 0.1$ (blue solid line), and $\theta = 0.5$ (magenta solid line), predicted by exploiting Th. 25. The dashed curves represent the values of the critical dimensionality estimated empirically.

empirical one for decreasing ϵ and that the rate of convergence is directly proportional to θ. Interestingly, it can be seen that in different cases, the reported critical dimensionality is quite high (e.g., consider $\epsilon = 0.01$). Because approximate nearest neighbors must be associated with small values of θ (e.g., consider $\theta = 0.01$) to be considered meaningful, it can be concluded that the notion of approximate nearest neighbor can be considered meaningful even in high-dimensional spaces provided that the approximation factor ϵ is sufficiently small.

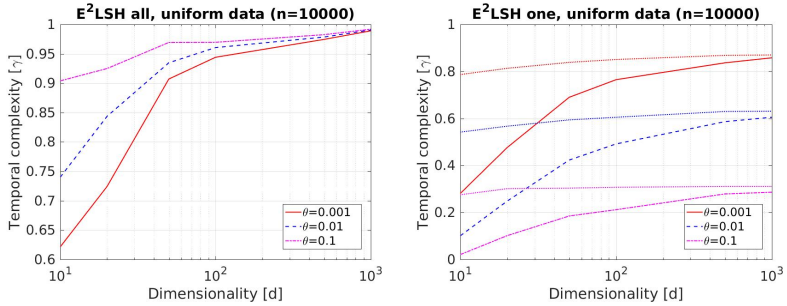
Unfortunately, this does not imply that algorithms perform efficiently in these cases. To illustrate, the researchers have proposed different algorithms for (approximate) nearest neighbor search problems. Most of these algorithms are randomized; that is, they are associated with a failure probability δ. Specifically, the *approximate near(est) neighbor search problem with failure probability* δ is defined as the problem to construct a data structure over a set of points $S \subseteq \mathbb{R}^d$ such that, given any query point $x \in \mathbb{R}^d$, with probability $1 - \delta$ reports:

P1. some $y \in S$ with $\mathrm{dist}(x, y) \leq (1 + \epsilon)r$ (*ϵ-approximate r-near neighbor*);

P2. some $y \in S$ with $\mathrm{dist}(x, y) \leq (1+\epsilon)\cdot\mathrm{dist}(x, nn(x, S))$ (*ϵ-approximate nearest neighbor*);

P3. each point $y \in S$ with $\mathrm{dist}(x, y) \leq r$ (*r-near neighbor reporting*).

The proposed algorithms offer trade-offs between the approximation factor, the space and the query time (Andoni, 2009). From the practical perspective, the space used by an algorithm should be as close to linear as possible. In this case, the best-existing solutions are based on locality-sensitive hashing (LSH) (Indyk and Motwani, 1998; Har-Peled et al., 2012). The idea of the LSH approach is to hash the points in a way that the probability of collision is much higher for points that are close (with the distance r) to each other than for those that are far apart (with distance at least $(1 + \epsilon)r$). Under different assumptions involving the parameters employed (Har-Peled et al., 2012), the LSH algorithm solves the ϵ-approximate r-near neighbor problem using $O(n^{1+\rho_\epsilon})$ extra space, $O(dn^{\rho_\epsilon})$ query time,



(a) (b)

Figure 5: Temporal complexity of the E^{2}LSH algorithm on uniform data for different selectivity values, namely $\theta = 0.1$ (red solid line), $\theta = 0.01$ (blue dashed line), and $\theta = 0.001$ (magenta dash-dotted line), and dimensions $d \in [10, 10^3]$, estimated by using $n = 10{,}000$ data points and $m = n$ query points. The plot on the left concerns the cost of reporting all the neighbors. The plot on the right concerns the cost of reporting just one neighbor. In the latter plot, dotted curves represent the complexity of sampling until a neighbor is retrieved.

and failure probability $\delta = 1/e+1/3$.[3] As for the value of the exponent ρ_ϵ, for the Euclidean distance, it is possible to achieve $\rho_\epsilon = 1/(1+\epsilon)^2 + o_\epsilon(1)$ (Andoni and Indyk, 2006), and it is known this bound is tight.

For example, consider that if $\epsilon = 0.01$, then $\rho_\epsilon = 0.980$. Because meaningfulness in intrinsically high-dimensional spaces requires smaller and smaller ϵ values, this means that, if we wish to maintain a pre-defined level of selectivity θ, we expect that the efficiency of LSH-based schemes will diminish with the intrinsic dimensionality of the space.

To empirically illustrate the relationship among selectivity θ, the intrinsic dimensionality d, and the temporal complexity γ of the search algorithm,[4] we analyzed the performances of the E^{2}LSH method as a function of the expected size θ of the r-neighborhood. The E^{2}LSH package solves the randomized r-near neighbor reporting problem exploiting the basic LSH scheme.[5] After preprocessing the data set, E^{2}LSH answers queries, typically in

3. The failure probability δ can be made arbitrarily small, say $\delta < 1/n$, by running $O(\log(n+m))$ copies of the basic LSH algorithm for *P1*, where n and m denote an upper bound on the number of points in the data structure and on the number of queries performed at any time. Moreover, *P2* can be solved by using as building blocks $O(\log n)$ copies of an algorithm for *P1*, achieving failure probability $O(\delta \log n)$ (Har-Peled et al., 2012). A similar strategy allows solving the nearest neighbor reporting problem (*P3*) by building on different data structures for *P1* associated with increasing values of r (Andoni and Indyk, 2008).
4. The *temporal complexity* is defined as the exponent $\gamma \geq 0$ such that the total number of distances D computed by the algorithm in order to report its answer is such that $D = n^\gamma$.
5. The E^{2}LSH package is available for download at `http://www.mit.edu/~andoni/LSH/`.

sub-linear time, with each near neighbor being reported with a certain probability $1 - \delta$ ($= 0.9$ by default). As for the values of the other parameters employed, we used the values determined automatically by the algorithm.

Figure 5 reports the results of the experiments on a family of uniformly distributed data sets composed of $n = 10{,}000$ points with $d \in [10, 10^3]$. We used $m = n$ different query points generated from the same distribution. We also varied the selectivity θ in $\{0.001, 0.01, 0.1\}$ by determining the radius r such that the expected fraction of r-near neighbors of the query points is θ. In Figure 5a, it can be seen that the complexity of the procedure increases with θ, and this can be explained by noting that the total number of points to be reported is directly proportional to θ. However, even if θ is held fixed, in all cases, the complexity of the algorithm for large d values tends to a linear scan of the data or to the cost γ_s of a random sampling procedure.[6] In Figure 5b, the algorithm has been enforced to report at most one near neighbor; hence, it stops the search as soon as it retrieves a near neighbor. It can now be seen that the complexity of the procedure decreases with θ, and this can be explained by noting that the probability of retrieving a neighbor is directly proportional to θ. The dotted curves represent the complexity γ_s of the procedure consisting in randomly selecting points until a r-near neighbor is retrieved. Additionally, in this case, it can be observed that the complexity degrades towards that of the random sampling procedure irrespectively of the selectivity value θ.[7]

The above analysis provides a picture of how much better an approximate search algorithm can perform than the pure random search, as a function of the selectivity and of the intrinsic dimensionality. Although the target neighborhood can be guaranteed to contain not too many points even in very large dimensional spaces, the best search algorithms may fail to perform better than random sampling. This can be explained by the poor separation of distances with the objects that are outside the approximate neighborhood.

In this regard, although the critical dimensionality is a construct with which to attempt to quantify the meaningless of a certain query, the relative contrast C_r (He et al., 2012) is a way to attempt to quantify its difficulty. Given a query point $\boldsymbol{x}_d$, the relative contrast is a measure of separability of the nearest neighbor of $\boldsymbol{x}_d$ from the rest of the data set points.

Definition 26 (Adapted from He et al., 2012) *Let DS be a data set consisting of n realizations of a random vector $\boldsymbol{Y}_d$. The relative contrast for the data set DS for a query $\boldsymbol{x}_d$, being the realization of a random vector $\boldsymbol{X}_d$, is defined as $C_r^k(\boldsymbol{x}_d) = \frac{\mathbf{E}[\mathrm{dist}(\boldsymbol{x}_d, DS)]}{\mathbf{E}[\mathrm{distnn}_k(\boldsymbol{x}_d, DS)]}$. Taking expectations with respect to queries, the relative contrast for the data set DS is*

$$C_r^k = \frac{\mathbf{E}[\mathrm{dist}(\boldsymbol{X}_d, DS)]}{\mathbf{E}[\mathrm{distnn}_k(\boldsymbol{X}_d, DS)]}.$$

He et al. (2012) provided an estimate of the relative contrast for a data set valid for independent dimensions and, moreover, provided bounds on the cost of LSH-based nearest neighbor search algorithms taking into account the relative contrast. They also noted that

6. Indeed, the expected number n^{γ_s} of points to be randomly picked in order to retrieve the $1-\delta$ fraction of the $n\theta$ data points that are r-near neighbors of the query point is $n^{\gamma_s} = n(1-\delta)$ and $\gamma_s = 1 + \log(1-\delta)/\log(n)$. E.g., for $1 - \delta = 0.9$, $\gamma_s = 0.9886$.
7. Note that, for a query having selectivity θ, the expected number of points to be randomly picked in order to retrieve exactly one r-near neighbor is $n^{\gamma_s} = 1/\theta$ and, hence, $\gamma_s = -\log(\theta)/\log(n)$.

the analysis of Beyer et al. (1999) and François et al. (2007) agree with the asymptotic behavior of the relative contrast. We refer to (He et al., 2012) for the details.

Here, we show that by exploiting the previous results, we can derive an approximation for the relative contrast C_r^k of a data set that results to be more accurate than the estimate provided by He et al. (2012). In addition, we can derive a closed form for the relative contrast $C_r^k(\boldsymbol{x}_d)$ of an individual query point.

Theorem 27 *Let $\boldsymbol{X}_d$ be a d-dimensional i.i.d. random vector with cdf F_X and let $\boldsymbol{Y}_d$ be a d-dimensional i.i.d. random vector with cdf F_Y. Assume, w.l.o.g., that F_Y has null mean $\mu_Y = 0$. Then, for large values of d,*

$$C_r^k \approx \sqrt{\frac{\sqrt{d}(\mu_{Y,2} + \mu_{X,2})}{\sqrt{d}(\mu_{Y,2} + \mu_{X,2}) + \Phi^{-1}\left(\frac{k}{n}\right)\sqrt{\mu_{Y,4} - \mu_{Y,2}^2 + 4\mu_{Y,2}\mu_{X,2} - 4\mu_{Y,3}\mu_X}}}.$$

Proof of Theorem 27. Consider the expected value of the squared distance separating a query point $\boldsymbol{X}_d$ from $nn_\varrho(\boldsymbol{X}_d, \boldsymbol{Y}_d)$ (leveraging Proposition 8 and Lemma 23):

$$\begin{aligned}
&\mathbf{E}[\|\boldsymbol{X}_d - nn_\varrho(\boldsymbol{X}_d, \boldsymbol{Y}_d)\|^2] = \mathbf{E}[\mathrm{distnn}_\varrho(\boldsymbol{X}_d, \boldsymbol{Y}_d)^2] = \mathbf{E}[\mathrm{dist}(\boldsymbol{X}_d, nn_\varrho(\boldsymbol{X}_d, \boldsymbol{Y}_d))^2] = \\
&\qquad = \int_{\mathbb{R}^d} Pr[\boldsymbol{X}_d = \boldsymbol{x}_d] \cdot \mathrm{dist}(\boldsymbol{x}_d, nn_\varrho(\boldsymbol{x}_d, \boldsymbol{Y}_d))^2 \, \mathrm{d}\boldsymbol{x}_d = \\
&\qquad = \int_0^{+\infty} \phi_{\|\boldsymbol{X}_d\|^2}(R) \cdot \left(\mu_{\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2} + \Phi^{-1}(\varrho)\, \sigma_{\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2}\right)\Big|_{\|\boldsymbol{x}_d\|^2 = R} \mathrm{d}R = \\
&= \int_0^{+\infty} \phi_{\|\boldsymbol{X}_d\|^2}(R) \cdot \left(R + d\mu_{Y,2} + \Phi^{-1}(\varrho)\sqrt{d(\mu_{Y,4} - \mu_{Y,2}^2) + 4\mu_{Y,2}R - 4\mu_{Y,3}\frac{\mu_X}{\mu_{X,2}}R}\right) \mathrm{d}R.
\end{aligned}$$

After approximating the R under the square root with the expected value $\mu_{\|\boldsymbol{X}_d\|^2} = d\mu_{X,2}$ of $\boldsymbol{X}_d$:

$$\begin{aligned}
&\mathbf{E}[\mathrm{dist}(\boldsymbol{X}_d, nn_\varrho(\boldsymbol{X}_d, \boldsymbol{Y}_d))^2] = \int_0^{+\infty} R \cdot \phi_{\|\boldsymbol{X}_d\|^2}(R)\, \mathrm{d}R + \\
&+ \left(d\mu_{Y,2} + \Phi^{-1}(\varrho)\sqrt{d(\mu_{Y,4} - \mu_{Y,2}^2) + 4d\mu_{Y,2}\mu_{X,2} - 4d\mu_{Y,3}\mu_X}\right) \cdot \int_0^{+\infty} \phi_{\|\boldsymbol{X}_d\|^2}(R)\, \mathrm{d}R = \\
&\qquad = d(\mu_{X,2} + \mu_{Y,2}) + \Phi^{-1}(\varrho)\sqrt{d(\mu_{Y,4} - \mu_{Y,2}^2 + 4\mu_{X,2}\mu_{Y,2} - 4\mu_{Y,3}\mu_X)}.
\end{aligned}$$

Indeed, the left hand integral above corresponds to the expected value $\mu_{\|\boldsymbol{X}_d\|^2} = d\mu_{X,2}$ of of the random variable $\|\boldsymbol{X}_d\|^2$, whereas the right hand integral evaluates to one.

According to the Jensen inequality (Johnson et al., 1994), if g is a concave function, then $\mathbf{E}[g(X)] \leq g(\mathbf{E}[X])$; moreover, the larger the relative variance σ_X/μ_X of X, the closer the two above values, i.e., $\mathbf{E}[g(X)] \approx g(\mathbf{E}[X])$. Specifically, $\mathbf{E}[\|\boldsymbol{X}_d\|] = \mathbf{E}[\sqrt{\sum_i X_i^2}] \leq \sqrt{\mathbf{E}[\sum_i X_i^2]} = \sqrt{\mathbf{E}[\|\boldsymbol{X}_d\|^2]}$ and, because $\sigma_{\|\boldsymbol{X}_d\|^2}/\mu_{\|\boldsymbol{X}_d\|^2} = O(d^{-1/2})$, for large values of d, $\mathbf{E}[\mathrm{dist}(\boldsymbol{X}_d, \boldsymbol{Y}_d)] \approx \sqrt{\mathbf{E}[\mathrm{dist}(\boldsymbol{X}_d, \boldsymbol{Y}_d)^2]}$, and $\mathbf{E}[\mathrm{distnn}_\varrho(\boldsymbol{X}_d, \boldsymbol{Y}_d)] \approx \sqrt{\mathbf{E}[\mathrm{distnn}_\varrho(\boldsymbol{X}_d, \boldsymbol{Y}_d)^2]}$. ■

We can also provide the relative contrast $C_r^k(\boldsymbol{x}_d)$ of an individual query point $\boldsymbol{x}_d$.

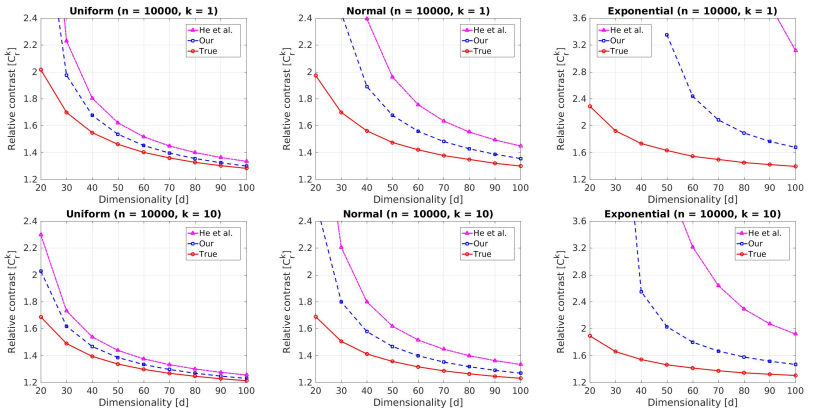


Figure 6: [Best viewed in color.] Comparison between the estimate of the relative contrast C_r provided in Theorem 27 (blue dashed line) and the estimate provided by He et al. (2012) (magenta dash-dotted line). The red solid line represents the value of the relative contrast estimated empirically.

Corollary 28 *Let $\boldsymbol{x}_d$ denote a realization of a d-dimensional i.i.d. random vector, and let $\boldsymbol{Y}_d$ be a d-dimensional i.i.d. random vector with cdf F_Y. Then, for large values of d, with high probability*

$$C_r^k(\boldsymbol{x}_d) \approx \sqrt{\frac{\|\boldsymbol{x}_d\|^2 + d\mu_2 - 2\mu\sum_{i=1}^d x_i}{\|\boldsymbol{x}_d\|^2 + d\mu_2 - 2\mu\sum_{i=1}^d x_i + \Phi^{-1}\left(\frac{k}{n}\right)\sqrt{d(\mu_4 - \mu_2^2) + 4\mu_2\|\boldsymbol{x}_d\|^2 - 4(\mu_3 - \mu\mu_2)\sum_{i=1}^d x_i}}}.$$

Proof of Corollary 28. Following the same line of reasoning of Theorem 27, $C_r^k(\boldsymbol{x}_d) \approx \sqrt{\frac{\mathbf{E}[\mathrm{dist}(\boldsymbol{x}_d, \boldsymbol{Y}_d)^2]}{\mathbf{E}[\mathrm{distnn}_\varrho(\boldsymbol{x}_d, \boldsymbol{Y}_d)^2]}}$, and the statement follows by leveraging Theorem 15 and Lemma 23. ■

Figure 6 compares the approximation of the relative contrast provided in Theorem 27 to the approximation provided by He et al. (2012). In all the cases, the approximation of Theorem 27 is the more accurate. This can be understood by noting that He et al. (2012) estimated the relative contrast by considering the differences $X_i - Y_i$ between the components of a query point $\boldsymbol{X}_d$ and of a data point $\boldsymbol{Y}_d$ as novel random variables, and then by determining their expectations and standard deviations. This corresponds to ignoring the form of the distribution of distances from each individual query point and all the data points, a relationship that is conversely taken into account in Theorem 27, due to the leveraging of Theorem 16, Proposition 8, and Lemma 23.

4.3 On the Distribution of Reverse Nearest Neighbors for i.i.d. Data

Given a real number $\varrho \in [0,1]$, a d-dimensional random vector $\boldsymbol{Y}_d$, and a realization $\boldsymbol{x}_d$ of $\boldsymbol{Y}_d$, it is said that a realization $\boldsymbol{y}_d$ of $\boldsymbol{Y}_d$ is a *ϱ reverse nearest neighbor* of $\boldsymbol{x}_d$ w.r.t. $\boldsymbol{Y}_d$ if $\boldsymbol{x}_d \in \mathrm{NN}_\varrho(\boldsymbol{y}_d, \boldsymbol{Y}_d)$.

The size $\mathrm{N}_\varrho(\boldsymbol{x}_d, \boldsymbol{Y}_d)$, or $\mathrm{N}_\varrho(\boldsymbol{x}_d)$ whenever $\boldsymbol{Y}_d$ is clear from the context, of the *ϱ reverse nearest neighborhood* of $\boldsymbol{x}_d$ w.r.t. $\boldsymbol{Y}_d$, also called *reverse ϱ nearest neighbor count* or *ϱ-occurrences*, is the fraction of realizations $\boldsymbol{y}_d$ of $\boldsymbol{Y}_d$ such that $\boldsymbol{x}_d \in \mathrm{NN}_\varrho(\boldsymbol{y}_d, \boldsymbol{Y}_d)$.

As in the previous sections, in order to deal with finite sets of n points $\{\boldsymbol{Y}_d\}_n$, the integer parameter $k = \varrho n$ ($k \in \{1, \ldots, n\}$) must be employed in place of ϱ. In such a case, we speak of *k reverse nearest neighborhood*, *reverse k nearest neighbor count*, or *k-occurrences*.

Before going into the main results, the following expression provides the probability that a realization, having norm value R, of a d-dimensional i.i.d. random vector $\boldsymbol{Y}_d$ lies at distance not greater than δ from a given d-dimensional vector $\boldsymbol{x}_d$.

Lemma 29 *Let $\boldsymbol{x}_d$ denote a realization of a d-dimensional i.i.d. random vector $\boldsymbol{X}_d$ with cdf F_X, and let $\boldsymbol{Y}_d$ be a d-dimensional i.i.d. random vector with cdf F_Y. Assume, w.l.o.g., that F_Y has null mean $\mu_Y = 0$. Then, for large values of d, with high probability*

$$Pr\left[\mathrm{dist}(\boldsymbol{x}_d, \boldsymbol{Y}_d) \leq \delta \mid \|\boldsymbol{Y}_d\| = R\right] \approx \Phi\left(\frac{\delta^2 - R^2 - \|\boldsymbol{x}_d\|^2}{2\|\boldsymbol{x}_d\|\sqrt{\mu_2}}\right),$$

where moments are relative to the constrained random vector $\boldsymbol{Y}_d$.

Proof of Lemma 29. See the appendix. ∎

The noteworthiness of the above expression lies in the fact that, by combining it with Proposition 8, it is possible in some cases to replace multi-dimensional integrations involving the full event space $\mathbb{R}^d$ with one-dimensional integrations over the domain $\mathbb{R}_0^+$ of the squared-norm values. Specifically, it is essential to the proof of the following result.

Theorem 30 *Let $\boldsymbol{x}_d$ denote a realization of a d-dimensional i.i.d. random vector $\boldsymbol{Y}_d$, with cdf F_Y having, w.l.o.g., null mean $\mu = 0$. Consider the reverse k nearest neighbor count $N_\varrho(\boldsymbol{x}_d)$ of $\boldsymbol{x}_d$ w.r.t. $\boldsymbol{Y}_d$. Then, for large values of d, with high probability*

$$\mathrm{N}_\varrho(\boldsymbol{x}_d) \approx \Phi\left(\frac{\Phi^{-1}(\varrho)\sqrt{\mu_4 + 3\mu_2^2} - z_{\boldsymbol{x}_d}\sqrt{\mu_4 - \mu_2^2}}{2\mu_2}\right).$$

Proof of Theorem 30. First of all, note that $\mathrm{N}_\varrho(\boldsymbol{x}_d) = Pr[\boldsymbol{x}_d \in \mathrm{NN}_\varrho(\boldsymbol{Y}_d)]$. Consider the probability

$$\begin{aligned}
Pr[\boldsymbol{x}_d \in \mathrm{NN}_\varrho(\boldsymbol{Y}_d)] &= \int_{\mathbb{R}^d} Pr\left[\boldsymbol{x}_d \in \mathrm{NN}_\varrho(\boldsymbol{y}_d)\right] \cdot Pr[\boldsymbol{Y}_d = \boldsymbol{y}_d] \,\mathrm{d}\boldsymbol{y}_d = \\
&= \int_{\mathbb{R}^d} Pr\left[\mathrm{dist}(\boldsymbol{x}_d, \boldsymbol{y}_d) \leq \mathrm{dist}(\boldsymbol{y}_d, nn_\varrho(\boldsymbol{y}_d))\right] \cdot Pr[\boldsymbol{Y}_d = \boldsymbol{y}_d] \,\mathrm{d}\boldsymbol{y}_d = \\
&= \int_0^{+\infty} Pr\left[\mathrm{dist}(\boldsymbol{x}_d, \boldsymbol{Y}_d) \leq \mathrm{dist}(\boldsymbol{Y}_d, nn_\varrho(\boldsymbol{Y}_d)) \mid \|\boldsymbol{Y}_d\|^2 = R\right] \cdot Pr\left[\|\boldsymbol{Y}_d\|^2 = R\right] \mathrm{d}R.
\end{aligned}$$

By Lemma 23 and Proposition 22, for $\|\boldsymbol{Y}_d\|^2 = R$,

$$\mathrm{dist}(\boldsymbol{Y}_d, nn_\varrho(\boldsymbol{Y}_d))^2 = R + d\mu_2 + \Phi^{-1}(\varrho)\sqrt{d(\mu_4 - \mu_2^2) + 4\mu_2 R},$$

while by Lemma 29,

$$\begin{aligned}
Pr\left[\mathrm{dist}(\boldsymbol{x}_d, \boldsymbol{Y}_d) \leq \mathrm{dist}(\boldsymbol{Y}_d, nn_\varrho(\boldsymbol{Y}_d)) \mid \|\boldsymbol{Y}_d\|^2 = R\right] &\approx \\
&\approx \Phi\left(\frac{\mathrm{dist}(\boldsymbol{Y}_d, nn_\varrho(\boldsymbol{Y}_d))^2 - R - \|\boldsymbol{x}_d\|^2}{2\|\boldsymbol{x}_d\|\sqrt{\mu_2}}\right) = \\
&= \Phi\left(\frac{\Phi^{-1}(\varrho)\sqrt{d(\mu_4 - \mu_2^2) + 4\mu_2 R} + d\mu_2 - \|\boldsymbol{x}_d\|^2}{2\|\boldsymbol{x}_d\|\sqrt{\mu_2}}\right),
\end{aligned}$$

from which it follows that

$$Pr[\boldsymbol{x}_d \in nn_\varrho(\boldsymbol{Y}_d)] \approx \int_0^{+\infty} \Phi\left(\frac{\Phi^{-1}(\varrho)\sqrt{d(\mu_4 - \mu_2^2) + 4\mu_2 R} + d\mu_2 - \|\boldsymbol{x}_d\|^2}{2\|\boldsymbol{x}_d\|\sqrt{\mu_2}}\right) \phi_{\|\boldsymbol{Y}_d\|^2}(R) \,\mathrm{d}R,$$

where moments are relative to the constrained random vector $\boldsymbol{Y}_d$.

The proof proceeds by expressing $\|\boldsymbol{x}_d\|^2$ and R in terms of their standard scores with respect to the random variable $\|\boldsymbol{Y}_d\|^2$, i.e.,

$$\|\boldsymbol{x}_d\|^2 = \mu_{\|\boldsymbol{Y}_d\|^2} + z_{\boldsymbol{x}_d} \cdot \sigma_{\|\boldsymbol{Y}_d\|^2} \quad \text{and} \quad R = \mu_{\|\boldsymbol{Y}_d\|^2} + z_{R,\|\boldsymbol{Y}_d\|^2} \cdot \sigma_{\|\boldsymbol{Y}_d\|^2}.$$

By substituting in the left-hand side above, and by considering that for α and β being finite and d growing, $\sqrt{\alpha d} \approx \sqrt{\alpha d + \sqrt{\beta d}}$,

$$\begin{aligned}
&\Phi\left(\frac{\Phi^{-1}(\varrho)\sqrt{d(\mu_4 - \mu_2^2) + 4d\mu_2^2 + z_{R,\|\boldsymbol{Y}_d\|^2} 4\mu_2\sqrt{d(\mu_4 - \mu_2^2)}} + d\mu_2 - d\mu_2 - z_{\boldsymbol{x}_d}\sqrt{d(\mu_4 - \mu_2^2)}}{2\sqrt{\mu_2}\sqrt{d\mu_2 + z_{\boldsymbol{x}_d}\sqrt{d(\mu_4 - \mu_2^2)}}}\right) \approx \\
&\quad \approx \Phi\left(\frac{\Phi^{-1}(\varrho)\sqrt{d(\mu_4 + 3\mu_2^2)} - z_{\boldsymbol{x}_d}\sqrt{d(\mu_4 - \mu_2^2)}}{2\mu_2\sqrt{d}}\right) = \\
&\quad = \Phi\left(\frac{\Phi^{-1}(\varrho)\sqrt{\mu_4 + 3\mu_2^2} - z_{\boldsymbol{x}_d}\sqrt{\mu_4 - \mu_2^2}}{2\mu_2}\right) = C(z_{\boldsymbol{x}_d}, \varrho).
\end{aligned}$$

Since, for large values of d, moments tend to their unconstrained values (see proof of Lemma 29), the last expression depends on $z_{\boldsymbol{x}_d}$ and on ϱ, but not on R. Thus

$$Pr[\boldsymbol{x}_d \in \mathrm{NN}_\varrho(\boldsymbol{Y}_d)] \;\approx\; C(z_{\boldsymbol{x}_d}, \varrho) \int_0^{+\infty} \phi_{\|\boldsymbol{Y}_d\|^2}(R) \,\mathrm{d}R = C(z_{\boldsymbol{x}_d}, \varrho).$$

∎

As for the expression reported in the statement of Theorem 30, it does not explicitly depend on the dimensionality and on the exact position of the point $\boldsymbol{x}_d$ but only on the relative position of the squared norm of the point with respect to the expected value. Thus, the following definition naturally arises.

Definition 31 *Let z denote the standard score of the squared norm. Then, the infinite dimensional k-occurrences function $\mathrm{N}_\varrho^\infty : \mathbb{R} \mapsto [0,1]$, defined as*

$$\mathrm{N}_\varrho^\infty(z) \quad = \quad \Phi\left(\frac{\Phi^{-1}(\varrho)\sqrt{\mu_4 + 3\mu_2^2} - z\sqrt{\mu_4 - \mu_2^2}}{2\mu_2}\right), \tag{3}$$

represents the fraction of points having a point with squared norm standard score z among their ϱ nearest neighbors.

An alternative expression can be provided by leveraging the *kurtosis* $\kappa = \frac{\mu_4}{\mu_2^2}$, a well known measure of tailedness of a probability distribution that is

$$\mathrm{N}_\varrho^\infty(z) \quad = \quad \Phi\left(\Phi^{-1}(\varrho)\frac{\sqrt{\kappa+3}}{2} - z\frac{\sqrt{\kappa-1}}{2}\right). \tag{4}$$

In particular, it holds from the development of Theorem 30 that

$$\lim_{d\to\infty} Pr[\mathrm{N}_\varrho(\boldsymbol{X}_d) = \mathrm{N}_\varrho^\infty(z)] = \phi(z), \tag{5}$$

from which it can be seen that the point $z_0 \to -\infty$ is such that $\mathrm{N}_\varrho^\infty(z_0) \to 1$ and $\phi(z_0) \to 0$. This point precisely corresponds with the expected value of $\boldsymbol{X}_d$ (the origin of the space for variables with null mean) and is the point most likely to be selected as nearest neighbor by any other point. At the same time, it is the point least likely to be observed as a realization of $\boldsymbol{X}_d$ among those having norms smaller than the expected value.

As for the point $z_\infty \to \infty$, it is such that $\mathrm{N}_\varrho^\infty(z_\infty) \to 0$ and $\phi(z_\infty) \to 0$. Hence, it is the less likely to be observed as a realization of $\boldsymbol{X}_d$, but it is also the less likely to be selected as a nearest neighbor. This point is the furthest from the mean, and it is located on the boundary of a bounded region or ad infinitum.

Figure 7 shows the curves of $\mathrm{N}_\varrho^\infty$ (red lines) for i.i.d. data coming from different distributions, together with the empirical N_k/n values (black dots), with $k = \varrho n$, in a random sample of $n = 10{,}000$ points. Theoretical $\mathrm{N}_\varrho^\infty$ curves represent the picture of what happens ad infinitum, because they provide the values to which the k-occurrences counts converge for large dimensions. We observed that in most cases, the convergence arises very soon, often a few tens of dimensions suffice. Indeed, empirical values tend to distribute along the associated curves. While the first two distributions have null skewness, the same behavior is exhibited by the third one, which instead has non-null skewness, even if, in this case, convergence appears to be slower. In any case, it appears that the value of k-occurrences predicted by means of the function $\mathrm{N}_\varrho^\infty$ usually is in good agreement with the empirical evidence, even for the smallest dimension considered in the figure. For similar plots on real data sets, we refer to Figures 10, 11, and 12, reported in the following.

It is now of interest to obtain the cdf and pdf of $\mathrm{N}_\varrho(\boldsymbol{X}_d)$ for large values, together with the associated variance and expected value.

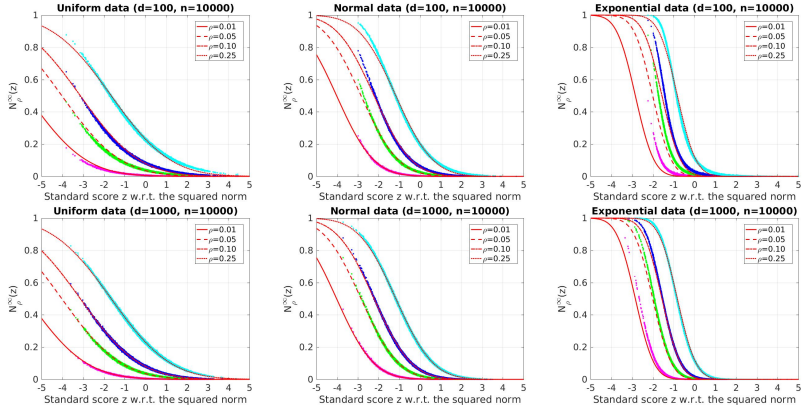


Figure 7: [Best viewed in color.] Comparison between the empirical values of the relative number of k-occurrences (N_k/n), estimated in a random sample of $n = 10{,}000$ points with $d \in \{100, 1{,}000\}$, and the values predicted by means of the function $\mathrm{N}_\varrho^\infty$ (red curves), for $\varrho = 0.01$ (magenta dots and solid red line), $\varrho = 0.01$ (green dots and dashed red line), $\varrho = 0.01$ (blue dots and dash-dotted red line), and $\varrho = 0.01$ (cyan dots and dotted red line).

Theorem 32

$$(i) \quad \lim_{d\to\infty} Pr[\mathrm{N}_\varrho(\boldsymbol{X}_d) \le \theta] = \Phi\left(\frac{\Phi^{-1}(\theta)2\mu_2 - \Phi^{-1}(\varrho)\sqrt{\mu_4 + 3\mu_2^2}}{\sqrt{\mu_4 - \mu_2^2}}\right),$$

$$(ii) \quad \lim_{d\to\infty} Pr[\mathrm{N}_\varrho(\boldsymbol{X}_d) = \theta] = \frac{2\mu_2}{\sqrt{\mu_4 - \mu_2^2}} \cdot \frac{1}{\phi(\Phi^{-1}(\theta))} \cdot \phi\left(\frac{\Phi^{-1}(\theta)2\mu_2 - \Phi^{-1}(\varrho)\sqrt{\mu_4 + 3\mu_2^2}}{\sqrt{\mu_4 - \mu_2^2}}\right),$$

$$(iii) \quad \lim_{d\to\infty} \sigma^2(\mathrm{N}_\varrho(\boldsymbol{X}_d)) = \varrho(1-\varrho) - 2T\left(\Phi^{-1}(\varrho), \frac{2\mu_2}{\sqrt{2(\mu_4 + \mu_2^2)}}\right), \text{ and}$$

$$(iv) \quad \lim_{d\to\infty} \mathbf{E}[\mathrm{N}_\varrho(\boldsymbol{X}_d)] = \varrho,$$

where $T(h,a) = \phi(h)\int_0^a \frac{\phi(hx)}{1+x^2}\,\mathrm{d}x$ *is the Owen's T function.*

Proof of Theorem 32. See the appendix. ■

Figure 8 shows the cdfs and the pdfs of the limiting distributions of the k-occurrences for uniform, normal, and exponential distributions. As expected, the probability of observing large values of N_ϱ increases with ϱ. Moreover, it can be observed from the pdf functions that for the exponential distribution, the probability of observing $\mathrm{N}_\varrho \approx 1$ is not negligible

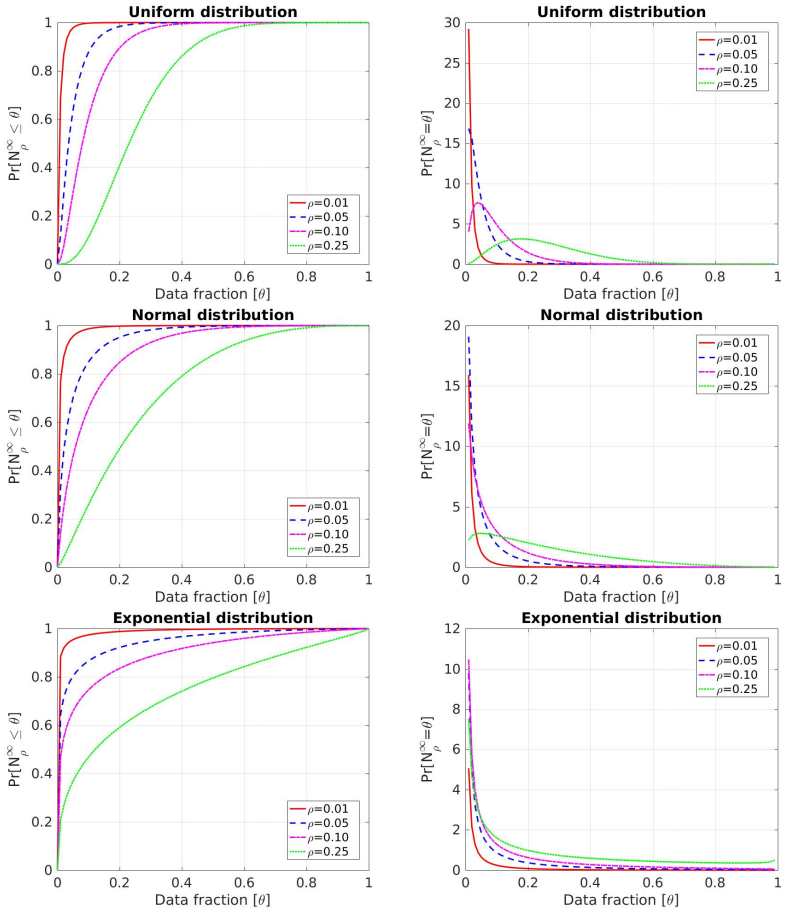


Figure 8: [Best viewed in color.] Cumulative distribution function (left column) and probability density function (right column) of the limiting distribution of the number of k-occurrences for i.i.d. random vectors (see Th. 32) according to different families of distributions, for $\varrho = 0.01$ (red solid line), $\varrho = 0.05$ (blue dashed line), $\varrho = 0.10$ (magenta dash-dotted line), and $\varrho = 0.25$ (green dotted line).

even for moderately large values of ϱ. This behavior can be better understood by looking at Figure 7, where theoretical curves of the exponential are approaching 1 earlier.

Corollary 33 *Let $\boldsymbol{X}_d$ and $\boldsymbol{Y}_d$ be two d-dimensional i.i.d. random vectors with common cdf F_Y having, w.l.o.g., null mean $\mu_Y = 0$. Then, for large values of d,*

$$Pr[\mathrm{N}_k(\boldsymbol{X}_d, \{\boldsymbol{Y}_d\}_n) \leq h] \approx \Phi\left(\frac{\Phi^{-1}(\frac{h}{n})2\mu_2 - \Phi^{-1}(\frac{k}{n})\sqrt{\mu_4 + 3\mu_2^2}}{\sqrt{\mu_4 - \mu_2^2}}\right).$$

Proof of Corollary 33. The statement follows immediately from Theorem 32. ■

In order to compare the solution here derived to the large dimensional limits of the function $\mathrm{N}_k^{n,d}$ provided by Newman et al. (1983), the same limits are derived next.

Corollary 34 *Let k be a fixed positive integer. Then*

$$(i)\ \lim_{n\to\infty}\lim_{d\to\infty} \mathrm{N}_k^{n,d} \xrightarrow{D} 0,\quad (ii)\ \lim_{n\to\infty}\lim_{d\to\infty} \sigma^2(\mathrm{N}_k^{n,d}) = \infty,\ \textit{and}\ (iii)\ \lim_{n\to\infty}\lim_{d\to\infty} \mathbf{E}[\mathrm{N}_k^{n,d}] = k.$$

Proof of Corollary 34. See the appendix. ■

Results provided by Newman et al. (1983), correspond to points (i) and (ii) above for the case $k = 1$. The point (iii) is reported only as a check, because k is expected by definition of the k-occurrences.

The interpretation of the above result provided by Tversky et al. (1983), which is typically how it is reported in the related literature, is that if the number of dimensions is large relative to the number of points, one may expect to have a large proportion of points with reverse nearest neighbor counts equaling 0, and a small proportion of points with high count values. However, according to the previous findings, the convergence in distribution to zero does not have to be motivated by the excess of the dimensions with respect to the sample size, but rather by the use of a fixed-size neighborhood parameter k in the presence of large samples. As a matter of fact, the curves reported in Figure 8 tend to the zero distribution only for $\varrho = k/n \to 0$. Moreover, large counts can also be achieved in the case of small samples and large dimensionalities, as shown in Figures 7 and 8. E.g., from Equation (5), the expected number of points such that $z \leq -1$ ($z \leq -2$, resp.) is about the 15.9% (2.3%, resp.) for any sample size n. All of this suggests that hubness is definitely not an artifact of a finite sample.

4.4 The Distribution of Independent Non-Identically Distributed Data

Previous results can be extended to independent non-identically distributed random vectors. With this aim, we consider the following proposition.

Given a sequence $W_1, W_2, \ldots, W_d$ of independent non-identically distributed random variables having non-null variances [8] and finite central moments $\hat{\mu}_{i,k}$ up to the eighth mo-

8. Clearly, variables having constant domain, hence null variance, can be disregarded because they do not alter distances.

ment, we say that the sequence has *comparable* central moments if there exist positive constants $\hat{\mu}_{\max} \geq \max_{i,k}\{|\hat{\mu}_{i,k}|\}$ and $\hat{\mu}_{\min} \leq \min_i\{|\hat{\mu}_{i,k}| : \hat{\mu}_{i,k} \neq 0\}$. Intuitively, this guarantees that the ratio between the greatest and the smallest non-null moment remains limited.[9]

Proposition 35 *Let $U_d = \sum_{i=1}^{d} W_i$ be a random variable defined as the summation of a sequence of independent, but not identically distributed, random variables W_i having comparable central moments. Then*

$$U_d \simeq \mathcal{N}\left(\sum_{i=1}^{d}\mu_{W_i},\ \sum_{i=1}^{d}\sigma^2_{W_i}\right) = \mathcal{N}\left(d\cdot\overline{\mu}_W,\ d\cdot\overline{\sigma^2}_W\right),$$

where $\overline{\mu}_W = (1/d)\sum_{i=1}^{d}\mu_{W_i}$ and $\overline{\sigma^2}_W = (1/d)\sum_{i=1}^{d}\sigma^2_{W_i}$.

Proof of Proposition 35. See the appendix. ■

François et al. (2007, cf. Proposition 2) affirmed that Theorem 3 holds for independent non-identically distributed variables provided that they are normalized. Authors justify this result by noting that norms will concentrate because normalization prevents variables from having too little effect on the distance values. According to this interpretation, normalization is essential for having comparable variances. (Recall that the variance is the second order central moment.)

Definition 36 *Let $\boldsymbol{Y}_d = (Y_1, Y_2, \ldots, Y_d)$ be an independent non-identically distributed d-dimensional random vector with cdfs $\boldsymbol{F}_Y = (F_{Y_1}, F_{Y_2}, \ldots, F_{Y_d})$ having k-th moments $\boldsymbol{\mu_k} = (\mu_{Y_{1,k}}, \mu_{Y_{2,k}}, \ldots, \mu_{Y_{d,k}}) = (\mu_{1,k}, \mu_{2,k}, \ldots, \mu_{d,k})$. Moreover, given a positive integer $h, k \geq 1$, let $\tilde{\boldsymbol{\mu}}^{\boldsymbol{h}}_{\boldsymbol{k}}$ denote the average h-th degree of the k-th central moments of $\boldsymbol{Y}_d$, also referred to as average central moment for simplicity, defined as*

$$\tilde{\boldsymbol{\mu}}^{\boldsymbol{h}}_{\boldsymbol{k}} = \frac{1}{d}\sum_{i=1}^{d}\hat{\mu}^h_{i,k} = \frac{1}{d}\sum_{i=1}^{d}\mathbf{E}[(Y_i - \mu_{Y_i})^k]^h,$$

where $\hat{\mu}_{i,k}$ denotes the k-th central moment of Y_i.

Because considering random variables having null means simplifies expressions, for the sake of simplicity, we next consider the case of independent non-identically distributed random vectors having common cdfs, but we note that a similar result also holds in the more general case $\boldsymbol{F_X} \neq \boldsymbol{F_Y}$.

Theorem 37 *Let $\boldsymbol{X}_d$ and $\boldsymbol{Y}_d$ be two independent non-identically distributed d-dimensional random vectors with common cdfs $\boldsymbol{F}$ having means $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_d)$, non null variances, and comparable central moments, and let $\boldsymbol{x}_d$ denote a realization of $\boldsymbol{X}_d$. The results of Sections 4.1, 4.2 and 4.3 can be applied to $\boldsymbol{X}_d$, $\boldsymbol{Y}_d$, and $\boldsymbol{x}_d$ by taking into account the average central moments of $\boldsymbol{X}_d$ and $\boldsymbol{Y}_d$ and the realization $\boldsymbol{x}_d - \boldsymbol{\mu}$.*

Proof of Theorem 37. See the appendix for details. ■

To illustrate the above results, real data sets having dimensionality varying at some order of magnitude are considered. The data sets are briefly described next. The *Statlog* (Landsat Satellite) data set[10] consists of multi-spectral values of pixels in 3×3 neighborhoods in a satellite image ($d = 36$, $n = 6{,}435$). The SIFT data set[11] consists of the base vectors of the `ANN_SIFT10` evaluation set used to evaluate the quality of approximate nearest neighbors search algorithms and consists of SIFT image descriptors ($d = 128$, $n = 10{,}000$). The MNIST data set[12] consists of handwritten digits which have been size-normalized and centered in a 28×28 image. The test examples have been employed ($d = 784$, $n = 10{,}000$). The *Sports* data set[13] consists of time series representing sensor measurements associated with activities performed by eight subjects for 5 minutes ($d = 5{,}625$, $n = 9{,}120$). The *NIPS* textual data set[14] consists of counts associated with words appearing in 5,812 NIPS conference papers published between 1987 and 2015 ($d = 11{,}463$, $n = 5{,}812$). The *RNA-Seq* data set[15] consists of gene expressions levels, measured by a illumina HiSeq sequencing platform, of patients having different types of tumors ($d = 20{,}531$, $n = 801$).

In the following, we also consider the *shuffled* version of the original data set. Specifically, the shuffled version of a given data set is obtained by randomly permuting the elements within every attribute. As already noted by François et al. (2007), the shuffled data set is marginally distributed as the original one, but because all the relationships between variables are destroyed, its components are independent, and its intrinsic dimension is equal to its embedding dimension.

Figure 9 reports the empirical cdf of the squared distance (solid line) associated with each data set, together with the theoretical cdf (dashed line) associated with independent but not identically data having the same average central moments of the original data, as reported in Theorem 37. The latter curve has been obtained by using the average central moments of the data according to Theorem 37. The empirical cdf of the shuffled data is also reported (dotted line).

From the linearity of the expected value, for any pair of d-dimensional random vectors, it follows that

$$\mathbf{E}[\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|^2] = d(\tilde{\boldsymbol{\mu}}_{\boldsymbol{X},\boldsymbol{2}} + \tilde{\boldsymbol{\mu}}_{\boldsymbol{Y},\boldsymbol{2}}) \quad \text{and} \quad \mathbf{E}[\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2] = \|\boldsymbol{x}_d - \boldsymbol{\mu}_{\boldsymbol{Y}}\|^2 + d\tilde{\boldsymbol{\mu}}_{\boldsymbol{Y},\boldsymbol{2}},$$

where the equality holds also for dependent and non-identically distributed random vectors. Hence, the expected value of the pairwise distances between data set points is identical to

9. This definition fits the Lyapunov condition. In general, given a sequence $W_1, W_2, \ldots, W_d$ of independent non-identically distributed random variables having non-null finite variances, then their standardized sum converges in distribution to a standard normal random variable if and only if the Feller-Lindeberg condition holds (Feller, 1971). According to this condition, the variance $\sigma(W_i)$ of any individual term never dominates their sum s_d; hence, $\lim_{d\to\infty}\max_{i=1}^{d}\sigma^2_i(W_i)/s^2_d = 0$. Because this both necessary and sufficient for the CLT to hold, the Feller-Lindeberg condition implies the Lyapunov condition.

10. Data available at `https://archive.ics.uci.edu/ml/datasets/Statlog+%28Landsat+Satellite%29`.
11. Data available at `http://corpus-texmex.irisa.fr/`.
12. Data available at `http://yann.lecun.com/exdb/mnist/`.
13. Data available at `https://archive.ics.uci.edu/ml/datasets/Daily+and+Sports+Activities`.
14. Data available at `https://archive.ics.uci.edu/ml/datasets/NIPS+Conference+Papers+1987-2015`.
15. Data available at `https://archive.ics.uci.edu/ml/datasets/gene+expression+cancer+RNA-Seq`.

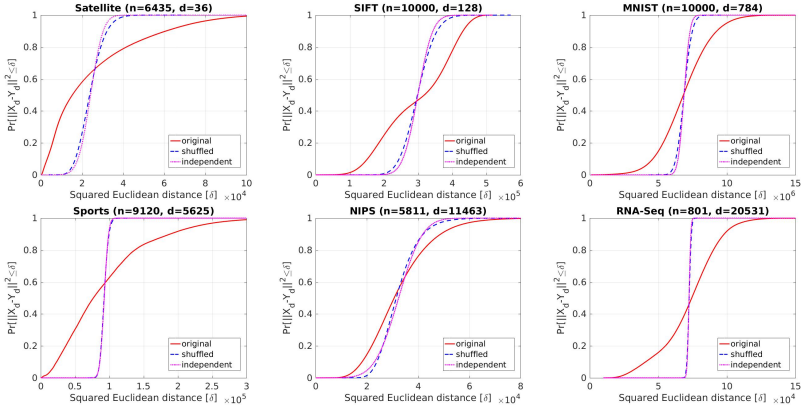


Figure 9: [Best viewed in color.] Pairwise distance distributions for real data sets, including the original data (red solid line), the shuffled data (blue dashed line), and the equivalent independent data (magenta dotted line). The last curve corresponds to the theoretical cdf associated with independent but not identically data having the same average central moments of the original data, as reported in Theorem 37.

the expected value of the theoretical distribution derived under the i.i.d. hypothesis, and also to that of the shuffled data.

The difference between the curves of the original and of the independent data suggests that the intrinsic dimensionality of the data at hand is smaller than that of the embedding space, because dependencies evidently exist between the attributes. Moreover, it can be seen that the empirical cdf of the shuffled data is very similar to that of the theoretical cdf.

These result confirm the correctness of Theorem 37, whose prediction agrees with the empirical observation on real independent data. Moreover, its approximation is accurate even in moderately large spaces, because the correspondence is good even for the smallest data set considered ($d = 36$). Moreover, these experiments testify to the meaningfulness of the analysis here accomplished as a worst-case analysis scenario, corresponding to the case in which relationships between variables are absent.

In order to verify if the data sets satisfy the requirements for the CLT to be applied, the value of the finite Lyapunov CLT condition (see Equation 1, for $\delta = 2$) has been determined on the data at hand (with the variable W_i taking value over all the terms $(x_{j,i} - x_{k,i})^2$ that can be formed with distinct pairs of data set points x_j and x_k, $1 \leq j < k \leq n$). Table 1 reports the value of the above condition (indicated as LC) together with the Relative Variance (RV) of the norm of data set points, for both the original data set (note that the shuffled data presents the same LC value as the original one) and its normalized version (obtained by substituting each attribute X_i with $(X_i - \mu_i)/\sigma_i$):

Data set	Original		Normalized	
	LC	RV	LC	RV
Satellite	0.012675	0.463934	0.014629	0.427250
SIFT	0.003382	0.063242	0.049320	0.090368
MNIST	0.000403	0.133234	25.163236	0.502937
Sports	0.072869	0.432149	0.546147	0.361839
NIPS	0.745970	0.257738	0.307322	0.241819
RNA-Seq	0.000412	0.131759	0.096444	0.175561

Table 1: Comparison between the values of the Lyapunov CLT Condition (LC) and the Relative Variance (RV) against those of the real data sets.

A small LC value (say, less than 1) indicates that the normal approximation is correct. This condition is met for all the configurations except for the normalized *MNIST* data set. Indeed, normalizing this data set is not meaningful, because attributes (corresponding to pixels) are already homogeneous (their domain consists of 256 gray levels encoded as byte values) and because the normalization has only the negative effect of exaggerating the range of variation of pixels whose domain is formed almost entirely of zeros. As a result, a few attributes dominate the distance, and this deteriorates convergence to normality. The relative variance has been reported for comparison, because it is a measure of the concentration of the data. (Note that the relative variance of the shuffled data is not coincident with that of the original one.)

Figure 10 reports the relative number of k-occurrences associated with data set points represented in terms of their squared norm standard score z. Different values for the parameter ϱ have been employed, Specifically, $\varrho \in \{0.01, 0.05, 0.10, 0.25\}$ (the color of points in the figure is magenta for $\varrho = 0.01$, green for $\varrho = 0.05$, blue for $\varrho = 0.10$ and cyan for $\varrho = 0.25$). The theoretical curves of $\mathrm{N}_\varrho^\infty(z)$, based on average central moments of the data, are also reported for comparison.

Interestingly, the real distributions of k-occurrences follow the trends of the theoretical curves; however, in contrast to the independent case, counts have much larger variability. The interpretation is that variability is associated with a lower intrinsic dimensionality and dependencies among variables, because independent data are much more widely distributed along the theoretical trend, as can be seen in Figure 7. Indeed, such behavior is also observed when considering the shuffled data set (see Figure 11). Moreover, in this case, the empirical evidence matches the behavior predicted by Theorem 37 for the independent case. In some cases, the trend appears to be different albeit generally analogous. It appears that the degree of agreement between the empirical evidence and the the theoretical prediction is directly proportional to the value of the LC condition reported in Table 1.

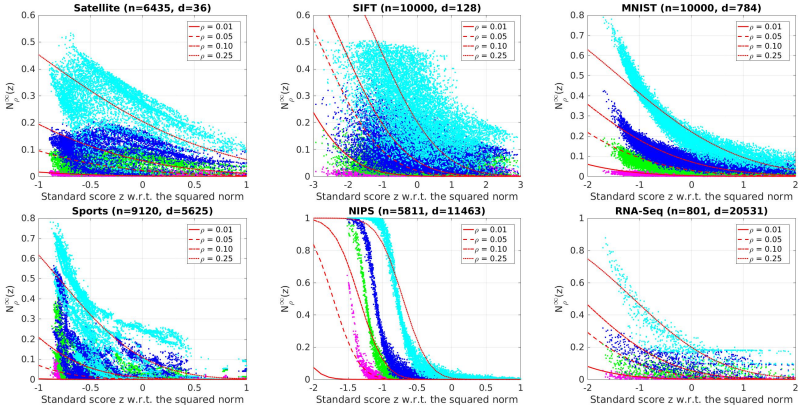


Figure 10: [Best viewed in color.] Relative number of k-occurrences associated with data set points, represented in terms of their squared norm standard score z (different colors), and the theoretical prediction according to the infinite dimensional k-occurrences function reported in Equation (4.3) (red lines), for the following values of $\varrho = k/n$: $\varrho_1 = 0.01$ (magenta-colored points and solid red line), $\varrho_2 = 0.05$ (green-colored points and dashed red line), $\varrho_3 = 0.10$ (blue-colored points and dash-dotted red line), and $\varrho_4 = 0.25$ (cyan-colored points and dotted red line).

4.5 Extension to Other Distances

In general, one may attempt to extend some of the properties discussed above to distances having the general form

$$\text{dist}(\boldsymbol{x}_d, \boldsymbol{y}_d) = h\left(\sum_{i=1}^{d} g(x_i, y_i)\right), \tag{6}$$

with $g : \mathbb{R}^2 \mapsto \mathbb{R}$ being commutative and not identically constant, and $h : \mathbb{R} \mapsto \mathbb{R}$ strictly monotonic and, hence, invertible. Indeed, let $\boldsymbol{X}_d$ and $\boldsymbol{Y}_d$ be d-dimensional i.i.d. random vectors, and consider the random variable

$$h^{-1}\left(\text{dist}(\boldsymbol{X}_d, \boldsymbol{Y}_d)\right) = \sum_{i=1}^{d} g(X_i, Y_i) = \sum_{i=1}^{d} W_i.$$

Because $W_1, W_2, W_3, \ldots$ is a sequence of i.i.d. random variables, by the CLT, it can be said that $h^{-1}\left(\text{dist}(\boldsymbol{X}_d, \boldsymbol{Y}_d)\right) \simeq \Phi\big(d \cdot \mathbf{E}[g(X_i, Y_i)],\ d \cdot \sigma^2(g(X_i, Y_i))\big)$ and, for large values of d,

$$Pr\left[\text{dist}(\boldsymbol{X}_d, \boldsymbol{Y}_d) \leq \delta\right] \approx \Phi\left(\frac{h(\delta) - d \cdot \mathbf{E}[g(X_i, Y_i)]}{\sqrt{d} \cdot \sigma(g(X_i, Y_i))}\right).$$

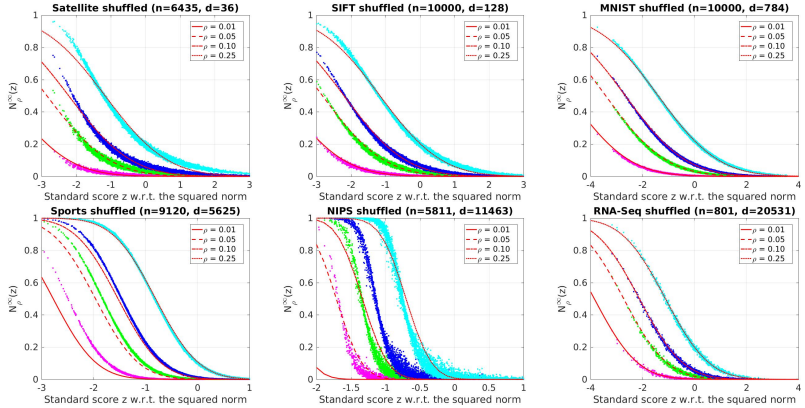


Figure 11: [Best viewed in color.] Relative number of k-occurrences associated with data set points (the shuffled version of each data set here being considered, which is obtained by randomly permuting the elements within every attribute), represented in terms of their squared norm standard score z (different colors), and theoretical prediction according to the infinite dimensional k-occurrences function reported in Equation (4.3), for the following values of $\varrho = k/n$: $\varrho_1 = 0.01$ (magenta-colored points and solid red line), $\varrho_2 = 0.05$ (green-colored points and dashed red line), $\varrho_3 = 0.10$ (blue-colored points and dash-dotted red line), and $\varrho_4 = 0.25$ (cyan-colored points and dotted red line).

As an example, consider the Minkowski norm L_p, $\|\boldsymbol{x}_d\|_p = \left(\sum_{i=1}^{d} |x_i|^p\right)^{1/p}$, with p a positive integer. Let, for the sake of simplicity, p be even, then

$$\begin{aligned}
\mathbf{E}[\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|_p^p] &= d\sum_{j=0}^{p}(-1)^j \binom{p}{j} \mu_{X,p-j}\mu_{Y,j}, \text{ and} \\
\sigma^2(\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|_p^p) &= d\sum_{j=0}^{p} \binom{p}{j}^2 \sigma^2(X_i^{p-j}Y_i^j) + \\
&\quad + d\sum_{j=0}^{p}\sum_{j\neq k=0}^{p} (-1)^{j+k}\binom{p}{j}\binom{p}{k} cov(X_i^{p-j}Y_i^j, X_i^{p-k}Y_i^k).
\end{aligned}$$

Newman and Rinott (1985) reported a generalized version of Theorem 4, in which distances of the form used in Equation (6) are considered.

Theorem 38 (Adapted from Newman and Rinott, 1985, cf. Theorem 3) *Consider the generalized distance function reported in Equation* (6)*. Let* $\beta_g = corr\big(g(X,Y), g(X,Z)\big)$ *be the correlation between* $g(X,Y)$ *and* $g(X,Z)$*, where* X*,* Y*,* Z *are i.i.d. random variables*

with common distribution F, and let $0 < \sigma^2(g(X,Y)) < \infty$. If $\beta_g > 0$, then Theorem 4 holds even if the generalized distance is employed instead of the Euclidean distance.

Both the Euclidean distance and all Minkowski's metrics with $p \neq 0$ respect the condition $\beta_g > 0$. This condition implies that the location of vector components plays a role when computing pairwise distances, because when it holds, the closer the coordinate value x_i of $\boldsymbol{x}_d$ to the the expected position of X_i, the more likely it is that the vector $\boldsymbol{x}_d$ will be close to the other realizations of the same random vector.

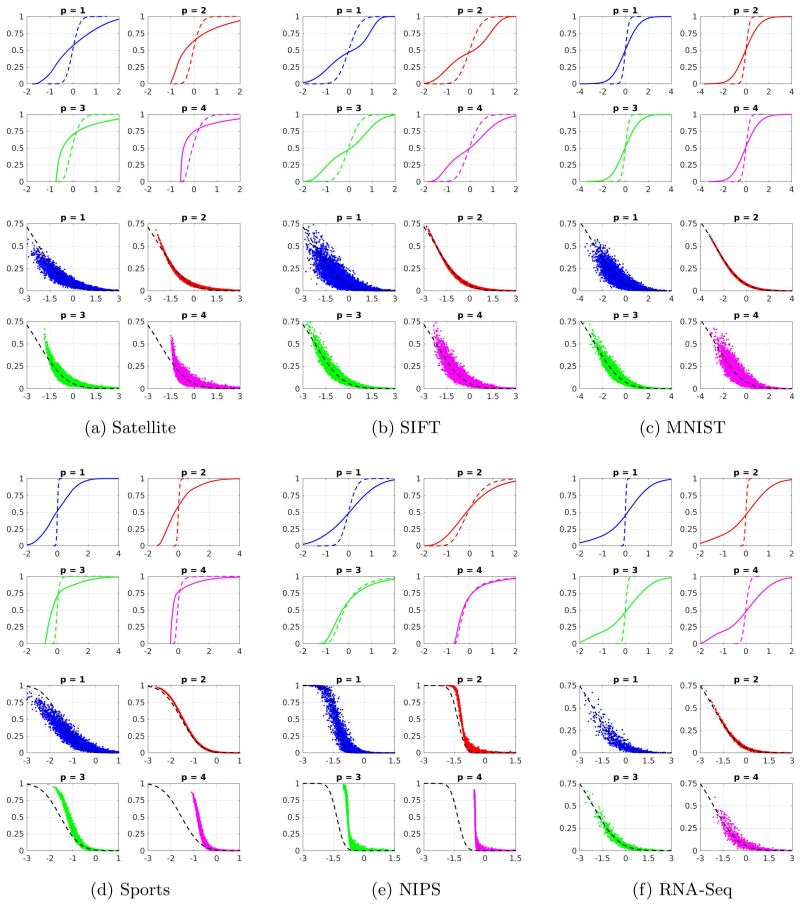
In contrast, the case $\beta_g = 0$ means that no vector occupies a special position. This condition is valid, for example, for Poisson process, which spread the vectors uniformly over $\mathbb{R}^d$. In this case, all positions within the space become equivalent and, hence, no concentration of distances is exhibited. As already noted by Radovanovic et al. (2010), the absence of spatial centrality can be intuitively used to explain the absence of hubness for cosine distance, since in this setting, no vector is more spatially central than any other, and the observation of the emergence of hubness for distance measures such as the L_p norm, Bray-Curtis, normalized Euclidean, and Canberra.

Because Theorems 16, 25 and 30, as well as related ones, are based on spatial centrality in that (the standard score of) the squared norm plays a special role in their formulation, it is conceivable that by following the same line here presented, similarly behaving closed forms can be obtained for any other distance presenting spatial centrality (namely, such that $\beta_g > 0$), expressed in terms of the standard score (or other degree) of some measure $\ell(\boldsymbol{x}_d)$ of centrality of $\boldsymbol{x}_d$. For example, for L_p norms, the natural measure of centrality is $\|\boldsymbol{x}_d\|_p^p$.

Figure 12 reports the distribution of pairwise distances and the relative number of k-occurrences for different Minkowski's metrics p. Namely, $p \in \{1,2,3,4\}$ (colors employed are blue for $p=1$, red for $p=2$, green for $p=3$, and magenta for $p=4$), on the real data sets considered in the previous section.

To facilitate the comparison of results involving different metrics, both distance values and norm values have been normalized. Specifically, let $\boldsymbol{X}_d$ and $\boldsymbol{Y}_d$ be two independent and not identically distributed random vectors whose components X_i and Y_i have the same central moments of the i-th attribute of the data set, and let $\boldsymbol{x}_d$ and $\boldsymbol{y}_d$ be two data set points. As for pairwise distance distributions, on the x-axis, the value $z_{dist}(\boldsymbol{x}_d, \boldsymbol{y}_d) = \frac{\|\boldsymbol{x}_d - \boldsymbol{y}_d\|_p^p - \mathbf{E}[\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|_p^p]}{\sigma(\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|_p^p)}$ is reported, whereas for the relative number of k-occurrences, the value $z_{norm}(\boldsymbol{x}_d) = \frac{\|\boldsymbol{x}_d\|_p^p - \mathbf{E}[\|\boldsymbol{X}_d\|_p^p]}{\sigma(\|\boldsymbol{X}_d\|_p^p)}$ is reported on the x-axis.

In Figure 12, plots concerning the pairwise distance distribution (that are, for each data set, the four plots on the top), report the cdf associated with the original data set (solid line) and the cdf associated with the shuffled data (dashed line). The curve of the cdf associated with the equivalent independent data—that is, the cdf of the normal distribution having mean $\mathbf{E}[\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|_p^p]$ and standard deviation $\sigma(\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|_p^p)$—is not reported for clarity, because its curve overlaps with that of the shuffled data. In Figure 12, plots concerning k-occurrences (which are, for each data set, the four plots on the bottom) report the relative number of k-occurrences (for $k = \varrho n$ and $\varrho = 0.1$) associated with the points of the shuffled data set (color varying with p) together with the value $\mathrm{N}_k^\infty(z_{norm})$ of the infinite dimensional k-occurrences function reported in Equation (4.3) evaluated in $z_{norm}(\boldsymbol{x}_d)$ (black dashed



(a) Satellite (b) SIFT (c) MNIST

(d) Sports (e) NIPS (f) RNA-Seq

Figure 12: [Best viewed in color.] Experimental results on real data for different Minkowski's metrics p (blue for $p=1$, red for $p=2$, green for $p=3$, and magenta for $p=4$). For each data set, the 4 plots on the top show the cdfs of pairwise distances for the original (solid line) and shuffled data (dashed line). Moreover, for each data set, the 4 plots on the bottom show the relative number of k-occurrences ($k = \varrho n$ with $\varrho = 0.1$) for the points of the shuffled data set (colored dots) and the value of the function $\mathrm{N}_\varrho^\infty$ (black dashed line) reported in Equation (4.3). The values on the x-axis are standard scores using $\|\cdot\|_p^p$ as measure of centrality.

line). Interestingly, as previously hypothesized, by representing the data in terms of the standard score of the measure of centrality $\|\cdot\|_p^p$, a similar behavior can also be observed for different Minkowski's metrics p. In general, the results for $p = 1$ are very similar to those for $p = 2$, whereas for $p > 2$, it seems that the degree of agreement is related to the value of the LC condition reported in the first column of Table 1.

5. Relationship with Hubness in Network Science

Because hubness is a phenomenon of primary importance in network science, we wondered if the findings relative to the distribution of the reverse nearest neighbors and the emergence of hubs in intrinsically high-dimensions have connections with the analogous phenomenon occurring in the context of networks.

It is well understood that complex networks (Barabási and Pósfai, 2016) arise from different natural and human-made systems, e.g., the Internet, the world-wide web, citation networks and some social networks. These networks exhibit as a major property a few nodes, called *hubs*, with unusually high degree as compared to the other nodes of the network.

In most cases, it has been observed that networks are approximatively scale-free that is, they have approximate power-law degree distributions. Specifically, in most cases, the approximation is true only for the tails of the node degree distribution. Tails are associated with the larger node degrees, which are also the less probable observations, whereas most of the probability mass is associated with the smallest degree values.

Early well-known random graph models, such as the Erdős and Rényi (1959) model, do not exhibit power laws. Thus, other models for generating scale-free networks have been proposed. The Bianconi and Barabási (2001) model generates scale-free networks based on three important concepts that have been observed in real networks: growth, preferential attachment, and fitness. Preferential attachment means that the more connected a node is, the more likely it is to receive new links. Fitness is an intrinsic value associated with each node, defined as the ability to attract new links.

At least two analogies between the study herein conducted and what was depicted above can be identified by regarding nodes as point in a high-dimensional space.

First, in some cases, the behavior of the theoretical pdf of the function N_k^∞ exhibits a transition (on the basis of the value $\varrho = k/n$) between the two aforementioned families of networks (see Theorem 32 (*ii*) and the bottom left plot in Figure 8 for uniform data); namely, the behavior is binomial-like for high ϱ values and skewed to the right with the emergence of hubs for small ϱ values.

Second, the squared norm standard score z for points can be conceptualized as a value of fitness that is assigned to nodes/points according to a certain probability ($\phi(z)$, that is the standard Normal pdf, in the case of points); consequently, $\mathrm{N}_k^\infty(z)$ represents the relative expected number of times that a certain node/point with fitness z will be referred by any other node of the network/data set. The more central the node (the closer the point to the mean), the higher its fitness and the higher is its probability of being selected as a neighbor by the rest of the points.

To ascertain whether the above analogies adhere to empirical evidence, we examined the node (in-)degree distribution of real networks. Given a directed network, consisting of

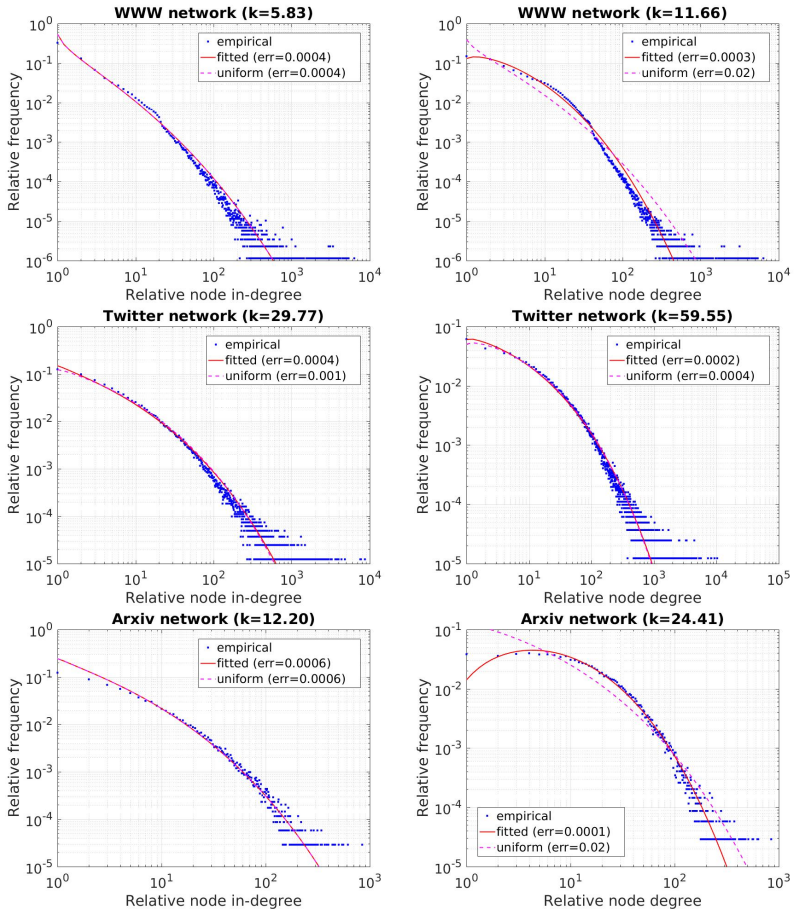


Figure 13: [Best viewed in color.] Relative node (in-)degrees (horizontal axis) and associated relative frequency (vertical axis) in log-log scale for different real-life complex networks (blue dots). Lines are associated with the probability of observing a point which has a certain number of reverse k-nearest neighbors in a set of n other points. Here, the average (in-)degree number is used as value for k. The magenta dashed line is associated with a uniform distribution in $[-0.5, +0.5]$. The red solid line is obtained using the moments minimizing the Cramér-von Mises distance between the theoretical and empirical cdfs.

n nodes and m arcs, let e_i (e_i^{in}, resp.) denote the number of arcs connected to (coming into, resp.) node n_i ($1 \leq i \leq n$).

Let $F_k(h)$ denote the discrete version of the cdf associated with the infinite dimensional k-occurrences function N_k^∞ (see Corollary 33). Then, $p_k(h) = F_k(h) - F_k(h-1)$ denotes the probability of observing a point which has exactly h reverse k-nearest neighbors in a set of n other points. As for the value of k, we used the average number of (incoming) arcs, that is $\overline{k} = \frac{1}{n}\sum_{i=1} e_i$ and $\overline{k}^{in} = \frac{1}{n}\sum_{i=1} e_i^{in}$, which is in general a rational number. This is consistent at least for incoming edges with Theorem 32 (*iv*), for which the expected value of N_k is indeed k.

We considered directed networks from the *Stanford Large Network Dataset Collection* of the Stanford Network Analysis Project (SNAP) (Leskovec and Krevl, 2014). We obtained similar results in different cases. Figure 13 reports the results concerning the following three directed networks: *WWW* (n = 875,713 nodes and m = 5,105,039 arcs), the `web-Google` web graph from Google; *Twitter* (n = 81,306 and m = 1,768,149), the `ego-Twitter` social circles from Twitter; and *Arxiv* (n = 34,546 and m = 421,578), the `cit-HepPh` arXiv high energy physics paper citation network.

Plots in Figure 13 are in a log-log scale and report on the horizontal axis the node degree and on the vertical axis the associated relative frequency. Blue dots represent the empirical values associated with each network. The magenta dashed line represents the function $p_k^u = p_k$ associated with a random variable uniformly distributed in $[-0.5, +0.5]$ ($\mu_2 = 0.08\overline{3}$, $\mu_4 = 0.0125$, and $\kappa = 1.8$). The red solid line represents the function $p_k^* = p_k$ using the values $\mu_2 = \mu_2^*$ and $\mu_4 = \mu_4^*$ (or $\kappa^* = \mu_4^*/(\mu_2^*)^2$) that minimize the Cramér-von Mises distance, also called *err* in the plots, between the cdf F_k and the empirical cdf of the node (in-)degree distribution. The Cramér-von Mises criterion corresponds to the integral of the squared difference between the empirical and the estimated distribution functions and is used to judge the goodness of fit of a cdf compared with an empirical cdf. This criterion depends on the entire cdf and gives more importance to the most probable observations. Alternative curves can be obtained by using other criteria.

Interestingly, we observed in different cases a good agreement between the empirical distribution of the number of incoming edges and the expected number of k-occurrences associated with the function p^u. In particular, the distance *err* for uniform data is in most cases similar to the distance for p^* associated with the best values for the parameters according to the Cramér-von mises statistics, and this is true especially for node in-degrees.

This suggests that for some real networks, the distribution of the incoming node degrees has connections with the herein-derived infinite-dimensional k-occurrences function N_k^∞, which models the number of reverse k-nearest neighbors in an arbitrarily large feature space of independent dimensions. Moreover, the function N_k^∞ appears to be suitable to be leveraged as a model for node-degrees distributions in complex real networks. We are currently investigating to what extent the above observations can be generalized and the use of these findings for the generation of realistic synthetic networks.

6. Concluding Discussion

This work investigated the distribution of distances in intrinsically high-dimensional spaces and leveraged this analysis to gain knowledge of phenomena related to the so-called dimensionality curse.

The study has been focused on independent data, because it is usually assumed that the number of independent dimensions dictates the intrinsic dimensionality of the data. By applying the central limit theorem to the Euclidean distance random variable, we obtained an approximation of the distance probability distribution between a given realization of a random vector and a random vector. The analysis of the error associated with the approximation highlights that, whereas the worst-case error always decreases with the dimensionality, there are configurations of n and d for which the hypothesized distance distribution can be considered equivalent, in terms of the expected empirical error, to the underlying distribution generating the observed inter-point distances.

With reference to the distribution of the nearest neighbors, we derived the expected distance of a point from its k-th nearest neighbor and the expected size of the ϵ-neighborhood in finite high-dimensional spaces, that is, the average number of points which emerge as ϵ-approximate neighbors of any other point, and then exploited it to determine the intrinsic dimensionality at which the neighborhood is expected to become unstable, called critical dimensionality. Also, a better estimate for the relative contrast for quantifying query difficulty has been obtained.

Moreover, the function N_k, or number of k-occurrences, representing the number of points which have a given point as one of their k nearest neighbors, has been investigated. Despite the extensive use of this function in many fields, including, among others, applied statistics and mathematical psychology, data mining and machine learning, information retrieval and computational geometry, the precise characterization of its form has been a longstanding problem. The limiting probability distribution of the function N_k has been derived, thereby providing full interpretability of the associated hubness problem.

It is well understood that complex networks arising from different natural conditions exhibit a few nodes, called *hubs*, of unusually high degree as compared to the other nodes of the network. Thus, we investigated if the findings relative to the distribution of the reverse nearest neighbors and the emergence of hubs in intrinsically high-dimensions is associated with the analogous phenomenon occurring in the context of networks. We concluded that for some real-life large-scale networks the distribution of the incoming node degrees is closely related to the herein-derived infinite-dimensional k-occurrences function N_k^∞ associated with uniform data and that this function is suitable to be leveraged as a model for node-degrees distribution in complex real networks.

We believe that the current study can be leveraged in several ways and in different contexts, such as direct and reverse nearest neighbor search, density estimation, anomaly and novelty detection, density-based clustering, and network analysis, as well as others, because almost all of them are based on the concepts of direct and reverse nearest neighbors. As for the study's possible applications, one is to obtain approximations of measures that are related to distance distributions. Another is to exploit the distributions for independent data as a worst-case scenario for data analysis and retrieval techniques in order to understand their behavior and limitations, in terms of meaningfulness or computational cost, as

dimensionality increases. Moreover, a deeper understanding of the behavior of intrinsically high dimensional spaces is fundamental to the design strategies that seek to mitigate the curse of dimensionality. For example, a line of research seeks to alleviate the problem by designing dissimilarity functions that suffer less on i.i.d. uniformly distributed features (François et al., 2007; Hsu and Chen, 2009). Moreover, from the discussion of Section 5, different applications within geometric models of complex networks can be devised.

To illustrate, the approximation of relative contrasts described in Section 4.2 results in estimates more accurate than those already provided, because the approach leverages a refined characterization of the distance distribution herein provided, which takes into account the relationship between the norm of the query point and its expected distance to the data points.

Additionally, the *Concentration Free Outlier Factor* (CFOF) recently introduced by Angiulli (2017) is a measure that aims to overcome the concentration problem in density estimation and outlier detection, whose behavior emerges from that of the k-occurrences function. Specifically, for a given parameter $\varrho \in [0,1]$ representing a fraction of the data population, the CFOF score of point $\boldsymbol{x}_d$ is $\mathrm{CFOF}(\boldsymbol{x}_d) = \min\{k/n : \mathrm{N}_k(\boldsymbol{x}_d) \geq n\varrho\}$, which is the smallest value for neighborhood parameter k (normalized on n) for which $\boldsymbol{x}_d$ presents a reverse neighborhood with a size of at least $n\varrho$. The intuition is that isolated points will require larger values of k than inliers in order to be selected as neighbors by an equal-sized fraction of the data population. In contrast to almost all known outlier detection measures, CFOF scores do not exhibit concentration. By leveraging the closed form of the function N_k it is possible to formally see that CFOF outliers are few in number and separated from inliers even in intrinsically high-dimensional spaces, whereas the direct use of the number of k-occurrences for outlier detection is prone to false positives. For further details, we refer to (Angiulli, 2017).

The understanding of properties characterizing high-dimensional spaces is also fundamental for enhancing intrinsic dimensionality estimation techniques. For example, Granata and Carnevale (2016), due to the difficulty of correctly working with the distance probability density function at small-length scales, propose to reconstruct that pdf at intermediate scales and then to compare it with a known pdf of a uniform distribution on a d-dimensional support.

Acknowledgments

The author would like to thank the anonymous reviewers for their meticulous work in reviewing the manuscript and for providing valuable suggestions that have helped to improve the presentation of the results.

Appendix A. Proofs

Proposition 11

$$cov\big(\|\boldsymbol{Y}_d\|^2, \langle \boldsymbol{X}_d, \boldsymbol{Y}_d\rangle\big) = d\mu(\mu_3 - \mu_2\mu) \\ \big(and\ cov\big(\|\boldsymbol{X}_d\|^2, \langle \boldsymbol{X}_d, \boldsymbol{Y}_d\rangle\big) = d\mu(\mu_3 - \mu_2\mu),\ for\ symmetry\big).$$

Proof of Proposition 11. Consider the covariance

$$\begin{aligned} cov\left(\|\boldsymbol{Y}_d\|^2, \langle \boldsymbol{X}_d, \boldsymbol{Y}_d\rangle\right) &= \mathbf{E}\big[\|\boldsymbol{Y}_d\|^2 \cdot \langle \boldsymbol{X}_d, \boldsymbol{Y}_d\rangle\big] - \mathbf{E}\big[\|\boldsymbol{Y}_d\|^2\big] \cdot \mathbf{E}\big[\langle \boldsymbol{X}_d, \boldsymbol{Y}_d\rangle\big] = \\ &= \mathbf{E}\left[\left(\sum_{i=1}^{d} Y_i^2\right) \cdot \left(\sum_{j=1}^{d} X_j Y_j\right)\right] - d\mu_2 \cdot d\mu^2 = \\ &= \mathbf{E}\left[\sum_{i=1}^{d} Y_i^3 X_i + \sum_{i=1}^{d}\sum_{i\neq j=1}^{d} X_j Y_j Y_i^2\right] - d^2\mu_2\mu^2 = \\ &= d\mu_3\mu + d(d-1)\mu_2\mu^2 - d^2\mu_2\mu^2 = d\mu(\mu_3 - \mu_2\mu). \end{aligned}$$

■

Proposition 17 *Let $\boldsymbol{X}_d$ be a d-dimensional i.i.d. random vector having cdf F_X. Moreover, let p and q be positive integers, and $\beta_0, \beta_1, \ldots, \beta_p$, $\alpha_0, \alpha_1, \ldots, \alpha_q$ be real coefficients such that $\beta_p \neq 0$ and $\alpha_q \neq 0$. Then, for any $\epsilon > 0$,*

$$\lim_{d\to\infty} Pr\left[\left|\frac{\sum_{i=1}^{d}\left(\sum_{j=0}^{p}\beta_j X_i^j\right)}{\left(\sum_{i=1}^{d}\left(\sum_{j=0}^{q}\alpha_j X_i^j\right)\right)^2}\right| \geq \epsilon\right] = 0.$$

Proof of Proposition 17. Let $U_i = \left(\sum_{j=0}^{p}\beta_j X_i^j\right)$ and $V_i = \left(\sum_{j=0}^{q}\alpha_j X_i^j\right)$ $(1 \leq i \leq d)$. Moreover, let U be $\sum_{i=1}^{d} U_i$ and let V be $\sum_{i=1}^{d} V_i$. Now it is shown that, for all $\epsilon > 0$,

$$\lim_{d\to\infty} Pr\left[\left|\frac{U}{V^2}\right| \geq \epsilon\right] = 0.$$

The mean and variance of V_i are as follows (mean and variance of U_i are similar):

$$\begin{aligned} \mathbf{E}[V_i] &= \mathbf{E}\left[\sum_{j=1}^{q}\alpha_j X_i^j\right] = \sum_{j=1}^{q}\alpha_j\mu_j, \\ \sigma^2(V_i) &= \mathbf{E}[V_i^2] - \mathbf{E}[V_i]^2 = \mathbf{E}\left[\left(\sum_{j=1}^{q}\alpha_j X_i^j\right)^2\right] - \left(\sum_{j=1}^{q}\alpha_j\mu_j\right)^2 = \\ &= \mathbf{E}\left[\sum_{j=1}^{q}\alpha_j^2 X_i^{2j} + \sum_{j=1}^{q}\sum_{k\neq j}^{q}\alpha_j\alpha_k X_i^j X_i^k\right] - \left(\sum_{j=1}^{q}\alpha_j^2\mu_j^2 + \sum_{j=1}^{q}\sum_{k\neq j}^{q}\alpha_j\alpha_k\mu_j\mu_k\right) = \\ &= \sum_{j=1}^{q}\alpha_j^2\mu_{2j} + \sum_{j=1}^{q}\sum_{k\neq j}^{q}\alpha_j\alpha_k\mu_j\mu_k - \sum_{j=1}^{q}\alpha_j^2\mu_j^2 - \sum_{j=1}^{q}\sum_{k\neq j}^{q}\alpha_j\alpha_k\mu_j\mu_k = \\ &= \sum_{j=1}^{q}\alpha_j^2\left(\mu_{2j} - \mu_j^2\right). \end{aligned}$$

We assume that moments up to $2\max\{p,q\}$ exist finite. Since both U and V are the sum of d independent identically distributed random variables, by the CLT, as $d \to \infty$,

$$U \approx \mathcal{N}\left(d\sum_{j=1}^{p}\alpha_j\mu_j,\ d\sum_{j=1}^{p}\alpha_j^2\left(\mu_{2j}-\mu_j^2\right)\right) \text{ and } V \approx \mathcal{N}\left(d\sum_{j=1}^{p}\beta_j\mu_j,\ d\sum_{j=1}^{p}\beta_j^2\left(\mu_{2j}-\mu_j^2\right)\right).$$

Consider now the random variable V^2, having mean

$$\mathbf{E}[V^2] \quad = \quad \mathbf{E}[V]^2+\sigma^2(V) = d^2\mu_{V_i}^2+d\sigma_{V_i}^2 = O(d^2).$$

Now it is essential to show that $\sigma^2(V^2)=O(d^3)$ and $cov(U,V^2)=O(d^2)$.

As for the variance $\sigma^2(V^2)=\mathbf{E}[V^4]-\mathbf{E}[V^2]^2$ of V^2, notice that the term of higher order in $\mathbf{E}[V^4]$ derives from the summation

$$\mathbf{E}\left[\sum_{i\neq j\neq k\neq h}V_iV_jV_kV_h\right] = d(d-1)(d-2)(d-3)\mu_{V_i}^4 = (d^4-6d^3+11d^2-6d)\mu_{V_i}^4,$$

and that all the other terms in $\mathbf{E}[V^4]$ are $O(d^3)$. As for $\mathbf{E}[V^2]^2 = (d^2\mu_{V_i}^2+d\sigma_{V_i}^2)^2 = d^4\mu_{V_i}^4+2d^3\mu_{V_i}^2\sigma_{V_i}^2+d^2\sigma_{V_i}^4$. Since both $\mathbf{E}[V^4]$ and $\mathbf{E}[V^2]^2$ contain as term of higher order $d^4\mu_{V_i}^4$, it then follows that $\sigma^2(V^2)=O(d^3)$.

As for the covariance $cov(U,V^2)=\mathbf{E}[U\cdot V^2]-\mathbf{E}[U]\mathbf{E}[V^2]$, similar considerations hold. Indeed, notice that the term of higher order in $\mathbf{E}[U\cdot V^2]$ derives from the summation:

$$\mathbf{E}\left[\sum_{i\neq j\neq k}U_iV_jV_k\right] = d(d-1)(d-2)\mu_{U_i}\mu_{V_i}^2$$

and that all the other terms in $\mathbf{E}[U\cdot V^2]$ are $O(d^2)$. As for $\mathbf{E}[U]\mathbf{E}[V^2] = d\mu_{U_i}(d^2\mu_{V_i}^2+d\sigma_{V_i}^2) = d^3\mu_{U_i}\mu_{V_i}^2+d^2\mu_{U_i}\sigma_{V_i}^2$. Since both $\mathbf{E}[U\cdot V^2]$ and $\mathbf{E}[U]\mathbf{E}[V^2]$ contain as term of higher order $d^3\mu_{U_i}\mu_{V_i}^2$, it then follows that $cov(U,V^2)=O(d^2)$.

By exploiting Taylor series, it can be written:

$$\begin{aligned}\mathbf{E}\left[\frac{U}{V^2}\right] &\approx \frac{\mathbf{E}[U]}{\mathbf{E}[V^2]}-\frac{cov(U,V^2)}{\mathbf{E}[V^2]^2}+\frac{\mathbf{E}[U]}{\mathbf{E}[V^2]^3}\sigma^2(V^2), \text{ and}\\ \sigma^2\left(\frac{U}{V^2}\right) &\approx \left(\frac{\mathbf{E}[U]}{\mathbf{E}[V^2]}\right)^2\left(\frac{\sigma^2[U]}{\mathbf{E}[U]^2}+\frac{\sigma^2(V^2)}{\mathbf{E}[V^2]^2}-\frac{2cov(U,V^2)}{\mathbf{E}[U]\mathbf{E}[V^2]}\right).\end{aligned}$$

Then

$$\begin{aligned}\lim_{d\to\infty}\mathbf{E}\left[\frac{U}{V^2}\right] &= \lim_{d\to\infty}\left(\frac{O(d)}{O(d^2)}-\frac{O(d^2)}{O(d^4)}+\frac{O(d)}{O(d^6)}O(d^3)\right) = \lim_{d\to\infty}O(d^{-1}) = 0, \text{ and}\\ \lim_{d\to\infty}\sigma^2\left(\frac{U}{V^2}\right) &= \lim_{d\to\infty}\left(\frac{O(d)}{O(d^2)}\right)^2\left(\frac{O(d)}{O(d^2)}+\frac{O(d^3)}{O(d^4)}-\frac{O(d^2)}{O(d^3)}\right) = \lim_{d\to\infty}O(d^{-3}) = 0.\end{aligned}$$

Since both $\mathbf{E}[U/V^2]$ and $\sigma^2(U/V^2)$ are vanishing as $d\to\infty$, by exploiting the Chebicheff theorem it can be proved that U/V^2 converges in probability to 0. Let $W_d = U/V^2$ then,

$\{W_d\}$ converges in probability towards the value zero, if for all $\epsilon>0$ the following limit evaluates to 0:

$$\lim_{d\to\infty}Pr\left[|W_d|\geq\epsilon\right] = \lim_{d\to\infty}Pr\left[|W_d-\mathbf{E}[W_d]|\geq\epsilon\right] \leq \lim_{d\to\infty}\frac{\sigma^2(W_d)}{\epsilon^2} = \lim_{d\to\infty}\frac{1}{\epsilon^2O(d^3)} = 0.$$

■

Proposition 19 *As $d\to\infty$, with high probability $\|\boldsymbol{Y}_d\|^2$ and $\langle\boldsymbol{x}_d,\boldsymbol{Y}_d\rangle$ are jointly normally distributed.*

Proof of Proposition 19. The proof follows a line similar to that exploited in Propositions 10 and 18. It must be shown that all linear combinations

$$Z = a\|\boldsymbol{Y}_d\|^2+b\langle\boldsymbol{x}_d,\boldsymbol{Y}_d\rangle = a\left(\sum_{i=1}^{d}Y_i^2\right)+b\left(\sum_{i=1}^{d}x_iY_i\right) = \sum_{i=1}^{d}(aY_i^2+bx_iY_i) = \sum_{i=1}^{d}W_i$$

are normally distributed, where $W_1,W_2,W_3,\ldots$ form a sequence of independent, but not identically distributed, random variables. The proof is completed by noticing that

$$\begin{aligned}\mathbf{E}\left[|W_i-\mathbf{E}[W_i]|^4\right] &= \mathbf{E}\left[W_i^4-4W_i^3\mathbf{E}[W_i]+6W_i^2\mathbf{E}[W_i]^2-4W_i\mathbf{E}[W_i]^3+\mathbf{E}[W_i]^4\right] =\\ &= \mathbf{E}[W_i^4]-4\mathbf{E}[W_i^3]\mathbf{E}[W_i]+6\mathbf{E}[W_i^2]\mathbf{E}[W_i]^2-3\mathbf{E}[W_i]^4 = \sum_{j=0}^{4}\alpha_jx^j,\end{aligned}$$

and

$$\sigma_{W_i}^2 \quad = \quad (b^2\mu_2)x_i^2+(2ab\mu_3)x_i+(a^2\mu_4-a\mu_2^2) = \beta_2x_i^2+\beta_1x_i+\beta_0,$$

from which the Lyapunov CLT condition (see Equation 1) for $\delta=2$:

$$\lim_{d\to\infty}\left.\frac{\mathbf{E}\left[|W_i-\mathbf{E}[W_i]|^{2+\delta}\right]}{s_d^{2+\delta}}\right|_{\delta=2} = \lim_{d\to\infty}\frac{\sum_{i=1}^{d}\left(\sum_{j=0}^{4}\alpha_jx_i^j\right)}{\left(\sum_{i=1}^{d}\left(\sum_{j=0}^{2}\beta_jx_i^j\right)\right)^2} = 0.$$

The above limit converges in probability to zero for the r.v. $\boldsymbol{X}_d$ by Proposition 17. ■

Proposition 20 $cov\left(\|\boldsymbol{Y}_d\|^2,\langle\boldsymbol{x}_d,\boldsymbol{Y}_d\rangle\right) = (\mu_3-\mu\mu_2)\sum_{i=1}^{d}x_i.$

Proof of Proposition 20. Consider the covariance

$$\begin{aligned} cov\left(\|\boldsymbol{Y}_d\|^2, \langle \boldsymbol{x}_d, \boldsymbol{Y}_d\rangle\right) &= \mathbf{E}\left[\|\boldsymbol{Y}_d\|^2 \cdot \langle \boldsymbol{x}_d, \boldsymbol{Y}_d\rangle\right] - \mathbf{E}\left[\|\boldsymbol{Y}_d\|^2\right] \cdot \mathbf{E}\left[\langle \boldsymbol{x}_d, \boldsymbol{Y}_d\rangle\right] = \\ &= \mathbf{E}\left[\left(\sum_{i=1}^{d} Y_i^2\right) \cdot \left(\sum_{j=1}^{d} x_j Y_j\right)\right] - d\mu_2 \cdot \mu \sum_{i=1}^{d} x_i = \\ &= \mathbf{E}\left[\sum_{i=1}^{d}\sum_{j=1}^{d} x_j Y_j Y_i^2\right] - d\mu\mu_2 \sum_{i=1}^{d} x_i = \\ &= \sum_{i=1}^{d} x_i \mathbf{E}[Y_i^3] + \sum_{i=1}^{d-1}\sum_{i\neq j=1}^{d} x_j \mathbf{E}[Y_j]\mathbf{E}[Y_i^2] - d\mu\mu_2 \sum_{i=1}^{d} x_i = \\ &= \mu_3 \sum_{i=1}^{d} x_i + (d-1)\mu\mu_2 \sum_{i=1}^{d} x_i - d\mu\mu_2 \sum_{i=1}^{d} x_i = (\mu_3 - \mu\mu_2)\sum_{i=1}^{d} x_i. \end{aligned}$$

■

Proposition 22 *Let $\boldsymbol{x}_d$ denote a realization of a d-dimensional i.i.d. random vector $\boldsymbol{X}_d$ with cdf F_X. Then, for large values of d, with high probability*

$$\frac{\sum_{i=1}^{d} x_i}{\|\boldsymbol{x}_d\|^2} \to \frac{\mu_X}{\mu_{X,2}}.$$

Proof of Proposition 22. Assume that in general $\mu \in \mathbb{R}$. By the CLT, following the same line of reasoning of Proposition 8, it can be seen that the random variable

$$U = \sum_{i=1}^{d} X_i \approx \mathcal{N}\left(d\mu, d(\mu_2 - \mu^2)\right).$$

Let $V = \|\boldsymbol{X}_d\|^2 = \sum_{i=1}^{d} X_i^2$. By Proposition 8, $V \approx \mathcal{N}\big(d\mu_2, d(\mu_4 - \mu_2^2)\big)$. As for the covariance $cov(U, V)$, it is $d(\mu_3 - \mu\mu_2)$. Consider the ratio U/V. Now it is shown that

$$\lim_{d\to\infty} Pr\left[\left|\frac{U}{V} - \frac{\mu}{\mu_2}\right| \geq \epsilon\right] = 0.$$

By exploiting Taylor series, the mean of U/V is:

$$\begin{aligned} \mathbf{E}\left[\frac{U}{V}\right] &\approx \frac{\mathbf{E}[U]}{\mathbf{E}[V]} - \frac{cov(U,V)}{\mathbf{E}[V]^2} + \frac{\mathbf{E}[U]}{\mathbf{E}[V]^3}\sigma^2(V) = \frac{d\mu}{d\mu_2} - \frac{d(\mu_3 - \mu\mu_2)}{d^2\mu_2^2} + \frac{d\mu}{d^3\mu_2^3} d(\mu_4 - \mu_2^2) = \\ &= \frac{\mu}{\mu_2} - \frac{\mu_3 - \mu\mu_2}{d\mu_2^2} + \frac{\mu}{d\mu_2^3}(\mu_4 - \mu_2^2) = \frac{\mu}{\mu_2} + \frac{1}{d} \cdot \frac{\mu(\mu_4 - \mu_2^2) - \mu_2(\mu_3 - \mu\mu_2)}{\mu_2^3}, \end{aligned}$$

while the variance of U/V is:

$$\begin{aligned} \sigma^2\left(\frac{U}{V}\right) &\approx \left(\frac{\mathbf{E}[U]}{\mathbf{E}[V]}\right)^2 \left(\frac{\sigma^2[U]}{\mathbf{E}[U]^2} + \frac{\sigma^2(V)}{\mathbf{E}[V]^2} - \frac{2cov(U,V)}{\mathbf{E}[U]\mathbf{E}[V]}\right) = \\ &= \left(\frac{d\mu}{d\mu_2}\right)^2 \left(\frac{d(\mu_2 - \mu^2)}{d^2\mu^2} + \frac{d(\mu_4 - \mu_2^2)}{d^2\mu_2^2} - \frac{2d(\mu_3 - \mu\mu_2)}{d^2\mu\mu_2}\right) = \\ &= \frac{1}{d} \cdot \frac{\mu_2^2(\mu_2 - \mu^2) + \mu(\mu_4 - \mu_2^2) - 2\mu\mu_2(\mu_3 - \mu\mu_2)}{\mu_2^4}. \end{aligned}$$

The statement then follows by applying the Chebicheff theorem to show that U/V converges in probability to μ/μ_2.

$$\lim_{d\to\infty} Pr\left[\left|\frac{U}{V} - \frac{\mu}{\mu_2}\right| \geq \epsilon\right] = \lim_{d\to\infty} Pr\left[\left|\frac{U}{V} - \mathbf{E}\left[\frac{U}{V}\right]\right| \geq \epsilon\right] \leq \lim_{d\to\infty} \frac{\sigma^2\left(\frac{U}{V}\right)}{\epsilon^2} = \lim_{d\to\infty} \frac{1}{\epsilon^2 O(d)} = 0.$$

■

Lemma 29 *Let $\boldsymbol{x}_d$ denote a realization of a d-dimensional i.i.d. random vector $\boldsymbol{X}_d$ with cdf F_X and let $\boldsymbol{Y}_d$ be a d-dimensional i.i.d. random vector with cdf F_Y. Assume, w.l.o.g., that F_Y has null mean $\mu_Y = 0$. Then, for large values of d, with high probability*

$$Pr\left[\text{dist}(\boldsymbol{x}_d, \boldsymbol{Y}_d) \leq \delta \mid \|\boldsymbol{Y}_d\| = R\right] \approx \Phi\left(\frac{\delta^2 - R^2 - \|\boldsymbol{x}_d\|^2}{2\|\boldsymbol{x}_d\|\sqrt{\mu_2}}\right),$$

where moments are relative to the random vector $\boldsymbol{Y}_d$.

Proof of Lemma 29. By Proposition 19, $\|\boldsymbol{Y}_d\|^2$ and $\langle \boldsymbol{x}_d, \boldsymbol{Y}_d\rangle$ are jointly normally distributed. Then,

$$\begin{aligned} Pr[\text{dist}(\boldsymbol{x}_d, \boldsymbol{Y}_d) \leq \delta \mid \|\boldsymbol{Y}_d\| = R] &= Pr[\|\boldsymbol{x}_d - \boldsymbol{Y}_d\| \leq \delta \mid \|\boldsymbol{Y}_d\| = R] = \\ &= Pr[\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2 \leq \delta^2 \mid \|\boldsymbol{Y}_d\|^2 = R^2] = \\ &= Pr[\|\boldsymbol{x}_d\|^2 + \|\boldsymbol{Y}_d\|^2 - 2\langle \boldsymbol{x}_d, \boldsymbol{Y}_d\rangle \leq \delta^2 \mid \|\boldsymbol{Y}_d\|^2 = R^2]. \end{aligned}$$

Notice that, for distributions F_Y having null skewness ($\mu_{Y,3} = 0$), by Proposition 20, $\|\boldsymbol{Y}_d\|^2$ and $\langle \boldsymbol{x}_d, \boldsymbol{Y}_d\rangle$ are both uncorrelated and independent, and it can be written that

$$Pr[\|\boldsymbol{x}_d\|^2 + \|\boldsymbol{Y}_d\|^2 - 2\langle \boldsymbol{x}_d, \boldsymbol{Y}_d\rangle \leq \delta^2 \mid \|\boldsymbol{Y}_d\|^2 = R^2] = Pr[\|\boldsymbol{x}_d\|^2 + R^2 - 2\langle \boldsymbol{x}_d, \boldsymbol{Y}_d\rangle \leq \delta^2],$$

from which the statement follows by exploiting Proposition 18.

More in general ($\mu_{Y,3} \neq 0$) by leveraging properties within Theorem 15 and by Proposition 22 the distribution of the squared norm $\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2$ subject to the constraint that $\|\boldsymbol{Y}_d\|^2 = R^2$, tends to the normal distribution with mean

$$\mu = \|\boldsymbol{x}_d\|^2 + R^2$$

and variance (by the assumption that F_X has null mean $\mu_X = 0$)

$$\sigma^2 = 4\mu_{Y,2}\|\boldsymbol{x}_d\|^2 - 4\mu_{Y,3}\frac{\mu_X}{\mu_{X,2}}\|\boldsymbol{x}_d\|^2 = 4\mu_{Y,2}\|\boldsymbol{x}_d\|^2,$$

from which the above expression again follows.

The above holds under the assumption that R itself is selected with high probability, that is $R^2 \approx \mathbf{E}[\|\boldsymbol{Y}_d\|^2]$, as also assumed in Theorem 30. For a generic R, the moments μ_2 and μ conditioned on $\|\boldsymbol{Y}_d\| = R$ must be used. As for $\mu_2 = \mathbf{E}[Y^2 \mid \|\boldsymbol{Y}_d\|^2 = R^2] = \mathbf{E}[Y^2 \mid d\mathbf{E}[Y^2] = R^2] = R^2/d$. As for μ conditioned on $\|\boldsymbol{Y}_d\|^2 = R^2$, it is $\mu = \mu_Y = 0$ for symmetric distributions, since for each $\boldsymbol{y}_d$ such that $\|\boldsymbol{y}_d\|^2 = R^2$, it holds that $\|-\boldsymbol{y}_d\|^2 = R^2$ and $f_Y(\boldsymbol{y}_d) = f_Y(-\boldsymbol{y}_d)$.

Moreover, the closer R^2 to $\mathbf{E}[\|\boldsymbol{Y}_d\|^2] = \mu_{\|\boldsymbol{Y}_d\|^2} = d\mu_{Y,2}$, the closer the moments to their unconditioned values. Indeed, for $k \geq 1$

$$\mathbf{E}[Y^k \mid \|\boldsymbol{Y}_d\|^2 = R^2] = \tfrac{1}{Pr[\|\boldsymbol{Y}_d\|^2 = R^2]} \left(\int_{\mathbb{R}} y^k \, f_Y(y) \, Pr\left[\left(\sum_{j>1}^{d} Y_j^2 \right) = R^2 - y^2 \right] \, \mathrm{d}y \right) \approx$$
$$\approx \tfrac{1}{\phi_{\|\boldsymbol{Y}_d\|^2}(R^2)} \left(\int_{\mathbb{R}} y^k \, f_Y(y) \, \phi_{\|\boldsymbol{Y}_d\|^2}\left(R^2 - y^2\right) \, \mathrm{d}y \right).$$

Hence, since μ_Y exists finite, as $d \to \infty$

$$\mathbf{E}\left[Y^k \mid \|\boldsymbol{Y}_d\|^2 = \mu_{\|\boldsymbol{Y}_d\|^2}\right] \approx \tfrac{1}{\phi_{\|\boldsymbol{Y}_d\|^2}\left(\mu_{\|\boldsymbol{Y}_d\|^2}\right)} \left(\int_{\mathbb{R}} y^k \, f_Y(y) \, \phi_{\|\boldsymbol{Y}_d\|^2}\left(\mu_{\|\boldsymbol{Y}_d\|^2} - y^2\right) \, \mathrm{d}y \right) =$$
$$= \tfrac{1}{\phi(0)} \left(\int_{\mathbb{R}} y^k \, f_Y(y) \, \phi\left(-y^2/\sigma_{\|\boldsymbol{Y}_d\|^2}\right) \, \mathrm{d}y \right) \approx \tfrac{\phi(0)}{\phi(0)} \int_{\mathbb{R}} y^k \, f_Y(y) \, \mathrm{d}y = \mu_{Y,k}.$$

■

Theorem 32

$$(i) \quad \lim_{d\to\infty} Pr[\mathrm{N}_\varrho(\boldsymbol{X}_d) \leq \theta] = \Phi\left(\frac{\Phi^{-1}(\theta) 2\mu_2 - \Phi^{-1}(\varrho)\sqrt{\mu_4 + 3\mu_2^2}}{\sqrt{\mu_4 - \mu_2^2}} \right),$$

$$(ii) \quad \lim_{d\to\infty} Pr[\mathrm{N}_\varrho(\boldsymbol{X}_d) = \theta] = \frac{2\mu_2}{\sqrt{\mu_4 - \mu_2^2}} \cdot \frac{1}{\phi(\Phi^{-1}(\theta))} \cdot \phi\left(\frac{\Phi^{-1}(\theta) 2\mu_2 - \Phi^{-1}(\varrho)\sqrt{\mu_4 + 3\mu_2^2}}{\sqrt{\mu_4 - \mu_2^2}} \right),$$

$$(iii) \quad \lim_{d\to\infty} \sigma^2(\mathrm{N}_\varrho(\boldsymbol{X}_d)) = \varrho(1-\varrho) - 2T\left(\Phi^{-1}(\varrho), \frac{2\mu_2}{\sqrt{2(\mu_4 + \mu_2^2)}} \right), \text{ and}$$

$$(iv) \quad \lim_{d\to\infty} \mathbf{E}[\mathrm{N}_\varrho(\boldsymbol{X}_d)] = \varrho,$$

where $T(h,a) = \phi(h) \int_0^a \frac{\phi(hx)}{1+x^2} \, \mathrm{d}x$ *is the Owen's T function.*

Proof of Theorem 32. Since

$$\mathrm{N}_\varrho^\infty(Z_\theta) \leq \theta \implies Z_\theta \leq \frac{\Phi^{-1}(\theta) 2\mu_2 - \Phi^{-1}(\varrho)\sqrt{\mu_4 + 3\mu_2^2}}{\sqrt{\mu_4 - \mu_2^2}},$$

point (i) corresponds to $\Phi(Z_\theta)$ and point (ii) to $\frac{\mathrm{d}}{\mathrm{d}\theta}\Phi(Z_\theta)$.

As for points (iii) and (iv), consider the Owen's Gaussian-type integrals reported in Equation (2) and in the following equation due to Owen (1980):

$$\int_{-\infty}^{+\infty} \Phi(a+bx)^2 \phi(x) \, \mathrm{d}x = \Phi\left(\frac{a}{\sqrt{1+b^2}} \right) - 2T\left(\frac{a}{\sqrt{1+b^2}}, \frac{1}{\sqrt{1+2b^2}} \right). \tag{7}$$

Let us consider first point (iv). Since

$$\lim_{d\to\infty} \mathbf{E}[\mathrm{N}_\varrho(\boldsymbol{X}_d)] = \int_{-\infty}^{+\infty} \Phi\left(\frac{\Phi^{-1}(\varrho)\sqrt{\mu_4 + 3\mu_2^2} - z\sqrt{\mu_4 - \mu_2^2}}{2\mu_2} \right) \phi(z) \, \mathrm{d}z,$$

by substituting $a = \frac{\Phi^{-1}(\varrho)\sqrt{\mu_4+3\mu_2^2}}{2\mu_2}$ and $b = -\frac{\sqrt{\mu_4-\mu_2^2}}{2\mu_2}$ in Equation (2):

$$\lim_{d\to\infty} \mathbf{E}[\mathrm{N}_\varrho(\boldsymbol{X}_d)] = \Phi\left(\frac{a}{\sqrt{1+b^2}} \right) = \Phi\left(\frac{\Phi^{-1}(\varrho)\sqrt{\mu_4 + 3\mu_2^2}}{\sqrt{\mu_4 + 3\mu_2^2}} \right) = \Phi(\Phi^{-1}(\varrho)) = \varrho.$$

As for point (iii),

$$\lim_{d\to\infty} \sigma^2(\mathrm{N}_\varrho(\boldsymbol{X}_d)) = \lim_{d\to\infty} \mathbf{E}[\mathrm{N}_\varrho(\boldsymbol{X}_d)^2] - \mathbf{E}[\mathrm{N}_\varrho(\boldsymbol{X}_d)]^2,$$

and by substituting a and b as above in the first (see Equation 2) and second (see Equation 7) Owen's formula the result is obtained. ■

Corollary 34 *Let k be a fixed positive integer. Then*

$$(i) \lim_{n\to\infty} \lim_{d\to\infty} \mathrm{N}_k^{n,d} \xrightarrow{D} 0, \quad (ii) \lim_{n\to\infty} \lim_{d\to\infty} \sigma^2(\mathrm{N}_k^{n,d}) = \infty, \text{ and } (iii) \lim_{n\to\infty} \lim_{d\to\infty} \mathbf{E}[\mathrm{N}_k^{n,d}] = k.$$

Proof of Corollary 34. All the points derive from Theorem 32. As for point (1) it suffices to show that $F_{N_k^{\infty,\infty}}(h) = Pr[N_k^{\infty,\infty} \leq h] = 1$ for $h > 0$, since $h = 0$ is not a continuity point for the cdf:

$$\begin{aligned}
\lim_{n\to\infty} \lim_{d\to\infty} Pr[\mathrm{N}_k^{n,d} > 0] &= 1 - \lim_{n\to\infty} \lim_{d\to\infty} Pr[\mathrm{N}_k^{n,d} \leq 0] = 1 - \lim_{n\to\infty} Pr[\mathrm{N}_k^{n,\infty} \leq 0] = \\
&= 1 - \lim_{n\to\infty} \Phi\left(\frac{\Phi^{-1}(0) 2\mu_2 - \Phi^{-1}(\frac{k}{n})\sqrt{\mu_4 + 3\mu_2^2}}{\sqrt{\mu_4 - \mu_2^2}} \right) = \\
&= 1 - \Phi\left(\Phi^{-1}(0) \frac{2\mu_2 - \sqrt{\mu_4 + 3\mu_2^2}}{\sqrt{\mu_4 - \mu_2^2}} \right) = 1 - \Phi(-\infty) = 1.
\end{aligned}$$

As for point (2),

$$\begin{aligned}
\lim_{n\to\infty} \lim_{d\to\infty} \sigma^2(\mathrm{N}_k^{n,d}) &= \lim_{n\to\infty} n^2 \left(\frac{k}{n} - \frac{k}{n^2} - 2T\left(\Phi^{-1}\left(\frac{k}{n}\right), \frac{2\mu_2}{\sqrt{2(\mu_4 + \mu_2^2)}} \right) \right) = \\
&= \lim_{n\to\infty} nk - k - 0 = \infty.
\end{aligned}$$

As for point (3),

$$\lim_{n\to\infty} \lim_{d\to\infty} \mathbf{E}[\mathrm{N}_k^{n,d}] = \lim_{d\to\infty} n\left(\frac{k}{n}\right) = k.$$

■

Proposition 35 *Let $U_d = \sum_{i=1}^{d} W_i$ be a random variable defined as the summation of a sequence of independent, but not identically distributed, random variables W_i having comparable central moments. Then*

$$U_d \simeq \mathcal{N}\left(\sum_{i=1}^{d} \mu_{W_i},\ \sum_{i=1}^{d} \sigma^2_{W_i}\right) = \mathcal{N}\left(d \cdot \overline{\mu}_W,\ d \cdot \overline{\sigma^2}_W\right),$$

where $\overline{\mu}_W = (1/d)\sum_{i=1}^{d} \mu_{W_i}$ and $\overline{\sigma^2}_W = (1/d)\sum_{i=1}^{d} \sigma^2_{W_i}$.

Proof of Proposition 35. For variables W_i having comparable central moments the Lyapunov CLT condition holds:

$$\lim_{d\to\infty} \frac{\sum_{i=1}^{d} \mathbf{E}\left[(W_i - \mathbf{E}[W_i])^4\right]}{\left(\sum_{i=1}^{d} \sigma^2(W_i)\right)^2} = \lim_{d\to\infty} \frac{\sum_{i=1}^{d} \hat{\mu}_{i,4}}{\left(\sum_{i=1}^{d} \hat{\mu}_{i,2}\right)^2} \leq \lim_{d\to\infty} \frac{d\hat{\mu}_{max}}{(d\hat{\mu}_{min})^2} = \lim_{d\to\infty} \frac{\hat{\mu}_{max}}{d\hat{\mu}^2_{min}} = 0.$$

■

Theorem 37. *Let $\boldsymbol{X}_d$ and $\boldsymbol{Y}_d$ be two independent non-identically distributed d-dimensional random vectors with common cdfs $\boldsymbol{F}$ having means $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_d)$, non null variances, and comparable central moments, and let $\boldsymbol{x}_d$ denote a realization of $\boldsymbol{X}_d$. The results of Sections 4.1, 4.2 and 4.3 can be applied to $\boldsymbol{X}_d$, $\boldsymbol{Y}_d$, and $\boldsymbol{x}_d$ by taking into account the average central moments of $\boldsymbol{X}_d$ and $\boldsymbol{Y}_d$ and the realization $\boldsymbol{x}_d - \boldsymbol{\mu}$.*

Proof of Theorem 37. W.l.o.g. assume that $\boldsymbol{\mu} = (0, \ldots, 0)$, for otherwise it is sufficient to replace vector $\boldsymbol{X}_d$ with $\hat{\boldsymbol{X}}_d = \boldsymbol{X}_d - \boldsymbol{\mu}$ and vector $\boldsymbol{Y}_d$ with $\hat{\boldsymbol{Y}}_d = \boldsymbol{Y}_d - \boldsymbol{\mu}$. Thus, from now $\mu_i = \mathbf{E}[X_i] = \mathbf{E}[Y_i] = 0$.

Let $\boldsymbol{\mu_k} = (\mu_{k,1}, \ldots, \mu_{k,d})$ denote the k-th moments of $\boldsymbol{X}_d$ ($\boldsymbol{Y}_d$, resp.). The result follows by taking into account that the variables X_i and Y_i ($1 \leq i \leq d$) are independent but not identically distributed and, hence, by exploiting the average moments to formulate expressions.

Let $\boldsymbol{p} = (p_1, \ldots, p_d)$ and $\boldsymbol{q}_d = (q_1, \ldots, q_d)$ two d-dimensional vectors, and let h be a positive integer. In the following we denote by $\boldsymbol{p^k}$ the vector $\boldsymbol{p^k} = (p_1^k, \ldots, p_d^h)$ and by $\boldsymbol{p} \cdot \boldsymbol{q}$ the scalar product $\langle \boldsymbol{p}, \boldsymbol{q} \rangle = \sum_{i=1}^{d} p_i q_i$ of $\boldsymbol{p}$ and $\boldsymbol{q}$.

Thus, as for the results of Section 4.1, we obtain the following expressions:

(**P37.1**) $\|\boldsymbol{Y}_d\|^2 \simeq \mathcal{N}\left(d\tilde{\boldsymbol{\mu}}_{\boldsymbol{2}},\ d(\tilde{\boldsymbol{\mu}}_{\boldsymbol{4}} - \tilde{\boldsymbol{\mu}}^{\boldsymbol{2}}_{\boldsymbol{2}})\right)$;

(**P37.2**) $\langle \boldsymbol{X}_d, \boldsymbol{Y}_d \rangle \simeq \mathcal{N}\left(0,\ \tilde{\boldsymbol{\mu}}^{\boldsymbol{2}}_{\boldsymbol{2}}\right)$;

(**P37.3**) $cov(\|\boldsymbol{Y}_d\|^2, \langle \boldsymbol{X}_d, \boldsymbol{Y}_d \rangle) = 0$;

(**P37.4**) $\|\boldsymbol{X}_d - \boldsymbol{Y}_d\|^2 \simeq \mathcal{N}\left(2d\tilde{\boldsymbol{\mu}}_{\boldsymbol{2}},\ 2d(\tilde{\boldsymbol{\mu}}_{\boldsymbol{4}} - \tilde{\boldsymbol{\mu}}^{\boldsymbol{2}}_{\boldsymbol{2}})\right)$;

(**P37.5**) $\langle \boldsymbol{x}_d, \boldsymbol{Y}_d \rangle \simeq \mathcal{N}\left(0,\ \boldsymbol{\mu_2} \cdot \boldsymbol{x^2}\right)$;

(**P37.6**) $cov(\boldsymbol{Y}_d, \langle \boldsymbol{x}_d, \boldsymbol{Y}_d \rangle) = \boldsymbol{\mu_3} \cdot \boldsymbol{x_d}$;

(**P37.7**) $\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2 \simeq \mathcal{N}\left(\|\boldsymbol{x}_d\|^2 + d\tilde{\boldsymbol{\mu}}_{\boldsymbol{2}},\ d(\tilde{\boldsymbol{\mu}}_{\boldsymbol{4}} - \tilde{\boldsymbol{\mu}}^{\boldsymbol{2}}_{\boldsymbol{2}}) + 4\boldsymbol{\mu_2} \cdot \boldsymbol{x^2_d} - 4\boldsymbol{\mu_3} \cdot \boldsymbol{x_d}\right)$.

Expression (**P37.1**) can be derived by exploiting $\mathbf{E}[\|\boldsymbol{Y}_d\|^2]$ and $\sigma(\|\boldsymbol{Y}_d\|^2)$ in terms of the average central moments of the random vectors, as illustrated next:

$$\begin{aligned}
\mathbf{E}[\|\boldsymbol{Y}_d\|^2] &= \mathbf{E}\left[\textstyle\sum_i Y_i^2\right] = \sum_i \mathbf{E}[Y_i^2] = \sum_i \mu_{i,2} = d\tilde{\boldsymbol{\mu}}_{\boldsymbol{2}},\\
\mathbf{E}[\|\boldsymbol{Y}_d\|^4] &= \mathbf{E}\left[\left(\textstyle\sum_i Y_i^2\right)^2\right] = \mathbf{E}\left[\textstyle\sum_i Y_i^4 + \sum_{i\neq j} Y_i^2 Y_j^2\right] = \textstyle\sum_{i=1}^{d} \mu_{i,4} + \sum_{i\neq j} \mu_{i,2}\mu_{j,2}, \text{ and}\\
\sigma^2\left(\|\boldsymbol{Y}_d\|^2\right) &= \mathbf{E}[\|\boldsymbol{Y}_d\|^4] - \mathbf{E}[\|\boldsymbol{Y}_d\|^2]^2 = \mathbf{E}[\|\boldsymbol{Y}_d\|^4] - \left(\textstyle\sum_i \mu_{i,2}\right)^2 =\\
&= \mathbf{E}[\|\boldsymbol{Y}_d\|^4] - \left(\textstyle\sum_i \mu_{i,2}^2 + \sum_{i\neq j} \mu_{i,2}\mu_{j,2}\right) = \textstyle\sum_i \mu_{i,4} + \sum_i \mu_{i,2}^2 = d(\tilde{\boldsymbol{\mu}}_{\boldsymbol{4}} + \tilde{\boldsymbol{\mu}}^{\boldsymbol{2}}_{\boldsymbol{2}}).
\end{aligned}$$

The other expressions can be obtained in an analogous manner.

Moreover, by using the same line of reasoning of Proposition 22 it can be shown that:

(**P37.8**) $\frac{\boldsymbol{\mu_3} \cdot \boldsymbol{x_d}}{\|\boldsymbol{x}_d\|^2} \to \frac{\boldsymbol{\mu_3} \cdot \boldsymbol{\mu}}{\boldsymbol{\mu_2}} = 0$, and (**P37.9**) $\frac{\boldsymbol{\mu_2} \cdot \boldsymbol{x^2_d}}{\|\boldsymbol{x}_d\|^2} \to \frac{\tilde{\boldsymbol{\mu}}^{\boldsymbol{2}}_{\boldsymbol{2}}}{\tilde{\boldsymbol{\mu}}_{\boldsymbol{2}}}$,

and, hence, (**P37.7'**) can be reformulated only in terms of the squared norm of $\boldsymbol{x}_d$:

(**P37.7'**) $\|\boldsymbol{x}_d - \boldsymbol{Y}_d\|^2 \simeq \mathcal{N}\left(\|\boldsymbol{x}_d\|^2 + d\tilde{\boldsymbol{\mu}}_{\boldsymbol{2}},\ d(\tilde{\boldsymbol{\mu}}_{\boldsymbol{4}} - \tilde{\boldsymbol{\mu}}^{\boldsymbol{2}}_{\boldsymbol{2}}) + 4\frac{\tilde{\boldsymbol{\mu}}^{\boldsymbol{2}}_{\boldsymbol{2}}}{\tilde{\boldsymbol{\mu}}_{\boldsymbol{2}}}\|\boldsymbol{x}_d\|^2\right)$.

Expressions of Sections 4.2 and 4.3 can be obtained by exploiting the above ones and by following the same line of reasoning. For completeness, we report the final expression of the number of k-occurrences:

(**P37.10**) $N_{\varrho}^{\infty}(z) = \Phi\left(\frac{\Phi^{-1}(\varrho)\sqrt{\tilde{\boldsymbol{\mu}}_{\boldsymbol{4}} + 3\tilde{\boldsymbol{\mu}}^{\boldsymbol{2}}_{\boldsymbol{2}}} - z\sqrt{\tilde{\boldsymbol{\mu}}_{\boldsymbol{4}} - \tilde{\boldsymbol{\mu}}^{\boldsymbol{2}}_{\boldsymbol{2}}}}{2\sqrt{\tilde{\boldsymbol{\mu}}^{\boldsymbol{2}}_{\boldsymbol{2}}}}\right)$.

■

References

Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional spaces. In *Proceedings of the International Conference on Database Theory (ICDT)*, pages 420–434, London, UK, 4-6 January 2001.

Alexandr Andoni. *NN search : the old, the new, and the impossible.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2009.

Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 459–468, Berkeley, California, USA, 21-24 October 2006.

Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1):117–122, 2008.

Fabrizio Angiulli. Concentration free outlier detection. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, pages 3–19, Skopje, Macedonia, 18-22 September 2017.

Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *Journal of the ACM*, 45(6):891–923, 1998.

Robert B. Ash and Catherine A. Doléans-Dade. *Probability & Measure Theory*. Academic Press, New York, NY, USA, 1999.

Jean-Julien Aucouturier and Francois Pachet. A scale-free distribution of false positives for a large class of audio similarity measures. *Pattern Recognition*, 41(1):272–284, 2007.

Albert-László Barabási and Márton Pósfai. *Network science*. Cambridge University Press, England, 2016.

Richard E. Bellmann. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, New Jersey, USA, 1961.

Kevin S. Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is "nearest neighbor" meaningful? In *Proceedings of the International Conference on Database Theory (ICDT)*, pages 217–235, Jerusalem, Israel, 10-12 January 1999.

Ginestra Bianconi and Albert-László Barabási. Competition and multiscaling in evolving networks. *Europhysics Letters*, 54, 2001.

Edgar Chávez, Gonzalo Navarro, Ricardo Baeza-Yates, and José Luis Marroquín. Searching in metric spaces. *ACM Computing Surveys*, 33(3):273–321, 2001.

Otfried Cheong, Antoine Vigneron, and Juyoung Yon. Reverse nearest neighbor queries in fixed dimension. *Internatinal Journal of Computational Geometry & Applications*, 21(2): 179–188, 2011.

Belur V. Dasarathy. *Nearest neighbor (NN) norms: NN pattern classification techniques*. IEEE Computer Society Press, Los Alamitos, CA, USA, 1990.

Pierre Demartines. *Analyse de Données par Réseaux de Neurones Auto-Organisés*. PhD thesis, Institut National Polytechnique de Grenoble, France, 1994.

George Doddington, Walter Liggett, Alvin Martin, Mark Przybocki, and Douglas Reynolds. Sheep, goats, lambs and wolves: A statistical analysis of speaker performance in the NIST 1998 speaker recognition evaluation. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, volume 4, pages 1351–1354, 1998.

Richar O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. John Wiley & Sons, New York, NY, USA, 2000.

Robert J. Durrant and Ata Kabán. When is 'nearest neighbour' meaningful: A converse theorem and implications. *Journal of Complexity*, 25(4):385–397, 2009.

Paul Erdős and Alfrèd Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

William Feller. *An Introduction to Probability Theory and Its Applications*, volume 2. Wiley, New York, 3rd edition, 1971.

Damien François, Vincent Wertz, and Michel Verleysen. The concentration of fractional distances. *IEEE Transactions on Knowledge and Data Engineering*, 19(7):873–886, 2007.

Chris Giannella. New instability results for high-dimensional nearest neighbor search. *Information Processing Letters*, 109(19):1109–1113, 2009.

Daniele Granata and Vincenzo Carnevale. Accurate estimation of the intrinsic dimension using graph distances: Unraveling the geometric complexity of datasets. *Scientific Reports*, 6, 2016.

Peter Grassberger and Itamar Procaccia. Measuring the strangeness of strange attractors. *Physica D Nonlinear Phenomena*, 9:189–208, 1983.

Sariel Har-Peled, Piotr Indyk, and Rajeev Motwani. Approximate nearest neighbor: Towards removing the curse of dimensionality. *Theory of Computing*, 8(1):321–350, 2012.

Ville Hautamäki, Ismo Kärkkäinen, and Pasi Fränti. Outlier detection using k-nearest neighbour graph. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 430–433, Cambridge, UK, 23-26 August 2004.

Junfeng He, Sanjiv Kumar, and Shih-Fu Chang. On the difficulty of nearest neighbor search. In *Proceedings of the International Conference on Machine Learning, (ICML)*, Edinburgh, Scotland, 26 June-1 July 2012.

Alexander Hinneburg, Charu C. Aggarwal, and Daniel A. Keim. What is the nearest neighbor in high dimensional spaces? In *Proceedings of International Conference on Very Large Data Bases (VLDB)*, pages 506–515, Cairo, Egypt, 10-14 September 2000.

Chih-Ming Hsu and Ming-Syan Chen. On the design and applicability of distance functions in high-dimensional data space. *IEEE Transactions on Knowledge and Data Engineering*, 21(4):523–536, 2009.

Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the ACM Symposium on the Theory of Computing (STOC)*, pages 604–613, Dallas, Texas, USA, 23-26 May 1998.

Norman L. Johnson, Samuel Kotz, and N. Balakrishnan. *Continuous Univariate Distributions*. Wiley, 1994.

Ata Kabán. Non-parametric detection of meaningless distances in high dimensional data. *Statistics and Computing*, 22(2):375–385, 2012.

Flip Korn and S. Muthukrishnan. Influence sets based on reverse nearest neighbor queries. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 201–212, Dallas, Texas, USA, 16-18 May 2000.

Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

Thomas Low, Christian Borgelt, Sebastian Stober, and Andreas Nürnberger. The hubness phenomenon: Fact or artifact? In *Towards Advanced Data Analysis by Combining Soft Computing and Statistics*, pages 267–278. Springer, Berlin, Heidelberg, 2013.

Laurence T. Maloney. Nearest neighbor analysis of point processes: Simulations and evaluations. *Journal of Mathematical Psychology*, 27(3):251–260, 1983.

Charles M. Newman and Yosef Rinott. Nearest neighbors and Voronoi volumes in high-dimensional point processes with various distance functions. *Advances in Applied Probability*, 17(4):794–809, 1985.

Charles M. Newman, Yosef Rinott, and Amos Tversky. Nearest neighbors and Voronoi regions in certain point processes. *Advances in Applied Probability*, 15(4):726–751, 1983.

Luca Oberto and Francesca Pennecchi. Estimation of the modulus of a complex-valued quantity. *Metrologia*, 43(6):531–538, 2006.

Donald B. Owen. A table of normal integrals. *Communications in Statistics: Simulation and Computation*, 89:389–419, 1980.

Vladimir Pestov. On the geometry of similarity search: Dimensionality curse and concentration of measure. *Information Processing Letters*, 73(1-2):47–51, 2000.

Franco P. Preparata and Michael I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag New York, Inc., New York, NY, USA, 1985.

Milos Radovanovic, Alexandros Nanopoulos, and Mirjana Ivanovic. Nearest neighbors in high-dimensional data: the emergence and influence of hubs. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 865–872, Montreal, Quebec, Canada, 14-18 June 2009.

Milos Radovanovic, Alexandros Nanopoulos, and Mirjana Ivanovic. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11: 2487–2531, 2010.

Milos Radovanovic, Alexandros Nanopoulos, and Mirjana Ivanovic. Reverse nearest neighbors in unsupervised distance-based outlier detection. *IEEE Transactions on Knowledge and Data Engineering*, 27(5):1369–1382, 2015.

Gregory Shakhnarovich, Trevor Darrell, and Piotr Indyk, editors. *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*. MIT Press, March 2006.

Amit Singh, Hakan Ferhatosmanoglu, and Ali Saman Tosun. High dimensional reverse nearest neighbor queries. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, pages 91–98, New Orleans, Louisiana, USA, 2-8 November 2003.

Yufei Tao, Dimitris Papadias, Xiang Lian, and Xiaokui Xiao. Multidimensional reverse k NN search. *The VLDB Journal*, 16(3):293–316, 2007.

Nenad Tomasev. Taming the empirical hubness risk in many dimensions. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, pages 891–899, Vancouver, BC, Canada, 30 April-2 May 2015.

Nenad Tomasev and Krisztian Buza. Hubness-aware kNN classification of high-dimensional data in presence of label noise. *Neurocomputing*, 160:157–172, 2015.

Nenad Tomasev, Milos Radovanovic, Dunja Mladenic, and Mirjana Ivanovic. The role of hubness in clustering high-dimensional data. *IEEE Transactions on Knowledge and Data Engineering*, 26(3):739–751, 2014.

Amos Tversky and John Wesley Hutchinson. Nearest neighbor analysis of psychological spaces. *Psychological Review*, 93(1):3–22, 1986.

Amos Tversky, Yosef Rinott, and Charles M. Newman. Nearest neighbor analysis of point processes: Applications to multidimensional scaling. *Journal of Mathematical Psychology*, 27(3):235–250, 1983.

Laurens van der Maaten, Eric Postma, and Jaapvan den Herik. Dimensionality reduction: A comparative review. Technical Report TiCC-TR 2009-005, Tilburg University, The Netherlands, 2009.

Roger Weber, Hans-Jörg Schek, and Stephen Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, pages 194–205, New York, NY, USA, 24-27 August 1998.

Graham J. Williams, Rohan A. Baxter, Hongxing He, Simon Hawkins, and Lifang Gu. A comparative study of RNN for outlier detection in data mining. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 709–712, Maebashi City, Japan, 9-12 December 2002.

Shiyu Yang, Muhammad A. Cheema, Xuemin Lin, and Wei Wang. Reverse k nearest neighbors query processing: Experiments and analysis. *Proceedings of the VLDB Endowment (PVLDB)*, 8(5):605–616, 2015.

Yi-Ching Yao and Gordon Simons. A large-dimensional independent and identically distributed property for nearest neighbor counts in poisson processes. *Annals of Applied Probability*, 6(2):561–571, 1996.

Journal of Machine Learning Research 18 (2018) 1-33 Submitted 8/17; Published 4/18

Convergence of Unregularized Online Learning Algorithms

Yunwen Lei yunwelei@cityu.edu.hk
Shenzhen Key Laboratory of Computational Intelligence
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, 518055, China
and
Department of Mathematics
City University of Hong Kong
Kowloon, Hong Kong, China

Lei Shi leishi@fudan.edu.cn
School of Mathematical Sciences
Shanghai Key Laboratory for Contemporary Applied Mathematics
Fudan University
Shanghai, 200433, China

Zheng-Chu Guo guozhengchu@zju.edu.cn
School of Mathematical Sciences
Zhejiang University
Hangzhou, 310027, China

Editor: Andreas Christmann

Abstract

In this paper we study the convergence of online gradient descent algorithms in reproducing kernel Hilbert spaces (RKHSs) without regularization. We establish a sufficient condition and a necessary condition for the convergence of excess generalization errors in expectation. A sufficient condition for the almost sure convergence is also given. With high probability, we provide explicit convergence rates of the excess generalization errors for both averaged iterates and the last iterate, which in turn also imply convergence rates with probability one. To our best knowledge, this is the first high-probability convergence rate for the last iterate of online gradient descent algorithms in the general convex setting. Without any boundedness assumptions on iterates, our results are derived by a novel use of two measures of the algorithm's one-step progress, respectively by generalization errors and by distances in RKHSs, where the variances of the involved martingales are cancelled out by the descent property of the algorithm.

Keywords: Learning theory, Online learning, Convergence analysis, Reproducing kernel Hilbert space

1. Introduction

Online gradient descent is a scalable method able to tackle large-scale data arriving in a sequential manner (Zhang, 2004; Kivinen et al., 2004; Duchi and Singer, 2009; Dieuleveut and Bach, 2016), which is becoming ubiquitous within the big data era. As a first-order method, it iteratively builds an unbiased estimate of the true gradient upon the arrival of a new example and uses this information to guide the learning process (Zinkevich, 2003; Zhang, 2004). As verified by theoretical and empirical analysis, online gradient descent enjoys comparable performance as compared to its batch counterpart such as gradient descent (Zhang, 2004; Yao, 2010; Shalev-Shwartz et al., 2011), while attaining a great computational speed-up since its gradient calculation involves only a single example. As a comparison, the gradient calculation in gradient descent requires to traverse all training examples. Recently, online gradient descent has received renewed attention due to the wide applications of its stochastic analogue, i.e., stochastic gradient descent, in training deep neural networks (Bottou, 1991; Ngiam et al., 2011; Sutskever et al., 2013).

In this paper, we are interested in the setting that training examples $\{z_t = (x_t, y_t)\}_{t\in\mathbb{N}}$ are sequentially and identically drawn from a probability measure ρ defined in the sample space $\mathcal{Z} = \mathcal{X}\times\mathcal{Y}$, where $\mathcal{X} \subset \mathbb{R}^d$ is the input space and $\mathcal{Y} \subset \mathbb{R}$ is the output space. We focus on the nonparametric setting, where the learning process is implemented in a reproducing kernel Hilbert space (RKHS) H_K associated with a Mercer kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which is assumed to be continuous, symmetric and positive semi-definite. The space H_K is defined as the completion of the linear span of the set of functions $\{K_x(\cdot) := K(x, \cdot) : x \in \mathcal{X}\}$ with the inner producing satisfying the reproducing property $f(x) = \langle f, K_x\rangle$ for any $x \in \mathcal{X}$ and $f \in H_K$. In this setting, the use of Mercer kernels provides a unifying way to measure similarities between pairs of objects (Cortes and Vapnik, 1995; Müller et al., 2001; Steinwart, 2001; Schölkopf and Smola, 2001), which turns out to be a key to the great success of kernel methods in many practical learning problems. We wish to build a prediction rule $f \in H_K$ after seeing a sequence of training examples, the performance of which at an example (x, y) can be quantitatively measured by a loss function $\phi : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}_+$ as $\phi(y, f(x))$. With a sequence $\{\eta_t\}_{t\in\mathbb{N}}$ of positive step sizes and $f_1 = 0$, online gradient descent is a realization of learning schemes by keeping a sequence of iterates as follows

$$f_{t+1} = f_t - \eta_t\phi'(y_t, f_t(x_t))K_{x_t}, \quad \forall t \in \mathbb{N}, \tag{1.1}$$

where ϕ' denotes the derivative of ϕ with respect to the second argument. Although our focus is on the nonparametric setting, it should be mentioned that the above algorithm also recovers the parametric case in which the kernel is taken to be the linear kernel with $K_x(x') = \langle x, x'\rangle, \forall x, x' \in \mathcal{X}$, to which our results also apply.

Despite its widespread applications, the theoretical understanding of the online gradient descent algorithms is still not satisfactory in the following three aspects. Firstly, boundedness assumptions on the iterates are often imposed in the literature, which may be violated in practical implementations if the underlying domain is not bounded. Although a projection of iterates onto a bounded domain guarantees the boundedness assumption, the projection operator may be time-consuming and this introduces an additional challenging problem of tuning the size of the domain. Secondly, most of the theoretical results are stated in expectation, while we are sometimes more interested in either almost sure convergence or convergence rates with high probability. Indeed, an algorithm may suffer from a high variability and should be used with caution if neither almost sure convergence nor high-probability bounds hold (Shamir and Zhang, 2013). In particular, an almost sure convergence is still lacking for online gradient descent algorithms applied to general convex problems (Ying and Zhou, 2017). Lastly, most existing convergence rates are stated for some average of iterates. Though taking average of iterates can improve the robustness

of the solution (Nemirovski et al., 2009), it can either destroy the sparsity of the solution which is crucial for a proper interpretation of models in many applications, or slow down the training speed in practical implementations (Rakhlin et al., 2012).

In this paper, we aim to take a further step to tackle the above mentioned problems. We establish a general sufficient condition and a necessary condition on the step sizes for the convergence of online gradient descent algorithms in expectation. With Doob's martingale convergence theorem and the Borel-Cantelli lemma, a sufficient condition for the almost sure convergence and explicit convergence rates with probability one are also established. Furthermore, we present high-probability bounds for both averaged iterates and the last iterate of online gradient descent algorithms. To our best knowledge, this is the first high-probability convergence rate for the last iterate of online gradient descent algorithms in the general convex setting. Our analysis does not impose any boundedness assumptions on the iterates. Indeed, we show that, although implemented in an unbounded domain, the iterates produced by (1.1) fall into a bounded domain with high probability (up to logarithmic factors). Our analysis is performed by viewing the one-step progress of online gradient descent algorithms from different yet unified perspectives: one in terms of generalization errors and one in terms of RKHS distances. For both viewpoints, we relate the one-step progress to a martingale difference sequence and a negative term due to the descent nature of the algorithm. Our novelty is to show that the dominant variance term appearing in the application of a Bernstein-type inequality to these martingales can be cancelled out by the negative terms in the one-step progress inequalities. Both viewpoints of the one-step progress are indispensable in our analysis.

The remaining parts of this paper are organized as follows. We present main results in Section 2. Discussions and comparisons with related work are given in Section 3. The proofs of main results are given in Section 4.

2. Main Results

Our convergence rates are stated for generalization errors, which, for a prediction rule $f : \mathcal{X} \to \mathbb{R}$, are defined as the expected error $\mathcal{E}(f) = \int_{\mathcal{Z}} \phi(y, f(x))d\rho$ incurred from using f to perform prediction. Our analysis requires to impose mild assumptions on the loss functions.

Assumption 1 *We assume the loss function $\phi : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}_+$ is convex and differentiable with respect to the second argument. Let $\alpha \in (0, 1]$ and $L > 0$ be two constants. We assume that the gradients of ϕ are (α, L)-Hölder continuous in the sense*

$$|\phi'(y, s) - \phi'(y, \tilde{s})| \le L|s - \tilde{s}|^\alpha, \quad \forall s, \tilde{s} \in \mathbb{R}, \forall y \in \mathcal{Y}. \tag{2.1}$$

We say ϕ is smooth if it satisfies (2.1) with $\alpha = 1$. Loss functions satisfying Assumption 1 are wildly used in machine learning. Smooth loss functions include the least squares loss $\phi(y, a) = \frac{1}{2}(y-a)^2$ and the Huber loss $\phi(y, a) = \frac{1}{2}(y-a)^2$ if $|y-a| \le 1$ and $|y-a|-\frac{1}{2}$ otherwise for regression, as well as the logistic loss $\phi(y, a) = \log(1 + \exp(-ya))$ and the quadratically smoothed hinge loss $\phi(y, a) = \max\{0, 1 - ya\}^2$ for classification (Zhang, 2004). If $p \in (1, 2]$, both the p-norm hinge loss $\phi(y, a) = \max\{0, 1 - ya\}^p$ for classification and the p-th power absolute distance $\phi(y, a) = |y - a|^p$ for regression satisfy (2.1) with $\alpha = p - 1$ (Chen et al., 2004; Steinwart and Christmann, 2008).

Throughout this paper, we assume that a minimizer $f_H = \arg\min_{f \in H_K} \mathcal{E}(f)$ exists in H_K. We also assume

$$\max\big\{\sup_{y \in \mathcal{Y}} \phi(y, 0), \sup_{z \in \mathcal{Z}} \phi(y, f_H(x))\big\} < \infty \text{ and } \kappa := \sup_{x \in \mathcal{X}} \sqrt{K(x, x)} < \infty,$$

which is satisfied if the sample space $\mathcal{Z}$ is bounded. Denote $\|\cdot\|$ as the norm in H_K. We always use the notation $A_t = \mathbb{E}[\mathcal{E}(f_t)] - \mathcal{E}(f_H)$ and $\hat{A}_t = \mathcal{E}(f_t) - \mathcal{E}(f_H), \forall t \in \mathbb{N}$ for brevity, which are referred to as the expected excess generalization errors and excess generalization errors, respectively.

In the following, we present the main results of this paper. We consider three types of convergence: convergence in expectation, almost sure convergence and convergence rates with high probability.

2.1 Convergence in Expectation

The first part of our main results to be proved in Section 4.1 establishes a general sufficient condition (Theorem 1) and a necessary condition (Theorem 2) on the step size sequence $\{\eta_t\}_{t \in \mathbb{N}}$ for the convergence of A_t to zero.

Theorem 1 *Let $\{f_t\}_{t \in \mathbb{N}}$ be the sequence produced by (1.1) and suppose Assumption 1 holds with $\alpha \in (0, 1]$. If*

$$\sum_{t=1}^{\infty} \eta_t = \infty \quad \text{and} \quad \lim_{t \to \infty} \eta_t^\alpha \sum_{k=1}^{t} \eta_k^2 = 0, \tag{2.2}$$

then $\lim_{t \to \infty} \mathbb{E}[\mathcal{E}(f_t)] - \mathcal{E}(f_H) = 0$.

Theorem 2 *Let $\{f_t\}_{t \in \mathbb{N}}$ be the sequence produced by (1.1). Suppose that for any $y \in \mathcal{Y}$, the function $\phi(y, \cdot) : \mathbb{R} \to \mathbb{R}_+$ is convex and its derivative $\phi'(y, \cdot)$ is $(1, L)$-Hölder continuous. Assume that the step size sequence satisfies $\eta_t \le 1/(6L\kappa^2), \forall t \in \mathbb{N}$ and $\mathcal{E}(f_1) \ne \mathcal{E}(f_H)$. If $\lim_{t \to \infty} \mathbb{E}[\mathcal{E}(f_t)] = \mathcal{E}(f_H)$, then $\sum_{t=1}^{\infty} \eta_t = \infty$.*

Remark 3 *We now illustrate the above theorems by considering the polynomially decaying step sizes $\eta_t = \eta_1 t^{-\theta}, t \in \mathbb{N}, \theta \ge 0$. The condition $\sum_{t=1}^{\infty} \eta_t = \infty$ requires $\theta \le 1$, while the condition $\lim_{t \to \infty} \eta_t^\alpha \sum_{k=1}^{t} \eta_k^2 = 0$ requires $\theta > \frac{1}{2+\alpha}$. Therefore, Theorem 1 shows that the iteration scheme (1.1) with $\eta_t = \eta_1 t^{-\theta}$ and $\theta \in \big(\frac{1}{2+\alpha}, 1\big]$ guarantees the convergence of $\{A_t\}_{t \in \mathbb{N}}$. Theorem 2 shows that the condition $\theta \le 1$ is also necessary for the convergence.*

2.2 Almost Sure Convergence

The second part of our main results focuses on a sufficient condition (Theorem 4) for the almost sure convergence of $\{\hat{A}_t\}_{t \in \mathbb{N}}$ to zero and convergence rates with probability 1 (Theorem 6). The proofs of results in this section can be found in Section 4.2.

Theorem 4 *Let $\{f_t\}_{t \in \mathbb{N}}$ be the sequence given by (1.1). If Assumption 1 holds with $\alpha \in (0, 1]$ and the step size sequence satisfies*

$$\sum_{t=1}^{\infty} \eta_t = \infty \quad \text{and} \quad \sum_{t=1}^{\infty} \eta_t^{1+\alpha} < \infty, \tag{2.3}$$

then $\lim_{t\to\infty}\mathcal{E}(f_t)=\mathcal{E}(f_H)$ *almost surely.*

Remark 5 *According to Theorem 4, we know that* $\{\hat{A}_t\}_{t\in\mathbb{N}}$ *would converge almost surely to* 0 *if we consider either the step sizes* $\eta_t=\eta_1 t^{-\theta}$ *with* $\theta\in(\frac{1}{1+\alpha},1]$ *or the step sizes* $\eta_t=\eta_1(t\log^\beta t)^{-\frac{1}{1+\alpha}}$ *with* $\beta>1$*. Specifically, if the loss function is smooth, then we can choose either* $\eta_t=\eta_1 t^{-\theta}$ *with* $\theta\in(\frac{1}{2},1]$ *or* $\eta_t=\eta_1\big(t\log^\beta t\big)^{-\frac{1}{2}}$ *with* $\beta>1$ *to guarantee the convergence of the algorithm* (1.1) *almost surely in the sense of generalization errors.*

Theorem 6 *Suppose that Assumption 1 holds with* $\alpha\in(0,1]$*. Let* $\{f_t\}_{t\in\mathbb{N}}$ *be the sequence given by* (1.1) *with* $\eta_t=\eta_1 t^{-\theta},\theta\in(\frac{1}{\alpha+1},1)$ *and* $\eta_1\le\frac{1}{A\kappa^2}$ *(*A *is defined in* (4.27)*). Then for any* $\epsilon>0$*,*

$$\lim_{t\to\infty} t^{\min\{(1-\theta),(\alpha+1)\theta-1\}-\epsilon}\hat{A}_t=0 \text{ almost surely.} \tag{2.4}$$

Specifically, if we choose $\theta=\frac{2}{2+\alpha}$*, then* $\lim_{t\to\infty}t^{\frac{\alpha}{2+\alpha}-\epsilon}\hat{A}_t=0$ *almost surely.*

2.3 Convergence Rates with High Probability

The last part of our main results is on high-probability bounds for the excess generalization errors, the proof of which is given in Section 4.3. With high probability, Theorem 7 establishes the boundedness (up to logarithmic factors) of the weighted summation $\sum_{t=1}^T\eta_t\hat{A}_t$, from which the decay rate of the excess generalization error $\mathcal{E}(\bar{f}_T^\eta)-\mathcal{E}(f_H)$ associated to a weighted average of the iterates $\bar{f}_T^\eta:=\frac{\sum_{t=1}^T\eta_t f_t}{\sum_{t=1}^T\eta_t}$ follows directly.

Theorem 7 *Let* $\{f_t\}_{t\in\mathbb{N}}$ *be the sequence given by* (1.1)*. Suppose that Assumption 1 holds with* $\alpha\in(0,1]$*. Assume the step size sequence satisfies* $\eta_t\le\frac{1}{A\kappa^2},\eta_{t+1}\le\eta_t$ *for all* $t\in\mathbb{N}$ *and* $\sum_{t=1}^\infty\eta_t^2<\infty$*. Then, there exists a constant* $\widetilde{C}$ *independent of* T *(explicitly given in the proof) such that for any* $\delta\in(0,1)$ *the following inequality holds with probability at least* $1-\delta$

$$\sum_{t=1}^T\eta_t\big[\mathcal{E}(f_t)-\mathcal{E}(f_H)\big]\le\widetilde{C}\log^{\frac{3}{2}}\frac{2T}{\delta}\quad\text{and}\quad\mathcal{E}(\bar{f}_T^\eta)-\mathcal{E}(f_H)\le\frac{\widetilde{C}\log^{\frac{3}{2}}\frac{2T}{\delta}}{\sum_{t=1}^T\eta_t}. \tag{2.5}$$

Remark 8 *For the step size sequence* $\eta_t=\eta_1 t^{-\theta},\theta>\frac{1}{2}$*, Theorem 7 implies that* $\mathcal{E}(\bar{f}_T^\eta)-\mathcal{E}(f_H)=O\big(T^{\theta-1}\log^{\frac{3}{2}}\frac{T}{\delta}\big)$ *with probability at least* $1-\delta$*. If we consider* $\eta_t=\eta_1\big(t\log^\beta(et)\big)^{-\frac{1}{2}}$ *with* $\beta>1$*, then with probability* $1-\delta$ *we have* $\mathcal{E}(\bar{f}_T^\eta)-\mathcal{E}(f_H)=O\big(T^{-\frac{1}{2}}\log^{\frac{3+\beta}{2}}\frac{T}{\delta}\big)$*.*

A key feature of Theorem 7 distinguishing it from the existing results is that it avoids boundedness assumptions on the iterates, which are always imposed in the literature (Nemirovski et al., 2009; Duchi et al., 2010). Indeed, an essential ingredient in proving Theorem 7 is to show that $\{f_t\}_{t\in\mathbb{N}}$ produced by (1.1) would fall into a bounded ball of H_K (up to logarithmic factors) with high probability, as shown in the following proposition.

Proposition 9 *Suppose assumptions in Theorem 7 hold. Then, there exists a constant* $\bar{C}\ge1$ *independent of* T *(explicitly given in the proof) such that for any* $\delta\in(0,1)$ *the following inequality holds with probability at least* $1-\delta$

$$\max_{1\le t\le T}\|f_t-f_H\|^2\le\bar{C}\log\frac{T}{\delta}.$$

A key ingredient to prove Proposition 9 is to establish the following one-step progress inequality in terms of the RKHS distances (see (4.37))

$$\|f_{t+1}-f_H\|^2\le\|f_t-f_H\|^2+C\eta_t^2+2\eta_t\big(\mathcal{E}(f_H)-\mathcal{E}(f_t)\big)+\xi_t,$$

where C is a constant and $\{\xi_t\}_{t\in\mathbb{N}}$ is a Martingale difference sequence. Our novelty in applying a Bernstein-type inequality to control the martingale $\sum_{t=1}^T\xi_t$ is to show that the associated variances can be cancelled out by the negative term $2\sum_{t=1}^T\eta_t\big(\mathcal{E}(f_H)-\mathcal{E}(f_t)\big)$ (see (4.38) and the last inequality of Proposition 23). Although Theorem 7 only considers the behavior of the weighted average $\bar{f}_T^\eta$ of iterates, it is possible to establish similar convergence rates for the uniform average of iterates $\bar{f}_T:=\frac{1}{T}\sum_{t=1}^T f_t$ (Proposition 24).

Theorem 10 establishes a general high-probability bound for the excess generalization error of the last iterate in terms of the step size sequence.

Theorem 10 *Suppose that the assumptions in Theorem 7 hold. Then, there exists a constant* $\widetilde{C}'$ *independent of* T *(explicitly given in the proof) such that for any* $\delta\in(0,1)$ *the following inequality holds with probability at least* $1-\delta$

$$\mathcal{E}(f_{T+1})-\mathcal{E}(f_H)\le\widetilde{C}'\max\Big\{\Big[\sum_{t=\lfloor\frac{T}{2}\rfloor}^T\eta_t\Big]^{-1},\eta_{\lfloor\frac{T}{2}\rfloor},\sum_{t=\lfloor\frac{T}{2}\rfloor}^T\eta_t^{1+\alpha}\Big\}\log^2\frac{3T}{\delta}, \tag{2.6}$$

where $\lfloor\frac{T}{2}\rfloor$ *denotes the largest integer not greater than* $\frac{T}{2}$*.*

To establish high-probability error bounds for the last iterate of online gradient descent algorithm is an interesting problem which is not well studied, to our best knowledge, in the general convex setting. The key ingredient in our analysis is the following one-step progress inequality in terms of generalization errors (see (4.47))

$$\hat{A}_{t+1}\le\hat{A}_t-\eta_t\|\nabla\mathcal{E}(f_t)\|^2+\bar{\xi}_t+C\eta_t^{1+\alpha},$$

where C is a constant and $\{\bar{\xi}_t\}$ is a martingale difference sequence. A key observation of our analysis is that the variance of the martingale $\sum_{t=1}^T\bar{\xi}_t$ can be cancelled out by the negative term $-\sum_{t=1}^T\eta_t\|\nabla\mathcal{E}(f_t)\|^2$ in the above one-step progress inequality (see (4.48) and (4.52)), paving the way for the application of a Bernstein-type inequality for martingales.

We can derive explicit convergence rates in Corollary 11 by considering polynomially decaying step sizes in Theorem 10.

Corollary 11 *Let* $\{f_t\}_{t\in\mathbb{N}}$ *be the sequence given by* (1.1) *with* $\eta_t=\eta_1 t^{-\theta},\theta\in(\frac{1}{2},1)$ *and* $\eta_1\le\frac{1}{A\kappa^2}$ *If Assumption 1 holds and* $\delta\in(0,1)$*, then the following inequality holds with probability* $1-\delta$

$$\mathcal{E}(f_{T+1})-\mathcal{E}(f_H)=O\Big(T^{\max\{\theta-1,1-(1+\alpha)\theta\}}\log^2\frac{T}{\delta}\Big).$$

If we choose $\theta=\frac{2}{2+\alpha}$*, then with probability at least* $1-\delta$ *we derive* $\mathcal{E}(f_{T+1})-\mathcal{E}(f_H)=O\Big(T^{-\frac{\alpha}{2+\alpha}}\log^2\frac{T}{\delta}\Big)$*.*

Remark 12 *It should be mentioned that, unlike Theorem 7, the convergence rates in Corollary 11 depend on the smoothness parameter α and are not able to attain the minimax optimal convergence rate $O(T^{-\frac{1}{2}})$ (Agarwal et al., 2009). Indeed, for smooth loss functions, Corollary 11 establishes the convergence rate $O\big(T^{-\frac{1}{3}}\log^2\frac{T}{\delta}\big)$ with high probability, which matches the bounds in-expectation $A_T = O(T^{-\frac{1}{3}})$ up to logarithmic factors established in Moulines and Bach (2011); Ying and Zhou (2017). It remains a challenging problem to further improve the high-probability bounds for $\hat{A}_T$.*

3. Discussions

In this section, we discuss related work on convergence of online/stochastic gradient descent algorithms from three viewpoints: convergence in expectation, almost sure convergence and convergence rates with high probability.

3.1 Related Work on Convergence in Expectation

Most studies of online gradient descent algorithms focus on convergence in expectation (Zhang, 2004; Ying and Zhou, 2006; Duchi and Singer, 2009; Shamir and Zhang, 2013; Lin et al., 2016; Hardt et al., 2016; Ying and Zhou, 2017). Convergence rates $O(T^{-\frac{1}{2}})$ were established for some averaged iterates produced by (1.1) in a parametric setting with the linear kernel $K_x = x$ (Zhang, 2004). These results were extended to online gradient descent algorithms in RKHSs with the specific least squares loss function (Ying and Pontil, 2008; Dieuleveut and Bach, 2016; Guo and Shi, 2017), and online mirror descent algorithms performing updates in Banach spaces (Duchi et al., 2010). Under boundedness assumptions on the iterates and (sub)gradients, the convergence rate $O(T^{-\frac{1}{2}}\log T)$ was established for the expected excess generalization error of the last iterate (Shamir and Zhang, 2013). Recently, a general condition on the step sizes as (2.3) was established for the convergence of the algorithm (1.1), in the sense $\lim_{t\to\infty} A_t = 0$, with loss functions satisfying Assumption 1 (Ying and Zhou, 2017). This sufficient condition is stricter than our condition (2.2). To see this clearly, we consider the polynomially decaying step sizes $\eta_t = \eta_1 t^{-\theta}$, for which the condition (2.3) requires $\theta \in (\frac{1}{1+\alpha}, 1]$ while our condition (2.2) requires $\theta \in (\frac{1}{2+\alpha}, 1]$. Furthermore, our discussion also implies a necessary condition for the convergence in expectation.

Implemented in either a parametric or a nonparametric setting, regularized online learning algorithms have also received considerable attention (Kivinen et al., 2004; Smale and Yao, 2006; Ying and Zhou, 2006; Smale and Zhou, 2009), which differ from (1.1) by introducing a regularization term to avoid overfitting. This algorithm updates iterates as follows

$$f_{t+1} = (1-\lambda\eta_t)f_t - \eta_t\phi'(y_t, f_t(x_t))K_{x_t}, \tag{3.1}$$

where $\lambda > 0$ is a regularization parameter and the term $\lambda f_t + \phi'(y_t, f_t(x_t))K_{x_t}$ is used as an unbiased estimator of the gradient for the regularized generalization error $\mathcal{E}^\lambda(f) := \mathcal{E}(f) + \frac{\lambda}{2}\|f\|^2$ at $f = f_t$. Convergence rates in expectation can be stated for either the excess regularized generalization error $\mathcal{E}^\lambda(f_T) - \mathcal{E}^\lambda(f_\lambda)$ (Shamir and Zhang, 2013) or the RKHS distance $\|f_T - f_\lambda\|$ (Smale and Yao, 2006; Ying and Zhou, 2006; Yao, 2010), where $f_\lambda = \arg\min_{f\in H_K}\mathcal{E}^\lambda(f)$ is the minimizer of the regularized generalization error. When the loss function is smooth, a sufficient and necessary condition as

$$\lim_{t\to\infty}\eta_t = 0 \quad \text{and} \quad \sum_{t=1}^{\infty}\eta_t = \infty \tag{3.2}$$

was recently established for the convergence of $\{\mathbb{E}[\|f_t - f_\lambda\|^2]\}_{t\in\mathbb{N}}$ to zero in the parametric case (Lei and Zhou, 2017). A disadvantage of the regularization scheme (3.1) is that it requires to tune two sequences of hyper-parameters: a regularization parameter and the step sizes. As a comparison, an implicit regularization can be attained in the unregularized scheme (1.1) by tuning only the step sizes.

3.2 Related Work on Almost Sure Convergence

Existing almost sure convergence of online learning algorithm is mainly stated for the RKHS distances, which requires to impose some type of strong convexity assumption on the objective function $\mathcal{E}(f)$. In the parametric setting with the learning scheme (1.1), a sufficient condition as

$$\sum_{t=1}^{\infty}\eta_t = \infty \quad \text{and} \quad \sum_{t=1}^{\infty}\eta_t^2 < \infty$$

was established for the almost sure convergence of $\|f_t - f_H\|^2$ if the objective function attains a unique minimizer and satisfies (Bottou, 1998)

$$\inf_{\|f-f_H\|^2>\epsilon}\langle f - f_H, \nabla\mathcal{E}(f)\rangle > 0, \quad \forall\epsilon > 0,$$
$$\mathbb{E}_Z\big[\|\phi'(Y, f(X))K_X\|^2\big] \le \widetilde{A} + \widetilde{B}\|f - f_H\|^2, \quad \forall f \in H_K,$$

where $\widetilde{A}$ and $\widetilde{B}$ are two constants. This result was extended to the online mirror descent setting under some convexity assumption on the objective function measured by Bregman distances induced by the associated mirror map (Lei and Zhou, 2017). For polynomially decaying step sizes $\eta_t = \eta_1 t^{-\theta}$ with $\theta \in (0,1)$, almost sure convergence of $\|f_t - f_\lambda\|$ was shown for regularized online learning algorithms (3.1) specified to the least squares loss function (Yao, 2010). The analysis in Yao (2010) roots its foundation on the martingale decompositions of the reminders $f_t - f_\lambda$, which only holds in the least squares regularization setting. Almost sure convergence was recently studied for the randomized Kaczmarz algorithm (Lin and Zhou, 2015), which is an instantiation of (1.1) with $\phi(y,a) = \frac{1}{2}(y-a)^2$ and $K_x = x$. The analysis there heavily depends on a restricted strong convexity of the objective function in a linear subspace where the learning takes place, which can not apply to general loss functions. As compared to the above mentioned results, our almost sure convergence is stated for the excess generalization errors with general loss functions and requires no assumptions on the strong convexity of the objective function $\mathcal{E}(f)$.

3.3 Related Work on Convergence Rates with High Probability

In this section, we survey related work on convergence rates with high probability. We divide our discussions into two parts according to the convexity of the objective function.

As far as we know, all existing high-probability convergence rates of online gradient descent algorithms with general convex functions focus on some average of iterates (here we

are not interested in probabilistic bounds with a polynomial dependence on $1/\delta$). The following online projected gradient descent algorithm with $K_x = x$ was studied in Nemirovski et al. (2009); Duchi et al. (2010)

$$f_{t+1} = \mathrm{Proj}_{\widetilde{H}}\Big[f_t - \eta_t \phi'(y_t, f_t(x_t))K_{x_t}\Big], \tag{3.3}$$

where $\widetilde{H}$ is a compact subset of H_K and $\mathrm{Proj}_{\widetilde{H}}(f) = \arg\min_{\tilde{f}\in\widetilde{H}} \|f - \tilde{f}\|$ is the projection of f onto $\widetilde{H}$. Under the boundedness assumption

$$\mathbb{E}\Big[\exp\big[\|\phi'(y, f(x))K_x\|^2/G^2\big]\Big] \le \exp(1) \quad \forall f \in \widetilde{H},$$

it was shown that the weighted average $\bar{f}_T^{\eta} = \frac{\sum_{t=1}^T \eta_t f_t}{\sum_{t=1}^T \eta_t}$ of iterates produced by (3.3) with a constant step size satisfies the following inequality with probability $1-\delta$

$$\mathcal{E}(\bar{f}_T^{\eta}) - \mathcal{E}(f_H) = O\big(GDT^{-\frac{1}{2}}\log\delta^{-1}\big),$$

where $D = \sup_{f,\tilde{f}\in\widetilde{H}} \|f - \tilde{f}\|$ is the diameter of the subspace $\widetilde{H}$. Under a stronger assumption $\|\phi'(y, f(x))K_x\| \le G$ for all $(x, y) \in \mathcal{Z}, f \in \widetilde{H}$, the uniform average $\bar{f}_T = \frac{1}{T}\sum_{t=1}^T f_t$ of iterates produced by (3.3) with step sizes $\eta_t = \eta_1 t^{-\frac{1}{2}}$ was shown to enjoy the bound $\mathcal{E}(\bar{f}_T) - \mathcal{E}(f_H) = O(DGT^{-\frac{1}{2}}\log^{\frac{1}{2}}\frac{1}{\delta})$ with probability at least $1-\delta$. In comparison with these results, the convergence rates in Theorem 7 are derived without the projection step and any boundedness assumption on the gradients. Indeed, most of the efforts in proving Theorem 7 is to show $\|f_t - f_H\|^2 = O(\log\frac{T}{\delta})$ with probability at least $1-\delta$. It is implied that the possibly computationally expensive projection step can be removed without harming the behavior of the online gradient descent algorithms. Furthermore, Theorem 10 gives, to our best knowledge, the first high-probability bounds for the last iterate of online gradient descent algorithms in the general convex setting. A framework to transfer regret bounds of online learning algorithms to high-probability bounds for the uniform average of iterates was established by Cesa-Bianchi et al. (2004).

Now we review some high-probability studies for online gradient descent algorithms in the strongly convex setting, for which some results for the last iterate can be found in the literature. For the online regularized algorithm (3.1) with the least squares loss function and $\eta_t = \eta_1 t^{-\theta}, \theta \in [0, 1)$, the following inequality was derived with probability at least $1-\delta$ (Yao, 2010)

$$\|f_T - f_\lambda\|^2 = O\Big(\lambda^{-2+\frac{1}{1-\theta}}T^{-\theta}\log\frac{1}{\delta}\Big).$$

The analysis in Yao (2010) is based on an integral operator approach, which can not be extended to general loss functions. Under almost sure boundedness assumption $\|(\phi'(y_t, f_t(x_t)) + \lambda)K_{x_t}\| \le G$ for all $t \in \mathbb{N}$, the following improved bound for the last iterate of (3.1) with general loss functions and step sizes $\eta_t = \eta_1(t\lambda)^{-1}$ was established with probability at least $1-\delta$ (Rakhlin et al., 2012)

$$\|f_T - f_\lambda\|^2 = O\Big(G^2\lambda^{-2}T^{-1}\log\frac{\log T}{\delta}\Big). \tag{3.4}$$

Although this bound enjoys a tight dependence on T, its dependence on the regularization parameter λ is suboptimal. To make a clear comparison between this result and ours, we consider here the specific least squares loss function and assume that the regression function $f_\rho(x) := \mathbb{E}[Y|X = x]$ belongs to H_K. In this case, Lemma 13 translates (3.4) to the following high-probability bounds on excess generalization errors

$$\mathcal{E}(f_T) + \frac{\lambda}{2}\|f_T\|^2 = \mathcal{E}(f_\lambda) + \frac{\lambda}{2}\|f_\lambda\|^2 + O\Big(G^2\lambda^{-2}T^{-1}\log\frac{\log T}{\delta}\Big). \tag{3.5}$$

The assumption $f_\rho \in H_K$ implies $D(\lambda) := \mathcal{E}(f_\lambda) - \mathcal{E}(f_\rho) + \frac{\lambda}{2}\|f_\lambda\|^2 = O(\lambda)$ (Cucker and Zhou, 2007) and therefore (3.5) reads as

$$\begin{aligned}\mathcal{E}(f_T) - \mathcal{E}(f_\rho) &= \Big(\mathcal{E}(f_T) - \mathcal{E}(f_\lambda) - \frac{\lambda}{2}\|f_\lambda\|^2\Big) + \Big(\mathcal{E}(f_\lambda) - \mathcal{E}(f_\rho) + \frac{\lambda}{2}\|f_\lambda\|^2\Big)\\ &= O\Big(G^2\lambda^{-2}T^{-1}\log\frac{\log T}{\delta}\Big) + O(\lambda).\end{aligned}$$

If we choose $\lambda = c\big(G^2T^{-1}\log\frac{\log T}{\delta}\big)^{\frac{1}{3}}$ for a constant $c > 0$, then the above inequality translates to $\mathcal{E}(f_T) - \mathcal{E}(f_\rho) = O\Big(\big(G^2T^{-1}\log\frac{\log T}{\delta}\big)^{\frac{1}{3}}\Big)$, which matches our convergence rates up to logarithmic factors. Note that the regularization parameter λ needs to be tuned according to T to balance the bias and variance in (3.5), which may not be accessible in practical implementations. To deal with this issue, a class of fully online regularized algorithms was proposed and investigated by allowing the regularization parameters to vary along the learning process (Ye and Zhou, 2007; Tarres and Yao, 2014). As a comparison, without a regularization parameter to tune, the unregularized online learning algorithm (1.1) achieves a bias-variance balance by tuning only the step sizes. Furthermore, the convergence rates (3.4) require to impose the non-intuitive boundedness assumptions on the gradients encountered during the iterations, which may be violated in practical implementations. This boundedness assumption is removed in our analysis.

4. Proofs

In this section, we present the proofs for the results given in Section 2. Our discussions require to use a property established in the following lemma on functions with (α, L)-Hölder continuous gradients. This lemma is motivated by similar results in the literature (see, e.g., Ying and Zhou, 2017) and we present the proof in Section A for completeness.

Lemma 13 *Let H be a Hilbert space associated with the inner product $\langle\cdot,\cdot\rangle$. Let $\mathcal{G} : H \to \mathbb{R}$ be a convex and differentiable functional satisfying*

$$\|\nabla\mathcal{G}(f) - \nabla\mathcal{G}(\tilde{f})\| \le L\|f - \tilde{f}\|^\alpha, \quad \forall f, \tilde{f} \in H,$$

where $L > 0, \alpha \in (0, 1], \nabla$ is the gradient operator and $\|\cdot\|$ is the norm induced by the inner product. Then, the following inequality holds for any $f, \tilde{f} \in H$

$$\frac{\alpha\|\nabla\mathcal{G}(f) - \nabla\mathcal{G}(\tilde{f})\|^{\frac{1+\alpha}{\alpha}}}{(1+\alpha)L^{\frac{1}{\alpha}}} \le \mathcal{G}(f) - \big[\mathcal{G}(\tilde{f}) + \langle f - \tilde{f}, \nabla\mathcal{G}(\tilde{f})\rangle\big] \le \frac{L\|f - \tilde{f}\|^{1+\alpha}}{1+\alpha}. \tag{4.1}$$

With Lemma 13, we can derive the following lemma on gradients of loss functions at iterates of the algorithm (1.1). Its power consists in bounding the gradients for the possibly unbounded iterates $\{f_t\}_{t\in\mathbb{N}}$ by the gradients for f_H and the excess generalization errors, the first of which can be considered as a constant while the second of which are exactly the terms we are interested in. For a random variable z, we use $\mathbb{E}_z[\cdot]$ to denote the conditional expectation with respect to z.

Lemma 14 *Suppose Assumption 1 holds and $\beta \in (0,1]$. Then,*

$$\mathbb{E}_{z_t}\big[|\phi'(y_t, f_t(x_t))|^{1+\beta}\big] \le 2^\beta L^{\frac{1}{\alpha}}(1+\beta)\big[\mathcal{E}(f_t)-\mathcal{E}(f_H)\big]+ \frac{2^\beta(1-\alpha\beta)}{1+\alpha} + 2^\beta \mathbb{E}_{z_t}\big[|\phi'(y_t, f_H(x_t))|^{1+\beta}\big], \quad \forall t\in\mathbb{N}. \tag{4.2}$$

Proof With the elementary inequality $|u+v|^{1+\beta} \le 2^\beta[|u|^{1+\beta}+|v|^{1+\beta}]$ and the Young's inequality

$$uv \le p^{-1}|u|^p + q^{-1}|v|^q, \quad \forall u,v\in\mathbb{R}, p^{-1}+q^{-1}=1, p\ge 0, \tag{4.3}$$

the term $|\phi'(y_t, f_t(x_t))|^{1+\beta}$ can be controlled by

$$\begin{aligned}
&|\phi'(y_t, f_t(x_t))|^{1+\beta} \le \Big[|\phi'(y_t, f_t(x_t))-\phi'(y_t, f_H(x_t))| + |\phi'(y_t, f_H(x_t))|\Big]^{1+\beta}\\
&\le 2^\beta |\phi'(y_t, f_t(x_t))-\phi'(y_t, f_H(x_t))|^{1+\beta} + 2^\beta|\phi'(y_t, f_H(x_t))|^{1+\beta}\\
&\le \frac{2^\beta\alpha(1+\beta)}{1+\alpha}|\phi'(y_t, f_t(x_t))-\phi'(y_t, f_H(x_t))|^{\frac{1+\alpha}{\alpha}} + \frac{2^\beta(1-\alpha\beta)}{1+\alpha} + 2^\beta|\phi'(y_t, f_H(x_t))|^{1+\beta}.
\end{aligned} \tag{4.4}$$

It follows from the first inequality of (4.1) that

$$\frac{\alpha}{1+\alpha}|\phi'(y_t, f_t(x_t))-\phi'(y_t, f_H(x_t))|^{\frac{1+\alpha}{\alpha}} \le L^{\frac{1}{\alpha}}\Big[\phi(y_t, f_t(x_t))-\phi(y_t, f_H(x_t))-\phi'(y_t, f_H(x_t))(f_t(x_t)-f_H(x_t))\Big].$$

Plugging the above inequality into (4.4) and taking expectations with respect to z_t (note f_t is independent of z_t), we get

$$\begin{aligned}
\mathbb{E}_{z_t}\big[|\phi'(y_t, f_t(x_t))|^{1+\beta}\big] &\le 2^\beta L^{\frac{1}{\alpha}}(1+\beta)\Big[\mathcal{E}(f_t)-\mathcal{E}(f_H)-\big\langle f_t-f_H, \mathbb{E}_{z_t}\big[\phi'(y_t, f_H(x_t))K_{x_t}\big]\big\rangle\Big]\\
&\quad + \frac{2^\beta(1-\alpha\beta)}{1+\alpha} + 2^\beta\mathbb{E}_{z_t}\big[|\phi'(y_t, f_H(x_t))|^{1+\beta}\big]\\
&= 2^\beta L^{\frac{1}{\alpha}}(1+\beta)\big[\mathcal{E}(f_t)-\mathcal{E}(f_H)\big] + \frac{2^\beta(1-\alpha\beta)}{1+\alpha} + 2^\beta\mathbb{E}_{z_t}\big[|\phi'(y_t, f_H(x_t))|^{1+\beta}\big].
\end{aligned}$$

Here the last identity holds since

$$\nabla\mathcal{E}(f_H) = \mathbb{E}_z\big[\phi'(y, f_H(x))K_x\big] = 0.$$

The proof is complete. ■

4.1 Proofs for Convergence in Expectation

Before proving Theorem 1 and Theorem 2 on convergence in expectation, we first present some preparatory results. Our first preliminary result is a weak result on convergence in expectation under a weak condition on the step size sequence (4.5). Eq. (4.6) implies the existence of a sub-index sequence $\{i_t\}_{t\in\mathbb{N}}$ satisfying $\lim_{t\to\infty} A_{i_t}=0$, while (4.7) shows the convergence of a weighted average of the expected excess generalization errors. This result is derived based on a one-step progress inequality in terms of distances in RKHSs (see (4.10)).

Proposition 15 *Let $\{f_t\}_{t\in\mathbb{N}}$ be the sequence given by (1.1) and suppose Assumption 1 holds. If*

$$\lim_{t\to\infty}\eta_t = 0 \quad \textit{and} \quad \sum_{t=1}^\infty \eta_t = \infty, \tag{4.5}$$

then

$$\liminf_{t\to\infty}\mathbb{E}[\mathcal{E}(f_t)-\mathcal{E}(f_H)] = 0 \tag{4.6}$$

and

$$\lim_{T\to\infty}\Big[\sum_{t=1}^T\eta_t\Big]^{-1}\sum_{t=1}^T\eta_t\mathbb{E}[\mathcal{E}(f_t)-\mathcal{E}(f_H)] = 0. \tag{4.7}$$

Lemma 16 *Let $\{\eta_t\}_{t\in\mathbb{N}}$ be a sequence of positive numbers. If $\lim_{t\to\infty}\eta_t=0$ and $\sum_{t=1}^\infty\eta_t=\infty$, then $\lim_{t\to\infty}\big[\sum_{k=1}^t\eta_k\big]^{-1}\sum_{k=1}^t\eta_k^2=0$.*

Proof of Proposition 15 According to the iteration strategy (1.1), we derive

$$\begin{aligned}
\|f_{t+1}-f_H\|^2 &= \|f_t-\eta_t\phi'(y_t, f_t(x_t))K_{x_t}-f_H\|^2\\
&\le \|f_t-f_H\|^2 + \eta_t^2|\phi'(y_t, f_t(x_t))|^2\kappa^2 - 2\eta_t\langle f_t-f_H, \phi'(y_t, f_t(x_t))K_{x_t}\rangle \qquad (4.8)\\
&\le \|f_t-f_H\|^2 + \eta_t^2|\phi'(y_t, f_t(x_t))|^2\kappa^2 + 2\eta_t\big[\phi(y_t, f_H(x_t))-\phi(y_t, f_t(x_t))\big]. \qquad (4.9)
\end{aligned}$$

Note that f_t is independent of z_t. Taking expectations with respect to z_t on both sides and using (4.2) with $\beta=1$, we derive

$$\begin{aligned}
&\mathbb{E}_{z_t}\big[\|f_{t+1}-f_H\|^2\big] \le \|f_t-f_H\|^2 + \eta_t^2\kappa^2\mathbb{E}_{z_t}\big[|\phi'(y_t, f_t(x_t))|^2\big] + 2\eta_t[\mathcal{E}(f_H)-\mathcal{E}(f_t)]\\
&\le \|f_t-f_H\|^2 + 4\eta_t^2\kappa^2L^{\frac{1}{\alpha}}[\mathcal{E}(f_t)-\mathcal{E}(f_H)] + \frac{2(1-\alpha)\eta_t^2\kappa^2}{1+\alpha}\\
&\qquad + 2\eta_t^2\kappa^2\mathbb{E}_{z_t}\big[|\phi'(y_t, f_H(x_t))|^2\big] + 2\eta_t[\mathcal{E}(f_H)-\mathcal{E}(f_t)]\\
&= \|f_t-f_H\|^2 + 2\eta_t\big(1-2\eta_t\kappa^2L^{\frac{1}{\alpha}}\big)[\mathcal{E}(f_H)-\mathcal{E}(f_t)] + 2\eta_t^2\kappa^2\Big(\mathbb{E}_{z_t}\big[|\phi'(y_t, f_H(x_t))|^2\big] + \frac{1-\alpha}{1+\alpha}\Big).
\end{aligned}$$

Since $\lim_{t\to\infty}\eta_t=0$, we can find an integer $t_1\in\mathbb{N}$ such that $\eta_t\le\frac{1}{4\kappa^2L^{\frac{1}{\alpha}}}, \forall t\ge t_1$. This together with $\mathcal{E}(f_H)\le\mathcal{E}(f_t)$ implies

$$\eta_t[\mathcal{E}(f_t)-\mathcal{E}(f_H)] \le \|f_t-f_H\|^2 - \mathbb{E}_{z_t}\big[\|f_{t+1}-f_H\|^2\big] + \gamma\eta_t^2, \quad \forall t\ge t_1, \tag{4.10}$$

where we introduce $\gamma = 2\kappa^2\Big(\mathbb{E}_{z_t}\big[|\phi'(y_t, f_H(x_t))|^2\big] + \frac{1-\alpha}{1+\alpha}\Big)$. Taking expectations followed with a summation from $t = t_1$ to $t = T$ gives

$$\sum_{t=t_1}^{T} \eta_t A_t \leq \mathbb{E}[\|f_{t_1} - f_H\|^2] + \gamma \sum_{t=t_1}^{T} \eta_t^2.$$

It then follows that

$$\begin{aligned}\lim_{T\to\infty} \big[\sum_{t=1}^{T} \eta_t\big]^{-1} \sum_{t=1}^{T} \eta_t A_t &= \lim_{T\to\infty} \big[\sum_{t=1}^{T} \eta_t\big]^{-1} \sum_{t=1}^{t_1-1} \eta_t A_t + \lim_{T\to\infty} \big[\sum_{t=1}^{T} \eta_t\big]^{-1} \sum_{t=t_1}^{T} \eta_t A_t \\ &\leq \lim_{T\to\infty} \big[\sum_{t=1}^{T} \eta_t\big]^{-1} \Big[\mathbb{E}[\|f_{t_1} - f_H\|^2] + \gamma \sum_{t=t_1}^{T} \eta_t^2\Big] = 0,\end{aligned}$$

where we have used $\lim_{t\to\infty} \big[\sum_{k=1}^{t} \eta_k\big]^{-1} \sum_{k=1}^{t} \eta_k^2 = 0$ established in Lemma 16. This establishes (4.7).

We now prove (4.6) by contradiction strategy. Suppose to the contrary that $\liminf_{t\to\infty} A_t = \tilde{a} > 0$. Then, there exists $\tilde{t} \in \mathbb{N}$ such that $A_t \geq 2^{-1}\tilde{a}, \forall t \geq \tilde{t}$, from which we derive from (4.7) that

$$0 = \lim_{T\to\infty} \frac{\sum_{t=1}^{T} \eta_t A_t}{\sum_{t=1}^{T} \eta_t} \geq \frac{\tilde{a}}{2} \lim_{T\to\infty} \frac{\sum_{t=\tilde{t}+1}^{T} \eta_t}{\sum_{t=1}^{T} \eta_t} = \frac{\tilde{a}}{2} - \frac{\tilde{a}}{2} \lim_{T\to\infty} \frac{\sum_{t=1}^{\tilde{t}} \eta_t}{\sum_{t=1}^{T} \eta_t} = \frac{\tilde{a}}{2}.$$

This leads to a contradiction. Therefore, $\liminf_{t\to\infty} A_t = 0$ and the proof is complete. ■

As our second preliminary result, Lemma 17 establishes an upper bound on $\mathbb{E}[\|f_t - f_H\|_2^2]$ in terms of the step size sequence, as well as a lower bound on $\mathbb{E}[\|\nabla\mathcal{E}(f_t)\|^2]$ in terms of the step size sequence and the expected excess generalization errors.

Lemma 17 *Let $\{f_t\}_{t\in\mathbb{N}}$ be the sequence given by (1.1). If Assumption 1 holds and $\lim_{t\to\infty} \eta_t = 0$, then there exist constants $\widehat{C}, \gamma > 0$ independent of t such that the following inequalities hold for any $t \in \mathbb{N}$*

$$\mathbb{E}[\|f_t - f_H\|^2] \leq \widehat{C} + \gamma \sum_{k=1}^{t} \eta_k^2 \tag{4.11}$$

and

$$\mathbb{E}[\|\nabla\mathcal{E}(f_t)\|^2] \geq \frac{\big(\mathbb{E}[\mathcal{E}(f_t) - \mathcal{E}(f_H)]\big)^2}{\widehat{C} + \gamma \sum_{k=1}^{t} \eta_k^2}. \tag{4.12}$$

Proof Since $\mathcal{E}(f_t) \geq \mathcal{E}(f_H)$ for all $t \in \mathbb{N}$, (4.10) implies

$$\mathbb{E}[\|f_{t+1} - f_H\|^2] \leq \mathbb{E}[\|f_t - f_H\|^2] + \eta_t^2\gamma, \quad \forall t \geq t_1,$$

where γ and t_1 are defined in the proof of Proposition 15. Taking a summation of the above inequality from $t = t_1$ to $t = T$ shows

$$\mathbb{E}[\|f_{T+1} - f_H\|^2] \leq \mathbb{E}[\|f_{t_1} - f_H\|^2] + \gamma \sum_{t=t_1}^{T} \eta_t^2 \leq \widehat{C} + \gamma \sum_{t=1}^{T} \eta_t^2,$$

where we introduce $\widehat{C} = \mathbb{E}[\|f_{t_1} - f_H\|^2]$. This establishes (4.11).

We now turn to (4.12). According to the convexity of $\mathcal{E}$ and Schwartz inequality, we get

$$\begin{aligned}\mathbb{E}[\mathcal{E}(f_t)] - \mathcal{E}(f_H) &\leq \mathbb{E}\big[\langle\nabla\mathcal{E}(f_t), f_t - f_H\rangle\big] \leq \mathbb{E}[\|\nabla\mathcal{E}(f_t)\|\|f_t - f_H\|] \\ &\leq \big(\mathbb{E}\big[\|\nabla\mathcal{E}(f_t)\|^2\big]\big)^{\frac{1}{2}} \big(\mathbb{E}\big[\|f_t - f_H\|^2\big]\big)^{\frac{1}{2}}.\end{aligned}$$

The above inequality together with (4.11) gives

$$\mathbb{E}[\big\|\nabla\mathcal{E}(f_t)\big\|^2] \geq \frac{\big(\mathbb{E}\big[\mathcal{E}(f_t) - \mathcal{E}(f_H)\big]\big)^2}{\mathbb{E}[\|f_t - f_H\|^2]} \geq \frac{\big(\mathbb{E}\big[\mathcal{E}(f_t) - \mathcal{E}(f_H)\big]\big)^2}{\widehat{C} + \gamma \sum_{k=1}^{t} \eta_k^2}.$$

This establishes (4.12) and completes the proof. ■

Remark 18 *Eq. (4.11) was derived in Ying and Zhou (2017) under an additional assumption $\sum_{k=1}^{\infty} \eta_k^{1+\alpha} < \infty$, which is removed in Lemma 17. For step sizes of the form $\eta_t = \eta_1 t^{-\theta}$ with $\frac{1}{2} < \theta \leq 1$, it was shown $\mathbb{E}[\|f_t - f_H\|^2] = O\big(t^{1-\theta}\big(\mathcal{E}(f_H) - \inf_f \mathcal{E}(f)\big)\big)$ (Lin and Zhou, 2018). As compared with the results in Ying and Zhou (2017); Lin and Zhou (2018), our discussion implies boundedness of $\mathbb{E}[\|f_t\|^2]$ under a milder condition $\sum_{k=1}^{\infty} \eta_k^2 < \infty$.*

We are now in a position to prove Theorem 1 for the convergence in expectation. Let $\epsilon > 0$ be an arbitrary small number. Our idea is to use Proposition 15, based on one-step progress in terms of the distances in RKHSs, to show that $\{A_t\}_{t\in\mathbb{N}}$ can be smaller than ϵ infinitely often. Once $A_{\tilde{t}} \leq \epsilon$ for a sufficiently large $\tilde{t}$, we can use the assumption $\lim_{t\to\infty} \eta_t^\alpha \sum_{k=1}^{t} \eta_k^2 = 0$ and the one-step progress inequality (4.15) in terms of generalization errors to show $A_t \leq \epsilon$ for any $t \geq \tilde{t}$.

Proof of Theorem 1 Since $\phi'(y, \cdot)$ is (α, L)-Hölder continuous, we can apply the second inequality of (4.1) to show that

$$\phi(y, f_{t+1}(x)) \leq \phi(y, f_t(x)) + (f_{t+1}(x) - f_t(x))\phi'(y, f_t(x)) + \frac{L}{1+\alpha}|f_{t+1}(x) - f_t(x)|^{1+\alpha}.$$

According to the reproducing property $f(x) = \langle f, K_x\rangle, \forall f \in H$ and the iteration scheme (1.1), we know

$$\begin{aligned}\phi(y, f_{t+1}(x)) &\leq \phi(y, f_t(x)) + \langle f_{t+1} - f_t, \phi'(y, f_t(x))K_x\rangle + \frac{L}{1+\alpha}|\langle f_{t+1} - f_t, K_x\rangle|^{1+\alpha} \\ &\leq \phi(y, f_t(x)) - \eta_t\langle\phi'(y_t, f_t(x_t))K_{x_t}, \phi'(y, f_t(x))K_x\rangle + \frac{L\kappa^{1+\alpha}}{1+\alpha}\|f_{t+1} - f_t\|^{1+\alpha} \\ &\leq \phi(y, f_t(x)) - \eta_t\langle\phi'(y_t, f_t(x_t))K_{x_t}, \phi'(y, f_t(x))K_x\rangle + \frac{L\kappa^{2(1+\alpha)}\eta_t^{1+\alpha}}{1+\alpha}|\phi'(y_t, f_t(x_t))|^{1+\alpha}.\end{aligned} \tag{4.13}$$

Putting (4.2) with $\beta = \alpha$ back into (4.13) followed with a conditional expectation with respect to z_t and z yields

$$\mathbb{E}_{z_t}[\mathcal{E}(f_{t+1})] = \mathbb{E}_{z_t,z}[\phi(y, f_{t+1}(x))] \le \mathbb{E}_z[\phi(y, f_t(x))] - \eta_t\big\langle \mathbb{E}_{z_t}[\phi'(y_t, f_t(x_t))K_{x_t}], \mathbb{E}_z[\phi'(y, f_t(x))K_x]\big\rangle$$
$$+ \frac{L\kappa^{2(1+\alpha)}\eta_t^{1+\alpha}}{1+\alpha}\Big[2^\alpha L^{\frac{1}{\alpha}}(1+\alpha)(\mathcal{E}(f_t) - \mathcal{E}(f_H)) + 2^\alpha(1-\alpha) + 2^\alpha \mathbb{E}_z[|\phi'(y, f_H(x))|^{1+\alpha}]\Big]$$
$$\le \mathcal{E}(f_t) - \eta_t\|\nabla\mathcal{E}(f_t)\|^2 + \frac{L\kappa^{2(1+\alpha)}2^\alpha\eta_t^{1+\alpha}\Big[L^{\frac{1}{\alpha}}(1+\alpha)(\mathcal{E}(f_t)-\mathcal{E}(f_H)) + (1-\alpha) + \mathbb{E}_z[|\phi'(y, f_H(x))|^{1+\alpha}]\Big]}{1+\alpha}.$$

Subtracting $\mathcal{E}(f_H)$ from both sides of the above inequality gives

$$\mathbb{E}_{z_t}[\mathcal{E}(f_{t+1})] - \mathcal{E}(f_H) \le \Big[1 + L^{1+\frac{1}{\alpha}}\kappa^{2(1+\alpha)}2^\alpha\eta_t^{1+\alpha}\Big](\mathcal{E}(f_t) - \mathcal{E}(f_H)) - \eta_t\|\nabla\mathcal{E}(f_t)\|^2$$
$$+ \frac{L\kappa^{2(1+\alpha)}2^\alpha\eta_t^{1+\alpha}}{1+\alpha}\Big[(1-\alpha) + \mathbb{E}_z[|\phi'(y, f_H(x))|^{1+\alpha}]\Big]. \quad (4.14)$$

Taking expectations over both sides, the above inequality can be written as

$$A_{t+1} \le (1 + a\eta_t^{1+\alpha})A_t + b\eta_t^{1+\alpha} - \eta_t\mathbb{E}[\|\nabla\mathcal{E}(f_t)\|^2], \quad (4.15)$$

where we introduce the notations

$$a = L^{1+\frac{1}{\alpha}}\kappa^{2(1+\alpha)}2^\alpha \quad \text{and} \quad b = \frac{L\kappa^{2(1+\alpha)}2^\alpha}{1+\alpha}\Big[(1-\alpha) + \mathbb{E}_z[|\phi'(y, f_H(x))|^{1+\alpha}]\Big]. \quad (4.16)$$

Plugging (4.12) into the above inequality gives

$$A_{t+1} \le (1 + a\eta_t^{1+\alpha})A_t + b\eta_t^{1+\alpha} - \frac{\eta_t A_t^2}{\widehat{C} + \gamma\sum_{k=1}^t \eta_k^2}, \quad (4.17)$$

where $\widehat{C}$ and γ are defined in the proof of Lemma 17. The assumption $\lim_{t\to\infty}\eta_t^\alpha\sum_{k=1}^t\eta_k^2 = 0$ implies $\lim_{t\to\infty}\eta_t = 0$ and therefore the assumptions of Proposition 15 hold. Let $\epsilon \in (0,1)$ be an arbitrary number. According to $\liminf_{t\to\infty} A_t = 0$ established in Proposition 15, we can find a $\tilde{t} \in \mathbb{N}$ ($\tilde{t}$ can be sufficiently large) such that $A_{\tilde{t}} \le \epsilon$ and

$$\eta_t^\alpha\big(\widehat{C} + \gamma\sum_{k=1}^t\eta_k^2\big) \le \frac{\epsilon^2}{4(a+b)}, \quad \eta_t^{1+\alpha} \le \frac{\epsilon}{2(a+b)} \quad \forall t \ge \tilde{t}. \quad (4.18)$$

We now prove by induction that $A_t \le \epsilon$ for all $t \ge \tilde{t}$. It suffices to show that $A_{t+1} \le \epsilon$ under the assumption $A_t \le \epsilon$ and $t \ge \tilde{t}$. Since $A_t \le 1$, we derive from (4.17) that

$$A_{t+1} \le A_t + (a+b)\eta_t^{1+\alpha} - \frac{\eta_t A_t^2}{\widehat{C} + \gamma\sum_{k=1}^t\eta_k^2}.$$

We now consider two cases. If $A_t^2 \ge (a+b)\eta_t^\alpha\big(\widehat{C} + \gamma\sum_{k=1}^t\eta_k^2\big)$, then we know $A_{t+1} \le A_t \le \epsilon$. Otherwise, we derive from (4.18) that

$$A_{t+1} \le A_t + (a+b)\eta_t^{1+\alpha} \le \sqrt{(a+b)\eta_t^\alpha\big(\widehat{C} + \gamma\sum_{k=1}^t\eta_k^2\big)} + (a+b)\eta_t^{1+\alpha} \le \epsilon.$$

Putting the above two cases together we derive $A_{t+1} \le \epsilon$. That is, $A_t \le \epsilon$ for all $t \ge \tilde{t}$. Since $\epsilon \in (0,1)$ is arbitrarily chosen, we get $\lim_{t\to\infty} A_t = 0$. ■

The necessary condition in Theorem 2 is established by applying the co-coercivity given in Lemma 13 to bound $\mathcal{E}(f_{t+1})$ in terms of $\mathcal{E}(f_t)$ from below.

Proof of Theorem 2 Since $\phi'(y,\cdot)$ is $(1,L)$-Hölder continuous for any $y \in \mathcal{Y}$, we have

$$\|\nabla\mathcal{E}(f) - \nabla\mathcal{E}(\tilde{f})\| = \big\|\mathbb{E}[\phi'(y, f(x))K_x - \phi'(y, \tilde{f}(x))K_x]\big\| \le \mathbb{E}\big[|\phi'(y, f(x)) - \phi'(y, \tilde{f}(x))|\|K_x\|\big]$$
$$\le L\mathbb{E}\big[|\langle f - \tilde{f}, K_x\rangle|\|K_x\|\big] \le L\kappa^2\|f - \tilde{f}\|. \quad (4.19)$$

That is, $\nabla\mathcal{E}$ is $(1, L\kappa^2)$-Hölder continuous. Lemma 13 with $\alpha = 1$ and $\nabla\mathcal{E}(f_H) = 0$ then yield the following inequality

$$\mathcal{E}(f_t) \ge \mathcal{E}(f_H) + \langle f_t - f_H, \nabla\mathcal{E}(f_H)\rangle + \frac{\|\nabla\mathcal{E}(f_t) - \nabla\mathcal{E}(f_H)\|^2}{2L\kappa^2} = \mathcal{E}(f_H) + \frac{\|\nabla\mathcal{E}(f_t)\|^2}{2L\kappa^2}. \quad (4.20)$$

It follows from the convexity of $\mathcal{E}$ and (1.1) that

$$\mathcal{E}(f_{t+1}) \ge \mathcal{E}(f_t) + \langle\nabla\mathcal{E}(f_t), f_{t+1} - f_t\rangle = \mathcal{E}(f_t) - \eta_t\big\langle\nabla\mathcal{E}(f_t), \phi'(y_t, f_t(x_t))K_{x_t}\big\rangle.$$

Taking expectations over both sides and using (4.20), we derive the following inequality for all $t \in \mathbb{N}$

$$\mathbb{E}[\mathcal{E}(f_{t+1})] \ge \mathbb{E}[\mathcal{E}(f_t)] - \eta_t\mathbb{E}[\|\nabla\mathcal{E}(f_t)\|^2] \ge \mathbb{E}[\mathcal{E}(f_t)] - 2L\kappa^2\eta_t\mathbb{E}[\mathcal{E}(f_t) - \mathcal{E}(f_H)].$$

Hence,

$$A_{t+1} \ge \big(1 - 2L\kappa^2\eta_t\big)A_t, \quad \forall t \in \mathbb{N}.$$

The assumption $\eta_t \le 1/(6L\kappa^2)$ and the elementary inequality $1 - \eta \ge \exp(-2\eta), \forall\eta \in (0, 1/3)$ (Lin and Zhou, 2015) then show

$$A_{t+1} \ge \exp\big(-4L\kappa^2\eta_t\big)A_t \ge \prod_{k=1}^t\exp\big(-4L\kappa^2\eta_k\big)A_1 = \exp\Big(-4L\kappa^2\sum_{k=1}^t\eta_k\Big)A_1,$$

which, together with the condition $\lim_{t\to\infty}A_t = 0$ and $A_1 \ne 0$, then establishes the necessary condition $\sum_{t=1}^\infty\eta_t = \infty$. ■

4.2 Proofs for Almost Sure Convergence

We use the following Doob's martingale convergence theorem (see, e.g., Doob, 1994, page 195) to prove Theorem 4 on almost sure convergence. Specifically, we will use the one-step progress inequality in terms of generalization errors to construct a supermartingale, whose almost sure convergence would imply the almost sure convergence of $\{\hat{A}_t\}_{t\in\mathbb{N}}$.

Lemma 19 *Let $\{\tilde{X}_t\}_{t\in\mathbb{N}}$ be a sequence of non-negative random variables with $\mathbb{E}[\tilde{X}_1] < \infty$ and let $\{\mathcal{F}_t\}_{t\in\mathbb{N}}$ be a nested sequence of sets of random variables with $\mathcal{F}_t \subset \mathcal{F}_{t+1}$ for all $t \in \mathbb{N}$. If $\mathbb{E}[\tilde{X}_{t+1}|\mathcal{F}_t] \le \tilde{X}_t$ for every $t \in \mathbb{N}$, then $\tilde{X}_t$ converges to a nonnegative random variable $\tilde{X}$ almost surely. Furthermore, $\tilde{X} < \infty$ almost surely.*

Proof of Theorem 4 Eq. (4.14) gives

$$\mathbb{E}_{z_t}[\hat{A}_{t+1}] \leq (1+a\eta_t^{1+\alpha})\hat{A}_t + b\eta_t^{1+\alpha}, \quad \forall t \in \mathbb{N}, \tag{4.21}$$

with a and b are defined in the proof of Theorem 1. Denote $c = \prod_{k=1}^{\infty}(1+a\eta_k^{1+\alpha})$, which, according to the elementary inequality $1+\tau \leq \exp(\tau), \tau \geq 0$ and (2.3), satisfies

$$c \leq \prod_{k=1}^{\infty} \exp(a\eta_k^{1+\alpha}) = \exp\big(a\sum_{k=1}^{\infty}\eta_k^{1+\alpha}\big) < \infty.$$

Multiplying both sides of (4.21) by $\prod_{k=t+1}^{\infty}(1+a\eta_k^{1+\alpha})$, we derive

$$\begin{aligned}\prod_{k=t+1}^{\infty}(1+a\eta_k^{1+\alpha})\mathbb{E}_{z_t}\big[\hat{A}_{t+1}\big] &\leq \prod_{k=t}^{\infty}(1+a\eta_k^{1+\alpha})\hat{A}_t + b\eta_t^{1+\alpha}\prod_{k=t+1}^{\infty}(1+a\eta_k^{1+\alpha}) \\ &\leq \prod_{k=t}^{\infty}(1+a\eta_k^{1+\alpha})\hat{A}_t + bc\eta_t^{1+\alpha}. \end{aligned} \tag{4.22}$$

Introduce the stochastic process

$$\hat{X}_t = \prod_{k=t}^{\infty}(1+a\eta_k^{1+\alpha})\hat{A}_t + bc\sum_{k=t}^{\infty}\eta_k^{1+\alpha}, \quad t \in \mathbb{N} \tag{4.23}$$

According to (2.3), we know $\mathbb{E}[\hat{X}_1] < \infty$. Eq. (4.22) implies $\mathbb{E}_{z_t}[\hat{X}_{t+1}] \leq \hat{X}_t$ for all $t \in \mathbb{N}$, that is, $\{\hat{X}_t\}_{t\in\mathbb{N}}$ is a supermartingale taking non-negative values. Lemma 19 then implies that $\lim_{t\to\infty}\hat{X}_t = \hat{X}$ for a non-negative random variable $\hat{X}$ almost surely. Let $\Omega = \{\omega = \{z_t\}_{t\in\mathbb{N}}\}$ be the set for which $\{\hat{X}_t(\omega)\}_t$ converges to $\hat{X}(\omega)$ as $t \to \infty$ and $\hat{X}(\omega) < \infty$. Then, $\Pr\{\Omega\} = 1$, where $\Pr\{\Omega\}$ denotes the probability with which the event Ω happens. Let $\omega \in \Omega$ and $\epsilon > 0$. Since $\sum_{t=1}^{\infty}\eta_t^{1+\alpha} < \infty$, we can find $\tilde{t} \in \mathbb{N}$ such that

$$\sum_{t=\tilde{t}}^{\infty}\eta_t^{1+\alpha} < \frac{\epsilon}{3bc}, \quad \prod_{k=\tilde{t}}^{\infty}(1+a\eta_k^{1+\alpha}) < 1 + \frac{\epsilon}{3\hat{X}(\omega)+\epsilon} \text{ and } |\hat{X}_t(\omega) - \hat{X}(\omega)| < \frac{\epsilon}{3}, \quad \forall t \geq \tilde{t}.$$

It then follows from (4.23) that

$$\begin{aligned}\hat{A}_t(\omega) \leq \hat{X}_t(\omega) &\leq \Big(1 + \frac{\epsilon}{3\hat{X}(\omega)+\epsilon}\Big)\hat{A}_t(\omega) + \frac{\epsilon}{3} \leq \hat{A}_t(\omega) + \frac{\epsilon\hat{X}_t(\epsilon)}{3\hat{X}(\omega)+\epsilon} + \frac{\epsilon}{3} \\ &\leq \hat{A}_t(\omega) + \frac{\epsilon\big(\hat{X}(\omega)+\frac{\epsilon}{3}\big)}{3\hat{X}(\epsilon)+\epsilon} + \frac{\epsilon}{3} \leq \hat{A}_t(\omega) + \frac{2\epsilon}{3}, \quad \forall t \geq \tilde{t},\end{aligned}$$

from which we derive

$$\hat{X}(\omega) - \epsilon \leq \hat{X}_t(\omega) - \frac{2\epsilon}{3} \leq \hat{A}_t(\omega) \leq \hat{X}(\omega) + \epsilon, \quad \forall t \geq \tilde{t}.$$

That is, $\lim_{t\to\infty}\hat{A}_t(\omega) = \hat{X}(\omega)$ for any $\omega \in \Omega$, i.e., $\lim_{t\to\infty}\hat{A}_t = \hat{X}$ almost surely. Since $\sum_{t=1}^{\infty}\eta_t^{1+\alpha} < \infty$, we know $\sum_{t=1}^{\infty}\eta_t^2 < \infty$ and $\lim_{t\to\infty}\eta_t = 0$. This further implies

$$\lim_{t\to\infty}\eta_t^{\alpha}\sum_{k=1}^{t}\eta_k^2 = 0$$

and therefore the assumptions in Theorem 1 hold. Theorem 1 shows that $\lim_{t\to\infty}\mathbb{E}[\hat{A}_t] = 0$. By Fatou's lemma, we get

$$0 \leq \mathbb{E}[\hat{X}] = \mathbb{E}\big[\lim_{t\to\infty}\hat{A}_t\big] \leq \liminf_{t\to\infty}\mathbb{E}[\hat{A}_t] = 0,$$

which implies that $\mathbb{E}[\hat{X}] = 0$ and therefore $\hat{X} = 0$ almost surely since $\hat{X}$ is non-negative. Combining the above deductions together, we know that $\lim_{t\to\infty}\hat{A}_t = 0$ almost surely. ■

Our proof of Theorem 6 is based on the following lemma which can be found in Lin and Zhou (2015) as an easy consequence of the Borel-Cantelli Lemma.

Lemma 20 *Let $\{\xi_t\}_{t\in\mathbb{N}}$ be a sequence of non-negative random variables and $\{\epsilon_t\}_{t\in\mathbb{N}}$ be a sequence of positive numbers satisfying $\lim_{t\to\infty}\epsilon_t = 0$. If $\sum_{t=1}^{\infty}\Pr\{\xi_t > \epsilon_t\} < \infty$, then ξ_t converges to 0 almost surely.*

Proof of Theorem 6 Introduce $\delta_t = t^{-2}$ for all $t \in \mathbb{N}$. According to Corollary 11, there exists a constant $\widetilde{C}_1$ such that

$$\Pr\Big\{t^{\min\{1-\theta,(\alpha+1)\theta-1\}-\epsilon}\hat{A}_t \geq \widetilde{C}_1 t^{-\epsilon}\log^2\frac{t}{\delta_t}\Big\} \leq \delta_t.$$

Since $\sum_{t=1}^{\infty}\delta_t < \infty$ and $\lim_{t\to\infty}t^{-\epsilon}\log^2\frac{t}{\delta_t} = 0$, we can apply Lemma 20 here to show (2.4). The proof is complete. ■

4.3 Proofs for Convergence Rates with High Probability

Our discussion on high-probability convergence rates roots its foundation on the following concentration inequalities of martingales. Part (a) is the Azuma-Hoeffding inequality for martingales with bounded differences (Hoeffding, 1963), while Part (b) is a Bernstein-type inequality which exploits information on variances to derive improved concentration inequalities for martingales (Zhang, 2005). A remarkable property of this Bernstein-type inequality is that it involves a conditional variance which itself is a random variable.

Lemma 21 *Let $z_1, \ldots, z_n$ be a sequence of random variables such that z_k may depend on the previous random variables $z_1, \ldots, z_{k-1}$ for all $k = 1, \ldots, n$. Consider a sequence of functionals $\xi_k(z_1, \ldots, z_k), k = 1, \ldots, n$.*

(a) Assume that $|\xi_k - \mathbb{E}_{z_k}[\xi_k]| \leq b_k$ for each k. Let $\delta \in (0,1)$. With probability at least $1-\delta$ we have

$$\sum_{k=1}^{n}\xi_k - \sum_{k=1}^{n}\mathbb{E}_{z_k}[\xi_k] \leq \Big(2\sum_{k=1}^{n}b_k^2\log\frac{1}{\delta}\Big)^{\frac{1}{2}}. \tag{4.24}$$

(b) Assume that $\xi_k - \mathbb{E}_{z_k}[\xi_k] \leq b$ for each k. Let $\rho > 0$ and $\delta \in (0,1)$. With probability at least $1-\delta$ we have

$$\sum_{k=1}^{n}\xi_k - \sum_{k=1}^{n}\mathbb{E}_{z_k}[\xi_k] \leq \frac{(e^{\rho}-\rho-1)\sigma_n^2}{\rho b} + \frac{b\log\frac{1}{\delta}}{\rho}, \tag{4.25}$$

where $\sigma_n^2 = \sum_{k=1}^{n}\mathbb{E}_{z_k}(\xi_k - \mathbb{E}_{z_k}\xi_k)^2$ is the conditional variance.

Since $\phi'(y,\cdot)$ is (α, L)-Hölder continuous, convex and non-negative, Proposition 1 in Ying and Zhou (2017) shows that $\phi(y,\cdot)$ satisfies the following self-bounding property

$$|\phi'(y,s)|^{\frac{1+\alpha}{\alpha}} \leq \frac{(1+\alpha)^{1+\frac{1}{\alpha}}}{\alpha} L^{\frac{1}{\alpha}} \phi(y,s), \quad \forall y \in \mathcal{Y}, s \in \mathbb{R}.$$

The Young's inequality (4.3) then implies

$$\begin{aligned} |\phi'(y,s)|^2 &\leq \alpha^{-\frac{2\alpha}{1+\alpha}} (1+\alpha)^2 L^{\frac{2}{1+\alpha}} \phi(y,s)^{\frac{2\alpha}{1+\alpha}} \\ &\leq \alpha^{-\frac{2\alpha}{1+\alpha}} L^{\frac{2}{1+\alpha}} (1+\alpha) \Big(2\alpha \phi(y,s) + 1 - \alpha \Big) = A\phi(y,s) + B, \end{aligned} \tag{4.26}$$

where

$$A = 2\alpha^{\frac{1-\alpha}{1+\alpha}} L^{\frac{2}{1+\alpha}} (1+\alpha) \quad \text{and} \quad B = \alpha^{-\frac{2\alpha}{1+\alpha}} L^{\frac{2}{1+\alpha}} (1-\alpha^2). \tag{4.27}$$

Below we will use Part (b) of Lemma 21 to show almost boundedness of $\{f_t\}_{t\in\mathbb{N}}$ with high probability (Proposition 9). To this aim, we first establish a crude bound on the iterates $\{f_t\}_{t\in\mathbb{N}}$ in terms of the step size sequence.

Lemma 22 *Let $\{f_t\}_{t\in\mathbb{N}}$ be the sequence given by (1.1). Assume $\eta_t \leq \frac{1}{A\kappa^2}$ for all $t \in \mathbb{N}$. Then, the following inequalities hold for all $t \in \mathbb{N}$*

$$\|f_{t+1} - f_H\|^2 \leq C_1 \sum_{k=0}^{t} \eta_k \quad \text{and} \quad \|f_{t+1}\|^2 \leq C_1 \sum_{k=1}^{t} \eta_k, \tag{4.28}$$

where we introduce for brevity $\eta_0 = 1$ and

$$C_1 = \|f_H\|_2^2 + A^{-1}B + 2\max\Big\{ \sup_{y\in\mathcal{Y}} \phi(y,0), \sup_{z\in\mathcal{Z}} \phi(y, f_H(x)) \Big\}. \tag{4.29}$$

Furthermore, if $\eta_t \leq \frac{1}{A\kappa^2}$ and $\eta_{t+1} \leq \eta_t$ for all $t \in \mathbb{N}$, we have

$$\sum_{k=1}^{t} \eta_k^2 \phi(y_k, f_k(x_k)) \leq \eta_1 \|f_H\|^2 + C_2 \sum_{k=1}^{t} \eta_k^2, \tag{4.30}$$

where we introduce

$$C_2 = 2\sup_{z\in\mathcal{Z}} \phi(y, f_H(x)) + \eta_1 \kappa^2 B. \tag{4.31}$$

Proof Plugging (4.26) into (4.9) gives

$$\begin{aligned} \|f_{t+1} - f_H\|^2 &\leq \|f_t - f_H\|^2 + \eta_t^2 \kappa^2 [A\phi(y_t, f_t(x_t)) + B] + 2\eta_t [\phi(y_t, f_H(x_t)) - \phi(y_t, f_t(x_t))] \\ &= \|f_t - f_H\|^2 + 2\eta_t \phi(y_t, f_H(x_t)) + \eta_t^2 \kappa^2 B + \eta_t (A\eta_t \kappa^2 - 2)\phi(y_t, f_t(x_t)) \\ &\leq \|f_t - f_H\|^2 + 2\eta_t \phi(y_t, f_H(x_t)) + \eta_t^2 \kappa^2 B \leq \|f_t - f_H\|^2 + \eta_t \big(2\phi(y_t, f_H(x_t)) + A^{-1} B \big), \end{aligned} \tag{4.32}$$

where the last two inequalities follow from the assumption $\eta_t \leq \frac{1}{A\kappa^2}$. According to the definitions of C_1 in (4.29) and η_0, it then follows that

$$\|f_{t+1} - f_H\|^2 = \|f_H\|^2 + \sum_{k=1}^{t} \big[\|f_{k+1} - f_H\|^2 - \|f_k - f_H\|^2 \big] \leq C_1 \sum_{k=0}^{t} \eta_k.$$

This establishes the first inequality in (4.28). We now prove the second inequality in (4.28). Notice that (4.9) also holds if we replace f_H with 0. This, together with (4.26) and $\eta_t \leq \frac{1}{A\kappa^2}$, gives

$$\begin{aligned} \|f_{t+1}\|^2 &\leq \|f_t\|^2 + \eta_t^2 \kappa^2 [A\phi(y_t, f_t(x_t)) + B] + 2\eta_t [\phi(y_t, 0) - \phi(y_t, f_t(x_t))] \\ &= \|f_t\|^2 + 2\eta_t \phi(y_t, 0) + \eta_t^2 \kappa^2 B + \eta_t (A\eta_t \kappa^2 - 2)\phi(y_t, f_t(x_t)) \\ &\leq \|f_t\|^2 + 2\eta_t \phi(y_t, 0) + \eta_t A^{-1} B. \end{aligned}$$

It is now clear

$$\|f_{t+1}\|^2 = \sum_{k=1}^{t} \big[\|f_{k+1}\|^2 - \|f_k\|^2 \big] \leq C_1 \sum_{k=1}^{t} \eta_k.$$

We now show (4.30). Applying $\eta_t \leq \frac{1}{A\kappa^2}$ in (4.32) gives

$$\eta_t \phi(y_t, f_t(x_t)) \leq \|f_t - f_H\|^2 - \|f_{t+1} - f_H\|^2 + 2\eta_t \phi(y_t, f_H(x_t)) + \eta_t^2 \kappa^2 B. \tag{4.33}$$

Multiplying both sides of the above inequality by η_t and using $\eta_{t+1} \leq \eta_t$, we derive

$$\begin{aligned} \eta_t^2 \phi(y_t, f_t(x_t)) &\leq \eta_t \big[\|f_t - f_H\|^2 - \|f_{t+1} - f_H\|^2 \big] + 2\eta_t^2 \phi(y_t, f_H(x_t)) + \eta_t^3 \kappa^2 B \\ &\leq \eta_t \|f_t - f_H\|^2 - \eta_{t+1} \|f_{t+1} - f_H\|^2 + 2\eta_t^2 \phi(y_t, f_H(x_t)) + \eta_t^3 \kappa^2 B. \end{aligned}$$

Taking a summation of the above inequality gives (4.30). The proof is complete. ■

Based on the above lemma, Proposition 23 gives a high-probability bound on $\|f_{t+1} - f_H\|^2$ in terms of $\sum_{k=1}^{t} \eta_k^2 \|f_k - f_H\|^2$. Proposition 23 is proved based on a one-step progress inequality (4.37) in terms of the RKHS distances, where the involved martingale is controlled by a Bernstein-type inequality with the dominant variance term cancelled out by the negative term $-2\sum_{k=1}^{t} \eta_k A_k$ existing in the one-step progress inequality.

Proposition 23 *Suppose assumptions in Theorem 7 hold. Let $\delta \in (0,1)$ and C_η, C_3, C_4 be constants defined by*

$$C_\eta = \sup_{k\in\mathbb{N}} \eta_k \sum_{j=0}^{k} \eta_j < \infty, \tag{4.34}$$

$$C_3 = \sup_{z_k \in \mathcal{Z}} \big\| \phi'(y_k, f_H(x_k)) K_{x_k} - \mathbb{E}_z[\phi'(y, f_H(x)) K_x] \big\|, \; C_4 = \frac{2(1-\alpha)\kappa^2}{1+\alpha} + 2\kappa^2 \mathbb{E}_z\big[|\phi'(y, f_H(x))|^2 \big]. \tag{4.35}$$

Then, there exists a constant ρ_1 (explicitly given in the proof and independent of t as well as the step size sequence) such that the following inequality holds with probability at least $1-\delta$

$$\begin{aligned} \|f_{t+1} - f_H\|^2 \leq (\eta_1 \kappa^2 A + 1)\|f_H\|^2 + (AC_2 + B)\kappa^2 \sum_{k=1}^{t} \eta_k^2 + \frac{C_4 \sum_{k=1}^{t} \big[\eta_k^2 \|f_H - f_k\|^2 \big]}{2C_1 C_\eta \kappa^2 L^{\frac{1}{\alpha}}} \\ + \frac{\big(2C_3 C_1^{\frac{1}{2}} C_\eta + 4L(C_1^{\frac{1}{2}}\kappa)^{\alpha+1} C_\eta \big) \log \frac{1}{\delta}}{\rho_1}. \end{aligned} \tag{4.36}$$

Proof The assumption $\sum_{t=1}^{\infty}\eta_t^2<\infty$ implies that C_η in (4.34) is well defined since $\eta_k\sum_{j=1}^k\eta_j\le\sum_{j=1}^k\eta_j^2<\infty$. According to (4.8) and (4.26), we derive

$$\begin{aligned}\|f_{k+1}-f_H\|^2\le\|f_k-f_H\|^2+\eta_k^2\kappa^2\big(A\phi(y_k,f_k(x_k))+B\big)+2\eta_k\langle f_H-f_k,\mathbb{E}_{z_k}[\phi'(y_k,f_k(x_k))K_{x_k}]\rangle\\+2\eta_k\big\langle f_H-f_k,\phi'(y_k,f_k(x_k))K_{x_k}-\mathbb{E}_{z_k}[\phi'(y_k,f_k(x_k))K_{x_k}]\big\rangle.\quad(4.37)\end{aligned}$$

Using the convexity of ϕ followed with a summation from $k=1$ to t gives

$$\begin{aligned}\|f_{t+1}-f_H\|^2&\le\|f_H\|^2+\kappa^2\sum_{k=1}^t\eta_k^2\big(A\phi(y_k,f_k(x_k))+B\big)+2\sum_{k=1}^t\eta_k\big[\mathcal{E}(f_H)-\mathcal{E}(f_k)\big]\\&\quad+2\sum_{k=1}^t\eta_k\big\langle f_H-f_k,\phi'(y_k,f_k(x_k))K_{x_k}-\mathbb{E}_{z_k}\big[\phi'(y_k,f_k(x_k))K_{x_k}\big]\big\rangle\\&\le(\eta_1A\kappa^2+1)\|f_H\|^2+(AC_2+B)\kappa^2\sum_{k=1}^t\eta_k^2+2\sum_{k=1}^t\eta_k\big[\mathcal{E}(f_H)-\mathcal{E}(f_k)\big]\\&\quad+2\sum_{k=1}^t\eta_k\big\langle f_H-f_k,\phi'(y_k,f_k(x_k))K_{x_k}-\mathbb{E}_{z_k}\big[\phi'(y_k,f_k(x_k))K_{x_k}\big]\big\rangle,\quad(4.38)\end{aligned}$$

where the last inequality is due to (4.30). We now estimate the last term of the above inequality with Lemma 21. To this aim, we need to control both the magnitudes and variances for the martingale difference sequences.

Introduce a sequence of functionals $\xi_k,k\in\mathbb{N}$ as follows

$$\xi_k=\eta_k\big\langle f_H-f_k,\phi'(y_k,f_k(x_k))K_{x_k}-\mathbb{E}_{z_k}\big[\phi'(y_k,f_k(x_k))K_{x_k}\big]\big\rangle.$$

It is clear

$$\begin{aligned}&\big\|\phi'(y_k,f_k(x_k))K_{x_k}-\mathbb{E}_{z_k}[\phi'(y_k,f_k(x_k))K_{x_k}]\big\|\le\big\|\phi'(y_k,f_k(x_k))K_{x_k}-\phi'(y_k,f_H(x_k))K_{x_k}\big\|\\&\quad+\big\|\phi'(y_k,f_H(x_k))K_{x_k}-\mathbb{E}_z[\phi'(y,f_H(x))K_x]\big\|+\mathbb{E}_z[\|(\phi'(y,f_H(x))-\phi'(y,f_k(x)))K_x\|]\\&\le\sup_{z_k\in\mathcal{Z}}\big\|\phi'(y_k,f_H(x_k))K_{x_k}-\mathbb{E}_z[\phi'(y,f_H(x))K_x]\big\|+2L\kappa\sup_{x\in\mathcal{X}}|f_k(x)-f_H(x)|^\alpha,\end{aligned}$$

where we have used the Jensen's inequality in the first step. But

$$|f_k(x)-f_H(x)|=|\langle f_k-f_H,K_x\rangle|\le\|f_k-f_H\|\kappa.$$

Combining the above two inequalities and using the definition of C_3 in (4.35) give

$$\big\|\phi'(y_k,f_k(x_k))K_{x_k}-\mathbb{E}_{z_k}[\phi'(y_k,f_k(x_k))K_{x_k}]\big\|\le C_3+2L\|f_k-f_H\|^\alpha\kappa^{\alpha+1}.\quad(4.39)$$

It then follows from (4.28) and $\mathbb{E}_{z_k}[\xi_k]=0$ that (note $\eta_0=1$)

$$\begin{aligned}\xi_k-\mathbb{E}_{z_k}[\xi_k]=\xi_k&\le\eta_k\|f_H-f_k\|\big\|\phi'(y_k,f_k(x_k))K_{x_k}-\mathbb{E}_{z_k}\big[\phi'(y_k,f_k(x_k))K_{x_k}\big]\big\|\\&\le\eta_kC_3\|f_H-f_k\|+2L\eta_k\kappa^{\alpha+1}\|f_H-f_k\|^{1+\alpha}\quad(4.40)\\&\le\eta_kC_3C_1^{\frac12}\big(\sum_{j=0}^{k-1}\eta_j\big)^{\frac12}+2L(C_1^{\frac12}\kappa)^{\alpha+1}\eta_k\big(\sum_{j=0}^{k-1}\eta_j\big)^{\frac{1+\alpha}{2}}\\&\le C_3C_1^{\frac12}C_\eta+2L(C_1^{\frac12}\kappa)^{\alpha+1}C_\eta.\end{aligned}$$

Here we have used the definition of C_η given in (4.34). Furthermore, according to Lemma 14 with $\beta=1$ and the definition of C_4 in (4.35), the conditional variances can be controlled by (note $\mathbb{E}[(\xi-\mathbb{E}[\xi])^2]<\mathbb{E}[\xi^2]$ for a real-valued random variable ξ)

$$\begin{aligned}\sum_{k=1}^t\mathbb{E}_{z_k}(\xi_k-\mathbb{E}_{z_k}[\xi_k])^2&\le\sum_{k=1}^t\eta_k^2\mathbb{E}_{z_k}\big[\big\langle f_H-f_k,\phi'(y_k,f_k(x_k))K_{x_k}\big\rangle^2\big]\\&\le\sum_{k=1}^t\eta_k^2\|f_H-f_k\|^2\kappa^2\mathbb{E}_{z_k}[|\phi'(y_k,f_k(x_k))|^2]\\&\le\sum_{k=1}^t\eta_k^2\|f_H-f_k\|^2\Big(4\kappa^2L^{\frac1\alpha}[\mathcal{E}(f_k)-\mathcal{E}(f_H)]+C_4\Big).\end{aligned}$$

According to (4.28) and the definition of C_η in (4.34), we can further get

$$\begin{aligned}\sum_{k=1}^t\mathbb{E}_{z_k}(\xi_k-\mathbb{E}_{z_k}[\xi_k])^2&\le4L^{\frac1\alpha}C_1\kappa^2\sum_{k=1}^t\Big[\eta_k^2\big(\sum_{j=0}^{k-1}\eta_j\big)\big(\mathcal{E}(f_k)-\mathcal{E}(f_H)\big)\Big]+C_4\sum_{k=1}^t\eta_k^2\|f_H-f_k\|^2\\&\le4L^{\frac1\alpha}C_1C_\eta\kappa^2\sum_{k=1}^t\eta_k\big(\mathcal{E}(f_k)-\mathcal{E}(f_H)\big)+C_4\sum_{k=1}^t\eta_k^2\|f_H-f_k\|^2.\end{aligned}$$

Let ρ_1 be the largest positive constant such that (such ρ_1 exists since $\lim_{\rho\to0}\frac{e^\rho-\rho-1}{\rho}=0$)

$$\frac{(e^{\rho_1}-\rho_1-1)L^{\frac1\alpha}C_1^{\frac12}\kappa^2}{C_3+2LC_1^{\frac\alpha2}\kappa^{\alpha+1}}\le\frac{\rho_1}{4}.$$

Since C_1 and C_3 do not depend on the step size sequence, ρ_1 is also a constant independent of the step size sequence. Plugging the above estimates on the magnitudes and variances of ξ_k into Part (b) of Lemma 21, we derive the following inequality with probability at least $1-\delta$

$$\begin{aligned}\sum_{k=1}^t\xi_k&\le\frac{(e^{\rho_1}-\rho_1-1)}{\rho_1\big(C_3C_1^{\frac12}C_\eta+2L(C_1^{\frac12}\kappa)^{\alpha+1}C_\eta\big)}\Big[4L^{\frac1\alpha}C_1C_\eta\kappa^2\sum_{k=1}^t\eta_k\big(\mathcal{E}(f_k)-\mathcal{E}(f_H)\big)\\&\qquad+C_4\sum_{k=1}^t\eta_k^2\|f_H-f_k\|^2\Big]+\frac{\big(C_3C_1^{\frac12}C_\eta+2L(C_1^{\frac12}\kappa)^{\alpha+1}C_\eta\big)\log\frac1\delta}{\rho_1}\\&\le\sum_{k=1}^t\eta_k\big(\mathcal{E}(f_k)-\mathcal{E}(f_H)\big)+\frac{C_4\sum_{k=1}^t\big[\eta_k^2\|f_H-f_k\|^2\big]}{4C_1C_\eta\kappa^2L^{\frac1\alpha}}+\frac{\big(C_3C_1^{\frac12}C_\eta+2L(C_1^{\frac12}\kappa)^{\alpha+1}C_\eta\big)\log\frac1\delta}{\rho_1}.\end{aligned}$$

Plugging this inequality into (4.38) gives the stated inequality with probability at least $1-\delta$. ■

According to Proposition 9 and the assumption $\sum_{k=1}^\infty\eta_k^2<\infty$, one can show essentially that $\max_{1\le t\le T}\|f_t-f_H\|^2\le\frac12\max_{1\le t\le T}\|f_t-f_H\|^2+c\log T$ for a constant $c>0$, from which one can establish the boundedness of the iterates with high probability (up to logarithmic factors).

Proof of Proposition 9 We define the subset $\Omega \subset \mathcal{Z}^T$ by

$$\Omega = \Big\{(z_1,\ldots,z_T) : \|f_{t+1}-f_H\|^2 \le C_5 + \frac{C_4\sum_{k=1}^t \big[\eta_k^2\|f_H-f_k\|^2\big]}{2C_1C_\eta\kappa^2L^{\frac{1}{\alpha}}} + C_6\log\frac{T}{\delta} \text{ for all } t=1,\ldots,T\Big\},$$

where we introduce

$$C_5 = (\eta_1\kappa^2A+1)\|f_H\|^2 + (AC_2+B)\kappa^2\sum_{k=1}^\infty \eta_k^2, \quad C_6 = \frac{2C_3C_1^{\frac{1}{2}}C_\eta + 4L(C_1^{\frac{1}{2}}\kappa)^{\alpha+1}C_\eta}{\rho_1}. \tag{4.41}$$

Applying Proposition 23 together with union bounds on probabilities of events, we have $\Pr\{\Omega\} \ge 1-\delta$. Since $\sum_{t=1}^\infty \eta_t^2 < \infty$, there exists a $t_2 \in \mathbb{N}$ such that

$$C_4\sum_{k=t_2}^\infty \eta_k^2 \le C_1C_\eta\kappa^2L^{\frac{1}{\alpha}}.$$

Under the event Ω, we know

$$\begin{aligned}\|f_{t+1}-f_H\|^2 &\le C_5 + \frac{C_4\sum_{k=1}^{t_2}\big[\eta_k^2\|f_H-f_k\|^2\big]}{2C_1C_\eta\kappa^2L^{\frac{1}{\alpha}}} + \frac{C_4\sum_{k=t_2+1}^{t}\big[\eta_k^2\|f_H-f_k\|^2\big]}{2C_1C_\eta\kappa^2L^{\frac{1}{\alpha}}} + C_6\log\frac{T}{\delta}\\ &\le C_5 + C_7 + \frac{1}{2}\max_{t_2<k\le t}\|f_k-f_H\|^2 + C_6\log\frac{T}{\delta}\\ &\le C_5 + C_7 + \frac{1}{2}\max_{1\le k\le T}\|f_k-f_H\|^2 + C_6\log\frac{T}{\delta}, \qquad \forall t=1,\ldots,T.\end{aligned}$$

where we have used the inequality

$$\frac{C_4\sum_{k=1}^{t_2}\big[\eta_k^2\|f_H-f_k\|^2\big]}{2C_1C_\eta\kappa^2L^{\frac{1}{\alpha}}} \le \frac{C_4C_1\sum_{k=1}^{t_2}\big[\eta_k^2\sum_{j=0}^{k-1}\eta_j\big]}{2C_1C_\eta\kappa^2L^{\frac{1}{\alpha}}} := C_7.$$

Under the event Ω, it is now clear that

$$\max_{1\le t\le T}\|f_t-f_H\|^2 \le C_5 + C_7 + \frac{1}{2}\max_{1\le k\le T}\|f_k-f_H\|^2 + C_6\log\frac{T}{\delta}.$$

Solving the above linear inequality yields the stated inequality with $\bar{C} = \max\{2(C_5+C_6+C_7),1\}$ with probability at least $1-\delta$. ■

We are now in a position to prove Theorem 7 on general high-probability convergence rates for a weighted average of iterates. The underlying idea is to construct a modified martingale difference sequence by imposing a constraint on the iterates, which is then estimated by applying the Azuma-Hoeffding inequality on martingales. Furthermore, according to Proposition 9, this modified martingale difference sequence would be identical to the original martingale difference sequence with high probability. Let $\mathbb{I}_{\mathcal{A}}$ denote the indicator function of an event $\mathcal{A}$.

Proof of Theorem 7 We now introduce the following sequence of functionals $\xi_k', k = 1,\ldots,T$ by

$$\xi_k' = \eta_k\big\langle f_H - f_k, \phi'(y_k,f_k(x_k))K_{x_k} - \mathbb{E}_{z_k}\big[\phi'(y_k,f_k(x_k))K_{x_k}\big]\big\rangle\mathbb{I}_{\{\|f_k-f_H\|^2\le\bar{C}\log\frac{2T}{\delta}\}},$$

where $\bar{C}$ is defined in Proposition 9. Analogous to (4.40), we have

$$\begin{aligned}|\xi_k'| &\le \Big[\eta_kC_3\|f_H-f_k\| + 2L\eta_k\kappa^{\alpha+1}\|f_H-f_k\|^{1+\alpha}\Big]\mathbb{I}_{\{\|f_k-f_H\|^2\le\bar{C}\log\frac{2T}{\delta}\}}\\ &\le \big(C_3+2L\kappa^{\alpha+1}\big)\eta_k\max\Big(\|f_H-f_k\|^2,1\Big)\mathbb{I}_{\{\|f_k-f_H\|^2\le\bar{C}\log\frac{2T}{\delta}\}}\\ &\le \big(C_3+2L\kappa^{\alpha+1}\big)\eta_k\bar{C}\log\frac{2T}{\delta} := b_k.\end{aligned} \tag{4.42}$$

It is clear that $\mathbb{E}_{z_k}[\xi_k'] = 0$ and ξ_k' only depends on $z_1,\ldots,z_k$. According to Part (a) of Lemma 21, there exists a subset $\Omega' = \{(z_1,\ldots,z_T) : z_1,\ldots,z_T \in \mathcal{Z}\} \subset \mathcal{Z}^T$ with probability measure $\Pr\{\Omega'\} \ge 1-\frac{\delta}{2}$ such that for any $(z_1,\ldots,z_T) \in \Omega'$ the following inequality holds

$$\sum_{k=1}^T \xi_k' \le \Big(2\sum_{k=1}^T b_k^2\log\frac{2}{\delta}\Big)^{\frac{1}{2}} \le \big(C_3+2L\kappa^{\alpha+1}\big)\bar{C}\log\frac{2T}{\delta}\Big(2\log\frac{2}{\delta}\sum_{k=1}^T\eta_k^2\Big)^{\frac{1}{2}}.$$

According to Proposition 9, there exists a subset $\Omega = \{(z_1,\ldots,z_T) : z_1,\ldots,z_T \in \mathcal{Z}\} \subset \mathcal{Z}^T$ with probability measure $\Pr\{\Omega\} \ge 1-\frac{\delta}{2}$ such that for any $(z_1,\ldots,z_T) \in \Omega$ the following inequality holds

$$\max_{1\le k\le T}\|f_k-f_H\|^2 \le \bar{C}\log\frac{2T}{\delta}.$$

Let $\{\xi_k\}_k$ be the martingale difference sequence defined in the proof of Proposition 23. For any $(z_1,\ldots,z_T) \in \Omega\cap\Omega'$, we then have

$$\sum_{k=1}^T\xi_k = \sum_{k=1}^T\xi_k' \le \big(C_3+2L\kappa^{\alpha+1}\big)\bar{C}\log\frac{2T}{\delta}\Big(2\log\frac{2}{\delta}\sum_{k=1}^T\eta_k^2\Big)^{\frac{1}{2}}.$$

Under this intersection of these two events, it follows from (4.38) and the definition of C_5 given in (4.41) that

$$\begin{aligned}2\sum_{k=1}^T\eta_k\big[\mathcal{E}(f_k)-\mathcal{E}(f_H)\big] &\le (\eta_1A\kappa^2+1)\|f_H\|^2 + (AC_2+B)\kappa^2\sum_{k=1}^T\eta_k^2 + 2\sum_{k=1}^T\xi_k\\ &\le C_5 + 2\big(C_3+2L\kappa^{\alpha+1}\big)\bar{C}\log\frac{2T}{\delta}\Big(2\log\frac{2}{\delta}\sum_{k=1}^T\eta_k^2\Big)^{\frac{1}{2}}.\end{aligned}$$

But $\Pr\{\Omega\cap\Omega'\} \ge 1-\delta$. Therefore, the first inequality of (2.5) holds with probability at least $1-\delta$ and

$$\widetilde{C} = \frac{C_5}{2} + \big(C_3+2L\kappa^{\alpha+1}\big)\bar{C}\Big(2\sum_{k=1}^\infty\eta_k^2\Big)^{\frac{1}{2}}.$$

The second inequality of (2.5) follows from the convexity of $\mathcal{E}(\cdot)$. The proof is complete. ■

Other than the high-probability bounds for the weighted average of iterates $\bar{f}_T^{\eta}$, we can also derive similar results for the uniform average of iterates $\bar{f}_T$. If we choose the step sizes $\eta_t = \eta_1(t\log^\beta t)^{-\frac{1}{2}}$ with $\beta > 1$, then Proposition 24 implies $\mathcal{E}(\bar{f}_T) - \mathcal{E}(f_H) = O(T^{-\frac{1}{2}}\log^{\frac{3}{2}}\frac{T}{\delta})$ with probability at least $1-\delta$. We present the proof in the appendix due to its similarity to the proof of Theorem 7.

Proposition 24 *Suppose assumptions in Theorem 7 hold. Then, for any $\delta \in (0,1)$, with probability at least $1-\frac{\delta}{2}$ we have*

$$\sum_{t=1}^T[\mathcal{E}(f_t)-\mathcal{E}(f_H)] = O\big(\big(T^{\frac{1}{2}}+\sum_{t=1}^T\eta_t\big)\log^{\frac{3}{2}}\frac{2T}{\delta}\big) \quad \text{and} \quad \mathcal{E}(\bar{f}_T)-\mathcal{E}(f_H) = O\big(\big(T^{-\frac{1}{2}}+T^{-1}\sum_{t=1}^T\eta_t\big)\log^{\frac{3}{2}}\frac{2T}{\delta}\big).$$

Theorem 10 is a specific case of Proposition 25 with $\widetilde{T} = \lfloor\frac{T}{2}\rfloor$. The step-stone in proving this proposition is the inequality (4.48) following from the one-step progress (4.47) in terms of generalization errors. The first term on the right hand side of (4.48) can be tackled by Theorem 7 on a weighted summation of $\hat{A}_t$ deduced from the one-step analysis in terms of RKHS distances. The variance of the martingales $\sum_{t=\tilde{t}}^T\bar{\xi}_t$ can be controlled by $\sum_{t=\tilde{t}}^T\eta_t\|\nabla\mathcal{E}(f_t)\|^2$, which is then cancelled out by the third term $-\sum_{t=\tilde{t}}^T\eta_t\|\nabla\mathcal{E}(f_t)\|^2$. A notable fact is that the martingale difference $\bar{\xi}_t - \mathbb{E}_{z_t}[\bar{\xi}_t]$ is bounded by $O(\eta_{\widetilde{T}})$ for all $t \geq \widetilde{T}$ with high probability, which would be small if $\widetilde{T}$ is large. We can balance the three terms on the right hand side of (4.43) by choosing an appropriate $\widetilde{T}$.

Proposition 25 *Suppose that the assumptions in Theorem 7 hold. Let $\widetilde{T} \in \mathbb{N}$ satisfy $1 \leq \widetilde{T} \leq T$. Then, there exists a constant $\widetilde{C}'$ independent of T and $\widetilde{T}$ (explicitly given in the proof) such that for any $\delta \in (0,1)$ the following inequality holds with probability at least $1-\delta$*

$$\mathcal{E}(f_{T+1}) - \mathcal{E}(f_H) \leq \widetilde{C}'\max\Big\{\big[\sum_{t=\widetilde{T}}^T\eta_t\big]^{-1}, \eta_{\widetilde{T}}, \sum_{t=\widetilde{T}}^T\eta_t^{1+\alpha}\Big\}\log^2\frac{3T}{\delta}. \tag{4.43}$$

Proof Recall that $\hat{A}_t = \mathcal{E}(f_t) - \mathcal{E}(f_H)$. According to the proof of Theorem 7, there exists a subset $\Omega = \{(z_1,\ldots,z_T) : z_1,\ldots,z_T \in \mathcal{Z}\} \subset \mathcal{Z}^T$ with $\Pr\{\Omega\} \geq 1 - \frac{2\delta}{3}$ such that for any $(z_1,\ldots,z_T) \in \Omega$, we have

$$\sum_{t=1}^T\eta_t\hat{A}_t \leq \widetilde{C}\log^{\frac{3}{2}}\frac{3T}{\delta} \quad \text{and} \quad \max_{1\leq t\leq T}\|f_t - f_H\|^2 \leq \bar{C}\log\frac{3T}{\delta}, \tag{4.44}$$

where $\widetilde{C}$ and $\bar{C}$ are constants independent of T and δ. Under the event of Ω, we have $\sum_{t=\widetilde{T}}^T\eta_t\hat{A}_t \leq \widetilde{C}\log^{\frac{3}{2}}\frac{3T}{\delta}$. Therefore, there exists a $\tilde{t} \in \mathbb{N}$ satisfying $\widetilde{T} \leq \tilde{t} \leq T$ and

$$\hat{A}_{\tilde{t}} \leq \big[\sum_{t=\widetilde{T}}^T\eta_t\big]^{-1}\widetilde{C}\log^{\frac{3}{2}}\frac{3T}{\delta}. \tag{4.45}$$

Taking expectations only with respect to z over both sides of (4.13) gives

$$\hat{A}_{t+1} \leq \hat{A}_t - \eta_t\big\langle\phi'(y_t,f_t(x_t))K_{x_t},\nabla\mathcal{E}(f_t)\big\rangle + \frac{L\kappa^{2(1+\alpha)}\eta_t^{1+\alpha}}{1+\alpha}|\phi'(y_t,f_t(x_t))|^{1+\alpha}. \tag{4.46}$$

According to (4.4), the term $|\phi'(y_t,f_t(x_t))|^{1+\alpha}$ can be controlled by

$$\begin{aligned}|\phi'(y_t,f_t(x_t))|^{1+\alpha} &\leq 2^\alpha|\phi'(y_t,f_t(x_t)) - \phi'(y_t,f_H(x_t))|^{1+\alpha} + 2^\alpha|\phi'(y_t,f_H(x_t))|^{1+\alpha}\\ &\leq 2^\alpha L^{1+\alpha}|\langle f_t - f_H, K_{x_t}\rangle|^{\alpha(1+\alpha)} + 2^\alpha|\phi'(y_t,f_H(x_t))|^{1+\alpha}\\ &\leq 2^\alpha L^{1+\alpha}\kappa^{\alpha(1+\alpha)}\|f_t - f_H\|^{\alpha(1+\alpha)} + 2^\alpha|\phi'(y_t,f_H(x_t))|^{1+\alpha}.\end{aligned}$$

Plugging the above bound into (4.46) gives

$$\begin{aligned}\hat{A}_{t+1} \leq \hat{A}_t - \eta_t\|\nabla\mathcal{E}(f_t)\|^2 + \eta_t\big\langle\nabla\mathcal{E}(f_t) - \phi'(y_t,f_t(x_t))K_{x_t},\nabla\mathcal{E}(f_t)\big\rangle\\ + \Big(\tilde{a}\|f_t - f_H\|^{\alpha(1+\alpha)} + \tilde{b}\Big)\eta_t^{1+\alpha},\end{aligned} \tag{4.47}$$

where we introduce

$$\tilde{a} = 2^\alpha L^{2+\alpha}\kappa^{(2+\alpha)(1+\alpha)}(1+\alpha)^{-1} \quad \text{and} \quad \tilde{b} = 2^\alpha L\kappa^{2(1+\alpha)}(1+\alpha)^{-1}\sup_{z\in\mathcal{Z}}|\phi'(y,f_H(x))|^{1+\alpha}.$$

Taking a summation from $t=\tilde{t}$ to T yields

$$\hat{A}_{T+1} \leq \hat{A}_{\tilde{t}} + \sum_{t=\tilde{t}}^T\Big(\tilde{a}\|f_t - f_H\|^{\alpha(1+\alpha)} + \tilde{b}\Big)\eta_t^{1+\alpha} - \sum_{t=\tilde{t}}^T\eta_t\|\nabla\mathcal{E}(f_t)\|^2 + \sum_{t=\tilde{t}}^T\bar{\xi}_t, \tag{4.48}$$

where we introduce the following two sequences of functionals

$$\begin{aligned}\bar{\xi}_t &= \eta_t\big\langle\nabla\mathcal{E}(f_t) - \phi'(y_t,f_t(x_t))K_{x_t},\nabla\mathcal{E}(f_t)\big\rangle,\\ \bar{\xi}_t' &= \eta_t\big\langle\nabla\mathcal{E}(f_t) - \phi'(y_t,f_t(x_t))K_{x_t},\nabla\mathcal{E}(f_t)\big\rangle\mathbb{I}_{\{\|f_t-f_H\|^2\leq\bar{C}\log\frac{3T}{\delta}\}}.\end{aligned}$$

Under the event Ω, it is clear $\bar{\xi}_t = \bar{\xi}_t'$. In the following, we will use Part (b) of Lemma 21 to estimate $\sum_{t=\tilde{t}}^T\bar{\xi}_t'$. It is clear that $\mathbb{E}_{z_t}[\bar{\xi}_t'] = 0$ for all $t \in \mathbb{N}$. Let $\bar{t}$ be any integer in $[\widetilde{T}, T]$. It follows from Lemma 14 with $\beta = 1$ and the definition of C_4 given in (4.35) that (note $\mathbb{E}[(\xi - \mathbb{E}[\xi])^2] < \mathbb{E}[\xi^2]$ for a real-valued random variable ξ)

$$\begin{aligned}\sum_{t=\bar{t}}^T\mathbb{E}_{z_t}\big(\bar{\xi}_t' - \mathbb{E}_{z_t}[\bar{\xi}_t']\big)^2 &\leq \sum_{t=\bar{t}}^T\eta_t^2\mathbb{E}_{z_t}\big[\big\langle\phi'(y_t,f_t(x_t))K_{x_t},\nabla\mathcal{E}(f_t)\big\rangle^2\big]\mathbb{I}_{\{\|f_t-f_H\|^2\leq\bar{C}\log\frac{3T}{\delta}\}}\\ &\leq \sum_{t=\bar{t}}^T\eta_t^2\kappa^2\|\nabla\mathcal{E}(f_t)\|^2\mathbb{E}_{z_t}\big[|\phi'(y_t,f_t(x_t))|^2\big]\mathbb{I}_{\{\|f_t-f_H\|^2\leq\bar{C}\log\frac{3T}{\delta}\}}\\ &\leq \sum_{t=\bar{t}}^T\eta_t^2\|\nabla\mathcal{E}(f_t)\|^2\big(4\kappa^2L^{\frac{1}{\alpha}}\big[\mathcal{E}(f_t) - \mathcal{E}(f_H)\big] + C_4\big)\mathbb{I}_{\{\|f_t-f_H\|^2\leq\bar{C}\log\frac{3T}{\delta}\}}.\end{aligned} \tag{4.49}$$

Analyzing analogously to (4.19), one can show that $\nabla\mathcal{E}$ is $(\alpha, L\kappa^{1+\alpha})$-Hölder continuous. Then, Lemma 13 together with $\nabla\mathcal{E}(f_H)=0$ shows that

$$\hat{A}_t = \mathcal{E}(f_t)-\mathcal{E}(f_H) \le \frac{L\kappa^{1+\alpha}\|f_t-f_H\|^{1+\alpha}}{1+\alpha}. \tag{4.50}$$

Plugging the above inequality into (4.49) shows

$$\sum_{t=\bar{t}}^{T}\mathbb{E}_{z_t}\big(\bar{\xi}'_t-\mathbb{E}_{z_t}[\bar{\xi}'_t]\big)^2 \le C_8\log\frac{3T}{\delta}\sum_{t=\bar{t}}^{T}\eta_t^2\|\nabla\mathcal{E}(f_t)\|^2 \le \eta_{\widetilde{T}}C_8\log\frac{3T}{\delta}\sum_{t=\bar{t}}^{T}\eta_t\|\nabla\mathcal{E}(f_t)\|^2, \tag{4.51}$$

where we have used $\bar{t}\ge\widetilde{T}$ and introduced

$$C_8 = \frac{4\kappa^{3+\alpha}L^{1+\frac{1}{\alpha}}\bar{C}}{1+\alpha}+C_4.$$

According to (4.39), there holds

$$\begin{aligned}\bar{\xi}'_t-\mathbb{E}_{z_t}[\bar{\xi}'_t] &\le \eta_t\big|\langle\phi'(y_t,f_t(x_t))K_{x_t}-\nabla\mathcal{E}(f_t),\nabla\mathcal{E}(f_t)\rangle\big|\mathbb{I}_{\{\|f_t-f_H\|^2\le\bar{C}\log\frac{3T}{\delta}\}}\\ &\le \eta_t\|\nabla\mathcal{E}(f_t)\|\big\|\phi'(y_t,f_t(x_t))K_{x_t}-\mathbb{E}_{z_t}[\phi'(y_t,f_t(x_t))K_{x_t}]\big\|\mathbb{I}_{\{\|f_t-f_H\|^2\le\bar{C}\log\frac{3T}{\delta}\}}\\ &\le \big(C_3+2L\|f_t-f_H\|^{\alpha}\kappa^{\alpha+1}\big)\eta_t\|\nabla\mathcal{E}(f_t)\|\mathbb{I}_{\{\|f_t-f_H\|^2\le\bar{C}\log\frac{3T}{\delta}\}},\quad \forall t\ge\bar{t}.\end{aligned}$$

Due to the $(\alpha, L\kappa^{1+\alpha})$-Hölder continuity of $\nabla\mathcal{E}$

$$\|\nabla\mathcal{E}(f_t)\| = \|\nabla\mathcal{E}(f_t)-\nabla\mathcal{E}(f_H)\| \le L\kappa^{1+\alpha}\|f_t-f_H\|^{\alpha},$$

we further get

$$\begin{aligned}\bar{\xi}'_t-\mathbb{E}_{z_t}[\bar{\xi}'_t] &\le \eta_t\big(C_3+2L\kappa^{\alpha+1}\big)\max\big(\|f_t-f_H\|^{\alpha},1\big)L\kappa^{1+\alpha}\|f_t-f_H\|^{\alpha}\mathbb{I}_{\{\|f_t-f_H\|^2\le\bar{C}\log\frac{3T}{\delta}\}}\\ &\le \eta_{\widetilde{T}}\big(C_3+2L\kappa^{\alpha+1}\big)L\kappa^{1+\alpha}\bar{C}\log\frac{3T}{\delta} := \eta_{\widetilde{T}}C_9\log\frac{3T}{\delta},\quad \forall t\ge\bar{t}.\end{aligned}$$

We can find a $\rho_2>0$ independent of T such that $(e^{\rho_2}-\rho_2-1)C_8\le\rho_2C_9$. Applying Part (b) of Lemma 21 with the above bounds on variances and magnitudes of $\bar{\xi}'_k$ followed with union bounds on probabilities, we can find a subset $\Omega'=\{(z_1,\ldots,z_T): z_1,\ldots,z_T\in\mathcal{Z}\}\subset\mathcal{Z}^T$ with $\Pr\{\Omega'\}\ge 1-\frac{\delta}{3}$ such that for any $(z_1,\ldots,z_T)\in\Omega'$ there holds (note $\mathbb{E}_{z_t}[\bar{\xi}'_t]=0$)

$$\begin{aligned}\sum_{t=\bar{t}}^{T}\bar{\xi}'_t &\le \frac{\eta_{\widetilde{T}}(e^{\rho_2}-\rho_2-1)C_8\log\frac{3T}{\delta}\sum_{t=\bar{t}}^{T}\eta_t\|\nabla\mathcal{E}(f_t)\|^2}{\eta_{\widetilde{T}}\rho_2C_9\log\frac{3T}{\delta}}+\frac{\eta_{\widetilde{T}}C_9\log^2\frac{3T}{\delta}}{\rho_2}\\ &\le \sum_{t=\bar{t}}^{T}\eta_t\|\nabla\mathcal{E}(f_t)\|^2+\frac{\eta_{\widetilde{T}}C_9\log^2\frac{3T}{\delta}}{\rho_2},\quad \forall\bar{t}\in[\widetilde{T},T].\end{aligned} \tag{4.52}$$

Under the event $\Omega\cap\Omega'$, we can plug the above inequality with $\bar{t}=\tilde{t},\bar{\xi}'_t=\bar{\xi}_t$ and $\|f_t-f_H\|^2\le\bar{C}\log\frac{3T}{\delta},\forall t=1,\ldots,T$ into (4.48) to derive

$$\begin{aligned}\hat{A}_{T+1} &\le \hat{A}_{\tilde{t}}+\Big(\tilde{a}\bar{C}\log\frac{3T}{\delta}+\tilde{b}\Big)\sum_{t=\tilde{t}}^{T}\eta_t^{1+\alpha}+\frac{\eta_{\widetilde{T}}C_9\log^2\frac{3T}{\delta}}{\rho_2}\\ &\le \big[\sum_{t=\widetilde{T}}^{T}\eta_t\big]^{-1}\widetilde{C}\log^{\frac{3}{2}}\frac{3T}{\delta}+\Big(\tilde{a}\bar{C}+\tilde{b}\Big)\log\frac{3T}{\delta}\sum_{t=\tilde{t}}^{T}\eta_t^{1+\alpha}+\frac{\eta_{\widetilde{T}}C_9\log^2\frac{3T}{\delta}}{\rho_2},\end{aligned}$$

where the last inequality is due to (4.45). This establishes the stated inequality with probability $1-\delta$ and

$$\widetilde{C}' = \widetilde{C}+\tilde{a}\bar{C}+\tilde{b}+C_9\rho_2^{-1}.$$

It is clear that $\widetilde{C}'$ is independent of T and $\widetilde{T}$. The proof is complete. ■

Proof of Corollary 11 The polynomially decaying step sizes $\eta_t=\eta_1t^{-\theta}(\theta>\frac{1}{2})$ satisfies the monotonicity and $\sum_{t=1}^{\infty}\eta_t^2<\infty$ Furthermore, we have

$$\big[\sum_{t=\lfloor\frac{T}{2}\rfloor}^{T}\eta_t\big]^{-1}\le\frac{2}{T\eta_T}=O(T^{\theta-1})\quad\text{and}\quad\sum_{t=\lfloor\frac{T}{2}\rfloor}^{T}\eta_t^{1+\alpha}\le\frac{(T+1)\eta_{\lfloor\frac{T}{2}\rfloor}^{1+\alpha}}{2}=O(T^{1-(1+\alpha)\theta}).$$

The proof is complete if we plug the above estimates into Theorem 10. ■

Acknowledgments

We thank the anonymous referees for their constructive suggestions. The work described in this paper is supported partially by the National Natural Science Foundation of China (Grants No. 11401524, 11531013, 11571078, 11631015, 11771012). Yunwen Lei is also supported by the Science and Technology Innovation Committee Foundation of Shenzhen (Grant No. ZDSYS201703031748284). Lei Shi is also supported by the Joint Research Fund by National Natural Science Foundation of China and Research Grants Council of Hong Kong (Project No. 11461161006 and RGC Project No. N_CityU120/14), Program of Shanghai Subject Chief Scientist (Project No.18XD1400700), and Zhuo Xue program of Fudan University. The corresponding author is Zheng-Chu Guo.

Appendix A. Some Additional Proofs

Proof of Lemma 16 Let $\epsilon>0$ be an arbitrary number. Since $\lim_{t\to\infty}\eta_t=0$ we can find a $t_3\in\mathbb{N}$ such that $\eta_t\le\frac{\epsilon}{2}$ for all $t\ge t_3$. Since $\sum_{t=1}^{\infty}\eta_t=\infty$, we can also find a $t_4>t_3$ such that $\sum_{k=1}^{t_3}\eta_k^2\le\frac{\epsilon}{2}\sum_{k=1}^{t_4}\eta_k$. Then, for any $t\ge t_4$, it holds

$$\begin{aligned}\big[\sum_{k=1}^{t}\eta_k\big]^{-1}\sum_{k=1}^{t}\eta_k^2 &= \big[\sum_{k=1}^{t}\eta_k\big]^{-1}\sum_{k=1}^{t_3}\eta_k^2+\big[\sum_{k=1}^{t}\eta_k\big]^{-1}\sum_{k=t_3+1}^{t}\eta_k^2\\ &\le \frac{\epsilon}{2}+\frac{\epsilon}{2}\big[\sum_{k=1}^{t}\eta_k\big]^{-1}\sum_{k=t_3+1}^{t}\eta_k\le\epsilon.\end{aligned}$$

Since $\epsilon>0$ is arbitrarily chosen, the proof is complete. ■

Proof of Lemma 13 Fix $f, \tilde{f} \in H$. Define a function $g : \mathbb{R} \to \mathbb{R}$ by $g(t) = \mathcal{G}(\tilde{f}+t(f-\tilde{f}))$. It is clear that $g'(t) = \langle f-\tilde{f}, \nabla\mathcal{G}(\tilde{f}+t(f-\tilde{f}))\rangle$ and

$$\begin{aligned}|g'(t)-g'(\tilde{t})| &= \Big\langle f-\tilde{f}, \nabla\mathcal{G}(\tilde{f}+t(f-\tilde{f})) - \nabla\mathcal{G}(\tilde{f}+\tilde{t}(f-\tilde{f}))\Big\rangle \\ &\le \|f-\tilde{f}\|\big\|\nabla\mathcal{G}\big(\tilde{f}+t(f-\tilde{f})\big) - \nabla\mathcal{G}\big(\tilde{f}+\tilde{t}(f-\tilde{f})\big)\big\| \\ &\le L\|f-\tilde{f}\|^{1+\alpha}|t-\tilde{t}|^{\alpha}.\end{aligned}$$

It then follows that

$$\begin{aligned}g(1)-g(0)-g'(0) &= \int_0^1 [g'(t)-g'(0)]dt \le \int_0^1 |g'(t)-g'(0)|dt \\ &\le L\|f-\tilde{f}\|^{1+\alpha}\int_0^1 t^{\alpha}dt = \frac{L\|f-\tilde{f}\|^{1+\alpha}}{1+\alpha},\end{aligned}$$

which amounts to the second inequality in (4.1)

$$\mathcal{G}(f) \le \mathcal{G}(\tilde{f}) + \langle f-\tilde{f}, \nabla\mathcal{G}(\tilde{f})\rangle + \frac{L\|f-\tilde{f}\|^{1+\alpha}}{1+\alpha}. \tag{A.1}$$

We now turn to the first inequality in (4.1). Fix f and $\tilde{f} \in H$. Define a functional $\mathcal{L} : H \to \mathbb{R}$ by $\mathcal{L}(\bar{f}) = \mathcal{G}(\bar{f}) - \langle \bar{f}, \nabla\mathcal{G}(f)\rangle$. It is clear that $\mathcal{L}$ is a convex function and $\nabla\mathcal{L}(f) = \nabla\mathcal{G}(f) - \nabla\mathcal{G}(f) = 0$. According to the first-order optimality condition, we know $\mathcal{L}$ attains its minimum at f and

$$\begin{aligned}\mathcal{L}(f) &= \min_{\bar{f}\in H}\mathcal{L}(\bar{f}) = \min_{\bar{f}\in H}\big[\mathcal{G}(\bar{f}) - \langle \bar{f}, \nabla\mathcal{G}(f)\rangle\big] \\ &\le \min_{\bar{f}\in H}\Big[\mathcal{G}(\tilde{f}) + \langle \bar{f}-\tilde{f}, \nabla\mathcal{G}(\tilde{f})\rangle + \frac{L\|\tilde{f}-\bar{f}\|^{1+\alpha}}{1+\alpha} - \langle \bar{f}, \nabla\mathcal{G}(f)\rangle\Big] \\ &= \mathcal{L}(\tilde{f}) + \min_{\bar{f}\in H}\Big[\langle \tilde{f}-\bar{f}, \nabla\mathcal{G}(f) - \nabla\mathcal{G}(\tilde{f})\rangle + \frac{L\|\tilde{f}-\bar{f}\|^{1+\alpha}}{1+\alpha}\Big] \\ &= \mathcal{L}(\tilde{f}) + \min_{\bar{f}\in H}\Big[\langle \bar{f}, \nabla\mathcal{G}(f) - \nabla\mathcal{G}(\tilde{f})\rangle + \frac{L\|\bar{f}\|^{1+\alpha}}{1+\alpha}\Big],\end{aligned}$$

where the inequality follows from (A.1). Taking $\bar{f} = L^{-\frac{1}{\alpha}}\|\nabla\mathcal{G}(\tilde{f}) - \nabla\mathcal{G}(f)\|^{\frac{1-\alpha}{\alpha}}\big(\nabla\mathcal{G}(\tilde{f}) - \nabla\mathcal{G}(f)\big)$ in the above inequality, we derive

$$\begin{aligned}\mathcal{L}(f) &\le \mathcal{L}(\tilde{f}) - L^{-\frac{1}{\alpha}}\|\nabla\mathcal{G}(\tilde{f}) - \nabla\mathcal{G}(f)\|^{\frac{1+\alpha}{\alpha}} + \frac{L^{-\frac{1}{\alpha}}\|\nabla\mathcal{G}(\tilde{f}) - \nabla\mathcal{G}(f)\|^{\frac{1+\alpha}{\alpha}}}{1+\alpha} \\ &= \mathcal{L}(\tilde{f}) - \frac{\alpha L^{-\frac{1}{\alpha}}}{1+\alpha}\|\nabla\mathcal{G}(f) - \nabla\mathcal{G}(\tilde{f})\|^{\frac{1+\alpha}{\alpha}}.\end{aligned}$$

This establishes the first inequality in (4.1). The proof is complete. ■

Proof of Proposition 24 Consider the following sequence of functionals $\tilde{\xi}_k, k = 1, \ldots, T$ by

$$\tilde{\xi}_k = \big\langle f_H - f_k, \phi'(y_k, f_k(x_k))K_{x_k} - \mathbb{E}_{z_k}\big[\phi'(y_k, f_k(x_k))K_{x_k}\big]\big\rangle,$$

where $\bar{C}$ is defined in Proposition 9. Eq. (4.42) implies that

$$|\tilde{\xi}_k|\mathbb{I}_{\{\|f_k-f_H\|^2\le\bar{C}\log\frac{2T}{\delta}\}} \le \big(C_3 + 2L\kappa^{\alpha+1}\big)\bar{C}\log\frac{2T}{\delta}.$$

By Part (a) of Lemma 21, there exists a subset $\Omega' = \{(z_1, \ldots, z_T) : z_1, \ldots, z_T \in \mathcal{Z}\} \subset \mathcal{Z}^T$ with probability measure $\Pr\{\Omega'\} \ge 1 - \frac{\delta}{2}$ such that for any $(z_1, \ldots, z_T) \in \Omega'$ the following inequality holds

$$\sum_{k=1}^T \tilde{\xi}_k\mathbb{I}_{\{\|f_k-f_H\|^2\le\bar{C}\log\frac{2T}{\delta}\}} \le \big(C_3 + 2L\kappa^{\alpha+1}\big)\bar{C}\log\frac{2T}{\delta}\Big(2T\log\frac{2}{\delta}\Big)^{\frac{1}{2}}.$$

According to Proposition 9, there exists a subset $\Omega = \{(z_1, \ldots, z_T) : z_1, \ldots, z_T \in \mathcal{Z}\} \subset \mathcal{Z}^T$ with probability measure $\Pr\{\Omega\} \ge 1 - \frac{\delta}{2}$ such that for any $(z_1, \ldots, z_T) \in \Omega$ there holds the inequality $\max_{1\le k\le T}\|f_k - f_H\|^2 \le \bar{C}\log\frac{2T}{\delta}$. Under the event $\Omega \cap \Omega'$, we then have

$$\sum_{k=1}^T \tilde{\xi}_k \le \big(C_3 + 2L\kappa^{\alpha+1}\big)\bar{C}\log\frac{2T}{\delta}\Big(2T\log\frac{2}{\delta}\Big)^{\frac{1}{2}}. \tag{A.2}$$

Furthermore, it follows from (4.37) that

$$2[\mathcal{E}(f_k) - \mathcal{E}(f_H)] \le \eta_k^{-1}\big[\|f_k - f_H\|^2 - \|f_{k+1} - f_H\|^2\big] + \eta_k\kappa^2\big(A\phi(y_k, f_k(x_k)) + B\big) + 2\tilde{\xi}_k.$$

Taking a summation of the above inequality from $k = 1$ to T yields the following inequality under the event $\Omega \cap \Omega'$

$$\begin{aligned}2\sum_{k=1}^T[\mathcal{E}(f_k) - \mathcal{E}(f_H)] \le \sum_{k=1}^{T-1}\big(\eta_{k+1}^{-1} - \eta_k^{-1}\big)\|f_{k+1} - f_H\|^2 + \eta_1^{-1}\|f_1 - f_H\|^2 \\ + \kappa^2\sum_{k=1}^T \eta_k\big(A\phi(y_k, f_k(x_k)) + B\big) + 2\sum_{k=1}^T\tilde{\xi}_k.\end{aligned} \tag{A.3}$$

It follows from (4.33) that

$$\sum_{k=1}^T \eta_k\phi(y_k, f_k(x_k)) \le \|f_H\|^2 + 2\sum_{k=1}^T \eta_k\phi(y_k, f_H(x_k)) + \kappa^2 B\sum_{k=1}^T \eta_k^2.$$

Plugging the above bound into (A.3) and using the monotonicity of η_k together with (A.2), we derive the following inequality with probability at least $1 - \delta$

$$\begin{aligned}2\sum_{k=1}^T[\mathcal{E}(f_k) - \mathcal{E}(f_H)] \le (A\kappa^2 + \eta_1^{-1})\|f_H\|^2 + \kappa^2\sum_{k=1}^T\Big(2A\eta_k\sup_z\phi(y, f_H(x)) + B\eta_k + AB\kappa^2\eta_k^2\Big) \\ + \big(C_3 + 2L\kappa^{\alpha+1}\big)\bar{C}(8T)^{\frac{1}{2}}\log^{\frac{3}{2}}\frac{2T}{\delta}.\end{aligned}$$

The proof is complete. ■

References

Alekh Agarwal, Martin J Wainwright, Peter L Bartlett, and Pradeep K Ravikumar. Information-theoretic lower bounds on the oracle complexity of convex optimization. In *Advances in Neural Information Processing Systems*, pages 1–9, 2009.

Léon Bottou. Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nımes*, 91(8), 1991.

Léon Bottou. Online learning and stochastic approximations. *On-line Learning in Neural Networks*, 17(9):142, 1998.

Nicolo Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004.

Di-Rong Chen, Qiang Wu, Yiming Ying, and Ding-Xuan Zhou. Support vector machine soft margin classifiers: error analysis. *Journal of Machine Learning Research*, 5:1143–1175, 2004.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3): 273–297, 1995.

Felipe Cucker and Ding-Xuan Zhou. *Learning Theory: An Approximation Theory Viewpoint*. Cambridge University Press, 2007.

Aymeric Dieuleveut and Francis Bach. Nonparametric stochastic approximation with large step-sizes. *The Annals of Statistics*, 44(4):1363–1399, 2016.

Joseph L Doob. *Measure Theory, Graduate Texts in Mathematics*. Springer, 1994.

John Duchi and Yoram Singer. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, 10(Dec):2899–2934, 2009.

John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Ambuj Tewari. Composite objective mirror descent. In *Conference on Learning Theory*, pages 14–26, 2010.

Zheng-Chu Guo and Lei Shi. Fast and strong convergence of online learning algorithms. *submitted*, 2017.

Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pages 1225–1234, 2016.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Jyrki Kivinen, Alexander J Smola, and Robert C Williamson. Online learning with kernels. *IEEE Transactions on Signal Processing*, 52(8):2165–2176, 2004.

Yunwen Lei and Ding-Xuan Zhou. Convergence of online mirror descent algorithms. *submitted*, 2017.

Junhong Lin and Ding-Xuan Zhou. Learning theory of randomized Kaczmarz algorithm. *Journal of Machine Learning Research*, 16:3341–3365, 2015.

Junhong Lin and Ding-Xuan Zhou. Online learning algorithms can converge comparably fast as batch learning. *IEEE Transactions on Neural Networks and Learning Systems*, in press, 2018. doi: 10.1109/TNNLS.2017.2677970.

Junhong Lin, Raffaello Camoriano, and Lorenzo Rosasco. Generalization properties and implicit regularization for multiple passes sgm. In *International Conference on Machine Learning*, pages 2340–2348, 2016.

Eric Moulines and Francis R Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems*, pages 451–459, 2011.

Klaus-Robert Müller, Sebastian Mika, Gunnar Ratsch, Koji Tsuda, and Bernhard Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, 2001.

Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

Jiquan Ngiam, Adam Coates, Ahbik Lahiri, Bobby Prochnow, Quoc V Le, and Andrew Y Ng. On optimization methods for deep learning. In *International Conference on Machine Learning*, pages 265–272, 2011.

Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In *International Conference on Machine Learning*, pages 449–456, 2012.

Bernhard Schölkopf and Alexander J Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT press, 2001.

Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical Programming*, 127(1):3–30, 2011.

Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization convergence results and optimal averaging schemes. In *International Conference on Machine Learning*, pages 71–79, 2013.

Steve Smale and Yuan Yao. Online learning algorithms. *Foundations of Computational Mathematics*, 6(2):145–170, 2006.

Steve Smale and Ding-Xuan Zhou. Online learning with markov sampling. *Analysis and Applications*, 7(01):87–113, 2009.

Ingo Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2(Nov):67–93, 2001.

Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer Science & Business Media, 2008.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International Conference on Machine Learning*, pages 1139–1147, 2013.

Pierre Tarres and Yuan Yao. Online learning as stochastic approximation of regularization paths: optimality and almost-sure convergence. *IEEE Transactions on Information Theory*, 60(9):5716–5735, 2014.

Yuan Yao. On complexity issues of online learning algorithms. *IEEE Transactions on Information Theory*, 56(12):6470–6481, 2010.

Gui-Bo Ye and Ding-Xuan Zhou. Fully online classification by regularization. *Applied and Computational Harmonic Analysis*, 23(2):198–214, 2007.

Yiming Ying and Massimiliano Pontil. Online gradient descent learning algorithms. *Foundations of Computational Mathematics*, 8(5):561–596, 2008.

Yiming Ying and Ding-Xuan Zhou. Online regularized classification algorithms. *IEEE Transactions on Information Theory*, 52(11):4775–4788, 2006.

Yiming Ying and Ding-Xuan Zhou. Unregularized online learning algorithms with general loss functions. *Applied and Computational Harmonic Analysis*, 2(42):224–244, 2017.

Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *International Conference on Machine Learning*, pages 919–926, 2004.

Tong Zhang. Data dependent concentration bounds for sequential prediction algorithms. In *Conference on Learning Theory*, pages 173–187, 2005.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning*, pages 928–936, 2003.

Journal of Machine Learning Research 18 (2018) 1-38 Submitted 11/16; Revised 2/18; Published 4/18

Convergence Analysis of Distributed Inference with Vector-Valued Gaussian Belief Propagation

Jian Du JIAND@ANDREW.CMU.EDU
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Shaodan Ma SHAODANMA@UMAC.MO
Department of Electrical and Computer Engineering
University of Macau
Avenida da Universidade, Taipa, Macau

Yik-Chung Wu YCWU@EEE.HKU.HK
Department of Electrical and Electronic Engineering
The University of Hong Kong
Pokfulam Road, Hong Kong

Soummya Kar SOUMMYAK@ANDREW.CMU.EDU
José M. F. Moura MOURA@ANDREW.CMU.EDU
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Editor: Qiang Liu

Abstract

This paper considers inference over distributed linear Gaussian models using factor graphs and Gaussian belief propagation (BP). The distributed inference algorithm involves only local computation of the information matrix and of the mean vector, and message passing between neighbors. Under broad conditions, it is shown that the message information matrix converges to a unique positive definite limit matrix for arbitrary positive semidefinite initialization, and it approaches an arbitrarily small neighborhood of this limit matrix at an exponential rate. A necessary and sufficient convergence condition for the belief mean vector to converge to the optimal centralized estimator is provided under the assumption that the message information matrix is initialized as a positive semidefinite matrix. Further, it is shown that Gaussian BP always converges when the underlying factor graph is given by the union of a forest and a single loop. The proposed convergence condition in the setup of distributed linear Gaussian models is shown to be strictly weaker than other existing convergence conditions and requirements, including the Gaussian Markov random field based walk-summability condition, and applicable to a large class of scenarios.

Keywords: Graphical Model, Large-Scale Networks, Linear Gaussian Model, Markov Random Field, Walk-summability.

1. Introduction

Inference based on a set of measurements from multiple agents on a distributed network is a central issue in many problems. While centralized algorithms can be used in small-scale networks, they face difficulties in large-scale networks, imposing a heavy communication burden when all the data is to be transported to and processed at a central processing unit. Dealing with highly distributed data has been recognized by the U.S. National Research Council as one of the big challenges for processing big data (National Research Council, 2013). Therefore, distributed inference techniques that only involve local communication and computation are important for problems arising in distributed networks.

In large-scale linear parameter learning with Gaussian measurements, Gaussian Belief Propagation (BP) (Weiss and Freeman, 2001a) provides an efficient distributed algorithm for computing the marginal means of the unknown parameters, and it has been adopted in a variety of topics including image interpolation (Xiong et al., 2010), distributed power system state inference (Hu et al., 2011), distributed beamforming (Ng et al., 2008), distributed synchronization (Du and Wu, 2013b), fast solver for system of linear equations (Shental et al., 2008a), distributed rate control in ad-hoc networks (Zhang et al., 2010), factor analyzer network (Frey, 1999), sparse Bayesian learning (Tan and Li, 2010), inter-cell interference mitigation (Lehmann, 2012), and peer-to-peer rating in social networks (Bickson and Malkhi, 2008).

Although with great empirical success (Murphy et al., 1999), it is known that a major challenge that hinders BP is the lack of theoretical guarantees of convergence in loopy networks (Chertkov and Chernyak, 2006; Gómez et al., 2007). Convergence of other forms of loopy BP are analyzed by Ihler et al. (2005), Mooij and Kappen (2005, 2007), Noorshams and Wainwright (2013), and Ravanbakhsh and Greiner (2015), but their analyses are not directly applicable to Gaussian BP. Sufficient convergence conditions for Gaussian BP have been developed in Weiss and Freeman (2001a); Malioutov et al. (2006); Moallemi and Roy (2009a); Su and Wu (2015) when the underlying Gaussian distribution is expressed in terms of pairwise connections between *scalar* variables, i.e., it is a Markov random field (MRF). However, depending on how the underlying joint Gaussian distribution is factorized, Gaussian BP may exhibit different convergence properties as different factorizations (different Gaussian models) lead to fundamentally different recursive update structures. In this paper, we study the convergence of Gaussian BP derived from the distributed linear Gaussian model. The motivation is twofold. From the factorization viewpoint, by specifically employing a factorization based on the linear Gaussian model, we are able to bypass difficulties in existing convergence analyses ((Malioutov et al., 2006) and references therein) based on Gaussian Markov random field factorization. From the distributed inference viewpoint, the linear Gaussian model and associated message passing requirements for implementing the Gaussian BP readily conform to the physical network topology arising in large-scale networks such as in (Hu et al., 2011; Ng et al., 2008; Du and Wu, 2013b; Shental et al., 2008a; Zhang et al., 2010; Frey, 1999; Tan and Li, 2010; Lehmann, 2012; Bickson and Malkhi, 2008), thus it is practically important.

Recently, Giscard et al. (2012, 2013, 2016) present a path-sum method to compute the information matrix inverse of a joint Gaussian distribution. Then, the marginal mean is obtained using the information matrix inverse. The path-sum method converges for an

arbitrary valid Gaussian model, however, it is not clear how to adapt it to the distributed and parallel inference setup. In contrast, Gaussian BP is a parallel and fully distributed method that computes the marginal means by computing only the block diagonal elements of the information matrix inverse. Though the block diagonal elements computed by Gaussian BP may not be correct, it is shown that the belief mean still converges to the correct value once Gaussian BP converges. This explains the popularity of Gaussian BP in distributed inference applications, even though its convergence properties are not fully understood.

To fill this gap, this paper studies the convergence of Gaussian BP for linear Gaussian models. Specifically, for the first time, by establishing certain contractive properties of the distributed information matrix (inverse covariance matrix) updates with respect to the Birkhoff metric, we show that, with arbitrary positive semidefinite (p.s.d.) initial message information matrix, the belief covariance for each local variable converges to a unique positive definite limit, and it approaches an arbitrarily small neighborhood of this limit matrix at an exponential rate. Consequently, the recursive equation for the message mean, which depends on the information matrix, can be reduced to a linear recursive equation. Further, we derive a necessary and sufficient convergence condition for this linear recursive equation under the assumption that the initial message information matrix is p.s.d. Furthermore, we show that, when the structure of the factor graph is the union of a single loop and a forest, Gaussian BP always converges. Finally, it is demonstrated that the proposed convergence condition for the linear Gaussian model encompasses the walk-summable convergence condition for Gaussian MRFs (Malioutov et al., 2006).

Note that there exist other distributed estimation frameworks, e.g., consensus+innovations (Kar and Moura, 2013; Kar et al., 2013) and diffusion algorithms (Cattivelli and Sayed, 2010) that enable distributed estimation of parameters and processes in multi-agent networked environments. The consensus+innovation algorithms converge in mean square sense to the centralized optimal solution under the assumption of global observability of the (aggregate) sensing model and connectivity (on the average) of the inter-agent communication network. In particular, these algorithms allow the communication or message exchange network to be different from the physical coupling network of the field being estimated where either networks can be arbitrarily connected with cycles. The results in Kar and Moura (2013); Kar et al. (2013) imply that the unknown field or parameter can be reconstructed completely at each agent in the network. For large-scale networks with high dimensional unknown variable, it may be impractical though to estimate all the unknowns at every agent. Reference (Kar, 2010, section 3.4) develops approaches to address this problem, where under appropriate conditions, each agent can estimate only a subset of the unknown parameter variables. This paper studies a different distributed inference problem where each agent learns only its own unknown random variables; this leads to lower dimensional data exchanges between neighbors.

The rest of this paper is organized as follows. Section 2 presents the system model for distributed inference. Section 3 derives the vector-valued distributed inference algorithm based on Gaussian BP. Section 4 establishes convergence conditions, and Section 5 discloses the relationship between the derived results and existing convergence conditions of Gaussian BP. Finally, Section 6 presents our conclusions.

Notation: Boldface uppercase and lowercase letters represent matrices and vectors, respectively. For a matrix $\mathbf{A}$, $\mathbf{A}^{-1}$ and $\mathbf{A}^T$ denote its inverse (if it exists) and transpose,

respectively. The symbol $\mathbf{I}_N$ denotes the $N \times N$ identity matrix, and $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{R})$ stands for the probability density function (PDF) of a Gaussian random vector $\mathbf{x}$ with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{R}$. The notation $\|\mathbf{x}-\mathbf{y}\|^2_{\mathbf{W}}$ stands for $(\mathbf{x}-\mathbf{y})^T \mathbf{W} (\mathbf{x}-\mathbf{y})$. The symbol $\propto$ represents the linear scalar relationship between two real valued functions. For Hermitian matrices $\mathbf{X}$ and $\mathbf{Y}$, $\mathbf{X} \succeq \mathbf{Y}$ ($\mathbf{X} \succ \mathbf{Y}$) means that $\mathbf{X}-\mathbf{Y}$ is positive semidefinite (definite). The sets $[\mathbf{A}, \mathbf{B}]$ are defined by $[\mathbf{A}, \mathbf{B}] = \{\mathbf{X} : \mathbf{B} \succeq \mathbf{X} \succeq \mathbf{A}\}$. The symbol $\texttt{Bdiag}\{\cdot\}$ stands for block diagonal matrix with elements listed inside the bracket; $\otimes$ denotes the Kronecker product; and $\mathbf{X}_{i,j}$ denotes the component of matrix $\mathbf{X}$ on the i-th row and j-th column.

2. Problem Statement and Markov Random Field

Consider a general connected network[1] of M agents, with $\mathcal{V} = \{1, \ldots, M\}$ denoting the set of agents, and $\mathcal{E}_{\text{Net}} \subset \mathcal{V} \times \mathcal{V}$ the set of all undirected communication links in the network, i.e., if i and j can communicate or exchange information directly, $(i,j) \in \mathcal{E}_{\text{Net}}$. At every agent $n \in \mathcal{V}$, the local observations are given by a linear Gaussian model:

$$\mathbf{y}_n = \sum_{i \in n \cup \mathcal{I}(n)} \mathbf{A}_{n,i} \mathbf{x}_i + \mathbf{z}_n, \tag{1}$$

where $\mathcal{I}(n)$ denotes the set of neighbors of agent n (i.e., all agents i with $(n,i) \in \mathcal{E}_{\text{Net}}$), $\mathbf{A}_{n,i}$ is a known coefficient matrix with full column rank, $\mathbf{x}_i$ is the local unknown parameter at agent i with dimension $N_i \times 1$ and with prior distribution $\mathbf{x}_i \sim \mathcal{N}(\mathbf{x}_i|\mathbf{0}, \mathbf{W}_i)$ ($\mathbf{W}_i \succ \mathbf{0}$), and $\mathbf{z}_n$ is the additive noise with distribution $\mathbf{z}_n \sim \mathcal{N}(\mathbf{z}_n|\mathbf{0}, \mathbf{R}_n)$, where $\mathbf{R}_n \succ 0$. It is assumed that $p(\mathbf{x}_i, \mathbf{x}_j) = p(\mathbf{x}_i)\, p(\mathbf{x}_j)$ and $p(\mathbf{z}_i, \mathbf{z}_j) = p(\mathbf{z}_i)\, p(\mathbf{z}_j)$ for $i \neq j$, and the x_i's and z_j's are independent for all i and j. The goal is to learn $\mathbf{x}_i$, based on $\mathbf{y}_n$, $p(\mathbf{x}_i)$, and $p(\mathbf{z}_n)$.[2]

In centralized estimation, all the observations $\mathbf{y}_n$'s at different agents are forwarded to a central processing unit. Define vectors $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ as the stacking of $\mathbf{x}_n$, $\mathbf{y}_n$, and $\mathbf{z}_n$ in ascending order with respect to n, respectively; then, we obtain

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{z}, \tag{2}$$

where $\mathbf{A}$ is constructed from $\mathbf{A}_{n,i}$, with specific arrangement dependent on the network topology. Assuming $\mathbf{A}$ is of full column rank, and since (2) is a standard linear model, the optimal minimum mean squared error estimate $\widehat{\mathbf{x}} \triangleq \left[\widehat{\mathbf{x}}_1^T, \ldots, \widehat{\mathbf{x}}_M^T\right]^T$ of $\mathbf{x}$ is given by (Murphy, 2012)

$$\widehat{\mathbf{x}} = \int \mathbf{x} \frac{p(\mathbf{x})\, p(\mathbf{y}|\mathbf{x})}{\int p(\mathbf{x})\, p(\mathbf{y}|\mathbf{x})\, \mathrm{d}\mathbf{x}} \mathrm{d}\mathbf{x} = \left(\mathbf{W}^{-1} + \mathbf{A}^T \mathbf{R}^{-1} \mathbf{A}\right)^{-1} \mathbf{A}^T \mathbf{R}^{-1} \mathbf{y}, \tag{3}$$

where $\mathbf{W}$ and $\mathbf{R}$ are block diagonal matrices containing $\mathbf{W}_i$ and $\mathbf{R}_i$ as their diagonal blocks, respectively. Although well-established, centralized estimation in large-scale networks has

1. A connected network is one where any two distinct agents can communicate with each other through a finite number of hops.
2. By slightly modifying (1), the local model would allow two neighboring agents to share a common observation and the analyses in the following sections still apply. Please refer to Du et al. (2017b) for details, and Du et al. (2017a) for the corresponding models and associated (distributed) convergence conditions.

several drawbacks including: 1) the transmission of $\mathbf{y}_n$, $\mathbf{A}_{n,i}$ and $\mathbf{R}_n$ from peripheral agents to the computation center imposes large communication overhead; 2) knowledge of global network topology is needed in order to construct $\mathbf{A}$; 3) the computation burden at the computation center scales up due to the matrix inversion required in (3) with complexity order $\mathcal{O}\left(\left(\sum_{i=1}^{|\mathcal{V}|} N_i\right)^3\right)$, i.e., cubic in the dimension in general.

On the other hand, Gaussian BP running over graphical models representing the joint posterior distribution of all $\mathbf{x}_i$'s provides a distributed way to learn $\mathbf{x}_i$ locally, thereby mitigating the disadvantages of the centralized approach. In particular, with Gaussian MRF, the joint distribution $p(\mathbf{x})\,p(\mathbf{y}|\mathbf{x})$ is expressed in a pairwise form (Malioutov et al., 2006):

$$p(\mathbf{x})\,p(\mathbf{y}|\mathbf{x}) = \prod_{n\in\mathcal{V}} \psi_n\left(\mathbf{x}_n, \{\mathbf{y}_i\}_{i\in\{n\cup\mathcal{I}(n)\}}\right) \prod_{(n,i)\in\mathcal{E}_{\text{MRF}}} \psi_{n,i}(\mathbf{x}_n, \mathbf{x}_i), \tag{4}$$

where

$$\mathcal{E}_{\text{MRF}} \triangleq \mathcal{E}_{\text{Net}} \cup \{(n,i)\,|\exists k, k\neq n, k\neq i, \text{such that } (n,k)\in\mathcal{E}_{\text{Net}}, \text{and } (i,k)\in\mathcal{E}_{\text{Net}}\}; \tag{5}$$

$$\psi_n\left(\mathbf{x}_n, \{\mathbf{y}_i\}_{i\in n\cup\mathcal{I}(n)}\right) = \exp\left\{\frac{1}{2}\left(\mathbf{x}_n^T\mathbf{W}_n^{-1}\mathbf{x}_n + \sum_{i\in n\cup\mathcal{I}(n)} \mathbf{y}_i^T\mathbf{R}_i^{-1}\mathbf{x}_n\right)\right\} \tag{6}$$

is the potential function at agent n, and

$$\begin{aligned}\psi_{n,i}(\mathbf{x}_n,\mathbf{x}_i) = \exp -\Big\{\frac{1}{2}\big[&(\mathbf{A}_{n,n}\mathbf{x}_n)^T\mathbf{R}_n^{-1}(\mathbf{A}_{n,i}\mathbf{x}_i) + (\mathbf{A}_{i,n}\mathbf{x}_n)^T\mathbf{R}_i^{-1}(\mathbf{A}_{i,i}\mathbf{x}_i)\\ &+ \sum_{\substack{k\in\{\widetilde{k}\,|(\widetilde{k},i)\in\mathcal{E}_{\text{Net}},\\ (\widetilde{k},n)\in\mathcal{E}_{\text{Net}}\}}} (\mathbf{A}_{k,n}\mathbf{x}_n)^T\mathbf{R}_k^{-1}(\mathbf{A}_{k,i}\mathbf{x}_i)\big]\Big\}\end{aligned} \tag{7}$$

is the edge potential between $\mathbf{x}_n$ and $\mathbf{x}_i$. After setting up the graphical model representing the joint distribution in (4), messages are exchanged between pairs of agents n and i with $(n,i)\in\mathcal{E}_{\text{MRF}}$. More specifically, according to the standard derivation of Gaussian BP, at the ℓ-th iteration, the message passed from agent n to agent i is

$$w_{n\to i}^{(\ell)}(\mathbf{x}_i) = \int \psi_n\left(\mathbf{x}_n, \{\mathbf{y}_k\}_{k\in n\cup\mathcal{I}(n)}\right)\psi_{n,i}(\mathbf{x}_n,\mathbf{x}_i) \prod_{k\in\mathcal{I}(n)\setminus i} w_{k\to n}^{(\ell-1)}(\mathbf{x}_n)\,\mathrm{d}\mathbf{x}_n. \tag{8}$$

As shown by (8), Gaussian BP is iterative with each agent alternatively receiving messages from its neighbors and forwarding out updated messages. At each iteration, agent i computes its belief on variable $\mathbf{x}_i$ as

$$b_{\text{MRF}}^{(\ell)}(\mathbf{x}_i) \propto \psi_i\left(\mathbf{x}_i, \{\mathbf{y}_n\}_{n\in i\cup\mathcal{I}(i)}\right) \prod_{k\in\mathcal{I}(n)} w_{k\to i}^{(\ell)}(\mathbf{x}_i). \tag{9}$$

It is known that, as the messages (8) converge, the mean of the belief (9) is the exact mean of the marginal distribution of $\mathbf{x}_i$ (Weiss and Freeman, 2001a).

It might seem that our distributed inference problem is now solved, as a solution is readily available. However, there are two serious limitations for the Gaussian MRF approach.

First, messages are passed between pairs of agents in $\mathcal{E}_{\text{MRF}}$, which according to the definition (5) includes not only those direct neighbors, but also pairs that are two hops away but share a common neighbor. This is illustrated in Fig. 1, where Fig. 1(a) shows a network of 4 agents with a line between two neighboring agents indicating the availability of a physical communication link, and Fig. 1(b) shows the equivalent pairwise graph. For this example, in the physical network, there is no direct connection between agents 1 and 4, nor between agents 1 and 3. But in the pairwise representation, those connections are present. We summarize the above observations in the following remark.

Remark 1 *For a network with communication edge set $\mathcal{E}_{Net}$ and local observations following (1), the corresponding MRF graph edge set satisfies $\mathcal{E}_{MRF}\supseteq\mathcal{E}_{Net}$. Thus, Gaussian BP for Gaussian MRFs cannot be applied to the distributed inference problem with the local observation model (1).*[3]

The consequence of the above findings is that, not only does information need to be shared among agents two hops away from each other to construct the edge potential function in (7), but also the messages (8) may be required to be exchanged among non-direct neighbors, where a physical communication link is not available. This complicates significantly the message exchange scheduling.

Secondly, even if the message scheduling between non-neighboring agents can be realized, the convergence of (8) is not guaranteed in loopy networks. For Gaussian MRF with scalar variables, sufficient convergence conditions have been proposed in (Weiss and Freeman, 2001a; Malioutov et al., 2006; Su and Wu, 2015). However, depending on how the factorization of the underlying joint Gaussian distribution is performed, Gaussian BP may exhibit different convergence properties as different factorizations (different Gaussian models) lead to fundamentally different recursive update structures. Furthermore, these results apply only to scalar Gaussian BP, and extension to vector-valued Gaussian BP is nontrivial as we show in this paper.

The next section derives distributed vector inference based on Gaussian BP with high order interactions (beyond pairwise connections), where information sharing and message exchange requirement conform to the physical network topology. Furthermore, convergence conditions will be studied in Section 4, and we show in Section 5 that the convergence condition obtained is strictly weaker than, i.e., subsumes the convergence conditions in (Weiss and Freeman, 2001a; Malioutov et al., 2006; Su and Wu, 2015).

3. In Section 5, we further show that the convergence condition of Gaussian BP obtained in this paper for model (1) encompasses all existing convergence conditions of Gaussian BP for the corresponding Gaussian MRF.

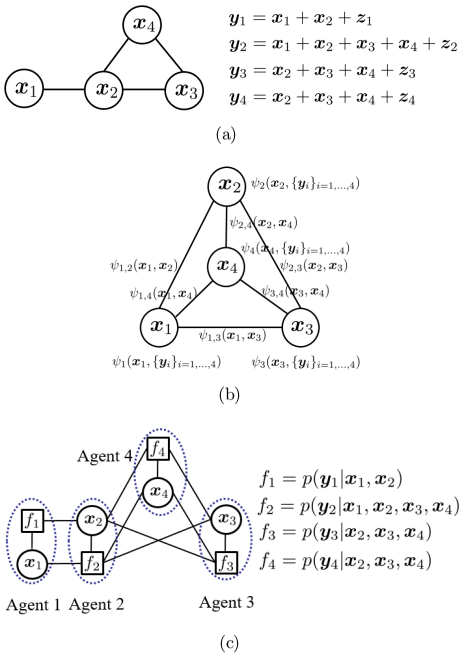


Figure 1: (a) A physical network with 4 agents, where $\{1,2\}$ and $\{2,3,4\}$ are two groups of agents that are within the communication range of each other, respectively. $\mathbf{x}_i$ is the local unknown vector, and $\mathbf{y}_i$ is the local observation at agent i that follows (1); (b) The corresponding MRF of Fig. 1 (a) with $\psi_n\left(\mathbf{x}_n, \{\mathbf{y}_i\}_{i\in n\cup\mathcal{I}(n)}\right)$ and $\psi_{n,i}\left(\mathbf{x}_n, \mathbf{x}_i\right)$ defined in (6) and (7), respectively. (c) The corresponding factor graph of Fig. 1 (a) with f_i defined in (10). Since $p(\mathbf{x}_i)$ does not involve message passing, the $p(\mathbf{x}_i)$ associated to each variable node is not drawn to keep the figure simple.

3. Distributed Inference with Vector-Valued Gaussian BP and Non-Pairwise Interaction

The joint distribution $p\left(\mathbf{x}\right)p\left(\mathbf{y}|\mathbf{x}\right)$ is first written as the product of the prior distribution and the likelihood function of each local linear Gaussian model in (1) as

$$p\left(\mathbf{x}\right)p\left(\mathbf{y}|\mathbf{x}\right) = \prod_{n\in\mathcal{V}} p\left(\mathbf{x}_n\right) \prod_{n\in\mathcal{V}} \underbrace{p\left(\mathbf{y}_n|\left\{\mathbf{x}_i\right\}_{i\in n\cup\mathcal{I}(n)}\right)}_{\triangleq f_n}. \tag{10}$$

To facilitate the derivation of the distributed inference algorithm, the factorization in (10) is expressed in terms of a factor graph (Kschischang et al., 2001), where every vector variable $\mathbf{x}_i$ is represented by a circle (called variable node) and the probability distribution of a vector variable or a group of vector variables is represented by a square (called factor node). A variable node is connected to a factor node if the variable is involved in that particular factor. For example, Fig. 1(c) shows the factor graph representation for the network in Fig. 1(a).

We derive the Gaussian BP algorithm over the corresponding factor graph to learn $\mathbf{x}_n$ for all $n \in \mathcal{V}$ (Kschischang et al., 2001). It involves two types of messages: one is the message from a variable node $\mathbf{x}_j$ to its neighboring factor node f_n, defined as

$$m^{(\ell)}_{j\to f_n}\left(\mathbf{x}_j\right) = p\left(\mathbf{x}_j\right) \prod_{f_k\in\mathcal{B}(j)\setminus f_n} m^{(\ell-1)}_{f_k\to j}\left(\mathbf{x}_j\right), \tag{11}$$

where $\mathcal{B}\left(j\right)$ denotes the set of neighbouring factor nodes of $\mathbf{x}_j$, and $m^{(\ell-1)}_{f_k\to j}\left(\mathbf{x}_j\right)$ is the message from f_k to $\mathbf{x}_j$ at time $\ell-1$. The second type of message is from a factor node f_n to a neighboring variable node $\mathbf{x}_i$, defined as

$$m^{(\ell)}_{f_n\to i}\left(\mathbf{x}_i\right) = \int\cdots\int f_n \times \prod_{j\in\mathcal{B}(f_n)\setminus i} m^{(\ell)}_{j\to f_n}\left(\mathbf{x}_j\right) \mathrm{d}\left\{\mathbf{x}_j\right\}_{j\in\mathcal{B}(f_n)\setminus i}, \tag{12}$$

where $\mathcal{B}\left(f_n\right)$ denotes the set of neighboring variable nodes of f_n. The process iterates between equations (11) and (12). At each iteration ℓ, the approximate marginal distribution, also referred to as belief, on $\mathbf{x}_i$ is computed locally at $\mathbf{x}_i$ as

$$b^{(\ell)}_{\mathrm{BP}}\left(\mathbf{x}_i\right) = p\left(\mathbf{x}_i\right) \prod_{f_n\in\mathcal{B}(i)} m^{(\ell)}_{f_n\to i}\left(\mathbf{x}_i\right). \tag{13}$$

In the sequel, we derive the exact expressions for the messages $m^{(\ell)}_{j\to f_n}\left(\mathbf{x}_j\right)$, $m^{(\ell)}_{f_n\to i}\left(\mathbf{x}_i\right)$, and belief $b^{(\ell)}_{\mathrm{BP}}\left(\mathbf{x}_i\right)$. First, let the initial messages at each variable node and factor node be in Gaussian function forms as

$$m^{(0)}_{f_n\to i}\left(\mathbf{x}_i\right) \propto \exp\left\{-\frac{1}{2}||\mathbf{x}_i - \mathbf{v}^{(0)}_{f_n\to i}||^2_{\mathbf{J}^{(0)}_{f_n\to i}}\right\}. \tag{14}$$

In Appendix A, it is shown that the general expression for the message from variable node j to factor node f_n is

$$m^{(\ell)}_{j\to f_n}\left(\mathbf{x}_j\right) \propto \exp\left\{-\frac{1}{2}||\mathbf{x}_j - \mathbf{v}^{(\ell)}_{j\to f_n}||^2_{\mathbf{J}^{(\ell)}_{j\to f_n}}\right\}, \tag{15}$$

with

$$\mathbf{J}_{j\to f_n}^{(\ell)} = \mathbf{W}_j^{-1} + \sum_{f_k\in\mathcal{B}(j)\setminus f_n} \mathbf{J}_{f_k\to j}^{(\ell-1)}, \tag{16}$$

$$\mathbf{v}_{j\to f_n}^{(\ell)} = \left[\mathbf{J}_{j\to f_n}^{(\ell)}\right]^{-1}\left[\sum_{f_k\in\mathcal{B}(j)\setminus f_n} \mathbf{J}_{f_k\to j}^{(\ell-1)}\mathbf{v}_{f_k\to j}^{(\ell-1)}\right], \tag{17}$$

where $\mathbf{J}_{f_k\to j}^{(\ell-1)}$ and $\mathbf{v}_{f_k\to j}^{(\ell-1)}$ are the message information matrix (inverse of covariance matrix) and mean vector received at variable node j at the $(\ell-1)$-th iteration, respectively. Furthermore, the message from factor node f_n to variable node i is given by

$$m_{f_n\to i}^{(\ell)}(\mathbf{x}_i) \propto \alpha_{f_n\to i}^{(\ell)} \exp\left\{-\frac{1}{2}||\mathbf{x}_i - \mathbf{v}_{f_n\to i}^{(\ell)}||^2_{\mathbf{J}_{f_n\to i}^{(\ell)}}\right\}, \tag{18}$$

with

$$\mathbf{J}_{f_n\to i}^{(\ell)} = \mathbf{A}_{n,i}^T\left[\mathbf{R}_n + \sum_{j\in\mathcal{B}(f_n)\setminus i}\mathbf{A}_{n,j}\left[\mathbf{J}_{j\to f_n}^{(\ell)}\right]^{-1}\mathbf{A}_{n,j}^T\right]^{-1}\mathbf{A}_{n,i}, \tag{19}$$

$$\mathbf{v}_{f_n\to i}^{(\ell)} = \left[\mathbf{J}_{f_n\to i}^{(\ell)}\right]^{-1}\mathbf{A}_{n,i}^T\left[\mathbf{R}_n + \sum_{j\in\mathcal{B}(f_n)\setminus i}\mathbf{A}_{n,j}\left[\mathbf{J}_{j\to f_n}^{(\ell)}\right]^{-1}\mathbf{A}_{n,j}^T\right]^{-1}\left(\mathbf{y}_n - \sum_{j\in\mathcal{B}(f_n)\setminus i}\mathbf{A}_{n,j}\mathbf{v}_{j\to f_n}^{(\ell)}\right), \tag{20}$$

and

$$\alpha_{f_n\to i}^{(\ell)} \propto \int\cdots\int \exp\left\{-\frac{1}{2}\mathbf{z}^T\mathbf{\Lambda}_{f_n\to i}^{(\ell)}\mathbf{z}\right\}\mathrm{d}\mathbf{z}. \tag{21}$$

In (21), $\mathbf{\Lambda}_{f_n\to i}^{(\ell)}$ is a diagonal matrix containing the eigenvalues of $\mathbf{A}_{n,\{\mathcal{B}(f_n)\setminus i\}}^T\mathbf{R}_n^{-1}\mathbf{A}_{n,\{\mathcal{B}(f_n)\setminus i\}} + \mathbf{J}_{\{\mathcal{B}(f_n)\setminus i\}\to f_n}^{(\ell)}$, with $\mathbf{A}_{n,\{\mathcal{B}(f_n)\setminus i\}}$ denoting a row block matrix containing $\mathbf{A}_{n,j}$ as row elements for all $j\in\mathcal{B}(f_n)\setminus i$ arranged in ascending order, and $\mathbf{J}_{\{\mathcal{B}(f_n)\setminus i\}\to f_n}^{(\ell)}$ denoting a block diagonal matrix with $\mathbf{J}_{j\to f_n}^{(\ell)}$ as its block diagonal elements for all $j\in\mathcal{B}(f_n)\setminus i$ arranged in ascending order.

Obviously, the validity of (18) depends on the existence of $\alpha_{f_n\to i}^{(\ell)}$. It is evident that (21) is the integral of a Gaussian distribution and equals to a constant when $\mathbf{\Lambda}_{f_n\to i}^{(\ell)} \succ \mathbf{0}$ or equivalently $\mathbf{A}_{n,\{\mathcal{B}(f_n)\setminus i\}}^T\mathbf{R}_n^{-1}\mathbf{A}_{n,\{\mathcal{B}(f_n)\setminus i\}} + \mathbf{J}_{\{\mathcal{B}(f_n)\setminus i\}\to f_n}^{(\ell)} \succ \mathbf{0}$. Otherwise, $\alpha_{f_n\to i}^{(\ell)}$ does not exist. Therefore, the necessary and sufficient condition for the existence of $m_{f_n\to i}^{(\ell)}(\mathbf{x}_i)$ is

$$\mathbf{A}_{n,\{\mathcal{B}(f_n)\setminus i\}}^T\mathbf{R}_n^{-1}\mathbf{A}_{n,\{\mathcal{B}(f_n)\setminus i\}} + \mathbf{J}_{\{\mathcal{B}(f_n)\setminus i\}\to f_n}^{(\ell)} \succ \mathbf{0}. \tag{22}$$

In general, the necessary and sufficient condition is difficult to be verified, as $\mathbf{J}_{\{\mathcal{B}(f_n)\setminus i\}\to f_n}^{(\ell)}$ changes in each iteration. However, as $\mathbf{R}_n^{-1} \succ \mathbf{0}$, it can be decomposed as $\mathbf{R}_n^{-1} = \widetilde{\mathbf{R}}_n^T\widetilde{\mathbf{R}}_n$. Then

$$\mathbf{A}_{n,\{\mathcal{B}(f_n)\setminus i\}}^T\mathbf{R}_n^{-1}\mathbf{A}_{n,\{\mathcal{B}(f_n)\setminus i\}} = \left(\widetilde{\mathbf{R}}_n\mathbf{A}_{n,\{\mathcal{B}(f_n)\setminus i\}}\right)^T\left(\widetilde{\mathbf{R}}_n\mathbf{A}_{n,\{\mathcal{B}(f_n)\setminus i\}}\right) \succeq \mathbf{0}.$$

Hence, one simple sufficient condition to guarantee (22) is $\mathbf{J}_{\{\mathcal{B}(f_n)\setminus i\}\to f_n}^{(\ell)} \succ \mathbf{0}$ or equivalently its diagonal block matrix $\mathbf{J}_{j\to f_n}^{(\ell)} \succ \mathbf{0}$ for all $j\in\mathcal{B}(f_n)\setminus i$. The following lemma shows that setting the initial message covariances $\mathbf{J}_{f_n\to i}^{(0)} \succeq \mathbf{0}$ for all $(n,i)\in\mathcal{E}_{\mathrm{Net}}$ guarantees $\mathbf{J}_{j\to f_n}^{(\ell)} \succ \mathbf{0}$ for $\ell\geq 1$ and all $(n,j)\in\mathcal{E}_{\mathrm{Net}}$.

Lemma 2 *Let the initial messages at factor node f_k be in Gaussian forms with the initial message information matrix $\boldsymbol{J}_{f_k\to j}^{(0)} \succeq \boldsymbol{0}$ for all $k\in\mathcal{V}$ and $j\in\mathcal{B}(f_k)$. Then $\boldsymbol{J}_{j\to f_n}^{(\ell)} \succ \boldsymbol{0}$ and $\boldsymbol{J}_{f_k\to j}^{(\ell)} \succ \boldsymbol{0}$ for all $\ell\geq 1$ with $j\in\mathcal{V}$ and $f_n, f_k\in\mathcal{B}(j)$. Furthermore, in this case, all messages $m_{j\to f_n}^{(\ell)}(\boldsymbol{x}_j)$ and $m_{f_k\to j}^{(\ell)}(\boldsymbol{x}_i)$ are well defined.*

Proof See Appendix B. ■

For this factor graph based approach, according to the message updating procedure (15) and (18), message exchange is only needed between neighboring agents (an agent refers to a variable-factor pair as shown in Fig. 1 (c)). For example, the messages transmitted from agent n to its neighboring agent i are $m_{f_n\to i}^{(\ell)}(\mathbf{x}_i)$ and $m_{n\to f_i}^{(\ell)}(\mathbf{x}_n)$. Thus, the factor graph does impose a clear messaging schedule, and the message passing scheme given in (11) and (12) conforms with the network topology. Furthermore, if the messages $m_{j\to f_n}^{(\ell)}(\mathbf{x}_j)$ and $m_{f_n\to i}^{(\ell)}(\mathbf{x}_i)$ exist for all ℓ (which can be achieved using Lemma 2), the messages are Gaussian, therefore only the corresponding mean vectors and information matrices (inverse of covariance matrices) are needed to be exchanged.

Finally, if the Gaussian BP messages exist, according to the definition of belief in (13), $b_{\mathrm{BP}}^{(\ell)}(\mathbf{x}_i)$ at iteration ℓ is computed as

$$\begin{aligned}\mathbf{b}_{\mathrm{BP}}^{(\ell)}(\mathbf{x}_i) &= p(\mathbf{x}_i)\prod_{f_n\in\mathcal{B}(i)} m_{f_n\to i}^{(\ell)}(\mathbf{x}_i),\\ &\propto \mathcal{N}\left(\mathbf{x}_i|\boldsymbol{\mu}_i^{(\ell)}, \mathbf{P}_i^{(\ell)}\right),\end{aligned}$$

where the belief covariance matrix

$$\mathbf{P}_i^{(\ell)} = \left[\mathbf{W}_i^{-1} + \sum_{f_n\in\mathcal{B}(i)}\mathbf{J}_{f_n\to i}^{(\ell)}\right]^{-1}, \tag{23}$$

and mean vector

$$\boldsymbol{\mu}_i^{(\ell)} = \mathbf{P}_i^{(\ell)}\left[\sum_{f_n\in\mathcal{B}(i)}\mathbf{J}_{f_n\to i}^{(\ell)}\mathbf{v}_{f_n\to i}^{(\ell)}\right]. \tag{24}$$

The iterative algorithm based on Gaussian BP is summarized as follows. The algorithm is started by setting the messages from factor nodes to variable nodes as in (14). At each round of message exchange, every variable node computes the output messages to its neighboring factor nodes according to (16) and (17). After receiving the messages from its neighboring variable nodes, each factor node computes its output messages according to (19) and (20). The iterative computation terminates when the iterates in (15) or (18) tend to approach a fixed value or the maximum number of iterations is reached.

Remark 3 *We assume that $\boldsymbol{R}_n \succ \boldsymbol{0}$ in this paper. If, however, some of the observations are noiseless, for example, $\boldsymbol{R}_n = \boldsymbol{0}$, the local observation is $\boldsymbol{y}_n = \sum_{i\in n\cup\mathcal{I}(n)} \boldsymbol{A}_{n,i}\boldsymbol{x}_i$. Then the corresponding local likelihood function is represented by the Dirac measure $\delta(\boldsymbol{y}_n - \sum_{i\in n\cup\mathcal{I}(n)} \boldsymbol{A}_{n,i}\boldsymbol{x}_i)$. Suppose, for example, there is only one agent with $\boldsymbol{R}_n = \boldsymbol{0}$, and all others are $\boldsymbol{R}_i \succ \boldsymbol{0}$. The the joint distribution is written as*

$$p(\boldsymbol{x})\, p(\boldsymbol{y}|\boldsymbol{x}) = \delta(\boldsymbol{y}_n - \sum_{i\in n\cup\mathcal{I}(n)} \boldsymbol{A}_{n,i}\boldsymbol{x}_i) \prod_{j\in\mathcal{V}} p(\boldsymbol{x}_j) \prod_{k\in\mathcal{V}} p\left(\boldsymbol{y}_k | \{\boldsymbol{x}_i\}_{i\in k\cup\mathcal{I}(k)}\right).$$

In this case, if $\boldsymbol{A}_{n,n}$ is invertible, then, by the definition of the Dirac measure, we have $\boldsymbol{x}_n = \boldsymbol{A}_{n,n}^{-1}\left(\boldsymbol{y}_n - \sum_{i\in\mathcal{I}(n)} \boldsymbol{A}_{n,i}\boldsymbol{x}_i\right)$. By substituting this equation into all of the likelihood functions involving $\boldsymbol{x}_n$, we have the equivalent joint distribution as in (10) with all the likelihood functions having a positive definite noise covariance. We thereafter can apply Gaussian BP to this new factorization and the convergence analysis in this paper still applies. Therefore, without loss of generality, we assume all $\boldsymbol{R}_n \succ \boldsymbol{0}$. Note that when $\boldsymbol{R}_n = \boldsymbol{0}$ for all n, this problem is equivalent to solving algebraic equations, which has been studied in (Shental et al., 2008b) using Gaussian BP.

4. Convergence Analysis

The challenge of deploying the Gaussian BP algorithm for large-scale networks is in determining whether it will converge or not. In particular, it is generally known that if the factor graph contains cycles, the Gaussian BP algorithm may diverge. Thus, determining convergence conditions for the Gaussian BP algorithm is very important. Sufficient conditions for the convergence of Gaussian BP with scalar variables in loopy graphs are available in (Weiss and Freeman, 2001a; Malioutov et al., 2006; Su and Wu, 2015). However, these conditions are derived based on pairwise graphs with local functions in the form of (6) and (7). This contrasts with the model considered in this paper, where the f_n in (10) involves high-order interactions between vector variables, and thus the convergence results in (Weiss and Freeman, 2001a; Malioutov et al., 2006; Su and Wu, 2015) cannot be applied to the factor graph based vector-form Gaussian BP.

Due to the recursive updating property of $m^{(\ell)}_{j\to f_n}(\mathbf{x}_j)$ and $m^{(\ell)}_{f_n\to i}(\mathbf{x}_i)$ in (15) and (18), the message evolution can be simplified by combining these two kinds of messages into one. By substituting $\mathbf{J}^{(\ell)}_{j\to f_n}$ in (16) into (19), the updating of the message covariance matrix inverse, referred to as message information matrix in the following, can be denoted as

$$\begin{aligned}\mathbf{J}^{(\ell)}_{f_n\to i} &= \mathbf{A}^T_{n,i}\Big[\mathbf{R}_n + \sum_{j\in\mathcal{B}(f_n)\setminus i} \mathbf{A}_{n,j}\Big[\mathbf{W}_j^{-1} + \sum_{f_k\in\mathcal{B}(j)\setminus f_n} \mathbf{J}^{(\ell-1)}_{f_k\to j}\Big]^{-1}\mathbf{A}^T_{n,j}\Big]^{-1}\mathbf{A}_{n,i}\\ &\triangleq \mathcal{F}_{n\to i}\left(\left\{\mathbf{J}^{(\ell-1)}_{f_k\to j}\right\}_{(f_k,j)\in\widetilde{\mathcal{B}}(f_n,i)}\right),\end{aligned} \tag{25}$$

where $\widetilde{\mathcal{B}}(f_n,i) = \{(f_k,j)\,|j\in\mathcal{B}(f_n)\setminus i, f_k\in\mathcal{B}(j)\setminus f_n\}$. Observing that $\mathbf{J}^{(\ell)}_{f_n\to i}$ in (25) is independent of $\mathbf{v}^{(\ell)}_{j\to f_n}$ and $\mathbf{v}^{(\ell)}_{f_n\to i}$ in (17) and (18), so we can first focus on the convergence property of $\mathbf{J}^{(\ell)}_{f_n\to i}$ alone and then later on that of $\mathbf{v}^{(\ell)}_{f_n\to i}$. With the convergence characterization of $\mathbf{J}^{(\ell)}_{f_n\to i}$ and $\mathbf{v}^{(\ell)}_{f_n\to i}$, we will further investigate the convergence of belief covariances and means in (23) and (24), respectively.

Note that computing $\mathbf{P}^{(\ell)}_j$ requires all the incoming messages from neighboring nodes including $\mathbf{J}^{(\ell)}_{f_n\to j}$ as shown in (23) by replacing the subscript i with j in (23). However, according to (25), when computing $\mathbf{J}^{(\ell)}_{f_n\to i}$ the quantity $\mathbf{J}^{(\ell-1)}_{f_n\to j}$ is excluded, i.e., the quantity inside the inner square brackets equals $[\mathbf{P}^{(\ell-1)}_j]^{-1} - \mathbf{J}^{(\ell-1)}_{f_n\to j}$. Therefore, one cannot compute $\mathbf{J}^{(\ell)}_{f_n\to i}$ from $\mathbf{P}^{(\ell)}_j$ alone.

4.1 Convergence of Message Information Matrices

To efficiently represent the updates of all message information matrices, we introduce the following definitions. Let

$$\mathbf{J}^{(\ell-1)} \triangleq \mathtt{Bdiag}\left(\left\{\mathbf{J}^{(\ell-1)}_{f_n\to i}\right\}_{n\in\mathcal{V}, i\in\mathcal{B}(f_n)}\right)$$

be a block diagonal matrix with diagonal blocks being the message information matrices in the network at time $\ell-1$ with index arranged in ascending order first on n and then on i. Using the definition of $\mathbf{J}^{(\ell-1)}$, the term $\sum_{f_k\in\mathcal{B}(j)\setminus f_n}\mathbf{J}^{(\ell-1)}_{f_k\to j}$ in (25) can be written as $\boldsymbol{\Xi}_{n,j}\mathbf{J}^{(\ell-1)}\boldsymbol{\Xi}^T_{n,j}$, where $\boldsymbol{\Xi}_{n,j}$ is for selecting appropriate components from $\mathbf{J}^{(\ell-1)}$ to form the summation. Further, define $\mathbf{H}_{n,i} = \left[\{\mathbf{A}_{n,j}\}_{j\in B(f_n)\setminus i}\right]$, $\boldsymbol{\Psi}_{n,i} = \mathtt{Bdiag}\left(\left\{\mathbf{W}_j^{-1}\right\}_{j\in B(f_n)\setminus i}\right)$ and $\mathbf{K}_{n,i} = \mathtt{Bdiag}\left(\{\boldsymbol{\Xi}_{n,j}\}_{j\in B(f_n)\setminus i}\right)$, all with component blocks arranged with ascending order on j. Then (25) can be written as

$$\mathbf{J}^{(\ell)}_{f_n\to i} = \mathbf{A}^T_{n,i}\left\{\mathbf{R}_n + \mathbf{H}_{n,i}\left[\boldsymbol{\Psi}_{n,i} + \mathbf{K}_{n,i}\left(\mathbf{I}_{|\mathcal{B}(f_n)|-1}\otimes\mathbf{J}^{(\ell-1)}\right)\mathbf{K}^T_{n,i}\right]^{-1}\mathbf{H}^T_{n,i}\right\}^{-1}\mathbf{A}_{n,i}. \tag{26}$$

Now, we define the function $\mathcal{F} \triangleq \{\mathcal{F}_{1\to k},\ldots,\mathcal{F}_{n\to i},\ldots,\mathcal{F}_{n\to M}\}$ that satisfies $\mathbf{J}^{(\ell)} = \mathcal{F}\left(\mathbf{J}^{(\ell-1)}\right)$. Then, by stacking $\mathbf{J}^{(\ell)}_{f_n\to i}$ on the left side of (26) for all n and i as the block diagonal matrix $\mathbf{J}^{(\ell)}$, we obtain

$$\begin{aligned}\mathbf{J}^{(\ell)} &= \mathbf{A}^T\{\boldsymbol{\Omega} + \mathbf{H}\left[\boldsymbol{\Psi} + \mathbf{K}\left(\mathbf{I}_\varphi\otimes\mathbf{J}^{(\ell-1)}\right)\mathbf{K}^T\right]^{-1}\mathbf{H}^T\}^{-1}\mathbf{A},\\ &\triangleq \mathcal{F}\left(\mathbf{J}^{(\ell-1)}\right),\end{aligned} \tag{27}$$

where $\mathbf{A}$, $\mathbf{H}$, $\boldsymbol{\Psi}$, and $\mathbf{K}$ are block diagonal matrices with block elements $\mathbf{A}_{n,i}$, $\mathbf{H}_{n,i}$, $\boldsymbol{\Psi}_{n,i}$, and $\mathbf{K}_{n,i}$, respectively, arranged in ascending order, first on n and then on i (i.e., the same order as $\mathbf{J}^{(\ell)}_{f_n\to i}$ in $\mathbf{J}^{(\ell)}$). Furthermore, $\varphi = \sum_{n=1}^M |\mathcal{B}(f_n)|\,(|\mathcal{B}(f_n)|-1)$ and $\boldsymbol{\Omega}$ is a block diagonal matrix with diagonal blocks $\mathbf{I}_{|\mathcal{B}(f_n)|}\otimes\mathbf{R}_n$ with ascending order on n. We first present some properties of the updating operator $\mathcal{F}(\cdot)$, the proofs being provided in Appendix C.

Proposition 4 *The updating operator $\mathcal{F}(\cdot)$ satisfies the following properties:*

P 4.1: $\mathcal{F}\left(\mathbf{J}^{(\ell)}\right) \succeq \mathcal{F}\left(\mathbf{J}^{(\ell-1)}\right)$, if $\mathbf{J}^{(\ell)} \succeq \mathbf{J}^{(\ell-1)} \succeq \mathbf{0}$.

P 4.2: $\alpha\mathcal{F}\left(\mathbf{J}^{(\ell)}\right) \succ \mathcal{F}\left(\alpha\mathbf{J}^{(\ell)}\right)$ and $\mathcal{F}\left(\alpha^{-1}\mathbf{J}^{(\ell)}\right) \succ \alpha^{-1}\mathcal{F}\left(\mathbf{J}^{(\ell)}\right)$, if $\mathbf{J}^{(\ell)} \succ \mathbf{0}$ and $\alpha > 1$.

P 4.3: Define $\mathbf{U} \triangleq \mathbf{A}^T\mathbf{\Omega}^{-1}\mathbf{A}$ and $\mathbf{L} \triangleq \mathbf{A}^T\left[\mathbf{\Omega} + \mathbf{H}\mathbf{\Psi}^{-1}\mathbf{H}^T\right]^{-1}\mathbf{A}$. With arbitrary $\mathbf{J}^{(0)} \succeq \mathbf{0}$, $\mathcal{F}\left(\mathbf{J}^{(\ell)}\right)$ is bounded by $\mathbf{U} \succeq \mathcal{F}\left(\mathbf{J}^{(\ell)}\right) \succeq \mathbf{L} \succ \mathbf{0}$ for $\ell \geq 1$.

Based on the above properties of $\mathcal{F}(\cdot)$, we can establish the convergence of the information matrices.

Theorem 5 *There exists a unique positive definite fixed point $\boldsymbol{J}^*$ for the mapping $\mathcal{F}(\cdot)$.*

Proof The set $[\mathbf{L}, \mathbf{U}]$ is a compact set. Further, according to Proposition 4, P 4.3, for arbitrary $\mathbf{J}^{(0)} \succeq \mathbf{0}$, $\mathcal{F}$ maps $[\mathbf{L}, \mathbf{U}]$ into itself starting from $\ell \geq 1$. Next, we show that $[\mathbf{L}, \mathbf{U}]$ is a convex set. Suppose that $\mathbf{X}$, $\mathbf{Y} \in [\mathbf{L}, \mathbf{U}]$, and $0 \leq t \leq 1$, then $t\mathbf{X} - t\mathbf{L}$ and $(1-t)\mathbf{Y} - (1-t)\mathbf{L}$ are positive semidefinite (p.s.d.) matrices. Since the sum of two p.s.d. matrices is a p.s.d. matrix, $t\mathbf{X} + (1-t)\mathbf{Y} \succeq \mathbf{L}$. Likewise, it can be shown that $t\mathbf{X} + (1-t)\mathbf{Y} \preceq \mathbf{U}$. Thus, the continuous function $\mathcal{F}$ maps a compact convex subset of the Banach space of positive definite matrices into itself. Therefore, the mapping $\mathcal{F}$ has a fixed point in $[\mathbf{L}, \mathbf{U}]$ according to Brouwer's Fixed-Point Theorem (Zeidler, 1985), and the fixed point is positive definite (p.d.).

Next, we prove the uniqueness of the fixed point. Suppose that there exist two fixed points $\mathbf{J}^* \succ \mathbf{0}$ and $\widetilde{\mathbf{J}}^* \succ \mathbf{0}$. Since $\mathbf{J}^*$ and $\widetilde{\mathbf{J}}^*$ are p.d., their components $\mathbf{J}^*_{f_n \to i}$ and $\widetilde{\mathbf{J}}^*_{f_n \to i}$ are also p.d. matrices. For the component blocks of $\mathbf{J}^*$ and $\widetilde{\mathbf{J}}^*$, there are two possibilities: 1) $\widetilde{\mathbf{J}}^*_{f_n \to i} - \mathbf{J}^*_{f_n \to i} \succ \mathbf{0}$ or $\widetilde{\mathbf{J}}^*_{f_n \to i} - \mathbf{J}^*_{f_n \to i}$ is indefinite for some $n, i \in \mathcal{V}$, and 2) $\widetilde{\mathbf{J}}^*_{f_n \to i} - \mathbf{J}^*_{f_n \to i} \preceq 0$ for all $n, i \in \mathcal{V}$.

For the first case, there must exist $\xi_{f_n,i} > 1$ such that $\xi_{f_n,i}\mathbf{J}^*_{f_n \to i} - \widetilde{\mathbf{J}}^*_{f_n \to i}$ has one or more zero eigenvalues, while all other eigenvalues are positive. Pick the component matrix with the maximum $\xi_{f_n,i}$ among those falling into this case, say $\xi_{f_\varrho,\tau}$, then, we can write

$$\xi_{f_\varrho,\tau}\mathbf{J}^*_{f_\varrho \to \tau} - \widetilde{\mathbf{J}}^*_{f_\varrho \to \tau} \succeq \mathbf{0}, \tag{28}$$

or in terms of the information matrices for the whole network

$$\xi_{f_\varrho,\tau}\mathbf{J}^* \succeq \widetilde{\mathbf{J}}^* \succ \mathbf{0}, \quad \xi_{f_\varrho,\tau} > 1. \tag{29}$$

Applying $\mathcal{F}$ on both sides of (29), according to the monotonic property of $\mathcal{F}(\cdot)$ as shown in Proposition 4, P 4.1, we have

$$\mathcal{F}\left(\xi_{f_\varrho,\tau}\mathbf{J}^*\right) \succeq \mathcal{F}\left(\widetilde{\mathbf{J}}^*\right) = \widetilde{\mathbf{J}}^*, \tag{30}$$

where the equality is due to $\widetilde{\mathbf{J}}^*$ being a fixed point. According to Proposition 4, P 4.2, $\xi_{f_\varrho,\tau}\mathcal{F}(\mathbf{J}^*) \succ \mathcal{F}\left(\xi_{f_\varrho,\tau}\mathbf{J}^*\right)$. Therefore, from (30), we obtain $\xi_{f_\varrho,\tau}\mathbf{J}^* \succ \widetilde{\mathbf{J}}^*$. Consequently,

$$\xi_{f_\varrho,\tau}\mathbf{J}^*_{f_\varrho,\tau} \succ \widetilde{\mathbf{J}}^*_{f_\varrho,\tau}.$$

But this contradicts with $\xi_{f_\varrho,\tau}\mathbf{J}^*_{f_\varrho,\tau} - \widetilde{\mathbf{J}}^*_{f_\varrho,\tau}$ having one or more zero eigenvalues as discussed before (28). Therefore, we must have $\mathbf{J}^* = \widetilde{\mathbf{J}}^*$.

On the other hand, if we have case two, which is $\widetilde{\mathbf{J}}^*_{f_n \to i} - \mathbf{J}^*_{f_n \to i} \preceq 0$ for all $n, i \in \mathcal{V}$, we can repeat the above derivation with the roles of $\widetilde{\mathbf{J}}^*$ and $\mathbf{J}^*$ reversed, and we would again obtain $\mathbf{J}^* = \widetilde{\mathbf{J}}^*$. Consequently, $\mathbf{J}^*$ is unique. ■

Lemma 2 states that with arbitrary p.s.d. initial message information matrices, the message information matrices will be kept as p.d. at every iteration. On the other hand, Theorem 5 indicates that there exists a unique fixed point for the mapping $\mathcal{F}$. Next, we will show that, with arbitrary initial value $\mathbf{J}^{(0)} \succeq 0$, $\mathbf{J}^{(\ell)}$ converges to a unique p.d. matrix.

Theorem 6 *The matrix sequence $\left\{\boldsymbol{J}^{(\ell)}\right\}_{l=0,1,\ldots}$ defined by (27) converges to a unique positive definite matrix $\boldsymbol{J}^*$ for any initial covariance matrix $\boldsymbol{J}^{(0)} \succeq \mathbf{0}$.*

Proof With arbitrary initial value $\mathbf{J}^{(0)} \succeq \mathbf{0}$, following Proposition 4, P 4.3, we have $\mathbf{U} \succeq \mathbf{J}^{(1)} \succeq \mathbf{L} \succ \mathbf{0}$. On the other hand, according to Theorem 5, (27) has a unique fixed point $\mathbf{J}^* \succ \mathbf{0}$. Notice that we can always choose a scalar $\alpha > 1$ such that

$$\alpha\mathbf{J}^* \succeq \mathbf{J}^{(1)} \succeq \mathbf{L}. \tag{31}$$

Applying $\mathcal{F}(\cdot)$ to (31) ℓ times, and using Proposition 4, P 4.1, we have

$$\mathcal{F}^\ell(\alpha\mathbf{J}^*) \succeq \mathcal{F}^{\ell+1}\left(\mathbf{J}^{(0)}\right) \succeq \mathcal{F}^\ell(\mathbf{L}), \tag{32}$$

where $\mathcal{F}^\ell(\mathbf{X})$ denotes applying $\mathcal{F}$ on $\mathbf{X}$ ℓ times.

We start from the left inequality in (32). According to Proposition 4, P 4.2, $\alpha\mathbf{J}^* \succ \mathcal{F}(\alpha\mathbf{J}^*)$. Applying $\mathcal{F}$ again gives $\mathcal{F}(\alpha\mathbf{J}^*) \succ \mathcal{F}^2(\alpha\mathbf{J}^*)$. Applying $\mathcal{F}(\cdot)$ repeatedly, we can obtain $\mathcal{F}^2(\alpha\mathbf{J}^*) \succ \mathcal{F}^3(\alpha\mathbf{J}^*) \succ \mathcal{F}^4(\alpha\mathbf{J}^*)$, etc. Thus $\mathcal{F}^\ell(\alpha\mathbf{J}^*)$ is a non-increasing sequence with respect to the partial order induced by the cone of p.s.d. matrices as ℓ increases. Furthermore, since $\mathcal{F}(\cdot)$ is bounded below by $\mathbf{L}$, $\mathcal{F}^\ell(\alpha\mathbf{J}^*)$ converges. Finally, since there exists only one fixed point for $\mathcal{F}(\cdot)$, $\lim_{l\to\infty}\mathcal{F}^\ell(\alpha\mathbf{J}^*) = \mathbf{J}^*$. On the other hand, for the right hand side of (32), as $\mathcal{F}(\cdot) \succeq \mathbf{L}$, we have $\mathcal{F}(\mathbf{L}) \succeq \mathbf{L}$. Applying $\mathcal{F}$ repeatedly gives successively $\mathcal{F}^2(\mathbf{L}) \succeq \mathcal{F}(\mathbf{L})$, $\mathcal{F}^3(\mathbf{L}) \succeq \mathcal{F}^2(\mathbf{L})$, etc. So, $\mathcal{F}^\ell(\mathbf{L})$ is an non-decreasing sequence (with respect to the partial order induced by the cone of p.s.d. matrices). Since $\mathcal{F}(\cdot)$ is upper bounded by $\mathbf{U}$, $\mathcal{F}^\ell(\mathbf{L})$ is a convergent sequence. Again, due to the uniqueness of the fixed point, we have $\lim_{l\to\infty}\mathcal{F}^\ell(\mathbf{L}) = \mathbf{J}^*$. Finally, taking the limit with respect to ℓ on (32), we have $\lim_{l\to\infty}\mathcal{F}^\ell\left(\mathbf{J}^{(0)}\right) = \mathbf{J}^*$, for arbitrary initial $\mathbf{J}^{(0)} \succeq \mathbf{0}$. ■

Remark 7 *According to Theorem 6, the information matrix $\boldsymbol{J}^{(\ell)}_{f_n \to i}$ converges if all initial information matrices are p.s.d., i.e., $\boldsymbol{J}^{(0)}_{f_n \to i} \succeq \boldsymbol{0}$ for all $i \in \mathcal{V}$ and $f_n \in \mathcal{B}(i)$. However, for the pairwise model, the messages are derived based on the classical Gaussian MRF based factorization (in the form of equations (6) and (7)) of the joint distribution. This differs from the model considered in this paper, where the factor f_n follows equation (10), which leads to intrinsically different recursive equations. More specifically, for BP on the*

Gaussian MRF based factorization, the information matrix does not necessarily converge for all initial nonnegative values (for the scalar variable case) as shown in (Malioutov et al., 2006; Moallemi and Roy, 2009a).

Remark 8 *Due to the computation of $\boldsymbol{J}_{f_n\to i}^{(\ell)}$ being independent of the local observations $\boldsymbol{y}_n$, as long as the network topology does not change, the converged value $\boldsymbol{J}_{f_n\to i}^{*}$ can be precomputed offline and stored at each agent, and there is no need to re-compute $\boldsymbol{J}_{f_n\to i}^{*}$ even if $\boldsymbol{y}_n$ varies.*

Another fundamental question is how fast the convergence is, and this is the focus of the discussion below. Since the convergence of a dynamic system is often studied with respect to the part metric (Chueshov, 2002), in the following, we start by introducing the part metric.

Definition 9 *Part (Birkhoff) Metric (Chueshov, 2002): For arbitrary symmetric matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ with the same dimension, if there exists $\alpha \geq 1$ such that $\alpha\boldsymbol{X} \succeq \boldsymbol{Y} \succeq \alpha^{-1}\boldsymbol{X}$, $\boldsymbol{X}$ and $\boldsymbol{Y}$ are called the parts, and $\mathrm{d}(\boldsymbol{X}, \boldsymbol{Y}) \triangleq \inf\left\{\log\alpha : \alpha\boldsymbol{X} \succeq \boldsymbol{Y} \succeq \alpha^{-1}\boldsymbol{X}, \alpha \geq 1\right\}$ defines a metric called the part metric.*

As it is useful to have an estimate of the convergence rate of $\mathbf{J}^{(\ell)}$ in terms of the more standard induced matrix norms, we further introduce the notion of monotone norms. The norms $||\cdot||_2$ and $||\cdot||_F$ (Frobenus norm) are monotone norms.

Definition 10 *Monotone Norm (Ciarlet, 1989, 2.2-10): A matrix norm $\|\cdot\|$ is monotone if*

$$\boldsymbol{X} \succeq \boldsymbol{0}, \boldsymbol{Y} \succeq \boldsymbol{X} \Rightarrow \|\boldsymbol{Y}\| \geq \|\boldsymbol{X}\|.$$

Next, for arbitrary $\epsilon > 0$, we will show that $\left\{\mathbf{J}^{(\ell)}\right\}_{l=1,..}$ approaches the ϵ-neighborhood of the fixed point $\mathbf{J}^{\star}$ exponentially fast with respect to the monotone norm. To this end, for a fixed $\epsilon > 0$, define the set

$$\mathcal{C} = \left\{\mathbf{J}^{(\ell)} | \mathbf{U} \succeq \mathbf{J}^{(\ell)} \succeq \mathbf{J}^{*} + \epsilon\mathbf{I}\right\} \cup \left\{\mathbf{J}^{(\ell)} | \mathbf{J}^{*} - \epsilon\mathbf{I} \succeq \mathbf{J}^{(\ell)} \succeq \mathbf{L}\right\}. \tag{33}$$

Theorem 11 *With the initial message information matrix set to be an arbitrary p.s.d. matrix, i.e., $\boldsymbol{J}_{f_n\to i}^{(0)} \succeq \boldsymbol{0}$, the sequence $\left\{\boldsymbol{J}^{(\ell)}\right\}_{l=0,1,...}$ approaches an arbitrarily small neighborhood of the fixed positive definite matrix $\boldsymbol{J}^{*}$ at an exponential rate with respect to any matrix norm.*

Proof Fix $\epsilon > 0$ and consider the set $\mathcal{C}$ defined in (33). It suffices to show that the quantity $\|\mathbf{J}^{(\ell)} - \mathbf{J}^{*}\|$, where $\|\cdot\|$ is a monotone norm as defined in Definition 10, decays exponentially as long as $\mathbf{J}^{(s)} \in \mathcal{C}$ for all $s \in \{0, 1, \cdots \ell\}$. To this end, for $\mathbf{J}^{(\ell)} \in \mathcal{C}$, and $\mathbf{J}^{*} \notin \mathcal{C}$ (necessarily), according to Definition 9, we have $\mathrm{d}\left(\mathbf{J}^{(\ell)}, \mathbf{J}^{*}\right) \triangleq \inf\left\{\log\alpha : \alpha\mathbf{J}^{(\ell)} \succeq \mathbf{J}^{*} \succeq \alpha^{-1}\mathbf{J}^{(\ell)}\right\}$. Since $\mathrm{d}\left(\mathbf{J}^{(\ell)}, \mathbf{J}^{*}\right)$ is the smallest number satisfying $\alpha\mathbf{J}^{(\ell)} \succeq \mathbf{J}^{*} \succeq \alpha^{-1}\mathbf{J}^{(\ell)}$, this is equivalent to

$$\exp\left\{\mathrm{d}\left(\mathbf{J}^{(\ell)}, \mathbf{J}^{*}\right)\right\}\mathbf{J}^{(\ell)} \succeq \mathbf{J}^{*} \succeq \exp\left\{-\mathrm{d}\left(\mathbf{J}^{(\ell)}, \mathbf{J}^{*}\right)\right\}\mathbf{J}^{(\ell)}. \tag{34}$$

Applying Proposition 4, P 4.1 to (34), we have

$$\mathcal{F}\left(\exp\left\{\mathrm{d}\left(\mathbf{J}^{(\ell)}, \mathbf{J}^{*}\right)\right\}\mathbf{J}^{(\ell)}\right) \succ \mathcal{F}(\mathbf{J}^{*}) \succ \mathcal{F}\left(\exp\left\{-\mathrm{d}\left(\mathbf{J}^{(\ell)}, \mathbf{J}^{*}\right)\right\}\mathbf{J}^{(\ell)}\right).$$

Then applying Proposition 4, P 4.2 and considering that $\exp\left\{\mathrm{d}\left(\mathbf{J}^{(\ell)}, \mathbf{J}^{*}\right)\right\} > 1$ and $\exp\left\{-\mathrm{d}\left(\mathbf{J}^{(\ell)}, \mathbf{J}^{*}\right)\right\} < 1$, we obtain

$$\exp\left\{\mathrm{d}\left(\mathbf{J}^{(\ell)}, \mathbf{J}^{*}\right)\right\}\mathcal{F}\left(\mathbf{J}^{(\ell)}\right) \succeq \mathcal{F}(\mathbf{J}^{*}) \succeq \exp\left\{-\mathrm{d}\left(\mathbf{J}^{(\ell)}, \mathbf{J}^{*}\right)\right\}\mathcal{F}\left(\mathbf{J}^{(\ell)}\right).$$

Notice that, for arbitrary p.d. matrices $\mathbf{X}$ and $\mathbf{Y}$, if $\mathbf{X} - k\mathbf{Y} \succ \mathbf{0}$, then, by definition, we have $\mathbf{x}^T\mathbf{X}\mathbf{x} - k\mathbf{x}^T\mathbf{Y}\mathbf{x} > 0$ for arbitrary $\mathbf{x} \neq \mathbf{0}$. Then, there must exist $o > 0$ that is small enough such that $\mathbf{x}^T\mathbf{X}\mathbf{x} - (k+o)\,\mathbf{x}^T\mathbf{Y}\mathbf{x} > 0$ or equivalently $\mathbf{X} \succ (k+o)\,\mathbf{Y}$. Thus, as $\exp(\cdot)$ is a continuous function, there must exist some $\triangle\mathrm{d} > 0$ such that

$$\exp\left\{-\triangle\mathrm{d} + \mathrm{d}\left(\mathbf{J}^{(\ell)}, \mathbf{J}^{*}\right)\right\}\mathcal{F}\left(\mathbf{J}^{(\ell)}\right) \succ \mathcal{F}(\mathbf{J}^{*}) \succ \exp\left\{\triangle\mathrm{d} - \mathrm{d}\left(\mathbf{J}^{(\ell)}, \mathbf{J}^{*}\right)\right\}\mathcal{F}\left(\mathbf{J}^{(\ell)}\right). \tag{35}$$

Now, using the definition of the part metric, (35) is equivalent to

$$-\triangle\mathrm{d} + \mathrm{d}\left(\mathbf{J}^{(\ell)}, \mathbf{J}^{*}\right) \geq \mathrm{d}\left(\mathcal{F}\left(\mathbf{J}^{(\ell)}\right), \mathcal{F}(\mathbf{J}^{*})\right).$$

Hence, we obtain $\mathrm{d}\left(\mathcal{F}\left(\mathbf{J}^{(\ell)}\right), \mathcal{F}(\mathbf{J}^{*})\right) < \mathrm{d}\left(\mathbf{J}^{(\ell)}, \mathbf{J}^{*}\right)$. Since this result holds for any $\mathbf{J}^{(\ell)} \in \mathcal{C}$, we also have $\mathrm{d}\left(\mathcal{F}\left(\mathbf{J}^{(\ell)}\right), \mathcal{F}(\mathbf{J}^{*})\right) < c\mathrm{d}\left(\mathbf{J}^{(\ell)}, \mathbf{J}^{*}\right)$, where $c = \sup_{\mathbf{J}^{(\ell)}\in\mathcal{C}} \frac{\mathrm{d}\left(\mathcal{F}\left(\mathbf{J}^{(\ell)}\right), \mathcal{F}(\mathbf{J}^{*})\right)}{\mathrm{d}\left(\mathbf{J}^{(\ell)}, \mathbf{J}^{*}\right)} < 1$. Since $\mathbf{J}^{(\ell+1)} = \mathcal{F}\left(\mathbf{J}^{(\ell)}\right)$ and $\mathbf{J}^{*} = \mathcal{F}(\mathbf{J}^{*})$, we have

$$\mathrm{d}\left(\mathbf{J}^{(\ell)}, \mathbf{J}^{*}\right) < c^{\ell}\mathrm{d}\left(\mathbf{J}^{(0)}, \mathbf{J}^{*}\right). \tag{36}$$

According to (Krause and Nussbaum, 1993, Lemma 2.3), the convergence rate of $||\mathbf{J}^{(\ell)} - \mathbf{J}^{*}||$ can be determined by that of $\mathrm{d}\left(\mathbf{J}^{(\ell)}, \mathbf{J}^{*}\right)$. More specifically,

$$||\mathbf{J}^{(\ell)} - \mathbf{J}^{*}|| \leq \left(2\exp\left\{\mathrm{d}\left(\mathbf{J}^{(\ell)}, \mathbf{J}^{*}\right)\right\} - \exp\left\{-\mathrm{d}\left(\mathbf{J}^{(\ell)}, \mathbf{J}^{*}\right)\right\} - 1\right)\min\left\{||\mathbf{J}^{(\ell)}||, ||\mathbf{J}^{*}||\right\}, \tag{37}$$

where $||\cdot||$ is a monotone norm defined on the p.s.d. cone:

As we show in Proposition 4, P 4.3 that $\mathbf{J}^{(\ell)}$ is bounded, then $||\mathbf{J}^{(\ell)}||$ and $||\mathbf{J}^{*}||$ must be finite. Let ζ be the largest value of $\min\left\{||\mathbf{J}^{(\ell)}||, ||\mathbf{J}^{*}||\right\}$ for all $\{\mathbf{J}^{(\ell)}\}$ with $\ell \geq 0$, then $\zeta > 0$. According to (36) and (37), we have that

$$||\mathbf{J}^{(\ell)} - \mathbf{J}^{*}|| < \zeta\left(2\exp\left\{c^{\ell}d_0\right\} - \exp\left\{-c^{\ell}d_0\right\} - 1\right), \tag{38}$$

with $0 < c < 1$ and $d_0 = \mathrm{d}\left(\mathbf{J}^{(0)}, \mathbf{J}^{*}\right)$, which is a constant. The above inequality is equivalent to

$$||\mathbf{J}^{(\ell)} - \mathbf{J}^{*}|| < \zeta\left(3\exp\left\{c^{\ell}d_0\right\} - \exp\left\{c^{\ell}d_0\right\} - \exp\left\{-c^{\ell}d_0\right\} - 1\right). \tag{39}$$

Since both $\exp\left\{c^\ell d_0\right\}$ and $\exp\left\{-c^\ell d_0\right\}$ are positive and $\exp\left\{c^\ell d_0\right\}\exp\left\{-c^\ell d_0\right\} = 1$, according to the arithmetic-geometric mean inequality, we have $\exp\left\{c^\ell d_0\right\} + \exp\left\{-c^\ell d_0\right\} \geq 2\left(\exp\left\{c^\ell d_0\right\}\exp\left\{-c^\ell d_0\right\}\right)^{1/2} = 2$. Then, the right-hand side of (39) is further amplified, and we obtain

$$||\mathbf{J}^{(\ell)} - \mathbf{J}^*|| < \zeta\left(3\exp\left\{c^\ell d_0\right\} - 3\right) = 3\zeta\left(\exp\left\{c^\ell d_0\right\} - 1\right).$$

Therefore, the sequence $\left\{\mathbf{J}^{(\ell)}\right\}_{l=0,1,\ldots}$ approaches the ϵ-neighborhood (and hence any arbitrarily small neighborhood) of the fixed positive definite matrix $\mathbf{J}^*$ at an exponential rate with respect to any matrix norm. ■

The physical meaning of Theorem 11 is that the distance between $\mathbf{J}^{(\ell)}$ and $\mathbf{J}^*$ decreases exponentially fast before $\mathbf{J}^{(\ell)}$ enters $\mathbf{J}^*$'s neighborhood, which can be chosen to be arbitrarily small. Next, we study how to choose the initial value $\mathbf{J}^{(0)}$ so that $\mathbf{J}^{(\ell)}$ converges faster.

Theorem 12 *With $\mathbf{0} \preceq \boldsymbol{J}^{(0)} \preceq \boldsymbol{L}$, $\boldsymbol{J}^{(\ell)}$ is a monotonic increasing sequence, and $\boldsymbol{J}^{(\ell)}$ converges most rapidly with $\boldsymbol{J}^{(0)} = \boldsymbol{L}$. Moreover, with $\boldsymbol{J}^{(0)} \succeq \boldsymbol{U}$, $\boldsymbol{J}^{(\ell)}$ is a monotonic decreasing sequence, and $\boldsymbol{J}^{(\ell)}$ converges most rapidly with $\boldsymbol{J}^{(0)} = \boldsymbol{U}$.*

Proof Following Proposition 4, P 4.3, it can be verified that for $\mathbf{0} \preceq \mathbf{J}^{(0)} \preceq \mathbf{L}$, we have $\mathbf{J}^{(1)} \succeq \mathbf{J}^{(0)}$. Then, according to Proposition 4, P 4.1, and by induction, this relationship can be extended to $\mathbf{J}^{(\ell)} \succeq \ldots \mathbf{J}^{(1)} \succeq \mathbf{J}^{(0)}$, which states that $\mathbf{J}^{(\ell)}$ is a monotonic increasing sequence. Now, suppose that there are two sequences $\mathbf{J}^{(\ell)}$ and $\widetilde{\mathbf{J}}^{(\ell)}$ that are started with different initial values $\mathbf{0} \preceq \mathbf{J}^{(0)} \prec \mathbf{L}$ and $\mathbf{0} \preceq \widetilde{\mathbf{J}}^{(0)} \prec \mathbf{L}$, respectively. Then these two sequences are monotonically increasing and bounded by $\mathbf{J}^*$. To prove that $\mathbf{J}^{(0)} = \mathbf{L}$ leads to the fastest convergence, it is sufficient to prove that $\mathbf{J}^{(\ell)} \succ \widetilde{\mathbf{J}}^{(\ell)}$ for $\ell = 0, 1\ldots$. First, note that $\mathbf{J}^{(0)} \succ \widetilde{\mathbf{J}}^{(0)}$. Assume $\mathbf{J}^{(n)} \succ \widetilde{\mathbf{J}}^{(n)}$ for some $n \geq 0$. According to Proposition 4, P 4.1, we have $\mathcal{F}\left(\mathbf{J}^{(n)}\right) \succeq \mathcal{F}\left(\widetilde{\mathbf{J}}^{(n)}\right)$, or equivalently $\mathbf{J}^{(n+1)} \succeq \widetilde{\mathbf{J}}^{(n+1)}$. Therefore, by induction, we have proven that, with $\mathbf{J}^{(0)} = \mathbf{L}$, $\mathbf{J}^{(\ell)}$ converges more rapidly than with any other initial value $\mathbf{0} \preceq \mathbf{J}^{(0)} \prec \mathbf{L}$.

With similar logic, we can show that, with $\mathbf{J}^{(0)} \succeq \mathbf{U}$, $\mathbf{J}^{(\ell)}$ is a monotonic decreasing sequence; and, with $\mathbf{J}^{(0)} = \mathbf{U}$, $\mathbf{J}^{(\ell)}$ converges more rapidly than that with any other initial value $\mathbf{J}^{(0)} \succ \mathbf{U}$. ■

Notice that it is a common practice in the Gaussian BP literature that the initial information matrix (or inverse variance for the scalar case) is set to be $\mathbf{0}$, i.e., $\mathbf{J}^{(0)}_{f_n \to i} = \mathbf{0}$ (Weiss and Freeman, 2001a; Malioutov et al., 2006). Theorem 12 reveals that there is a better choice to guarantee faster convergence.

4.2 Convergence of Message Mean Vector

According to Theorems 6 and 11, as long as we choose $\mathbf{J}^{(0)}_{f_k \to j} \succeq \mathbf{0}$ for all $j \in \mathcal{V}$ and $f_k \in \mathcal{B}(j)$, the distance between $\mathbf{J}^{(\ell)}_{f_k \to j}$ and $\mathbf{J}^*_{f_k \to j}$ decreases exponentially fast before $\mathbf{J}^{(\ell)}_{f_k \to j}$ enters $\mathbf{J}^*_{f_k \to j}$'s neighborhood, which can be chosen to be arbitrarily small. Furthermore,

according to (16), $\left[\mathbf{J}^{(\ell)}_{j \to f_n}\right]^{-1}$ also converges to a p.d. matrix once $\mathbf{J}^{(\ell)}_{f_k \to j}$ converges, and the converged value for $\left[\mathbf{J}^{(\ell)}_{j \to f_n}\right]^{-1}$ is denoted by $\left[\mathbf{J}^*_{j \to f_n}\right]^{-1}$. Then for arbitrary initial value $\mathbf{v}^{(0)}_{f_k \to j}$, the evolution of $\mathbf{v}^{(\ell)}_{j \to f_n}$ in (17) can be written in terms of the converged message information matrices, which is

$$\mathbf{v}^{(\ell)}_{j \to f_n} = \left[\mathbf{J}^*_{j \to f_n}\right]^{-1} \sum_{f_k \in \mathcal{B}(j) \setminus f_n} \mathbf{J}^*_{f_k \to j} \mathbf{v}^{(\ell-1)}_{f_k \to j}. \tag{40}$$

Using (20), and replacing indices j, i, n with z, j, k respectively, $\mathbf{v}^{(\ell-1)}_{f_k \to j}$ is given by

$$\mathbf{v}^{(\ell-1)}_{f_k \to j} = \left[\mathbf{J}^*_{f_k \to j}\right]^{-1} \mathbf{A}^T_{k,j} \underbrace{\left[\mathbf{R}_k + \sum_{z \in \mathcal{B}(f_k) \setminus j} \mathbf{A}_{k,z} \left[\mathbf{J}^*_{z \to f_k}\right]^{-1} \mathbf{A}^T_{k,z}\right]}_{\triangleq \mathbf{M}_{k,j}}{}^{-1} \left(\mathbf{y}_k - \sum_{z \in \mathcal{B}(f_k) \setminus j} \mathbf{A}_{k,z} \mathbf{v}^{(\ell-1)}_{z \to f_k}\right). \tag{41}$$

Putting (41) into (40), we have

$$\mathbf{v}^{(\ell)}_{j \to f_n} = \mathbf{b}_{j \to f_n} - \sum_{f_k \in \mathcal{B}(j) \setminus f_n} \sum_{z \in \mathcal{B}(f_k) \setminus j} \left[\mathbf{J}^*_{j \to f_n}\right]^{-1} \mathbf{A}^T_{k,j} \mathbf{M}^{-1}_{k,j} \mathbf{A}_{k,z} \mathbf{v}^{(\ell-1)}_{z \to f_k}, \tag{42}$$

where $\mathbf{b}_{j \to f_n} = \left[\mathbf{J}^*_{j \to f_n}\right]^{-1} \sum_{f_k \in \mathcal{B}(j) \setminus f_n} \mathbf{A}^T_{k,j} \mathbf{M}^{-1}_{k,j} \mathbf{y}_k$. The above equation can be further written in compact form as

$$\mathbf{v}^{(\ell)}_{j \to f_n} = \mathbf{b}_{j \to f_n} - \mathbf{Q}_{j \to f_n} \mathbf{v}^{(\ell-1)},$$

with the column vector $\mathbf{v}^{(\ell-1)}$ containing $\mathbf{v}^{(\ell-1)}_{z \to f_k}$ for all $z \in \mathcal{V}$ and $f_k \in \mathcal{B}(z)$ as subvector with ascending index first on z and then on k. The matrix $\mathbf{Q}_{j \to f_n}$ is a row block matrix with component block $\left[\mathbf{J}^*_{j \to f_n}\right]^{-1} \mathbf{A}^T_{k,j} \mathbf{M}^{-1}_{k,j} \mathbf{A}_{k,z}$ if $f_k \in \mathcal{B}(j) \setminus f_n$ and $z \in \mathcal{B}(f_k) \setminus j$, and $\mathbf{0}$ otherwise. Let $\mathbf{Q}$ be the block matrix that stacks $\mathbf{Q}_{j \to f_n}$ with the order first on j and then on n, and $\mathbf{b}$ be the vector containing $\mathbf{b}_{j \to f_n}$ with the same stacking order as $\mathbf{Q}_{j \to f_n}$. We have

$$\mathbf{v}^{(\ell)} = -\mathbf{Q}\mathbf{v}^{(\ell-1)} + \mathbf{b}, \quad \ell \geq 1, 2, \ldots. \tag{43}$$

It is known that for arbitrary initial value $\mathbf{v}^{(0)}$, $\mathbf{v}^{(\ell)}$ converges if and only if the spectral radius $\rho(\mathbf{Q}) < 1$ (Demmel, 1997, pp. 280). Since the elements of $\mathbf{v}^{(0)}$, i.e., $\mathbf{v}^{(0)}_{j \to f_n}$, depends on $\mathbf{v}^{(0)}_{f_k \to j}$, we can choose arbitrary $\mathbf{v}^{(0)}_{f_k \to j}$. Furthermore, as $\mathbf{v}^{(\ell)}$ depends on the convergence of $\mathbf{J}^{(\ell)}$, we have the following result.

Theorem 13 *The vector sequence $\left\{\boldsymbol{v}^{(\ell)}\right\}_{l=1,2,\ldots}$ defined by (43) converges to a unique value under any initial value $\left\{\boldsymbol{v}^{(0)}_{f_k \to j}\right\}_{k \in \mathcal{V}, j \in \mathcal{B}(f_k)}$ and initial message information matrix $\boldsymbol{J}^{(0)} \succeq \mathbf{0}$ if and only if $\rho(\mathbf{Q}) < 1$.*

The row block matrix $\mathbf{Q}_j$, a row block of $\mathbf{Q}$, contains only block entries $\mathbf{0}$ and $\mathbf{Q}_{j\to f_n}$. When the observation model (1) reduces to the pairwise model, where only two unknown variables are involved in each local observation, it can be shown that $\mathbf{Q}_j$ and $\mathbf{Q}_i$ are orthogonal if $i \neq j$. A distributed convergence condition is obtained utilizing this orthogonal property in Du et al. (2017a). However, for the more general case studied in this paper, properties of $\mathbf{Q}_j$ and $\mathbf{Q}$ need to be further exploited to show when $\rho(\mathbf{Q}) < 1$ is satisfied.

In the sequel, we will show that $\rho(\mathbf{Q}) < 1$ is satisfied for a single loop factor graph with multiple chains/trees (an example is shown in Fig. 2), thus Gaussian BP converges in such a topology. Although Weiss (2000) shows the convergence of Gaussian BP on the MRF with a single loop, the analysis cannot be applied here since the local observations model (1) is different from the pairwise model in (Weiss, 2000).

Theorem 14 *For any factor graph that is the union of a single loop and a forest, with arbitrary positive semi-definite initial information matrix, i.e., $\boldsymbol{J}^{(0)}_{f_n\to i} \succeq \boldsymbol{0}$ for all $i \in \mathcal{V}$ and $f_n \in \mathcal{B}(i)$, the message information matrix $\boldsymbol{J}^{(\ell)}_{f_n\to i}$ and mean vector $\boldsymbol{v}^{(\ell)}_{i\to f_n}$ is guaranteed to converge to their corresponding unique points.*

Proof In this proof, Fig. 2 is being used as a reference throughout. For a single loop factor graph with chains/trees as shown in Fig. 2 (a), there are two kinds of nodes. One is the factors/variables in the loop, and they are denoted by $f_n/\mathbf{x}_j$. The other is the factors/variables on the chains/trees but outside the loop, denoted as $\widetilde{f}_k/\widetilde{\mathbf{z}}_i$. Then message from a variable node to a neighboring factor node on the graph can be categorized into three groups:

1) message on a tree/chain passing towards the loop, e.g., $m^{*}_{\widetilde{z}\to f_k}(\widetilde{\mathbf{x}}_z)$ and $m^{*}_{\widetilde{s}\to \widetilde{f}_k}(\widetilde{\mathbf{x}}_s)$;

2) message on a tree/chain passing away from the loop, e.g., $m^{(\ell)}_{j\to \widetilde{f}_k}(\mathbf{x}_j)$, $m^{(\ell)}_{\widetilde{s}\to \widetilde{f}_s}(\mathbf{x}_s)$ and $m^{(\ell)}_{\widetilde{z}\to \widetilde{f}_z}(\mathbf{x}_z)$;

3) message in the loop, e.g., $m^{(\ell)}_{j\to f_n}(\mathbf{x}_j)$, $m^{(\ell)}_{z\to f_k}(\mathbf{x}_z)$ and $m^{(\ell)}_{i\to f_n}(\mathbf{x}_i)$.

According to (11), computation of the messages in the first group does not depend on messages in the loop and is thus convergence guaranteed. Therefore, the message iteration number is replaced with a $*$ to denote the converged message. Also, from the definition of message computation in (11), if messages in the third group converge, the second group messages should also converge. Therefore, we next focus on showing the convergence of messages in the third group.

For a factor node f_k in the loop with $\mathbf{x}_z$ and $\mathbf{x}_j$ being its two neighboring variable nodes in the loop and $\widetilde{\mathbf{x}}_z$ being its neighboring variable node outside the loop, according to the definition of message computation in (12), we have

$$\begin{aligned} m^{(\ell)}_{f_k\to j}(\mathbf{x}_j) &= \int\int f_k \times m^{(\ell)}_{z\to f_k}(\mathbf{x}_z) \prod_{\widetilde{z}\in\mathcal{B}(f_k)\setminus j} m^{*}_{\widetilde{z}\to f_k}(\widetilde{\mathbf{x}}_z)\ \mathrm{d}\{\widetilde{\mathbf{x}}_z\}_{\widetilde{z}\in\mathcal{B}(f_k)\setminus j}\ \mathrm{d}\mathbf{x}_z, \\ &= \int m^{(\ell)}_{z\to f_k}(\mathbf{x}_z) \left[\int f_k \times \prod_{\widetilde{z}\in\mathcal{B}(f_k)\setminus j} m^{*}_{\widetilde{z}\to f_k}(\widetilde{\mathbf{x}}_z)\ \mathrm{d}\{\widetilde{\mathbf{x}}_z\}_{\widetilde{z}\in\mathcal{B}(f_k)\setminus j}\right] \mathrm{d}\mathbf{x}_z. \end{aligned} \tag{44}$$

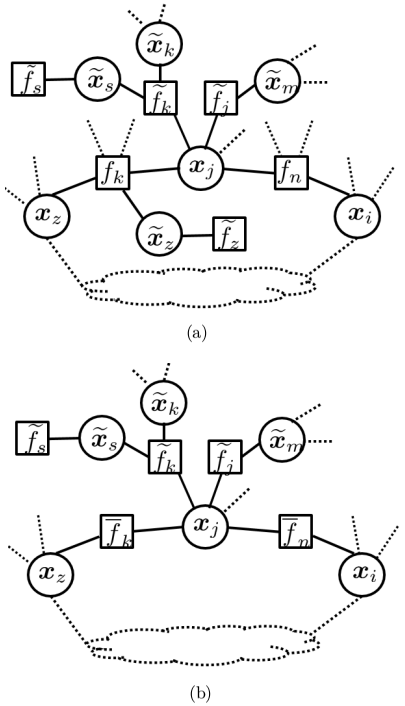


Figure 2: (a) An example of factor graph with a single loop and chains/trees topology, where the dashed line indicates possible chains/trees; (b) The equivalent factor graph of Fig 2 (a) with new factor functions that do not have neighboring variable nodes except those in the loop.

As shown in Lemma 2, $m^*_{\widetilde{z}\to f_k}(\widetilde{\mathbf{x}}_z)$ must be in Gaussian function form, which is denoted by $m^*_{\widetilde{z}\to f_k}(\widetilde{\mathbf{x}}_z) \propto \mathcal{N}\left(\widetilde{\mathbf{x}}_z|\mathbf{v}^*_{\widetilde{z}\to f_k}, \left[\mathbf{J}^*_{\widetilde{z}\to f_k}\right]^{-1}\right)$. Besides, from (1) we obtain

$$f_k = \mathcal{N}\left(\mathbf{y}_k|\mathbf{A}_{k,z}\mathbf{x}_z + \mathbf{A}_{k,j}\mathbf{x}_j + \sum_{\widetilde{z}\in\mathcal{B}(f_k)}\mathbf{A}_{k,\widetilde{z}}\widetilde{\mathbf{x}}_z, \mathbf{R}_k\right).$$

It can be shown that the inner integration in the second line of (44) is given by

$$\mathcal{N}\left(\overline{\mathbf{y}}_k|\overline{\mathbf{A}}_{k,z}\mathbf{x}_z + \overline{\mathbf{A}}_{k,j}\mathbf{x}_j, \overline{\mathbf{R}}_k\right) \triangleq \overline{f}_k,$$

where the overbar is used to denote the new constant matrix or vector. Then (44) can be written as

$$m^{(\ell)}_{f_k\to j}(\mathbf{x}_j) = \int \overline{f}_k \times m^{(\ell)}_{z\to f_k}(\mathbf{x}_z)\,\mathrm{d}\mathbf{x}_z. \tag{45}$$

Comparing (45) with (12), we obtain $m^{(\ell)}_{f_k\to j}(\mathbf{x}_j)$ as if $m^{(\ell)}_{\overline{f}_k\to j}(\mathbf{x}_j)$ is being passed to a factor node $\overline{f}_k$. Therefore, a factor graph with a single loop and multiple trees/chains is equivalent to a single loop factor graph in which each factor node has no neighboring variable node outside the loop. As a result, the example of Fig. 2 (a) is equivalent to Fig. 2 (b). In the following, we focus on this equivalent topology for the convergence analysis.

Note that, for arbitrary variable node j in the loop, there are two neighboring factor nodes in the loop. Further, using the notation for the equivalent topology, (42) is reduced to

$$\begin{aligned}\mathbf{v}^{(\ell)}_{j\to\overline{f}_n} =& -\left[\mathbf{J}^*_{j\to\overline{f}_n}\right]^{-1}\overline{\mathbf{A}}^T_{k,j}\mathbf{T}^{-1}_{k,j}\overline{\mathbf{A}}_{k,z}\mathbf{v}^{(\ell-1)}_{z\to\overline{f}_k}\\ &+\underbrace{\overline{\mathbf{b}}_{j\to\overline{f}_n} - \sum_{\widetilde{f}_k\in\mathcal{B}(j)\setminus f_n}\sum_{\widetilde{s}\in\mathcal{B}(\widetilde{f}_k)\setminus j}\left[\mathbf{J}^*_{j\to\overline{f}_n}\right]^{-1}\overline{\mathbf{A}}^T_{k,j}\overline{\mathbf{M}}^{-1}_{k,j}\overline{\mathbf{A}}_{k,\widetilde{s}}\mathbf{v}^*_{\widetilde{s}\to\widetilde{f}_k}}_{\triangleq\mathbf{c}_{j\to\overline{f}_n}},\end{aligned} \tag{46}$$

where $\mathbf{v}^*_{\widetilde{s}\to\widetilde{f}_k}$ is the converged mean vector on the chain/tree;

$$\overline{\mathbf{b}}_{j\to\overline{f}_n} = \left[\mathbf{J}^*_{j\to\overline{f}_n}\right]^{-1}\sum_{\overline{f}_k\in\mathcal{B}(j)\setminus\overline{f}_n}\overline{\mathbf{A}}^T_{k,j}\overline{\mathbf{M}}^{-1}_{k,j}\overline{\mathbf{y}}_k$$

with $\overline{\mathbf{M}}_{k,j} = \overline{\mathbf{R}}_k + \sum_{\widetilde{s}\in\mathcal{B}(\widetilde{f}_k)\setminus j}\overline{\mathbf{A}}_{k,\widetilde{s}}\left[\mathbf{J}^*_{\widetilde{s}\to\widetilde{f}_k}\right]^{-1}\overline{\mathbf{A}}^T_{k,\widetilde{s}}$, and

$$\mathbf{T}_{k,j} = \overline{\mathbf{R}}_k + \overline{\mathbf{A}}_{k,z}\left[\mathbf{J}^*_{z\to\overline{f}_k}\right]^{-1}\overline{\mathbf{A}}^T_{k,z}, \tag{47}$$

with $\mathbf{x}_z$ and $\overline{f}_k$ in the loop where $\overline{f}_k \in \mathcal{B}(j)\setminus\overline{f}_n$ and $\mathbf{x}_z \in \mathcal{B}\left(\overline{f}_k\right)\setminus j$. By multiplying $\left[\mathbf{J}^*_{j\to\overline{f}_n}\right]^{1/2}$ on both sides of (46), and defining $\boldsymbol{\beta}^{(\ell)}_{j\to\overline{f}_n} = \left[\mathbf{J}^*_{j\to\overline{f}_n}\right]^{1/2}\mathbf{v}^{(\ell)}_{j\to\overline{f}_n}$, we have

$$\boldsymbol{\beta}^{(\ell)}_{j\to\overline{f}_n} = -\left[\mathbf{J}^*_{j\to\overline{f}_n}\right]^{-1/2}\overline{\mathbf{A}}^T_{k,j}\mathbf{T}^{-1}_{k,j}\overline{\mathbf{A}}_{k,z}\left[\mathbf{J}^*_{z\to\overline{f}_k}\right]^{-1/2}\boldsymbol{\beta}^{(\ell-1)}_{z\to\overline{f}_k} + \left[\mathbf{J}^*_{j\to\overline{f}_n}\right]^{1/2}\mathbf{c}_{j\to\overline{f}_n}, \tag{48}$$

Let $\boldsymbol{\beta}^{(\ell-1)}$ contain $\boldsymbol{\beta}^{(\ell-1)}_{z\to\overline{f}_k}$ for all $\mathbf{x}_z$ with $z\in\mathcal{B}\left(\overline{f}_k\right)$ and $\overline{f}_k$ being in the loop, and the index is arranged first on k and then on z. Then, the above equation is written in a compact form as

$$\boldsymbol{\beta}^{(\ell)}_{j\to\overline{f}_n} = -\mathbf{Q}_{j\to\overline{f}_n}\boldsymbol{\beta}^{(\ell-1)} + \left[\mathbf{J}^*_{j\to\overline{f}_n}\right]^{1/2}\mathbf{c}_{j\to\overline{f}_n}, \tag{49}$$

where $\mathbf{Q}_{j\to\overline{f}_n}$ is a row block matrix with the only nonzero block

$$\left[\mathbf{J}^*_{j\to\overline{f}_n}\right]^{-1/2}\overline{\mathbf{A}}^T_{k,j}\mathbf{T}^{-1}_{k,j}\overline{\mathbf{A}}_{k,z}\left[\mathbf{J}^*_{z\to\overline{f}_k}\right]^{-1/2}$$

located at the position corresponding to the position $\boldsymbol{\beta}^{(\ell)}_{z\to\overline{f}_k}$ in $\boldsymbol{\beta}^{(\ell)}$. Then let $\mathbf{Q}$ be a matrix that stacks $\mathbf{Q}_{j\to\overline{f}_n}$ as its row, where j and $\overline{f}_n$ are in the loop with $j\in\mathcal{B}\left(\overline{f}_n\right)$. Besides, let $\mathbf{c}$ be the vector containing the subvector $\left[\mathbf{J}^*_{j\to\overline{f}_n}\right]^{1/2}\mathbf{c}_{j\to\overline{f}_n}$ with the same order as $\mathbf{Q}_{j\to\overline{f}_n}$ in $\mathbf{Q}$. We have

$$\boldsymbol{\beta}^{(\ell)} = -\mathbf{Q}\boldsymbol{\beta}^{(\ell-1)} + \mathbf{c}. \tag{50}$$

Since $\mathbf{Q}$ is a square matrix, $\rho(\mathbf{Q}) \le \sqrt{\rho\left(\mathbf{Q}\mathbf{Q}^T\right)}$ and therefore $\rho\left(\mathbf{Q}\mathbf{Q}^T\right) < 1$ is the sufficient condition for the convergence of $\boldsymbol{\beta}^{(\ell)}$. We next investigate the elements in $\mathbf{Q}\mathbf{Q}^T$.

Due to the single loop structure of the graph, every $\boldsymbol{\beta}^{(\ell)}_{j\to\overline{f}_n}$ in (48) would be dependent on a unique $\boldsymbol{\beta}^{(\ell)}_{z\to\overline{f}_k}$, where $\overline{f}_k\in\mathcal{B}(j)\setminus\overline{f}_n$ and $z\in\mathcal{B}\left(\overline{f}_k\right)\setminus j$ (i.e., the message two hops backward along the loop in the factor graph). Thus, the position of the non-zero block in $\mathbf{Q}_{j\to\overline{f}_n}$ will be different and non-overlapping for different combinations of $(j,\overline{f}_n)$. As a result, there exists a column permutation matrix $\boldsymbol{\Xi}$ such that $\mathbf{Q}\boldsymbol{\Xi}$ is a block diagonal matrix. Therefore, $(\mathbf{Q}\boldsymbol{\Xi})(\mathbf{Q}\boldsymbol{\Xi})^T = \mathbf{Q}\mathbf{Q}^T$ is also a diagonal matrix, and we can write

$$\mathbf{Q}\mathbf{Q}^T = \mathtt{Bdiag}\left\{\mathbf{Q}_{j\to\overline{f}_n}\mathbf{Q}^T_{j\to\overline{f}_n}\right\}_{j\,\in\,\mathcal{B}\left(\overline{f}_n\right)}.$$

As a consequence, $\rho\left(\mathbf{Q}\mathbf{Q}^T\right) < 1$ is equivalent to $\rho\left(\mathbf{Q}_{j\to\overline{f}_n}\mathbf{Q}^T_{j\to\overline{f}_n}\right) < 1$ for all j and $\overline{f}_n$ in the loop with $j\in\mathcal{B}\left(\overline{f}_n\right)$. Following the definition of $\mathbf{Q}_{j\to\overline{f}_n}$ below (49), we obtain

$$\begin{aligned}\mathbf{Q}_{j\to\overline{f}_n}\mathbf{Q}^T_{j\to\overline{f}_n} &= \left[\mathbf{J}^*_{j\to\overline{f}_n}\right]^{-1/2}\overline{\mathbf{A}}^T_{k,j}\mathbf{T}^{-1}_{k,j}\overline{\mathbf{A}}_{k,z}\left[\mathbf{J}^*_{z\to\overline{f}_k}\right]^{-1}\overline{\mathbf{A}}^T_{k,z}\mathbf{T}^{-1}_{k,j}\overline{\mathbf{A}}_{k,j}\left[\mathbf{J}^*_{j\to\overline{f}_n}\right]^{-1/2}\\ &= \left[\mathbf{J}^*_{j\to\overline{f}_n}\right]^{-1/2}\overline{\mathbf{A}}^T_{k,j}\mathbf{T}^{-1}_{k,j}\left(\mathbf{T}_{k,j} - \overline{\mathbf{R}}_k\right)\mathbf{T}^{-1}_{k,j}\overline{\mathbf{A}}_{k,j}\left[\mathbf{J}^*_{j\to\overline{f}_n}\right]^{-1/2},\end{aligned} \tag{51}$$

where the second equation follows from the definition of $\mathbf{T}_{k,j}$ in (47). Besides, since $\overline{\mathbf{R}}_k \succ \mathbf{0}$, we have $\mathbf{T}_{k,j} - \overline{\mathbf{R}}_k \prec \mathbf{T}_{k,j}$. Following P B.2 in Appendix B, and due to $\mathbf{T}_{k,j} = \mathbf{T}^T_{k,j}$, we have

$$\mathbf{T}^{-1/2}_{k,j}\left(\mathbf{T}_{k,j} - \overline{\mathbf{R}}_k\right)\mathbf{T}^{-1/2}_{k,j} \prec \mathbf{I}. \tag{52}$$

Applying P B.2 in Appendix B again to (52), and making use of (51), we obtain

$$\mathbf{Q}_{j\to\overline{f}_n}\mathbf{Q}^T_{j\to\overline{f}_n} \prec \left[\mathbf{J}^*_{j\to\overline{f}_n}\right]^{-1/2}\overline{\mathbf{A}}^T_{k,j}\mathbf{T}^{-1}_{k,j}\overline{\mathbf{A}}_{k,j}\left[\mathbf{J}^*_{j\to\overline{f}_n}\right]^{-1/2}. \tag{53}$$

According to (47), we have

$$\overline{\mathbf{A}}_{k,j}^T \mathbf{T}_{k,j}^{-1} \overline{\mathbf{A}}_{k,j} = \overline{\mathbf{A}}_{k,j}^T \left[\overline{\mathbf{R}}_k + \overline{\mathbf{A}}_{k,z} \left[\mathbf{J}^*_{z \to \overline{f}_k} \right]^{-1} \overline{\mathbf{A}}_{k,z}^T \right]^{-1} \overline{\mathbf{A}}_{k,j}. \tag{54}$$

On the other hand, using (19), due to $\mathcal{B}\left(\overline{f}_k\right) \setminus j = \mathbf{x}_z$ in the considered topology, the right hand side of (54) is $\mathbf{J}^*_{\overline{f}_k \to j}$. Therefore, (53) is further written as

$$\mathbf{Q}_{j \to \overline{f}_n} \mathbf{Q}_{j \to \overline{f}_n}^T \prec \left[\mathbf{J}^*_{j \to \overline{f}_n} \right]^{-1/2} \mathbf{J}^*_{\overline{f}_k \to j} \left[\mathbf{J}^*_{j \to \overline{f}_n} \right]^{-1/2}. \tag{55}$$

From (16), $\mathbf{J}^*_{j \to \overline{f}_n} = \mathbf{W}_j^{-1} + \mathbf{J}^*_{\overline{f}_k \to j} + \sum_{\widetilde{f}_k \in \mathcal{B}(j) \setminus \overline{f}_n} \mathbf{J}^*_{\widetilde{f}_k \to j}$, thus $\mathbf{J}^*_{\overline{f}_k \to j} \preceq \mathbf{J}^*_{j \to \overline{f}_n}$. Therefore, $\left[\mathbf{J}^*_{j \to \overline{f}_n} \right]^{-1/2} \mathbf{J}^*_{\overline{f}_k \to j} \left[\mathbf{J}^*_{j \to \overline{f}_n} \right]^{-1/2} \preceq \mathbf{I}$, and, together with (55), we have

$$\mathbf{Q}_{j \to \overline{f}_n} \mathbf{Q}_{j \to \overline{f}_n}^T \prec \mathbf{I}.$$

Hence $\rho\left(\mathbf{Q}_{j \to \overline{f}_n} \mathbf{Q}_{j \to \overline{f}_n}^T \right) < 1$ for all j and $\overline{f}_n$ in the loop and $j \in \mathcal{B}\left(\overline{f}_n\right)$, and equivalently $\rho(\mathbf{Q}) < 1$. This completes the proof. ■

4.3 Convergence of Belief Covariance and Mean Vector

As the computation of the belief covariance $\mathbf{P}_i^{(\ell)}$ depends on the message information matrix $\mathbf{J}^{(\ell)}_{f_n \to i}$, using Theorems 6 and 11, we can derive the convergence and uniqueness properties of $\mathbf{P}_i^{(\ell)}$.

Before we present the main result, we first present some properties of the part metric $\mathrm{d}(\mathbf{X}, \mathbf{Y})$, with positive definite arguments $\mathbf{X}$, $\mathbf{Y}$, and $\triangle\mathbf{X}$. The proofs are provided in Appendix D.

Proposition 15 *The part metric* $\mathrm{d}(\boldsymbol{X}, \boldsymbol{Y})$ *satisfies the following properties*

P 15.1: $\mathrm{d}(\mathbf{X}_1 + \mathbf{X}_2, \mathbf{Y}_1 + \mathbf{Y}_2) \leq \mathrm{d}(\mathbf{X}_1, \mathbf{Y}_1) + \mathrm{d}(\mathbf{X}_2, \mathbf{Y}_2)$;
P 15.2: $\mathrm{d}(\mathbf{X}, \mathbf{Y}) = \mathrm{d}\left(\mathbf{X}^{-1}, \mathbf{Y}^{-1}\right)$.

We now have the following result.

Corollary 16 *With arbitrary initial message information matrix* $\boldsymbol{J}^{(0)}_{f_n \to i} \succeq \boldsymbol{0}$ *for all* $i \in \mathcal{V}$ *and* $f_n \in \mathcal{B}(i)$*, the belief covariance matrix* $\boldsymbol{P}_i^{(\ell)}$ *converges to a unique p.d. matrix at an exponential rate with respect to any matrix norm before* $\boldsymbol{P}_i^{(\ell)}$ *enters* $\boldsymbol{P}_i^*$*'s neighborhood, which can be chosen to be arbitrarily small.*

Proof Since $\mathbf{J}^{(\ell)}_{f_n \to i}$ converges to a p.d. matrix, and according to (23), $\mathbf{P}_i^{(\ell)}$ also converges. Below, we study the convergence rate of $\mathbf{P}_i^{(\ell)}$. According to the definition of $\mathbf{P}_i^{(\ell)}$ in (23)

and part metric in Definition 9, we have

$$\mathrm{d}\left([\mathbf{P}_i^{(\ell)}]^{-1}, [\mathbf{P}_i^*]^{-1} \right) = \mathrm{d}\left(\mathbf{W}_i^{-1} + \sum_{f_n \in \mathcal{B}(i)} \mathbf{J}^{(\ell)}_{f_n \to i}, \mathbf{W}_i^{-1} + \sum_{f_n \in \mathcal{B}(i)} \mathbf{J}^*_{f_n \to i} \right).$$

By applying P 15.1 to the above equation, we obtain

$$\mathrm{d}\left([\mathbf{P}_i^{(\ell)}]^{-1}, [\mathbf{P}_i^*]^{-1} \right) \leq \mathrm{d}\left(\mathbf{W}_i^{-1}, \mathbf{W}_i^{-1} \right) + \sum_{f_n \in \mathcal{B}(i)} \mathrm{d}\left(\mathbf{J}^{(\ell)}_{f_n \to i}, \mathbf{J}^*_{f_n \to i} \right) = \sum_{f_n \in \mathcal{B}(i)} \mathrm{d}\left(\mathbf{J}^{(\ell)}_{f_n \to i}, \mathbf{J}^*_{f_n \to i} \right).$$

According to (36), for all $i \in \mathcal{V}$ and $f_n \in \mathcal{B}(i)$, there exist a $c < 1$ such that

$$\mathrm{d}\left(\mathbf{J}^{(\ell)}_{f_n \to i}, \mathbf{J}^*_{f_n \to i} \right) < c^\ell \mathrm{d}\left(\mathbf{J}^{(0)}_{f_n \to i}, \mathbf{J}^*_{f_n \to i} \right).$$

Applying the above inequality to compute $\left[\mathbf{P}_i^{(\ell)} \right]^{-1}$ in (24), we obtain

$$\mathrm{d}\left([\mathbf{P}_i^{(\ell)}]^{-1}, [\mathbf{P}_i^*]^{-1} \right) < c^\ell \sum_{f_n \in \mathcal{B}(i)} \mathrm{d}\left(\mathbf{J}^{(0)}_{f_n \to i}, \mathbf{J}^*_{f_n \to i} \right).$$

Following P 15.2, the above inequality is equivalent to

$$\mathrm{d}\left(\mathbf{P}_i^{(\ell)}, \mathbf{P}_i^* \right) < c^\ell \sum_{f_n \in \mathcal{B}(i)} \mathrm{d}\left(\mathbf{J}^{(0)}_{f_n \to i}, \mathbf{J}^*_{f_n \to i} \right),$$

where $\sum_{f_n \in \mathcal{B}(i)} \mathrm{d}\left(\mathbf{J}^{(0)}_{f_n \to i}, \mathbf{J}^*_{f_n \to i} \right)$ is a constant. Following the same procedure as that from (36) to (38), we can prove that $\mathbf{P}_i^{(\ell)}$ converges at an exponential rate with respect to the monotone norm before $\mathbf{P}_i^{(\ell)}$ enters $\mathbf{P}_i^*$'s neighborhood, which can be chosen to be arbitrarily small. ■

On the other hand, as shown in (24), the computation of the belief mean $\boldsymbol{\mu}_i^{(\ell)}$ depends on the belief covariance $\mathbf{P}_i^{(\ell)}$ and the message mean $\mathbf{v}^{(\ell)}_{f_n \to i}$. Thus, under the same condition as in Theorem 13, $\boldsymbol{\mu}_i^{(\ell)}$ is convergence guaranteed. Moreover, it is shown in (Weiss and Freeman, 2001b, Appendix) that, for Gaussian BP over a factor graph, the converged value of belief mean equals the optimal estimate in (3). Together with the convergence guaranteed topology revealed in Theorem 14, we have the following Corollary.

Corollary 17 *With arbitrary* $\boldsymbol{J}^{(0)}_{f_n \to i} \succeq \boldsymbol{0}$ *and arbitrary* $\boldsymbol{v}^{(0)}_{f_n \to i}$ *for all* $i \in \mathcal{V}$ *and* $f_n \in \mathcal{B}(i)$*, the mean vector* $\boldsymbol{\mu}_i^{(\ell)}$ *in (24) converges to the optimal estimate* $\widehat{\boldsymbol{x}}_i$ *in (3) if and only if* $\rho(\boldsymbol{Q}) < 1$*, where* $\boldsymbol{Q}$ *is defined in (43). Furthermore, a sufficient condition to guarantee* $\rho(\boldsymbol{Q}) < 1$ *is when the factor graph contains only one single loop connected to multiple chains/trees.*

5. Relationships with Existing Convergence Conditions

In this section, we show the relationship between our convergence condition for Gaussian BP and the recent proposed path-sum method (Giscard et al., 2016). We also show that our convergence condition is more general than the walk-summable condition (Malioutov et al., 2006) for the scalar case.

5.1 Relationship with the Path-sum Method

The path-sum method is proposed in (Giscard et al., 2012, 2013, 2016) to compute $\left(\mathbf{W}^{-1}+\mathbf{A}^T\mathbf{R}^{-1}\mathbf{A}\right)^{-1}$ in (3), in which the matrix inverse $\left(\mathbf{W}^{-1}+\mathbf{A}^T\mathbf{R}^{-1}\mathbf{A}\right)^{-1}$ is interpreted as the sum of simple paths and simple cycles on a weighted graph. The resulting formulation is guaranteed to converge to the correct value for any valid multivariate Gaussian distribution.

The BP message update equations (16), (17), (19), and (20) can be seen as a cut-off of path-sum by retaining only self-loops and backtracks (simple cycles of lengths one and two). In the presence of a graph with one or more loops, equations (16), (17), (19), and (20) do not include the terms related to simple cycles with length larger than 2. This may be a potential cause for the possible divergence of the Gaussian BP algorithm. From this perspective, the divergence can be averted if none of the walks going around the loop(s) have weight greater than one, or equivalently, that the spectral radius of the block matrix representing the loop(s) is strictly less than one. This is an intuitive explanation of the condition $\rho(\mathbf{Q})<1$ obtained in Theorem 13. It also immediately follows from these considerations that the convergence rate is at least geometric, with a cut-off of order ℓ yielding an $\mathcal{O}(\rho(\mathbf{Q})^{\ell})$ error[4].

While the path-sum framework provides an insightful interpretation of the results obtained in this paper, the path-sum algorithm may not be efficiently implementable in distributed and parallel settings, as it requires the summation over all the paths of any length. In contrast, Gaussian BP, while paying the price of non-convergence in general loopy models, makes it possible to realize parallel and fully distributed inference. In summary, though the path-sum method converges for arbitrary valid Gaussian models, it is difficult to be adapted to a distributed and parallel inference setup as the Gaussian BP method.

5.2 Relationship with the Walk-Summable Condition

We show next that, in the setup of linear Gaussian models, the condition $\rho(\mathbf{Q})<1$ as in Corollary 17 encompasses the Gaussian MRF based walk-summable (Malioutov et al. (2006)) in terms of convergence. As all existing results on Gaussian BP convergence (Malioutov et al., 2006; Moallemi and Roy, 2009b) only apply to scalar variables, we restrict the following discussion to only the scalar case. In (Malioutov et al., 2006), the starting point for the convergence analysis for Gaussian MRF is a joint multivariate Gaussian distribution

$$q(\mathbf{x})\propto\exp\Big\{-\frac{1}{2}\mathbf{x}^T\mathbf{J}\mathbf{x}+\mathbf{h}^T\mathbf{x}\Big\},\tag{56}$$

4. We thank an anonymous reviewer for this interpretation.

expressed in the normalized information form such that $\mathbf{J}_{i,i}=1$ for all i. The underlying Gaussian distribution is factorized as (Malioutov et al. (2006))

$$q(\mathbf{x})\propto\prod_{i\in\mathcal{V}}\psi_i(\mathbf{x}_i)\prod_{J_{i,j}\neq 0;i\leq j}\psi_{i,j}(\mathbf{x}_i,\mathbf{x}_j),\tag{57}$$

where

$$\psi_i(\mathbf{x}_i)=\exp\left\{-\frac{1}{2}\mathbf{J}_{i,i}\mathbf{x}_i^2+\mathbf{h}_i\right\}\quad\text{and}\quad\psi_{i,j}(\mathbf{x}_i,\mathbf{x}_j)=\exp\left\{-\mathbf{x}_i\mathbf{J}_{i,j}\mathbf{x}_j\right\}.$$

In Malioutov et al. (2006, Proposition 1), based on the interpretation that $\left[\mathbf{J}^{-1}\right]_{i,j}$ is the sum of the weights of all the walks from variable j to variable i on the corresponding Gaussian MRF, a sufficient Gaussian BP convergence condition known as walk-summability is provided, which is equivalent to

$$\mathbf{I}-|\mathbf{R}|\succ\mathbf{0},\tag{58}$$

together with the initial message variance inverse being set to 0, where $\mathbf{R}=\mathbf{I}-\mathbf{J}$ and $|\mathbf{R}|$ is matrix of entrywise absolute values of $\mathbf{R}$. In the following, we establish the relationship between walk-summable Gaussian MRF and linear Gaussian model by utilizing properties of H-matrices (Boman et al., 2005). We show that, with Gaussian MRF satisfying the walk-summable condition, the joint distribution $q(\mathbf{x})$ in (57) can be reformulated as a special case of the linear Gaussian model based factorization in (10). Moreover, Gaussian BP on this particular linear Gaussian model always converges.

Definition 18 *H-Matrices (Boman et al., 2005): A matrix $\boldsymbol{X}$ is an H-matrix if all eigenvalues of the matrix $\mathcal{H}(\boldsymbol{X})$, where $[\mathcal{H}(\boldsymbol{X})]_{i,i}=|\boldsymbol{X}_{i,i}|$, and $[\mathcal{H}(\boldsymbol{X})]_{i,j}=-|\boldsymbol{X}_{i,j}|$ have positive real parts.*

Proposition 19 *Factor width at most 2 factorization (Boman et al., 2005, Theorem 9): A symmetric H-matrix $\boldsymbol{X}$ that has non-negative diagonals can always be factorized as $\boldsymbol{X}=\boldsymbol{V}\boldsymbol{V}^T$, where $\boldsymbol{V}$ is a real matrix with each column of $\boldsymbol{V}$ containing at most 2 non-zeros.*

Let ω be an arbitrary positive value that is smaller than the minimum eigenvalue of $\mathbf{I}-|\mathbf{R}|$ and also satisfies $0<\omega<1$. According to (58), we have $(1-\omega)\mathbf{I}-|\mathbf{R}|\succ\mathbf{0}$. Furthermore, by applying $\mathcal{H}(\cdot)$ to $(1-\omega)\mathbf{I}-\mathbf{R}$, we have $[\mathcal{H}((1-\omega)\mathbf{I}-\mathbf{R})]_{i,i}=|(1-\omega)\mathbf{I}-\mathbf{R}|_{i,i}=1-\omega$ and $[\mathcal{H}((1-\omega)\mathbf{I}-\mathbf{R})]_{i,j}=-|[(1-\omega)\mathbf{I}-\mathbf{R})]_{i,j}|=-|\mathbf{R}_{i,j}|$. Thus, $\mathcal{H}((1-\omega)\mathbf{I}-\mathbf{R})=(1-\omega)\mathbf{I}-|\mathbf{R}|\succ\mathbf{0}$, and we conclude that $(1-\omega)\mathbf{I}-\mathbf{R}$ is an H-matrix. According to Proposition 19, $(1-\omega)\mathbf{I}-\mathbf{R}=\mathbf{J}-\omega\mathbf{I}=\mathbf{V}\mathbf{V}^T$, where each column of $\mathbf{V}$ contains at most 2 non-zeros. Now, we can rewrite the joint distribution in (57) as

$$\begin{aligned}q(\mathbf{x})&\propto\exp\left\{-\frac{1}{2}\mathbf{x}^T\left(\mathbf{J}-\omega\mathbf{I}\right)\mathbf{x}-\frac{1}{2}\omega\mathbf{x}^T\mathbf{x}+\mathbf{h}^T\mathbf{x}\right\}\\&=\exp\left\{-\frac{1}{2}\left(\mathbf{V}^T\mathbf{x}\right)^T\left(\mathbf{V}^T\mathbf{x}\right)-\frac{1}{2}\left(\omega\mathbf{x}^T\mathbf{x}-2\mathbf{h}^T\mathbf{x}\right)\right\}\\&\propto\exp\left\{-\frac{1}{2}\sum_{n=1}^{M}\left(V_{n,n_i}x_{n_i}+V_{n,n_j}x_{n_j}\right)^2-\frac{1}{2}\sum_{n=1}^{M}\omega(x_n-\frac{h_n}{\omega})^2\right\},\end{aligned}\tag{59}$$

where V_{n,n_i} and V_{n,n_j} denote the two possible non-zero elements on the n-th column and n_i-th and n_j-th rows, and M is the dimension of $\mathbf{x}$. Thus, a walk-summable Gaussian MRF in (56) (or equivalently (57)) can always be written as

$$q(\mathbf{x}) \propto \prod_{n=1}^{M} \underbrace{\mathcal{N}(x_n|\frac{1}{\omega}h_n, \frac{1}{\omega})}_{p(x_n)} \prod_{n=1}^{M} \underbrace{\mathcal{N}(0|V_{n,n_i}x_{n_i} + V_{n,n_j}x_{n_j}, 1)}_{f_n}. \tag{60}$$

Note that, in the above equation, $p(x_n)$ serves as the prior distribution for x_n as that in (10) and f_n is the local likelihood function with local observation being $y_n = 0$ and noise distribution $z_n \sim \mathcal{N}(z_n|0, 1)$[5]. Thus the above equation is a special case of the linear Gaussian model based factorization in (10) with the local likelihood function f_n containing only a pair of variables. For this pairwise linear Gaussian model with scalar variables, it is shown in (Moallemi and Roy, 2009b) that $\rho(\mathbf{Q}) < 1$ is fulfilled. Thus by Corollary 17, Gaussian BP always converges. In summary, for the factorization based on Gaussian MRF, if the walk-summable convergence condition is fulfilled, there is an equivalent joint distribution factorization based on linear Gaussian model; and Gaussian BP is convergence guaranteed for this linear Gaussian model.

In the following, we further demonstrate through an example that there exist Gaussian MRFs, in which the information matrix $\mathbf{J}$ fails to satisfy the walk-summable condition, but a convergence guaranteed Gaussian BP update based on the distributed linear Gaussian model representation can still be obtained. More specifically, consider the following information matrix $\mathbf{J}$ in a Gaussian MRF:

$$\mathbf{J} = \begin{bmatrix} 1 & \frac{1}{3\sqrt{2}} & \frac{1}{\sqrt{3}} & \frac{\sqrt{2}}{3} \\ \frac{1}{3\sqrt{2}} & 1 & 0 & \frac{1}{3} \\ \frac{1}{\sqrt{3}} & 0 & 1 & \frac{1}{\sqrt{6}} \\ \frac{\sqrt{2}}{3} & \frac{1}{3} & \frac{1}{\sqrt{6}} & 1 \end{bmatrix}. \tag{61}$$

The eigenvalues of $\mathbf{I} - |\mathbf{R}|$ to 4 decimal places are -0.0754, 0.9712, 1.4780, and 1.6262. According to the walk-summable definition in (58), it is non walk-summable and the convergence condition in (Malioutov et al., 2006) is inconclusive as to whether Gaussian BP converges. On the other hand, we can study the Gaussian BP convergence of this example by employing a linear Gaussian model representation, and rewriting $\mathbf{J}$ as $\mathbf{J} = \mathbf{A}^T\mathbf{R}^{-1}\mathbf{A} + \mathbf{W}^{-1}$, where

$$\mathbf{A} = \begin{bmatrix} \frac{2}{\sqrt{6}} & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{3}} & 0 & \frac{1}{\sqrt{3}} \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} 6 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix},$$

and $\mathbf{R} = \mathbf{I}$. In Fig. 3, the joint distribution of this example with Gaussian MRF and the corresponding linear Gaussian model are represented by factor graphs. As it is shown in Corollary 17, for a factor graph that is the union of a forest and a single loop, as in Fig. 3(b), Gaussian BP always converges to the exact value. This is in sharp contrast to

5. For a particular f_n, if there is only one non-zero coefficient, $f_n \times \mathcal{N}(x_n|\frac{1}{\omega}h_n, \frac{1}{\omega})$ is also proportional to a Gaussian distribution, which can be seen as a prior distribution in (10).

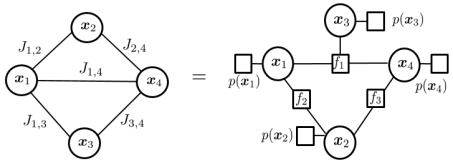


Figure 3: The Gaussian MRF corresponding to $\mathbf{J}$ in (61) with the factorization following (4); (b) The factor graph corresponding to $\mathbf{J}$ in (61) with the factorization following (10).

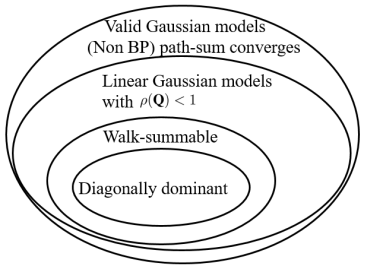


Figure 4: Venn diagram summarizing various subclasses of Gaussian models: the three inner most conditions are all for the BP algorithm while the path-sum in general does not constitute a BP algorithm.

the inconclusive convergence property when the same joint distribution is expressed using the classical Gaussian MRF in (4).

In summary, we have shown that the linear Gaussian model with $\rho(\mathbf{Q}) < 1$ encompasses walk-summable Gaussian MRF. Further, it is shown in (Malioutov et al., 2006) that the diagonally dominant convergence condition in (Weiss and Freeman, 2001a) for Gaussian BP is a special case of the walk-summable condition. Also, the convergence condition in (Su and Wu, 2015) is encompassed by walk-summable condition. Therefore, we have the Venn diagram in Fig. 4 summarizing the relations (in terms of convergence guarantees) between the convergence condition proposed in this paper and existing conditions.

6. Conclusions

This paper shows that, depending on how the factorization of the underlying joint Gaussian distribution is performed, Gaussian belief propagation (BP) may exhibit different convergence properties as different factorizations lead to fundamentally different recursive update structures. The paper studies the convergence of Gaussian BP derived from the factorization

based on the distributed linear Gaussian model. We show that the condition we present for convergence of the marginal mean based on factorizations using the linear Gaussian model is more general than the walk-summable condition (Malioutov et al., 2006) (and references therein) that is based on the Gaussian Markov random field factorization. Further, the linear Gaussian model that is studied in this paper readily conforms to the physical network topology arising in large-scale networks.

Further, the paper analyzes the convergence of the Gaussian BP based distributed inference algorithm. In particular, we show analytically that, with arbitrary positive semidefinite matrix initialization, the message information matrix exchanged among agents converges to a unique positive definite matrix, and it approaches an arbitrarily small neighborhood of this unique positive definite matrix at an exponential rate (with respect to any matrix norm). Regarding the initial information matrix, there exist positive definite initializations that guarantee faster convergence than the commonly used all-zero matrix. Moreover, under the positive semidefinite initial message information matrix condition, we present a necessary and sufficient condition of the belief mean vector to converge to the optimal centralized estimate. We also prove that Gaussian BP always converges if the corresponding factor graph is a union of a single loop and a forest. In particular, we show that the proposed convergence condition for Gaussian BP based on the linear Gaussian model leads to a strictly larger class of models in which Gaussian BP converges than those postulated by the Gaussian Markov random field based walk-summable condition. Finally, we discuss connections of Gaussian BP with the general path-sum algorithm. In the future, it will be interesting to explore if these path-sum interpretations can lead to modifications of the standard Gaussian BP algorithm that guarantee the convergence of Gaussian BP for larger classes of topologies while being also parallel and fully distributed.

Acknowledgments

We thank the reviewers for giving detailed comments and suggestions that have been helpful in improving this paper.

We would like to acknowledge support for this project from the National Science Foundation (NSF grant # CCF1513936), National Natural Science Foundation of China (NSFC grant 61601524), Macau Science and Technology Development Fund under grant FDCT 091/2015/A3, Research Committee of University of Macau under Grant MYRG2014-00146-FST and Grant MYRG2016-00146-FST, and the General Research Fund (GRF) from Hong Kong Research Grant Council (Project No.: 17212416).

Appendix A.

We first compute the first round updating message from variable node to factor node. Substituting $m^{(0)}_{f_n\to i}(\mathbf{x}_i) \propto \exp\left\{-\frac{1}{2}||\mathbf{x}_j - \mathbf{v}^{(0)}_{f_n\to i}||^2_{\mathbf{J}^{(0)}_{f_n\to i}}\right\}$ into (11) and, after algebraic manipulations, we obtain

$$m^{(1)}_{j\to f_n}(\mathbf{x}_j) \propto \exp\left\{-\frac{1}{2}||\mathbf{x}_j - \mathbf{v}^{(1)}_{j\to f_n}||^2_{\mathbf{J}^{(1)}_{j\to f_n}}\right\}, \tag{62}$$

with

$$\mathbf{J}^{(1)}_{j\to f_n} = \mathbf{W}_j^{-1} + \sum_{f_k\in\mathcal{B}(j)\setminus f_n} \mathbf{J}^{(0)}_{f_k\to j},$$

and

$$\mathbf{v}^{(1)}_{j\to f_n} = \left[\mathbf{J}^{(1)}_{j\to f_n}\right]^{-1}\left[\sum_{f_k\in\mathcal{B}(j)\setminus f_n} \mathbf{J}^{(0)}_{f_k\to j}\mathbf{v}^{(0)}_{f_k\to j}\right].$$

Next, we evaluate $m^{(1)}_{f_n\to i}(\mathbf{x}_i)$. By substituting $m^{(1)}_{j\to f_n}(\mathbf{x}_j)$ in (62) into (12), we obtain

$$\begin{aligned} m^{(1)}_{f_n\to i}(\mathbf{x}_i) \propto \int\ldots\int \exp\Big\{-\frac{1}{2}\big(\mathbf{y}_n - \sum_{j\in\mathcal{B}(f_n)}\mathbf{A}_{n,j}\mathbf{x}_j\big)^T\mathbf{R}^{-1}\big(\mathbf{y}_n - \sum_{j\in\mathcal{B}(f_n)}\mathbf{A}_{n,j}\mathbf{x}_j\big)\Big\}\times \\ \prod_{j\in\mathcal{B}(f_n)\setminus i}\exp\Big\{-\frac{1}{2}||\mathbf{x}_j - \mathbf{v}^{(1)}_{j\to f_n}||^2_{\mathbf{J}^{(1)}_{j\to f_n}}\Big\}\,\mathrm{d}\{\mathbf{x}_j\}_{j\in\mathcal{B}(f_n)\setminus i}. \end{aligned} \tag{63}$$

Let $\mathbf{x}_{\{\mathcal{B}(f_n)\setminus i\}}$ and $\mathbf{v}^{(1)}_{\{\mathcal{B}(f_n)\setminus i\}\to f_n}$ be stacked vectors containing $\mathbf{x}_j$ and $\mathbf{v}^{(1)}_{j\to f_n}$ as vector elements for all $j\in\mathcal{B}(f_n)\setminus i$ arranged in ascending order on j, respectively; $\mathbf{A}_{n,\{\mathcal{B}(f_n)\setminus i\}}$ denotes a row block matrix containing $\mathbf{A}_{n,j}$ as row elements for all $j\in\mathcal{B}(f_n)\setminus i$ arranged in ascending order; and $\mathbf{J}^{(1)}_{\{\mathcal{B}(f_n)\setminus i\}\to f_n}$ is a block diagonal matrix with $\mathbf{J}^{(1)}_{j\to f_n}$ as its block diagonal elements for all $j\in\mathcal{B}(f_n)\setminus i$ arranged in ascending order. Then, (63) can be reformulated as

$$\begin{aligned} m^{(1)}_{f_n\to i}(\mathbf{x}_i) \propto & \int\ldots\int \exp\left\{-\frac{1}{2}||\mathbf{y}_n - \mathbf{A}_{n,i}\mathbf{x}_i - \mathbf{A}_{n,\{\mathcal{B}(f_n)\setminus i\}}\mathbf{x}^T_{\{\mathcal{B}(f_n)\setminus i\}}||^2_{\mathbf{R}^{-1}}\right\} \\ & \times\exp\left\{-\frac{1}{2}||\mathbf{x}_{\{\mathcal{B}(f_n)\setminus i\}} - \mathbf{v}^{(1)}_{\{\mathcal{B}(f_n)\setminus i\}\to f_n})||^2_{\mathbf{J}^{(1)}_{\{\mathcal{B}(f_n)\setminus i\}\to f_n}}\right\}\mathrm{d}\mathbf{x}_{\{\mathcal{B}(f_n)\setminus i\}} \\ \propto & \exp\left\{-\frac{1}{2}||\mathbf{y}_n - \mathbf{A}_{n,i}\mathbf{x}_i||^2_{\mathbf{R}^{-1}}\right\} \\ & \times\int\ldots\int\exp\left\{\frac{1}{2}(\mathbf{x}^T_{\{\mathcal{B}(f_n)\setminus i\}}\mathbf{K}^{(1)}_{f_n\to i}\mathbf{x}_{\{\mathcal{B}(f_n)\setminus i\}} - 2[\mathbf{h}^{(1)}_{f_n\to i}]^T\mathbf{x}_{\{\mathcal{B}(f_n)\setminus i\}})\right\}\mathrm{d}\mathbf{x}_{\{\mathcal{B}(f_n)\setminus i\}}, \end{aligned} \tag{64}$$

where

$$\mathbf{K}^{(1)}_{f_n\to i} = \mathbf{A}^T_{n,\{\mathcal{B}(f_n)\setminus i\}}\mathbf{R}^{-1}\mathbf{A}_{n,\{\mathcal{B}(f_n)\setminus i\}} + \mathbf{J}^{(1)}_{\{\mathcal{B}(f_n)\setminus i\}\to f_n}$$

and

$$\mathbf{h}^{(1)}_{f_n\to i} = \mathbf{A}^T_{n,\{\mathcal{B}(f_n)\setminus i\}}\mathbf{R}^{-1}(\mathbf{y}_n - \mathbf{A}_{n,i}\mathbf{x}_i) + \mathbf{J}^{(1)}_{\{\mathcal{B}(f_n)\setminus i\}\to f_n}\mathbf{v}^{(1)}_{\{\mathcal{B}(f_n)\setminus i\}\to f_n}.$$

By completing the square for the integrand of (64), we obtain

$$m^{(1)}_{f_n\to i}(\mathbf{x}_i) \propto \alpha^{(1)}_{f_n\to i}\exp\left\{-\frac{1}{2}||\mathbf{y}_n-\mathbf{A}_{n,i}\mathbf{x}_i||^2_{\mathbf{R}_n^{-1}}+\frac{1}{2}\left[\mathbf{h}^{(1)}_{f_n\to i}\right]^T\left[\mathbf{K}^{(1)}_{f_n\to i}\right]^{-1}\mathbf{h}^{(\ell)}_{f_n\to i}\right\},(65)$$

with

$$\alpha^{(1)}_{f_n\to i}=\int\cdots\int\exp\left\{-\frac{1}{2}||\mathbf{x}_{\{\mathcal{B}(f_n)\backslash i\}}-\left[\mathbf{K}^{(1)}_{f_n\to i}\right]^{-1}\mathbf{h}^{(1)}_{f_n\to i}||^2_{\mathbf{K}^{(1)}_{f_n\to i}}\right\}\mathrm{d}\mathbf{x}_{\{\mathcal{B}(f_n)\backslash i\}}.$$

Next, by applying the spectral theorem to $\mathbf{K}^{(1)}_{f_n\to i}$ and after some algebraic manipulations, we simplify (65) as

$$m^{(1)}_{f_n\to i}(\mathbf{x}_i)\propto\alpha^{(1)}_{f_n\to i}\exp\left\{-\frac{1}{2}||\mathbf{x}_i-\mathbf{v}^{(1)}_{f_n\to i}||_{\mathbf{J}^{(1)}_{f_n\to i}}\right\},$$

with the inverse of the covariance, the information matrix

$$\mathbf{J}^{(1)}_{f_n\to i}=\mathbf{A}^T_{n,i}\left[\mathbf{R}_n+\sum_{j\in\mathcal{B}(f_n)\backslash i}\mathbf{A}_{n,j}\left[\mathbf{J}^{(1)}_{j\to f_n}\right]^{-1}\mathbf{A}^T_{n,j}\right]^{-1}\mathbf{A}_{n,i},$$

and the mean vector

$$\mathbf{v}^{(1)}_{f_n\to i}=\left[\mathbf{J}^{(1)}_{f_n\to i}\right]^{-1}\mathbf{A}^H_{n,i}\left[\mathbf{R}_n+\sum_{j\in\mathcal{B}(f_n)\backslash i}\mathbf{A}_{n,j}\left[\mathbf{J}^{(1)}_{j\to f_n}\right]^{-1}\mathbf{A}^T_{n,j}\right]^{-1}\left(\mathbf{y}_n-\sum_{j\in\mathcal{B}(f_n)\backslash i}\mathbf{A}_{n,j}\mathbf{v}^{(1)}_{j\to f_n}\right),$$

and

$$\alpha^{(1)}_{f_n\to i}\propto\int\cdots\int\exp\left\{-\frac{1}{2}\mathbf{z}^T\mathbf{\Lambda}^{(1)}_{f_n\to i}\mathbf{z}\right\}\mathrm{d}\mathbf{z},\quad(66)$$

where $\mathbf{\Lambda}^{(1)}_{f_n\to i}$ is a diagonal matrix containing the eigenvalues of $\mathbf{A}^T_{n,\{\mathcal{B}(f_n)\backslash i\}}\mathbf{R}_n^{-1}\mathbf{A}_{n,\{\mathcal{B}(f_n)\backslash i\}}+\mathbf{J}^{(1)}_{\{\mathcal{B}(f_n)\backslash i\}\to f_n}$.

By induction, and following similar derivations as in (62) to (66), we obtain the general updating expressions as in (15) to (21).

Appendix B.

Before going into the proof of Lemma 2, we note the following properties of positive definite (p.d.) matrices. If $\mathbf{X}\succ\mathbf{0}$, $\mathbf{Y}\succ\mathbf{0}$, $\mathbf{Z}\succeq\mathbf{0}$ are of the same dimension, then we have (Du and Wu, 2013a):

P B.1: $\mathbf{X}+\mathbf{Y}\succ\mathbf{0}$ and $\mathbf{X}+\mathbf{Z}\succ\mathbf{0}$.

P B.2: $\mathbf{A}^T\mathbf{X}\mathbf{A}\succ\mathbf{0}$, $\mathbf{A}^T\mathbf{Z}\mathbf{A}\succeq\mathbf{0}$, $\mathbf{A}\mathbf{X}\mathbf{A}^T\succeq\mathbf{0}$ and $\mathbf{A}\mathbf{Z}\mathbf{A}^T\succeq\mathbf{0}$ for any full column rank matrix $\mathbf{A}$ with compatible dimension.

Now, we prove Lemma 2. If $\mathbf{J}^{(\ell-1)}_{f_k\to j}\succeq\mathbf{0}$ for all $f_k\in\mathcal{B}(j)\setminus f_n$, according to P B.1, $\sum_{f_k\in\mathcal{B}(j)\backslash f_n}\mathbf{J}^{(\ell-1)}_{f_k\to j}\succeq\mathbf{0}$. As $\mathbf{W}_j^{-1}\succ\mathbf{0}$, we have $\mathbf{W}_j^{-1}+\sum_{f_k\in\mathcal{B}(j)\backslash f_n}\mathbf{J}^{(\ell-1)}_{f_k\to j}\succ\mathbf{0}$, which, according to (16), is equivalent to $\mathbf{J}^{(\ell)}_{j\to f_n}\succ\mathbf{0}$. Besides, as $\mathbf{A}_{n,j}$ is full column rank, if $\left[\mathbf{J}^{(\ell)}_{j\to f_n}\right]^{-1}\succ\mathbf{0}$

for all $j\in\mathcal{B}(f_n)\backslash i$, according to P B.2, $\mathbf{A}_{n,j}\left[\mathbf{J}^{(\ell)}_{j\to f_n}\right]^{-1}\mathbf{A}^T_{n,j}\succeq\mathbf{0}$. With $\mathbf{R}_n\succ\mathbf{0}$, following P B.1, we have $\left[\mathbf{R}_n+\sum_{j\in\mathcal{B}(f_n)\backslash i}\mathbf{A}_{n,j}\left[\mathbf{J}^{(\ell)}_{j\to f_n}\right]^{-1}\mathbf{A}^T_{n,j}\right]^{-1}\succ\mathbf{0}$. As $\mathbf{A}_{n,i}$ is of full column rank, by applying P B.2 again, we have $\mathbf{A}^T_{n,i}\left[\mathbf{R}_n+\sum_{j\in\mathcal{B}(f_n)\backslash i}\mathbf{A}_{n,j}\left[\mathbf{J}^{(\ell)}_{j\to f_n}\right]^{-1}\mathbf{A}^T_{n,j}\right]^{-1}\mathbf{A}_{n,i}\succ\mathbf{0}$, which according to (19) is equivalent to $\mathbf{J}^{(\ell)}_{f_n\to i}\succ\mathbf{0}$.

In summary, we have proved that 1) if $\mathbf{J}^{(\ell-1)}_{f_k\to j}\succeq\mathbf{0}$ for all $f_k\in\mathcal{B}(j)\setminus f_n$, then $\mathbf{J}^{(\ell)}_{j\to f_n}\succ\mathbf{0}$; 2) if $\left[\mathbf{J}^{(\ell)}_{j\to f_n}\right]^{-1}\succ\mathbf{0}$ for all $j\in\mathcal{B}(f_n)\setminus i$, then $\mathbf{J}^{(\ell)}_{f_n\to i}\succ\mathbf{0}$. Therefore, by setting $\mathbf{J}^{(0)}_{f_k\to j}\succeq\mathbf{0}$ for all $k\in\mathcal{V}$ and $j\in\mathcal{B}(f_k)$, according to the results of the first case, we have $\mathbf{J}^{(1)}_{j\to f_n}\succ\mathbf{0}$ for all $j\in\mathcal{V}$ and $f_n\in\mathcal{B}(j)$. Then, applying the second case, we further have $\mathbf{J}^{(1)}_{f_n\to i}\succ\mathbf{0}$ for all $n\in\mathcal{V}$ and $i\in\mathcal{B}(f_n)$. By repeatedly using the above arguments, it follows readily that $\mathbf{J}^{(\ell)}_{f_k\to j}\succ\mathbf{0}$ and $\mathbf{J}^{(\ell)}_{j\to f_n}\succ\mathbf{0}$ for $\ell\geq1$ and with $j\in\mathcal{V}$, $f_n,f_k\in\mathcal{B}(j)$. Furthermore, according to the discussion before Lemma 2, all messages $m^{(\ell)}_{j\to f_n}(\mathbf{x}_j)$ and $m^{(\ell)}_{f_n\to i}(\mathbf{x}_i)$ exist, and are in Gaussian form as in (15) and (18).

Appendix C.

First, Proposition 4, P 4.1 is proved. Suppose that $\mathbf{J}^{(\ell)}\succeq\mathbf{J}^{(\ell-1)}\succeq\mathbf{0}$, i.e., $\mathbf{J}^{(\ell)}_{f_k\to j}\succeq\mathbf{J}^{(\ell-1)}_{f_k\to j}\succeq\mathbf{0}$ for all $(f_k,j)\in\widetilde{\mathcal{B}}(f_n,i)$, we have

$$\mathbf{W}_j^{-1}+\sum_{f_k\in\mathcal{B}(j)\backslash f_n}\mathbf{J}^{(\ell)}_{f_k\to j}\succeq\mathbf{W}_j^{-1}+\sum_{f_k\in\mathcal{B}(j)\backslash f_n}\mathbf{J}^{(\ell-1)}_{f_k\to j}\succ\mathbf{0}.$$

Then, according to the fact that if $\mathbf{X}\succeq\mathbf{Y}\succ\mathbf{0}$, $\mathbf{Y}^{-1}\succeq\mathbf{X}^{-1}\succ\mathbf{0}$, we have

$$\left[\mathbf{W}_j^{-1}+\sum_{f_k\in\mathcal{B}(j)\backslash f_n}\mathbf{J}^{(\ell-1)}_{f_k\to j}\right]^{-1}\succeq\left[\mathbf{W}_j^{-1}+\sum_{f_k\in\mathcal{B}(j)\backslash f_n}\mathbf{J}^{(\ell)}_{f_k\to j}\right]^{-1}\succ\mathbf{0}.$$

Since $\mathbf{A}_{n,j}$ is of full column rank and following P B.2 in Appendix B, we have

$$\mathbf{A}_{n,j}\left[\mathbf{W}_j^{-1}+\sum_{f_k\in\mathcal{B}(j)\backslash f_n}\mathbf{J}^{(\ell-1)}_{f_k\to j}\right]^{-1}\mathbf{A}^T_{n,j}\succeq\mathbf{A}_{n,j}\left[\mathbf{W}_j^{-1}+\sum_{f_k\in\mathcal{B}(j)\backslash f_n}\mathbf{J}^{(\ell)}_{f_k\to j}\right]^{-1}\mathbf{A}^T_{n,j}\succ\mathbf{0}.$$

Following the same procedure of the proof above and due to $\mathbf{R}\succ\mathbf{0}$, we can further prove that

$$\begin{aligned}&\mathbf{A}^T_{n,i}\left[\mathbf{R}_n+\sum_{j\in\mathcal{B}(f_n)\backslash i}\mathbf{A}_{n,j}\left[\mathbf{W}_j^{-1}+\sum_{f_k\in\mathcal{B}(j)\backslash f_n}\mathbf{J}^{(\ell)}_{f_k\to j}\right]^{-1}\mathbf{A}^T_{n,j}\right]^{-1}\mathbf{A}_{n,i}\\ \succeq&\mathbf{A}^T_{n,i}\left[\mathbf{R}_n+\sum_{j\in\mathcal{B}(f_n)\backslash i}\mathbf{A}_{n,j}\left[\mathbf{W}_j^{-1}+\sum_{f_k\in\mathcal{B}(j)\backslash f_n}\mathbf{J}^{(\ell-1)}_{f_k\to j}\right]^{-1}\mathbf{A}^T_{n,j}\right]^{-1}\mathbf{A}_{n,i},\end{aligned}$$

which is equivalent to

$$\mathcal{F}_{n\to i}\left(\left\{\mathbf{J}_{f_k\to j}^{(\ell)}\right\}_{(f_k,j)\in\widetilde{\mathcal{B}}(f_n,i)}\right)\succeq\mathcal{F}_{n\to i}\left(\left\{\mathbf{J}_{f_k\to j}^{(\ell-1)}\right\}_{(f_k,j)\in\widetilde{\mathcal{B}}(f_n,i)}\right).$$

Since $\mathcal{F}$ contains $\mathcal{F}_{n\to i}(\cdot)$ as its component, Proposition 4, P 4.1 is proved.

Next, Proposition 4, P 4.2 is proved. Suppose that $\mathbf{J}_{f_k\to j}^{(\ell)}\succ\mathbf{0}$ for all $(f_k,j)\in\widetilde{\mathcal{B}}(f_n,i)$. As $\alpha>1$, we have

$$\alpha\mathbf{W}_j^{-1}+\sum_{f_k\in\mathcal{B}(j)\backslash f_n}\alpha\mathbf{J}_{f_k\to j}^{(\ell)}\succeq\mathbf{W}_j^{-1}+\sum_{f_k\in\mathcal{B}(j)\backslash f_n}\alpha\mathbf{J}_{f_k\to j}^{(\ell)}\succ\mathbf{0},$$

where the equality holds when $\mathbf{W}_j^{-1}=\mathbf{0}$, which corresponds to non-informative prior for $\mathbf{x}_j$. Applying the fact that if $\mathbf{X}\succeq\mathbf{Y}\succ\mathbf{0}$, $\mathbf{Y}^{-1}\succeq\mathbf{X}^{-1}\succ\mathbf{0}$, and, according to P B.2 in Appendix B, we obtain

$$\mathbf{A}_{n,j}\left[\mathbf{W}_j^{-1}+\sum_{f_k\in\mathcal{B}(j)\backslash f_n}\alpha\mathbf{J}_{f_k\to j}^{(\ell)}\right]^{-1}\mathbf{A}_{n,j}^T\succeq\mathbf{A}_{n,j}\left[\alpha\mathbf{W}_j^{-1}+\sum_{f_k\in\mathcal{B}(j)\backslash f_n}\alpha\mathbf{J}_{f_k\to j}^{(\ell)}\right]^{-1}\mathbf{A}_{n,j}^T\succeq\mathbf{0}.$$

Since $\mathbf{R}_n\succ\frac{1}{\alpha}\mathbf{R}_n\succ\mathbf{0}$, we have

$$\begin{aligned}&\left[\frac{1}{\alpha}\mathbf{R}_n+\sum_{j\in\mathcal{B}(f_n)\backslash i}\mathbf{A}_{n,j}\left[\alpha\mathbf{W}_j^{-1}+\sum_{f_k\in\mathcal{B}(j)\backslash f_n}\alpha\mathbf{J}_{f_k\to j}^{(\ell)}\right]^{-1}\mathbf{A}_{n,j}^T\right]^{-1}\\&\succ\left[\mathbf{R}_n+\sum_{j\in\mathcal{B}(f_n)\backslash i}\mathbf{A}_{n,j}\left[\mathbf{W}_j^{-1}+\sum_{f_k\in\mathcal{B}(j)\backslash f_n}\alpha\mathbf{J}_{f_k\to j}^{(\ell)}\right]^{-1}\mathbf{A}_{n,j}^T\right]^{-1}.\end{aligned}$$

Finally, applying P B.2 in Appendix B to the above equation and taking out the common factor α, we obtain

$$\begin{aligned}&\alpha\mathbf{A}_{n,i}^T\left[\mathbf{R}_n+\sum_{j\in\mathcal{B}(f_n)\backslash i}\mathbf{A}_{n,j}\left[\mathbf{W}_j^{-1}+\sum_{f_k\in\mathcal{B}(j)\backslash f_n}\mathbf{J}_{f_k\to j}^{(\ell)}\right]^{-1}\mathbf{A}_{n,j}^T\right]^{-1}\mathbf{A}_{n,i}\\&\succ\mathbf{A}_{n,i}^T\left[\mathbf{R}_n+\sum_{j\in\mathcal{B}(f_n)\backslash i}\mathbf{A}_{n,j}\left[\mathbf{W}_j^{-1}+\sum_{f_k\in\mathcal{B}(j)\backslash f_n}\alpha\mathbf{J}_{f_k\to j}^{(\ell)}\right]^{-1}\mathbf{A}_{n,j}^T\right]^{-1}\mathbf{A}_{n,i}.\end{aligned}$$

Therefore, $\alpha\mathcal{F}_{n\to i}\left(\left\{\mathbf{J}_{f_k\to j}^{(\ell)}\right\}_{(f_k,j)\in\widetilde{\mathcal{B}}(f_n,i)}\right)\succ\mathcal{F}_{n\to i}\left(\left\{\alpha\mathbf{J}_{f_k\to j}^{(\ell)}\right\}_{(f_k,j)\in\widetilde{\mathcal{B}}(f_n,i)}\right)$ if $\mathbf{J}_{f_k\to j}^{(\ell)}\succ\mathbf{0}$ for all $(f_k,j)\in\widetilde{\mathcal{B}}(f_n,i)$ and $\alpha>1$. As $\mathcal{F}$ contains $\mathcal{F}_{n\to i}(\cdot)$ as its component, Proposition 4, P 4.2 is proved. In the same way, we can prove $\mathcal{F}\left(\alpha^{-1}\mathbf{J}^{(\ell)}\right)\succ\alpha^{-1}\mathcal{F}\left(\mathbf{J}^{(\ell)}\right)$ if $\mathbf{J}^{(\ell)}\succ\mathbf{0}$ and $\alpha>1$.

At last, Proposition 4, P 4.3 is proved. From Lemma 2, if we have initial message information matrix $\mathbf{J}_{f_k\to j}^{(0)}\succeq\mathbf{0}$ for all $j\in\mathcal{V}$ and $f_k\in\mathcal{B}(j)$, then we have $\mathbf{J}_{f_k\to j}^{(\ell)}\succ\mathbf{0}$ for all

$j\in\mathcal{V}$ and $f_k\in\mathcal{B}(j)$. In such case, obviously, $\mathbf{J}^{(\ell)}\succeq\mathbf{0}$. Applying $\mathcal{F}$ to both sides of this equation, and using Proposition 4, P 4.1, we have $\mathcal{F}\left(\mathbf{J}^{(\ell)}\right)\succeq\mathcal{F}(\mathbf{0})$. On the other hand, using (27), it can be easily checked that $\mathcal{F}(\mathbf{0})=\mathbf{A}^T\left[\mathbf{\Omega}+\mathbf{H}\mathbf{\Psi}^{-1}\mathbf{H}^T\right]^{-1}\mathbf{A}\succ\mathbf{0}$, where the inequality is from Lemma 2. For proving the upper bound, we start from the fact that

$$\sum_{j\in\mathcal{B}(f_n)\backslash i}\mathbf{A}_{n,j}\left[\mathbf{W}_j^{-1}+\sum_{f_k\in\mathcal{B}(j)\backslash f_n}\mathbf{J}_{f_k\to j}^{(\ell-1)}\right]^{-1}\mathbf{A}_{n,j}^T$$

in (25), and equivalently the corresponding term

$$\mathbf{H}_{n,i}\left[\mathbf{W}_{n,i}+\mathbf{K}_{n,i}\left(\mathbf{I}_{|\mathcal{B}(f_n)|-1}\otimes\mathbf{J}^{(\ell-1)}\right)\mathbf{K}_{n,i}^T\right]^{-1}\mathbf{H}_{n,i}^T$$

in (26), are p.s.d. matrices. In (27), since

$$\mathbf{H}\left[\mathbf{\Psi}+\mathbf{K}\left(\mathbf{I}_{\sum_{n=1}^M|B(f_n)|(|B(f_n)|-1)}\otimes\mathbf{J}^{(\ell-1)}\right)\mathbf{K}^T\right]^{-1}\mathbf{H}^T$$

contains $\mathbf{H}_{n,i}\left[\mathbf{W}_{n,i}+\mathbf{K}_{n,i}\left(\mathbf{I}_{|\mathcal{B}(f_n)|-1}\otimes\mathbf{J}^{(\ell-1)}\right)\mathbf{K}_{n,i}^T\right]^{-1}\mathbf{H}_{n,i}^T$ as its block diagonal elements, it is also a p.s.d. matrix. With $\mathbf{\Omega}\succ\mathbf{0}$, adding to the above result gives

$$\mathbf{\Omega}+\mathbf{H}\left[\mathbf{\Psi}+\mathbf{K}\left(\mathbf{I}_\varphi\otimes\mathbf{J}^{(\ell)}\right)\mathbf{K}^T\right]^{-1}\mathbf{H}^T\succeq\mathbf{\Omega}\succ\mathbf{0}.$$

Inverting both sides, we obtain $\mathbf{\Omega}^{-1}\succeq\left[\mathbf{\Omega}+\mathbf{H}\left[\mathbf{\Psi}+\mathbf{K}\left(\mathbf{I}_\varphi\otimes\mathbf{J}^{(\ell)}\right)\mathbf{K}^T\right]^{-1}\mathbf{H}^T\right]^{-1}$. Finally, applying P B.2 again gives

$$\mathbf{A}^T\mathbf{\Omega}^{-1}\mathbf{A}\succeq\mathbf{A}^T\left[\mathbf{\Omega}+\mathbf{H}\left[\mathbf{\Psi}+\mathbf{K}\left(\mathbf{I}_\varphi\otimes\mathbf{J}^{(\ell)}\right)\mathbf{K}^T\right]^{-1}\mathbf{H}^T\right]^{-1}\mathbf{A}^T=\mathcal{F}\left(\mathbf{J}^{(\ell)}\right).$$

Therefore, we have $\mathbf{A}^T\mathbf{\Omega}^{-1}\mathbf{A}\succeq\mathcal{F}\left(\mathbf{J}^{(\ell)}\right)\succeq\mathbf{A}^T\left[\mathbf{\Omega}+\mathbf{H}\mathbf{\Psi}^{-1}\mathbf{H}^T\right]^{-1}\mathbf{A}\succ\mathbf{0}$.

Appendix D.

Let $\mathrm{d}(\mathbf{X}_1,\mathbf{Y}_1)=\exp\{a_1\}$ and $\mathrm{d}(\mathbf{X}_2,\mathbf{Y}_2)=\exp\{a_2\}$, and $\mathrm{d}(\mathbf{X}_1+\mathbf{X}_2,\mathbf{Y}_1+\mathbf{Y}_2)=\exp\{a_3\}$. First, P 15.1 is proved. According to the definition of part metric in Definition 9, for arbitrary symmetric p.d matrix $\mathbf{X}_1$, $\mathbf{X}_2$, $\mathbf{Y}_1$, and $\mathbf{Y}_2$, we have $\mathrm{d}(\mathbf{X}_1,\mathbf{Y}_1)$, $\mathrm{d}(\mathbf{X}_2,\mathbf{Y}_2)$, and $\mathrm{d}(\mathbf{X}_1+\mathbf{X}_2,\mathbf{Y}_1+\mathbf{Y}_2)$ correspond to

$$a_1\mathbf{X}_1\succeq\mathbf{Y}_1\succeq\frac{1}{a_1}\mathbf{X}_1,\quad a_2\mathbf{X}_2\succeq\mathbf{Y}_2\succeq\frac{1}{a_2}\mathbf{X}_2,\tag{67}$$

$$a_3(\mathbf{X}_1+\mathbf{X}_2)\succeq\mathbf{Y}_1+\mathbf{Y}_2\succeq\frac{1}{a_3}(\mathbf{X}_1+\mathbf{X}_2).\tag{68}$$

Since $\mathrm{d}(\mathbf{X}_1, \mathbf{Y}_1) > 0$ and $\mathrm{d}(\mathbf{X}_2, \mathbf{Y}_2) > 0$, we have $a_1, a_2 \geq 1$. And therefore $a_1 + a_2 > a_1$ and $a_1 + a_2 > a_2$. Then, according to (67), we have

$$(a_1 + a_2)(\mathbf{X}_1 + \mathbf{X}_2) \succeq \mathbf{Y}_1 + \mathbf{Y}_2 \succeq \frac{1}{a_1 + a_2}(\mathbf{X}_1 + \mathbf{X}_2). \tag{69}$$

Following the definition of part matric, a_3 is the smallest value satisfy the inequality in (68). Thus, by comparing (69) with (68), we obtain $a_1 + a_2 \geq a_3$ Hence, $\mathrm{d}(\mathbf{X}_1 + \mathbf{X}_2, \mathbf{Y}_1 + \mathbf{Y}_2) \leq \mathrm{d}(\mathbf{X}_1, \mathbf{Y}_1) + \mathrm{d}(\mathbf{X}_2, \mathbf{Y}_2)$.

Next, P 15.2 is proved. Following the part metric definition of $\mathrm{d}(\mathbf{X}_1, \mathbf{Y}_1)$, $a_1\mathbf{X}_1 \succeq \mathbf{Y}_1 \succeq \frac{1}{a_1}\mathbf{X}_1$, which is equivalent to $\mathbf{Y}_1^{-1} \succeq \frac{1}{a_1}\mathbf{X}_1^{-1}$ and $a_1\mathbf{X}_1^{-1} \succeq \mathbf{Y}_1^{-1}$. Thus, $\mathrm{d}(\mathbf{X}, \mathbf{Y}) = \mathrm{d}(\mathbf{X}^{-1}, \mathbf{Y}^{-1})$.

References

D. Bickson and D. Malkhi. A unifying framework for rating users and data items in peer-to-peer and social networks. *Peer-to-Peer Networking and Applications (PPNA) Journal*, 1(2):93–103, 2008.

E. G. Boman, D. Chen, O. Parekh, and S. Toledo. On factor width and symmetric H-matrices. *Linear algebra and its applications*, 405:239–248, 2005.

F. S. Cattivelli and A. H. Sayed. Diffusion LMS strategies for distributed estimation. *IEEE Trans. Signal Processing*, 58(3):1035–1048, 2010.

M. Chertkov and V. Y. Chernyak. Loop series for discrete statistical models on graphs. *Journal of Statistical Mechanics: Theory and Experiment*, 2006(06):P06009, 2006.

I. Chueshov. *Monotone Random Systems Theory and Applications*. New York: Springer, 2002.

P. G. Ciarlet. *Introduction to Numerical Linear Algebra and Optimisation*. Cambridge University Press, 1989.

J. W. Demmel. *Applied Numerical Linear Algebra*. SIAM, 1997.

J. Du and Y. C. Wu. Network-wide distributed carrier frequency offsets estimation and compensation via belief propagation. *IEEE Trans. Signal Processing*, 61(23):5868–5877, December 2013a.

J. Du and Y. C. Wu. Distributed clock skew and offset estimation in wireless sensor networks: Asynchronous algorithm and convergence analysis. *IEEE Trans. Wireless Commun.*, 12(11):5908–5917, Nov 2013b.

J. Du, S. Kar, and J. M. F. Moura. Distributed convergence verification for Gaussian belief propagation. In *Asilomar Conference on Signals, Systems, and Computers*, 2017a.

J. Du, S. Ma, Y. C. Wu, S. Kar, and J. M. F. Moura. Convergence analysis of belief propagation for pairwise linear Gaussian models. In *IEEE Global Conference on Signal and Information Processing*, 2017b.

B. J. Frey. Local probability propagation for factor analysis. In *Neural Information Processing Systems (NIPS)*, pages 442–448, December 1999.

P.-L. Giscard, S. Thwaite, and D. Jaksch. Walk-sums, continued fractions and unique factorisation on digraphs. *arXiv preprint arXiv:1202.5523*, 2012.

P.-L. Giscard, S. Thwaite, and D. Jaksch. Evaluating matrix functions by resummations on graphs: the method of path-sums. *SIAM Journal on Matrix Analysis and Applications*, 34(2):445–469, 2013.

P.-L. Giscard, Z. Choo, S. Thwaite, and D. Jaksch. Exact inference on Gaussian graphical models of arbitrary topology using path-sums. *Journal of Machine Learning Research*, 7 (2):1–19, February 2016.

V. Gómez, J. M. Mooij, and H. J. Kappen. Truncating the loop series expansion for belief propagation. *Journal of Machine Learning Research*, 8:1987–2016, 2007.

Y. Hu, A. Kuh, T. Yang, and A. Kavcic. A belief propagation based power distribution system state estimator. *IEEE Comput. Intell. Mag.*, 2011.

A. T. Ihler, J. W. Fisher III, and A. S. Willsky. Loopy belief propagation: Convergence and effects of message errors. *Journal of Machine Learning Research*, 6:905–936, 2005.

S. Kar. *Large Scale Networked Dynamical Systems: Distributed Inference*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, Department of Electrical and Computer Engineering, June 2010.

S. Kar and J. M. F. Moura. Consensus + innovations distributed inference over networks: cooperation and sensing in networked systems. *IEEE Signal Process. Mag.*, 30(3):99–109, 2013.

S. Kar, J. M. F. Moura, and H.V. Poor. Distributed linear parameter estimation: asymptotically efficient adaptive strategies. *SIAM Journal on Control and Optimization*, 51(3): 2200–2229, 2013.

U. Krause and R. Nussbaum. A limit set trichotomy for self-mappings of normal cones in Banach spaces. *Nonlinear Analysis, Theory, Methods & Applications*, 20(7):855–870, 1993.

F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Information Theory*, 47(2):498–519, February 2001.

F. Lehmann. Iterative mitigation of intercell interference in cellular networks based on Gaussian belief propagation. *IEEE Trans. Veh. Technol.*, 61(6):2544–2558, July 2012.

D. M. Malioutov, J. K. Johnson, and A. S. Willsky. Walk-sums and belief propagation in Gaussian graphical models. *Journal of Machine Learning Research*, 7(2):2031–2064, February 2006.

C. C. Moallemi and B. Van Roy. Convergence of min-sum message passing for quadratic optimization. *IEEE Trans. Information Theory*, 55(5):2413–2423, 2009a.

C. C. Moallemi and B. Van Roy. Convergence of min-sum message passing for quadratic optimization. *IEEE Transactions on Information Theory*, 55(5):2413–2423, 2009b.

J. M. Mooij and H. J. Kappen. Sufficient conditions for convergence of loopy belief propagation. In F. Bacchus and T. Jaakkola, editors, *Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pages 396–403, Corvallis, Oregon, 2005. AUAI Press.

J. M. Mooij and H. J. Kappen. Sufficient conditions for convergence of the sum-product algorithm. *IEEE Transactions on Information Theory*, 53(12):4422–4437, December 2007.

K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: an empirical study. In *15th Conf. Uncertainty in Artificial Intelligence (UAI), Stockholm, Sweden*, pages 467–475, July 1999.

National Research Council. *Frontiers in Massive Data Analysis*. Washington, DC: The National Academies Press, 2013.

B. L. Ng, J. S. Evans, S. V. Hanly, and D. Aktas. Distributed downlink beamforming with cooperative base stations. *IEEE Trans. Information Theory*, 54(12):5491–5499, December 2008.

N. Noorshams and M. J. Wainwright. Belief propagation for continuous state spaces: Stochastic message-passing with quantitative guarantees. *Journal of Machine Learning Research*, 14:2799–2835, 2013.

S. Ravanbakhsh and R. Greiner. Perturbed message passing for constraint satisfaction problem. *Journal of Machine Learning Research*, 16:1249–1274, 2015.

O. Shental, P. H. Siegel, J. K. Wolf, D. Bickson, and D. Dolev. Gaussian belief propagation solver for systems of linear equations. In *2008 IEEE International Symposium on Information Theory (ISIT 2008)*, pages 1863–1867, July 2008a.

O. Shental, P. H. Siegel, J. K. Wolf, D. Bickson, and D. Dolev. Gaussian belief propagation solver for systems of linear equations. In *2008 IEEE International Symposium on Information Theory*, pages 1863–1867, 2008b.

Q. Su and Y. C. Wu. On convergence conditions of Gaussian belief propagation. *IEEE Trans. Signal Processing*, 63(5):1144–1155, March 2015.

X. Tan and J. Li. Computationally efficient sparse Bayesian learning via belief propagation. *IEEE Trans. Signal Processing*, 58(4):2010–2021, April 2010.

Y. Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation,*, 12:1–41, 2000.

Y. Weiss and W. T. Freeman. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. *Neural Computation*, 13(10):2173–2200, March 2001a.

Y. Weiss and W. T. Freeman. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *IEEE Trans. Information Theory*, 47(2): 736–744, February 2001b.

R. Xiong, W. Ding, S. Ma, and W. Gao. A practical algorithm for tanner graph based image interpolation. In *2010 IEEE International Conference on Image Processing (ICIP 2010)*, pages 1989–1992, 2010.

E. Zeidler. *Nonlinear Analysis and its Applications IV-Applications to Mathematical Physics*. Springer-Verlag New York Inc., 1985.

G. Zhang, W. Xu, and Y. Wang. Fast distributed rate control algorithm with QoS support in ad-hoc networks. In *2010 IEEE Global Telecommunications Conference (GLOBECOM 2010)*, pages 1–5, December 2010.

Journal of Machine Learning Research 18 (2018) 1-5 Submitted 7/17; Revised 12/17; Published 4/18

auDeep: Unsupervised Learning of Representations from Audio with Deep Recurrent Neural Networks

Michael Freitag FREITAGM@FIM.UNI-PASSAU.DE
Shahin Amiriparian SHAHIN.AMIRIPARIAN@TUM.DE
Sergey Pugachevskiy PUGACHEV@FIM.UNI-PASSAU.DE
Nicholas Cummins NICHOLAS.CUMMINS@IEEE.ORG
Björn Schuller SCHULLER@IEEE.ORG
Chair of Embedded Intelligence for Health Care & Wellbeing
Augsburg University, Augsburg, Germany &
Chair of Complex & Intelligent Systems
Universität Passau, 94032 Passau, Germany &
GLAM – Group on Language, Audio & Music
Imperial College London, London, UK

Editor: Geoff Holmes

Abstract

auDeep is a Python toolkit for deep unsupervised representation learning from acoustic data. It is based on a recurrent sequence to sequence autoencoder approach which can learn representations of time series data by taking into account their temporal dynamics. We provide an extensive command line interface in addition to a Python API for users and developers, both of which are comprehensively documented and publicly available at `https://github.com/auDeep/auDeep`. Experimental results indicate that auDeep features are competitive with state-of-the art audio classification.

Keywords: deep feature learning, sequence to sequence learning, recurrent neural networks, autoencoders, audio processing

1. Introduction

Machine learning approaches for audio processing commonly operate on a variety of hand-crafted features computed from raw audio signals. Considerable effort has been put into developing high-performing feature sets for specific tasks. Recently, representation learning, in particular deep representation learning, has received significant attention as a highly effective alternative to using such conventional feature sets (Bengio et al., 2013; Schmitt and Schuller, 2017). These techniques have been shown to be superior to feature engineering for a plethora of tasks, including speech recognition and music transcription (Bengio et al., 2013; Amiriparian et al., 2016). Sequential data such as audio, however, poses challenges for deep neural networks, as they typically require inputs of fixed dimensionality. In this regard, sequence to sequence learning with *recurrent neural networks* (RNNs) has been proposed in machine translation, for learning fixed-length representations of variable-length sequences (Sutskever et al., 2014).

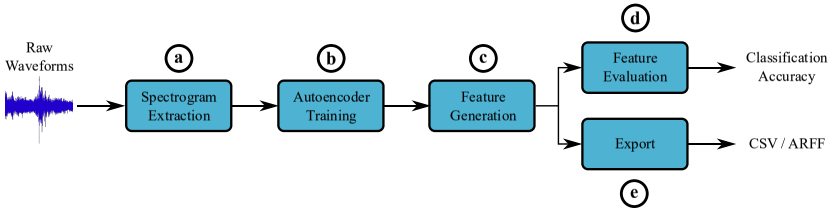


Figure 1: Illustration of the feature learning procedure with auDeep. A detailed description of the procedure is given in Section 3.1.

In this paper, we present auDeep, a first of its kind TensorFlow based Python toolkit for deep unsupervised representation learning from acoustic data. This is achieved using a recurrent sequence to sequence autoencoder approach. auDeep can be used both through its Python API as well as through an extensive command line interface.

2. Recurrent Sequence to Sequence Autoencoders

Our implementation of sequence to sequence autoencoders extends the RNN encoder-decoder model proposed by Sutskever et al. (2014). The input sequence is fed to a multi-layered *encoder* RNN which collects key information about the input sequence in its hidden state. The final hidden state of the encoder RNN is then passed through a fully connected layer, the output of which is used to initialise the hidden state of the multilayered *decoder* RNN. The function of the decoder RNN is to reconstruct the input sequence based on the information contained in the initial hidden state. The network is trained to minimise the root mean squared error between the input sequence and the reconstruction. In order to accelerate model convergence, the expected decoder output from the previous step is fed back as the input into the decoder RNN (Sutskever et al., 2014). Once training is complete, the activations of the fully connected layer are used as the representation of an input sequence.

For the purposes of representation learning from acoustic data, we train sequence to sequence autoencoders built of *long short-term memory* cells or *gated recurrent units* on spectrograms, which are viewed as time dependent sequences of frequency vectors. Two of the key strengths of this approach are (i) fully unsupervised training, and (ii) the ability to account for the temporal dynamics of sequences.

3. System Overview

auDeep contains at its core a high-performing implementation of sequence to sequence autoencoders which is not specifically constrained to acoustic data. Based on this domain-independent implementation, we provide extensive additional functionality for representation learning from audio.

3.1 Practical Usage

An illustration of the feature learning procedure with auDeep is shown in Figure 1. First, spectrograms are extracted from raw audio files (cf. Figure 1a). Then, a sequence to sequence autoencoder, as previously described, is trained on the extracted spectrograms (cf. Figure 1b), and the learned representation of each instance is extracted as its feature vector (cf. Figure 1c). If instance labels are available, a classifier can then be trained and evaluated on the extracted features (cf. Figure 1d). Finally, the extracted features, and any associated metadata, can be exported to CSV or ARFF for further processing, such as classification with alternate algorithms (cf. Figure 1e).

While our system is capable of learning representations of the extracted spectrograms entirely without additional metadata, we provide the possibility to parse instance labels, data set partitions, or a cross-validation setup from a variety of common formats. If available, metadata is stored alongside the extracted spectrograms, and can be used, e. g. for evaluation of a classifier on the learned representations. To demonstrate the strength of the learned representations two conventional classifiers are built into the toolkit: a *Multilayer Perceptron* (MLP) with softmax output, and an interface to LibLINEAR (Fan et al., 2008).

3.2 Design

auDeep provides a highly modularised Python library for deep unsupervised representation learning from audio. The core sequence to sequence autoencoder models are implemented using TensorFlow. This implementation substantially extends the built-in sequence to sequence learning capabilities of TensorFlow; for example, the RNNs allow probabilistic feedback and are also reusable, self-contained modules. Furthermore, diversely structured data sets are handled by the system in a unified way without requiring time-consuming manual adjustments. The topology and parameters of autoencoders are stored as TensorFlow checkpoints which can be reused in other applications. This enables users, e. g. to pretrain the encoder RNN with auDeep and subsequently apply custom retraining. Data sets are represented in binary format using the platform-independent NetCDF standard, but we provide tools for converting data between NetCDF and CSV/ARFF.

Users are given fine-grained control over the representation learning process through the Python API and the command line interface. The system is platform-independent, and has been tested on Windows and various Linux distributions on desktop PCs and in a cluster environment. auDeep is capable of running on CPU only, and GPU-acceleration is leveraged automatically when available.

4. Experiments

We demonstrate the capabilities of auDeep on three audio classification tasks. First, we perform acoustic scene classification on the development partition of the TUT Acoustic Scenes 2017 (TUT AS 2017) data set (Mesaros et al., 2017). Furthermore, we conduct environmental sound classification (ESC) on the ESC-10 and ESC-50 data sets (Piczak, 2015b), and, finally, we perform music genre classification on the GTZAN data set (Tzanetakis and Cook, 2002).

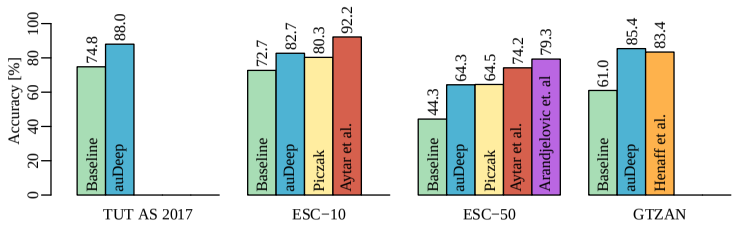


Figure 2: Comparison of auDeep to baselines and state-of-the-art on four data sets.

We train multiple autoencoder configurations using auDeep, and perform feature-level fusion of the learned representations. The fused representations are evaluated using the built-in MLP with the same cross-validation setup as used for the baseline systems on the TUT AS 2017, ESC-10, and ESC-50 data sets. For GTZAN, no predefined cross-validation setup is available, therefore we randomly generate five stratified cross-validation folds (cf. Figure 2, auDeep). Due to space limitations, we refrain from detailing our full experimental setup. However, to ensure reproducibility, we distribute the codes and parameter choices used for these experiments with auDeep.

Finally, we compare the performance of auDeep with baseline and state-of-the-art approaches for the different datasets (cf. Figure 2, identified by authors' names). We observe that auDeep either matches or outperforms a *convolutional neural network* approach (Piczak, 2015a) and a representation learning approach for ESC-10 and ESC-50, and a sparse coding approach for GTZAN (Henaff et al., 2011). The SoundNet (Aytar et al., 2016) and L3 (Arandjelovic and Zisserman, 2017) systems did achieve stronger performances on ESC-10 and ESC-50. However, this is not a straightforward comparison, as auDeep was trained using ESC-10 and ESC-50 data only whilst L3 and SoundNet were pre-trained on external corpora of 500 000 and 2+ million videos, respectively. For further details and comparisons with state-of-the-art for the TUT Acoustic Scenes 2017 corpus, the reader is referred to Amiriparian et al. (2017).

5. Conclusions

auDeep is an easy-to-use, open-source toolkit for deep unsupervised representation learning from audio with competitive performance on various audio classification tasks. Our long-term goal is to grow auDeep into a general-purpose deep audio toolkit, by integrating other deep representation learning algorithms such as conditional variational sequence to sequence autoencoders or *deep convolutional generative adversarial networks*, and by extending the feature learning functionality to regression tasks on continuously labelled data.

Acknowledgments

This research has received funding from the EU's 7^{th} Framework Programme through the ERC Starting Grant No. 338164 (iHEARu) and from the Innovative Medicines Initiative 2 Joint Undertaking under grant agreement No 115902.

References

S. Amiriparian, J. Pohjalainen, E. Marchi, S. Pugachevskiy, and B. Schuller. Is deception emotional? An emotion-driven predictive approach. In *INTERSPEECH*, pages 2011–2015. ISCA, 2016.

S. Amiriparian, M. Freitag, N. Cummins, and B. Schuller. Sequence to sequence autoencoders for unsupervised representation learning from audio. In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop*, pages 17–21. IEEE, 2017.

R. Arandjelovic and A. Zisserman. Look, listen and learn. In *2017 IEEE International Conference on Computer Vision*, pages 609–617. IEEE, 2017.

Y. Aytar, C. Vondrick, and A. Torralba. SoundNet: Learning sound representations from unlabeled video. In *Advances in Neural Information Processing Systems 29*, pages 892–900. Curran Associates, Inc., 2016.

Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8): 1798–1828, 2013.

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

M. Henaff, K. Jarrett, K. Kavukcuoglu, and Y. LeCun. Unsupervised learning of sparse features for scalable audio classification. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, pages 681–686. ISMIR, 2011.

A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen. DCASE 2017 challenge setup: Tasks, datasets and baseline system. In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop*. IEEE, 2017.

K. J. Piczak. Environmental sound classification with convolutional neural networks. In *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing*, pages 1–6. IEEE, 2015a.

K. J. Piczak. ESC: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM International Conference on Multimedia*, pages 1015–1018. ACM, 2015b.

M. Schmitt and B. W. Schuller. openXBOW—Introducing the Passau open-source crossmodal bag-of-words toolkit. *Journal of Machine Learning Research*, 18:1–5, 2017.

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014.

G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5):293–302, 2002.

Journal of Machine Learning Research 18 (2018) 1-54 Submitted 9/17; Revised 2/18; Published 4/18

On the Stability of Feature Selection Algorithms

Sarah Nogueira SARAH.NOGUEIRA@MANCHESTER.AC.UK
Konstantinos Sechidis KONSTANTINOS.SECHIDIS@MANCHESTER.AC.UK
Gavin Brown GAVIN.BROWN@MANCHESTER.AC.UK
School of Computer Science
University of Manchester
Manchester M13 9PL, UK

Editor: Isabelle Guyon

Abstract

Feature Selection is central to modern data science, from exploratory data analysis to predictive model-building. The "stability" of a feature selection algorithm refers to the *robustness* of its feature preferences, with respect to data sampling and to its stochastic nature. An algorithm is 'unstable' if a *small* change in data leads to *large* changes in the chosen feature subset. Whilst the idea is simple, *quantifying* this has proven more challenging—we note numerous proposals in the literature, each with different motivation and justification. We present a rigorous statistical treatment for this issue. In particular, with this work we consolidate the literature and provide (1) a deeper understanding of existing work based on a small set of properties, and (2) a clearly justified statistical approach with several novel benefits. This approach serves to identify a stability measure obeying all desirable properties, and (for the first time in the literature) allowing confidence intervals and hypothesis tests on the stability, enabling rigorous experimental comparison of feature selection algorithms.

Keywords: stability, feature selection

1. Introduction

High-dimensional data sets are the norm in data-intensive scientific domains. In application areas from bioinformatics to business analytics, it is common to collect many more measurements ("features" or "variables") than a study is able to easily cope with. This is a natural consequence of exploratory data analysis, but brings challenges of computational overhead, model interpretability and overfitting. Modern statistical regularisation methods can often control the model fit, but if the task is to identify *meaningful* subsets of features, there is a jungle of heuristics and domain-specific feature selection methods from which to pick, surveyed in several previous works, e.g. Guyon and Elisseeff (2003); Brown et al. (2012). Many authors have addressed the question of how sensitive each feature selection method is, with respect to small changes in the training data. If, with a different sample from the same training data, the chosen subset of features changes radically, then it is regarded as being an *unstable* procedure. Conversely, if the feature subset is almost static with respect to data changes, it is a *stable* procedure. Whilst the intuition here is clear, there is to date no single agreed measure to *quantify* stability, and numerous proposals in the literature.

The first published work to consider the *stability of feature selection procedures* was Kalousis et al. (2005), with extended experimental results published later as Kalousis et al. (2007). A slightly earlier technical report (Dunne et al., 2002) examined the idea in the limited scope of wrapper-based feature selection, but Kalousis et al. (2005) were the first to discuss stability in depth, independent of the particular feature selection algorithm. They defined stability as follows:

> *"We define the stability of a feature selection algorithm as the robustness of the feature preferences it produces to differences in training sets drawn from the same generating distribution P(X,C). Stability quantifies how different training sets affect the feature preferences."* (Kalousis et al., 2005, pg 2)

Kalousis et al. (2005) provided an excellent review of the issues, which we will not repeat here. One important point is how the feature preferences are represented—as a ranking, a weighting or a subset. Since any ranking or weighting can be thresholded to obtain a subset (which is often the case), the scope of this particular article is the stability of feature *subset* selection, with the other representations left for future work. The seminal work of Kalousis was followed by a flurry of publications in application areas where stability turns out to be critical, such as microarray classification (Davis et al., 2006), molecular profiling (Jurman et al., 2008) and linguistics (Wichmann and Kamholz, 2008). But, perhaps more interesting for this paper, there was also a flurry of *methodological* papers, addressing how best to quantify stability.

1.1 The Problem: How to Quantify and Estimate Stability

The *measurement* of stability is important, as it addresses a fundamental question in data science—*how much can we trust an algorithm?* If tiny changes to initial conditions result in significantly different conclusions, perhaps we should not trust the output as reflective of the true underlying mechanism. This is important, not just for pure interest's sake in machine learning, but a true interdisciplinary challenge. In biomedical fields, this is a proxy for *reproducible research* (Lee et al., 2012) indicating that whatever biological features the algorithm has found are likely to be a data artefact, not a real clinical signal worth pursuing with further resources. Jurman et al. (2008) argue that having a *stable* selected gene set is equally important as their predictive power, while Goh and Wong (2016, pg 1) state:

> *"Identifying reproducible yet relevant features is a major challenge in biological research.*[...] *We recommend augmenting statistical feature selection methods with concurrent analysis on stability and reproducibility to improve the quality of the selected features prior to experimental validation."*

This is the intuitive concept and motivation to study stability. However intuitive, precisely *quantifying* it has proven somewhat challenging. In a literature search, conducted in early 2018, we identified at least 15 different measures used to quantify the stability of feature selection algorithms (Dunne et al., 2002; Shi et al., 2006; Davis et al., 2006; Kalousis et al., 2007; Krízek et al., 2007; Kuncheva, 2007; Yu et al., 2008; Zucknick et al., 2008; Zhang et al., 2009; Lustgarten et al., 2009; Somol and Novovičová, 2010; Guzmán-Martínez and Alaiz-Rodríguez, 2011; Wald et al., 2013; Lausser et al., 2013; Goh and Wong, 2016). Most of these

were justified and evaluated, though there has been little cross-comparison. The question arises: which should we trust, in which situation? If we do not understand the properties of these measures, it leads to a questionable interpretation of the stability values obtained, and questionable research in general. As acknowledged by Boulesteix and Slawski (2009), a multiplicity[1] of methods for stability assessment may lead to publication bias—in that researchers may (hopefully unintentionally) be drawn toward the metric that reports their feature selection algorithm as more stable. Furthermore, rarely do authors acknowledge that the stability value obtained is *an estimate of a true stability*, based on the number of feature sets sampled. Any measure is an *estimator of an underlying random variable*—therefore we should be able to discuss statistical concepts such as the population parameter being estimated and the convergence properties of the estimator. In this paper, we provide such an estimator and a theoretical analysis of its properties.

1.2 Our Approach to the Problem

Our approach to this problem is to propose a small set of properties, describing desirable behaviours from a stability measure. We will argue that the properties are generic enough to be desirable in all reasonable feature selection scenarios, and that they are critical for useful comparison and interpretation of stability values. We proceed to prove whether the 5 properties hold for each of 15 measures already proposed in the literature, and find no measure satisfying all. We then propose a novel measure for which all properties provably hold, in Section 4. The proposed measure has several desirable characteristics which distinguish it from previous proposals:

1. It is based on 5 well-defined properties (Section 3) which we will argue are essential requirements for a stability measure in most (if not all) feature selection scenarios.

2. It has a clean statistical interpretation in terms of the sample variance of a set of Bernoulli variables. The clean interpretation allows us to derive a set of tools for practitioners including

   - confidence intervals for the true stability;
   - a hypothesis test to check if the true stability is above a user-defined threshold;
   - a hypothesis test to compare the stability of two algorithms on a given data set.

3. It is a proper generalization of several existing measures (and therefore the statistical tools we develop are also applicable to those measures).

4. It is computable in linear time as opposed to quadratic, as is the case for most measures in the literature.

5. Given the theoretical and computational properties above, it can reliably be used to select hyperparameters for feature selection algorithms, such as LASSO or Stability Selection (Meinshausen and Bühlmann, 2010).

1. The R package `OmicsMarkeR` provides 7 different options for measuring stability, with no guidance for which to use, in which situation (c.f. `www.rdocumentation.org/packages/OmicsMarkeR`).

In the following sections we explain our framework, first embarking on a brief review of existing literature. For a more thorough review and an extended version of this work, refer to Nogueira (2018). We also provide:

- The code in R and Matlab at `github.com/nogueirs/JMLR2018` for the proposed measure and associated statistical tools. The code for all experiments is also available, enabling reproducible research.

- A Python package and a demonstration notebook using the package at `github.com/nogueirs/JMLR2018/tree/master/python/`

- A demonstration web page at `www.cs.man.ac.uk/~gbrown/stability`

2. Background

We assume a data set of n examples $\{\mathbf{x}_i, y_i\}_{i=1}^n$ where each $\mathbf{x}_i$ is a d-dimensional feature vector and y_i is the associated label. The task of feature subset selection is to identify a subset of the dimensions, of size $k < d$, that conveys the maximum information about the label y. The challenge of feature selection can be tackled in various ways, commonly grouped in three families: filters, wrappers, and embedded methods (Guyon and Elisseeff, 2003). Filters assign a score to a feature subset (or a feature) based on statistics of the data, *independently* of any particular classifier—for example mutual information-based methods. Wrappers, on the other hand, evaluate a subset based on an error criterion (and therefore are classifier-specific). Because there are $2^d - 1$ possible feature subsets, filters and wrappers often use a search procedure such as a forward or backward search to only evaluate some of the subsets. Finally, embedded methods sit in between these two, choosing a subset as an integral part of learning a prediction model—for example LASSO and other penalized likelihood methods. The output of a feature selection algorithm is therefore either a weighting (or scoring) on the features, a ranking on the features or a subset of the features. Sorting the weights naturally gives a ranking on the features, and selecting the top-k ranked features gives a subset of the features. This way, the output of any weighting or ranking feature selection technique can be treated as a subset selection one (while the reverse is not true). The input to any given procedure is the data set, which itself is assumed to be a finite sample from a generating distribution. If the sample varies, logically the selected feature subset may vary—this variation is the *stability*.

It is important to note *why* instability may occur, and what is commonly done about it—the sources of, and solutions to instability. Several authors study how stability is influenced by data characteristics, such as noise (Shanab et al., 2011), data dimensionality, sample size (Alelyani, 2013), imbalance of the data set (Dittman et al., 2012) or feature redundancy (Gulgezen et al., 2009; Wald et al., 2013). Additionally, several works have been proposed to *increase* stability. These include variance reduction frameworks (Han and Yu, 2012), sample weighting (Yu et al., 2012), ensemble feature selection (Ditzler et al., 2015; Saeys et al., 2008) and multi-objective optimization (Baldassarre et al., 2017; Gulgezen et al., 2009; Kalousis et al., 2007).

It can be noted that each of these works, by definition, mandates the authors to *measure* stability—to know whether they have increased it, or decreased it. A typical approach to

measure stability is to first take M bootstrap samples of the provided data set, to apply feature selection to each one of them, and then to measure the variability in the M feature sets obtained. Approaches other than taking bootstraps have been considered, such as noise injection (Wald et al., 2012b; Altidor et al., 2011) or random subsampling (Wald et al., 2012a). However, in this article, we aim at measuring the variation with respect to data sampling. For this reason, we adopt the bootstrap approach due to its well understood properties and familiarity to the community.

Let $\mathcal{Z} = \{s_1, ..., s_M\}$ be a collection of feature sets, where each s_i is a subset of the d features and let a stability measure $\hat{\Phi}$ be a function taking as input $\mathcal{Z}$ and returning a stability value[2]. How can we define $\hat{\Phi}$ so that it measures the variability of the feature sets in $\mathcal{Z}$? In the literature, we find two approaches to this problem—the *similarity-based* approach and the *frequency-based* approach that we introduce in the next two sections.

2.1 Similarity-based Measures

First introduced by Dunne et al. (2002), the similarity-based approach consists in defining $\hat{\Phi}$ as the average pairwise *similarity* between the $M(M-1)$ possible pairs of feature sets in $\mathcal{Z}$, that is

$$\underbrace{\hat{\Phi}(\mathcal{Z})}_{\text{Stability measure}} = \frac{1}{M(M-1)} \sum_{i=1}^{M} \sum_{\substack{j=1 \\ j \neq i}}^{M} \underbrace{\phi(s_i, s_j)}_{\textit{Similarity} \text{ measure}} ,$$

where ϕ is a function taking two feature sets as input and returning a similarity value. The more the feature sets in $\mathcal{Z}$ are *similar* to each other on average, the larger the value of $\hat{\Phi}(\mathcal{Z})$ will be. Therefore, the definition of $\hat{\Phi}$ and its properties critically depend on the choice of a similarity measure ϕ. Dunne et al. (2002) proposed to use the relative Hamming distance between two feature sets as a measure of their similarity. Since it was introduced, this approach gained popularity in the literature and we found to date 8 other proposals of similarity measures to be used in this context (c.f. Appendix 2.1 for more details). Kalousis et al. (2007) proposed to use the Jaccard index, Yu et al. (2008) the Dice-Sørenson index, Zucknick et al. (2008) the Ochiai index and Shi et al. (2006) the *POG* index (*Percentage of Overlapping Genes*). Kuncheva (2007) analysed some of the existing stability measures and demonstrated that they were behaving in undesirable ways. It pioneered the property-based approach, proposing a new similarity measure based on a set of 3 properties, proven to be essential to the correct comparison and interpretation of stability values. Because of its well-known properties, the proposed measure was used in many works that followed. Nevertheless, Kuncheva's measure was only defined for feature selection algorithms that guarantee to return a constant number of features. Other authors proposed to extend this measure to more general scenarios where the number of features selected is not pre-determined by the user (Lustgarten et al., 2009; Wald et al., 2012b). Nevertheless, we will see in Section 3 that the proposed extensions somehow lose some of the desirable properties.

2. We include the 'hat' notation $\hat{\Phi}$ to acknowledge that the value is an *estimate* of an underlying quantity, dependent on the sample size M.

2.2 Frequency-based Measures

An alternative representation for a feature set is to regard the feature choices as a binary string of length d, where a 1 at the f^{th} position means feature X_f has been selected in the set and a 0 means it has not been selected. This representation has driven the frequency-based approach, where one can measure stability by (for example) looking at the frequencies of selection of each feature over the M feature sets. The collection of the M feature sets can therefore be modelled as a binary matrix $\mathcal{Z}$ of size $M \times d$, where a row represents a feature set and a column represents the selection of a given feature over the M repeats as follows

$$\mathcal{Z} = \begin{pmatrix} z_{1,1} & z_{1,2} & \cdots & z_{1,d} \\ z_{2,1} & z_{2,2} & \cdots & z_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ z_{M,1} & z_{M,2} & \cdots & z_{M,d} \end{pmatrix}.$$

In the remainder of this paper, we will denote the observed frequency of selection of feature X_f by $\hat{p}_f = \frac{1}{M} \sum_{i=1}^{M} z_{i,f}$, the mean of the f^{th} column of $\mathcal{Z}$. We also model the selection of the f^{th} feature by a Bernoulli variable Z_f with unknown mean p_f. Since these two alternative representations are equivalent, with a slight abuse of notation, we will interchangeably denote by $\mathcal{Z}$ a collection of M feature sets or the matrix of M binary vectors. Intuitively, frequencies closer to 0 or 1 will indicate higher stability as it will mean that a feature is either selected on almost all M feature sets or on almost none of them; but as we will later see, defining a measure in this category can also prove to be challenging. In total, we identified 6 stability measures in this category (c.f. Appendix A.2 for more details), which are all either a function the observed frequencies of selection of each feature (Davis et al., 2006; Goh and Wong, 2016; Guzmán-Martínez and Alaiz-Rodríguez, 2011) or of the observed frequencies of selection of each feature set (Krízek et al., 2007).

3. A New Set of Properties for Stability Measures

Given the variety of stability measures published, it is sensible to ask whether any one is more valid than the others. This seems somewhat of a philosophical question—what does it mean for one stability measure to be more "correct" than another? To answer this, we adopt the perspective that a measure should (1) provably obey certain *properties* that are desirable in the domain of application and (2) provide capabilities that other measures do not. In this section, we aggregate and generalize the requirements of the literature into a set of 5 properties[3], given in Table 1, that we will show to be critical to the interpretation and comparison of stability values. For each property, we also study which one of the existing measures satisfy the property and we summarize our findings in Table 2. Our results show that none of the existing measures possess all 5 properties, which leads us to Section 4.1 where we propose a new stability measure.

3. As opposed to what has been done in some previous works (Kuncheva, 2007), these properties are functions of the *stability* measure rather than the *set similarity* measure—this will allow to compare the properties of both pairwise and non-pairwise stability measures in a single framework.

1. *Fully defined.* The stability estimator $\hat{\Phi}$ should be defined for any collection $\mathcal{Z}$ of feature sets, thus allowing for the total number of features selected to vary.
2. *Strict Monotonicity.* The stability estimator $\hat{\Phi}$ should be a strictly decreasing function of the sample variances s_f^2 of the variables Z_f.
3. *Bounds.* The stability $\hat{\Phi}$ should be upper/lower bounded by constants not dependent on the overall number of features or the number of features selected.
4. *Maximum Stability* $\leftrightarrow$ *Deterministic Selection.* A measure should achieve its maximum if-and-only-if all feature sets in $\mathcal{Z}$ are identical.
5. *Correction for Chance.* Under the Null Model of Feature Selection H_0, the expected value of $\hat{\Phi}$ should be constant.

Table 1: Proposed Properties for a Stability Measure.

3.1 Fully Defined

The first property, *Fully Defined*, is that a stability measure $\hat{\Phi}$ should be able to cope with any collection $\mathcal{Z}$ of feature sets. We observed that some of the stability measures do not obey this property. More specifically, the measures proposed by Kuncheva (2007), Krízek et al. (2007), Guzmán-Martínez and Alaiz-Rodríguez (2011) and Lausser et al. (2013) are only defined for a feature selection algorithm that would always return a constant number of features (c.f. definitions in Appendix A). Stability measures not having this property will not be defined for a wide class of feature selection algorithms, such as $L1$-regularization, and therefore such stability measures cannot be used to compare the stability of feature selection algorithms of different types.

3.2 Strict Monotonicity

By definition, all 9 similarity measures proposed to quantify stability are a strictly increasing function of the size of the intersection $|s_i \cap s_j|$ between the two sets s_i and s_j given as input (c.f. Appendix A.1). Kuncheva (2007) explicitly states this as a required property for a similarity measure. This property is implicitly defining what similarity (and therefore stability) is. Since the stability is defined as the average pairwise similarities, in the more general case, this property would naturally translate to: *For a given collection of feature sets $\mathcal{Z}$, the stability $\hat{\Phi}$ should be an increasing function of the average pairwise intersection size* $\frac{1}{M(M-1)} \sum_{i=1}^{M} \sum_{\substack{j=1 \\ j \neq i}}^{M} |s_i \cap s_j|$. Even though this property can be applicable to any stability measure (as it is not phrased in terms of a similarity measure any more), it is not straightforward to comprehend the meaning of this property, neither to verify if a stability measure possesses this property, especially for frequency-based measures. We can therefore wonder what this does translate to in the frequency-based representation and how can

we verify that non-pairwise measures possess this property? Theorem 1 bridges the two approaches and justifies our second property, *Strict Monotonicity*[4].

Theorem 1 *The average pairwise intersection between the M feature sets is as a linear function of the sample variances of the selection of each feature. More precisely,*

$$\frac{1}{M(M-1)} \sum_{i=1}^{M} \sum_{\substack{j=1 \\ j \neq i}}^{M} |s_i \cap s_j| = \bar{k} - \sum_{f=1}^{d} s_f^2, \tag{1}$$

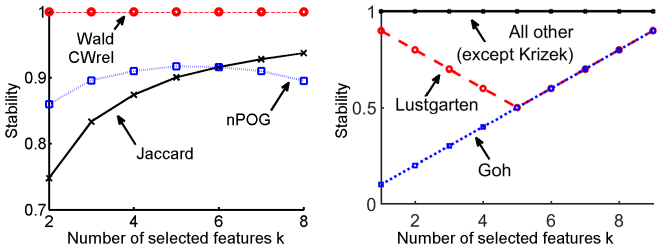
where $s_f^2 = \frac{M}{M-1} \hat{p}_f (1 - \hat{p}_f)$ is the unbiased sample variance of the selection of the f^{th} feature[5] *and where $\bar{k} = \frac{1}{M} \sum_{i=1}^{M} \sum_{f=1}^{d} z_{i,f}$ is the average number of features selected over the M feature sets.*

As we can see from Theorem 1, phrasing Monotonicity in terms of the average pairwise intersections is equivalent to phrasing it in terms of the average variance of the selection of each feature, which led to the definition of this property as stated in Table 1. This property defines *what* a stability measure should measure, rather than stating a necessary condition for a stability measure; therefore other proposals could be made for this purpose. For this reason, we will not discard the measures not having this property. However, it is interesting to note that, as we can see in Table 2, most stability measures have this property, showing some agreement upon the definition of stability across the literature, even if never stated as such. In some way, we can say that most existing measures of the literature implicitly aim at measuring the same quantity. In summary, we showed that: (1) interestingly, most stability measures of the literature possess this property, even though they were not explicitly built to that end, (2) the variance of the selection of each feature is an intuitive and simple way of measuring the variability in the choice features and (3) such a definition will allow us to derive a statistical framework for stability estimates, as we will later see in Section 4.2.

3.3 Bounds

This property was stated as necessary in several works. Somol and Novovičová (2010); Zucknick et al. (2008); Guzmán-Martínez and Alaiz-Rodríguez (2011) require a stability measure to be bounded by constants, while Kuncheva (2007) express the same requirement but for a similarity measure. Some of the measures such as Krízek et al. (2007) do not possess this property, since its maximum value depends on the number of features selected k and on the total number of features d (c.f. Appendix A.2). Unbounded measures do not allow for meaningful interpretation of stability values across problems or for different number of features selected, which can be restrictive in many applications.

4. To clarify: strict monotonicity is such that for a function g defined on a set D_g, g is a strictly monotonically (decreasing) function if $\forall x_1, x_2 \in D_g, x_1 < x_2 \Rightarrow g(x_1) > g(x_2)$. A counter-example showing the need for *strict* monotonicity (as opposed to monotonicity only) is to take any constant function: it will be monotonic as it will be an non-decreasing function of $|s_i \cap s_j|$ but cannot be interpreted as the similarity between two feature sets.
5. This expression of the sample variance is derived from a Bernoulli distribution.



(a) Forward implication (b) Backward implication

Figure 1: Illustration of the Maximum property. On the left, demonstration that Wald's measure and CW_{rel} (Somol and Novovičová, 2010) violate the forward implication. On the right, demonstration that Lustgarten's (Lustgarten et al., 2009) and Goh's (Goh and Wong, 2016) measures violate the backward implication[7].

3.4 Maximum Stability ↔ Deterministic Selection

For meaningful *interpretation* of a stability measure and comparison across problems, the range of values of a stability measure should be known and finite. Kuncheva (2007) states that IF two feature sets s_i and s_j are identical THEN their *similarity* is maximal, a desirable behaviour. Similarly, Guzmán-Martínez and Alaiz-Rodríguez (2011) also require that $\hat{\Phi}(\mathcal{Z})$ reaches its maximum whenever all the feature sets in $\mathcal{Z}$ are identical. We make this requirement a bi-implication, which translates in the general case to: $\hat{\Phi}(\mathcal{Z})$ reaches its maximum, *if-and-only-if* all feature sets in $\mathcal{Z}$ are identical, as stated in Table 1. We illustrate the need for the bi-implication below.

We used two scenarios, distinguishing the forward implication from the backward implication. First, we generated a collection of feature sets $\mathcal{Z}$ in which half of the feature sets are $\{X_1, ..., X_k\}$ and the other half are $\{X_1, ..., X_{k-1}\}$. Since there is clearly some variation in the features selected, the selection is not deterministic and we would want stability values not to be equal to their maximum. We plotted stability values against k for $d = 10$ and $M = 100$ in Figure 1a for some stability measures. We can see that Wald's measure (Wald et al., 2013) and CW_{rel} measure (Somol and Novovičová, 2010) still return their maximum value of 1. Therefore, these two measures violate the forward implication of this property. Second, we generated a collection of feature sets $\mathcal{Z}$ in which the same k features are selected on every repeat (for $d = 10$). Since the feature selection is now deterministic, we would want all stability values to be equal to their maximum. We plotted the stability values against the number of features selected k in Figure 1b. Even though the selection is completely deterministic, we can see that Lustgarten's and Goh's measure takes variable stability values depending on the number of selected features k. This shows that the two measures violate the backward implication of this property.

7. We ignored Krizek's measure in the right sub-figure as it is the only measure for which lower values correspond to higher stability and therefore, it should reach its minimum here instead of its maximum.

3.5 Correction for Chance

The fifth property, *Correction for chance*, was a novel concept introduced in the field by Kuncheva (2007) (but already existing in the statistical literature, c.f. Berry et al., 2016). This states that whenever we have *independently drawn subsets* at random, the stability value should be constant in expectation. This property was also later required by Lustgarten et al. (2009); Zhang et al. (2009); Guzmán-Martínez and Alaiz-Rodríguez (2011). When the number of features selected is constant, the notion of purely *random* feature selection is intuitive: it means that on each sample, given the number of selected features k, each one of the $\binom{d}{k}$ feature sets is equally likely to be chosen by the procedure. Now, let us take the case when the procedure does *not guarantee* to return a constant number of features, and thus produces a collection $\mathcal{Z}$ of feature sets, of varying size. We can still define the concept of *randomness* in that case: given the cardinality k_i of the i^{th} set, each one of the $\binom{d}{k_i}$ possible feature sets has an equal probability of being selected. We note that this is the assumption (sometimes implicitly) made by different authors using the concept of correction for chance, e.g. Kuncheva (2007); Lustgarten et al. (2009); Zhang et al. (2009); Guzmán-Martínez and Alaiz-Rodríguez (2011). Since we will use this concept multiple times in the remainder of the paper, we formalize this in Definition 2 and will refer to this assumption as the *Null Model of Feature Selection*.

Definition 2 (The Null Model of Feature Selection H_0) *We define the Null Model of Feature Selection as the situation where all possible permutations of the observed bits in a row of $\mathcal{Z}$ are equally likely. In other words, for all rows i in $\mathcal{Z}$, all subsets of size k_i have an equal probability of being drawn.*

Our final property, *Correction for chance* is that: under the Null Model of Feature Selection H_0, the expected value of $\hat{\Phi}$ should be constant, which for convenience we set to 0 (as done by other authors). The motivation behind such a property is that we would not want a stability measure to reflect the similarity between feature sets that might occur by chance, but only that due to the systematic decision-making of the feature selection algorithm. For illustrative purposes, we reproduced the experiment of Kuncheva (2007) in Figure 2. Let us assume that a feature selection procedure **randomly** selects k features out of $d = 10$ features and that we estimate the stability based on the $M = 100$ feature sets obtained for different values of k. As we can see, even though the feature selection procedure is random and therefore corresponds to a fully unstable situation (i.e. we are under the Null Model of Feature Selection H_0), some stability measures are strongly biased by the feature set size. For instance, we can see that using the Dice similarity measure, the stability systematically increases with the number of features selected, thus being in favour of larger feature sets. On the other hand, Kuncheva's measure that is corrected for chance gives a stability close to 0, no matter what the feature set size is as it is corrected for chance. Non-corrected measures make stability values neither comparable nor interpretable in different settings. To prove whether a measure has this property or not, we derived the value of $\mathbb{E}\left[\hat{\Phi}|H_0\right]$ for each of the existing measures in Appendix C.5 and reported the results in Table 2.

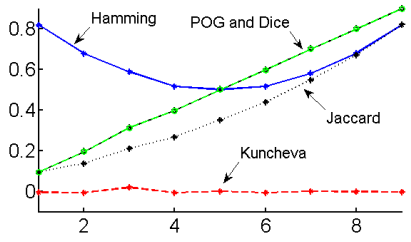


Figure 2: Illustration of the Correction-for-chance property. Stability values using Hamming, *POG*, Jaccard, Dice and Kuncheva similarity measures against the number k of features selected for $M = 100$ repeats.

Name	**Fully defined**	**Monotonicity**	**Bounds**	**Maximum**	**Correction**
Hamming	✓	✓	✓	✓	
Jaccard	✓	✓	✓	✓	
Dice	✓	✓	✓	✓	
Ochiai	✓	✓	✓	✓	
POG	✓	✓	✓	✓	
Kuncheva		✓	✓	✓	✓
Lustgarten	✓	✓	✓		✓
Wald	✓	✓			✓
nPOG	✓	✓		✓	✓
Goh	✓		✓		
Davis	✓		✓		
Krízek				✓	
Guzmán			✓	✓	✓
CW_{rel}	✓	✓	✓		
Lausser		✓	✓	✓	

Table 2: Properties of Stability Measures proposed in the literature 2002–2018. For each of the 15 measures, and for each of the 5 properties, we prove which measure satisfies which property — full proofs available in Appendix C.

3.6 Summary

In the literature, many authors advocated for the need of stability measures with well-known properties. In this section, we aggregated and generalised the properties stated as desirable in the literature into a set of desirable 5 properties[8]—applicable to any stability measure (whether similarity-based or not) and to any any feature selection algorithm (whether it selects a constant number of features or not)—in Table 1. Then, Table 2 summarizes the properties of each one of the stability measures. The first 5 measures are similarity-based measures and as we can see, they all possess all properties except Correction for chance. This weakness, noticed by Kuncheva (2007), gave rise to corrected-by-chance similarity-based measures, which are the 4 following measures of Table 2. As we can see, even though these 4 proposals all possess the Correction-for-chance property, they somehow lost some of the other desirable properties. Finally, the 6 following measures in the table are the frequency-based measures, which are more diverse in terms of the set of properties they satisfy. We note that even though CW_{rel} (Somol and Novovičová, 2010) does not possess the Correction-for-chance property, when the number of features selected is constant, we show in Appendix C.5 that it is asymptotically (as M approaches infinity) corrected for chance. We can therefore conclude that none of the stability measures in the literature possess all desired properties (even when discarding the Monotonicity property). Based on these results, we derive a novel stability measure in the next section, that not only has all desired properties, but also will allow us to develop a statistical framework for the quantification of stability.

4. A Novel Stability Measure

In this section, we propose a measure of stability which provably attains all desirable properties as discussed in the previous section. We recognise the stability measure $\hat{\Phi}$ as an estimator of a random variable, and aim to make explicit the corresponding population parameter Φ. By identifying the sampling distribution, we are able to provide tools for practitioners such as confidence intervals and hypothesis tests. This provides confidence in what the true value may be, and allows us to reliably compare stability across feature selection procedures. In the remainder of the paper, we refer to the stability measure as the *stability estimator* and to the parameter being estimated as the *population stability* or the *true stability*.

4.1 Proposed Stability Estimator

As required by the Monotonicity property, the stability should be a strictly decreasing function of the variances of the selection of each feature—for simplicity, we just negate the mean of the sample variances. As required by Correction for Chance, we rescale it by its expected value under the Null Model of Feature Selection. Finally for convenience of

8. Another property sometimes stated as desirable in the literature concerns the symmetry of the similarity measure ϕ (i.e. $\phi(s_i, s_j) = \phi(s_j, s_i)$) (Alelyani, 2013; P. and Perumal, 2016; Zucknick et al., 2008). We note that for any non-symmetric similarity measure ϕ, taking the arithmetic mean of $\phi(s_i, s_j)$ and $\phi(s_j, s_i)$ gives a symmetric similarity measure holding the same average pairwise value. Thus, the symmetry of a similarity measure ϕ is of little importance when comparing the properties of the corresponding stability measure $\hat{\Phi}$.

interpretation, we ensure that (asymptotically) the value is in the range $[0, 1]$ by taking one minus the resulting expression. Making use of Theorem 3 and by linearity of the expectation, this gives us our stability estimator as given in Definition 4.

Theorem 3 *Under the Null Model of Feature Selection H_0, for all f, the expected value of the sample variance of Z_f is $\mathbb{E}\left[s_f^2|H_0\right] = \frac{\bar{k}}{d}\left(1 - \frac{\bar{k}}{d}\right)$.*

Definition 4 (Novel Measure) *We define the stability estimator as*

$$\hat{\Phi}(\mathcal{Z}) = 1 - \frac{\frac{1}{d}\sum_{f=1}^{d} s_f^2}{\mathbb{E}\left[\frac{1}{d}\sum_{f=1}^{d} s_f^2|H_0\right]} = 1 - \frac{\frac{1}{d}\sum_{f=1}^{d} s_f^2}{\frac{\bar{k}}{d}\left(1 - \frac{\bar{k}}{d}\right)}, \tag{2}$$

where $s_f^2 = \frac{M}{M-1}\hat{p}_f(1-\hat{p}_f)$ is the unbiased sample variance of the selection of the f^{th} feature.

We now verify that the proposed measure possesses all 5 properties. First, by construction, we can see that the measure is defined for all collections $\mathcal{Z}$, unless the denominator $\frac{\bar{k}}{d}\left(1 - \frac{\bar{k}}{d}\right)$ is equal to zero. This happens whenever $\bar{k} = 0$ or $\bar{k} = d$, which are the two limit cases in which the algorithm does not select any features over the M feature sets in $\mathcal{Z}$ or selects all of the features on every feature set of $\mathcal{Z}$. Since by definition, a feature selection procedure will always select a non-empty proper subset of the set of all available features, the first property *Fully defined* holds for the proposed definition of $\hat{\Phi}$. Second, by construction, $\hat{\Phi}$ is a linear function of the sample variances s_f^2 with a strictly negative slope. Therefore, the Monotonicity property holds for $\hat{\Phi}$. Third, to conform to the Bounds property—by inspection, one can see the numerator and denominator in the measure are positive quantities, therefore the upper bound is 1. For the lower bound, Appendix D shows that the minimum of $\hat{\Phi}$ is equal to $-\frac{1}{M-1}$. This shows that $\hat{\Phi}$ is bounded by -1, but is asymptotically bounded by 0 (as M approaches infinity). Fourth, let us assume for some $\mathcal{Z}$, the measure achieves its maximum, $\hat{\Phi}(\mathcal{Z}) = 1$. This state is equivalent to $\sum_{f=1}^{d} s_f^2 = 0$. Since the variance is a positive quantity, this is in turn equivalent to $s_f^2 = 0$ for all f. This case corresponds to the situation where each column of $\mathcal{Z}$ contains either all 1s or all 0s. Thus $\hat{\Phi}$ is equal to its maximum if-and-only-if all feature sets in $\mathcal{Z}$ are identical. Fifth, under the Null Model of Feature Selection H_0, by linearity of the expectation and using Theorem 3, $\mathbb{E}\left[\hat{\Phi}|H_0\right] = 0$. Therefore, the proposed measure is corrected for chance.

Theorem 5 shows that the proposed measure of stability $\hat{\Phi}$ is a generalization of some other widely used measures to feature sets of varying cardinality, hence being consistent with previous literature. As a result, all statistical tools of Section 4.2 are also valid for Kuncheva's measure and for the other equivalent measures, hence unifying some of the theory on the measurement of stability. The reformulation of Kuncheva's measure using our definition also provides a computational advantage, being $O(Md)$ instead of $O(M^2d)$ (which is the computational complexity of all pairwise measures).

Theorem 5 *When the number of features selected is constant:*

- *The stability estimator $\hat{\Phi}$ is equal to the stability measures proposed by Kuncheva (2007), Wald et al. (2013) and to nPOG (Zhang et al., 2009).*

- *The stability estimator $\hat{\Phi}$ and CW_{rel} (Somol and Novovičová, 2010) are asymptotically equivalent.*

In the next section, we provide the population parameter Φ estimated by $\hat{\Phi}$, that is the *true* stability of the feature selection procedure considered.

4.2 Statistical Tools

In the previous section, we proposed a new stability measure that possesses all desirable properties and is a generalization of some of the existing measures of the literature. We can also ask how it relates to other parts of the literature and which other fields deal with similar problems. This section demonstrates and exploits a relationship between stability and inter-rater agreement (Fleiss, 1971).

4.2.1 Viewing Stability as Inter-rater Agreement

Imagine a medical scenario—we have M doctors (more formally called *raters*) assigning a nominal category $\{1, 2, ..., q\}$ to each member of a set of d patients (called *subjects*). A useful indication of the agreement of the M raters is given by *inter-rater agreement coefficients*. We can view stability in this light—when the number of categories q is equal to 2, and each row of $\mathcal{Z}$ represents a rater, placing the d subjects into category 0 or 1. Interestingly[9], in this special case, we prove with Theorem 6 that a popular measure of inter-rater agreement, Fleiss' Kappa (Fleiss, 1971) reduces to our estimator, Definition 4. As a result, any statistical result previously derived for Fleiss' Kappa also holds for $\hat{\Phi}$. Using this relationship, we can use the work of Gwet (2008) which shows the asymptotic normality of Fleiss' Kappa, hence guaranteeing the validity of confidence intervals and hypothesis tests for large samples.

Theorem 6 *When there are only two categories (0/1), Fleiss' Kappa is equal to $\hat{\Phi}(\mathcal{Z})$.*

4.2.2 The Sampling Distribution of Stability

Let us assume each row of the matrix $\mathcal{Z}$ is an independent sample from the joint distribution $(Z_1, ..., Z_d)$, where Z_f is a Bernoulli variable with unknown population parameter p_f, where we make no assumption of independence between d covariates. In the original paper, Fleiss (1971) derives the variance of Fleiss' Kappa, but only when Φ is equal to 0, which is of little use in our case. Later on, Gwet (2008) provides a variance estimate and the asymptotic distribution of Fleiss' Kappa in the general case. In his work, Gwet (2008) assumes that the raters (samples) and subjects (features) are sampled from a larger population and then derives the variance due to the sampling of raters and the variance due to the sampling of subjects. Using the multivariate Central Limit Theorem and a linear approximation of $\hat{\Phi}$, Gwet (2008) shows that $\hat{\Phi}$ is asymptotically normal. Gwet (2008) also verifies the validity of this result for the construction of confidence intervals with Monte Carlo simulations. In our case, we assume that there is no sampling of the subjects and that the number of categories

9. Another interesting relationship that could be used in future work is that Fleiss' Kappa has also been linked to the Intra-Class Correlation Coefficient (ICC) in the binary case (Fleiss et al., 2004), so any result that applies to the ICC can also be applied to the proposed stability estimator.

$q = 2$. Under these assumptions, the variance due to the sampling of subjects derived by Gwet (2008) becomes zero and the asymptotic distribution of the stability estimator $\hat{\Phi}$ becomes the one given by Theorem 7.

Theorem 7 (Asymptotic Distribution) *As $M \to \infty$, the statistic $\hat{\Phi}$ weakly converges to a normal distribution, that is*

$$\frac{\hat{\Phi} - \Phi}{\sqrt{v(\hat{\Phi})}} \xrightarrow{\mathcal{L}} \mathcal{N}(0,1),$$

where:

- $\Phi = 1 - \frac{\frac{1}{d}\sum_{f=1}^{d} p_f(1-p_f)}{\bar{p}(1-\bar{p})}$ *is the mean of the estimator $\hat{\Phi}$ in which $\bar{p} = \frac{1}{d}\sum_{f=1}^{d} p_f$ is the average mean parameter of the d Bernoulli variables;*
- $v(\hat{\Phi}) = \frac{4}{M^2}\sum_{i=1}^{M}(\hat{\Phi}_{(i)} - \bar{\hat{\Phi}}_{(.)})^2$ *is an estimate of the variance of $\hat{\Phi}$ in which:*
  - $\hat{\Phi}_{(i)} = \frac{1}{\frac{k_i}{d}\left(1-\frac{k_i}{d}\right)}\left[\frac{1}{d}\sum_{f=1}^{d} z_{i,f}\hat{p}_f - \frac{k_i\bar{k}}{d^2} + \frac{\hat{\Phi}}{2}\left(\frac{2\bar{k}k_i}{d^2} - \frac{k_i}{d} - \frac{\bar{k}}{d} + 1\right)\right]$;
  - *and $\bar{\hat{\Phi}}_{(.)}$ is the average value of $\hat{\Phi}_{(i)}$, that is $\frac{1}{M}\sum_{i=1}^{M}\hat{\Phi}_{(i)}$.*

This asymptotic distribution allows us to identify $\hat{\Phi}$ as being an estimator of an unknown population quantity Φ. In the remainder of the paper, we refer to $\hat{\Phi}$ as being the sample stability and to Φ as being the population (or *true*) stability. The asymptotic convergence shows that $\hat{\Phi}$ is a consistent estimator of the population stability Φ. This means that as M approaches infinity, we are assured that the stability estimator $\hat{\Phi}$ will converge in probability to the population stability Φ.

4.2.3 Confidence Intervals

The asymptotic convergence to a normal distribution allows us to derive approximate confidence intervals for the population stability Φ, given below in Corollary 8. Though the provided confidence intervals are only approximate, we will see in Section 5.2 that even for relatively small values of M, the given intervals still have a good coverage probability.

Corollary 8 (Confidence Intervals) *A $(1-\alpha)$%-approximate confidence interval for Φ is*

$$\left[\hat{\Phi} - z^*_{\left(1-\frac{\alpha}{2}\right)}\sqrt{v(\hat{\Phi})}, \hat{\Phi} + z^*_{\left(1-\frac{\alpha}{2}\right)}\sqrt{v(\hat{\Phi})}\right],$$

where $z^*_{\left(1-\frac{\alpha}{2}\right)}$ is the inverse cumulative of a standard normal distribution at $1-\frac{\alpha}{2}$.

4.2.4 Hypothesis Testing

In a first scenario, let us assume a practitioner applies a feature selection procedure to M bootstrap samples, generating a matrix $\mathcal{Z}$ of size $M \times d$, and computes the stability estimate $\hat{\Phi}(\mathcal{Z})$. How can we know whether the true stability Φ is significantly greater than

a fixed value Φ_0? This can be defined formally in terms of a null hypothesis significance test.

Is the Population Stability Φ Greater than a Given Value Φ_0? In this case the hypothesis tested is

$$\begin{cases} H_0 : \Phi = \Phi_0 \\ H_1 : \Phi > \Phi_0 \end{cases}$$

Under H_0, $\Phi = \Phi_0$ and therefore the statistic $V_M = \frac{\hat{\Phi}-\Phi_0}{\sqrt{v(\hat{\Phi})}}$ is asymptotically standard normal (c.f. Theorem 7). Therefore we can apply a one-tail test as follows:

1. Compute the statistic V_M.
2. Reject H_0 if $V_M \geq z^*_{(1-\alpha)}$, where the critical value $z^*_{(1-\alpha)}$ is the $(1-\alpha)$th percentile of the standard normal distribution.

In addition, it is very common to compare stability values between algorithms. For example, Saeys et al. (2008) conclude that *"RELIEF is one of the less stable algorithms"* and *"Random Forests clearly outperform other feature selection methods regarding robustness"*. So given two stability estimates $\hat{\Phi}(\mathcal{Z}_1)$ and $\hat{\Phi}(\mathcal{Z}_2)$, can we conclude that the true stability of the first is significantly different than the second?

Do Two Feature Selection Algorithms Have Identical Stabilities? Let $\mathcal{Z}_1$ and $\mathcal{Z}_2$ be the output of two feature selection procedures. In this case, we wish to test the following hypothesis

$$\begin{cases} H_0 : \Phi_1 = \Phi_2 \\ H_1 : \Phi_1 \neq \Phi_2 \end{cases}$$

Using the asymptotic distribution of $\hat{\Phi}(\mathcal{Z}_1)$ and of $\hat{\Phi}(\mathcal{Z}_2)$ given by Theorem 7, we can derive Theorem 9. Using the given test statistic T_M, we reject H_0 if $|T_M| \geq \theta$, where θ is the $\left(1-\frac{\alpha}{2}\right)^{th}$ percentile of the standard normal distribution.

Theorem 9 *The test statistic for comparing stabilities is*

$$T_M = \frac{\hat{\Phi}(\mathcal{Z}_2) - \hat{\Phi}(\mathcal{Z}_1)}{\sqrt{v(\hat{\Phi}(\mathcal{Z}_1)) + v(\hat{\Phi}(\mathcal{Z}_2))}}.$$

Under H_0, the statistic T_M asymptotically (as M approaches infinity) follows a standard normal distribution.

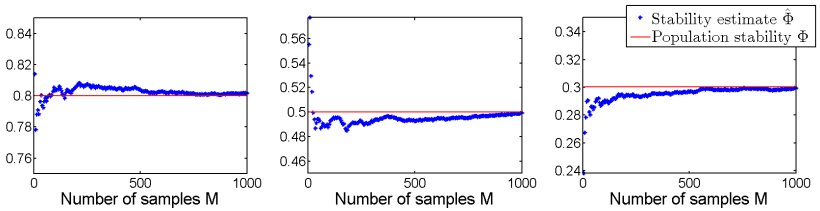
5. Empirical Validation of the Statistical Tools

Fleiss et al. (2004) propose a benchmark scale for interpretation of the value of Fleiss' Kappa. We will use the same scale for $\hat{\Phi}(\mathcal{Z})$, provided in Table 3. Stability values above 0.75 represent an excellent agreement of the feature sets beyond chance, while values below 0.4 represent a poor agreement between sampled feature sets.

In the remainder of this section, we verify the tools of the previous section, using toy data for a population stability Φ in each one of the categories. To be able to generate

Φ	**Strength of Agreement**
< 0.40	Poor
0.40 to 0.75	Intermediate to good
> 0.75	Excellent

Table 3: Benchmark scale for stability.



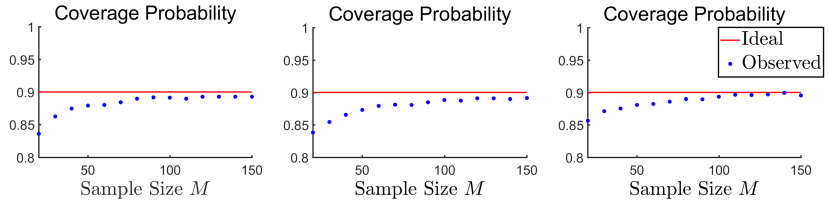
(a) Case 1: $\Phi = 0.8$ (b) Case 2: $\Phi = 0.5$ (c) Case 3: $\Phi = 0.3$

Figure 3: Consistency of the stability estimate $\hat{\Phi}(\mathcal{Z})$ for the 3 toy cases. As M increases, the value of $\hat{\Phi}(\mathcal{Z})$ gets closer to the population parameter Φ.

Bernoulli variables with a specified population stability Φ_0, we first need to chose d Bernoulli parameters $p_1, ..., p_d$ such that $\Phi = \Phi_0$. We picked 3 test cases for the values of Φ equal 0.8, 0.5 and 0.3 and for $d = 100$.

5.1 Validation of Consistency of the Estimator

In this section, we empirically show the consistency of the stability estimator $\hat{\Phi}$ in the 3 test cases described. In each case, we take M samples from the $d = 100$ Bernoulli variables with mean parameters $(p_1, ..., p_d)$. This gives us a binary matrix $\mathcal{Z}$ of size $M \times d$. We then plot the value of the stability estimate $\hat{\Phi}(\mathcal{Z})$ as we increase the number of samples M in Figure 3. As we can see, as the number of samples M increases, the value of $\hat{\Phi}(\mathcal{Z})$ approaches the true stability Φ. We chose similar scales for the 3 test cases. We observe that for relatively small values of M, (generally for $M \leq 10$), the absolute value of the difference $|\hat{\Phi}(\mathcal{Z}) - \Phi|$ is within 5% and for $M \leq 100$, within 1%. Of course, these cannot be used as a general rule of thumb for other chosen parameters p_f and d but it gives an idea of the rate of convergence of the stability estimates. In real applications, the population stability is unknown and we need tools to be able to determine which interval of values the population stability takes. This is the topic of the next section where we study the confidence intervals.



(a) Case 1: $\Phi = 0.8$ (b) Case 2: $\Phi = 0.5$ (c) Case 3: $\Phi = 0.3$

Figure 4: Coverage probabilities for the 3 test cases. We set the nominal level to be 0.90 (90%-confidence interval). The x-axis represents the sample size M and the y-axis the estimated coverage probability with 10000 repeats.

5.2 Validation of Confidence Intervals

The coverage probability of a confidence interval for Φ is the proportion of time the interval built from the data will actually contain the population stability Φ. To verify the results of Theorem 8 providing the confidence intervals, we adopted the following procedure:

1. Compute the population stability Φ using the true Bernoulli parameters $(p_1, ..., p_d)$.
2. Repeat 10,000 times:
   - Take M samples from the d variables $Z_1, ..., Z_d$ with mean $p_1, ..., p_d$.
   - Compute the $(1 - \alpha)$-approximate confidence interval using Corollary 8.
3. The estimated coverage probability is the fraction of times (from 10,000) that the true stability Φ was within the confidence intervals.

If we had exact confidence intervals, we should have an exact coverage probability of $(1-\alpha)$. However, the confidence intervals derived are only approximate. Figure 4 gives the estimated coverage probabilities in the 3 test cases for $\alpha = 10\%$, i.e. a 90%-confidence interval.

As we can see, the estimated probability quickly approaches $(1-\alpha) = 0.9$ as expected. In Table 4, we further show the observed coverage probabilities in the 3 test cases for $M = 100$ and for different values of α. The same behaviour can be seen in this table: the values get very close to the expected coverage probability of $(1-\alpha)$. We observed the same behaviour for a larger number of features ($d = 10000$) and a same number of feature sets M.

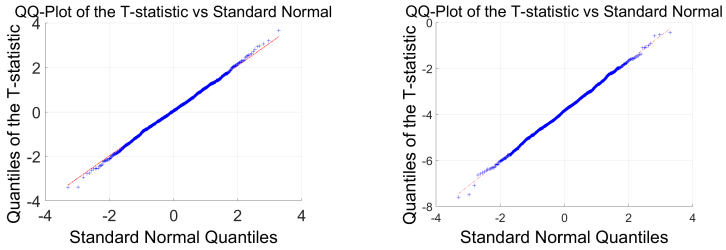
5.3 Validation of the Second Hypothesis Test

In this section, we verify the asymptotic distribution of the test statistic T_M as given by Theorem 9. To verify this result we proceed as follows:

1. Pick two set of parameters $p_1, ..., p_d$ with the desired values of Φ_1 and Φ_2.
2. Repeat 1000 times:

Case 1	$\Phi = 0.8$	98.5%	94.3%	89.0%
Case 2	$\Phi = 0.5$	98.6%	93.8%	89.0%
Case 3	$\Phi = 0.3$	98.6%	94.0%	89.3%
Ideal		99%	95%	90%

Table 4: Coverage probabilities for the 3 test cases with $M = 100$, $d = 100$, estimated via 10,000 repeats for different nominal confidence intervals.



(a) $\Phi_1 = \Phi_2 = 0.8$ (b) $\Phi_1 = 0.8$ and $\Phi_2 = 0.75$

Figure 5: QQ-plots illustrating the convergence of the statistic T_M to a standard normal distribution when $\Phi_1 = \Phi_2 = 0.8$ [LEFT] and the convergence to a non-standard Gaussian when $\Phi_1 \neq \Phi_2$ [RIGHT]. We took $d = 100$, $M = 1000$ and 1000 repeats. We note that the range of values on the y-axis of the right plot is not the same as the one of the left plot.

- Take M samples from the d variables, for each of $\mathcal{Z}_1$ and $\mathcal{Z}_2$
- Compute the corresponding statistic T_M.

We then order the 1000 estimates of T_M and plot the quantiles against that of a standard normal distribution in a Quantile-Quantile plot (QQ-plot). Figure 5 provides the result for two test cases. In the left sub-figure, the population stabilities are taken to be identical (i.e. $\Phi_1 = \Phi_2$). In that situation, the QQ-plot shows that the quantiles of the test statistic T_M are identical to the ones of a standard normal distribution, which is the result we expected. On the right sub-figure, we chose different population stabilities (i.e. $\Phi_1 \neq \Phi_2$). In this case, the QQ-plot shows that the quantiles are still the ones of a normal distribution but with a mean different from 0. Indeed, if we have a closer look at the right sub-figure, we can see that the range of values taken on the y-axis are all negative. The observed median of the statistic T_M in that case is of -3.8 and the statistic takes values in the interval $[-7.6, -0.4]$. Therefore the statistic T_M is not standard normal in that situation.

6. Experiments

The experiments of this section illustrate how the tools presented in this paper can be used by practitioners to select hyperparameters with higher stability or to compare the stability of different feature selection algorithms. This section contains two sets of experiments[10]:

- Section 6.1 focuses on the LASSO and the Elastic Net, which are both regularized regression models that select features as part of the training process. The Elastic Net is known to yield more stable coefficients than the LASSO in the presence of redundant features (Zhou, 2013). This section shows that the proposed stability measure captures this and that choosing a good trade-off between stability and accuracy can reduce the number of irrelevant features in the model with negligible loss in accuracy.
- Section 6.2 focuses on a popular technique, *Stability Selection* (Meinshausen and Bühlmann, 2010) which we apply to LASSO. The proposed framework defines the stable set as the set of features being selected with high frequency across a set of regularizing parameters. We propose to look at the stability of the *stable set* for different hyperparameters and compare it to the stability of LASSO.

6.1 The Stability of $L1/L2$ Regularized Logistic Regression

In this section, we observe how the degree of redundancy in data can affect the stability of LASSO and Elastic Net, and how we can optimize hyperparameters so that both log-likelihood and stability are taken into account. We show that stability can help recover the *true* set of relevant features. To be able to control the set of relevant features and the redundancy between them, we use a synthetic data set as described in the next section.

6.1.1 Description of the Data Set

We use a synthetic data set (Kamkar et al., 2015)—a binary classification problem, with 2000 instances and $d = 100$ features, where only the first 50 features are relevant to the target class. Instances of the positive class are identically and independently drawn from a normal distribution with mean $\mu_+ = (\underbrace{1, ..., 1}_{50}, \underbrace{0, ..., 0}_{50})$ and covariance matrix

$$\Sigma = \begin{bmatrix} \Sigma^*_{50\times 50} & \mathbf{0}_{50\times 50} \\ \mathbf{0}_{50\times 50} & \mathbf{I}_{50\times 50} \end{bmatrix},$$

where $\Sigma^*_{50\times 50}$ is the matrix with ones on the diagonal and ρ (a parameter in $[0, 1]$) controlling the degree of redundancy everywhere else. The mean for the negative class is taken equal to $\mu_- = (\underbrace{-1, ..., -1}_{50}, \underbrace{0, ..., 0}_{50})$. The larger the value of ρ, the more the 50 relevant features will be correlated to each other.

10. You can reproduce all these experiments (in Matlab or R) with the code given at `github.com/nogueirs/JMLR2018`

6.1.2 Stability of LASSO

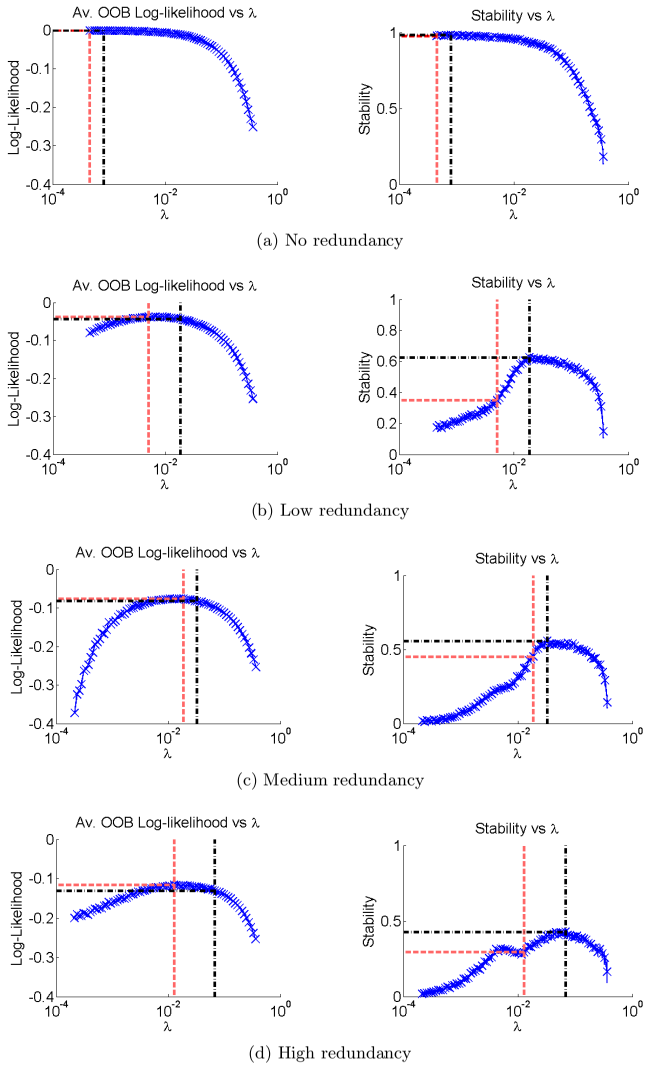
We use a $L1$-regularized logistic regression where λ is the regularizing parameter, influencing the amount of features selected—as λ increases, more and more coefficients are equal to zero and therefore less and less features are selected. We take 2000 samples and divide them into 1000 for model selection (i.e. to select the regularizing parameter λ) and 1000 for selection of the final set of features. The model selection set can be used simply to optimize error, or to optimize error/stability simultaneously—the experiments will demonstrate that the latter provides a lower false positive rate in the final selection of features. We study 4 degrees of redundancy: $\rho = 0$ (no redundancy, the features are independent from each other), $\rho = 0.3$ (low redundancy), $\rho = 0.5$ (medium) and $\rho = 0.8$ (high). We apply $L1$-logistic regression to $M = 100$ bootstrap samples of the data set. We then compute the average out-of-bag (OOB) log-likelihood[11] and the stability of the feature selection.

Figure 6 shows the average log-likelihood (left column) and the stability (right column) versus the regularization parameter λ for the 4 degrees of redundancy chosen. For each degree of redundancy, the pink dashed-line represents the parameter λ maximizing the likelihood and the black one represents the parameter maximizing the stability. In the case of no redundancy, we can see that these two parameters produce similar values of likelihood and stability. But, as we change the degree of redundancy, this is not the case. The result can most strongly be seen in Figure 6b, where $\lambda = 0.0051$ optimizes likelihood, but if we increase to $\lambda = 0.0187$, we sacrifice a negligible amount of likelihood for a quite significant increase in stability to $\hat{\Phi} = 0.63$.

Figure 7 shows an alternative view of these results, plotting stability against likelihood. When there is no redundancy in the data (sub-figure (a)), stability seems to be an increasing function of the likelihood. For higher levels of redundancy, this results in at least *two* values of λ which achieve the *same likelihood*, but clearly one results in higher stability. In practice, to choose a hyperparameter λ, we have to pick a trade-off between likelihood of the model and stability. For this purpose, we can identify the pareto front of likelihood and stability, which is the set of points such that there is no other point with higher likelihood and higher stability. Hence, each point in the pareto front represents a different trade-off between likelihood and stability. Figure 8 summarizes the pareto fronts for the 4 degrees of redundancy. In a classic scenario, we would pick the value of λ that maximizes the likelihood only, which corresponds to the rightmost point in the figure. However, we can see that sacrificing a small amount of likelihood allows us to considerably increase stability. As we increase the degrees of redundancy, we see the best case for stability is lower. Nevertheless, all the points in a given pareto front have a similar likelihood and a similar misclassification rate. All these observations show that *stability can potentially be increased without loss of predictive power.*

We also observe that pursuing stability may help identifying the *relevant* set of features. Figure 9 gives the observed frequencies of selection $\hat{p}_f$ of each feature over the $M = 100$ bootstraps—where features $1-50$ are relevant, and $51-100$ are irrelevant in the case of high redundancy. We picked the value of λ that maximized the likelihood for the left sub-figure and a value of λ in the pareto front of stability and likelihood for the right sub-figure. On the right figure, all 50 irrelevant features have a frequency of selection equal to 0 (i.e. the

11. In all the presented results, the log-likelihood is rescaled by the number of examples n.



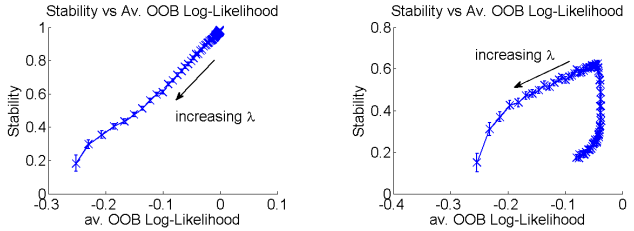
(a) No redundancy

(b) Low redundancy

(c) Medium redundancy

(d) High redundancy

Figure 6: Average OOB log-likelihood [left column] and stability [right column] against the regularizing parameter λ for 4 degrees of redundancy. For each degree of redundancy, the pink dashed-line corresponds to the λ value that maximizes the likelihood and the black one corresponds to the value of λ that maximizes stability. As we can see, by choosing latter parameter λ, we gain in stability with only a small loss in likelihood.



(a) No redundancy (b) Low redundancy

(c) Medium redundancy (d) High redundancy

Figure 7: Stability (with 95%-confidence intervals) against average OOB log-likelihood for 4 degrees of redundancy.

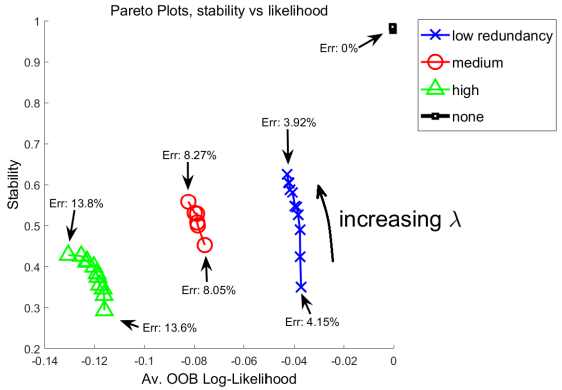
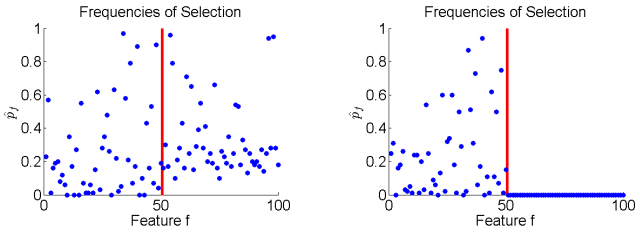


Figure 8: Summary of the pareto fronts for the 4 degrees of redundancy. The average OOB misclassification error is given for the two extreme points of each pareto front.



(a) $\lambda = 1.3 \times 10^{-2}$ (b) $\lambda = 6.8 \times 10^{-2}$

Figure 9: The observed frequencies of selection $\hat{p}_f$ of each feature optimizing only the likelihood [LEFT] and choosing a trade-off between stability and likelihood [RIGHT] for high redundancy ($\rho = 0.8$). The features on the left of the red vertical line correspond to the 50 relevant features and the ones on the right to the 50 irrelevant ones.

false positive rate 0), which means they have not been selected on any of the $M = 100$ samples. This is not the case when only maximizing the likelihood: we cannot discriminate the set of relevant features from the set of irrelevant ones by looking at the frequencies of selection $\hat{p}_f$. Using the 1000 holdout set, we applied $L1$-regularised logistic regression using these two values of λ. Table 5 provides the false positives and false negatives for the 4 degrees of redundancy. The results show a decrease in the false positives when optimizing both stability and likelihood while having a limited effect on the false negatives.

Redundancy	Optimizing likelihood	Optimizing both
none	$FP = 0$, $FN = 0$	IDEM
low	$FP = 14$, $FN = 19$	$FP = 0$, $FN = 20$
medium	$FP = 0$, $FN = 26$	$FP = 0$, $FN = 26$
high	$FP = 12$, $FN = 38$	$FP = 0$, $FN = 39$

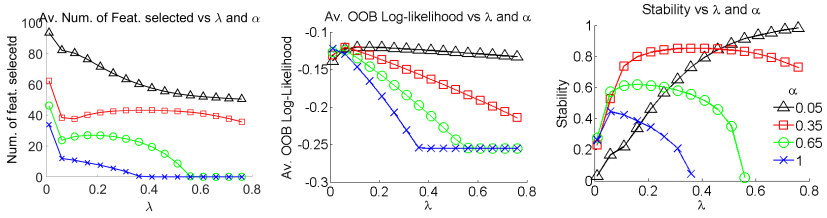
Table 5: False positives and false negatives of the final feature set for different degrees of redundancy ρ when optimizing only the likelihood against when optimizing both likelihood and stability.

6.1.3 Stability of the Elastic Net

$L1$-regularization has the effect of forcing regression coefficients to zero, hence selecting a subset of the available features. $L2$-regularization (*ridge regression*), is known to have a grouping effect: correlated features will have similar coefficients (Zhou, 2013). The Elastic Net is a convex combination of a $L1$ and a $L2$-regularization. It has two parameters, α and

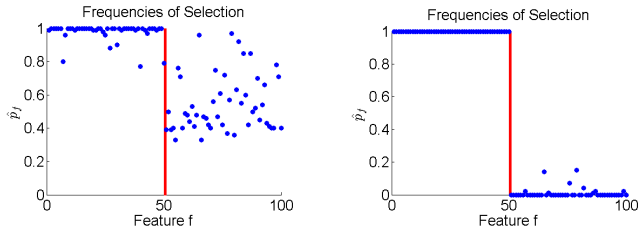
λ, where λ controls the overall weight of regularization and where α controls the balance between the two regularizing terms. When $\alpha = 1$, it becomes $L1$ (LASSO), and when $\alpha = 0$ it is $L2$ (ridge regression). As α varies, the Elastic net blends between the two and offers the advantages of both techniques—it forces some of the coefficients to be zero like LASSO while having the grouping effect of the ridge regression. Correlated features are a source of instability (Gulgezen et al., 2009; Wald et al., 2013), as feature selection procedures will tend to select a different feature from a same group of correlated features on different samples. Therefore, we expect the Elastic net to mitigate this, hence increasing stability.

In this section, we reproduce some of the experiments of the last section for the Elastic Net, optimizing the two regularizing parameters α and λ. We proceed as before, taking $M = 100$ bootstraps, and focus on the most challenging case of high redundancy ($\rho = 0.8$). We confirm in Figure 10a that as λ increases, the overall regularization increases and less features are selected. Figures 10b and 10c respectively give the average OOB log-likelihood and the stability against the values of λ for different values of α. We can see that no matter what is the value of λ chosen, $\alpha = 0.05$ has a higher likelihood than the other values of α in most cases and reaches high stability (greater than 0.90) for values of λ greater than 0.56. Interestingly, for these values of α and λ, we can see in Figure 10a that the number of features selected is around 50, which is the total number of relevant features. Let us have a closer look at $\alpha = 0.05$. If we were only optimizing the likelihood, we would pick $\lambda = 0.05$, which yields a stability of 0.34. The corresponding average misclassification error is 14%. If we wanted to also optimize stability, we could sacrifice a small amount of likelihood by picking $\lambda = 0.76$ which yields a stability of 0.98. The corresponding average OOB misclassification error is also 14%. Figure 11 gives the observed frequencies of selection $\hat{p}_f$ for $\lambda = 0.16$ on the left sub-figure (which is the value of λ that maximizes the likelihood) and for $\lambda = 0.76$ on the right sub-figure (which is a value of λ optimizing both the likelihood and the stability). We can see on the right sub-figure that when also optimizing stability, the whole set of relevant features is selected on each one of the $M = 100$ samples and irrelevant features are only rarely selected. On the left sub-figure, even though the likelihood for the given hyperparameters and the misclassification error are similar, we can see that non-relevant features are a lot more often selected in the model.



(a) Av. num. of feat. selected $\bar{k}$ (b) Likelihood (c) Stability

Figure 10: Plots against λ where each line corresponds to a different value of α in the high redundancy case ($\rho = 0.8$). We pick the $\alpha = 0.05$ as it reaches a higher likelihood for most values of λ and it can also achieve high stability.



(a) $\alpha = 0.05$ and $\lambda = 0.16$ (b) $\alpha = 0.05$ and $\lambda = 0.76$

Figure 11: The observed frequencies of selection $\hat{p}_f$ of each feature optimizing only the likelihood [LEFT] and choosing a trade-off between stability and likelihood [RIGHT] for high redundancy ($\rho = 0.8$). The features on the left of the red vertical line correspond to the 50 relevant features and the ones on the right to the 50 irrelevant ones. We can see that optimizing both stability and likelihood [RIGHT] helps recovering the set of relevant features.

6.1.4 Conclusions

In this section, we proposed a methodology to select hyperparameters using stability along with the error. On the data set used, we showed that it is possible to select a hyperparameter yielding much higher stability values without loss of predictive power. When the stability is optimized along with the likelihood of the model, the false positive rate was lower. Using the Elastic Net, we were able to achieve high stability for a similar loss than using LASSO. Enforcing high stability had the effect of discriminating the set of relevant features, helping to *recover* the true set of relevant features.

6.2 How Stable is Stability Selection?

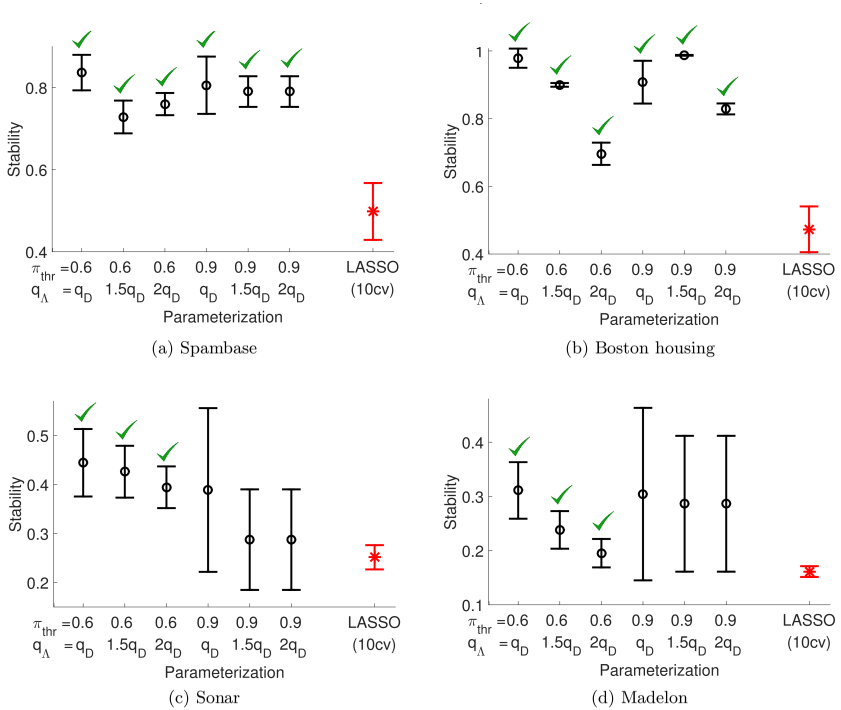
In high dimensional data sets, picking a regularizing parameter λ that recovers the *true* set of relevant features has proven to be challenging. For this reason, Meinshausen and Bühlmann (2010) introduced a technique called "*Stability Selection*", a popular and generic approach that can also be used for solving other problems of structure estimation such as graphical modelling. In this section, we focus on the use of Stability Selection in the context of feature selection with LASSO. It proposes to apply LASSO to M random sub-samples of size $\lfloor \frac{n}{2} \rfloor$ of the original data set (where n is the sample size) for a set of regularizing parameters $\lambda \in \Lambda$, where Λ is a subset of $\mathbb{R}^+$. This method considers the frequencies of selection of each feature $\hat{p}_f$ for each value of $\lambda \in \Lambda$ and defines the **set of stable variables** as *the set of all variables having a frequency of selection $\hat{p}_f \geq \pi_{thr}$ for at least one of the regularizing parameters $\lambda \in \Lambda$* (where π_{th} is a user-defined threshold). Then they propose to use the identified stable set as an approximation of the true relevant set.

The proposed technique is effectively an ensemble feature selection technique where the final set is made of all features having a high frequency of selection $\hat{p}_f$ for at least

one the chosen regularizing parameter. The main contribution of Stability Selection is that it provides a control over false discovery error rates (i.e. the number of irrelevant features identified as relevant), and as a result, a principled way to choose the amount of regularization for variable selection. In relation to our work, we can point out the following interesting facts about this work: (1) it uses the concept of frequency of selection $\hat{p}_f$ to detect relevant features; (2) it uses the underlying idea that the *stable set* does not only help recovering the true feature set but also will be more robust to the choice of regularizing parameters; (3) It provides an upper bound and an exact control of the number of false positives (i.e. the number of irrelevant features falsely selected). Therefore, this work implicitly uses the idea that selecting stable features in the final set will help recover the *true* set of relevant features, which is intimately linked to the results of the previous section where we have shown that enforcing stability could potentially reduce the number of false positives. Intuitively, the final set of variables picked by Stability Selection should be more stable in the sense of our definition $\hat{\Phi}(\mathcal{Z})$, as they select the variables showing a consensus across multiple repeats of the data with perturbations and for different regularizing parameters.

In this section, we use our proposed measure to *quantify* just how stable their *stable set* can be. To that end, we look at how much the final set picked by Stability Selection varies in the context of LASSO and we will show on 4 data sets that it will indeed yield more stable results (in the sense of $\hat{\Phi}(\mathcal{Z})$) than its non-ensemble version (LASSO). Nevertheless, we remind the reader that these experiments are purely illustrative of the concepts discussed in the paper and do not claim to be an exhaustive empirical study. Stability selection possesses 3 hyperparameters[12]: (1) the cut-off value π_{thr}, (2) the average number of features selected q_Λ over the all values of $\lambda \in \Lambda$ and (3) the set of regularizing parameters Λ (where the two last hyperparameters are dependent). We used the values suggested by the original authors: that is $\pi_{thr} \in (0.6 - 0.9)$ and q_Λ around $\sqrt{0.8d}$. Figure 12 compares the two approaches for variable selection in 4 data sets, three binary classification problems (Spambase/Sonar/Madelon) and one regression (Boston housing). To derive the 95%-interval estimate of stability, we ran the algorithms on $M = 100$ bootstraps, and used the tools presented in Section 4.2.3. The first observation is that, no matter the parametrisation, the average stability of stability selection is always higher than the stability of LASSO. Furthermore, we performed hypothesis tests at a level of significance of 5% to check for which hyperparameters Stability Selection achieved higher stability than LASSO. A green tick indicates where the null hypothesis (equal stability) was rejected. On the Sonar and Madelon data sets, the stability of Stability Selection is consistently higher than LASSO, but can present high variability for some hyperparameters (as shown by the large confidence intervals). In those cases, it failed to reject the null hypothesis. These experiments highlight the importance of statistical significance when quantifying stability and the need for such statistical tools.

12. We note that the first two hyperparameters listed hereafter effectively control the upper bound on the amount of false positives.



(a) Spambase (b) Boston housing

(c) Sonar (d) Madelon

Figure 12: Comparing the stability of LASSO and different parametrisations of Stability Selection in four classification/regression data sets. For LASSO (red star), we optimised the regularisation parameter using 10-fold cross validation and the *one-standard-error* rule—picking up the most parsimonious model within one standard error of the minimum (Hastie et al., 2009). For stability selection (black circle), we explored different parameters: for the cut-off threshold $\pi_{\text{thr}} \in \{0.60, 0.90\}$, while for the average number of selected variable $q_\Lambda \in \{q_D, 1.5q_D, 2q_D\}$, where $q_D = \sqrt{0.8d}$ is the default value. We performed hypothesis tests to check whether the stability of Stability Selection is significantly different from LASSO. The green tick means that the null hypothesis (i.e. the population stabilities are equal) has been rejected at level of significance 5%.

7. Conclusions and Future Work

In this section, we present the conclusions and then consider possible areas of future work.

7.1 Conclusions

We have provided a rigorous statistical treatment for the concept of stability in feature selection. Following a property-based approach suggested in previous works, we identified a set of 5 properties—and argued for them as desirable in most (if not all) scenarios. Then, we compared all existing stability measures in terms of properties. It emerged that no existing measure satisfied all desirable properties—to counter this, we constructed a novel measure, Definition 4. This turns out to be a generalization of several existing measures, and possesses all 5 properties. In addition, it provides new capabilities, not previously possible in the literature. We provided confidence intervals and hypothesis testing of *true* stability, and finally, we showed how these tools could be used to choose hyperparameters. An important conclusion is that optimizing stability can be potentially achieved without significant loss of accuracy, and can help identifying the *true* underlying set of features.

7.2 Future Work

In some data scenarios, the user might not want to measure the instability due to the redundancy of the features. In that scenario, the user is more interested in knowing whether features belonging to a same group of correlated features have been consistently selected, rather than looking at the selection of each feature independently. For this purpose, stability measures taking into account feature redundancy have been proposed in the literature. The relative POG (also called $POGR$) and the relative $nPOG$ (also called $nPOGR$) are both extensions of the POG measure and of the $nPOG$ measure respectively (Zhang et al., 2009) as they both reduce to their original version in the case of no redundancy. This case study is not in the scope of this paper. Nevertheless, we note that since they reduce to POG and $nPOG$, these two measures will also not possess the 5 properties. Future work might consider extending the measure we propose to take redundancy into account.

Another avenue of investigation could be the extension of the present work to other types of feature selection outputs—such as feature rankings or feature weights. A popular measure to quantify the stability of feature rankings is the pairwise Spearman's Rho. In the case of untied ranks, Nogueira et al. (2017) show that this measure can be re-written as

$$1 - \frac{\frac{1}{d}\sum_{f=1}^{d} s_f^2}{V_{rank}},$$

where s_f^2 is the sample variance of the rank of feature X_f and where V_{rank} is the expected variance under the assumption that each ranking was generated at random (i.e. each ranking is equally likely). The above equation has a similar form to the measure proposed in this paper and provides a promising direction for unifying stability of ranking and subset selection.

Acknowledgments

This research was conducted with support from the Centre for Doctoral Training (CDT) in Computer Science, funded by Engineering and Physical Sciences Research Council (EPSRC) grant [EP/I028099/1]. GB was supported by the EPSRC LAMBDA project [EP/N035127/1]. We would like to thank Dr. Adam Pocock for his valuable feedback on our work.

Appendix A. Existing Stability Measures

In the remainder of the appendices, we will add a subscript to the stability $\hat{\Phi}$ giving the author or the name of the measure for disambiguation. Whenever the subscript is omitted, we refer to our proposed stability measure as given by Definition 4.

In this section, we first review the existing similarity-based stability measures and then we focus on the frequency-based ones.

A.1 Similarity-based Measures

We remind the reader that given a similarity measure ϕ between two feature sets s_i and s_j, the resulting stability $\hat{\Phi}(\mathcal{Z})$ is taken as the average pairwise similarities between the feature sets in $\mathcal{Z}$, that is

$$\hat{\Phi}(\mathcal{Z}) = \frac{1}{M(M-1)} \sum_{i=1}^{M} \sum_{\substack{j=1 \\ j \neq i}}^{M} \phi(s_i, s_j).$$

For simplicity, we introduce notations that will be used in the remainder of the appendices. Let $r_{i,j}$ be the short notation for $|s_i \cap s_j|$, the size of the intersection between feature sets s_i and s_j. Let k_i denote the size of feature set s_i (when the size of the feature set is assumed constant, we will simply denote it k). Table 6 provides all 9 similarity measures used in the literature in the context of stability along with their minimum and maximum value. These definitions will be later used in the proof of properties in Appendix C.

A.2 Frequency-based Measures

In this section, we give the definitions of the frequency-based measures. As these require to introduce a lot of new notations, we do not summarize them in a table like we did for similarity-based measures.

A.2.1 Goh's Measure

Goh and Wong (2016) propose to use the frequency of selection, averaged over all features, that is $\hat{\Phi}_{Goh}(\mathcal{Z}) = \frac{1}{d}\sum_{f=1}^{d} \hat{p}_f$. This measure take values in $[0, 1]$.

First used in	**Name**	**Measure**	[min, max]
Dunne et al. (2002)	Hamming	$1 - \frac{\lvert s_i \setminus s_j \rvert + \lvert s_i \setminus s_j \rvert}{d}$	$[0, 1]$
Kalousis et al. (2005)	Jaccard	$\frac{r_{i,j}}{\lvert s_i \cup s_j \rvert}$	$[0, 1]$
Yu et al. (2008)	Dice-Sørenson	$\frac{2r_{i,j}}{k_i + k_j}$	$[0, 1]$
Zucknick et al. (2008)	Ochiai	$\frac{r_{i,j}}{\sqrt{k_i k_j}}$	$[0, 1]$
Shi et al. (2006)	POG	$\frac{r_{i,j}}{k_i}$	$[0, 1]$
Kuncheva (2007)	Consistency	$\frac{r_{i,j} - \frac{k^2}{d}}{k - \frac{k^2}{d}}$	$[-1, 1]$
Lustgarten et al. (2009)	Lustgarten	$\frac{r_{i,j} - \frac{k_i k_j}{d}}{min(k_i, k_j) - max(0, k_i + k_j - d)}$	$[-1, 1]$
Wald et al. (2013)	Wald	$\frac{r_{i,j} - \frac{k_i k_j}{d}}{min(k_i, k_j) - \frac{k_i k_j}{d}}$	$[1 - d, 1]$
Zhang et al. (2009)	nPOG	$\frac{r_{i,j} - \frac{k_i k_j}{d}}{k_i - \frac{k_i k_j}{d}}$	$[1 - d, 1]$

Table 6: Similarity measures proposed in the literature 2002–2018, using the pairwise formulation. In some cases the measure is extremely simple, (e.g. percentage overlap of features) and authors are chosen simply as the first known usage of the measures in the context of stability. We note that as opposed as what can be found in some literature (Alelyani, 2013; P. and Perumal, 2016), the minimum for Wald's and $nPOG$ similarity measures is equal to $1 - d$ and not 0 (and is reached for $k_i = 1$, $k_j = d - 1$ and $r_{i,j} = 0$).

A.2.2 Davis' Measure

Davis et al. (2006, pg2) penalize the frequency to *"account for the artificial increase in stability that occurs with increasingly long gene signatures"*, as follows

$$\hat{\Phi}_{Davis}(\mathcal{Z}) = \max\left(0,\ \frac{1}{F}\sum_{f=1}^{d}\hat{p}_f - \alpha\frac{median(k_1, ..., k_M)}{d}\right), \tag{3}$$

where F is the number of features selected at least in one of the M feature sets (in other words, $F = |\cup_{i\in\{1,...,M\}} s_i|$) and where α is a hyperparameter chosen by the user. This measure also takes values in the interval $[0, 1]$.

A.2.3 Krizek's Measure

Krízek et al. (2007) treat each possible of the $\binom{d}{k}$ feature sets of k features as a random variable and estimate its Shannon entropy as

$$\hat{\Phi}_{Krizek}(\mathcal{Z}) = -\sum_{s_i\in\mathcal{Z}}\hat{p}(s_i)\log_2\hat{p}(s_i),$$

where $\hat{p}(s_i)$ is the frequency of occurrence of subset s_i in $\mathcal{Z}$ over all the $\binom{d}{k}$ possible combinations of k features taken amongst d features. It takes values in $\left[0, \log(\min(M, \binom{d}{k}))\right]$.

A.2.4 Guzmán's Measure

A measure is proposed by Guzmán-Martínez and Alaiz-Rodríguez (2011), using frequencies to compute Jensen-Shannon divergences. Originally created for feature rankings, they extend it to feature sets of k features (top-k lists of genes in the literature), using the JS-divergence, as follows

$$\hat{\Phi}(\mathcal{Z}) = 1 - \frac{D_{JS}(\mathbf{q}_1, ..., \mathbf{q}_M)}{D^*_{JS}(\mathbf{q}_1, ..., \mathbf{q}_M)}.$$

Here, each $\mathbf{q}_i$ is a distribution over features, formed by taking the bitstring on the i^{th} row of the matrix $\mathcal{Z}$, and dividing through by k, the number of bits set. D^*_{JS} is a normalizing term—according to the authors *"the divergence value for a [feature set] that is completely random"*. This ensures the value is in $[0, 1]$, but most interestingly, this is yet another work correcting the measure for chance, as introduced by Kuncheva.

A.2.5 Relative Weighted Consistency CW_{rel}

With a property-based analysis, Somol and Novovičová (2010) constructed a new stability measure called the relative weighted consistency (CW_{rel}) as

$$\hat{\Phi}(\mathcal{Z}) = \frac{d\left(M\bar{k} - D + \sum_{f=1}^{d} M\hat{p}_f(M\hat{p}_f - 1)\right) - (M\bar{k})^2 + D^2}{d\left(H^2 + M(M\bar{k} - H) - D\right) - (M\bar{k})^2 + D^2}, \tag{4}$$

where $D = (M\bar{k})\ mod\ d$ and $H = (M\bar{k})\ mod\ M$.

A.2.6 Lausser's Measure

Finally, Lausser et al. (2013) proposed a measure for feature sets of fixed size k as follows:

$$\hat{\Phi}(\mathcal{Z}) = \frac{1}{M^2 k}\sum_{i=1}^{M} i^2 a^{(i)},$$

where $a^{(i)} = \sum_{f=1}^{d}\mathbb{1}\{\sum_{j=1}^{M} z_{j,f} = i\}$ is the number of features selected exactly i times.

Appendix B. Proof of Theorems

In this section, we provide the proofs or proof sketches for the theorems and corollaries in the paper. Before we proceed to the proofs, we give the following equation, that will be repeatedly used in the proofs

$$\sum_{f=1}^{d} \hat{p}_f = \frac{1}{M} \sum_{i=1}^{M} \sum_{f=1}^{d} z_{i,f} = \frac{1}{M} \sum_{i=1}^{M} k_i = \bar{k}. \tag{5}$$

B.1 Proof of Theorem 1

In this appendix we prove the following theorem from Section 3.

Theorem 1 *The average pairwise intersection between the M feature sets is as a linear function of the sample variances of the selection of each feature, as follows*

$$\frac{1}{M(M-1)} \sum_{i=1}^{M} \sum_{\substack{j=1 \\ j \neq i}}^{M} r_{i,j} = \bar{k} - \sum_{f=1}^{d} s_f^2, \tag{6}$$

where $s_f^2 = \frac{M}{M-1} \hat{p}_f (1 - \hat{p}_f)$ is the sample variance of the selection of the f^{th} feature.

Proof. To prove Theorem 1, we start by calculating the average pairwise size of the intersection and show that we get the results presented.

$$\begin{aligned}
\frac{1}{M(M-1)} \sum_{i=1}^{M} \sum_{\substack{j=1 \\ j \neq i}}^{M} r_{i,j} &= \frac{1}{M(M-1)} \sum_{i=1}^{M} \sum_{j=1}^{M} r_{i,j} - \frac{1}{M(M-1)} \sum_{i=1}^{M} r_{i,i} \\
&= \frac{1}{M(M-1)} \sum_{i=1}^{M} \sum_{j=1}^{M} r_{i,j} - \frac{1}{M(M-1)} \sum_{i=1}^{M} k_i \\
&= \frac{1}{M(M-1)} \sum_{i=1}^{M} \sum_{j=1}^{M} r_{i,j} - \frac{1}{M-1} \bar{k}
\end{aligned}$$

Since the i^{th} feature set $\mathcal{Z}_{(i,:)}$ and the j^{th} feature set $\mathcal{Z}_{(j,:)}$ are binary vectors, the size of their intersection $r_{i,j}$ is the number of 1s occurring at the same position in both vectors. In other words, $r_{i,j}$ is the dot product of the two feature sets, that is $\mathcal{Z}_{(i,:)} \cdot \mathcal{Z}_{(j,:)} = \sum_{f=1}^{d} z_{i,f} z_{j,f}$. By substituting in the previous equation, we get

$$\begin{aligned}
\frac{1}{M(M-1)} \sum_{i=1}^{M} \sum_{\substack{j=1 \\ j \neq i}}^{M} r_{i,j} &= \frac{1}{M(M-1)} \sum_{i=1}^{M} \sum_{j=1}^{M} \sum_{f=1}^{d} z_{i,f} z_{j,f} - \frac{1}{M-1} \bar{k} \\
&= \frac{1}{M(M-1)} \sum_{f=1}^{d} \left(\sum_{i=1}^{M} z_{i,f} \right) \left(\sum_{j=1}^{M} z_{j,f} \right) - \frac{1}{M-1} \bar{k} \\
&= \frac{1}{M(M-1)} \sum_{f=1}^{d} \left(\sum_{i=1}^{M} z_{i,f} \right)^2 - \frac{1}{M-1} \bar{k}
\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{M(M-1)} \sum_{f=1}^{d} (M \hat{p}_f)^2 - \frac{1}{M-1} \bar{k} \\
&= \frac{M}{M-1} \sum_{f=1}^{d} \left(\hat{p}_f^2 - \hat{p}_f + \hat{p}_f \right) - \frac{1}{M-1} \bar{k} \\
&= \frac{M}{M-1} \sum_{f=1}^{d} -\hat{p}_f (1 - \hat{p}_f) + \frac{M}{M-1} \sum_{f=1}^{d} \hat{p}_f - \frac{1}{M-1} \bar{k} \\
&= -\frac{M}{M-1} \sum_{f=1}^{d} \hat{p}_f (1 - \hat{p}_f) + \frac{M}{M-1} \bar{k} - \frac{1}{M-1} \bar{k} \\
&= \bar{k} - \sum_{f=1}^{d} s_f^2.
\end{aligned}$$

■

B.2 Proof of Theorem 3

In this appendix we prove the following theorem from Section 4.1.

Theorem 3 *Under the Null Model of Feature Selection H_0, for all f, the expected value of the sample variance of Z_f is $\mathbb{E}\left[s_f^2 | H_0\right] = \frac{\bar{k}}{d}\left(1 - \frac{\bar{k}}{d}\right)$.*

Proof. Since s_f^2 is the unbiased sample variance of the Bernoulli variable Z_f, we have $\mathbb{E}\left[s_f^2 | H_0\right] = Var[Z_f | H_0]$, which is the value of $p_f(1 - p_f)$ under the Null Model of Feature Selection H_0.

To calculate p_f under H_0, we will start by carrying out the calculations on a simple example to clarify what is the sample space we are looking at under H_0. Let us assume we observe the matrix

$$\mathcal{Z} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}.$$

By definition, under H_0, for each row i, all permutations of the bit-string on row i are equally likely. For the first row, we have 3 possible permutations that are 100, 010, 001 and for the second row, we also have 3 possible permutations that are 110, 101, 011. Therefore, under H_0, all the $N = 3 \times 3 = 9$ following matrices are equally likely to be observed:

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix},$$
$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix},$$
$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \quad \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}.$$

Since all these matrices are equally likely to be observed under H_0, to calculate p_f, we can simply count the proportion of times feature Z_f equals 1 over the $MN = 2 \times 9 = 18$ rows. Here we would get $p_f = \frac{9}{18} = \frac{1}{2}$ for any f.

Let N be the total number of possible matrices under H_0. In the more general case, to compute p_f under H_0, we can count the proportion of times $Z_f = 1$ over the MN rows. We can decompose this into a sum over the row numbers as follows:

$$p_f = \Pr[Z_f = 1|H_0] = \frac{\sum_{i=1}^{M} N \times \Pr[Z_f = 1|\text{on row i }, H_0]}{MN}$$

$$p_f = \frac{1}{M}\sum_{i=1}^{M} \Pr[Z_f = 1|\text{on row i }, H_0], \tag{7}$$

where $\Pr[Z_f = 1|\text{on row i }, H_0]$ is the proportion of times Z_f is equal to 1 on all the permutations of the i^{th} row of $\mathcal{Z}$. Since the i^{th} row of $\mathcal{Z}$ has k_i bits set to 1, this is equal to

$$\frac{\#\{\text{bit-strings with } k_i \text{ 1s where } Z_f = 1\}}{\#\{\text{bit-strings with } k_i \text{ 1s}\}}.$$

The denominator is equal to $\binom{d}{k_i}$. For the numerator, we know X_f is set equal to 1, which means we have now $k_i - 1$ bits left to set to 1 from the remaining $d-1$ bits. Therefore the numerator is equal to $\binom{d-1}{k_i-1}$. Replacing these in the previous equation, we get that on the i^{th} row, all features have an equal probability of being selected equal to $\frac{\binom{d-1}{k_i-1}}{\binom{d}{k_i}} = \frac{k_i}{d}$. Now, replacing this in Equation (7), we get that under H_0,

$$p_f = \frac{1}{M}\sum_{i=1}^{M}\frac{k_i}{d} = \frac{\bar{k}}{d}.$$

We can verify this in our previous example, where we had $k_1 = 2$, $k_2 = 1$, $d = 3$, which gives $p_f = \frac{\bar{k}}{d} = \frac{1}{2}$ as computed previously.

Using this, we can now compute $\mathbb{E}\left[s_f^2|H_0\right] = Var[Z_f|H_0] = p_f(1-p_f) = \frac{\bar{k}}{d}\left(1-\frac{\bar{k}}{d}\right)$. ■

B.3 Proof of Theorem 5

In this appendix we prove the following theorem from Section 4.1.

Theorem 5 *When the number of features selected is constant:*

- *The stability estimator $\hat{\Phi}$ is equal to the stability measures derived by Kuncheva (2007), Wald et al. (2013) and to nPOG (Zhang et al., 2009).*
- *The stability estimator $\hat{\Phi}$ and CW_{rel} (Somol and Novovičová, 2010) are asymptotically equivalent.*

Proof. First, we show that when the number of features selected is constant equal to k, then Kuncheva's measure is equal to the proposed stability measure. The stability measure

defined by Kuncheva (2007) is

$$\hat{\Phi}_{Kuncheva}(\mathcal{Z}) = \frac{1}{M(M-1)}\sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M}\frac{r_{i,j}-\frac{k^2}{d}}{k-\frac{k^2}{d}} = \frac{1}{M(M-1)}\left(\sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M}\frac{r_{i,j}}{k-\frac{k^2}{d}}\right) - \frac{\frac{k^2}{d}}{k-\frac{k^2}{d}}$$

$$= \frac{1}{k-\frac{k^2}{d}}\left(\frac{1}{M(M-1)}\sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M} r_{i,j}\right) - \frac{\frac{k^2}{d}}{k-\frac{k^2}{d}}.$$

Using Theorem 1, we can replace the term between parenthesis in the latter equation by $k - \sum_{f=1}^{d} s_f^2$ (since the number of features selected is constant, $\bar{k} = k$). We get that

$$\hat{\Phi}_{Kuncheva}(\mathcal{Z}) = \frac{1}{k-\frac{k^2}{d}}\left(k - \sum_{f=1}^{d} s_f^2\right) - \frac{\frac{k^2}{d}}{k-\frac{k^2}{d}} = \frac{k-\frac{k^2}{d}}{k-\frac{k^2}{d}} - \frac{\sum_{f=1}^{d} s_f^2}{k-\frac{k^2}{d}} = 1 - \frac{\frac{1}{d}\sum_{f=1}^{d} s_f^2}{\frac{\bar{k}}{d}\left(1-\frac{\bar{k}}{d}\right)},$$

which is our proposed stability measure for $\bar{k} = k$. Then, using the definitions of Wald and $nPOG$ measures given in Table 6, by replacing the cardinalities of the sets by k in the equations, they reduce to Kuncheva's measure, which proves the first part of the theorem.

We now prove the equivalence of our measure with CW_{rel}. Using Equation (4), we can rewrite CW_{rel} as

$$\hat{\Phi}_{CW_{rel}}(\mathcal{Z}) = \frac{-\frac{1}{d}\frac{M-1}{M}\sum_{f=1}^{d} s_f^2 + \frac{\bar{k}}{d}\left(1-\frac{\bar{k}}{d}\right) - \frac{D}{M^2 d}\left(1-\frac{D}{d}\right)}{\frac{\bar{k}}{d}\left(1-\frac{\bar{k}}{d}\right) - \frac{D}{M^2 d}\left(1-\frac{D}{d}\right) + \frac{H^2}{dM^2} - \frac{H}{Md}}. \tag{8}$$

Assuming that the number of features selected is constant equal to k, we then have that $\bar{k} = k$ and hence that $H = (Mk)\ mod\ M = 0$. Therefore the above equation becomes

$$\hat{\Phi}_{CW_{rel}}(\mathcal{Z}) = 1 - \frac{\frac{M-1}{M}\sum_{f=1}^{d} s_f^2}{k\left(1-\frac{k}{d}\right) - \frac{D}{M^2}\left(1-\frac{D}{d}\right)}.$$

We have that $D = (Mk)\ mod\ d$ which implies that D is a constant number between 0 and $d-1$. Therefore the limit of the term $\frac{D}{M^2}\left(1-\frac{D}{d}\right)$ as M approaches infinity is 0. Therefore, taking the limit of the above equation, we get

$$\lim_{M\to\infty}\hat{\Phi}_{CW_{rel}}(\mathcal{Z}) = \lim_{M\to\infty}\left[1 - \frac{\frac{M-1}{M}\sum_{f=1}^{d} s_f^2}{k\left(1-\frac{k}{d}\right)}\right] = \lim_{M\to\infty}\hat{\Phi}(\mathcal{Z}).$$

This shows that $\lim_{M\to+\infty}\frac{\hat{\Phi}_{CW_{rel}}(\mathcal{Z})}{\hat{\Phi}(\mathcal{Z})} = 1$ and therefore we obtain that $\hat{\Phi}_{CW_{rel}}(\mathcal{Z}) \underset{M\to+\infty}{\sim} \hat{\Phi}(\mathcal{Z})$, which is what we wanted to prove. ■

B.4 Proof of Theorem 6

In this appendix we prove the following theorem from Section 4.2.1.

Theorem 6 *When there are only two categories (0/1), Fleiss' Kappa is equal to $\hat{\Phi}(\mathcal{Z})$.*

Proof. To prove this, we start from the definition of Fleiss' Kappa as given in the original paper (Fleiss, 1971) and show that when the number of categories is equal to 2, it reduces to the proposed definition of stability (c.f. Definition 4). Fleiss (1971) defines Kappa as

$$\kappa = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e}, \tag{9}$$

where:

- $\bar{P}_e = \sum_{j=1}^{q} p_j^2$ in which
  - $q = 2$ is the number of categories;
  - $p_j = \frac{1}{Md} \sum_{f=1}^{d} n_{fj}$;
  - n_{fj} is the number of samples that assign f to the j^{th} category. Therefore, in our case, we have $n_{f1} = M\hat{p}_f$ and $n_{f0} = M - M\hat{p}_f$.
- $\bar{P} = \frac{1}{dM(M-1)} \sum_{f=1}^{d} \sum_{j=1}^{q} n_{fj}(n_{fj} - 1)$.

Now we can re-write this using our notation. First, we have that

- $p_1 = \frac{1}{Md} \sum_{f=1}^{d} n_{f1} = \frac{1}{Md} \sum_{f=1}^{d} M\hat{p}_f = \frac{1}{d} \sum_{f=1}^{d} \hat{p}_f = \frac{\bar{k}}{d}$;
- $p_0 = \frac{1}{Md} \sum_{f=1}^{d} n_{f0} = \frac{1}{Md} \sum_{f=1}^{d} (M - M\hat{p}_f) = 1 - \frac{1}{d} \sum_{f=1}^{d} \hat{p}_f = 1 - \frac{\bar{k}}{d}$.

Therefore,

$$\bar{P}_e = p_0^2 + p_1^2 = \frac{\bar{k}^2}{d^2} + \left(1 - \frac{\bar{k}}{d}\right)^2 = \frac{\bar{k}^2}{d^2} + 1 - 2\frac{\bar{k}}{d} + \frac{\bar{k}^2}{d^2} = 1 - 2\frac{\bar{k}}{d}\left(1 - \frac{\bar{k}}{d}\right).$$

Let us now calculate $\bar{P}$.

$$\bar{P} = \frac{1}{dM(M-1)} \sum_{f=1}^{d} \sum_{j=1}^{q} n_{fj}(n_{fj} - 1)$$

$$\bar{P} = \frac{1}{dM(M-1)} \sum_{f=1}^{d} (n_{f0}(n_{f0} - 1) + n_{f1}(n_{f1} - 1))$$

$$\bar{P} = \frac{1}{dM(M-1)} \sum_{f=1}^{d} ((M - M\hat{p}_f)(M - M\hat{p}_f - 1) + M\hat{p}_f(M\hat{p}_f - 1))$$

$$\bar{P} = 1 - \frac{2}{d} \sum_{f=1}^{d} \frac{M}{M-1} \hat{p}_f(1 - \hat{p}_f) = 1 - \frac{2}{d} \sum_{f=1}^{d} s_f^2.$$

Now, substituting the two last equations back into Equation (9), we finally get that

$$\kappa = \frac{1 - \frac{2}{d}\sum_{f=1}^{d} s_f^2 - 1 + 2\frac{\bar{k}}{d}\left(1 - \frac{\bar{k}}{d}\right)}{1 - 1 + 2\frac{\bar{k}}{d}\left(1 - \frac{\bar{k}}{d}\right)} = \frac{-\frac{1}{d}\sum_{f=1}^{d} s_f^2 + \frac{\bar{k}}{d}\left(1 - \frac{\bar{k}}{d}\right)}{\frac{\bar{k}}{d}\left(1 - \frac{\bar{k}}{d}\right)}$$

$$\kappa = 1 - \frac{\frac{1}{d}\sum_{f=1}^{d} s_f^2}{\frac{\bar{k}}{d}\left(1 - \frac{\bar{k}}{d}\right)} = \hat{\Phi}(\mathcal{Z}),$$

which is what we wanted to prove. ■

B.5 Proof of Theorem 7

Theorem 7 (Asymptotic Distribution) *As $M \to \infty$, the statistic $\hat{\Phi}$ weakly converges to a normal distribution:*

$$V_M = \frac{\hat{\Phi} - \Phi}{\sqrt{v(\hat{\Phi})}} \xrightarrow{\mathcal{L}} \mathcal{N}(0, 1),$$

where:

- $\Phi = 1 - \frac{\frac{1}{d}\sum_{f=1}^{d} p_f(1-p_f)}{\bar{p}(1-\bar{p})}$ *is the mean of the estimator $\hat{\Phi}$ in which $\bar{p} = \frac{1}{d}\sum_{f=1}^{d} p_f$ is the average mean parameter of the d Bernoulli variables;*
- $v(\hat{\Phi}) = \frac{4}{M^2}\sum_{i=1}^{M}(\hat{\Phi}_{(i)} - \hat{\bar{\Phi}}_{(.)})^2$ *is an estimate of the variance of $\hat{\Phi}$ in which:*
  - $\hat{\Phi}_{(i)} = \frac{1}{\frac{\bar{k}}{d}\left(1-\frac{\bar{k}}{d}\right)}\left[\frac{1}{d}\sum_{f=1}^{d} z_{i,f}\hat{p}_f - \frac{k_i\bar{k}}{d^2} + \frac{\hat{\Phi}}{2}\left(\frac{2\bar{k}k_i}{d^2} - \frac{k_i}{d} - \frac{\bar{k}}{d} + 1\right)\right]$;
  - *and $\hat{\bar{\Phi}}_{(.)}$ is the average value of $\hat{\Phi}_{(i)}$, that is: $\frac{1}{M}\sum_{i=1}^{M}\hat{\Phi}_{(i)}$.*

Proof. We prove this using Theorem 6 showing the equality between our proposed measure and Fleiss' Kappa and the work of Gwet (2008) that derives the asymptotic distribution of Fleiss' Kappa, as explained in the first paragraph of Section 4.2.2. ■

B.6 Proof of Theorem 9

In this appendix we prove the following theorem from Section 4.2.4.

Theorem 9 *The test statistic for comparing stabilities is*

$$T_M = \frac{\hat{\Phi}(\mathcal{Z}_2) - \hat{\Phi}(\mathcal{Z}_1)}{\sqrt{v(\hat{\Phi}(\mathcal{Z}_1)) + v(\hat{\Phi}(\mathcal{Z}_2))}}.$$

Under H_0, the statistic T_M asymptotically (as M approaches infinity) follows a standard normal distribution.

Proof. We know from Theorem 7 that $\hat{\Phi}(\mathcal{Z}_1)$ is asymptotically normal with unknown mean Φ_1 and variance σ_1^2 and that $\hat{\Phi}(\mathcal{Z}_2)$ is asymptotically normal with unknown mean Φ_2 and variance σ_2^2. Therefore, the difference $\hat{\Phi}(\mathcal{Z}_2) - \hat{\Phi}(\mathcal{Z}_1)$ is normal with unknown mean $\Phi_2 - \Phi_1$ and with variance $\sigma_1^2 + \sigma_2^2$. Under H_0, $\Phi_2 - \Phi_1 = 0$ and we estimate this variance by $v(\hat{\Phi}(\mathcal{Z}_1)) + v(\hat{\Phi}(\mathcal{Z}_2))$ using the result of Theorem 7. This gives us the asymptotic distribution of the statistic T_M. ■

Appendix C. Proof of Properties

In this section, for each one of the 5 properties given in Section 3, we determine which measures possess the property.

C.1 First property: Fully defined

This property directly follows from the definitions of the stability measures given Appendix A. Kuncheva's, Krízek's, Guzmán's and Lausser's measures are only defined when the number of features selected is fixed, and therefore do not possess this property.

C.2 Second property: Monotonicity

Since the proofs will all be similar for similarity-based measures, we first provide the proofs for the similarity based measures and then we look at frequency-based ones.

C.2.1 Similarity-based Measures

We start by calculating the derivative for each one of the 9 the similarity measures and provide the results in Table 7. As we can see, for all 9 similarity measures, assuming that the cardinalities of the feature sets are always in $\{1,..,d-1\}$, we have that $\frac{d\phi(s_i,s_j)}{dr_{i,j}} > 0$ (some derivatives are undefined otherwise, which correspond to the limit cases where no features are selected or all the features are selected). Therefore the derivative of the stability measure $\hat{\Phi}(\mathcal{Z})$ will be positive since

$$\frac{d\hat{\Phi}(\mathcal{Z})}{d\left(\sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M} r_{i,j}\right)} = \frac{1}{M(M-1)}\sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M}\frac{\partial\phi(s_i,s_j)}{\partial r_{i,j}}$$

and since a sum of strictly positive quantities is strictly positive. Therefore, all similarity-based stability measures have the Monotonicity property.

	Hamming	Jaccard	Dice	Ochiai	*POG*
$\frac{d\phi(s_i,s_j)}{dr_{i,j}}$	$\frac{2}{d}$	$\frac{k_i+k_j}{(k_i+k_j-r_{i,j})^2}$	$\frac{2}{k_i+k_j}$	$\frac{1}{\sqrt{k_i k_j}}$	$\frac{1}{k_i}$

	Kuncheva	Lustgarten	Wald	*nPOG*
$\frac{d\phi(s_i,s_j)}{dr_{i,j}}$	$\frac{1}{k-\frac{k^2}{d}}$	$\frac{1}{\min(k_i,k_j)-\max(0,k_i+k_j-d)}$	$\frac{1}{\min(k_i,k_j)-\frac{k_i k_j}{d}}$	$\frac{1}{k_i-\frac{k_i k_j}{d}}$

Table 7: Derivatives for each one of the similarity measures.

C.2.2 Goh's Measure

Using the definition of Goh's measure (c.f. Appendix A.2) and Equation (5), we have that

$$\hat{\Phi}_{Goh}(\mathcal{Z}) = \frac{1}{d}\sum_{f=1}^{d}\hat{p}_f = \frac{\bar{k}}{d}. \tag{10}$$

This is not a function of the variances of selection of each feature s_f^2, therefore the measure does not have the Monotonicity property.

C.2.3 Davis' Measure

When $\alpha = 0$, this measure reduces to Goh's measure, which does not possess the property. Therefore this measure does not possess the Monotonicity property either.

C.2.4 Krízek's Measure

To prove that this measure does not possess the Monotonicity property, we give a counter-example. Let us assume we have a procedure that selects $k = 2$ features out of $d = 4$ features in total. The two binary matrices $\mathcal{Z}_1$ and $\mathcal{Z}_2$ illustrate two different scenarios with $M = 4$ as follows

$$\mathcal{Z}_1 = \begin{bmatrix}1&1&0&0\\1&1&0&0\\0&0&1&1\\0&0&1&1\end{bmatrix} \quad \text{and} \quad \mathcal{Z}_2 = \begin{bmatrix}1&0&1&0\\1&0&1&0\\0&0&1&1\\0&0&1&1\end{bmatrix}.$$

Using Krízek's measure, we get a stability of 1 for both $\mathcal{Z}_1$ and $\mathcal{Z}_2$ (using log base 2). Now by computing the sum of variances of selection of the 4 features for the two test cases, we get $\sum_{f=1}^{4} s_f^2 = \frac{4}{3}$ for $\mathcal{Z}_1$ and $\sum_{f=1}^{4} s_f^2 = \frac{2}{3}$ for $\mathcal{Z}_2$. Therefore $\mathcal{Z}_1$ and $\mathcal{Z}_2$ have the same stability value but different sums of variance, which is a counter-example of the property.

C.2.5 Guzmán's Measure

Before proving this, we re-write Guzmán's measure using our notation (this re-writing will also be useful for later proofs). For feature sets of fixed cardinality k, the stability is defined by Guzmán-Martínez and Alaiz-Rodríguez (2011) as

$$\hat{\Phi}_{Guzman}(\mathcal{Z}) = 1 - \frac{D_{JS}(\mathbf{q}_1,...,\mathbf{q}_M)}{D^*_{JS}(\mathbf{q}_1,...,\mathbf{q}_M)},$$

where:

- $D^*_{JS}(\mathbf{q}_1,...,\mathbf{q}_M) = \log\frac{d}{k}$;
- $D_{JS}(\mathbf{q}_1,...,\mathbf{q}_M) = \frac{1}{M}\sum_{i=1}^{M}\sum_{f=1}^{d} q_{f,i}\log\frac{q_{f,i}}{\bar{q_f}}$;
- $q_{f,i} = \frac{1}{k}$ if the f^{th} feature is selected on the i^{th} run and 0 otherwise;
- $\bar{q_f} = \frac{1}{M}\sum_{i=1}^{M} q_{f,i}$.

Therefore, using our notation, we get that $q_{f,i} = z_{i,f}\frac{1}{k}$ and that $\bar{q_f} = \frac{1}{M}\frac{1}{k}\sum_{i=1}^{M} z_{i,f} = \frac{\hat{p}_f}{k}$. Therefore,

$$\begin{aligned}
\hat{\Phi}_{Guzman}(\mathcal{Z}) =& 1 - \frac{\frac{1}{kM}\sum_{i=1}^{M}\sum_{f=1}^{d} z_{i,f}\log\frac{z_{i,f}}{\hat{p}_f}}{\log\frac{d}{k}}\\
\hat{\Phi}_{Guzman}(\mathcal{Z}) =& 1 - \frac{\frac{1}{kM}\sum_{i=1}^{M}\sum_{f=1}^{d} z_{i,f}\log z_{i,f}}{\log\frac{d}{k}} + \frac{\frac{1}{kM}\sum_{i=1}^{M}\sum_{f=1}^{d} z_{i,f}\log\hat{p}_f}{\log\frac{d}{k}}\\
\hat{\Phi}_{Guzman}(\mathcal{Z}) =& 1 - \frac{\frac{1}{kM}\sum_{i=1}^{M}\sum_{f=1}^{d} z_{i,f}\log z_{i,f}}{\log\frac{d}{k}} + \frac{\frac{1}{k}\sum_{f=1}^{d}\hat{p}_f\log\hat{p}_f}{\log\frac{d}{k}}.
\end{aligned}$$

Since $z_{i,f}$ is binary, $z_{i,f}\log z_{i,f} = 0$. Therefore, the previous equation becomes

$$\hat{\Phi}_{Guzman}(\mathcal{Z}) = 1 + \frac{\frac{1}{k}\sum_{f=1}^{d}\hat{p}_f\log\hat{p}_f}{\log\frac{d}{k}} = 1 - \frac{\frac{1}{d}\sum_{f=1}^{d}\hat{p}_f\log\hat{p}_f}{\frac{k}{d}\log\frac{k}{d}}. \tag{11}$$

Now, to prove that this measure does not have the Monotonicity property, we will give a counter-example. Let $\mathcal{Z}_1$ and $\mathcal{Z}_2$ be the two following binary matrices

$$\mathcal{Z}_1 = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \qquad \text{and} \qquad \mathcal{Z}_2 = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}.$$

We show that using Guzmán's measure, $\mathcal{Z}_1$ has a lower stability than $\mathcal{Z}_2$ but also a lower average variance, thus violating the Monotonicity property. Indeed, we have $\hat{\Phi}_{Guzman}(\mathcal{Z}_1) \simeq 0.24$ and $\hat{\Phi}_{Guzman}(\mathcal{Z}_2) \simeq 0.31$ while we have a sum of variances $\simeq 0.15$ for $\mathcal{Z}_1$ and $\simeq 0.17$ for $\mathcal{Z}_2$.

C.2.6 Relative Weighted Consistency CW_{rel}

Since H, D and $\bar{k}$ only depend on the feature set cardinalities $k_1, ..., k_M$, on M and on d, we can see from Equation (8) that $\hat{\Phi}_{CW_{rel}}(\mathcal{Z})$ is a linear and strictly decreasing function of s_f^2. Therefore, CW_{rel} possesses the Monotonicity property.

C.2.7 Lausser's measure

Similarly to what has been done for Guzmán's measure, we will re-write this measure as a function of the frequencies of selection $\hat{p}_f$. This will help us understand the measure and also will be useful for other proofs involving this measure. We remind the reader that Lausser's measure is defined as

$$\hat{\Phi}_{Lausser}(\mathcal{Z}) = \frac{1}{M^2 k} \sum_{f=1}^{d} \sum_{i=1}^{M} i^2 \mathbb{1}\{\sum_{j=1}^{M} z_{j,f} = i\} = \frac{1}{M^2 k} \sum_{f=1}^{d} \left[\sum_{i=1}^{M} i^2 \mathbb{1}\{M\hat{p}_f = i\} \right].$$

Let us look at the term in between brackets that depends on the row index i. We note that the indicator term $\mathbb{1}\{M\hat{p}_f = i\}$ is equal to 1 only when i is equal to $M\hat{p}_f$ and is equal to 0 for any other value of i in $\{1, ..., M\}$. Therefore we can make the sum over i disappear since it is always equal to $(M\hat{p}_f)^2$. Hence, Lausser's Measure can be re-written as

$$\hat{\Phi}_{Lausser}(\mathcal{Z}) = \frac{1}{M^2 k} \sum_{f=1}^{d} (M\hat{p}_f)^2 = \frac{1}{k} \sum_{f=1}^{d} \hat{p}_f^2. \tag{12}$$

This simple expression helps us understand what is this measure actually measuring. Let us now show that this is a strictly decreasing function of the sum of variances $\sum_{f=1}^{d} \hat{p}_f(1-\hat{p}_f)$. We can re-write the sum of variances as follows

$$\frac{M-1}{M} \sum_{f=1}^{d} s_f^2 = \sum_{f=1}^{d} \hat{p}_f(1-\hat{p}_f) = \sum_{f=1}^{d} \hat{p}_f - \sum_{f=1}^{d} \hat{p}_f^2 = k - \sum_{f=1}^{d} \hat{p}_f^2 = k(1 - \hat{\Phi}_{Lausser}(\mathcal{Z})).$$

Therefore, Lausser's measure has the Monotonicity property.

C.3 Third property: Bounds

In this section, we verify which measures have the Bounds property as given by Table 1.

C.3.1 Similarity-based measures

If a similarity measure ϕ is bounded, i.e. if $\exists (a, b) \in \mathbb{R}^2, a \leq \phi \leq b$, then it follows that the corresponding stability measure will also be bounded. Indeed:

$$a \leq \phi \leq b \quad \Rightarrow \quad a \leq \frac{1}{M(M-1)} \sum_{i=1}^{M} \sum_{\substack{j=1 \\ j \neq i}}^{M} \phi(s_i, s_j) \leq b \quad \Rightarrow \quad a \leq \hat{\Phi}(\mathcal{Z}) \leq b.$$

As given in Table 6, we can see that all similarity measures except Wald's measure (Wald et al., 2013) and $nPOG$ measure (Zhang et al., 2009) are bounded. Therefore, we know that their corresponding stability measures will also be bounded.

The contrary is not necessarily true. If a similarity measure is not bounded, this does not imply that the corresponding stability measure is not bounded. Nevertheless, we prove that the stability measures using Wald's and $nPOG$ similarity measures are not bounded using a counter-example. Let us assume we have the following scenario

$$\mathcal{Z} = \begin{bmatrix} 1 & 1 & 1 & 1 & ... & 1 & 0 \\ & & \vdots & & & & \\ 1 & 1 & 1 & 1 & ... & 1 & 0 \\ 0 & 0 & 0 & 0 & ... & 0 & 1 \\ & & \vdots & & & & \\ 0 & 0 & 0 & 0 & ... & 0 & 1 \end{bmatrix},$$

where the first $\frac{M}{2}$ feature sets are all identical and select the first $d-1$ features and the $\frac{M}{2}$ following ones are also identical but only select the first feature. In this situation, using Wald's similarity, the first block will give $\frac{M}{2}\left(\frac{M}{2}-1\right)$ similarities of 1 (as all feature sets in the first block are identical), the second block of feature sets will also give $\frac{M}{2}\left(\frac{M}{2}-1\right)$ similarities of 1 as all feature sets in the second block are also identical. Then the $\frac{M^2}{2}$ remaining pairs of feature sets (coming from the inter block pairs) have an intersection $r_{i,j} = 0$ and therefore a similarity equal to

$$\frac{0 - \frac{d-1}{d}}{1 - \frac{d-1}{d}} = \frac{1-d}{d-d+1} = 1 - d.$$

So overall, the stability using Wald's similarity measure is equal to

$$\hat{\Phi}_{Wald}(\mathcal{Z}) = \frac{1}{M(M-1)} \left[2\frac{M}{2}\left(\frac{M}{2}-1\right) + \frac{M^2}{2}(1-d) \right] = \frac{\frac{M}{2}-1}{M-1} + \frac{M}{2(M-1)}(1-d).$$

We can see that that the value of the stability decreases with d. Therefore, we can conclude that Wald's stability measure is not bounded by constants. Using the same scenario, we can similarly show that the $nPOG$ measure is not bounded.

C.3.2 Frequency-based measures

The range of values of all the frequency-based measures are given in the literature and recapitulated in Appendix A.2. Krízek's measure has a maximum depending on M, d and k and therefore is not bounded. All five other frequency measures (CW_{rel}, Davis' and Goh's measures) take values in $[0, 1]$ and therefore are bounded.

C.4 Fourth Property: Maximum

In this section, we show which one of the stability measures possess the Maximum property, as given in Table 1.

C.4.1 Similarity-based Measures

For the backward implication (Deterministic Selection $\rightarrow$ Maximum Stability), let us assume that all the feature sets in $\mathcal{Z}$ are identical with cardinality k, therefore $|s_i \cap s_j| = r_{i,j} = k$. By definition, for all similarity measures given in Table 1 except Lustgarten's measure, for all $i, j \in \{1, ..., M\}$, $\phi(s_i, s_j) = 1$ which means that the average pairwise similarity is also 1. Therefore all similarity-based measures have this property except Lustgarten's measure (as it is shown with a counter-example in Figure 1b).

For the forward implication (Maximum Stability $\rightarrow$ Deterministic Selection), showing that Wald's stability measure does not have this property can easily be done with a counter-example as done in the paper (c.f. Figure 1a). All other similarity-based stability measures have a maximum equal to 1. Let us assume that $\hat{\Phi}(\mathcal{Z}) = max(\hat{\Phi}) = 1$. We want to show that this implies that all feature sets in $\mathcal{Z}$ are identical.

$$\begin{aligned}
\hat{\Phi}(\mathcal{Z}) = 1 \Rightarrow & \quad \frac{1}{M(M-1)} \sum_{i=1}^{M} \sum_{\substack{j=1 \\ j \neq i}}^{M} \phi(s_i, s_j) = 1 \\
\Rightarrow & \quad \sum_{i=1}^{M} \sum_{\substack{j=1 \\ j \neq i}}^{M} \phi(s_i, s_j) = M(M-1) \\
\Rightarrow & \quad \forall i \in \{0,1\}^d, \forall j \in \{0,1\}^d, j \neq i, \phi(s_i, s_j) = 1.
\end{aligned}$$

Then using the constraint that $r_{i,j}$ is a natural number less or equal than $\min(k_i, k_j)$ (since that is the maximal possible size of intersection between two sets of size k_i and k_j), it can be shown for Jaccard, Dice, *POG*, *nPOG* and Kuncheva, that this implies that $k_i = k_j = r_{i,j}$ which means that $s_i = s_j$.

C.4.2 Goh's Measure

Using Equation (10), we have that $\hat{\Phi}_{Goh}(\mathcal{Z}) = \frac{\bar{k}}{d}$. Therefore, when all feature sets in $\mathcal{Z}$ are identical, $\hat{\Phi}_{Goh}(\mathcal{Z})$ only reaches its maximal value of 1 if all features are selected (i.e., $\hat{p}_f = 1$ for all $f \in \{1, ..., d\}$). Therefore, this measure does not have the Maximum property.

C.4.3 Davis' Measure

Taking $\alpha = 0$, this measure is equal to Goh's measure (seen in the previous section). Therefore this stability measure does not have the Maximum property either.

C.4.4 Krízek's Measure

Let us show that the property is true for Krízek's stability measure. We note this measure is the only one for which lower values correspond to a higher stability and the maximum

stability is reached for a stability of 0.

$$\begin{aligned}
\hat{\Phi}_{Krizek}(\mathcal{Z}) = 0 \Leftrightarrow & \quad -\sum_{s_i \in \mathcal{Z}} \hat{p}(s_i) \log_2 \hat{p}(s_i) = 0 \\
\Leftrightarrow & \quad \forall j \in \{1, ..., \binom{d}{k}\}, \hat{p}(s_i) \log_2 \hat{p}(s_i) = 0 \\
\Leftrightarrow & \quad \forall j \in \{1, ..., \binom{d}{k}\}, \hat{p}(s_i) = 0 \text{ or } \hat{p}(s_i) = 1 \\
\Leftrightarrow & \quad \text{All feature sets in } \mathcal{Z} \text{ are identical.}
\end{aligned}$$

Therefore, Krízek's measure has the Maximum property.

C.4.5 Relative Weighted Consistency CW_{rel}

Using Equation (8), we have

$$\begin{aligned}
\hat{\Phi}_{CW_{rel}}(\mathcal{Z}) = 1 \quad \Leftrightarrow & \quad \frac{-\frac{1}{d}\frac{M-1}{M}\sum_{f=1}^{d} s_f^2 + \frac{\bar{k}}{d}\left(1 - \frac{\bar{k}}{d}\right) - \frac{D}{M^2 d}\left(1 - \frac{D}{d}\right)}{\frac{\bar{k}}{d}\left(1 - \frac{\bar{k}}{d}\right) - \frac{D}{M^2 d}\left(1 - \frac{D}{d}\right) + \frac{H^2}{M^2 d} - \frac{H}{Md}} = 1 \\
\Leftrightarrow & \quad \sum_{f=1}^{d} s_f^2 = \frac{H}{M-1} - \frac{H^2}{M(M-1)}.
\end{aligned}$$

When all feature sets in $\mathcal{Z}$ are identical, we have $\bar{k} = k$ and therefore $H = (M\bar{k}) \mod M = 0$. Therefore the right-hand side of the above equation is 0 and the left-hand side is also 0. This proves that CW_{rel} possesses the backward implication of the Maximum property.

The forward implication is not true. We give the following counter-example

$$\mathcal{Z} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix},$$

It can easily be shown that $\hat{\Phi}_{CW_{rel}}(\mathcal{Z}) = 1$, even though all rows in $\mathcal{Z}$ are not identical. Therefore CW_{rel} does not have the Maximum property.

C.4.6 Lausser's Measure

To prove that Lausser's measure has this property, we will use the expression given in Equation (12), that is

$$\hat{\Phi}_{Lausser}(\mathcal{Z}) = \frac{1}{k} \sum_{f=1}^{d} \hat{p}_f^2.$$

Let us first assume that all feature sets in $\mathcal{Z}$ are identical. This implies that we will have exactly k features for which the value of $\hat{p}_f$ will be 1 and $d - k$ features for which it will be 0. Hence in that case, $\hat{\Phi}_{Lausser}(\mathcal{Z}) = 1$. Now let us assume that the stability is equal to 1. This means that we have $\sum_{f=1}^{d} \hat{p}_f^2 = k$. The only solution to that is when we have exactly k features with a frequency of selection equal to 1 and the other features have a frequency of selection equal to 0. Therefore Lausser's measure possesses the Maximum property.

C.5 Fifth Property: Correction for Chance

In order to prove the property of Correction for chance, we calculate the expected value of $\hat{\Phi}(\mathcal{Z})$ under the Null Model of Feature Selection H_0 for each one of the existing stability measures. If $\mathbb{E}\left[\hat{\Phi}(\mathcal{Z})|H_0\right]$ is not constant (i.e. if it depends on parameters of the problem like k or d), then it does not have this property.

C.5.1 Similarity-based Measures

It has been shown in the literature that under H_0, the intersection follows a hypergeometric distribution with known expected value equal to $\mathbb{E}\left[r_{i,j}|H_0\right] = \frac{k_i k_j}{d}$ (Lustgarten et al., 2009). Using the linearity of the expectation, we will have that $\mathbb{E}\left[\hat{\Phi}(\mathcal{Z})|H_0\right]$ of the Normalized Hamming distance, the Jaccard index, the Dice-Sørensen index, Ochiai index and the *POG* measures will depend on k_i, k_j and d. Therefore, all these stability measures will not have the property of Correction for chance. All other similarity measures will verify $\mathbb{E}\left[\hat{\Phi}(\mathcal{Z})|H_0\right] = 0$ and will have the property of Correction for chance. We detail all calculations below.

For the Hamming similarity measure, we have

$$\begin{aligned}
\mathbb{E}\left[\hat{\Phi}_{Hamming}(\mathcal{Z})|H_0\right] &= 1 - \frac{1}{M(M-1)}\sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M}\frac{k_i + k_j - \mathbb{E}\left[r_{i,j}|H_0\right]}{d} \\
&= 1 - \frac{1}{M(M-1)}\sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M}\frac{k_i + k_j - \frac{k_i k_j}{d}}{d} \\
&= 1 - \frac{1}{M(M-1)d}\left[\sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M}k_i + \sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M}k_j - \sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M}\frac{k_i k_j}{d}\right] \\
&= 1 - \frac{1}{M(M-1)d}\left[M(M-1)\bar{k} + M(M-1)\bar{k} - \frac{1}{d}\sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M}k_i k_j\right] \\
&= 1 - 2\frac{\bar{k}}{d} + \frac{1}{M(M-1)d^2}\sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M}k_i k_j \\
&= 1 - 2\frac{\bar{k}}{d} + \frac{1}{M(M-1)d^2}\left(M^2\bar{k}^2 - \sum_{i=1}^{M}k_i^2\right) \\
&= 1 - 2\frac{\bar{k}}{d} + \frac{M}{M-1}\frac{\bar{k}^2}{d^2} - \frac{1}{M(M-1)}\sum_{i=1}^{M}\frac{k_i^2}{d^2}.
\end{aligned}$$

In particular, when the number of features selected is constant equal to k, the above equation becomes $1 - 2\frac{k}{d}\left(1 - \frac{k}{d}\right)$, which depends on k and d. Therefore, this measure measure does not have the Correction-for-chance property.

For the Jaccard similarity measure, we have

$$\begin{aligned}
\mathbb{E}\left[\hat{\Phi}_{Jaccard}(\mathcal{Z})|H_0\right] &= \frac{1}{M(M-1)}\sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M}\mathbb{E}\left[\frac{r_{i,j}}{k_i + k_j - r_{i,j}}|H_0\right] \\
&= \frac{1}{M(M-1)}\sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M}\sum_{n=1}^{d}\frac{n}{k_i + k_j - n}\mathbb{P}(r_{i,j} = n|H_0).
\end{aligned}$$

Since we know that under H_0, the intersection $r_{i,j}$ follows a central hypergeometric distribution, we have that

$$\mathbb{P}(r_{i,j} = n|H_0) = \frac{\binom{k_i}{n}\binom{d-k_i}{k_j-n}}{\binom{d}{k_j}}.$$

Therefore, the expected value of the average pairwise Jaccard index under the Null Model of Feature Selection H_0 is

$$\mathbb{E}\left[\hat{\Phi}_{Jaccard}(\mathcal{Z})|H_0\right] = \frac{1}{M(M-1)}\sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M}\sum_{n=1}^{d}\frac{n}{k_i + k_j - n}\frac{\binom{k_i}{n}\binom{d-k_i}{k_j-n}}{\binom{d}{k_j}}.$$

In particular, when the number of features selected is constant equal to k, the above equation becomes $\sum_{n=1}^{d}\frac{n}{2k-n}\frac{\binom{k}{n}\binom{d-k}{k-n}}{\binom{d}{k}}$, which depends on k and d. Therefore this measure does not have the Correction-for-chance property.

For the Dice coefficient, we have

$$\mathbb{E}\left[\hat{\Phi}_{Dice}(\mathcal{Z})|H_0\right] = \frac{1}{M(M-1)}\sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M}\frac{2\mathbb{E}\left[r_{i,j}|H_0\right]}{k_i + k_j} = \frac{2}{M(M-1)d}\sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M}\frac{k_i k_j}{k_i + k_j}.$$

In particular, when the number of features selected is constant equal to k, the above equation becomes $\frac{k}{d}$, which depends on k and d. Therefore this measure does not have the Correction-for-chance property.

For Ochiai's index, we have

$$\begin{aligned}
\mathbb{E}\left[\hat{\Phi}_{Ochiai}(\mathcal{Z})|H_0\right] &= \frac{1}{M(M-1)}\sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M}\frac{\mathbb{E}\left[r_{i,j}|H_0\right]}{\sqrt{k_i k_j}} = \frac{1}{M(M-1)d}\sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M}\frac{k_i k_j}{\sqrt{k_i k_j}} \\
&= \frac{1}{M(M-1)d}\sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M}\sqrt{k_i k_j} = -\frac{1}{M-1}\frac{\bar{k}}{d} + \frac{1}{M(M-1)}\left(\sum_{i=1}^{M}\sqrt{\frac{k_i}{d}}\right)^2.
\end{aligned}$$

In particular, when the number of features selected is constant equal to k, the above equation becomes $\frac{k}{d}$, which depends on k and d. Therefore this measure does not have the Correction-for-chance property.

For *POG* measure, we use its symmetrical version equal to $\frac{r_{i,j}}{2k_i} + \frac{r_{i,j}}{2k_j}$ to carry out calculations. This results in the same stability value.

$$\begin{aligned}\mathbb{E}\left[\hat{\Phi}_{POG}(\mathcal{Z})|H_0\right] &= \frac{1}{M(M-1)}\sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M}\left(\frac{\mathbb{E}\left[r_{i,j}|H_0\right]}{2k_i} + \frac{\mathbb{E}\left[r_{i,j}|H_0\right]}{2k_j}\right)\\ &= \frac{1}{M(M-1)}\sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M}\left(\frac{k_ik_j}{2dk_i} + \frac{k_ik_j}{2dk_j}\right) = \frac{\bar{k}}{d},\end{aligned}$$

which depends on the number of feature selected and d. Therefore this measure does not have the Correction-for-chance property.

C.5.2 Frequency-based Measures

We showed in the proof of Theorem 3 (c.f. Appendix B.2), that under the Null Model of Feature Selection H_0, that p_f was equal to $\frac{\bar{k}}{d}$. Therefore, $\mathbb{E}\left[\hat{p}_f|H_0\right] = p_f = \frac{\bar{k}}{d}$. This will be used repeatedly in the proofs below.

For Goh's measure, since we have $\hat{\Phi}_{Goh}(\mathcal{Z}) = \frac{\bar{k}}{d}$ (c.f. Equation 10), this gives $\mathbb{E}\left[\hat{\Phi}_{Goh}(\mathcal{Z})|H_0\right] = \hat{\Phi}_{Goh}(\mathcal{Z}) = \frac{\bar{k}}{d}$, which is not constant. Therefore this measure does not have the property.

For Davis' Measure, we have $\mathbb{E}\left[\hat{\Phi}_{Davis}(\mathcal{Z})|H_0\right] = \hat{\Phi}_{Davis}(\mathcal{Z})$ as well. Therefore this measure does not have the Correction-for-chance property.

For Krízek's Measure, when the feature selection procedure is randomly selecting feature sets of cardinality k, the expected value of the frequency of occurrence of a feature set is equal to $\frac{1}{\binom{d}{k}}$. Therefore

$$\mathbb{E}\left[\hat{\Phi}_{Krizek}(\mathcal{Z})|H_0\right] = -\sum_{j=1}^{\binom{d}{k}}\frac{1}{\binom{d}{k}}\log\frac{1}{\binom{d}{k}} = -\log\frac{1}{\binom{d}{k}} = \log\binom{d}{k}.$$

Therefore Krízek's measure is not corrected for chance.

For Guzmán-Martinez's Measure, the stability measure is said to *"take the value zero for completely random rankings"* (Guzmán-Martínez and Alaiz-Rodríguez, 2011, pg 602), so we expect this measure to possess the Correction-for-chance property. We show this below.

$$\begin{aligned}\mathbb{E}\left[\hat{\Phi}_{Guzman}(\mathcal{Z})|H_0\right] &= 1 - \frac{\frac{1}{d}}{\frac{k}{d}\log(\frac{k}{d})}\sum_{f=1}^{d}\mathbb{E}\left[\hat{p}_f\log\hat{p}_f|H_0\right]\\ &= 1 - \frac{\frac{1}{d}}{\frac{k}{d}\log(\frac{k}{d})}\sum_{f=1}^{d}\mathbb{E}\left[\frac{\sum_{i=1}^{M}z_{i,f}}{M}\log\frac{\sum_{i=1}^{M}z_{i,f}}{M}|H_0\right].\end{aligned} \tag{13}$$

Under the Null Model of Feature Selection H_0, we have that $z_{i,f}$ follows a Bernoulli distribution with parameter $\frac{k}{d}$. Since we assumed that the samples $z_{1,f},...,z_{M,f}$ are independent and identically distributed (i.i.d.), we have that $\sum_{i=1}^{M}z_{i,f}$ follows a Binomial distribution with parameters M and $\frac{k}{d}$. Let $Y_f = \sum_{i=1}^{M}z_{i,f}$. Using this latter equation, we can calculate the expected value term of Equation (13),

$$\mathbb{E}\left[\frac{\sum_{i=1}^{M}z_{i,f}}{M}\log\frac{\sum_{i=1}^{M}z_{i,f}}{M}|H_0\right] = \mathbb{E}\left[\frac{Y_f}{M}\log\frac{Y_f}{M}|H_0\right].$$

Let $g: y \longmapsto \frac{y}{M}\log\frac{y}{M}$, we have

$$\mathbb{E}\left[\frac{\sum_{i=1}^{M}z_{i,f}}{M}\log\frac{\sum_{i=1}^{M}z_{i,f}}{M}|H_0\right] = \mathbb{E}\left[\frac{Y_f}{M}\log\frac{Y_f}{M}|H_0\right] = \mathbb{E}\left[g(Y_f)|H_0\right]. \tag{14}$$

Since g is a convex function[13] of y on the interval $(0,1]$, we can use Jensen's inequality, which gives

$$\begin{aligned}&\mathbb{E}\left[g(Y_f)|H_0\right] \geq g\left(\mathbb{E}\left[Y_f|H_0\right]\right) \Rightarrow \mathbb{E}\left[g(Y_f)|H_0\right] \geq g\left(M\frac{k}{d}\right) \Rightarrow \mathbb{E}\left[g(Y_f)|H_0\right] \geq \frac{k}{d}\log\frac{k}{d}\\ \Rightarrow\quad &\frac{1}{d}\sum_{f=1}^{d}\mathbb{E}\left[g(Y_f)|H_0\right] \geq \frac{k}{d}\log\frac{k}{d} \Rightarrow \frac{\frac{1}{d}}{\frac{k}{d}\log(\frac{k}{d})}\sum_{f=1}^{d}\mathbb{E}\left[g(Y_f)|H_0\right] \geq 1\\ \Rightarrow\quad &\frac{\frac{1}{d}}{\frac{k}{d}\log(\frac{k}{d})}\sum_{f=1}^{d}\mathbb{E}\left[g(Y_f)|H_0\right] \leq -1 \Rightarrow 1 - \frac{\frac{1}{d}}{\frac{k}{d}\log(\frac{k}{d})}\sum_{f=1}^{d}\mathbb{E}\left[g(Y_f)|H_0\right] \leq 0\\ \Rightarrow\quad &1 - \mathbb{E}\left[\frac{\frac{1}{d}}{\frac{k}{d}\log(\frac{k}{d})}\sum_{f=1}^{d}g(Y_f)|H_0\right] \leq 0.\end{aligned}$$

As shown by Equations (13) and (14), the left-hand-side term is equal to $\mathbb{E}\left[\hat{\Phi}_{Guzman}(\mathcal{Z})|H_0\right]$, therefore we get $\mathbb{E}\left[\hat{\Phi}_{Guzman}(\mathcal{Z})|H_0\right] \leq 0$. Since we know that $\hat{\Phi}_{Guzman}(\mathcal{Z})$ is a positive quantity, this gives us that $\mathbb{E}\left[\hat{\Phi}_{Guzman}(\mathcal{Z})|H_0\right] = 0$.

For the Relative Weighted Consistency CW_{rel}, using Equation (8), the result of Theorem 3 and by linearity of the expectation, we get

$$\mathbb{E}\left[\hat{\Phi}_{CW_{rel}}(\mathcal{Z})|H_0\right] = \frac{-\frac{1}{d}\frac{M-1}{M}\sum_{f=1}^{d}\mathbb{E}\left[s_f^2|H_0\right] + \frac{\bar{k}}{d}\left(1-\frac{\bar{k}}{d}\right) - \frac{D}{M^2d}\left(1-\frac{D}{d}\right)}{\frac{\bar{k}}{d}\left(1-\frac{\bar{k}}{d}\right) - \frac{D}{M^2d}\left(1-\frac{D}{d}\right) + \frac{H^2}{M^2d} - \frac{H}{Md}}$$

13. Indeed, its second derivative $g''(y) = \frac{1}{y\ln a}$ where a is the logarithm base used is non-negative for $y \in (0,1]$. Therefore g is convex on that interval.

$$= \frac{-\frac{1}{d}\frac{M-1}{M}\sum_{f=1}^{d}\frac{\bar{k}}{d}\left(1-\frac{\bar{k}}{d}\right)+\frac{\bar{k}}{d}\left(1-\frac{\bar{k}}{d}\right)-\frac{D}{M^2d}\left(1-\frac{D}{d}\right)}{\frac{\bar{k}}{d}\left(1-\frac{\bar{k}}{d}\right)-\frac{D}{M^2d}\left(1-\frac{D}{d}\right)+\frac{H^2}{M^2d}-\frac{H}{Md}}$$
$$= \frac{\frac{1}{M}\frac{\bar{k}}{d}\left(1-\frac{\bar{k}}{d}\right)-\frac{D}{M^2d}\left(1-\frac{D}{d}\right)}{\frac{\bar{k}}{d}\left(1-\frac{\bar{k}}{d}\right)-\frac{D}{M^2d}\left(1-\frac{D}{d}\right)+\frac{H^2}{M^2d}-\frac{H}{Md}}.$$

Since $H = (M\bar{k}) \mod M$, H is such that $M\bar{k} = \lfloor\bar{k}\rfloor M + H$. Therefore $H = M(\bar{k} - \lfloor\bar{k}\rfloor)$. Replacing in the previous equation, we get

$$\mathbb{E}\left[\hat{\Phi}_{CW_{rel}}(\mathcal{Z})|H_0\right] = \frac{\frac{1}{M}\frac{\bar{k}}{d}\left(1-\frac{\bar{k}}{d}\right)-\frac{D}{M^2d}\left(1-\frac{D}{d}\right)}{\frac{\bar{k}}{d}\left(1-\frac{\bar{k}}{d}\right)-\frac{D}{M^2d}\left(1-\frac{D}{d}\right)+\frac{(\bar{k}-\lfloor\bar{k}\rfloor)^2}{d}-\frac{\bar{k}-\lfloor\bar{k}\rfloor}{d}},$$

which is not constant. Nevertheless, when $\lfloor\bar{k}\rfloor = \bar{k}$, we have $\mathbb{E}\left[\hat{\Phi}_{CW_{rel}}(\mathcal{Z})|H_0\right] \xrightarrow[M\to\infty]{} 0$ and therefore the relative weighted consistency CW_{rel} is asymptotically corrected for chance. This is a result we expect since when the number of selected features is constant, this measure is asymptotically equivalent to our proposed measure (c.f. Theorem 5).

For Lausser's Measure, using Equation (12), we have

$$\begin{aligned}
\mathbb{E}\left[\hat{\Phi}_{Lausser}(\mathcal{Z})|H_0\right] &= \mathbb{E}\left[\frac{1}{k}\sum_{f=1}^{d}\hat{p}_f^2|H_0\right] = -\frac{1}{k}\sum_{f=1}^{d}\mathbb{E}\left[\hat{p}_f-\hat{p}_f^2-\hat{p}_f|H_0\right] \\
&= -\frac{1}{k}\sum_{f=1}^{d}\mathbb{E}\left[\hat{p}_f(1-\hat{p}_f)|H_0\right]+\frac{1}{k}\sum_{f=1}^{d}\mathbb{E}\left[\hat{p}_f|H_0\right] \\
&= -\frac{1}{k}\frac{M-1}{M}\sum_{f=1}^{d}\mathbb{E}\left[\frac{M}{M-1}\hat{p}_f(1-\hat{p}_f)|H_0\right]+\frac{1}{k}\sum_{f=1}^{d}p_f \\
&= -\frac{1}{k}\frac{M-1}{M}\sum_{f=1}^{d}p_f(1-p_f)+\frac{1}{k}\sum_{f=1}^{d}p_f.
\end{aligned}$$

As shown earlier, $\mathbb{E}\left[\hat{p}_f|H_0\right] = \frac{k}{d}$, therefore

$$\mathbb{E}\left[\hat{\Phi}_{Lausser}(\mathcal{Z})|H_0\right] = -\frac{M-1}{M}\left(1-\frac{k}{d}\right)+1 = \frac{1}{M}+\frac{M-1}{M}\frac{k}{d},$$

which is not constant. Therefore, Lausser's measure does not have this property.

Appendix D. Proof of the Lower Bound of the Proposed Measure

In this section, we prove the lower bound of the proposed stability measure given in Definition 4. To do so, we first prove the lemma below that will be used later on.

Lemma 10 $\frac{1}{d^2}\sum_{f<f'}(\hat{p}_f-\hat{p}_{f'})^2 = \frac{1}{d}\sum_{f=1}^{d}\hat{p}_f^2 - \left(\frac{1}{d}\sum_{f=1}^{d}\hat{p}_f\right)^2$.

Proof.
Starting from the right-hand term, we get

$$\begin{aligned}
&\frac{1}{d}\sum_{f=1}^{d}\hat{p}_f^2-\left(\frac{1}{d}\sum_{f=1}^{d}\hat{p}_f\right)^2 = \left(\frac{1}{d}\sum_{f=1}^{d}\hat{p}_f^2\right)-\frac{1}{d^2}\sum_{f=1}^{d}\sum_{f'=1}^{d}\hat{p}_f\hat{p}_{f'} = \frac{1}{d^2}\sum_{f=1}^{d}\left(d\hat{p}_f^2-\hat{p}_f\sum_{f'=1}^{d}\hat{p}_{f'}\right) \\
&= \frac{1}{d^2}\sum_{f=1}^{d}\sum_{f'=1}^{d}\hat{p}_f\left(\hat{p}_f-\hat{p}_{f'}\right).
\end{aligned}$$

Since the term $(\hat{p}_f - \hat{p}_{f'})$ is equal to zero when $f = f'$, by splitting the sum in two terms, this is equal to

$$\frac{1}{d^2}\sum_{f=1}^{d}\sum_{f'=1}^{f-1}\hat{p}_f\left(\hat{p}_f-\hat{p}_{f'}\right)+\frac{1}{d^2}\sum_{f=1}^{d}\sum_{f'=f+1}^{d}\hat{p}_f\left(\hat{p}_f-\hat{p}_{f'}\right) \quad .$$

The left term $\sum_{f=1}^{d}\sum_{f'=1}^{f-1}\hat{p}_f(\hat{p}_f-\hat{p}_{f'})$ is equal to $\sum_{f=1}^{d}\sum_{f'=f+1}^{d}-\hat{p}_{f'}(\hat{p}_f-\hat{p}_{f'})$. Therefore the previous equation becomes

$$\begin{aligned}
&\frac{1}{d^2}\sum_{f=1}^{d}\sum_{f'=f+1}^{d}-\hat{p}_{f'}\left(\hat{p}_f-\hat{p}_{f'}\right)+\frac{1}{d^2}\sum_{f=1}^{d}\sum_{f'=f+1}^{d}\hat{p}_f\left(\hat{p}_f-\hat{p}_{f'}\right) \\
&= \frac{1}{d^2}\sum_{f=1}^{d}\sum_{f'=f+1}^{d}\left(-\hat{p}_{f'}\left(\hat{p}_f-\hat{p}_{f'}\right)+\hat{p}_f\left(\hat{p}_f-\hat{p}_{f'}\right)\right) \\
&= \frac{1}{d^2}\sum_{f=1}^{d}\sum_{f'=f+1}^{d}\left(\hat{p}_f-\hat{p}_{f'}\right)^2 = \frac{1}{d^2}\sum_{f<f'}\left(\hat{p}_f-\hat{p}_{f'}\right)^2.
\end{aligned}$$

■

Since a sum of squares is always positive, using this lemma we have that

$$\begin{aligned}
&\frac{1}{d^2}\sum_{f<f'}(\hat{p}_f-\hat{p}_{f'})^2 \geq 0 \\
\Rightarrow\quad &\frac{1}{d}\sum_{f=1}^{d}\hat{p}_f^2-\left(\frac{1}{d}\sum_{f=1}^{d}\hat{p}_f\right)^2 \geq 0 \\
\Rightarrow\quad &\frac{1}{d}\sum_{f=1}^{d}\hat{p}_f^2-\left(\frac{\bar{k}}{d}\right)^2 \geq 0 \\
\Rightarrow\quad &\frac{1}{d}\sum_{f=1}^{d}\hat{p}_f^2 \geq \left(\frac{\bar{k}}{d}\right)^2 \\
\Rightarrow\quad &\frac{1}{d}\sum_{f=1}^{d}\hat{p}_f^2-\frac{1}{d}\sum_{f=1}^{d}\hat{p}_f \geq \left(\frac{\bar{k}}{d}\right)^2-\frac{1}{d}\sum_{f=1}^{d}\hat{p}_f
\end{aligned}$$

$$
\begin{aligned}
\Rightarrow \quad & -\frac{1}{d}\sum_{f=1}^{d}\hat{p}_f(1-\hat{p}_f) \geq \left(\frac{\bar{k}}{d}\right)^2 - \frac{\bar{k}}{d} \\
\Rightarrow \quad & -\frac{1}{d}\sum_{f=1}^{d}\hat{p}_f(1-\hat{p}_f) \geq -\frac{\bar{k}}{d}\left(1-\frac{\bar{k}}{d}\right) \\
\Rightarrow \quad & \frac{\frac{1}{d}\sum_{f=1}^{d}\hat{p}_f(1-\hat{p}_f)}{\frac{\bar{k}}{d}\left(1-\frac{\bar{k}}{d}\right)} \leq 1 \\
\Rightarrow \quad & 1 - \frac{\frac{1}{d}\frac{M}{M-1}\sum_{f=1}^{d}\hat{p}_f(1-\hat{p}_f)}{\frac{\bar{k}}{d}\left(1-\frac{\bar{k}}{d}\right)} \geq 1 - \frac{M}{M-1} \\
\Rightarrow \quad & \hat{\Phi}(\mathcal{Z}) \geq -\frac{1}{M-1}.
\end{aligned}
$$

Hence, $\hat{\Phi}(\mathcal{Z})$ is lower bounded by -1 (as $M \geq 2$), but asymptotically bounded by 0. ■

References

Salem Alelyani. *On Feature Selection Stability: A Data Perspective*. PhD thesis, Arizona State University, 2013.

Wilker Altidor, Taghi M. Khoshgoftaar, and Amri Napolitano. A noise-based stability evaluation of threshold-based feature selection techniques. In *IEEE International Conference on Information Reuse & Integration (IRI'11)*, pages 240–245, 2011.

Luca Baldassarre, Massimiliano Pontil, and Janaina Mouro-Miranda. Sparsity is better with stability: Combining accuracy and stability for model selection in brain decoding. *Frontiers in Neuroscience*, 11:62, 2017.

Kenneth J Berry, Paul W Mielke, Jr, and Janis E Johnston. *Permutation statistical methods: an integrated approach*. Springer, 2016.

Anne-Laure Boulesteix and Martin Slawski. Stability and aggregation of ranked gene lists. *Briefings in Bioinformatics*, 10(5):556–68, 2009.

Gavin Brown, Adam Pocock, Ming-Jie Zhao, and Mikel Luján. Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *The Journal of Machine Learning Research*, 13(1):27–66, 2012.

Chad A. Davis, Fabian Gerick, Volker Hintermair, Caroline C. Friedel, Katrin Fundel, Robert Kffner, and Ralf Zimmer. Reliable gene signatures for microarray classification: assessment of stability and performance. *Bioinformatics*, 22(19):2356–63, 2006.

David J. Dittman, Taghi M. Khoshgoftaar, Randall Wald, and Amri Napolitano. Similarity analysis of feature ranking techniques on imbalanced dna microarray datasets. In *2012 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1–5, 2012.

Gregory Ditzler, Robi Polikar, and Gail Rosen. A bootstrap based neyman-pearson test for identifying variable importance. *IEEE Transactions on Neural Networks and Learning Systems*, 26(4):880–886, 2015.

Kevin Dunne, Padraig Cunningham, and Francisco Azuaje. Solutions to instability problems with sequential wrapper-based approaches to feature selection. Technical Report TCD-CS-2002-28, Trinity College Dublin, School of Computer Science, 2002.

Joseph L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.

Joseph L. Fleiss, Bruce Levin, and Myunghee Cho Paik. *The Measurement of Interrater Agreement*, pages 598–626. John Wiley & Sons, Inc., 2004.

Wilson Wen Bin Goh and Limsoon Wong. Evaluating feature-selection stability in next-generation proteomics. *Journal of Bioinformatics and Computational Biology*, 14(05): 1650029, 2016.

Gokhan Gulgezen, Zehra Cataltepe, and Lei Yu. Stable and accurate feature selection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, pages 455–468, 2009.

Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3:1157–1182, 2003.

Roberto Guzmán-Martínez and Rocío Alaiz-Rodríguez. Feature selection stability assessment based on the jensen-shannon divergence. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, pages 597–612. 2011.

Kilem Li Gwet. Variance estimation of nominal-scale inter-rater reliability with random selection of raters. *Psychometrika*, 73(3):407, 2008.

Yue Han and Lei Yu. A variance reduction framework for stable feature selection. *Statistical Analysis and Data Mining*, 5(5):428–445, 2012.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer, 2nd edition, 2009.

Giuseppe Jurman, Stefano Merler, Annalisa Barla, Silvano Paoli, Antonio Galea, and Cesare Furlanello. Algebraic stability indicators for ranked lists in molecular profiling. *Bioinformatics*, 24(2):258–64, 2008.

Alexandros Kalousis, Julien Prados, and Melanie Hilario. Stability of feature selection algorithms. In *IEEE International Conference on Data Mining (ICDM'05)*, 2005.

Alexandros Kalousis, Julien Prados, and Melanie Hilario. Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowledge and Information Systems*, 12(1):95–116, 2007.

Iman Kamkar, Sunil Kumar Gupta, Dinh Phung, and Svetha Venkatesh. Stable feature selection with support vector machines. In *Australasian Joint Conference on Artificial Intelligence (AI 2015)*, volume 9457 of *LNCS*, pages 298–308, 2015.

Pavel Krízek, Josef Kittler, and Václav Hlavác. Improving stability of feature selection methods. In *International Conference on Computer Analysis of Images and Patterns (CAIP)*, volume 4673 of *LNCS*, pages 929–936, 2007.

Ludmila I. Kuncheva. A stability index for feature selection. In *Artificial Intelligence and Applications (AIAP'07)*, pages 390–395, 2007.

Ludwig Lausser, Christoph Müssel, Markus Maucher, and Hans A. Kestler. Measuring and visualizing the stability of biomarker selection techniques. *Computational Statistics*, 28 (1):51–65, 2013.

Hae Woo Lee, Carl Lawton, Young Jeong Na, and Seongkyu Yoon. Robustness of chemometrics-based feature selection methods in early cancer detection and biomarker discovery. *Statistical Applications in Genetics and Molecular Biology*, 12(2):207–23, 2012.

Jonathan L Lustgarten, Vanathi Gopalakrishnan, and Shyam Visweswaran. Measuring stability of feature selection in biomedical datasets. *AMIA Annual Symposium Proceedings.*, pages 406–410, 2009.

Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

Sarah Nogueira. *Quantifying the Stability of Feature Selection.* PhD thesis, University of Manchester, 2018.

Sarah Nogueira, Konstantinos Sechidis, and Gavin Brown. On the use of Spearman's rho to measure the stability of feature rankings. In *Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*, pages 381–391, 2017.

Mohana Chelvan P. and Karuppasamy Perumal. A survey on feature selection stability measures. *International Journal of Computer and Information Technology*, 5(1):98–103, 2016.

Yvan Saeys, Thomas Abeel, and Yves Van de Peer. Robust feature selection using ensemble feature selection techniques. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD*, volume 5212 of *LNCS*, pages 313–325, 2008.

Ahmad A. Shanab, Taghi M. Khoshgoftaar, and Randall Wald. Impact of noise and data sampling on stability of feature selection. In *International Conference on Machine Learning and Applications and Workshops (ICMLA)*, pages 172–177, 2011.

Leming Shi, Laura H. Reid, Wendell D. Jones, et al. The MicroArray Quality Control (MAQC) project shows inter- and intraplatform reproducibility of gene expression measurements. *Nature Biotechnology*, 24(9):1151–61, September 2006.

Petr Somol and Jana Novovičová. Evaluating stability and comparing output of feature selectors that optimize feature subset cardinality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(11):1921–1939, 2010.

Randall Wald, Taghi M. Khoshgoftaar, and David J. Dittman. A new fixed-overlap partitioning algorithm for determining stability of bioinformatics gene rankers. In *International Conference on Machine Learning and Applications (ICMLA)*, 2012a.

Randall Wald, Taghi M. Khoshgoftaar, and Ahmad Abu Shanab. The effect of measurement approach and noise level on gene selection stability. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2012b.

Randall Wald, Taghi M. Khoshgoftaar, and Amri Napolitano. Stability of filter- and wrapper-based feature subset selection. In *International Conference on Tools with Artificial Intelligence*. IEEE Computer Society, 2013.

Søren Wichmann and David Kamholz. A stability metric for typological features. *STUF-Language Typology and Universals Sprachtypologie und Universalienforschung*, 61(3):251–262, 2008.

Lei Yu, Chris H. Q. Ding, and Steven Loscalzo. Stable feature selection via dense feature groups. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2008.

Lei Yu, Yue Han, and Michael E. Berens. Stable gene selection from microarray data via sample weighting. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 9(1):262–272, 2012.

Min Zhang, Lin Zhang, Jinfeng Zou, Chen Yao, Hui Xiao, Qing Liu, Jing Wang, Dong Wang, Chenguang Wang, and Zheng Guo. Evaluating reproducibility of differential expression discoveries in microarray studies by considering correlated molecular changes. *Bioinformatics*, 2009.

Ding-Xuan Zhou. On grouping effect of elastic net. *Statistics & Probability Letters*, 83(9): 2108 – 2112, 2013.

Manuela Zucknick, Sylvia Richardson, and Euan A. Stronach. Comparing the characteristics of gene expression profiles derived by univariate and multivariate classification methods. *Statistical Applications in Genetics and Molecular Biology*, 7(1), 2008.

Journal of Machine Learning Research 18 (2018) 1-11 Submitted 12/16; Revised 12/16; Published 4/18

Maximum Likelihood Estimation for Mixtures of Spherical Gaussians is NP-hard

Christopher Tosh CTOSH@CS.UCSD.EDU
Sanjoy Dasgupta DASGUPTA@CS.UCSD.EDU
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093-0404, USA

Editor: Mikhail Belkin

Abstract

This paper presents NP-hardness and hardness of approximation results for maximum likelihood estimation of mixtures of spherical Gaussians.

Keywords: Mixtures of Gaussians, maximum likelihood, NP-completeness

1. Introduction

A *spherical Gaussian* in $\mathbb{R}^d$ is a distribution specified by its mean $\mu \in \mathbb{R}^d$ and variance $\sigma^2 > 0$, with density

$$N(x; \mu, \sigma^2) = \left(\frac{1}{2\pi\sigma^2}\right)^{d/2} \exp\left(-\frac{\|x-\mu\|^2}{2\sigma^2}\right).$$

(The standard notation for this Gaussian is $N(\mu, \sigma^2 I_d)$, but we will drop the identity matrix as a shorthand.)

When data arise from several sources, or form several clusters, it is common to model each source or cluster by a spherical Gaussian. If there are k sources, the resulting overall distribution is a mixture of k Gaussians,

$$\pi_1 N(\mu_1, \sigma_1^2) + \pi_2 N(\mu_2, \sigma_2^2) + \cdots + \pi_k N(\mu_k, \sigma_k^2),$$

where $\mu_i \in \mathbb{R}^d$ and σ_i^2 are the mean and variance of the ith component, and π_i is the fraction of the distribution that arises from this component. In what follows, we will often package the parameters together as $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_k)$, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_k)$, $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_k)$.

A standard statistical task is to fit a mixture of k Gaussians to a given data set. This is typically formulated as an optimization problem (Dempster et al., 1977), where given data points $x_1, \ldots, x_n \in \mathbb{R}^d$, the goal is to find the parameters $(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma})$ that maximize the *log-likelihood*

$$LL(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \sum_{i=1}^{n} \ln\left(\sum_{j=1}^{k} \pi_j N(x_i; \mu_j, \sigma_j^2)\right). \tag{1}$$

In this brief note, we establish the computational hardness of this estimation problem. This is in contrast with various positive results showing that, when data is in fact generated from a Gaussian mixture, it is possible to efficiently recover the mixture from a sample of polynomial size, under certain conditions; relevant work includes, for instance, Belkin and Sinha (2010), Moitra and Valiant (2010), Hsu and Kakade (2013), and Hardt and Price (2015), among others.

1.1 Gaussians with the same variance

We start with the simplest subcase, where the variances of the components are constrained to be the same.

> mixtures of spherical Gaussians with same variance: mog-sv
> *Input*: Points $x_1, \ldots, x_n \in \mathbb{R}^d$; positive integer k; unary parameter b.
> *Output*: A mixture of k spherical Gaussians with the same variance, $(\boldsymbol{\pi}, \boldsymbol{\mu}, \sigma)$, whose log-likelihood
>
> $$LL(\boldsymbol{\pi}, \boldsymbol{\mu}, \sigma) = \sum_{i=1}^{n} \ln\left(\sum_{j=1}^{k} \pi_j N(x_i; \mu_j, \sigma^2)\right)$$
>
> is within an additive factor $1/b$ of optimal.

Since the parameters of the optimal mixture are real-valued, they can only be provided to within some precision. The role of the input parameter b is to specify the desired level of accuracy. It is worth pointing out, however, that in our reductions, the coordinates of data points take values in $\{-1, 0, 1\}$ and the hardness does not stem from precision issues but rather from underlying combinatorial structure.

mog-sv is similar to the k-means clustering problem, which is NP-hard (Aloise et al., 2009).

> k-means
> *Input*: Points $x_1, \ldots, x_n \in \mathbb{R}^d$; positive integer k.
> *Output*: A collection of k "centers" $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_k)$ in $\mathbb{R}^d$ that minimize the cost function
>
> $$\Phi(\boldsymbol{\mu}) = \sum_{i=1}^{n} \min_{1 \le j \le k} \|x_i - \mu_j\|^2.$$

The biggest difference between the two problems is that k-means assigns each data point x_i to a single center μ_j (a "hard" clustering), while the mixture of Gaussians effectively spreads it out over all the centers (a "soft" clustering). Earlier work (Arora and Kannan, 2001) has established that a "hard clustering" version of the mixture of Gaussians problem is NP-hard. Here we consider the more standard formulation, and show that it is hard even when $k = 2$.

Theorem 1 mog-sv *is NP-hard on instances with* $k = 2$.

The proof follows from the observation that an additive approximation to the best mog-sv solution yields a multiplicative approximation to the best k-means solution:

Lemma 2 *Fix any data set $x_1, \ldots, x_n \in \mathbb{R}^d$ and any positive integer k. Let LL_{OPT} denote the log-likelihood of the optimal solution to* MOG-SV*, and Φ_{OPT} the lowest achievable k-means cost. For any parameters $(\boldsymbol{\pi}, \boldsymbol{\mu}, \sigma)$, we have*

$$\ln \frac{\Phi(\boldsymbol{\mu})}{\Phi_{OPT}} \;\leq\; \frac{4 \ln k}{d} + \frac{2}{nd}\left(LL_{OPT} - LL(\boldsymbol{\pi}, \boldsymbol{\mu}, \sigma)\right).$$

The first term on the right-hand side comes from the discrepancy between hard and soft clustering. It can be made negligible by increasing the dimension, for instance by padding each point with extra zero-valued coordinates.

Lemma 2 can also be combined with a recent hardness of approximation result for k-means (Awasthi et al., 2015) to show that, if k is allowed to be large, MOG-SV cannot be approximated within an additive factor of $o(nd)$.

Theorem 3 *There is a family of* MOG-SV *instances with the following properties:*

- *An instance with n points has dimension $O(n)$.*
- *Each point is $\{0,1\}$-valued and has $O(1)$ nonzero coordinates.*
- *$k = \Theta(n)$.*

For some absolute constant c_o, it is NP-hard to approximate MOG-SV *on such instances within an additive factor of $c_o dn$.*

The specific form of this result (additive versus multiplicative approximation, interpoint distances that are small constants) is motivated by the unusual properties of the log-likelihood objective. To begin with, consider the problem of fitting a single Gaussian to a data set $\mathcal{X} \subset \mathbb{R}^d$ of size n. A quick calculation shows that the log-likelihood (of the maximum likelihood estimate) is

$$\frac{dn}{2} \ln \frac{d}{2\pi e} - \frac{dn}{2} \ln \text{radius}(\mathcal{X}), \quad \textit{where} \quad \text{radius}(\mathcal{X}) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \|x - \text{mean}(\mathcal{X})\|^2.$$

Depending on the scale of the data, this log-likelihood could be positive, negative, or zero. When fitting a mixture of k Gaussians, the log-likelihood has a term of this sort for each cluster, plus an additional term of size $\pm n \ln k$ due to the mixing weights. For the kind of instance described in the theorem, any cluster with at least two points has radius $\Theta(1)$ and thus the log-likelihoods of all reasonable mixture models lie in an interval of size $O(dn)$.

The proofs of these results appear in Section 2.

1.2 Gaussians with differing variances

When the different Gaussian components are allowed to have different variances, and $k > 1$, the maximum-likelihood solution is always degenerate. This is because it is possible to make the log-likelihood go to infinity by centering one of the Gaussians at a single data point and letting its variance go to zero. Thus, in order for the problem to be well-defined, an additional constraint must be introduced. One option is to force all variances to be non-negligible.

mixtures of spherical Gaussians with constrained variances: MOG
Input: Points $x_1, \ldots, x_n \in \mathbb{R}^d$; positive integer k; value $\sigma_o > 0$; unary integer b.
Output: A mixture of k spherical Gaussians $(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma})$ whose log-likelihood is within an additive factor $1/b$ of optimal, subject to the constraint $\sigma_1, \ldots, \sigma_k \geq \sigma_o$.

This problem is slightly further from k-means, but remains intractable.

Theorem 4 MOG *is NP-hard on instances with $k = 2$.*

The proof appears in Section 3.

2. Mixtures of spherical Gaussians with the same variance

2.1 Induced partitions

We start with a basic relation between hard and soft clustering that applies to arbitrary mixture models, not just those with Gaussian components of the same variance.

Although a mixture model represents a soft clustering, it also induces a natural hard partition. For data set $\mathcal{X}$ and mixture of Gaussians $(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma})$, this hard partition has clusters

$$\mathcal{X}_j \;=\; \left\{ x \in \mathcal{X} \,:\, j = \operatorname*{argmax}_{\ell} \; \pi_\ell N(x; \mu_\ell, \sigma_\ell^2) \right\} \tag{2}$$

(breaking ties arbitrarily). The log-likelihood of a mixture is easily bounded in terms of the likelihood of the corresponding hard partition.

Lemma 5 *Pick any mixture $(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma})$ and data set $\mathcal{X} = \{x_1, \ldots, x_n\}$.*

(a) For any partition $(\mathcal{X}'_1, \ldots, \mathcal{X}'_k)$ of $\mathcal{X}$, we have

$$LL(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}) \;\geq\; \sum_{j=1}^{k} \sum_{x \in \mathcal{X}'_j} \ln(\pi_j N(x; \mu_j, \sigma_j^2)).$$

(b) For the partition $(\mathcal{X}_1, \ldots, \mathcal{X}_k)$ induced by $(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma})$, as in Eq (2), we have

$$LL(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}) \;\leq\; n \ln k + \sum_{j=1}^{k} \sum_{x \in \mathcal{X}_j} \ln(\pi_j N(x; \mu_j, \sigma_j^2)).$$

Proof Recall from (1) that the contribution of each data point x_i to $LL(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma})$ is

$$\ln \left(\sum_{j=1}^{k} \pi_j N(x_i; \mu_j, \sigma_j^2) \right).$$

For $x_i \in \mathcal{X}'_j$, we can lower-bound this contribution by $\ln(\pi_j N(x_i; \mu_j, \sigma_j^2))$. Similarly, if $x_i \in \mathcal{X}_j$, then we can upper-bound the contribution by $\ln(k\pi_j N(x_i; \mu_j, \sigma_j^2))$, by the manner in which the hard partition $(\mathcal{X}_1, \ldots, \mathcal{X}_k)$ is defined. ■

2.2 Proof of Lemma 2

As in the statement of Lemma 2, fix data $x_1, \ldots, x_n \in \mathbb{R}^d$, and define LL_{OPT} to be the log-likelihood of the optimal solution of MOG-SV. Let Φ_{OPT} be the optimal k-means cost.

Pick any parameters $(\boldsymbol{\pi}, \boldsymbol{\mu}, \sigma)$, and let $(\mathcal{X}_1, \ldots, \mathcal{X}_k)$ be the induced hard partition of the data set, as per Eq (2). From Lemma 5,

$$\begin{aligned} LL(\boldsymbol{\pi}, \boldsymbol{\mu}, \sigma) &\leq n \ln k + \sum_{j=1}^{k} \sum_{x \in \mathcal{X}_j} \left(\ln \pi_j + \frac{d}{2} \ln \left(\frac{1}{2\pi\sigma^2} \right) - \frac{\|x - \mu_j\|^2}{2\sigma^2} \right) \\ &\leq n \ln k + \frac{nd}{2} \ln \left(\frac{1}{2\pi\sigma^2} \right) - \frac{1}{2\sigma^2} \sum_{j=1}^{k} \sum_{x \in \mathcal{X}_j} \|x - \mu_j\|^2 \\ &\leq n \ln k + \frac{nd}{2} \ln \left(\frac{1}{2\pi\sigma^2} \right) - \frac{\Phi(\boldsymbol{\mu})}{2\sigma^2} \\ &\leq n \ln k + \frac{nd}{2} \ln \left(\frac{nd}{2\pi\Phi(\boldsymbol{\mu})} \right) - \frac{nd}{2}, \end{aligned}$$

where the last inequality comes from solving for the optimal value of σ^2 (namely, $\Phi(\boldsymbol{\mu})/nd$) in the preceding line.

Suppose the optimal k-means solution is realized by centers $\boldsymbol{\mu}^* = (\mu_1^*, \ldots, \mu_k^*)$. Let $\pi_1^* = \cdots = \pi_k^* = 1/k$ and $\sigma^{*2} = \Phi(\boldsymbol{\mu}^*)/nd$. To bound the log-likelihood of the mixture model $(\boldsymbol{\pi}^*, \boldsymbol{\mu}^*, \sigma^*)$, we look at the hard partition that it induces, $(\mathcal{X}_1^*, \ldots, \mathcal{X}_k^*)$, and notice that $\mathcal{X}_j^*$ consists of points whose closest center is μ_j^*. We then apply Lemma 5 to get

$$\begin{aligned} LL(\boldsymbol{\pi}^*, \boldsymbol{\mu}^*, \sigma^*) &\geq \sum_{j=1}^{k} \sum_{x \in \mathcal{X}_j^*} \left(\ln \pi_j^* + \frac{d}{2} \ln \left(\frac{1}{2\pi\sigma^{*2}} \right) - \frac{\|x - \mu_j^*\|^2}{2\sigma^{*2}} \right) \\ &= -n \ln k + \frac{nd}{2} \ln \left(\frac{1}{2\pi\sigma^{*2}} \right) - \frac{1}{2\sigma^{*2}} \sum_{j=1}^{k} \sum_{x \in \mathcal{X}_j^*} \|x - \mu_j^*\|^2 \\ &= -n \ln k + \frac{nd}{2} \ln \left(\frac{1}{2\pi\sigma^{*2}} \right) - \frac{1}{2\sigma^{*2}} \Phi(\boldsymbol{\mu}^*) \\ &= -n \ln k + \frac{nd}{2} \ln \left(\frac{nd}{2\pi\Phi(\boldsymbol{\mu}^*)} \right) - \frac{nd}{2}, \end{aligned}$$

where the last equality comes from substituting in the value of σ^{*2}. Combining our bounds for the two mixtures, we get

$$\begin{aligned} LL_{OPT} - LL(\boldsymbol{\pi}, \boldsymbol{\mu}, \sigma) &\geq LL(\boldsymbol{\pi}^*, \boldsymbol{\mu}^*, \sigma^*) - LL(\boldsymbol{\pi}, \boldsymbol{\mu}, \sigma) \\ &\geq \frac{nd}{2} \ln \left(\frac{\Phi(\boldsymbol{\mu})}{\Phi(\boldsymbol{\mu}^*)} \right) - 2n \ln k. \end{aligned}$$

Rearranging terms yields the lemma statement.

2.3 Proof of Theorem 1

With Lemma 2 in place, a reduction from k-means to MOG-SV is almost immediate. There are various hardness results available for k-means (Aloise et al., 2009; Dasgupta and Freund, 2009; Mahajan et al., 2009; Awasthi et al., 2015); of these, we use Aloise et al. (2009) as a starting point.

Theorem 6 (Aloise et al. (2009)) *There exists a family of k-means instances with the following properties, for some low-order polynomials $\alpha(\cdot)$ and $\beta(\cdot)$:*

- *For an instance containing n points, each point has dimension at most $\alpha(n)$, with individual coordinates taking values in $\{-1, 0, 1\}$.*
- *It is NP-hard to approximate the best k-means solution, with $k = 2$, within a factor of $1 + 1/\beta(n)$.*

To prove Theorem 1, we reduce the problem of finding a $(1 + 1/\beta(n))$-approximate k-means solution to MOG-SV. Given an instance $x_1, \ldots, x_n$ of k-means:

- Pad each point with additional zero-valued coordinates until the dimension d exceeds $16\beta(n) \ln k$. This has no effect on interpoint distances or on the optimal k-means cost.
- Solve MOG-SV for these modified points, with precision parameter $b = 1$. This yields $(\boldsymbol{\pi}, \boldsymbol{\mu}, \sigma)$ such that $LL_{OPT} - LL(\boldsymbol{\pi}, \boldsymbol{\mu}, \sigma) \leq 1$, where LL_{OPT} is the optimal log-likelihood. It follows from Lemma 2 that

  $$\ln \frac{\Phi(\boldsymbol{\mu})}{\Phi_{OPT}} \leq \frac{4 \ln k}{d} + \frac{2}{nd} \leq \frac{1}{2\beta(n)},$$

  whereupon $\Phi(\boldsymbol{\mu}) \leq \Phi_{OPT}(1 + 1/\beta(n))$.

2.4 Proof of Theorem 3

A recent hardness of approximation result for k-means shows the following.

Theorem 7 (Awasthi et al. (2015)) *There is a family of k-means instances with the following properties:*

- *An instance with n points has dimension at most n, points that are $\{0, 1\}$-valued (and have at most two non-zero coordinates), and a target number of clusters $k = \Omega(n)$.*
- *It is NP-hard to approximate the optimal k-means solution within a factor c, for some absolute constant $c > 1$.*

Pick any $c_o < (1/2) \ln c$. To see that it is hard to approximate MOG-SV within an additive factor $c_o nd$, we reduce from k-means as follows. Start with an instance $x_1, \ldots, x_n \in \mathbb{R}^d$ of the type described in Theorem 7. Then:

- If necessary, pad points with zero-valued coordinates to bring the dimension up to

  $$d \geq \frac{4 \ln k}{(\ln c) - 2c_o}.$$

- Obtain an approximate solution $(\boldsymbol{\pi}, \boldsymbol{\mu}, \sigma)$ to MOG-SV on these points such that $LL_{OPT} - LL(\boldsymbol{\pi}, \boldsymbol{\mu}, \sigma) \leq c_o nd$.

- Return the centers $\boldsymbol{\mu}$.

By Lemma 2, we have

$$\ln \frac{\Phi(\boldsymbol{\mu})}{\Phi_{OPT}} \;\leq\; \frac{4 \ln k}{d} + \frac{2}{nd} c_o nd \;\leq\; \ln c,$$

so that $\boldsymbol{\mu}$ is a c-approximate solution to the k-means instance.

3. The general case

We now consider the case where the variances are allowed to differ but are uniformly lower bounded. Specifically, a mixture model $(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma})$ is *admissible* if all $\sigma_j \geq \sigma_o$, where σ_o is supplied as part of the input.

The basic reduction still applies, with an additional device to force all variances to be close to the lower bound—and therefore approximately equal.

3.1 Controlling the variances

Lemma 8 *Fix any data set $\mathcal{X} = \{x_1, \ldots x_n\}$ in $\mathbb{R}^d$, and let $D = \max_{i \neq i'} \|x_i - x_{i'}\|$ denote its diameter. Pick any $\Delta, \delta > 0$. If the dimension d satisfies*

$$d \geq \frac{4}{\delta} \left(\frac{nD^2}{2\sigma_o^2} + n \ln k + \Delta \right), \tag{3}$$

then any admissible mixture $(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma})$ within an additive factor Δ of optimal (that is, $LL(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}) \geq LL_{OPT} - \Delta$) has the following property: in the associated hard partition $(\mathcal{X}_1, \ldots, \mathcal{X}_k)$, any nonempty cluster $\mathcal{X}_j$ has $\sigma_j^2 \leq \sigma_o^2(1+\delta)$.

Proof Pick any admissible mixture $(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma})$ that is within Δ of optimal, and let $(\mathcal{X}_1, \ldots, \mathcal{X}_k)$ be the associated hard partition. Let $\widetilde{\mu}_j$ denote the cluster means:

$$\widetilde{\mu}_j \;=\; \frac{1}{|\mathcal{X}_j|} \sum_{x \in \mathcal{X}_j} x.$$

Using Lemma 5, we can compare the log-likelihood of $(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma})$ to that of the adjusted parameters $(\boldsymbol{\pi}, \widetilde{\boldsymbol{\mu}}, \widetilde{\boldsymbol{\sigma}})$, where each $\widetilde{\sigma}_j = \sigma_o$.

$$\begin{aligned}
&LL(\boldsymbol{\pi}, \widetilde{\boldsymbol{\mu}}, \widetilde{\boldsymbol{\sigma}}) - LL(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}) \\
&\geq \sum_{j=1}^{k} \sum_{x \in \mathcal{X}_j} \left(\ln(\pi_j N(x; \widetilde{\mu}_j, \sigma_o^2)) - \ln(\pi_j N(x; \mu_j, \sigma_j^2)) \right) - n \ln k \\
&= \sum_{j=1}^{k} \sum_{x \in \mathcal{X}_j} \left(\frac{d}{2} \ln \frac{1}{2\pi\sigma_o^2} - \frac{\|x - \widetilde{\mu}_j\|^2}{2\sigma_o^2} - \frac{d}{2} \ln \frac{1}{2\pi\sigma_j^2} + \frac{\|x - \mu_j\|^2}{2\sigma_j^2} \right) - n \ln k \\
&= \sum_{j=1}^{k} \left(\frac{d|\mathcal{X}_j|}{2} \ln \frac{\sigma_j^2}{\sigma_o^2} + \sum_{x \in \mathcal{X}_j} \left(\frac{\|x - \mu_j\|^2}{2\sigma_j^2} - \frac{\|x - \widetilde{\mu}_j\|^2}{2\sigma_o^2} \right) \right) - n \ln k \\
&\geq \sum_{j=1}^{k} \left(\frac{d|\mathcal{X}_j|}{2} \ln \frac{\sigma_j^2}{\sigma_o^2} + \left(\frac{1}{2\sigma_j^2} - \frac{1}{2\sigma_o^2} \right) \sum_{x \in \mathcal{X}_j} \|x - \widetilde{\mu}_j\|^2 \right) - n \ln k \\
&\geq \sum_{j=1}^{k} |\mathcal{X}_j| \left(d \ln \frac{\sigma_j}{\sigma_o} - \frac{D^2}{2\sigma_o^2} \right) - n \ln k \;\;\geq\;\; d \ln \frac{\max_{j: \mathcal{X}_j \neq \emptyset} \sigma_j}{\sigma_o} - \frac{nD^2}{2\sigma_o^2} - n \ln k.
\end{aligned}$$

In the second-last line, we have exploited the fact that $\widetilde{\mu}_j$ is the mean of cluster $\mathcal{X}_j$, so that $\sum_{x \in \mathcal{X}_j} \|x - \widetilde{\mu}_j\|^2 \leq \sum_{x \in \mathcal{X}_j} \|x - \mu_j\|^2$, and for the last line we have used $\|x - \widetilde{\mu}_j\| \leq D$.

The difference above is at most Δ, and thus for each nonempty cluster $\mathcal{X}_j$,

$$d \ln \frac{\sigma_j}{\sigma_o} - \frac{nD^2}{2\sigma_o^2} - n \ln k \leq \Delta,$$

whereupon $\sigma_j^2 \leq \sigma_o^2(1+\delta)$ given the bound (3) on the dimension d. ■

This observation allows us to prove following analog of Lemma 2.

Lemma 9 *Following the terminology of Lemma 8, pick $\delta, \Delta > 0$ and suppose that the dimension satisfies (3). Pick any admissible mixture $(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma})$ whose log-likelihood is within an additive factor Δ of the optimal. Then*

$$\Phi(\boldsymbol{\mu}) \;\leq\; (1+\delta) \left(2\sigma_o^2 (\Delta + 2n \ln k) + \Phi_{OPT} \right).$$

Proof Let $(\mathcal{X}_1, \ldots, \mathcal{X}_k)$ be the hard partition of the data set induced by $(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma})$. By Lemma 8, we know that for any nonempty cluster $\mathcal{X}_j$, the variance σ_j^2 is at most $(1+\delta)\sigma_o^2$.

Thus, using Lemma 5, we have

$$\begin{aligned} LL(\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\sigma}) &\leq n\ln k + \sum_{j=1}^{k}\sum_{x\in\mathcal{X}_j}\left(\ln\pi_j + \frac{d}{2}\ln\frac{1}{2\pi\sigma_j^2} - \frac{\|x-\mu_j\|^2}{2\sigma_j^2}\right) \\ &\leq n\ln k + \sum_{j=1}^{k}\left(\frac{|\mathcal{X}_j|d}{2}\ln\frac{1}{2\pi\sigma_j^2} - \frac{1}{2\sigma_j^2}\sum_{x\in\mathcal{X}_j}\|x-\mu_j\|^2\right) \\ &\leq n\ln k + \frac{nd}{2}\ln\frac{1}{2\pi\sigma_o^2} - \frac{1}{2(1+\delta)\sigma_o^2}\sum_{j=1}^{k}\sum_{x\in\mathcal{X}_j}\|x-\mu_j\|^2 \\ &\leq n\ln k + \frac{nd}{2}\ln\left(\frac{1}{2\pi\sigma_o^2}\right) - \frac{\Phi(\boldsymbol{\mu})}{2(1+\delta)\sigma_o^2} \end{aligned}$$

Let $\mu_1^*,\ldots,\mu_k^*$ be an optimal k-means solution and let $(\mathcal{X}_1^*,\ldots,\mathcal{X}_k^*)$ be the hard partition of the data set induced by the mixture model $(\boldsymbol{\pi}^*,\boldsymbol{\mu}^*,\boldsymbol{\sigma}^*)$ where $\pi_j^* = 1/k$ and $\sigma_j^* = \sigma_o$ for all j. Again using Lemma 5,

$$\begin{aligned} LL(\boldsymbol{\pi}^*,\boldsymbol{\mu}^*,\boldsymbol{\sigma}^*) &\geq \sum_{j=1}^{k}\sum_{x\in\mathcal{X}_j^*}\left(\ln\pi_j^* + \frac{d}{2}\ln\frac{1}{2\pi\sigma_j^{*2}} - \frac{\|x-\mu_j^*\|^2}{2\sigma_j^{*2}}\right) \\ &= -n\ln k + \frac{nd}{2}\ln\left(\frac{1}{2\pi\sigma_o^2}\right) - \frac{\Phi(\boldsymbol{\mu}^*)}{2\sigma_o^2} \end{aligned}$$

Then by the near-optimality of $(\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\sigma})$, we have

$$\begin{aligned} \Delta &\geq LL_{OPT} - LL(\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\sigma}) \\ &\geq LL(\boldsymbol{\pi}^*,\boldsymbol{\mu}^*,\boldsymbol{\sigma}^*) - LL(\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\sigma}) \\ &\geq \left(-n\ln k + \frac{nd}{2}\ln\frac{1}{2\pi\sigma_o^2} - \frac{\Phi(\boldsymbol{\mu}^*)}{2\sigma_o^2}\right) - \left(n\ln k + \frac{nd}{2}\ln\left(\frac{1}{2\pi\sigma_o^2}\right) - \frac{\Phi(\boldsymbol{\mu})}{2(1+\delta)\sigma_o^2}\right) \\ &= \frac{\Phi(\boldsymbol{\mu})}{2(1+\delta)\sigma_o^2} - \frac{\Phi_{OPT}}{2\sigma_o^2} - 2n\ln k \end{aligned}$$

Rearranging gives the theorem statement. ■

3.2 Proof of Theorem 4

Once again we reduce from k-means, using the hardness result of Aloise et al. (2009), summarized in Theorem 6. Recall that the family of instances for which k-means was shown to be hard has $k = 2$, $d = \text{poly}(n)$, and points with $\{-1, 0, 1\}$-valued coordinates.

Starting with such an instance $x_1,\ldots,x_n \in \mathbb{R}^d$, we show how MOG can be used to find a $(1 + 1/\beta(n))$-approximate solution to k-means.

- Let D denote the diameter of the points; it is polynomial in n.
- Set $\delta = 1/(5\beta(n))$ and
$$\sigma_o^2 = \frac{\delta}{2(1+2n\ln k)}.$$

- Pad the points with zero-valued coordinates to bring the dimension up to at least
$$d = \frac{4}{\delta}\left(\frac{nD^2}{2\sigma_o^2} + n\ln k + 1\right).$$
- Invoke MOG on these modified points, with target precision $b = 1$ and variance lower bound σ_o^2. This returns a mixture $(\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\sigma})$ whose log-likelihood is at least $LL_{OPT} - 1$, subject to the variance constraint.
- Return centers $\boldsymbol{\mu}$.

Lemma 9, with $\Delta = 1$, asserts that

$$\Phi(\boldsymbol{\mu}) \;\leq\; (1+\delta)(2\sigma_o^2(1+2n\ln k) + \Phi_{OPT}) \;\leq\; (1+\delta)(\delta + \Phi_{OPT}) \;\leq\; (1+5\delta)\Phi_{OPT},$$

which is at most $(1 + 1/\beta(n))\Phi_{OPT}$. For the last inequality, we have used the fact that $\Phi_{OPT} \geq 1/2$ since all interpoint distances are ≥ 1.

Acknowledgments

The authors are grateful to the anonymous reviewers for their feedback and to the NSF for support under grants IIS-1162581 and DGE-1144086.

References

D. Aloise, A. Deshpande, P. Hansen, and P. Popat. NP-hardness of Euclidean sum-of-squares clustering. *Machine Learning*, 75(2):245–248, 2009.

S. Arora and R. Kannan. Learning mixtures of arbitrary Gaussians. In *Proceedings of the 33rd ACM Symposium on Theory of Computing*, pages 247–257, 2001.

P. Awasthi, M. Charikar, R. Krishnaswamy, and A. Sinop. The hardness of approximation of Euclidean k-means. In *Proceedings of the 31st International Symposium on Computational Geometry*, 2015.

M. Belkin and K. Sinha. Polynomial learning of distribution families. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, pages 103–112, 2010.

S. Dasgupta and Y. Freund. Random projection trees for vector quantization. *IEEE Transactions on Information Theory*, 55(7):3229–3242, 2009.

A.P. Dempster, N. M. Laird, and D. B. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, 39:1–38, 1977.

M. Hardt and E. Price. Tight bounds for learning a mixture of two Gaussians. In *Proceedings of the 47th Annual ACM on Symposium on Theory of Computing*, pages 753–760, 2015.

D.J. Hsu and S.M. Kakade. Learning mixtures of spherical Gaussians: Moment methods and spectral decompositions. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, pages 11–20, 2013.

M. Mahajan, P. Nimbhorkar, and K. Varadarajan. The planar k-means problem is NP-hard. In *Proceedings of the 3rd International Workshop on Algorithms and Computation*, pages 274–285, 2009.

A. Moitra and G. Valiant. Settling the polynomial learnability of mixtures of Gaussians. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, pages 93–102, 2010.

Journal of Machine Learning Research 18 (2018) 1-36 Submitted 10/16; Revised 8/17; Published 4/18

The DFS Fused Lasso: Linear-Time Denoising over General Graphs

Oscar Hernan Madrid Padilla OMADRID@BERKELEY.EDU
Department of Statistics
University of California
Berkeley, CA 94720, USA

James Sharpnack JSHARPNA@UCDAVIES.EDU
Department of Statistics
University of California
Davis, CA 95616, USA

James G. Scott JAMES.SCOTT@MCCOMBS.UTEXAS.EDU
Department of Information, Risk,
and Operations Management;
Department of Statistics and Data Sciences
University of Texas
Austin, TX 78712, USA

Ryan J. Tibshirani RYANTIBS@CMU.EDU
Machine Learning Department; Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Editor: Massimiliano Pontil

Abstract

The fused lasso, also known as (anisotropic) total variation denoising, is widely used for piecewise constant signal estimation with respect to a given undirected graph. The fused lasso estimate is highly nontrivial to compute when the underlying graph is large and has an arbitrary structure. But for a special graph structure, namely, the chain graph, the fused lasso—or simply, 1d fused lasso—can be computed in linear time. In this paper, we revisit a result recently established in the online classification literature (Herbster et al., 2009; Cesa-Bianchi et al., 2013) and show that it has important implications for signal denoising on graphs. The result can be translated to our setting as follows. Given a general graph, if we run the standard depth-first search (DFS) traversal algorithm, then the total variation of any signal over the chain graph induced by DFS is no more than twice its total variation over the original graph.

This result leads to several interesting theoretical and computational conclusions. Letting m and n denote the number of edges and nodes, respectively, of the graph in consideration, it implies that for an underlying signal with total variation t over the graph, the fused lasso (properly tuned) achieves a mean squared error rate of $t^{2/3}n^{-2/3}$. Moreover, precisely the same mean squared error rate is achieved by running the 1d fused lasso on the DFS-induced chain graph. Importantly, the latter estimator is simple and computationally cheap, requiring $O(m)$ operations to construct the DFS-induced chain and $O(n)$ operations to compute the 1d fused lasso solution over this chain. Further, for trees that have bounded maximum degree, the error rate of $t^{2/3}n^{-2/3}$ cannot be improved, in the sense that it is the minimax rate for signals that have total variation t over the tree. Finally, several related results also hold—for example, the analogous result holds for a roughness measure defined by the ℓ_0 norm of differences across edges in place of the total variation metric.

Keywords: fused lasso, total variation denoising, graph denoising, depth-first search

1. Introduction

We study the graph denoising problem, i.e., estimation of a signal $\theta_0 \in \mathbb{R}^n$ from noisy data

$$y_i = \theta_{0,i} + \epsilon_i, \quad i = 1, \ldots, n, \tag{1}$$

when the components of θ_0 are associated with the vertices of an undirected, connected graph $G = (V, E)$. Without a loss of generality, we denote $V = \{1, \ldots, n\}$. Versions of this problem arise in diverse areas of science and engineering, such as gene expression analysis, protein mass spectrometry, and image denoising. The problem is also archetypal of numerous internet-scale machine learning tasks that involve propagating labels or information across edges in a network (e.g., a network of users, web pages, or YouTube videos).

Methods for graph denoising have been studied extensively in machine learning and signal processing. In machine learning, graph kernels have been proposed for classification and regression, in both supervised and semi-supervised data settings (e.g., Belkin and Niyogi (2002); Smola and Kondor (2003); Zhu et al. (2003); Zhou et al. (2005)). In signal processing, a considerable focus has been placed on the construction of wavelets over graphs (e.g., Crovella and Kolaczyk (2003); Coifman and Maggioni (2006); Gavish et al. (2010); Hammond et al. (2011); Sharpnack et al. (2013); Shuman et al. (2013)). We will focus our study on the *fused lasso* over graphs, also known as (anisotropic) *total variation* denoising over graphs. Proposed by Rudin et al. (1992) in the signal processing literature, and Tibshirani et al. (2005) in the statistics literature, the fused lasso estimate is defined by the solution of a convex optimization problem,

$$\hat{\theta}_G = \underset{\theta \in \mathbb{R}^n}{\operatorname{argmin}} \ \frac{1}{2}\|y - \theta\|_2^2 + \lambda \|\nabla_G \theta\|_1, \tag{2}$$

where $y = (y_1, \ldots, y_n) \in \mathbb{R}^n$ the vector of observed data, $\lambda \geq 0$ is a tuning parameter, and $\nabla_G \in \mathbb{R}^{m \times n}$ is the edge incidence matrix of the graph G. Note that the subscript on the incidence matrix ∇_G and the fused lasso solution $\hat{\theta}_G$ in (2) emphasize that these quantities are defined with respect to the graph G. The edge incidence matrix ∇_G can be defined as follows, using some notation and terminology from algebraic graph theory (e.g., Godsil and Royle (2001)). First, we assign an arbitrary orientation to edges in the graph, i.e., for each edge $e \in E$, we arbitrarily select one of the two joined vertices to be the head, denoted e^+, and the other to be the tail, denoted e^-. Then, we define a row $(\nabla_G)_e$ of ∇_G, corresponding to the edge e, by

$$(\nabla_G)_{e,e^+} = 1, \ \ (\nabla_G)_{e,e^-} = -1, \ \ (\nabla_G)_{e,v} = 0 \ \text{ for all } v \neq e^+, e^-,$$

for each $e \in E$. Hence, for an arbitrary $\theta \in \mathbb{R}^n$, we have

$$\|\nabla_G \theta\|_1 = \sum_{e \in E} |\theta_{e^+} - \theta_{e^-}|.$$

We can see that the particular choice of orientation does not affect the value $\|\nabla_G \theta\|_1$, which we refer to as the *total variation* of θ over the graph G.

1.1 Summary of results

We will wait until Section 1.3 to give a detailed review of literature, both computational and theoretical, on the fused lasso. Here we simply highlight a key computational aspect of the fused lasso to motivate the main results in our paper. The fused lasso solution in (2), for a graph G of arbitrary structure, is highly nontrivial to compute. For a chain graph, however, the fused lasso solution can be computed in linear time (e.g., using dynamic programming or specialized taut-string methods).

The question we address is: how can we use this fact to our advantage, when seeking to solve (2) over an arbitrary graph? Given a generic graph structure G that has m edges and n nodes, it is obvious that we can define a chain graph based on the ordering of nodes produced by depth-first search (DFS). Far less obvious is that for any signal, its total variation along the DFS-induced chain graph never exceeds twice its total variation over the original graph. This fact follows closely from a similar result found in the online graph-structured classification literature (Herbster et al., 2009; Cesa-Bianchi et al., 2013), and for our purposes, it has three notable consequences, described below.

1. No matter the structure of G, we can denoise any signal defined over this graph in $O(m+n)$ operations: $O(m)$ operations for DFS and $O(n)$ operations for the 1d fused lasso on the induced chain. We call the corresponding estimator—the 1d fused lasso run on the DFS-induced chain—the *DFS fused lasso*.

2. For an underlying signal θ_0 that generates the data, as in (1), such that $\theta_0 \in \mathrm{BV}_G(t)$, where $\mathrm{BV}_G(t)$ is the class of signals with total variation at most t, defined in (4), the DFS fused lasso estimator has mean squared error (MSE) on the order of $t^{2/3}n^{-2/3}$.

3. For an underlying signal $\theta_0 \in \mathrm{BV}_G(t)$, the fused lasso estimator over the original graph, in (2), also has MSE on the order of $t^{2/3}n^{-2/3}$.

The fact that such a fast rate, $t^{2/3}n^{-2/3}$, applies for the fused lasso estimator over *any* connected graph structure is somewhat surprising. It implies that the chain graph represents the hardest graph structure for denoising signals of bounded variation—at least, hardest for the fused lasso, since as we have shown, error rates on general connected graphs can be no worse than the chain rate of $t^{2/3}n^{-2/3}$.

We also complement these MSE upper bounds with the following minimax lower bound over trees.

4. When G is a tree of bounded max degree, the minimax MSE over the class $\mathrm{BV}_G(t)$ scales at the rate $t^{2/3}n^{-2/3}$. Hence, in this setting, the DFS fused lasso estimator attains the optimal rate, as does the fused lasso estimator over G.

Lastly, we prove the following for signals with a bounded number of nonzero edge differences.

5. For an underlying signal $\theta_0 \in \mathrm{BD}_G(s)$, where $\mathrm{BD}_G(s)$ is the class of signals with at most s nonzero edge differences, defined in (5), the DFS fused lasso (under a condition on the spacing of nonzero differences over the DFS-induced chain) has MSE on the order of $s(\log s + \log\log n)\log n/n + s^{3/2}/n$. When G is a tree, the minimax MSE over the class $\mathrm{BD}_G(s)$ scales as $s\log(n/s)/n$. Thus, in this setting, the DFS fused lasso estimator is only off by a $\log\log n$ factor provided that s is small.

Thus DFS fused lasso gives us an $O(n)$ time algorithm for nearly minimax rate-optimal denoising over trees. On paper, this only saves a factor of $O(\log n)$ operations, as recent work (to be described in Section 1.3) has produced an $O(n\log n)$ time algorithm for the fused lasso over trees, by extending earlier dynamic programming ideas over chains. However, dynamic programming on a tree is (a) much more complex than dynamic programming on a chain (since it relies on sophisticated data structures), and (b) noticeably slower in practice than dynamic programming over a chain, especially for large problem sizes. Hence there is still a meaningful difference—both in terms of simplicity and practical computational efficiency—between the DFS fused lasso estimator and the fused lasso over a generic tree.

For a general graph structure, we cannot claim that the statistical rates attained by the DFS fused lasso estimator are optimal, nor can we claim that they match those of fused lasso over the original graph. As an example, recent work (to be discussed in Section 1.3) studying the fused lasso over grid graphs shows that estimation error rates for this problem can be much faster than those attained by the DFS fused lasso (and thus the minimax rates over trees). What should be emphasized, however, is that the DFS fused lasso can still be a practically useful method for any graph, running in linear time (in the number of edges) no matter the graph structure, a scaling that is beneficial for truly large problem sizes.

1.2 Assumptions and notation

Our theory will be primarily phrased in terms of the mean squared error (MSE) for an estimator $\hat{\theta}$ of the mean parameter θ_0 in (1), assuming that $\epsilon = (\epsilon_1, \ldots, \epsilon_n)$ has i.i.d. mean zero sub-Gaussian components, i.e.,

$$\mathbb{E}(\epsilon_i) = 0, \quad \text{and} \quad \mathbb{P}(|\epsilon_i| > t) \leq M\exp\big(-t^2/(2\sigma^2)\big), \text{ all } t \geq 0, \quad \text{for } i = 1, \ldots, n, \tag{3}$$

for constants $M, \sigma > 0$. The MSE of $\hat{\theta}$ will be denoted, with a slight abuse of notation, by

$$\|\hat{\theta} - \theta_0\|_n^2 = \frac{1}{n}\|\hat{\theta} - \theta_0\|_2^2.$$

(In general, for a vector $x \in \mathbb{R}^n$, we denote its scaled ℓ_2 norm by $\|x\|_n = \|x\|_2/\sqrt{n}$.) Of course, the MSE will depend not only on the estimator $\hat{\theta}$ in question but also on the assumptions that we make about θ_0. We will focus our study on two classes of signals. The first is the *bounded variation class*, defined with respect to the graph G, and a radius parameter $t > 0$, as

$$\mathrm{BV}_G(t) = \{\theta \in \mathbb{R}^n : \|\nabla_G\theta\|_1 \leq t\}. \tag{4}$$

The second is the *bounded differences class*, defined again with respect to the graph G, and a now a sparsity parameter $s > 0$, as

$$\mathrm{BD}_G(s) = \{\theta \in \mathbb{R}^n : \|\nabla_G\theta\|_0 \leq s\}. \tag{5}$$

We call measure of roughness used in the bounded differences class the *cut metric*, given by replacing the ℓ_1 norm used to define the total variation metric by the ℓ_0 norm, i.e.,

$$\|\nabla_G \theta\|_0 = \sum_{e \in E} 1\{\theta_{e^+} \neq \theta_{e^-}\},$$

which counts the number of nonzero edge differences that appear in θ. Hence, we may think of the former class in (4) as representing a type of weak sparsity across these edge differences, and the latter class in (5) as representing a type of strong sparsity in edge differences.

When dealing with the chain graph, on n vertices, we will use the following modifications to our notation. We write $\nabla_{1\mathrm{d}} \in \mathbb{R}^{(n-1)\times n}$ for the edge incidence matrix of the chain, i.e.,

$$\nabla_{1\mathrm{d}} = \begin{bmatrix} -1 & 1 & 0 & \ldots & 0 \\ 0 & -1 & 1 & \ldots & 0 \\ \vdots & & \ddots & \ddots & \\ 0 & 0 & \ldots & -1 & 1 \end{bmatrix}. \tag{6}$$

We also write $\hat{\theta}_{1\mathrm{d}}$ for the solution of the fused lasso problem in (2) over the chain, also called the *1d fused lasso* solution, i.e., to be explicit,

$$\hat{\theta}_{1\mathrm{d}} = \operatorname*{argmin}_{\theta \in \mathbb{R}^n} \frac{1}{2}\|y - \theta\|_2^2 + \lambda \sum_{i=1}^{n-1} |\theta_{i+1} - \theta_i|. \tag{7}$$

We write $\mathrm{BV}_{1\mathrm{d}}(t)$ and $\mathrm{BD}_{1\mathrm{d}}(s)$ for the bounded variation and bounded differences classes with respect to the chain, i.e., to be explicit,

$$\mathrm{BV}_{1\mathrm{d}}(t) = \{\theta \in \mathbb{R}^n : \|\nabla_{1\mathrm{d}}\theta\|_1 \leq t\},$$
$$\mathrm{BD}_{1\mathrm{d}}(s) = \{\theta \in \mathbb{R}^n : \|\nabla_{1\mathrm{d}}\theta\|_0 \leq s\}.$$

Lastly, in addition to the standard notation $a_n = O(b_n)$, for sequences a_n, b_n such that a_n/b_n is upper bounded for n large enough, we use $a_n \asymp b_n$ to denote that both $a_n = O(b_n)$ and $a_n^{-1} = O(b_n^{-1})$. Also, for random sequences A_n, B_n, we use $A_n = O_{\mathbb{P}}(B_n)$ to denote that A_n/B_n is bounded in probability.

1.3 Related work

Since its inception in the signal processing and statistics communities in Rudin et al. (1992) and Tibshirani et al. (2005), respectively, there has been an impressive amount of work on total variation penalization and the fused lasso. We do not attempt to give a complete coverage, but point out some relevant computational and theoretical advances, covering the two categories separately.

Computational. On the computational side, it is first worth pointing out that there are multiple efficient algorithms for solving the fused lasso problem over a chain graph, i.e., the 1d fused lasso problem. Davies and Kovac (2001) derived an algorithm based on a "taut string" perspective that solves the 1d fused lasso problem in $O(n)$ time (but, the fact

that their taut string method solves the 1d fused lasso problem was not explicitly stated in the work). This was later extended by Condat (2012); Barbero and Sra (2014) to allow for arbitrary weights in both of the individual penalty and loss terms. Johnson (2013) proposed an entirely different $O(n)$ time algorithm for the fused lasso based on dynamic programming. The taut string and dynamic programming algorithms are extremely fast in practice (e.g., they can solve a 1d fused lasso problem with n in the tens of millions in just a few seconds on a standard laptop).

Kolmogorov et al. (2016) extended the dynamic programming approach of Johnson (2013) to solve the fused lasso problem on a tree. Their algorithm is theoretically very efficient, with $O(n \log n)$ running time, but the implementation that achieves this running time (we have found) can be practically slow for large problem sizes, compared to dynamic programming on a chain graph. Alternative implementations are possible, and may well improve practical efficiency, but as far as we see it, they will all involve somewhat sophisticated data structures in the "merge" steps in the forward pass of dynamic programming.

Barbero and Sra (2014) extended (though not in the same direct manner) fast 1d fused lasso optimizers to work over grid graphs, using operator splitting techniques like Douglas-Rachford splitting. Their techniques appear to be quite efficient in practice, and the authors provide thorough comparisons and a thorough literature review of related methods. Over general graphs structures, many algorithms have been proposed, e.g., to highlight a few: Chambolle and Darbon (2009) described a direct algorithm based on a reduction to parametric max flow programming; Hoefling (2010); Tibshirani and Taylor (2011) gave solution path algorithms (tracing out the solution in (2) over all $\lambda \in [0, \infty]$); Chambolle and Pock (2011) described what can be seen as a kind of preconditioned ADMM-style algorithm; Kovac and Smith (2011) described an active set approach; Tansey and Scott (2015) leveraged fast 1d fused lasso solvers in an ADMM decomposition over trails of the graph; most recently, Landrieu and Obozinski (2015) derived a new method based on graph cuts. We emphasize that, even with the advent of these numerous clever computational techniques for the fused lasso over general graphs, it is still far slower to solve the fused lasso over an arbitrary graph than it is to solve the fused lasso over a chain.

Theoretical. On the theoretical side, it seems that the majority of statistical theory on the fused lasso can be placed into two categories: analysis of changepoint recovery, and analysis of MSE. Some examples of works focusing on changepoint recovery are Rinaldo (2009); Harchaoui and Levy-Leduc (2010); Qian and Jia (2012); Rojas and Wahlberg (2014). The statistical theory will concern MSE rates, and hence we give a more detailed review of related literature for this topic.

We begin with results for chain graphs. Mammen and van de Geer (1997) proved, when $\theta_0 \in \mathrm{BV}_{1\mathrm{d}}(t)$, that the 1d fused lasso estimator estimator $\hat{\theta}_{1\mathrm{d}}$ with $\lambda \asymp t^{-1/3} n^{1/3}$ satisfies

$$\|\hat{\theta}_{1\mathrm{d}} - \theta_0\|_n^2 = O_{\mathbb{P}}(t^{2/3} n^{-2/3}). \tag{8}$$

This is indeed the minimax MSE rate for the class $\mathrm{BV}_{1\mathrm{d}}(t)$, as implied by the minimax results in Donoho and Johnstone (1998). (For descriptions of the above upper bound and this minimax rate in a language more in line with that of the current paper, see Tibshirani (2014).) Recently, Lin et al. (2016) improved on earlier results for the bounded differences class in Dalalyan et al. (2014), and proved that when $\theta_0 \in \mathrm{BD}_{1\mathrm{d}}(s)$, the 1d fused lasso

estimator $\hat{\theta}_{1\mathrm{d}}$ with $\lambda \asymp (nW_n)^{1/4}$ satisfies

$$\|\hat{\theta}_{1\mathrm{d}} - \theta_0\|_n^2 = O_{\mathbb{P}}\bigg(\frac{s}{n}\Big((\log s + \log\log n)\log n + \sqrt{n/W_n}\Big)\bigg), \tag{9}$$

where W_n denotes the minimum distance between positions at which nonzero differences occur in θ_0, more precisely, $W_n = \min\{|i-j| : (\nabla_{1\mathrm{d}}\theta_0)_i \neq 0, (\nabla_{1\mathrm{d}}\theta_0)_j \neq 0\}$. When these nonzero differences or "jumps" in θ_0 are evenly spaced apart, we have $W_n \asymp n/s$, and the above becomes, for $\lambda \asymp \sqrt{n}s^{-1/4}$,

$$\|\hat{\theta}_{1\mathrm{d}} - \theta_0\|_n^2 = O_{\mathbb{P}}\bigg(\frac{s(\log s + \log\log n)\log n}{n} + \frac{s^{3/2}}{n}\bigg). \tag{10}$$

This is quite close to the minimax lower bound, whose rate is $s\log(n/s)/n$, that we establish for the class $\mathrm{BD}_{1\mathrm{d}}(s)$, in Theorem 8. (The minimax lower bound that we prove in this theorem actually holds beyond the chain graph, and applies to tree graphs). We can see that the 1d fused lasso rate in (10) is only off by a factor of $\log\log n$, provided that s does not grow too fast (specifically, $s = O((\log n \log\log n)^2)$).

Beyond chain graphs, the story is in general much less clear, however, interesting results are known in special cases. For a d-dimensional grid graph, with $d \geq 2$, Hutter and Rigollet (2016) recently improved on results of Wang et al. (2016), showing that for $\theta_0 \in \mathrm{BV}_G(t) \cap \mathrm{BD}_G(s)$, the fused lasso estimator $\hat{\theta}_G$ over G satisfies

$$\|\hat{\theta}_G - \theta_0\|_n^2 = O_{\mathbb{P}}\bigg(\min\{t, s\}\frac{\log^a n}{n}\bigg). \tag{11}$$

when $\lambda \asymp \log^{a/2} n$, where $a = 2$ if $d = 2$, and $a = 1$ if $d \geq 3$. A minimax lower bound on the MSE rate for the $\mathrm{BV}_G(t)$ class over a grid G of dimension $d \geq 2$ was established to be $t\sqrt{\log(n/t)/n}$, by Sadhanala et al. (2016). This makes the rate achieved by the fused lasso in (11) nearly optimal for bounded variation signals, off by at most a $\log^{3/2} n$ factor when $d = 2$, and a $\log n$ factor when $d \geq 3$.

Wang et al. (2016); Hutter and Rigollet (2016) also derived MSE rates for the fused lasso over several other graph structures, such as Erdos-Renyi random graphs, Ramanujan d-regular graphs, star graphs, and complete graphs. As it is perhaps the most relevant to our goals in this paper, we highlight the MSE bound from Wang et al. (2016) that applies to arbitrary connected graphs. Their Theorem 3 implies, for a generic connected graph G, $\theta_0 \in \mathrm{BV}_G(t)$, that the fused lasso estimator $\hat{\theta}_G$ over G with $\lambda \asymp \sqrt{n\log n}$ satisfies

$$\|\hat{\theta}_G - \theta_0\|_n^2 = O_{\mathbb{P}}\bigg(t\sqrt{\frac{\log n}{n}}\bigg). \tag{12}$$

(See Appendix A.1 for details.) In Theorem 4, we show that the universal $tn^{-1/2}$ rate (ignoring log terms) in (12) for the fused lasso over an arbitrary connected graph can be improved to $t^{2/3}n^{-2/3}$. In Theorem 3, we show that the same rate can indeed be achieved by a simple, linear-time algorithm: the DFS fused lasso.

1.4 Outline

In Section 2, we review a simple but key lemma relating the ℓ_1 norm (and ℓ_0) norm of differences on a tree and a chain induced by running DFS. (This follows closely from an analogous result on binary signals, in Herbster et al. (2009).) We then define the DFS fused lasso estimator. In Section 3, we derive MSE rates for the DFS fused lasso, and the fused lasso over the original graph G under consideration, for signals of bounded variation. We also derive lower bounds for the minimax MSE over trees. In Section 4, we proceed similarly but for signals in the bounded differences class. In Section 5, we present numerical experiments. In Section 6, we summarize our work and also describe some potential extensions.

2. The DFS fused lasso

In this section, we define the DFS-induced chain graph and the DFS fused lasso.

2.1 Tree and chain embeddings

We start by studying some of the fundamental properties associated with total variation on general graphs, and embedded trees and chains. Given a graph $G = (V, E)$, let $T = (V, E_T)$ be an arbitrary spanning tree of G. It is clear that for any signal, its total variation over T is no larger than its total variation over G,

$$\|\nabla_T\theta\|_1 = \sum_{e\in E_T} |\theta_{e^+} - \theta_{e^-}| \leq \sum_{e\in E} |\theta_{e^+} - \theta_{e^-}| = \|\nabla_G\theta\|_1, \quad \text{for all } \theta \in \mathbb{R}^n. \tag{13}$$

The above inequality, albeit very simple, reveals to us the following important fact: if the underlying mean θ_0 in (1) is assumed to be smooth with respect to the graph G, inasmuch as $\|\nabla_G\theta_0\|_1 \leq t$, then it must also be smooth with respect to any spanning tree T of G, since $\|\nabla_T\theta_0\|_1 \leq t$. Roughly speaking, computing the fused lasso solution in (2) over a spanning tree T, instead of G, would therefore still be reasonable for the denoising purposes, as the mean θ_0 would still be smooth over T according to the total variation metric.

The same property as in (14) also holds if we replace total variation by the cut metric:

$$\|\nabla_T\theta\|_0 = \sum_{e\in E_T} 1\{\theta_{e^+} \neq \theta_{e^-}\} \leq \sum_{e\in E} 1\{\theta_{e^+} \neq \theta_{e^-}\} = \|\nabla_G\theta\|_0, \quad \text{for all } \theta \in \mathbb{R}^n. \tag{14}$$

Thus for the mean θ_0, the property $\|\nabla_G\theta_0\|_0 \leq s$ again implies $\|\nabla_T\theta_0\|_0 \leq s$ for any spanning tree T of G, and this would again justify solving the fused lasso over T, in place of G, assuming smoothness of θ_0 with respect to the cut metric in the first place.

Here we go one step further than (13), (14), and assert that analogous properties actually hold for specially embedded chain graphs. The next lemma gives the key result.

Lemma 1 *Let $G = (V, E)$ be a connected graph, where recall we write $V = \{1, \ldots, n\}$. Consider depth-first search (DFS) run on G, and denote by $v_1, \ldots, v_n$ the nodes in the order in which they are reached by DFS. Hence, DFS first visits v_1, then v_2, then v_3, etc. This induces a bijection $\tau : \{1, \ldots, n\} \to \{1, \ldots, n\}$, such that*

$$\tau(i) = v_i, \quad \textit{for all } i = 1, \ldots, n.$$

Let $P \in \mathbb{R}^{n \times n}$ denote the permutation associated with τ. Then it holds that

$$\|\nabla_{1\mathrm{d}} P\theta\|_1 \leq 2\|\nabla_G \theta\|_1, \quad \textit{for all } \theta \in \mathbb{R}^n, \tag{15}$$

as well as

$$\|\nabla_{1\mathrm{d}} P\theta\|_0 \leq 2\|\nabla_G \theta\|_0, \quad \textit{for all } \theta \in \mathbb{R}^n. \tag{16}$$

Proof The proof is simple. Observe that

$$\|\nabla_{1\mathrm{d}} P\theta\|_1 = \sum_{i=1,\ldots,n-1} |\theta_{\tau(i+1)} - \theta_{\tau(i)}|, \tag{17}$$

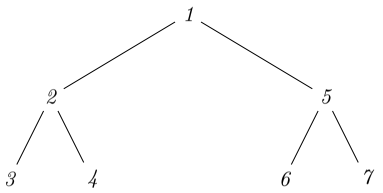
and consider an arbitrary summand $|\theta_{\tau(i+1)} - \theta_{\tau(i)}|$. There are now two cases to examine. First, suppose $\tau(i)$ is not a leaf in the spanning tree constructed by DFS; then there is an edge $e \in E$ such that $\{e^-, e^+\} = \{\tau(i), \tau(i+1)\}$, and $|\theta_{\tau(i+1)} - \theta_{\tau(i)}| = |\theta_{e^+} - \theta_{e^-}|$. Second, suppose that $\tau(i)$ is a leaf node in the DFS tree; then there is a path $p = \{p_1, \ldots, p_r\}$ in the graph such that $p_1 = \tau(i)$, $p_r = \tau(i+1)$, and each $\{p_j, p_{j+1}\} \in E$, $j = 1, \ldots, r-1$, so that by the triangle inequality

$$|\theta_{\tau(i+1)} - \theta_{\tau(i)}| \leq \sum_{j=1}^{r-1} |\theta_{p_{j+1}} - \theta_{p_j}|.$$

Applying this logic over all terms in the sum in (17), and invoking the fundamental property that DFS visits each edge exactly twice (e.g., Chapter 22 of Cormen et al. (2001)), we have established (15). The proof for (16) follows from precisely the same arguments. ■

Remark 2 *Lemma 1 has essentially already appeared in the literature, in a form for binary signals and a roughness metric given by the quadratic form in the graph Laplacian (in place the total variation metric or cut metric), see Herbster et al. (2009). The key idea behind this result and ours is the same, and the proof of Lemma 1 only requires minor modifications. For completeness, we have presented Lemma 1 and its proof anyway. More generally, we should note that graph embeddings—in particular, using chains and trees—are well-known in the online graph-based classification setting. See, e.g., Herbster et al. (2009); Cesa-Bianchi et al. (2013), and references therein.*

Example 1 *The idea behind Lemma 1 can also be clearly demonstrated through an example. We consider G to be a binary tree graph with $n = 7$ nodes, shown below, where we have labeled the nodes according to the order in which they are visited by DFS (i.e., so that here P is the identity).*



In this case,

$$\begin{aligned}
\|\Delta_{1\mathrm{d}}\theta\|_1 &= \sum_{i=1}^{6} |\theta_{i+1} - \theta_i| \\
&\leq |\theta_2 - \theta_1| + |\theta_3 - \theta_2| + \big(|\theta_3 - \theta_2| + |\theta_4 - \theta_2|\big) + \big(|\theta_4 - \theta_2| + |\theta_2 - \theta_1| + |\theta_5 - \theta_1|\big) \\
&\quad + |\theta_6 - \theta_5| + \big(|\theta_6 - \theta_5| + |\theta_7 - \theta_5|\big) \\
&\leq 2 \sum_{e \in G} |\theta_{e^+} - \theta_{e^-}| = 2\|\nabla_G \theta\|_1,
\end{aligned}$$

where in the inequality above, we have used triangle inequality for each term in parentheses individually.

2.2 The DFS fused lasso

We define the DFS fused lasso estimator, $\hat{\theta}_{\mathrm{DFS}}$, to be the fused lasso estimator over the chain graph induced by running DFS on G. Formally, if τ denotes the bijection associated with the DFS ordering (as described in Lemma 1), then the DFS-induced chain graph can be expressed as $C = (V, E_C)$ where $V = \{1, \ldots, n\}$ and $E_C = \{\{\tau(1), \tau(2)\}, \ldots, \{\tau(n-1), \tau(n)\}\}$. Denoting by P the permutation matrix associated with τ, the edge incidence matrix of C is simply $\nabla_C = \nabla_{1\mathrm{d}} P$, and the DFS fused lasso estimator is given by

$$\begin{aligned}
\hat{\theta}_{\mathrm{DFS}} &= \operatorname*{argmin}_{\theta \in \mathbb{R}^n} \frac{1}{2}\|y - \theta\|_2^2 + \lambda \|\nabla_{1\mathrm{d}} P\theta\|_1 \\
&= P^\top \left(\operatorname*{argmin}_{\theta \in \mathbb{R}^n} \frac{1}{2}\|Py - \theta\|_2^2 + \lambda \sum_{i=1}^{n-1} |\theta_{i+1} - \theta_i| \right).
\end{aligned} \tag{18}$$

Therefore, we only need to compute the 1d fused lasso estimator on a permuted data vector Py, and apply the inverse permutation operator $P^\top$, in order to compute $\hat{\theta}_{\mathrm{DFS}}$.

Given the permutation matrix P, the computational cost of (18) is $O(n)$, since, to recall the discussion in Section 1.3, the 1d fused lasso problem (7) can be solved in $O(n)$ operations with dynamic programming or taut string algorithms. The permutation P is obtained by running DFS, which requires $O(m)$ operations, and makes the total computation cost of the DFS fused lasso estimator $O(m+n)$.

It should be noted that, when multiple estimates are desired over the same graph G, we must only run DFS once, and all subsequent estimates on the induced chain require just $O(n)$ operations.

The bounds in (15), (16) for the DFS chain are like those in (13), (14) for spanning trees, and carry the same motivation as that discussed above for spanning trees, beneath (13), (14): if the mean θ_0 is assumed to be smooth with respect to t, insofar as its total variation satisfies $\|\nabla_G \theta_0\|_1 \leq t$, then denoising with respect to C would also be reasonable, in that $\|\nabla_{1\mathrm{d}} P\theta_0\|_1 \leq 2t$; the same can be said for the cut metric. However, it is the rapid $O(m+n)$ computational cost of the DFS fused lasso, and also the simplicity of the dynamic programming and taut string algorithms for the 1d fused lasso problem (7), that make (15), (16) particularly appealing compared to (13), (14). To recall the discussion in Section 1.3, the fused lasso can in principle be computed efficiently over a tree, in $O(n \log n)$ operations

using dynamic programming, but this requires a much more cumbersome implementation and in practice we have found it to be noticeably slower.

2.3 Running DFS on a spanning tree

We can think of the induced chain graph, as described in the last section, as being computed in two steps:

(i) run DFS to compute a spanning tree T of G;

(ii) run DFS on the spanning tree T to define the chain C.

Clearly, this is the same as running DFS on G to define the induced chain C, so decomposing this process into two steps as we have done above may seem odd. But this decomposition provides a useful perspective because it leads to the idea that we could compute the spanning tree T in Step (i) in any fashion, and then proceed with DFS on T in Step (ii) in order to define the chain C. Indeed, any spanning tree in Step (i) will lead to a chain C that has the properties (15), (16) as guaranteed by Lemma 1. This may be of interest if we could compute a spanning tree T that better represents the topology of the original graph G, so that the differences over the eventual chain C better mimicks those over G.

An example of a spanning tree whose topology is designed to reflect that of the original graph is a low-stretch spanning tree. Current interest on low-stretch spanning trees began with the breakthrough results in Elkin et al. (2008); most recently, Abraham and Neiman (2012) showed that a spanning tree with average stretch $O(\log n \log\log n)$ can be computed in $O(m \log n \log\log n)$ operations.

In Section 6.4, we discuss a setting in which the fused lasso problem (2) has arbitrary penalty weights, which gives rise to a weighted graph G. In this setting, an example of a spanning tree that can be crafted so that its edges represent important differences in the original graph is a maximum spanning tree. Prim's and Kruskal's minimum spanning tree algorithms, each of which take $O(m \log n)$ time (Cormen et al., 2001), can be used to compute a maximum spanning tree after we negate all edge weights.

2.4 Averaging multiple DFS estimators

Notice that several DFS-induced chains can be formed from a single seed graph G, by running DFS itself on G with different random starts (or random decisions about which edge to follow at each step in DFS), or by computing different spanning trees T of G (possibly themselves randomized) on which we run DFS, or by some combination, etc. Denoting by $\hat{\theta}^{(1)}_{\mathrm{DFS}}, \hat{\theta}^{(2)}_{\mathrm{DFS}}, \ldots, \hat{\theta}^{(K)}_{\mathrm{DFS}}$ the DFS fused lasso estimators fit to K different induced chains, we might believe that the average estimator, $(1/K)\sum_{k=1}^K \hat{\theta}^{(k)}_{\mathrm{DFS}}$, will have good denoising performance, as it incorporates fusion at each node in multiple directions. In Section 5, we demonstrate that this intuition holds true (at least, across the set of experiments we consider).

3. Analysis for signals of bounded variation

Throughout this section, we assume that the underlying mean θ_0 in (1) satisfies $\theta_0 \in \mathrm{BV}_G(t)$ for a generic connected graph G. We derive upper bounds on the MSE rates of the DFS fused lasso and the fused lasso over G. We also derive a tight lower bound on the minimax MSE when G is a tree that has bounded degree.

3.1 The DFS fused lasso

The analysis for the DFS fused lasso estimator is rather straightforward. By assumption, $\|\nabla_G \theta_0\|_1 \leq t$, and thus $\|\nabla_{1\mathrm{d}} P\theta_0\|_1 \leq 2t$ by (15) in Lemma 1. Hence, we may think of our model (1) as giving us i.i.d. data Py around $P\theta_0 \in \mathrm{BV}_{1\mathrm{d}}(2t)$, and we may apply existing results from Mammen and van de Geer (1997) on the 1d fused lasso for bounded variation signals, as described in (8) in Section 1.3. This establishes the following.

Theorem 3 *Consider a data model* (1)*, with i.i.d. sub-Gaussian errors as in* (3)*, and* $\theta_0 \in \mathrm{BV}_G(t)$*, where* G *is a generic connected graph. Then for any DFS ordering of* G *yielding a permutation matrix* P*, the DFS fused lasso estimator* $\hat{\theta}_{\mathrm{DFS}}$ *in* (18)*, with a choice of tuning parameter* $\lambda \asymp t^{-1/3} n^{1/3}$*, has MSE converging in probability at the rate*

$$\|\hat{\theta}_{\mathrm{DFS}} - \theta_0\|_n^2 = O_{\mathbb{P}}(t^{2/3} n^{-2/3}). \tag{19}$$

We note that, if multiple DFS fused lasso estimators $\hat{\theta}^{(1)}_{\mathrm{DFS}}, \hat{\theta}^{(2)}_{\mathrm{DFS}}, \ldots, \hat{\theta}^{(K)}_{\mathrm{DFS}}$ are computed across multiple different DFS-induced chains on G, then the average estimator clearly satisfies the same bound as in (19),

$$\left\| \frac{1}{K} \sum_{k=1}^K \hat{\theta}^{(k)}_{\mathrm{DFS}} - \theta_0 \right\|_n^2 = O_{\mathbb{P}}(t^{2/3} n^{-2/3}),$$

provided that K is held constant, by the triangle inequality.

3.2 The graph fused lasso

Interestingly, the chain embedding result (15) in Lemma 1 is not only helpful for establishing the MSE rate for the DFS fused lasso estimator in Theorem 3, but it can also be used to improve the best known rate for the original fused lasso estimator over the graph G. In Section 1.3, we described a result (12) that follows from Wang et al. (2016), establishing an MSE rate of $tn^{-1/2}$ rate (ignoring log terms) for the fused lasso estimator over a connected graph G, when $\|\nabla_G \theta_0\|_1 \leq t$. In fact, as we will now show, this can be improved to a rate of $t^{2/3} n^{-2/3}$, just as in (19) for the DFS fused lasso.

Wang et al. (2016) present a framework for deriving fast MSE rates for fused lasso estimators based on entropy. They show in their Lemma 9 that a bound in probability on the sub-Gaussian complexity

$$\max_{x \in \mathcal{S}_G(1)} \frac{\epsilon^\top x}{\|x\|_2^{1-w/2}}, \tag{20}$$

for some $0 < w < 2$, where $\mathcal{S}_G(1) = \{x \in \mathrm{row}(\nabla_G) : \|\nabla_G x\|_1 \leq 1\}$, leads to a bound in probability on the MSE of the fused lasso estimator $\hat{\theta}_G$ over G. (Wang et al. (2016) actually assume Gaussian errors, but their Lemma 9, Theorem 10, Lemma 11, and Corollary 12 still hold for sub-Gaussian errors as in (3)). The sub-Gaussian complexity in (20) is typically controlled via an entropy bound on the class $\mathcal{S}_G(1)$. Typically, one thinks of controlling

entropy by focusing on specific classes of graph structures G. Perhaps surprisingly, Lemma 1 shows we can uniformly control the sub-Gaussian complexity (20) over all connected graphs.

For any DFS-induced chain C constructed from G, note first that

$$\mathrm{row}(\nabla_G) = \mathrm{span}\{\mathbb{1}\}^\perp = \mathrm{row}(\nabla_C),$$

where $\mathbb{1} = (1, \ldots, 1) \in \mathbb{R}^n$ is the vector of all 1s. This, and (15) in Lemma 1, imply that

$$\max_{x \in \mathcal{S}_G(1)} \frac{\epsilon^\top x}{\|x\|_2^{1-w/2}} \leq \max_{x \in \mathcal{S}_C(2)} \frac{\epsilon^\top x}{\|x\|_2^{1-w/2}}.$$

Now, taking $w = 1$,

$$\max_{x \in \mathcal{S}_C(2)} \frac{\epsilon^\top x}{\|x\|_2^{1/2}} = \max_{\substack{x \,:\, \mathbb{1}^\top x = 0, \\ \|\nabla_{1\mathrm{d}} P x\|_1 \leq 2}} \frac{\epsilon^\top x}{\|x\|_2^{1/2}} = \max_{\substack{x \,:\, \mathbb{1}^\top x = 0, \\ \|\nabla_{1\mathrm{d}} x\|_1 \leq 1}} \frac{2^{-1/2}(P\epsilon)^\top x}{\|x\|_2^{1/2}} = O_{\mathbb{P}}(n^{1/4}).$$

The last step (asserting that the penultimate term is $O_{\mathbb{P}}(n^{1/4})$) holds by first noting that $P\epsilon$ is equal in law to ϵ (as we have assumed i.i.d. components of the error vector), and then applying results on the chain graph in Theorem 10, Lemma 11, and Corollary 12 of Wang et al. (2016). Applying Lemma 9 of Wang et al. (2016), we have now established the following result.

Theorem 4 *Consider a data model* (1)*, with i.i.d. sub-Gaussian errors as in* (3)*, and* $\theta_0 \in \mathrm{BV}_G(t)$*, where* G *is a generic connected graph. Then the fused lasso estimator* $\hat{\theta}_G$ *over* G*, in* (2)*, under a choice of tuning parameter* $\lambda \asymp t^{-1/3} n^{1/3}$*, has MSE converging in probability at the rate*

$$\|\hat{\theta}_G - \theta_0\|_n^2 = O_{\mathbb{P}}(t^{2/3} n^{-2/3}). \tag{21}$$

In a sense, the above theorem suggests that the chain graph is among the hardest graphs for denoising bounded variation signals, since the fused lasso estimator on any connected graph G will achieve an MSE rate in that is at least as good as in the chain rate, if not better. In this vein, it is worth emphasizing that the MSE bound in (21) is not tight for certain graph structures; a good example is the 2d grid, where we must compare (21) from the theorem to the known MSE bound in (11) from Hutter and Rigollet (2016), the latter being only log factors from optimal, as shown in Sadhanala et al. (2016). It is natural for the 2d grid graph to consider the scaling $t \asymp \sqrt{n}$ (as argued in Sadhanala et al. (2016)), in which case the rates for the fused lasso estimator are $n^{-1/3}$ from Theorem 4 versus $(\log^2 n) n^{-1/2}$ from Hutter and Rigollet (2016).

3.3 Minimax lower bound over trees

We derive a lower bound for the MSE over the class $\mathrm{BV}_G(t)$ when G is a tree graph. The proof applies Assouad's Lemma (Yu, 1997), over a discrete set of probability measures constructed by a careful partitioning of the vertices of G, that balances both the sizes of each partition element and the number of edges crossing in between partition elements. It is deferred until Appendix A.2.

Theorem 5 *Consider a data model* (1)*, with i.i.d. Gaussian errors* $\epsilon_i \sim N(0, \sigma^2)$*,* $i = 1, \ldots, n$*, and with* $\theta_0 \in \mathrm{BV}_G(t)$*, where* G *is a tree graph, having maximum degree* $d_{\max}$*. Then there exists absolute constants* $N, C > 0$*, such that for* $n/(td_{\max}) > N$*,*

$$\inf_{\hat{\theta}} \sup_{\theta_0 \in \mathrm{BV}_G(t)} \mathbb{E}\|\hat{\theta} - \theta_0\|_n^2 \geq C\left(\frac{t}{\sigma d_{\max}^2 n}\right)^{2/3}. \tag{22}$$

The theorem demonstrates that, for trees of bounded degree, such as the chain and balanced d-ary trees, the fused lasso estimator over the tree achieves achieves the minimax rate, as does the DFS fused lasso.

4. Analysis for signals with bounded differences

We assume that the underlying mean θ_0 in (1) satisfies $\theta_0 \in \mathrm{BD}_G(s)$ for a generic connected graph G. We analyze the MSE of the DFS fused lasso, as well as (a particular formulation of) wavelet denoising over G. We again establish a lower bound on the minimax MSE when G is a tree.

4.1 The DFS fused lasso

As it was for the bounded variation case, the analysis for the DFS fused lasso estimator is straightforward. By assumption, $\|\nabla_G \theta_0\|_0 \leq s$, thus $\|\nabla_{1\mathrm{d}} P\theta_0\|_0 \leq 2s$ by (16) in Lemma 1, and we may think of our model (1) as having i.i.d. data Py around $P\theta_0 \in \mathrm{BD}_{1\mathrm{d}}(2s)$. Applying an existing result on the 1d fused lasso for bounded differences signals, as described in (9), from Lin et al. (2016), gives the following result.

Theorem 6 *Consider a data model* (1)*, with i.i.d. sub-Gaussian errors as in* (3)*, and* $\theta_0 \in \mathrm{BD}_G(s)$*, for a connected graph* G*. Consider an arbitrary DFS ordering of* G*, that defines a permutation matrix* P *and the DFS fused lasso estimator* $\hat{\theta}_{\mathrm{DFS}}$ *in* (18)*. Denote by* $W_n = \min\{|i - j| : (\nabla_{1\mathrm{d}} P\theta_0)_i \neq 0, (\nabla_{1\mathrm{d}} P\theta_0)_j \neq 0\}$ *the minimum distance between positions, measured along the DFS-induced chain, at which nonzero differences or jumps occur in* θ_0*. Then, under a choice of tuning parameter* $\lambda \asymp (nW_n)^{1/4}$*, the DFS fused lasso estimator has MSE converging in probability at the rate*

$$\|\hat{\theta}_{\mathrm{DFS}} - \theta_0\|_n^2 = O_{\mathbb{P}}\left(\frac{s}{n}\left((\log s + \log\log n)\log n + \sqrt{n/W_n}\right)\right). \tag{23}$$

Hence, if the s *jumps along the DFS chain are evenly spaced apart, i.e.,* $W_n \asymp n/s$*, then for* $\lambda \asymp \sqrt{n} s^{-1/4}$*,*

$$\|\hat{\theta}_{\mathrm{DFS}} - \theta_0\|_n^2 = O_{\mathbb{P}}\left(\frac{s(\log s + \log\log n)\log n}{n} + \frac{s^{3/2}}{n}\right). \tag{24}$$

An undesirable feature of applying existing 1d fused lasso results for signals with bounded differences, in the above result, is the dependence on W_n in the DFS fused lasso error bound (23) (we applied the result (9) from Lin et al. (2016), but the bounds from Dalalyan et al. (2014) also depend on W_n, and as far as we can tell, so should any analysis of the 1d fused

lasso for signals with bounded differences). In the 1d setting, assuming that $W_n \asymp n/s$, which says that jumps in θ_0 occur at roughly equally spaced positions, is fairly reasonable; but to assume the same when the jumps are measured with respect to the DFS-induced chain, as we must in order to establish (24), is perhaps not. Even if the differences apparent in θ_0 over edges in G are somehow (loosely speaking) spaced far apart, running DFS could well produce an ordering such that jumps in $P\theta_0$ occur at positions very close together. We reiterate that the MSE bounds for the DFS fused lasso for bounded variation signals, in Theorem 3, do not suffer from any such complications.

4.2 Graph wavelet denoising

We compare the performances of the DFS fused lasso and wavelet denoising using spanning tree wavelets, for signals with bounded differences. For spanning tree wavelets, the construction starts with a spanning tree and carefully defines a hierarchical decomposition by recursively finding and splitting around a balancing vertex, which is a vertex whose adjacent subtrees are of size at most half of the original tree; this decomposition is used to construct an unbalanced Haar wavelet basis, as in Singh et al. (2010). In Sharpnack et al. (2013), it was shown that for any connected graph G, the constructed wavelet basis $W \in \mathbb{R}^{n \times n}$ satisfies

$$\|W\theta\|_0 \leq \lceil \log d_{\max} \rceil \lceil \log n \rceil \|\nabla_G \theta\|_0, \quad \text{for all } \theta \in \mathbb{R}^n, \tag{25}$$

where $d_{\max}$ is the maximum degree of G, and the above holds regardless of choice of spanning tree in the wavelet construction. Now consider the wavelet denoising estimator

$$\hat{\theta}_W = \operatorname*{argmin}_{\theta \in \mathbb{R}^n} \frac{1}{2}\|y - \theta\|_2^2 + \lambda \|W\theta\|_1. \tag{26}$$

The following is an immediate consequence of (25), the fact that the wavelet basis W is orthonormal, and standard results about soft-thresholding (e.g., Lemma 2.8 in (Johnstone, 2011)).

Theorem 7 *Consider a data model (1), with i.i.d. Gaussian errors $\epsilon_i \sim N(0, \sigma^2)$, $i = 1, \ldots, n$, and with $\theta_0 \in \mathrm{BV}_G(t)$, where G is a connected graph, having maximum degree $d_{\max}$. Then the spanning tree wavelet estimator $\hat{\theta}_W$ in (26), with a choice $\lambda \asymp \sqrt{\log n}$, has MSE converging in expectation at the rate*

$$\mathbb{E}\|\hat{\theta}_W - \theta_0\|_n^2 = O\left(\frac{s \log d_{\max} \log^2 n}{n}\right). \tag{27}$$

The result in (27) has the advantage over the DFS fused lasso result in (23) that it does not depend on a hard-to-interpret quantity like W_n, the minimum spacing between jumps along the DFS-induced chain. But when (say) $d_{\max} \asymp 1$, $s \asymp 1$, and we are willing to assume that $W_n \asymp n$ (meaning the jumps of θ_0 occur at positions evenly spaced apart on the DFS chain), we can see that the spanning tree wavelet rate in (27) is just slightly slower than the DFS fused lasso rate in (24), by a factor of $\log n / \log\log n$.

While the comparison between the DFS fused lasso and wavelet rates, (23) and (27), show an advantage to spanning tree wavelet denoising, as it does not require assumptions about the spacings between nonzero differences in θ_0, we have found nonetheless that the

DFS fused lasso to performs well in practice compared to spanning tree wavelets, and indeed often outperforms the latter in terms of MSE. Experiments comparing the two methods are presented Section 5.

4.3 Minimax lower bound for trees

We now derive a lower bound for the MSE over the class $\mathrm{BD}_G(s)$ when G is a tree graph. The proof relates the current denoising problem to one of estimating sparse normal means, with a careful construction of the sparsity set using degree properties of trees. It is deferred until Appendix A.3.

Theorem 8 *Consider a data model (1), with i.i.d. Gaussian errors $\epsilon_i \sim N(0, \sigma^2)$, $i = 1, \ldots, n$, and with $\theta_0 \in \mathrm{BD}_G(s)$, where G is a tree. Then there are absolute constants $N, C > 0$, such that for $n/s > N$,*

$$\inf_{\hat{\theta}} \sup_{\theta_0 \in \mathrm{BD}_G(s)} \mathbb{E}\|\hat{\theta} - \theta_0\|_n^2 \geq C\sigma^2 \frac{s}{n} \log\left(\frac{n}{s}\right). \tag{28}$$

The MSE lower bound in (28) shows that, when we are willing to assume that $W_n \asymp n/s$ in the DFS-induced chain, the DFS fused lasso estimator is a $\log\log n$ factor away from the optimal rate, provided that s is not too large, namely $s = O((\log n \log\log n)^2)$. The spanning tree wavelet estimator, on the other hand, is a $\log n$ factor from optimal, without any real restrictions on s, i.e., it suffices to have $s = O(n^a)$ for some $a > 0$. It is worth remarking that, for large enough s, the lower bound in (28) is perhaps not very interesting, as in such a case, we may as well consider the bounded variation lower bound in (22), which will likely be tighter (faster).

5. Experiments

In this section we compare experimentally the speed and accuracy of two approaches for denoising signals on graphs: the graph fused lasso, and the fused lasso along the chain graph induced by a DFS ordering. In our experiments, we see that the DFS-based denoiser sacrifices a modest amount in terms of mean squared error, while providing gains (sometimes considerable) in computational speed. This shows that our main theorem, in addition to providing new insights on MSE rates for the graph fused lasso, also has important practice consequences. For truly massive problems, where the full graph denoising problem is impractical to solve, we may use the linear-time DFS fused lasso denoiser, and obtain a favorable tradeoff of accuracy for speed.

5.1 Generic graphs

We begin by considering three examples of large graphs of (more or less) generic structure, derived from road networks in three states: California, Pennsylvania, and Texas. Data on these road networks are freely available at `https://snap.stanford.edu`. In these networks, intersections and endpoints are represented by nodes, and roads connecting these intersections or endpoints are represented by undirected edges; see Leskovec et al. (2009) for more details. For each network, we use the biggest connected component as our graph structure to run comparisons. The graph corresponding to California has $n = 1957027$ nodes

and $m = 2760388$ edges, the one for Pennsylvania has $n = 1088092$ nodes and $m = 1541898$ edges, and the graph for Texas has $n = 1351137$ nodes and $m = 1879201$ edges. We compare Laplacian smoothing versus the fused lasso over a DFS-induced chain, on the graphs from the three states. We do not compare with the fused lasso over the original graphs, due to its prohibitive computational cost at such large scales.

We used the following procedure to construct a synthetic signal $\theta_0 \in \mathbb{R}^n$ on each of the road network graphs, of piecewise constant nature:

- an initial seed node v_1 is selected uniformly at random from the nodes $V = \{1, \dots, n\}$ in the graph;
- a component C_1 is formed based on the $\lfloor n/10 \rfloor$ nodes closest to v_1 (where the distance between two nodes in the graph is given by the length of the shortest path between them);
- a second seed node v_2 is selected uniformly at random from $G \setminus C_1$;
- a component C_2 is formed based on the $\lfloor n/10 \rfloor$ nodes closest to v_2 (again in shortest path distance);
- this process is repeated[1] until we have a partition $C_1, \dots, C_{10}$ of the node set V into components of (roughly) equal size, and $\theta_0 \in \mathbb{R}^n$ is defined to take constant values on each of these components.

In our experiments, we considered 20 values of the total variation for the underlying signal. For each, the signal θ_0 was scaled appropriately to achieve the given total variation value, and data $y \in \mathbb{R}^n$ was generated by adding i.i.d. $N(0, 0.2^2)$ noise to the components of θ_0. For each data instance y, the DFS fused lasso and Laplacian smoothing estimators, the former defined by (18) and the latter by

$$\hat{\theta}_{\text{Lap}} = \operatorname*{argmin}_{\theta \in \mathbb{R}^n} \frac{1}{2}\|y - \theta\|_2^2 + \lambda \theta^\top L_G \theta, \tag{29}$$

where $L_G = \nabla_G^\top \nabla_G$ is the Laplacian matrix of the given graph G, and each estimator is computed over 20 values of its own tuning parameter. Then, the value of the tuning parameter minimizing the average MSE, over 50 draws of data y around θ_0, was selected for each method. Finally, this optimized MSE, averaged over the 50 draws of data y, and further, over 10 repetitions of the procedure for constructing the signal θ_0 explained above, was recorded. Figure 1 displays the optimized MSE for the DFS fused lasso and Laplacian smoothing, as the total variation of the underlying signal varies, for the three road network graphs.

As we can see from the figure, for low values of the underlying total variation, i.e., low signal-to-noise ratio (SNR) levels, Laplacian smoothing and the DFS fused lasso, each tuned to optimality, perform about the same. This is because at low enough SNR levels, each will be approximating θ_0 by something like $\bar{y}\mathbb{1}$, with $\bar{y}$ being the sample average of the data vector y. But as the SNR increases, we see that the DFS fused lasso outpeforms

1. Here, whenever isolated small components are unintentionally created, we add them to the big components

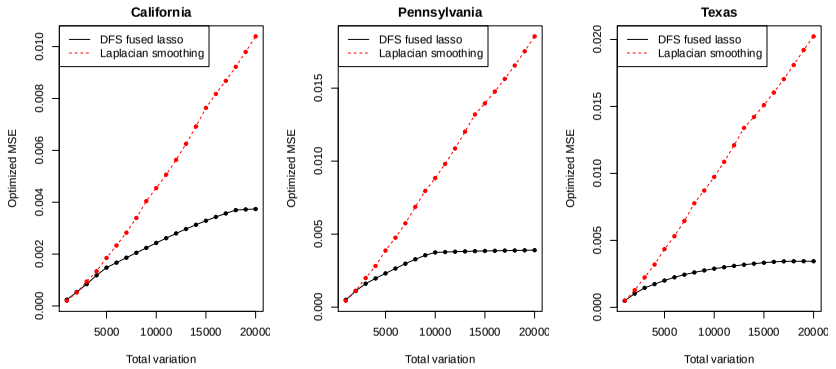


Figure 1: *The optimized MSE for the DFS fused lasso and Laplacian smoothing (i.e., MSE achieved by these methods under optimal tuning) is plotted as a function of the total variation of the underlying signal, for each of the three road network graphs. This has been averaged over 50 draws of data y for each construction of the underlying signal θ_0, and 10 repetitions in constructing θ_0 itself. For low values of the underlying total variation, i.e., low SNR levels, the two methods perform about the same, but as the SNR increases, the DFS fused lasso outperforms Laplacian smoothing by a considerable margin.*

Laplacian smoothing by a considerable amount. This might seem surprising, as Laplacian smoothing uses information from the entire graph, whereas the DFS fused lasso reduces the rich structure of the road network graph in each case to that of an embedded chain. However, Laplacian smoothing is a linear smoother (meaning that $\hat{\theta}_{\text{Lap}}$ in (29) is a linear function of the data y), and therefore it comes with certain limitations when estimating signals of bounded variation (e.g., see the seminal work of Donoho and Johnstone (1998), and the more recent graph-based work of Sadhanala et al. (2016)). In contrast, the DFS fused lasso is a nonlinear estimator, and while it discards some information in the original graph structure, it retains enough of the strong adaptivity properties of the fused lasso over the original graph to statistically dominate a linear estimator like Laplacian smoothing.

Lastly, in terms of computational time, it took an average of 82.67 seconds, 44.02 seconds, and 54.49 seconds to compute the 20 DFS fused lasso solutions (i.e., over the 20 tuning parameter values) for the road network graphs from California, Pennsylvania, and Texas, respectively (the averages are taken over the 50 draws of data y around each signal θ_0, and the 10 repetitions in constructing θ_0). By comparison, it took an average of 2748.26 seconds, 1891.97 seconds, and 1487.36 seconds to compute the 20 Laplacian smoothing solutions for the same graphs. The computations and timings were performed on a standard laptop computer (with a 2.80GHz Intel Core i7-2640M processor). For the DFS fused lasso, in each problem instance, we first computed a DFS ordering using the `dfs` function from the R package `igraph`, which is an R wrapper for a C++ implementation of DFS, and initialized the algorithm at a random node for the root. We then computed the appropriate 1d fused lasso solutions using the `trendfilter` function from the R package `glmgen`, which

is an R wrapper for a C++ implementation of the fast (linear-time) dynamic programming algorithm in Johnson (2013). For Laplacian smoothing, we used the `solve` function from the R package `Matrix`, which is an R wrapper for a C++ implementation of the sparse Cholesky-based solver in Davis and Hager (2009). For such large graphs, alternative algorithms, such as (preconditioned) conjugate gradient methods, could certainly be more efficient in computing Laplacian smoothing solutions; our reported timings are only meant to indicate that the DFS fused lasso is efficiently computable at problem sizes that are large enough that even a simple linear method like Laplacian smoothing becomes nontrivial.

5.2 2d grid graphs

Next we consider a denoising example on a 2d grid graph of dimension 1000×1000, so that the number of nodes is $n = 1000000$ and the number of edges is $m = 1998000$. We generated a synthetic piecewise constant signal $\theta_0 \in \mathbb{R}^{1000 \times 1000}$ over the 2d grid, shown in the top left corner of Figure 2, where a color scale (displayed in the accompanying color legend) is used, with red denoting the smallest possible value and yellow the largest possible value. Data $y \in \mathbb{R}^{1000 \times 1000}$ was generated by adding i.i.d. $N(0, 1)$ noise to the components of θ_0, displayed in the top middle panel of Figure 2. We then computed the 2d fused lasso solution, i.e., the fused lasso solution over the full 2d grid[2] graph, as well as three DFS-based variations: the DFS fused lasso solution using a random DFS ordering (given by running DFS beginning at a random node), labeled as "1 random DFS" in the figure; the average of DFS fused lasso solutions over 5 random DFS orderings, labeled "5 random DFS" in the figure; and the average of DFS fused lasso solutions over 2 "snake" DFS orderings (one given by collecting and joining all horizontal edges and the other all vertical edges) labeled "2 snake DFS" in the figure. The tuning parameter for each method displayed in the figure was chosen to minimize the average MSE over 100 draws of the data y from the specified model. Visually, we can see that the full 2d fused lasso solution is the most accurate, however, the 1 random DFS, 5 random DFS, and 2 snake DFS solutions all still clearly capture the structure inherent in the underlying signal. Of the three DFS variations, the 5 random DFS estimator is visually most accurate; the 1 random DFS estimator is comparably "blotchy", and the 2 snake DFS estimator is comparably "stripey".

The left panel of 3 shows the optimized MSE for each method, i.e., the minimum of the average MSE over 100 draws of the data y, when we consider 20 choices for the tuning parameter. This optimized MSE is plotted as a function of the sample size, which runs from $n = 2500$ (a 50×50 grid) to $n = 1000000$ (a 1000×1000 grid), and in each case the underlying signal is formed by taking an appropriate (sub)resolution of the image in the top left panel of Figure 2. The 2d fused lasso provides the fastest decrease in MSE as n grows, followed by the 5 random DFS estimator, then the 1 random DFS estimator, and the 2 snake DFS estimator. This is not a surprise, since the 2d fused lasso uses the information from the full 2d grid. Indeed, comparing (11) and (19), we recall that the 2d fused lasso enjoys an MSE rate of $t \log^2 n/n$ when θ_0 has 2d total variation t, whereas the

2. In the previous subsection, we remarked that the fused lasso was prohibitive to compute over the road network graphs, which each have in between 1 or 2 million nodes and 1.5 and 3 million edges. Here, we are able to compute the fused lasso over a graph with 1 million nodes and nearly 2 million edges. The difference is the specialized 2d grid structure in the present example, which is leveraged by the proximal stacking technique in Barbero and Sra (2011) that we use to compute the 2d fused lasso solutions

DFS fused lasso has an MSE rate of only $(t/n)^{2/3}$ in this setting. When $t \asymp \sqrt{n}$, which is a natural scaling for the underlying total variation in 2d and also the scaling considered in the experimental setup for the figure, these rates are $(\log^2 n)n^{-1/2}$ for the 2d fused lasso, and $n^{-1/3}$ for the DFS fused lasso. The figure uses a log-log plot, so the MSE curves all appear to have linear trends, and the fitted slopes roughly match these theoretical MSE rates (-0.58 for the 2d fused lasso, and -0.39, -0.40, and -0.36 for the three DFS variations).

The right panel of Figure 3 shows the runtimes for each method (averaged over 100 draws of the data y), as a function of the sample size n. The runtime for each method counts the total time taken to compute solutions across 20 tuning parameter values. The computations and timings were carried out on a standard desktop computer (with a 3.40GHz Intel Core i7-4770 processor). To compute 2d fused lasso solutions, we used the `TVgen` function in the Matlab package `proxTV`, which is a Matlab wrapper for a C++ implementation of the proximal stacking technique described in Barbero and Sra (2014). For the DFS fused lasso, we computed initial DFS orderings using the `dfs` function from the Matlab package `MathBGL`, and then, as before, used the C++ implementation available through `glmgen` to compute the appropriate 1d fused lasso solutions. The figure uses a log-log plot, and hence we can see that all DFS-based estimators are quite a bit more efficient than the 2d fused lasso estimator.

5.3 Tree graphs

We finish with denoising comparisons on tree graphs, for sample sizes varying from $n = 100$ to $n = 5300$. For each sample size n, a random tree is constructed via a sequential process in which each node is assigned a number of children between 2 and 10 (uniformly at random). Given a tree, an underlying signal $\theta_0 \in \mathbb{R}^n$ is constructed to be piecewise constant with total variation $5\sqrt{n}$ (the piecewise constant construction here is made easy because the oriented incidence matrix of a tree is invertible). Data $y \in \mathbb{R}^n$ was generated by adding i.i.d. $N(0, 1)$ noise to θ_0. We compared the fused lasso estimator over the full tree, 1 random DFS and 5 random DFS estimators (using the terminology from the last subsection), and the wavelet smoothing estimator defined in (26). For each estimator, we computed the entire solution path using the path algorithm of Tibshirani and Taylor (2011) implemented in the R package `genlasso`, and selected the step along the path to minimize the average MSE over 50 draws of data y around θ_0, and 10 repetitions in constructing θ_0. (The full solution path can be computed here because each estimator can be cast as a generalized lasso problem, and because the problem sizes considered here are not enormous.)

The left panel of Figure 4 plots this optimized MSE as a function of the sample size n. We see that the fused lasso estimator over the full tree and the 5 random DFS estimator perform more or less equivalently over all sample sizes. The 1 random DFS estimator is slightly worse, and the wavelet smoothing estimator is considerably worse. The right panel shows the the MSE as a function of the effective degrees of freedom of each estimator, for a particular data instance with $n = 5300$. We see that both the tree fused lasso and 1 random DFS estimators achieve their optimum MSEs at solutions of low complexity (degrees of freedom), whereas wavelet smoothing does not come close to achieving this MSE across its entire path of solutions.

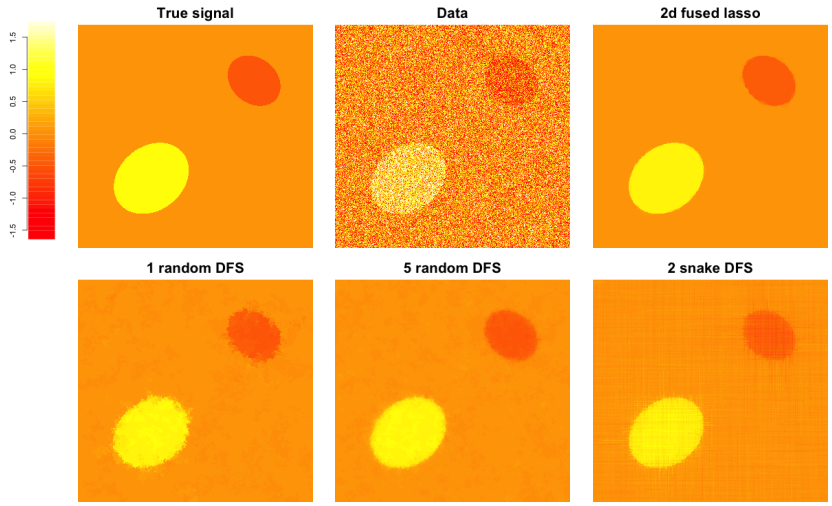


Figure 2: *Underlying signal, data, and solutions from the 2d fused lasso and different variations on the DFS fused lasso fit over a* 1000×1000 *grid.*

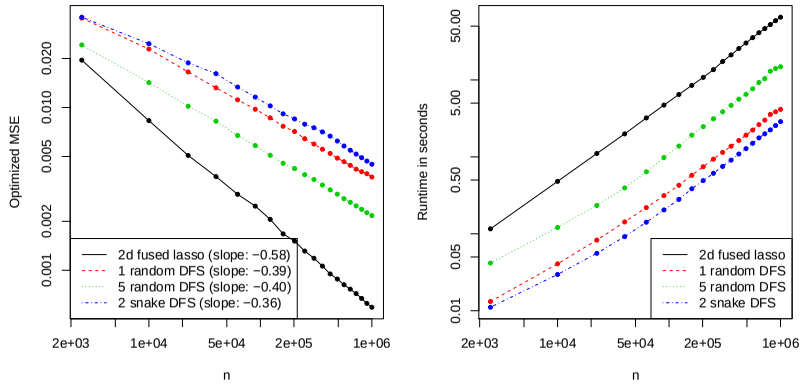


Figure 3: *Optimized MSE and runtime for the 2d fused lasso and DFS fused lasso estimators over a 2d grid, as the grid size n (total number of nodes) varies.*

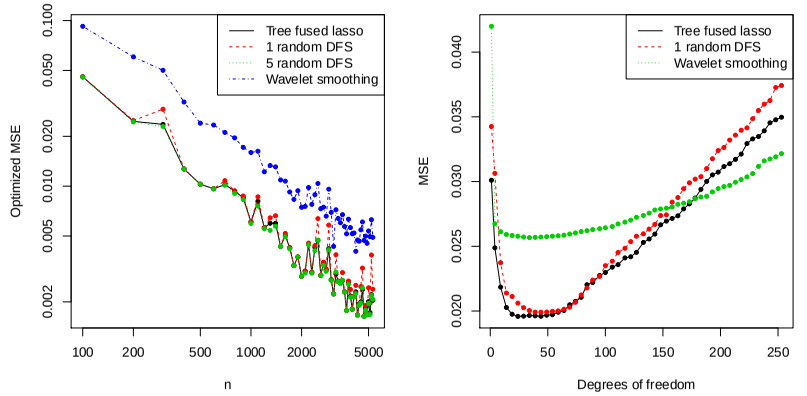


Figure 4: *The left panel shows the optimized MSE as a function of the sample size n for the fused lasso over a tree graph, as well as the 1 random DFS and 5 random DFS estimators, and wavelet smoothing. The right panel shows the MSE as a function of the degrees of freedom of each estimator, for a particular data example with* $n = 5300$.

6. Discussion

Recently, there has been a significant amount on interest on graph-structured denoising. Much of this work has focused on the construction of graph kernels or wavelet bases. We have proposed and studied a simple method, defined by computing the 1d fused lasso over a particular DFS-induced ordering of the nodes of a general graph. This linear-time algorithm comes with strong theoretical guarantees for signals of bounded variation (achieving optimal MSE rates for trees of bounded degree), as well as guarantees for signals with a bounded number of nonzero differences (achieving nearly optimal rates under a condition on the spacings of jumps along the DFS-induced chain). We summarize our theoretical results in Table 1.

Practically, we have seen that the DFS fused lasso can often represent a useful trade-off between computational efficiency and statistical accuracy, versus competing methods that offer better statistical denoising power but are more computationally expensive, especially for large problems. A simple trick like averaging multiple DFS fused lasso fits, over multiple random DFS-induced chains, often improves statistical accuracy at little increased computational cost. Several extensions along these lines, and other lines, are possible. To study any of them in detail is beyond the scope of this paper. We discuss them briefly below, leaving detailed follow-up to future work.

6.1 Beyond simple averaging

Given multiple DFS fused lasso estimators, $\hat{\theta}^{(1)}_{\mathrm{DFS}}, \ldots, \hat{\theta}^{(K)}_{\mathrm{DFS}}$, obtained using multiple DFS-induced chains computed on the same graph G, there are several possibilities for intelligently combining these estimators beyond the simple average, denoted (say) $\bar{\theta}^{(K)}_{\mathrm{DFS}} = (1/K) \sum_{k=1}^{K} \hat{\theta}^{(k)}_{\mathrm{DFS}}$.

	$\mathrm{BV}_G(t)$, $t \asymp 1$	$\mathrm{BD}_G(s)$, $s \asymp 1$
Fused lasso, $\hat{\theta}_G$	$n^{-2/3}$	unknown
Spanning tree wavelets, $\hat{\theta}_W$	unknown	$(\log^2 n \log d_{\max})/n$
DFS fused lasso, $\hat{\theta}_{\mathrm{DFS}}$	$n^{-2/3}$	$(\log n \log\log n)/n^*$
Tree lower bound	$n^{-2/3} d_{\max}^{-4/3}$	$\log n/n$

Table 1: *A summary of the theoretical results derived in this paper. All rates are on the mean squared error (MSE) scale ($\mathbb{E}\|\hat{\theta} - \theta_0\|_n^2$ for an estimator $\hat{\theta}$), and for simplicity, are presented under a constant scaling for t, s, the radii in the $\mathrm{BV}_G(t), \mathrm{BD}_G(s)$ classes, respectively. The superscript "$*$" in the $\mathrm{BD}_G(s)$ rate for the DFS fused lasso is used to emphasize that this rate only holds under the assumption that $W_n \asymp n$. Also, we write $d_{\max}$ to denote the max degree of the graph in question.*

To better preserve edges in the combined estimator, we could run a simple nonlinear filter—for example, a median filter, over $\hat{\theta}^{(1)}_{\mathrm{DFS}}, \ldots, \hat{\theta}^{(K)}_{\mathrm{DFS}}$ (meaning that the combined estimator is defined by taking medians over local neighborhoods of all of the individual estimators). A more sophisticated approach would be to compute the DFS fused lasso estimators sequentially, using the $(k-1)$st estimator to modify the response in some way in the 1d fused lasso problem that defines the kth DFS fused lasso estimator. We are intentionally vague here with the specifics, because such a modification could be implemented in various ways; for example, it could be useful to borrow ideas from the boosting literature, which would have us treat each DFS fused lasso estimator as a weak learner.

6.2 Distributed algorithm

For large graphs, we should be able to both compute a DFS ordering over G, and solve the DFS fused lasso problem in (18), in a distributed fashion. There are many algorithms for distributed DFS, offering a variety of communication and time complexities; see, e.g., Tsin (2002) for a survey. Distributed algorithms for the 1d fused lasso are not as common, though we can appeal to the now well-studied framework for distributed optimization via the alternating direction method of multipliers (ADMM) from Boyd et al. (2011). Different formulations for the auxiliary variables present us with different options for communication costs. We have found that, for a formulation that requires $O(1)$-length messages to be communicated between processors, the algorithm typically converges in a reasonably small number of iterations.

6.3 Theory for piecewise constant signals

The bounded differences class $\mathrm{BD}_G(s)$ in (5) is defined in terms of the cut metric $\|\nabla_G \theta\|_0$ of a parameter θ, which recall, counts the number of nonzero differences occurring in θ over edges in the graph G. The cut metric measures a notion of strong sparsity (compared to the weaker notion measured by the total variation metric) in a signal θ, over edge differences; but, it may not be measuring sparsity on the "right" scale for certain graphs G. Specifically,

the cut metric $\|\nabla_G \theta\|_0$ can actually be quite large for a parameter θ that is piecewise constant over G, with a small number of pieces—these are groups of connected nodes that are assigned the same constant value in θ. Over the 2d grid graph, e.g., one can easily define a parameter θ that has only (say) two constant pieces but on the order of $\sqrt{n}$ nonzero edge differences. Therefore, for such a "simple" configuration of the parameter θ, the cut metric $\|\nabla_G \theta\|_0$ is deceivingly large.

To formally define a metric that measures the number of constant pieces in a parameter θ, with respect to a graph $G = (V, E)$, we introduce a bit of notation. Denote by $Z(\theta) \subseteq E$ the subset of edges over which θ exhibits differences of zero, i.e., $Z(\theta) = \{e \in E : \theta_{e^+} = \theta_{e^-}\}$. Also write $(\nabla_G)_{Z(\theta)}$ for the submatrix of the edge incidence matrix ∇_G with rows indexed by $Z(\theta)$. We consider a metric defined by

$$\rho_G(\theta) = \mathrm{nullity}\big((\nabla_G)_{Z(\theta)}\big),$$

where $\mathrm{nullity}(\cdot)$ denotes the dimension of the null space of its argument. An equivalent definition is

$$\rho_G(\theta) = \text{the number of connected components in } (V, E \setminus Z(\theta)).$$

We may now define the *piecewise constant class*, with respect to G, and a parameter $s > 0$,

$$\mathrm{PC}_G(s) = \{\theta \in \mathbb{R}^n : \rho_G(\theta) \leq s\}.$$

It is not hard to to see that $\mathrm{BV}_G(s) \subseteq \mathrm{PC}_G(s)$ (assuming only that G is connected), but for certain graph topologies, the latter class $\mathrm{PC}_G(s)$ will be much larger. Indeed, to repeat what we conveyed above, for the 2d grid one can naturally define a parameter θ such that $\theta \in \mathrm{BD}_G(\sqrt{n})$ and $\theta \in \mathrm{PC}_G(2)$.

We conjecture that the fused lasso estimator over G can achieve a fast MSE rate when the mean θ_0 in (1) exhibits a small number of constant pieces, i.e., $\theta_0 \in \mathrm{PC}_G(s)$, provided that these pieces are of roughly equal size.

6.4 Weighted graphs

The key result in Lemma 1 can be extended to the setting of a weighted graph $G = (V, E, w)$, with $w_e \geq 0$ denoting the edge weight associated to an edge $e \in E$.

Lemma 9 *Let $G = (V, E, w)$ be a connected weighted graph, where recall we write $V = \{1, \ldots, n\}$, and we assume all edge weights are nonnegative. Consider running DFS on G, and denote by $\tau : \{1, \ldots, n\} \to \{1, \ldots, n\}$ the induced permutation, so that if $v_1, \ldots, v_n$ are the nodes in the order that they are traversed by DFS, then*

$$\tau(i) = v_i, \quad \text{for all } i = 1, \ldots, n.$$

Denote $w_{\min} = \min_{e \in E} w_e$, the minimum edge weight present in the graph, and define

$$\tilde{w}_{\tau(i),\tau(i+1)} = \begin{cases} w_e & \text{if } e = \{\tau(i), \tau(i+1)\} \in E, \\ w_{\min} & \text{otherwise,} \end{cases} \quad \text{for all } i = 1, \ldots, n-1. \tag{30}$$

It holds that

$$\sum_{i=1}^{n-1} \tilde{w}_{\tau(i),\tau(i+1)} \big|\theta_{\tau(i+1)} - \theta_{\tau(i)}\big| \leq 2 \sum_{e \in E} w_e |\theta_{e^+} - \theta_{e^-}|, \quad \textit{for all } \theta \in \mathbb{R}^n, \tag{31}$$

as well as

$$\sum_{i=1}^{n-1} \tilde{w}_{\tau(i),\tau(i+1)} 1\big\{\theta_{\tau(i+1)} \neq \theta_{\tau(i)}\big\} \leq 2 \sum_{e \in E} w_e 1\{\theta_{e^+} \neq \theta_{e^-}\}, \quad \textit{for all } \theta \in \mathbb{R}^n. \tag{32}$$

Remark 10 *As with Lemma 1, Lemma 9 has also essentially appeared in the literature, in a form for binary signals and the roughness metric being the quadratic form in the graph Laplacian, see Cesa-Bianchi et al. (2013). The main idea behind this result and Lemma 9 is the same, and the latter follows from the former with only minor modifications.*

For simplicity, when edges in the DFS chain do not appear in the original graph, we have defined their corresponding weights to be the minimum of the weights in the original graph. Instead, we could have taken the approach in Cesa-Bianchi et al. (2013), and defined the weight for such an edge in the DFS chain to be the minimum of weights from original edges in its backtracking path.

The bounds in (31), (32) are the analogies of (15), (16) but for a weighted graph G; indeed we see that we can still embed a DFS chain into G, but this chain itself comes with edge weights, as in (30). These new edge weights in the chain do not cause any computational issues; the 1d fused lasso problem with arbitrary penalty weights can still be solved in $O(n)$ time using the taut string algorithm in Barbero and Sra (2014). Thus, in principle, all of the results in this paper should carry over in some form to weighted graphs.

6.5 Robustness to signal perturbation

Here we briefly explore some robustness properties of the DFS fused lasso estimator, under perbutations of the mean parameter θ_0 in (1) with respect to a small number of nodes. This is based on the approach taken by Cesa-Bianchi et al. (2013) to study robustness in their online graph-structured classification setting. Let $G = (V, E, w)$ a weighted graph, and write $\|\nabla_G \theta\|_1$ for the weighted total variation of a signal θ, i.e.,

$$\|\nabla_G \theta\|_1 = \sum_{e \in E} w_e |\theta_{e^+} - \theta_{e^-}|.$$

Denote by θ^δ a perturbed version of θ, where any number of entries are given by adding an amount $\delta > 0$ to the corresponding entries of θ. (More general forms of perturbations can also be investigated but slightly complicate the discussion that follows, so for simplicity, we will limit ourselves to considering this simple model for perturbations.) Denote by

$$I(\theta, \theta^\delta) = \Big\{ i \in \{1, \ldots, n\} \,:\, \theta_i \neq \theta_i^\delta \Big\}$$

the subset of nodes at which the components of θ and θ^δ differ. Observe

$$\begin{aligned}
\big|\|\nabla_G \theta\|_1 - \|\nabla_G \theta^\delta\|_1\big| &= \bigg| \sum_{\{e^+, e^-\} \cap I(\theta,\theta^\delta) \neq \emptyset} w_e \left(|\theta_{e^+} - \theta_{e^-}| - |\theta^\delta_{e^+} - \theta^\delta_{e^-}|\right) \bigg| \\
&= \bigg| \sum_{|\{e^+, e^-\} \cap I(\theta,\theta^\delta)| = 1} w_e \left(|\theta_{e^+} - \theta_{e^-}| - |\theta^\delta_{e^+} - \theta^\delta_{e^-}|\right) \bigg| \\
&\leq \sum_{|\{e^+, e^-\} \cap I(\theta,\theta^\delta)| = 1} w_e \left(|\theta_{e^+} - \theta^\delta_{e^+}| + |\theta_{e^-} - \theta^\delta_{e^-}|\right) \\
&= \delta \, \mathrm{cut}_G(I(\theta, \theta^\delta)),
\end{aligned}$$

where, for a subset $S \subset V$, we use $\mathrm{cut}_G(S)$ to denote the cost of the cut in between S and S^c in the graph G, i.e., the sum of edge weights among edges with one endpoint in S and the other in S^c.

Thus we have shown that the (weighted) total variation metric associated with G has modulus of continuity $\delta \, \mathrm{cut}_G(I(\theta, \theta^\delta))$ with respect to perturbations θ^δ of θ. This scales linearly in δ, but how does it scale in $|I(\theta, \theta^\delta)|$ (the number of discrepancies)? The answer to this depends on the geometry of G, in particular, it depends on whether perturbations occur at nodes having high (weighted) degree. Even when $|I(\theta, \theta^\delta)|$ is small, $\mathrm{cut}_G(I(\theta, \theta^\delta))$ can be very large—an example is given below with G being a star graph. Further, the modulus of continuity $\delta \, \mathrm{cut}_G(I(\theta, \theta^\delta))$ is tight, i.e., it can always be achieved[3], and thus for graphs G with nodes of high (weighted) degrees, in a worst-case sense, small perturbations of θ can lead to big differences in $\|\nabla_G \theta\|_1$.

For the DFS-induced chain graph derived from G, the worst-case is not nearly as bad. Letting C denote this chain and P the corresponding permutation matrix, by the same argument as above,

$$\Big|\|\nabla_C P\theta\|_1 - \|\nabla_C P\theta^\delta\|_1\Big| \leq \delta \, \mathrm{cut}_C(I(\theta, \theta^\delta)).$$

But the key difference now is that $\mathrm{cut}_C(I(\theta, \theta^\delta))$ sums the edge weights of at most $2\,|I(\theta, \theta^\delta)|$ edges in C, due to the special structure of the chain. Denoting by

$$v_G(k) = \max_{F \subset E, |F| = k} \sum_{e \in F} w_e$$

the sum of the k largest edge weights in G, and similarly for C, we can proceed to upper bound the right-hand side above,

$$\Big|\|\nabla_C P\theta\|_1 - \|\nabla_C P\theta^\delta\|_1\Big| \leq \delta \, v_C(2\,|I(\theta, \theta^\delta)|) \leq \delta \, v_C(2\,|I(\theta, \theta^\delta)|),$$

the second inequality following from the definition of the weights in the chain graph, as in (30). When all weights in the original graph G are unity, e.g., this says that perturbing k

3. By this we mean that, for any graph G, any subset $S \subset V$ of nodes at which perturbations are to occur, there exist θ, θ^δ such that $I(\theta, \theta^\delta) = S$ and $\big|\|\nabla_G \theta\|_1 - \|\nabla_G \theta^\delta\|_1\big| = \delta \, \mathrm{cut}_G(I(\theta, \theta^\delta))$.

of the entries of any input signal by an amount δ results in a total variation difference over the DFS-induced chain of at most $2\delta k$.

Consider the following instructive example, borrowed from Cesa-Bianchi et al. (2013). Let G be a star graph, having one center node that is connected to each of the remaining $n-1$ nodes, and no other edges in the graph. Assume that all weights of G are unity, and define θ to take a value 0 on the center node, and 1 on all others. Then by flipping the value of the center node from 0 to 1, the total variation over G changes from $n-1$ to 0. However, for any DFS-induced chain, we see from the above analysis that the total variation can change by at most 2, making it more robust to such a perturbation. We can rephrase this robustness property in an interesting way. Suppose that the mean θ_0 in (1) is defined over the star graph G to take a value 0 on the center node, and 1 on all others. Then θ_0 is "close" to a signal of bounded variation, since changing only its value at the center node makes it have zero total variation, and yet in its current configuration it has total variation $n-1$. Hence, we would not expect the fused lasso estimator over G to be consistent, when fit to data drawn around θ_0. But the total variation of θ_0 as measured over the DFS-induced chain C is at most 2 (as the total variation over C is zero once we perturb the value of center node from 0 to 1). With respect to C then, the parameter θ_0 is certainly of bounded variation and using the DFS fused lasso estimator, we will achieve an MSE rate of $n^{-2/3}$. The DFS fused lasso thus exhibits a robustness property, in that it allows us to accurately estimate signals that are "close to" the set of bounded variation signals over G.

6.6 Potts and energy minimization

Replacing the total variation metric by the cut metric in the fused lasso problem (2) gives us

$$\tilde{\theta}_G = \operatorname*{argmin}_{\theta \in \mathbb{R}^n} \frac{1}{2}\|y-\theta\|_2^2 + \lambda \|\nabla_G \theta\|_0, \tag{33}$$

often called the *Potts minimization* problem. Because the 1d Potts minimization problem

$$\tilde{\theta}_{1\mathrm{d}} = \operatorname*{argmin}_{\theta \in \mathbb{R}^n} \frac{1}{2}\|y-\theta\|_2^2 + \lambda \|\nabla_{1\mathrm{d}} \theta\|_0 \tag{34}$$

can be solved efficiently, e.g., in worst-case $O(n^2)$ time with dynamic programming Bellman (1961); Johnson (2013), the same strategy that we have proposed in this paper can be applied to reduce the graph Potts problem (33) to a 1d Potts problem (34), via a DFS ordering of the nodes. This may be especially interesting as the original Potts problem (33) is non–convex and generally intractable (i.e., intractable to solve to global optimality) for an arbitrary graph structure, so a reduction to a worst-case quadratic-time denoiser is perhaps very valuable.

When the optimization domain in (33) is a discrete set, the problem is often called an *energy minimization* problem, as in Boykov et al. (2001). It has not escaped our notice that our technique of denoising over DFS-induced chains could be useful for this setting, as well.

Acknowledgments

The authors thank the Associate Editor and Referees for their helpful comments, especially for pointing out the connections to the previous papers of Herbster et al. (2009); Cesa-Bianchi et al. (2013), and the signal perturbation angle covered in Section 6.5. The last author thanks Veeranjaneyulu Sadhanala and Yu-Xiang Wang for early stimulating discussions. Ryan J. Tibshirani was supported by NSF grant DMS-1554123.

Appendix A. Proofs

A.1 Derivation of (12) from Theorem 3 in Wang et al. (2016)

We first establish a result on the exact form for the inverse of (an augmented version of) the edge incidence matrix of a generic tree $T = (V, E_T)$, where, recall $V = \{1, \dots, n\}$. Without a loss of generality, we may assume that the root of T is at node 1. For $m \leq n$, we define a path in T, of length m, to be a sequence $p_1, \dots, p_m$ such that $\{p_r, p_{r+1}\} \in E_T$ for each $r = 1, \dots, m-1$. We allow for the possibility that $m = 1$, in which case the path has just one node. For any $j, k, \ell = 1, \dots, n$, we say that j is on the path from k to ℓ if there exists a path $p_1, \dots, p_m$ such that $p_1 = k$, $p_m = \ell$ and $p_r = j$ for some $r = 1, \dots, m$. For each node $i = 2, \dots, n$ (each node other than the root), we define its parent $p(i)$ to be the node connected to i which is on the path from the root to i.

We can also assume without a loss of generality that for each $i = 2, \dots, n$, the $(i-1)$st row of ∇_T corresponds to the edge $\{p(i), i\}$, and thus we can write

$$(\nabla_T)_{i-1,j} = \begin{cases} -1 & \text{if } j = p(i), \\ 1 & \text{if } j = i, \\ 0 & \text{if } j \in \{1, \dots, n\} \setminus \{i, p(i)\}. \end{cases}$$

for each $j = 1, \dots, n$. The next lemma describes the inverse of ∇_T, in the appropriate sense.

Lemma 11 *Let $e_1 = (1, 0, \dots, 0) \in \mathbb{R}^n$, and define the matrix $A_T \in \mathbb{R}^{n \times n}$ by*

$$(A_T)_{i,j} = \begin{cases} 1 & \textit{if } j \textit{ is on the path from the root to } i, \\ 0 & \textit{otherwise}, \end{cases} \tag{35}$$

for each $i, j = 1, \dots, n$. Then

$$A_T = \begin{pmatrix} e_1^\top \\ \nabla_T \end{pmatrix}^{-1}.$$

Proof We will prove that the product

$$B = \begin{pmatrix} e_1^\top \\ \nabla_T \end{pmatrix} A_T$$

is the identity. As the root of T corresponds to node 1, we have that by definition of A_T that its first column is

$$(A_T)_{\cdot,1} = (1, \dots, 1),$$

which implies that the first column of B is

$$B_{\cdot,1} = e_1.$$

Moreover, by definition of A_T, its first row is

$$(A_T)_{1,\cdot} = e_1^\top,$$

which implies that the first row of B is

$$B_{1,\cdot} = e_1^\top.$$

Let us now assume that i, j are each not the root. We proceed to consider three cases.

Case 1. Let $j \neq i$, and j be on the path from the root to i. Then j is also on the path from the root to $p(i)$. This implies that

$$B_{ij} = \begin{pmatrix} e_1^\top \\ \nabla_T \end{pmatrix}_{i,\cdot} (A_T)_{\cdot,j} = (\nabla_T)_{i-1,\cdot}(A_T)_{\cdot,j} = 1 - 1 = 0.$$

Case 2. Let $j \neq i$, and j not be on the path from the root to i. Then j is not on the path from the root to $p(i)$, which implies that

$$B_{ij} = \begin{pmatrix} e_1^\top \\ \nabla_T \end{pmatrix}_{i,\cdot} (A_T)_{\cdot,j} = (\nabla_T)_{i-1,\cdot}(A_T)_{\cdot,j} = 0 - 0 = 0.$$

Case 3. Let $j = i$. Then j is on the path from the root to i, and j is not on the path from the root to $p(i)$. Hence,

$$B_{ij} = \begin{pmatrix} e_1^\top \\ \nabla_T \end{pmatrix}_{i,\cdot} (A_T)_{\cdot,j} = (\nabla_T)_{i-1,\cdot}(A_T)_{\cdot,j} = -1 \cdot 0 + 1 \cdot 1 = 1.$$

Assembling these three cases, we have shown that $B = I$, completing the proof. ■

We now establish (12).

Proof [Proof of (12)] The proof of Theorem 3 in Wang et al. (2016) proceeds as in standard basic inequality arguments for the lasso, and arrives at the step

$$\|\Pi^\perp(\hat{\theta}_G - \theta_0)\|_2^2 \leq 2\epsilon^\top \Pi^\perp(\hat{\theta}_G - \theta_0) + 2\lambda\|\nabla_G\theta_0\|_1 - 2\lambda\|\nabla_G\hat{\theta}_G\|_1,$$

where $\Pi^\perp$ is the projection matrix onto the space $\mathrm{span}\{\mathbb{1}\}^\perp$, i.e., the linear space of all vectors orthogonal to the vector $\mathbb{1} = (1, \ldots, 1) \in \mathbb{R}^n$ of all 1s. The proof in Wang et al. (2016) uses the identity $\Pi^\perp = \nabla_G^\dagger \nabla_G$, where $\nabla_G^\dagger$ denotes the pseudoinverse of ∇_G. However, notice that we may also write $\Pi^\perp = \nabla_T^\dagger \nabla_T$ for any spanning tree T of G. Then, exactly the same arguments as in Wang et al. (2016) produce the MSE bound

$$\|\hat{\theta}_G - \theta_0\|_n^2 = O_{\mathbb{P}}\bigg(\frac{M(\nabla_T)\sqrt{\log n}}{n}\|\nabla_G\theta_0\|_1\bigg),$$

where $M(\nabla_T)$ is the maximum ℓ_2 norm among the columns of $\nabla_T^\dagger$. We show below, using Lemma 11, that $M(\nabla_T) \leq \sqrt{n}$, and this gives the desired MSE rate.

For any $b \in \mathbb{R}^{n-1}$, we may characterize $\nabla_T^\dagger b$ as the unique solution $x \in \mathbb{R}^n$ to the linear system

$$\nabla_T x = b,$$

such that $\mathbb{1}^\top x = 0$, i.e., the unique solution to the linear system

$$\begin{pmatrix} e_1^\top \\ \nabla_T \end{pmatrix} x = \begin{pmatrix} a \\ b \end{pmatrix},$$

for a value of $a \in \mathbb{R}$ such that $\mathbb{1}^\top x = 0$. By Lemma 11, we may write

$$x = A_T \begin{pmatrix} a \\ b \end{pmatrix},$$

so that the constraint $0 = \mathbb{1}^\top x = na + \mathbb{1}^\top (A_T)_{\cdot,2:n} b$ gives $a = -(\mathbb{1}/n)^\top (A_T)_{\cdot,2:n} b$, and

$$x = (I - \mathbb{1}\mathbb{1}^\top/n)(A_T)_{\cdot,2:n} b.$$

Evaluating this across $b = e_1, \ldots, e_n$, we find that the maximum ℓ_2 norm of columns of $\nabla_T^\dagger$ is bounded by the maximum ℓ_2 norm of columns of $(A_T)_{\cdot,2:n}$, which, from the definition in (35), is at most $\sqrt{n}$. ■

A.2 Proof of Theorem 5

We first present two preliminary lemmas.

Lemma 12 *Let $S_1, \ldots, S_m$ be a partition of the nodes of G such that the total number of edges with ends in distinct elements of the partition is at most s. Let $k \leq \min_{i=1,\ldots,m} |S_i|$. Then*

$$\inf_{\hat{\theta}} \sup_{\theta_0 \in \mathrm{BV}_G(t)} \mathbb{E}\|\hat{\theta} - \theta_0\|_2^2 \geq \frac{kmt^2}{4\sigma^2 s^2} \exp\bigg(-\frac{kt^2}{\sigma^2 s^2}\bigg).$$

Proof For each $\eta \in \{-1, 1\}^m$, define

$$\theta_\eta = \frac{\delta}{2} \sum_{i=1}^m \eta_i \frac{1_{S_i}}{\sqrt{|S_i|}},$$

where $\delta > 0$ will be specified shortly. Also define the class $\mathcal{P} = \{N(\theta_\eta, \sigma^2 I) : \eta \in \{-1, 1\}^m\}$. Note that $\|\nabla_G \theta_\eta\|_1 \leq \delta s/\sqrt{k}$, so to embed $\mathcal{P}$ into the class $\{N(\theta, \sigma^2 I) : \theta \in \mathrm{BV}_G(t)\}$, we set $\delta = t\sqrt{k}/s$.

Let $\eta, \eta' \in \{-1, 1\}^m$ differ in only one coordinate. Then the KL divergence between the corresponding induced measures in $\mathcal{P}$ is $\|\theta_\eta - \theta_{\eta'}\|_2^2/\sigma^2 \leq \delta^2/\sigma^2$. Hence by Assouad's Lemma (Yu, 1997), and a well-known lower bound on the affinity between probability measures in terms of KL divergence,

$$\inf_{\hat{\theta}} \sup_{\theta_0 \in \mathrm{BV}_G(t)} \mathbb{E}\|\hat{\theta} - \theta_0\|_2^2 \geq \frac{\delta^2 m}{4\sigma^2} \exp\bigg(-\frac{\delta^2}{\sigma^2}\bigg).$$

The result follows by plugging in the specified value for δ. ■

Lemma 13 *Let G be a tree with maximum degree $d_{\max}$, and $k \in \{1, \dots, n\}$ be arbitrary. Then there exists a partition as in Lemma 12, $s = m - 1$, and*

$$k \leq \min_{i=1,\dots,m} |S_i| \leq k(d_{\max} + 1).$$

Proof Our proof proceeds inductively. We begin by constructing S'_1, the smallest subtree among all those having size at least k, and generated by a cut of size 1 (i.e., separated from the graph by the removal of 1 edge). Note that $|S'_1| \leq kd_{\max}$, because if not then S'_1 has at least k internal nodes, and we can remove its root to produce another subtree whose size is smaller but still at least k.

For the inductive step, assume $S'_1, \dots, S'_\ell$ have been constructed. We consider two cases. (For a subgraph G' of G, we denote by $G - G'$ the complement subgraph, given by removing all nodes in G', and all edges incident to a node in G'.)

Case 1. If $|G - \cup_{i=1}^{\ell} S'_i| > k$, then we construct $S'_{\ell+1}$, the smallest subtree of $G - \cup_{i=1}^{l} S'_i$ among all those having size at least k, and generated by a cut of size 1. As before, we obtain that $|S'_{\ell+1}| \leq kd_{\max}$.

Case 2. If $|G - \cup_{i=1}^{\ell} S'_i| \leq k$, then the process is stopped. We define $S_i = S'_i$, $i = 1, \dots, \ell - 1$, as well as $S_\ell = S'_\ell \cup (G - \cup_{i=1}^{\ell} S'_i)$. With $m = \ell$, the result follows. ■

We now demonstrate a more precise characterization of the lower bound in Theorem 5, from which the result in the theorem can be derived.

Theorem 14 *Let G be a tree with maximum degree $d_{\max}$. Then*

$$\inf_{\hat{\theta}} \sup_{\theta_0 \in \mathrm{BV}_G(t)} \mathbb{E}\|\hat{\theta} - \theta_0\|_2^2 \geq \frac{t^2}{4e\sigma^2 n} \left(\left(\frac{\sigma n}{2t(d_{\max} + 1)} \right)^{2/3} - 1 \right)^2.$$

Proof Set $s = m - 1$ and

$$k = \left\lfloor \left(\frac{\sigma n}{2t(d_{\max} + 1)} \right)^{2/3} \right\rfloor.$$

By Lemmas 12 and 13,

$$\begin{aligned}
\inf_{\hat{\theta}} \sup_{\theta_0 \in \mathrm{BV}_G(t)} \mathbb{E}\|\hat{\theta} - \theta_0\|_2^2 &\geq \frac{kmt^2}{4\sigma^2 s^2} \exp\left(- \frac{kt^2}{\sigma^2 s^2} \right) \\
&\geq \frac{kt^2}{4\sigma^2 m} \exp\left(- \frac{kt^2}{\sigma^2 (m-1)^2} \right) \\
&\geq \frac{kt^2}{4\sigma^2 m} \exp\left(- \frac{t^2 k^3 (d_{\max} + 1)^2}{\sigma^2 n^2} \frac{m^2}{(m-1)^2} \right) \\
&\geq \frac{kt^2}{4\sigma^2 n} \exp\left(- \frac{4t^2 k^3 (d_{\max} + 1)^2}{\sigma^2 n^2} \right) \\
&\geq \frac{k^2 t^2}{4\sigma^2 n} \exp(-1).
\end{aligned}$$

In the above, the third line uses $n/m \leq kd_{\max}$ as given by Lemma 13, the fourth line simply uses $m \leq n$ and $m^2/(m-1)^2 \leq 4$ (as $m \geq 2$), and the last line uses the definition of k. Thus, because

$$k \geq \left(\frac{\sigma n}{2t(d_{\max} + 1)} \right)^{2/3} - 1,$$

we have established the desired result. ■

A.3 Proof of Theorem 8

First we establish that, as G is a tree, the number of nodes of degree at most 2 is at least $n/2$. Denote by d_i be the degree of the node i, for each $i = 1, \dots, n$. Then

$$2(n-1) = \sum_{i=1}^{n} d_i = \sum_{i:\, d_i \leq 2} d_i + \sum_{i:\, d_i \geq 3} d_i \geq |\{i : d_i \leq 2\}| + 3|\{i : d_i \geq 3\}| = 3n - 2|\{i : d_i \leq 2\}|.$$

Hence, rearranging, we find that $|\{i : d_i \leq 2\}| \geq n/2 + 1$.

Let $\mathcal{I} = \{i : d_i \leq 2\}$ so that $|\mathcal{I}| \geq \lceil n/2 \rceil$ and stipulate that $|\mathcal{I}|$ is even without loss of generality. Let k be the largest even number such that $k \leq s/2$. Define

$$\mathcal{B} = \{z \in \mathbb{R}^n : z_{\mathcal{I}} \in \{-1, 0, +1\}^{|\mathcal{I}|},\ z_{\mathcal{I}^c} = 0,\ \|z\|_0 = k\}.$$

Note that by construction $\mathcal{B} \subseteq \mathrm{BD}_G(s)$.

Assume $s \leq n/6$. Then this implies $k/2 \leq n/6 \leq |\mathcal{I}|/3$. By Lemma 4 in Raskutti et al. (2011), there exists $\tilde{\mathcal{B}} \subseteq \mathcal{B}$ such that

$$\log |\tilde{\mathcal{B}}| \geq \frac{k}{2} \log \left(\frac{|\mathcal{I}| - k}{k/2} \right),$$

and $\|z - z'\|_2^2 \geq k/2$ for all $z, z' \in \tilde{\mathcal{B}}$. Defining $\mathcal{B}_0 = 2\delta\tilde{\mathcal{B}}$, for $\delta > 0$ to be specified shortly, we now have $\|z - z'\|_2^2 \geq 2\delta^2 k$ for all $z, z' \in \mathcal{B}_0$.

For $\theta \in \mathcal{B}_0$, let us consider comparing the measure $P_\theta = N(\theta, \sigma^2 I)$ against $P_0 = N(0, \sigma^2 I)$: the KL divergence between these two satisfies $K(P_\theta || P_0) = \|\theta\|_2^2 / \sigma^2 = 2\delta^2 k / \sigma^2$. Let $\delta = \sqrt{\alpha\sigma^2/(2k) \log |\mathcal{B}_0|}$, for a parameter $\alpha < 1/8$ that we will specify later. We have

$$\frac{1}{|\mathcal{B}_0|} \sum_{\theta \in \mathcal{B}_0} K(P_\theta || P_0) \leq \alpha \log |\mathcal{B}_0|.$$

Hence by Theorem 2.5 in Tsybakov (2009),

$$\inf_{\hat{\theta}} \sup_{\theta_0 \in \mathrm{BD}_G(s)} \mathbb{P}(\|\hat{\theta} - \theta_0\|_2^2 \geq \delta^2 k) \geq \frac{\sqrt{|\mathcal{B}_0|}}{1 + \sqrt{|\mathcal{B}_0|}} \left(1 - 2\alpha - \sqrt{\frac{2\alpha}{\log |\mathcal{B}_0|}} \right). \tag{36}$$

It holds that

$$\delta^2 k = \frac{\alpha\sigma^2}{2} \log |\mathcal{B}_0| \geq \frac{\alpha\sigma^2 k}{4} \log \frac{|\mathcal{I}| - k}{k/2} \geq C\sigma^2 s \log \left(\frac{n}{s} \right),$$

for some constant $C > 0$ depending on α alone. Moreover, the right-hand side in (36) can be lower bounded by (say) 1/4 by taking α to be small enough and assuming n/s is large enough. Thus we have established

$$\inf_{\hat{\theta}} \sup_{\theta_0 \in \mathrm{BD}_G(s)} \mathbb{P}\left(\|\hat{\theta} - \theta_0\|_2^2 \geq C\sigma^2 s \log\left(\frac{n}{s}\right) \right) \geq \frac{1}{4},$$

and the result follows by Markov's inequality.

References

Ittai Abraham and Ofer Neiman. Using petal-decompositions to build a low stretch spanning tree. *ACM Symposium on Theory of Computing*, 44:395–406, 2012.

Alvaro Barbero and Suvrit Sra. Fast Newton-type methods for total variation regularization. *International Conference on Machine Learning*, 28:313–320, 2011.

Álvaro Barbero and Suvrit Sra. Modular proximal optimization for multidimensional total-variation regularization. *arXiv preprint arXiv:1411.0589*, 2014.

Mikhail Belkin and Partha Niyogi. Using manifold structure for partially labelled classification. *Advances in Neural Information Processing Systems*, 15, 2002.

Richard Bellman. On the approximation of curves by line segments using dynamic programming. *Communications of the ACM*, 4(6):284, 1961.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11): 1–18, 2001.

Nicolo Cesa-Bianchi, Claudio Gentile, Fabio Vitale, and Giovanni Zappella. Random spanning trees and the prediction of weighted graphs. *Journal of Machine Learning Research*, 14(1):1251–1284, 2013.

Antonin Chambolle and Jérôme Darbon. On total variation minimization and surface evolution using parametric maximum flows. *International Journal of Computer Vision*, 84(3):288–307, 2009.

Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40: 120–145, 2011.

Ronald Coifman and Mauro Maggioni. Diffusion wavelets. *Applied and Computational Harmonic Analysis*, 21(2):53–94, 2006.

Laurent Condat. A direct algorithm for 1d total variation denoising. *HAL preprint hal-00675043*, 2012.

Thomas Cormen, Clifford Stein, Ronald Rivest, and Charles Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.

Mark Crovella and Eric Kolaczyk. Graph wavelets for spatial traffic analysis. *Annual Joint Conference of the IEEE Computer and Communications IEEE Societies*, 3:1848–1857, 2003.

Arnak Dalalyan, Mohamed Hebiri, and Johannes Lederer. On the prediction performance of the lasso. *To appear, Bernoulli*, 2014.

P. Laurie Davies and Arne Kovac. Local extremes, runs, strings and multiresolution. *Annals of Statistics*, 29(1):1–65, 2001.

Timothy Davis and William Hager. Dynamic supernodes in sparse Cholesky update/downdate and triangular solves. *ACM Transactions on Mathematical Software*, 35 (4):1–23, 2009.

David L Donoho and Iain M Johnstone. Minimax estimation via wavelet shrinkage. *Annals of Statistics*, 26(8):879–921, 1998.

Michael Elkin, Yuval Emek, Daniel Spielman, and Shang-Hua Teng. Lower-stretch spanning trees. *SIAM Journal on Computing*, 38(2):608–628, 2008.

Matan Gavish, Boaz Nadler, and Ronald Coifman. Multiscale wavelets on trees, graphs and high dimensional data: Theory and applications to semi supervised learning. *International Conference on Machine Learning*, 27, 2010.

Chris Godsil and Gordon Royle. *Algebraic Graph Theory*. Springer, 2001.

David Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.

Zaid Harchaoui and Celine Levy-Leduc. Multiple change-point estimation with a total variation penalty. *Journal of the American Statistical Association*, 105(492):1480–1493, 2010.

Mark Herbster, Guy Lever, and Massimiliano Pontil. Online prediction on large diameter graphs. In *Advances in Neural Information Processing Systems*, pages 649–656, 2009.

Holger Hoefling. A path algorithm for the fused lasso signal approximator. *Journal of Computational and Graphical Statistics*, 19(4):984–1006, 2010.

Jan-Christian Hutter and Philippe Rigollet. Optimal rates for total variation denoising. *Annual Conference on Learning Theory*, 29:1115–1146, 2016.

Nicholas Johnson. A dynamic programming algorithm for the fused lasso and l_0-segmentation. *Journal of Computational and Graphical Statistics*, 22(2):246–260, 2013.

Iain Johnstone. Gaussian estimation: sequence and wavelet models. *Unpublished manuscript*, 2011.

Vladimir Kolmogorov, Thomas Pock, and Michal Rolinek. Total variation on a tree. *SIAM Journal of Imaging Sciences*, 9(2):605–636, 2016.

Arne Kovac and Andrew Smith. Nonparametric regression on a graph. *Journal of Computational and Graphical Statistics*, 20(2):432–447, 2011.

Loic Landrieu and Guillaume Obozinski. Cut pursuit: fast algorithms to learn piecewise constant functions on general weighted graphs. *HAL preprint hal-01306779*, 2015.

Jure Leskovec, Kevin J Lang, Anirban Dasgupta, and Michael W Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.

Kevin Lin, James Sharpnack, Alessandro Rinaldo, and Ryan J Tibshirani. Approximate recovery in changepoint problems, from ℓ_2 estimation error rates. *arXiv preprint arXiv:1606.06746*, 2016.

Enno Mammen and Sara van de Geer. Locally apadtive regression splines. *Annals of Statistics*, 25(1):387–413, 1997.

Junyang Qian and Jinzhu Jia. On pattern recovery of the fused lasso. *arXiv preprint arXiv:1211.5194*, 2012.

Garvesh Raskutti, Martin Wainwright, and Bin Yu. Minimax rates of estimation for high-dimensional linear regression over ℓ_q-balls. *IEEE Transactions on Information Theory*, 57(10):6976–6994, 2011.

Alessandro Rinaldo. Properties and refinements of the fused lasso. *The Annals of Statistics*, 37(5):2922–2952, 2009.

Cristian R Rojas and Bo Wahlberg. On change point detection using the fused lasso method. *arXiv preprint arXiv:1401.5408*, 2014.

Leonid Rudin, Stanley Osher, and Emad Faterni. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.

Veeranjaneyulu Sadhanala, Yu-Xiang Wang, and Ryan J. Tibshirani. Total variation classes beyond 1d: Minimax rates, and the limitations of linear smoothers. *To appear, Neural Information Processing Systems*, 2016.

James Sharpnack, Akshay Krishnamurthy, and Aarti Singh. Detecting activations over graphs using spanning tree wavelet bases. *International Conference on Artificial Intelligence and Statistics*, 16:536–544, 2013.

David Shuman, Sunil Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30 (3):83–98, 2013.

Aarti Singh, Robert Nowak, and Robert Calderbank. Detecting weak but hierarchically-structured patterns in networks. *International Conference on Artificial Intelligence and Statistics*, 13:749–756, 2010.

Alexander Smola and Risi Kondor. Kernels and regularization on graphs. *Annual Conference on Learning Theory*, 16, 2003.

Wesley Tansey and James Scott. A fast and flexible algorithm for the graph-fused lasso. *arXiv preprint arXiv:1505.06475*, 2015.

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B*, 67 (1):91–108, 2005.

Ryan J. Tibshirani. Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics*, 42(1):285–323, 2014.

Ryan J. Tibshirani and Jonathan Taylor. The solution path of the generalized lasso. *Annals of Statistics*, 39(3):1335–1371, 2011.

Y. H. Tsin. Some remarks on distributed depth-first search. *Information Processing Letters*, 82:173–178, 2002.

Alexander Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009.

Yu-Xiang Wang, James Sharpnack, Alex Smola, and Ryan J Tibshirani. Trend filtering on graphs. *Journal of Machine Learning Research*, 17(105):1–41, 2016.

Bin Yu. Assouad, Fano, and Le Cam. In *Festschrift for Lucien Le Cam*, pages 423–435. Springer, 1997.

Dengyong Zhou, Jiayuan Huang, and Bernhard Scholkopf. Learning from labeled and unlabeled data on a directed graph. *International Conference on Machine Learning*, 22: 1036–1043, 2005.

Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. *International Conference on Machine Learning*, 20: 912–919, 2003.

Journal of Machine Learning Research 18 (2018) 1-86 Submitted 9/16; Revised 3/17; Published 4/18

Community Detection and Stochastic Block Models: Recent Developments

Emmanuel Abbe EABBE@PRINCETON.EDU
Program in Applied and Computational Mathematics
and Department of Electrical Engineering
Princeton University
Princeton, NJ 08544, USA

Editor: Edoardo M. Airoldi

Abstract

The stochastic block model (SBM) is a random graph model with planted clusters. It is widely employed as a canonical model to study clustering and community detection, and provides generally a fertile ground to study the statistical and computational tradeoffs that arise in network and data sciences.

This note surveys the recent developments that establish the fundamental limits for community detection in the SBM, both with respect to information-theoretic and computational thresholds, and for various recovery requirements such as exact, partial and weak recovery (a.k.a., detection). The main results discussed are the phase transitions for exact recovery at the Chernoff-Hellinger threshold, the phase transition for weak recovery at the Kesten-Stigum threshold, the optimal distortion-SNR tradeoff for partial recovery, the learning of the SBM parameters and the gap between information-theoretic and computational thresholds.

The note also covers some of the algorithms developed in the quest of achieving the limits, in particular two-round algorithms via graph-splitting, semi-definite programming, linearized belief propagation, classical and nonbacktracking spectral methods. A few open problems are also discussed.

Keywords: Community detection, clustering, stochastic block models, random graphs, unsupervised learning, spectral algorithms, computational gaps, network data analysis.

Contents

1. Introduction

1.1 Community detection

The[1] most basic task of *community detection*, or *graph clustering*, consists in partitioning the vertices of a graph into clusters that are more densely connected. From a more general point of view, community structures may also refer to groups of vertices that connect similarly to the rest of the graphs without having necessarily a higher inner density, such as disassortative communities that have higher external connectivity. Note that the terminology of 'community' is sometimes used only for assortative clusters in the literature, but we adopt here the more general definition. Community detection may also be performed on graphs where edges have labels or intensities, and if these labels represent similarities among data points, the problem may be called *data clustering*. In this monograph, we will use the terms communities and clusters exchangeably. Further one may also have access to interactions that go beyond pairs of vertices, such as in hypergraphs, and communities may not always be well separated due to overlaps. In the most general context, community detection refers to the problem of inferring similarity classes of vertices in a network by having access to measurements of local interactions.

Community detection and clustering are central problems in machine learning and data mining. A vast amount of data sets can be represented as a network of interacting items, and one of the first features of interest in such networks is to understand which items are "alike," as an end or as preliminary step towards other learning tasks. Community detection is used in particular to understand sociological behavior Goldenberg et al. (2010); Fortunato (2010); Newman et al., protein to protein interactions Chen and Yuan (2006); Marcotte et al. (1999), gene expressions Cline et al. (2007); Jiang et al. (2004), recommendation systems Linden et al. (2003); Sahebi and Cohen (2011); Wu et al. (2015), medical prognosis Sorlie et al. (2001), DNA 3D folding Cabreros et al. (2015), image segmentation Shi and Malik (1997), natural language processing Ball et al. (2011a), product-customer segmentation Clauset et al. (2004), webpage sorting Kumar et al. (1999) and more.

The field of community detection has been expanding greatly since the 80's, with a remarkable diversity of models and algorithms developed in different communities such as machine learning, network science, social science and statistical physics. These rely on various benchmarks for finding clusters, in particular cost functions based on cuts or Girvan-Newman modularity Girvan and Newman (2002). We refer to Newman (2010); Fortunato (2010); Goldenberg et al. (2010); Newman et al. for an overview of these developments.

Various fundamental questions remain nonetheless unsettled such as:

- Are there really communities? Algorithms may output community structures, but are these meaningful or artefacts?
- Can we always extract the communities when they are present; fully, partially?

1. This manuscript is the evolution of notes written for our tutorial at ISIT 2015 with M. Wainwright on Machine Learning and Information Theory, a review article for the Information Theory Newsletter, and now an extended version for a Special Issue of the Journal of Machine Learning Research. The initial goal was to explain without all the technical details and in a more general context some of the recent papers that had been written with several collaborators, but it ended up as a broader overview paper on the recent developments for the stochastic block model (with a few new additions). It mainly covers results up to 2016.

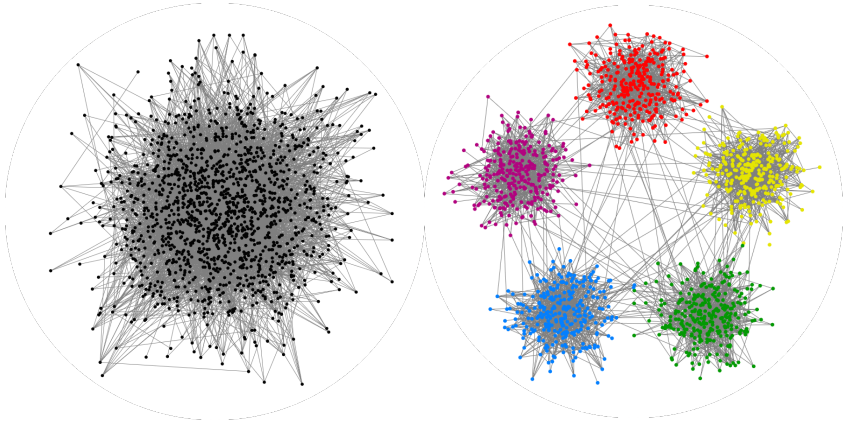


Figure 1: The above two graphs are the same graph re-organized and drawn from the SBM model with 1000 vertices, 5 balanced communities, within-cluster probability of 1/50 and across-cluster probability of 1/1000. The goal of community detection in this case is to obtain the right graph (with the true communities) from the left graph (scrambled) up to some level of accuracy. In such a context, community detection may be called graph clustering. In general, communities may not only refer to denser clusters but more generally to groups of vertices that behave similarly.

- What is a good benchmark to measure the performance of algorithms, and how good are the current algorithms?

The goal of this survey is to describe recent developments aiming at answering these questions in the context of the stochastic block model. The stochastic block model (SBM) has been used widely as a canonical model for community detection. It is arguably the simplest model of a graph with communities (see definitions in the next section). Since the SBM is a generative model for the data, it benefits from a ground truth for the communities, which allows to consider the previous questions in a formal context. On the flip side, one has to hope that the model represents a good fit for real data, which does not mean necessarily a realistic model but at least an insightful one. We believe that, similarly to the role of the discrete memoryless channel in communication theory, the SBM provides such a level of insightful abstraction. The basic model captures some of the key bottleneck phenomena, and it can be extended to more advanced and realistic refinements (such edge labels or overlapping communities). Our focus will be here on the fundamental understanding of the core SBM, without diving too much into the refined extensions.

The core SBM is defined as follows. For positive integers n, k, a probability vector p of dimension k, and a symmetric matrix W of dimension $k \times k$ with entries in $[0, 1]$, the model $\mathrm{SBM}(n, p, W)$ defines an n-vertex random graph with labelled vertices, where each vertex is assigned a community label in $\{1, \ldots, k\}$ independently under the community prior p, and

pairs of vertices with labels i and j connect independently with probability $W_{i,j}$. Further generalizations allow for labelled edges and continuous vertex labels, connecting to low-rank approximation models and 'graphons.'

A first hint on the centrality of the SBM comes from the fact that the model appeared independently in numerous scientific communities. It appeared under the SBM terminology in the context of social networks, in the machine learning and statistics literature Holland et al. (1983), while the model is typically called the planted partition model in theoretical computer science Bui et al. (1987); Dyer and Frieze (1989); Boppana (1987), and the inhomogeneous random graph in the mathematics literature Bollobás et al. (2007). The model takes also different interpretations, such as a planted spin-glass model Decelle et al. (2011), a sparse-graph code Abbe and Sandon (2015a,b) or a low-rank (spiked) random matrix model McSherry (2001); Vu (2014); Deshpande et al. (2015) among others.

In addition, the SBM has recently turned into more than a model for community detection. It provides a fertile ground for studying various central questions in machine learning, computer science and statistics: It is rich in phase transitions Decelle et al. (2011); Massoulié (2014); Mossel et al. (2014a); Abbe et al. (2016); Abbe and Sandon (2015b), allowing to study the interplay between statistical and computational barriers Y. Chen (2014); Abbe and Sandon (2015c); Banks et al. (2016); Abbe and Sandon (2017), as well as the discrepancies between probabilstic and adversarial models Moitra et al. (2016), and it serves as a test bed for algorithms, such as SDPs Abbe et al. (2016); B. Hajek (2014); Hajek et al. (2015c); Guédon and Vershynin (2016); Amini and Levina (2014); Montanari and Sen (2016); Perry and Wein (2015), spectral methods Vu (2014); Massoulié (2014); Krzakala et al. (2013); Bordenave et al. (2015); Yun and Proutiere (2014), and belief propagation Decelle et al. (2011); Abbe and Sandon (2015).

1.2 Inference on graphs

Variants of block models where edges can have labels, or where communities can overlap, allow to cover at broad set of problems in machine learning. For example, a spiked Wigner model with observation $Y = XX^T + Z$, where X is an unknown vector and Z is Wigner, can be viewed as a labeled graph where edge-(i, j)'s label is given by $Y_{ij} = X_i X_j + Z_{ij}$. If the X_i's take discrete values, e.g., $\{1, -1\}$, this is closely related to the stochastic block model—see Deshpande et al. (2015) for a precise connection. The classical data clustering problem Shalev-Shwartz and Ben-David (2014), with a matrix of similarities or dissimilarities between n points, can also be viewed as a graph with labeled edges, and generalized block models provide probabilistic models to generate such graphs, when requiring continuous labels to model Euclidean connectivity kernels. In general, models where a collection of variables $\{X_i\}$ have to be recovered from noisy observations $\{Y_{ij}\}$ that are stochastic functions of X_i, X_j, or more generally that depend on local interactions of some of the X_i's, can be viewed as inverse problems on graphs or hypergraphs that bear similarities with the basic community detection problems discussed here. This concerns in particular topic modelling, ranking, synchronization problems and other unsupervised learning problems. The specificity of the core stochastic block model is that the input variables are usually discrete.

A general abstraction of these problems can also be obtained from an information theoretic point of view, with graphical channels Abbe and Montanari (2015), themselves a special case of conditional random fields Lafferty (2001), which model conditional distributions between a collection of vertex variables X^V and a collection of edge variables Y^E on a hyper-graph $G = (V, E)$, where the conditional probability distributions factors over each edge with a local kernel Q:

$$P(y^E|x^V) = \prod_{I \in E} Q_I(y_I|x[I]),$$

where y_I is the realization of Y on the hyperedge I and $x[I]$ is the realization of X^V over the vertices incident to the hyperedge I. Our goal in this note is to discuss tools and methods for the SBM that are likely to extend to the analysis of such general models.

1.3 Fundamental limits, phase transitions and algorithms

This note focus on the *fundamental limits* of community detection, with respect to various recovery requirements. The term 'fundamental limit' here is used to emphasize the fact that we seek conditions for recovery that are *necessary and sufficient*. In the information-theoretic sense, this means finding conditions under which a given task can or cannot be solved irrespective of complexity or algorithmic considerations, whereas in the computational sense, this further constraints the algorithms to run in polynomial time in the number of vertices. As we shall see in this note, such fundamental limits are often expressed through *phase transition* phenomena, which provide sharp transitions in the relevant regimes between phases where the given task can or cannot be resolved. In particular, identifying the bottleneck regime and location of the phase transition will typically characterize the behavior of the problem in almost any other regime.

Phase transitions have proved to be often instrumental in the developments of algorithms. A prominent example is Shannon's coding theorem Shannon (1948), that gives a sharp threshold for coding algorithms at the channel capacity, and which has led the development of coding algorithms for more than 60 years (e.g., LDPC, turbo or polar codes) at both the theoretical and practical level Richardson and Urbanke (2001). Similarly, the SAT threshold Achlioptas et al. (2005) has driven the developments of a variety of satisfiability algorithms such as survey propagation Mézard et al. (2003).

In the area of clustering and community detection, where establishing rigorous benchmarks is a long standing challenge, the quest of fundamental limits and phase transition is likely to impact the development of algorithms. In fact, this has already taken place as discussed in this note, such as with two-rounds algorithms or nonbacktracking spectral methods discussed in Section 3 and 4.

1.4 Network data analysis

This note focus on the fundamentals of community detection, but we want to illustrate here how the developed theory can impact real data applications. We use the blogosphere data set from the 2004 US political elections Adamic and Glance (2005) as an archetype example.

Consider the problem where one is interested in extracting features about a collection of items, in our case $n = 1,222$ individuals writing about US politics, observing only some form of their interactions. In our example, we have access to which blogs refers to which (via hyperlinks), but nothing else about the content of the blogs. The hope is to still extract knowledge about the individual features from these simple interactions.

To proceed, build a *graph of interaction* among the n individuals, connecting two individuals if one refers to the other, ignoring the direction of the hyperlink for simplicity. Assume next that the data set is generated from a stochastic block model; assuming two communities is an educated guess here, but one can also estimate the number of communities using the methods discussed in Section 7. The type of algorithms developed in Sections 4 and 3 can then be run on this data set, and two assortative communities are obtained. In the paper Adamic and Glance (2005), Adamic and Glance recorded which blogs are right or left leaning, so that we can check how much agreement these algorithm give with this partition of the blogs. The state-of-the-art algorithms give an agreement of roughly 95% with the groundtruth Newman (2011); Jin (2015); Gao et al. (2015). Therefore, by only observing simple pairwise interactions among these blogs, without any further information on the content of the blogs, we can infer about 95% of the blogs' political inclinations.

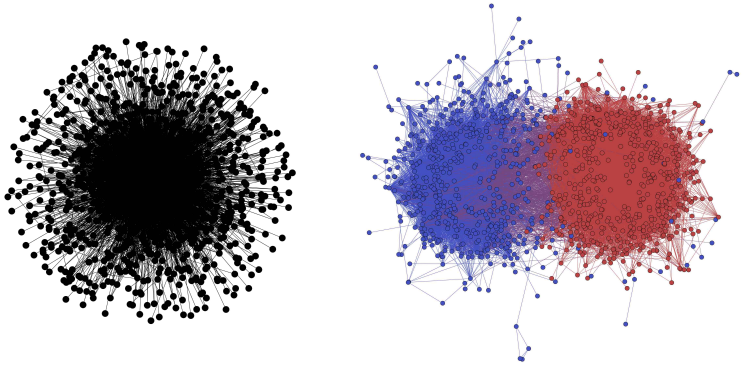


Figure 2: The above graphs represent the real data set of the political blogs from Adamic and Glance (2005). Each vertex represents a blog and each edge represents the fact that one of the blogs refers to the other. The left graph is plotted with a random arrangement of the vertices, and the right graph is the output of the ABP algorithm described in Section 4, which gives 95% accuracy on the reconstruction of the political inclination of the blogs (blue and red colors correspond to left and right leaning blogs).

Despite the fact that the blog data set is particularly 'well behaved'—there are two dominant clusters (potentially more with moderate blogs) that are balanced and well separated—the above approach can be applied to a broad collection of data sets to extract knowledge about the data from graphs of similarities. In some applications, the graph of similarity is obvious (such as in social networks with friendships), while in others, it is engineered from the data set based on metrics of similarity that need to be chosen properly. In any

case, the goal is to apply such an approach to problems for which the ground truth is unknown, such as to understand biological functionality of protein complexes; to find genetically related sub-populations; to make accurate recommendations; medical diagnosis; image classification; segmentation; page sorting; and more.

In such cases where the ground truth is not available, a key question is to understand how reliable the algorithms' outputs may be. On this matter, the theory discussed in this note gives a new perspective as follows. Following the definitions from Sections 4 and 3, the parameters estimated by fitting an SBM on this data set in the constant degree regime are

$$p_1 = 0.48,\ \ p_2 = 0.52, \quad Q = \begin{pmatrix} 52.06 & 5.16 \\ 5.16 & 47.43 \end{pmatrix}. \tag{1}$$

and in the logarithmic degree regime

$$p_1 = 0.48,\ \ p_2 = 0.52, \quad Q = \begin{pmatrix} 7.31 & 0.73 \\ 0.73 & 6.66 \end{pmatrix}. \tag{2}$$

Following the definitions of Theorem 42 from Section 4, we can now compute the SNR for these parameters in the constant-degree regime, obtaining $\lambda_2^2/\lambda_1 \approx 18$ which is much greater than 1. Thus, under an SBM model, the data is largely in a regime where communities can be detected, i.e., above the weak recovery threshold. Following the definitions of Theorem 14 from Section 3, we can also compute the CH-divergence for these parameters in the logarithmic-degree regime, obtaining $J(p, Q) \approx 2$ which is also greater than 1. Thus, under an SBM model, the data is a regime where the graph clusters could in fact be recovered entirely, i.e, above the exact recovery threshold. This does not answer whether the SBM is a good or bad model, but it gives that under this model, the data appears in a very good 'clustering regime.' This is of course counting on the fact that $n = 1,222$ is large enough to trust the asymptotic analysis. Had the SNR been too small, the model would have given us less confidence about the cluster outputs. This is the type of confidence insight that the study of fundamental limits can provide.

1.5 Brief historical overview of recent developments

This section provides a brief historical overview of the recent developments discussed in this monograph. The resurged interest in the SBM and its 'modern study' has been initiated in big part due to the paper of Decelle, Krzakala, Moore, Zdeborová Decelle et al. (2011), which conjectured[2] phase transition phenomena for the weak recovery (a.k.a. detection) problem at the Kesten-Stigum threshold and the information-computation gap at 4 symmetric communities in the symmetric case. These conjectures are backed in Decelle et al. (2011) with strong insights from statistical physics, based on the cavity method (belief propagation), and provide a detailed picture of the weak recovery problem, both for the

2. The conjecture of the Kesten-Stigum threshold in Decelle et al. (2011) was formulated with what we call in this note the max-detection criteria, asking for an algorithm to output a reconstruction of the communities that strictly improves on the trivial performance achieved by putting all the vertices in the largest community. This conjecture is formally incorrect for general SBMs, see Abbe and Sandon (2017) for a counter-example, as the notion of max-detection is too strong in some cases. The conjecture is always true for symmetric SBMs, as re-stated in Mossel et al. (2015), but it requires a different notion of detection to hold for general SBMs Abbe and Sandon (2017)—see Section 4.

algorithmic and information-theoretic behavior. Paper Decelle et al. (2011) opened a new research avenue driven by establishing such phase transitions.

One of the first paper that obtains a non-trivial algorithmic result for the weak recovery problem is Coja-Oghlan (2010) from 2010, which appeared before the conjecture (and does not achieve the threshold by a logarithmic degree factor). The first paper to make progress on the conjecture is Mossel et al. (2015) from 2012, which proves the impossibility part of the conjecture for two symmetric communities, introducing various key concepts in the analysis of block models. In 2013, Mossel et al. (2013) also obtains a result on the partial recovery of the communities, expressing the optimal fraction of mislabelled vertices when the signal-to-noise ratio is large enough in terms of the broadcasting problem on trees Kesten and Stigum (1966); Evans et al. (2000).

The positive part of the conjecture for efficient algorithm and two communities was first proved in 2014 with Massoulié (2014) and Mossel et al. (2014a), using respectively a spectral method from the matrix of self-avoiding walks and weighted non-backtracking walks between vertices.

In 2014, Abbe et al. (2014); Abbe et al. (2016) and Mossel et al. (2014b) found that the exact recovery problem for two symmetric communities has also a phase transition, in the logarithmic rather than constant degree regime, further shown to be efficiently achievable. This relates to a large body of work from the first decades of research on the SBM Bui et al. (1987); Dyer and Frieze (1989); Boppana (1987); Snijders and Nowicki (1997); Condon and Karp (1999); McSherry (2001); Bickel and Chen (2009); Choi et al. (2012); Vu (2014); Y. Chen (2014), driven by the exact or almost exact recovery problems without sharp thresholds.

In 2015, the phase transition for exact recovery is obtained for the general SBM Abbe and Sandon (2015b,c), and shown to be efficiently achievable irrespective of the number of communities. For the weak recovery problem, Bordenave et al. (2015) shows that the Kesten-Stigum threshold can be achieved with a spectral method based on the nonbacktracking (edge) operator in a fairly general setting (covering SBMs that are not necessarily symmetric), but falling short to settle the conjecture for more than two communities in the symmetric case due to technical reasons. The approach of Bordenave et al. (2015) is based on the 'spectral redemption' conjecture made in 2013 in Krzakala et al. (2013), which introduces the use of the nonbacktracking operator as a linearization of belief propagation. This is arguably the most elegant approach to the weak recovery problem, besides for the fact that the matrix is not symmetrical (to work with a symmetric matrix, the first proof of Massoulié (2014) provides also a clean description via self-avoiding walks). The general conjecture for arbitrary many symmetric or asymmetric communities is settled later in 2015 with Abbe and Sandon (2015); Abbe and Sandon (2016b), relying on a higher-order nonbacktracking matrix and a message passing implementation. It is further shown in Abbe and Sandon (2015); Abbe and Sandon (2017) that it is possible to cross information-theoretically the Kesten-Stigum threshold in the symmetric case at 4 communities, settling both positive parts of the conjectures from Decelle et al. (2011). Crossing at 5 communities is also obtained in Banks and Moore (2016); Banks et al. (2016), which further obtains the scaling of the information-theoretic threshold for a growing number of communities.

In 2016, a tight expression is obtained for partial recovery with two communities in the regime of finite SNR with diverging degrees in Deshpande et al. (2015) and Mossel and Xu

(2015) for different distortion measures. This also gives the threshold for weak recovery in the regime where the SNR in the regime of finite SNR with diverging degrees.

Other major lines of work on the SBM have been concerned with the performance of SDPs, with a precise picture obtained in Guédon and Vershynin (2016); Montanari and Sen (2016); Javanmard et al. (2016) for the weak recovery problem and in Abbe et al. (2016); B. Hajek (2014); Amini and Levina (2014); Bandeira (2015); Agarwal et al. (2015); Perry and Wein (2015) for the (almost) exact recovery problem, as well as spectral methods on classical operators McSherry (2001); Coja-Oghlan (2010); Chin et al. (2015); Xu et al. (2014); Vu (2014); Yun and Proutiere (2014); Yun and Proutiere (2015). A detailed picture has also been developed for the problem of a single planted community in Montanari (2015); Hajek et al. (2015b,a); Caltagirone et al. (2016). There is a much broader list of works on the SBMs that is not covered in this paper, specially before the 'recent developments' discussed above but also after. It is particularly challenging to track the vast literature on this subject as it is split between different communities of statistics, machine learning, mathematics, computer science, information theory, social sciences and statistical physics. This monograph covers developments mainly until 2016. There a few additional surveys available. Community detection and more generally statistical network models are discussed in Newman (2010); Fortunato (2010); Goldenberg et al. (2010), and C. Moore has a recent overview paper Moore (2017) that focuses on the weak recovery problem with emphasis on the cavity method.

The main thresholds proved for weak and exact recovery are summarized in the table below:

	Exact recovery (logarithmic degrees)
2-SSBM	$\lvert\sqrt{a}-\sqrt{b}\rvert>\sqrt{2}$ Abbe et al. (2014); Mossel et al. (2014b)
General SBM	$\min_{i<j} D_+((PQ)_i,(PQ)_j)>1$ Abbe and Sandon (2015b)

	Weak recovery (detection) (constant degrees)
2-SSBM	$(a-b)^2>2(a+b)$ Massoulié (2014); Mossel et al. (2014a)
General SBM	$\lambda_2^2(PQ)>\lambda_1(PQ)$ Bordenave et al. (2015); Abbe and Sandon (2015)

1.6 Outline

In the next section, we formally define the SBM and various recovery requirements for community detection, namely weak, partial and exact recovery. We then describe in Sections 3, 4, 5, 6 recent results that establish the fundamental limits for these recovery requirements. We further discuss in Section 7 the problem of learning the SBM parameters, and give a list of open problems in Section 8.

2. The Stochastic Block Model

The history of the SBM is long, and we omit a comprehensive treatment here. As mentioned earlier, the model appeared independently in multiple scientific communities: the terminology SBM, which seems to have dominated in the recent years, comes from the machine learning and statistics literature Holland et al. (1983), while the model is typically

called the planted partition model in theoretical computer science Bui et al. (1987); Dyer and Frieze (1989); Boppana (1987), and the inhomogeneous random graphs model in the mathematics literature Bollobás et al. (2007).

2.1 The general SBM

Definition 1 *Let n be a positive integer (the number of vertices), k be a positive integer (the number of communities), $p=(p_1,\dots,p_k)$ be a probability vector on $[k]:=\{1,\dots,k\}$ (the prior on the k communities) and W be a $k\times k$ symmetric matrix with entries in $[0,1]$ (the connectivity probabilities). The pair (X,G) is drawn under $\mathrm{SBM}(n,p,W)$ if X is an n-dimensional random vector with i.i.d. components distributed under p, and G is an n-vertex simple graph where vertices i and j are connected with probability W_{X_i,X_j}, independently of other pairs of vertices. We also define the community sets by $\Omega_i=\Omega_i(X):=\{v\in[n]: X_v=i\}, i\in[k]$.*

Thus the distribution of (X,G) where $G=([n],E(G))$ is defined as follows, for $x\in[k]^n$ and $y\in\{0,1\}^{\binom{n}{2}}$,

$$\mathbb{P}\{X=x\}:=\prod_{u=1}^n p_{x_u}=\prod_{i=1}^k p_i^{|\Omega_i(x)|} \tag{3}$$

$$\mathbb{P}\{E(G)=y|X=x\}:=\prod_{1\le u<v\le n} W_{x_u,x_v}^{y_{uv}}(1-W_{x_u,x_v})^{1-y_{uv}} \tag{4}$$

$$=\prod_{1\le i\le j\le k} W_{i,j}^{N_{ij}(x,y)}(1-W_{i,j})^{N_{ij}^c(x,y)} \tag{5}$$

where,

$$N_{ij}(x,y):=\sum_{u<v,x_u=i,x_v=j}\mathbb{1}(y_{uv}()=1), \tag{6}$$

$$N_{ij}^c(x,y):=\sum_{u<v,x_u=i,x_v=j}\mathbb{1}(y_{uv}=0)=|\Omega_i(x)||\Omega_j(x)|-N_{ij}(x,y),\quad i\neq j \tag{7}$$

$$N_{ii}^c(x,y):=\sum_{u<v,x_u=i,x_v=i}\mathbb{1}(y_{uv}=0)=|\Omega_i(x)|(|\Omega_i(x)|-1)/2-N_{ii}(x,y), \tag{8}$$

which are the number of edges and non-edges between any pair of communities. We may also talk about G drawn under $\mathrm{SBM}(n,p,W)$ without specifying the underlying community labels X.

Remark 2 *Besides for Section 8, we assume that p does not scale with n, whereas W typically does. As a consequence, the number of communities does not scale with n and the communities have linear size. Nonetheless, various results discussed in this note should extend (by inspection) to cases where k is growing slowly enough.*

Remark 3 *Note that by the law of large numbers, almost surely,*

$$\frac{1}{n}|\Omega_i|\to p_i.$$

Alternative definitions of the SBM require X to be drawn uniformly at random with the constraint that $\frac{1}{n}|\{v \in [n] : X_v = i\}| = p_i + o(1)$, or $\frac{1}{n}|\{v \in [n] : X_v = i\}| = p_i$ for consistent values of n and p (e.g., $n/2$ being an integer for two symmetric communities). For the purpose of this paper, these definitions are essentially equivalent.

2.2 The symmetric SBM

The SBM is called symmetric if p is uniform and if W takes the same value on the diagonal and the same value outside the diagonal.

Definition 4 *(X, G) is drawn under $\mathrm{SSBM}(n, k, A, B)$, if $p = \{1/k\}^k$ and W takes value A on the diagonal and B off the diagonal.*

Note also that if all entries of W are the same, then the SBM collapses to the Erdős-Rényi random graph, and no meaningful reconstruction of the communities is possible.

2.3 Recovery requirements

The goal of community detection is to recover the labels X by observing G, up to some level of accuracy. We next define the agreement, also called sometimes overlap.

Definition 5 (Agreement) *The agreement between two community vectors $x, y \in [k]^n$ is obtained by maximizing the common components between x and any relabelling of y, i.e.,*

$$A(x, y) = \max_{\pi \in S_k} \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(x_i = \pi(y_i)), \tag{9}$$

where S_k is the group of permutations on $[k]$.

Note that the relabelling permutation is used to handle symmetric communities such as in SSBM, as it is impossible to recover the actual labels in this case, but we may still hope to recover the *partition*. In fact, one can alternatively work with the community partition $\Omega = \Omega(X)$, defined earlier as the unordered collection of the k disjoint unordered subsets $\Omega_1, \ldots, \Omega_k$ covering $[n]$ with $\Omega_i = \{u \in [n] : X_u = i\}$. It is however often convenient to work with vertex labels. Further, upon solving the problem of finding the partition, the problem of assigning the labels is often a much simpler task. It cannot be resolved if symmetry makes the community label non identifiable, such as for SSBM, and it is trivial otherwise by using the community sizes and clusters/cuts densities.

For $(X, G) \sim \mathrm{SBM}(n, p, W)$ one can always attempt to reconstruct X without even taking into account G, simply drawing each component of $\hat{X}$ i.i.d. under p. Then the agreement satisfies almost surely

$$A(X, \hat{X}) \to \|p\|_2^2, \tag{10}$$

and $\|p\|_2^2 = 1/k$ in the case of p uniform. Thus an agreement becomes interesting only when it is above this value.

One can alternatively define a notion of component-wise agreement. Define the overlap between two random variables X, Y on $[k]$ as

$$O(X, Y) = \sum_{z \in [k]} (\mathbb{P}\{X = z, Y = z\} - \mathbb{P}\{X = z\}\mathbb{P}\{Y = z\}) \tag{11}$$

and $O^*(X, Y) = \max_{\pi \in S_k} O(X, \pi(Y))$. In this case, for $X, \hat{X}$ i.i.d. under p, we have $O^*(X, \hat{X}) = 0$.

All recovery requirement in this note are going to be asymptotic, taking place with high probability as n tends to infinity. We also assume in the following sections—except for Section 7—that the parameters of the SBM are known when designing the algorithms.

Definition 6 *Let $(X, G) \sim \mathrm{SBM}(n, p, W)$. The following recovery requirements are solved if there exists an algorithm that takes G as an input and outputs $\hat{X} = \hat{X}(G)$ such that*

- **Exact recovery:** $\mathbb{P}\{A(X, \hat{X}) = 1\} = 1 - o(1)$,
- **Almost exact recovery:** $\mathbb{P}\{A(X, \hat{X}) = 1 - o(1)\} = 1 - o(1)$,
- **Partial recovery:** $\mathbb{P}\{A(X, \hat{X}) \geq \alpha\} = 1 - o(1)$, $\alpha \in (0, 1)$.

In other words, exact recovery requires the entire partition to be correctly recovered, almost exact recovery allows for a vanishing fraction of misclassified vertices and partial recovery allows for a constant fraction of misclassified vertices. We call α the agreement or accuracy of the algorithm.

Different terminologies are sometimes used in the literature, with following equivalences:

- exact recovery $\iff$ strong consistency
- almost exact recovery $\iff$ weak consistency

Sometimes 'exact recovery' is also called just 'recovery' and 'almost exact recovery' is called 'strong recovery.'

As mentioned above, that values of α that are too small may not be interesting or possible. In the symmetric SBM with k communities, an algorithm that that ignores the graph and simply draws $\hat{X}$ i.i.d. under p achieves an accuracy of $1/k$. Thus the problem becomes interesting when $\alpha > 1/k$, leading to the following definition.

Definition 7 *Weak recovery or detection is solved in $\mathrm{SSBM}(n, k, A, B)$ if for $(X, G) \sim \mathrm{SSBM}(n, k, A, B)$, there exists $\varepsilon > 0$ and an algorithm that takes G as an input and outputs $\hat{X}$ such that $\mathbb{P}\{A(X, \hat{X}) \geq 1/k + \varepsilon\} = 1 - o(1)$.*

Equivalently, $\mathbb{P}\{O^*(X_V, \hat{X}_V) \geq \varepsilon\} = 1 - o(1)$ where V is uniformly drawn in $[n]$. Determining the counterpart of weak recovery in the general SBM requires some discussion. Consider an SBM with two communities of relative sizes $(0.8, 0.2)$. A random guess under this prior gives an agreement of $0.8^2 + 0.2^2 = 0.68$, however an algorithm that simply puts every vertex in the first community achieves an agreement of 0.8. In Decelle et al. (2011), the latter agreement is considered as the one to improve upon in order to detect communities, leading to the following definition:

Definition 8 *Max-detection is solved in $\mathrm{SBM}(n, p, W)$ if for $(X, G) \sim \mathrm{SBM}(n, p, W)$, there exists $\varepsilon > 0$ and an algorithm that takes G as an input and outputs $\hat{X}$ such that $\mathbb{P}\{A(X, \hat{X}) \geq \max_{i \in [k]} p_i + \varepsilon\} = 1 - o(1)$.*

As shown in Abbe and Sandon (2017), previous definition is however not the right definition to capture the Kesten-Stigum threshold in the general case. In other words, the conjecture that max-detection is always possible above the Kesten-Stigum threshold is not accurate in general SBMs. Back to our example with communities of relative sizes $(0.8, 0.2)$, an algorithm that could find a set containing $2/3$ of the vertices from the large community and $1/3$ of the vertices from the small community would not satisfy the above above detection criteria, while the algorithm produces nontrivial amounts of evidence on what communities the vertices are in. To be more specific, consider a two community SBM where each vertex is in community 1 with probability 0.99, each pair of vertices in community 1 have an edge between them with probability $2/n$, while vertices in community 2 never have edges. Regardless of what edges a vertex has it is more likely to be in community 1 than community 2, so detection according to the above definition is not impossible, but one can still divide the vertices into those with degree 0 and those with positive degree to obtain a non-trivial detection—see Abbe and Sandon (2017) for a formal counter-example.

A fix fo this issue is to consider a weighted notion of agreement, i.e.,

$$\tilde{A}(x,y) = \max_{\pi \in S_k} \frac{1}{k} \sum_{i=1}^{k} \frac{\sum_{u \in [n]} \mathbb{1}(x_u = \pi(y_u), x_u = i)}{\sum_{u \in [n]} \mathbb{1}(x_u = i)}, \tag{12}$$

which counts the number of agreeing components (up to relabellings) normalized by the size of the communities. Weak recovery (or detection) can then be defined as obtaining with high probability a weighted agreement of

$$\tilde{A}(X, \hat{X}(G)) = 1/k + \Omega_n(1),$$

and this applies to the general SBM. Another definition of detection that seems easier to manipulate and that implies the previous one is as follows; note that this definition requires a single partition even for the general SBM.

Definition 9 Weak recovery or detection *is solved in* $\mathrm{SBM}(n,p,W)$ *if for* $(X,G) \sim \mathrm{SBM}(n,p,W)$*, there exists* $\varepsilon > 0$*,* $i,j \in [k]$ *and an algorithm that takes* G *as an input and outputs a partition of* $[n]$ *into two sets* (S, S^c) *such that*

$$\mathbb{P}\{|\Omega_i \cap S|/|\Omega_i| - |\Omega_j \cap S|/|\Omega_j| \geq \epsilon\} = 1 - o(1),$$

where we recall that $\Omega_i = \{u \in [n] : X_u = i\}$.

In other words, an algorithm solves detection if it divides the graph's vertices into two sets such that vertices from two different communities have different probabilities of being assigned to one of the sets. With this definition, putting all vertices in one community does not detect, since $|\Omega_i \cap S|/|\Omega_i| = 1$ for all $i \in [k]$. Further, in the symmetric SBM, this definition implies Definition 7 provided that we fix the output:

Lemma 10 *If an algorithm solves detection in the sense of Definition 11 for a symmetric SBM, then it solves max-detection (or detection according to Decelle et al.'s definition), provided that we consider it as returning* $k-2$ *empty sets in addition to its actual output.*

See Abbe and Sandon (2016b) for the proof. The above is likely to extend to other weakly symmetric SBMs, i.e., that have constant expected degree, but not all.

Finally, note that our notion of detection requires to separate at least two communities $i,j \in [k]$. One may ask for a definition where two specific communities need to be separated:

Definition 11 *Separation of communities* i *and* j*, with* $i,j \in [k]$*, is solved in* $\mathrm{SBM}(n,p,W)$ *if for* $(X,G) \sim \mathrm{SBM}(n,p,W)$*, there exists* $\varepsilon > 0$ *and an algorithm that takes* G *as an input and outputs a partition of* $[n]$ *into two sets* (S, S^c) *such that*

$$\mathbb{P}\{|\Omega_i \cap S|/|\Omega_i| - |\Omega_j \cap S|/|\Omega_j| \geq \epsilon\} = 1 - o(1).$$

There are at least two additional questions that are natural to ask about SBMs, both can be asked for efficient or information-theoretic algorithms:

- **Distinguishability or testing:** Consider an hypothesis test where a random graph G is drawn with probability $1/2$ from an SBM model (with same expected degree in each community) and with probability $1/2$ from an Erdős-Rényi model with matching expected degree. Is is possible to decide with asymptotic probability $1/2+\varepsilon$ for some $\varepsilon > 0$ from which ensemble the graph is drawn? This requires the total variation between the two ensembles to be non-vanishing. This is also sometimes called 'detection', although we use here detection as an alternative terminology to weak recovery. Distinguishability is further discussed in Section 4.6.1.

- **Learnability:** Assume that G is drawn from an SBM ensemble, is it possible to obtain a consistent estimator for the parameters? E.g., can we learn k,p,Q from a graph drawn from $\mathrm{SBM}(n,p,Q/n)$? This is further discussed in Section 7.

The obvious implications are: exact recovery $\Rightarrow$ almost exact recovery $\Rightarrow$ partial recovery $\Rightarrow$ weak detection $\Rightarrow$ distinguishability. Moreover, for symmetric SBMs with two symmetric communities: learnability $\Leftrightarrow$ weak recovery $\Leftrightarrow$ distinguishability, but these are broken for general SBMs; see Section 7.

2.4 Model variants

There are various extensions of the basic SBM discussed in previous section, in particular:

- **Labelled SBMs:** allowing for edges to carry a label, which can model intensities of similarity functions between vertices (see for example Heimlicher et al. (2012); Xu et al. (2014); Jog and Loh (2015); Yun and Proutiere (2015) and further details in Section 3.5);

- **Degree-corrected SBMs:** allowing for a degree parameter for each vertex that scales the edge probabilities in order to makes expected degrees match the observed degrees (see for example Karrer and Newman (2011); **?**); **?**);

- **Overlapping SBMs:** allowing for the communities to overlap, such as in the mixed-membership SBM Airoldi et al. (2008), where each vertex has a profile of community memberships or a continuous label—see also Fortunato (2010); Newman and Peixoto (2015); Ball et al. (2011b); Peixoto (2015); Palla et al. (2005); Gopalan and Blei (2013); Abbe and Sandon (2015b) and further discussion in Section 3.5).

Another variant that circumvents the discussions about non-edges is to consider a **censored block model** (CBM), defined as follows (see Abbe et al. (2014a)).

Definition 12 (Binary symmetric CBM) *Let $G = ([n], E)$ be a graph and $\varepsilon \in [0,1]$. Let $X^n = (X_1, \ldots, X_n)$ with i.i.d. Bernoulli(1/2) components. Let Y be a random vector of dimension $\binom{n}{2}$ taking values in $\{0, 1, \star\}$ such that*

$$\mathbb{P}\{Y_{ij} = 1 | X_i = X_j, E_{ij} = 1\} = \mathbb{P}\{Y_{ij} = 0 | X_i \neq X_j, E_{ij} = 1\} = \varepsilon \tag{13}$$
$$\mathbb{P}\{Y_{ij} = \star | E_{ij} = 0\} = 1. \tag{14}$$

The case of an Erdos-Renyi graph is discussed in Heimlicher et al. (2012); Abbe et al. (2014a,b); Chin et al. (2015); Saade et al. (2015); Chen et al. (2014); Chen and Goldsmith (2014). Inserting the $\star$ symbol simplifies a few aspects compared to SBMs with respect to non-edges. In that sense, the CBM is a more convenient model than the SBM from a mathematical viewpoint, while behaving similarly to the SBM (when G is an Erdos-Renyi graph of degree $(a+b)/2$ and $\varepsilon = b/(a+b)$ for the two community symmetric case). The CBM can also be viewed as a synchronization model over the binary field, and more general synchronization models have been studied in **??**, with a complete description both at the fundamental and algorithmic level (generalizing in particular the results from Section 6.2).

Further, one can consider more general models of **inhomogenous random graphs** Bollobás et al. (2007), which attach to each vertex a label in a set that is not necessarily finite, and where edges are drawn independently from a given kernel conditionally on these labels. This gives in fact a way to model mixed-membership, and is also related to **graphons**, which corresponds to the case where each vertex has a continuous label.

It may be worth saying a few words about the theory of graphons and its implications for us. Lovász and co-authors introduced graphons Lovász and Szegedy (2006); Borgs et al. (2008); Lovász (2012) in the study of large graphs (also related to Szemerédi's Regularity Lemma Szemerédi (1976)), showing that[3] a convergent sequence of graphs admits a limit object, the graphon, that preserves many local and global properties of the sequence. Graphons can be represented by a measurable function $w : [0,1]^2 \to [0,1]$, which can be viewed as a continuous extensions of the connectivity matrix W used throughout this paper. Most relevant to us is that any network model that is invariant under node labelings, such as most models of practical interests, can be described by an edge distribution that is *conditionally independent* on hidden node labels, via such a measurable map w. This gives a de Finetti's theorem for label-invariant models Hoover (1979); Aldous (1981); Diaconis and Janson (2007), but does not require the topological theory behind it. Thus the theory of graphons may give a broader meaning to the study of block models, which are precisely building blocks to graphons, but for the sole purpose of studying exchangeable network models, inhomogeneous random graphs give enough degrees of freedom.

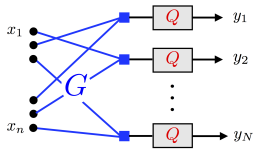
Further, many problems in machine learning and networks are also concerned with interactions of items that go beyond the pairwise setting. For example, citation or metabolic networks rely on interactions among k-tuples of vertices. In a broad context, one may thus cast the SBM and its variants into a a comprehensive class of conditional random field or

3. Initially in dense regimes and more recently for sparse regimes Borgs et al. (2014).

channel model, where edges labels depend on vertex labels,[4] such as developed in Abbe and Montanari (2015) with **graphical channels** as follows.

Let $V = [n]$ and $G = (V, E(G))$ be a hypergraph with $N = |E(G)|$. Let $\mathcal{X}$ and $\mathcal{Y}$ be two finite sets called respectively the input and output alphabets, and $Q(\cdot|\cdot)$ be a channel from $\mathcal{X}^k$ to $\mathcal{Y}$ called the kernel. To each vertex in V, assign a vertex-variable in $\mathcal{X}$, and to each edge in $E(G)$, assign an edge-variable in $\mathcal{Y}$. Let y_I denote the edge-variable attached to edge I, and $x[I]$ denote the k node-variables adjacent to I. We define a graphical channel with graph G and kernel Q as the channel $P(\cdot|\cdot)$ given by

$$P(y|x) \equiv \prod_{I \in E(G)} Q(y_I | x[I])$$
$$x \in \mathcal{X}^V, y \in \mathcal{Y}^{E(G)}$$



As we shall see for the SBM, two quantities are key to understand how much information can be carried in graphical channels: a measure on how "rich" the observation graph G is, and a measure on how "noisy" the connectivity kernel Q is. This survey quantifies the tradeoffs between these two quantities in the SBM (which corresponds to a discrete $\mathcal{X}$, a complete graph G and a specific kernel Q), in order to recover the input from the output. Similar tradeoffs are expected to take place in other graphical channels, such as in ranking, synchronization, topic modelling or other related models.

2.5 SBM regimes and topology

Before discussing when the various recovery requirements can be solved or not in SBMs, it is important to recall a few topological properties of the SBM graph.

When all the entries of W are the same and equal to w, the SBM collapses to the Erdős-Rényi model $G(n, w)$ where each edge is drawn independently with probability w. Let us recall a few basic results for this model derived mainly from Erdös and Rényi (1960):

- $G(n, c\ln(n)/n)$ is connected with high probability if and only if $c > 1$,
- $G(n, c/n)$ has a giant component (i.e., a component of size linear in n) if and only if $c > 1$,
- For $\delta < 1/2$, the neighborhood at depth $r = \delta \log_c n$ of a vertex v in $G(n, c/n)$, i.e., $B(v, r) = \{u \in [n] : d(u, v) \leq r\}$ where $d(u, v)$ is the length of the shortest path connecting u and v, tends in total variation to a Galton-Watson branching process of offspring distribution Poisson(c).

For SSBM(n, k, A, B), these results hold by essentially replacing c with the average degree.

- For $a, b > 0$, SSBM$(n, k, a\log n/n, b\log n/n)$ is connected with high probability if and only if $\frac{a+(k-1)b}{k} > 1$ (if a or b is equal to 0, the graph is of course not connected).

4. A recent paper Berthet et al. (2016) has also considered an Ising model with block structure, studying exact recovery and SDPs in this context.

- SSBM$(n, k, a/n, b/n)$ has a giant component (i.e., a component of size linear in n) if and only if $d := \frac{a+(k-1)b}{k} > 1$,
- For $\delta < 1/2$, the neighborhood at depth $r = \delta \log_d n$ of a vertex v in SSBM$(n, k, a/n, b/n)$ tends in total variation to a Galton-Watson branching process of offspring distribution Poisson(d) where d is as above.

Similar results hold for the general SBM, at least for the case of a constant excepted degrees. For connectivity, one has that SBM$(n, p, Q \log n/n)$ is connected with high probability if

$$\min_{i\in[k]} \|(\mathrm{diag}(p)Q)_i\|_1 > 1 \tag{15}$$

and is not connected with high probability if $\min_{i\in[k]} \|(\mathrm{diag}(p)Q)_i\|_1 < 1$, where $(\mathrm{diag}(p)Q)_i$ is the i-th column of $\mathrm{diag}(p)Q$.

These results are important to us as they already point regimes where exact or weak recovery is not possible. Namely, if the SBM graph is not connected, exact recovery is not possible (since there is no hope to label disconnected components with higher chance than $1/2$), hence exact recovery can take place only if the SBM parameters are in the logarithmic degree regime. In other words, exact recovery in SSBM$(n, k, a \log n/n, b \log n/n)$ is not solvable if $\frac{a+(k-1)b}{k} < 1$. This is however unlikely to provide a tight condition, i.e., exact recovery is not equivalent to connectivity, and next section will precisely investigate how much more than $\frac{a+(k-1)b}{k} > 1$ is needed to obtain exact recovery. Similarly, it is not hard to see that weak recovery is not solvable if the graph does not have a giant component, i.e., weak recovery is not solvable in SSBM$(n, k, a/n, b/n)$ if $\frac{a+(k-1)b}{k} < 1$, and we will see in Section 4 how much more is needed to go from the giant to weak recovery.

2.6 Challenges: spectral, SDP and message passing approaches

Consider the symmetric SBM with two communities, where the inner-cluster probability is a/n and the across-cluster probability is b/n, i.e., SSBM$(n, 2, a/n, b/n)$, and assume $a \geq b$. We next discuss basic approaches and the challenges that they face.

The spectral approach. Assume for simplicity that the two clusters have exactly size $n/2$, and index the first cluster with the first $n/2$ vertices. The expected adjacency matrix $\mathbb{E}A$ of this graph has four blocks given by

$$\mathbb{E}A = \begin{pmatrix} a/n \cdot 1^{n/2\times n/2} & b/n \cdot 1^{n/2\times n/2} \\ b/n \cdot 1^{n/2\times n/2} & a/n \cdot 1^{n/2\times n/2} \end{pmatrix}. \tag{16}$$

This matrix has three eigenvalues, namely $(a+b)/n$, $(a-b)/n$ and 0, where 0 has multiplicity $n-2$, and eigenvectors attached to the first two eigenvalues are

$$\left\{ \frac{a+b}{n}, \begin{pmatrix} 1^{n/2} \\ 1^{n/2} \end{pmatrix} \right\}, \left\{ \frac{a-b}{n}, \begin{pmatrix} 1^{n/2} \\ -1^{n/2} \end{pmatrix} \right\}. \tag{17}$$

Since permutations do not affect eigenvalues and permute eigenvectors, if one were to work with the expected adjacency matrix, communities could simply be recovered by taking an

eigenvector corresponding to the second largest eigenvalue, and assigning each vertex to a community depending on the sign of this eigenvector's components. Of course, we do not have access to the expected adjacency matrix, nor a tight estimate since we are observing a single shot of the SBM graph, but we can view the adjacency matrix A as a perturbation of $\mathbb{E}A$, i.e.,

$$A = \mathbb{E}A + Z$$

where $Z = (A - \mathbb{E}A)$ is the perturbation. One may hope that this perturbation is moderate, i.e., that carrying the same program as for the expected adjacency—taking the second eigenvector of A—still gives a meaningful reconstruction of the communities.

The Courant-Fisher theorem can be used to obtain a simple control on the perturbation of the eigenvalues of A from $\mathbb{E}A$. Denoting by $\lambda_1 \geq \cdots \geq \lambda_n$ and $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \cdots \geq \hat{\lambda}_n$ the ordered eigenvalues of $\mathbb{E}A$ and A respectively, we have for all $i \in [n]$,

$$|\lambda_i - \hat{\lambda}_i| \leq \|Z\|, \tag{18}$$

where $\|\cdot\|$ denotes the operator norm. Thus, if $\|Z\|$ is less than half of the least gap between the three eigenvalues $\frac{a+b}{n}$, $\frac{a-b}{n}$ and 0, the eigenvalues of A would have a preserved ordering. This would only give us hope that the eigenvectors would be correlated, but gives no guarantees so far. The Davis-Kahan Theorem can be used to that end, giving that the angle θ_i between the eigenvectors corresponding to the i-th eigenvalues has a sinus bounded as $\sin\theta_i \leq \frac{\|Z\|}{\min_{i\neq j}|\lambda_i - \lambda_j|/2}$ (this is in fact a slightly weaker statement than Davis-Kahan). Thus estimating the operator norm is crucial with this approach, and tools from random matrix theory can be used here Nadakuditi and Newman (2012); Vu (2007).

The problem with this naive approach is that it fails when a and b are too small, such as in the sparse regime (constant a, b) or even slowly growing degrees. One of the main reasons for this is that the eigenvalues are far from being properly ordered in such cases, in particular due to high degree nodes. In the sparse regime, there will be nodes of almost logarithmic degree, and these induce eigenvalues of order roughly root-logarithmic. To see this, note that an isolated star graph, i.e., a single vertex connected to k neighbors, has an eigenvalue of $\sqrt{k}$ with eigenvector having weight 1 on the center-vertex and $\sqrt{k}$ on the neighbors (recall that applying a vector to the adjacency matrix corresponds to diffusing each vector component to its neighbors).

Thus the spectrum of the adjacency matrix or standard Laplacians is blurred by 'outlier' eigenvalues in such regimes Kawamoto and Kabashima (2015), and we will need to rely on different operators that are not as sensitive. This is a real difficulty that is recurrent in practical applications of clustering algorithms. A possible approach is to try to regularize the spectrum by relying on regularized Laplacians such as in Joseph and Yu (2013); Le et al. (2015); Coja-Oghlan (2010); Vu (2014); Guédon and Vershynin (2016); Chin et al. (2015), by either trimming or shifting the matrix entries, but does not suffice in the most challenging regimes. The SBM will gives us rigorous benchmarks to understand how to design such operators. In particular, the nonbacktracking operator discussed in Section 4 will allow to run the above program (i.e., taking the second eigenvector) successfully, as it affords a 'clean' spectrum that is not contaminated by localized eigenvectors even in the weakest possible signal-to-noise regimes. As discussed next, this operator can in fact be derived as a linearization of belief propagation.

The message passing approach. Before describing the approach, let us remind ourselves of what our goals are. We defined different recovery requirements for the SBM, in particular weak and exact recovery. As discussed in Section 3, exact recovery yields a clear target for the SSBM, the optimal algorithm (minimizing the probability of failing) is the Maximum A Posteriori (MAP) algorithm that looks for the min-bisection:

$$\hat{x}_{\text{map}}(g) = \arg \max_{\substack{x \in \{+1,-1\}^n \\ x^t 1^n = 0}} x^t A x, \tag{19}$$

which is equivalent to finding a balanced ± 1 assignment to each vertex such that the number of edges with different end points is minimized. This is NP-hard in the worst-case (due to the integral constraint), but we will show that the min-bisection can be found efficiently 'with high probability' for the SBM whenever it is unique with high probability (i.e., no gap occurs!). Further, both the spectral approach described above (with some modifications) and the SDP approach described in Section 3.4 allow to achieve the threshold, and each can be viewed as a relaxation of the min-bisection problem (see Section 3.4).

For weak recovery instead, minimizing the error probability (i.e., the MAP estimator) is no longer optimal, and thus targeting the min-bisection is not necessarily the right approach. As discussed above, the spectral approach on the adjacency matrix can fail dramatically, as it will catch localized eigenvectors (e.g., high-degree nodes) rather than communities. The SDP approach seems more robust. While it does not detect communities in the most challenging regimes, it approaches the threshold fairly closely for two communities—see Javanmard et al. (2016). Nonetheless, it does not target the right figure of merit for weak recovery.

What is then the right figure of merit for weak recovery? Consider the agreement metric, i.e., minimizing the fraction of mislabelled vertices. Consider also a perturbation of the SBM parameters to a slightly asymmetric version, such that the labels can now be identified from the partition, to avoid the relabelling maximization over the communities. The agreement between the true clusters X and a reconstruction $\hat{X}$ is then given by $\sum_{v \in [n]} \mathbb{1}(X_v = \hat{X}_v(G))$, and upon observing $G = g$, the expected agreement is maximized by finding for each $v \in [n]$

$$\max_{\hat{x}_v} \mathbb{P}\{X_v = \hat{x}_v | G = g\}. \tag{20}$$

The reasons for considering an asymmetric SBM here is that the above expression is exactly equal to half in the symmetric case, which carries no information. To remediate to that, one should break the symmetry in the symmetric case by revealing some vertex labels (or use noisy labels as done in Deshpande et al. (2015), or parity of pairs of vertices). One may also relate the random agreement to its expectation with concentration arguments. These are overlooked here, but we get a hint on what the Bayes optimal algorithm should do: it should approximately maximize the posterior distribution of a single vertex given the graph. This is different than MAP which attempts to recover all vertices in one shot. In the latter context, the maximizer may not be 'typical' (e.g., the all-one vector is the most likely outcome of n i.i.d. Bernoulli(3/4) random variables, but it does not have a typical fraction of 1's).

Computing (20) is hard, as it requires computing the graph marginal which requires computing an exponential sum. This is where belief propagation comes into play, to provide a tight approximation. When the graph is a tree, the approximation is exact, and physicists have good reasons to believe that this remains true even in our context of a loopy graph Decelle et al. (2011). However, establishing such a claim rigorously is a long-standing challenge for message passing algorithms. Without a good guess on how to initialize BP, which we do not possess for the detection problem, one may simply take a random initial guess. In the symmetric case, this would still give a bias of roughly $\sqrt{n}$ vertices (from the Central Limit Theorem) towards the true partition, i.e., a initial belief of $1/2 + \Theta(1/\sqrt{n})$ towards the truth. Recall that in BP, each vertex will send its belief of being 0 or 1 to its neighbors, computing this belief using Bayes rule from the received beliefs at previous iteration, and factoring out the backtracking beliefs (i.e., do not take into account the belief of a specific vertex to update it in the next iteration). As the initial belief of $1/2$ is a trivial fix point in BP, one may attempt to approximate the update Bayes rule around the uniform beliefs, working out the linear approximation of the BP update. This gives raise to a linear operator, which is the nonbacktracking (NB) operator discussed in Section 4.5.1, and running linerazied BP corresponds to taking a power-iteration method with this operator on the original random guesses—see Section 4.5.1. Thus, we are back to a spectral method, but with an operator that is a linearization of the Bayes optimal approximation (this gave raise to the terminology 'spectral redemption' in Krzakala et al. (2013).)

The advantage of this linearized approach is that it is easier to analyze than the full BP, and one can prove statements about it, such as that it detects down to the optimal threshold Bordenave et al. (2015); Abbe and Sandon (2015). On the other hand, linerarized BP will loose some accuracy in contrast to full BP, but this can be improved by using a two-round algorithm: start with linearized BP to provably detect communities, and then enhance this reconstruction by feeding it to full BP—see for example Mossel et al. (2013); Abbe and Sandon (2016b). The latter approach gives raise to a new approach to analyzing belief propagation: can such two rounds approaches with linearization plus amplification be applied to other problems?

3. Exact Recovery

3.1 Fundamental limit and the CH threshold

Exact recovery for linear size communities has been one of the most studied problem for block models in its first decades. A partial list of papers is given by Bui et al. (1987); Dyer and Frieze (1989); Boppana (1987); Snijders and Nowicki (1997); Condon and Karp (1999); McSherry (2001); Bickel and Chen (2009); Choi et al. (2012); Vu (2014); Y. Chen (2014). In this line of work, the approach is mainly driven by the choice of the algorithms, and in particular for the model with two symmetric communities. The results look as follows[5]:

5. Some of the conditions have been borrowed from attended talks and papers and have not been checked

Bui, Chaudhuri, Leighton, Sipser '84	maxflow-mincut	$A = \Omega(1/n), B = o(n^{-1-4/((A+B)n)})$
Boppana '87	spectral meth.	$(A-B)/\sqrt{A+B} = \Omega(\sqrt{\log(n)/n})$
Dyer, Frieze '89	min-cut via degrees	$A - B = \Omega(1)$
Snijders, Nowicki '97	EM algo.	$A - B = \Omega(1)$
Jerrum, Sorkin '98	Metropolis aglo.	$A - B = \Omega(n^{-1/6+\epsilon})$
Condon, Karp '99	augmentation algo.	$A - B = \Omega(n^{-1/2+\epsilon})$
Carson, Impagliazzo '01	hill-climbing algo.	$A - B = \Omega(n^{-1/2}\log^4(n))$
McSherry '01	spectral meth.	$(A-B)/\sqrt{A} \geq \Omega(\sqrt{\log(n)/n})$
Bickel, Chen '09	N-G modularity	$(A-B)/\sqrt{A+B} = \Omega(\log(n)/\sqrt{n})$
Rohe, Chatterjee, Yu '11	spectral meth.	$A - B = \Omega(1)$

More recently, Vu Vu (2014) obtained a spectral algorithm that works in the regime where the expected degrees are logarithmic, rather than poly-logarithmic as in McSherry (2001); Choi et al. (2012). Note that exact recovery requires the node degrees to be at least logarithmic, as discussed in Section 2.5. Thus the results of Vu are tight in the scaling, and the first to apply in such great generality, but as for the other results in Table 1, they do not reveal the phase transition. The fundamental limit for exact recovery was derived first for the case of symmetric SBMs with two communities:

Theorem 13 *Abbe et al. (2014); Mossel et al. (2014b) Exact recovery in* $\mathrm{SSBM}(n, 2, a\ln(n)/n, b\ln(n)/n)$ *is solvable and efficiently so if* $|\sqrt{a}-\sqrt{b}| > \sqrt{2}$ *and unsolvable if* $|\sqrt{a}-\sqrt{b}| < \sqrt{2}$.

A few remarks regarding this result:

- At the threshold, one has to distinguish two cases: if $a, b > 0$, then exact recovery is solvable (and efficiently so) if $|\sqrt{a}-\sqrt{b}| = \sqrt{2}$ as first shown in Mossel et al. (2014b). If a or b are equal to 0, exact recovery is solvable (and efficiently so) if $\sqrt{a} > \sqrt{2}$ or $\sqrt{b} > \sqrt{2}$ respectively, and this corresponds to connectivity.
- Theorem 13 provides a necessary and sufficient condition for exact recovery, and covers all cases for exact recovery in $\mathrm{SSBM}(n, 2, A, B)$ were A and B may depend on n as long as not asymptotically equivalent (i.e., $A/B \nrightarrow 1$). For example, if $A = 1/\sqrt{n}$ and $B = \ln^3(n)/n$, which can be written as $A = \frac{\sqrt{n}}{\ln n}\frac{\ln n}{n}$ and $B = \ln^2 n \frac{\ln n}{n}$, then exact recovery is trivially solvable as $|\sqrt{a}-\sqrt{b}|$ goes to infinity. If instead $A/B \to 1$, then one needs to look at the second order terms. This is covered by Mossel et al. (2014b) for the 2 symmetric community case, which shows that for $a_n, b_n = \Theta(1)$, exact recovery is solvable if and only if $((\sqrt{a_n}-\sqrt{b_n})^2 - 1)\log n + \log\log n/2 = \omega(1)$.
- Note that $|\sqrt{a}-\sqrt{b}| > \sqrt{2}$ can be rewritten as $\frac{a+b}{2} > 1 + \sqrt{ab}$ and recall that $\frac{a+b}{2} > 2$ is the connectivity requirement in SSBM. As expected, exact recovery requires connectivity, but connectivity is not sufficient. The extra term $\sqrt{ab}$ is the 'over-sampling' factor needed to go from connectivity to exact recovery, and the connectivity threshold can be recovered by considering the case where $b = 0$. An information-theoretic interpretation of Theorem 13 is also discussed below.

We next provide the fundamental limit for exact recovery in the general SBM, in the regime of the phase transition where W scales like $\ln(n)Q/n$ for a matrix Q with positive entries.

Theorem 14 *Abbe and Sandon (2015b) Exact recovery in* $\mathrm{SBM}(n, p, \ln(n)Q/n)$ *is solvable and efficiently so if*

$$I_+(p, Q) := \min_{1\leq i<j\leq k} D_+((\mathrm{diag}(p)Q)_i \| (\mathrm{diag}(p)Q)_j) > 1$$

and is not solvable if $I_+(p, Q) < 1$, *where* D_+ *is defined by*

$$D_+(\mu\|\nu) := \max_{t\in[0,1]} \sum_x \nu(x) f_t(\mu(x)/\nu(x)), \quad f_t(y) := 1 - t + ty - y^t. \tag{21}$$

Remark 15 *Regarding the behavior at the threshold: If all the entries of* Q *are non-zero, then exact recovery is solvable (and efficiently so) if and only if* $I_+(p, Q) \geq 1$. *In general, exact recovery is solvable at the threshold, i.e., when* $I_+(p, Q) = 1$, *if and only if any two columns of* $\mathrm{diag}(p)Q$ *have a component that is non-zero and different in both columns.*

Remark 16 *In the symmetric case* $\mathrm{SSBM}(n, k, a\ln(n)/n, b\ln(n)/n)$, *the CH-divergence is maximized at the value of* $t = 1/2$, *and it reduces in this case to the Hellinger divergence between any two columns of* Q; *the theorem's inequality becomes*

$$\frac{1}{k}(\sqrt{a}-\sqrt{b})^2 > 1,$$

matching the expression obtained in Theorem 13 for 2 symmetric communities.

We discuss now some properties of the functional D_+ governing the fundamental limit for exact recovery in Theorem 14. For $t \in [0, 1]$, let

$$D_t(\mu\|\nu) := \sum_x \nu(x) f_t(\mu(x)/\nu(x)), \quad f_t(y) = 1 - t + ty - y^t, \tag{22}$$

and note that $D_+ = \max_{t\in[0,1]} D_t$. Since the function f_t satisfies

- $f_t(1) = 0$
- f_t is convex on $\mathbb{R}_+$,

the functional D_t is what is called an f-divergence Csiszár (1963), like the KL-divergence ($f(y) = y\log y$), the Hellinger divergence, or the Chernoff divergence. Such functionals have a list of common properties described in Csiszár (1963). For example, if two distributions are perturbed by additive noise (i.e., convolving with a distribution), then the divergence always increases, or if some of the elements of the distributions' support are merged, then the divergence always decreases. Each of these properties can interpreted in terms of community detection (e.g., it is easier to recovery merged communities, etc.). Since D_t collapses to the Hellinger divergence when $t = 1/2$ and since it matches the Chernoff divergence for probability measures, we call D_t the Chernoff-Hellinger (CH) divergence in Abbe and Sandon (2015b), and so for D_+ as well by a slight abuse of terminology.

Theorem 14 gives hence an operational meaning to a new f-divergence, showing that the fundamental limit for data clustering in SBMs is governed by the CH-divergence, similarly to the fundamental limit for data transmission in DMCs governed by the KL-divergence. If the columns of $\mathrm{diag}(p)Q$ are "different" enough, where difference is measured in CH-divergence, then one can separate the communities. This is analog to the channel coding theorem that says that when the output's distributions are different enough, where difference is measured in KL-divergence, then one can separate the codewords.

3.2 Proof techniques

Let $(X,G) \sim \mathrm{SBM}(n,p,W)$. Recall that to solve exact recovery, we need to find the partition of the vertices, but not necessarily the actual labels. Equivalently, the goal is to find the community partition $\Omega = \Omega(X)$ as defined in Section 2. Upon observing $G = g$, reconstructing Ω with $\hat{\Omega}(g)$ gives a probability of error given by

$$P_e := \mathbb{P}\{\Omega \neq \hat{\Omega}(G)\} = \sum_g \mathbb{P}\{\hat{\Omega}(g) \neq \Omega | G = g\}\mathbb{P}\{G = g\} \tag{23}$$

and thus an estimator $\hat{\Omega}_{\mathrm{map}}(\cdot)$ minimizing the above must minimize $\mathbb{P}\{\hat{\Omega}(g) \neq \Omega | G = g\}$ for every g. To minimize $\mathbb{P}\{\hat{\Omega}(g) \neq \Omega | G = g\}$, we must declare a reconstruction of s that maximizes the posterior distribution

$$\mathbb{P}\{\Omega = s | G = g\}, \tag{24}$$

or equivalently

$$\sum_{x \in [k]^n : \Omega(x) = s} \mathbb{P}\{G = g | X = x\} \prod_{i=1}^k p_i^{|\Omega_i(x)|}, \tag{25}$$

and any such maximizer can be chosen arbitrarily.

This defines the MAP estimator $\hat{\Omega}_{\mathrm{map}}(\cdot)$, which minimizes the probability of making an error for exact recovery. If MAP fails in solving exact recovery, no other algorithm can succeed. Note that to succeed for exact recovery, the partition shall be typical in order to make the last factor in (25) non-vanishing (i.e., communities of relative size $p_i + o(1)$ for all $i \in [k]$). Of course, resolving exactly the maximization in (24) requires comparing exponentially many terms, so the MAP estimator may not always reveal the computational threshold for exact recovery.

3.2.1 Converse: the genie-aided approach

We now describe how to obtain the impossibility part of Theorem 14. Imagine that in addition to observing G, a genie provides the observation of $X_{\sim u} = \{X_v : v \in [n] \setminus \{u\}\}$. Define now $\hat{X}_v = X_v$ for $v \in [n] \setminus \{u\}$ and

$$\hat{X}_{u,\mathrm{map}}(g, x_{\sim u}) = \arg\max_{i \in [k]} \mathbb{P}\{X_u = i | G = g, X_{\sim u} = x_{\sim u}\}, \tag{26}$$

where ties can be broken arbitrarily if they occur (we assume that an error is declared in case of ties to simplify the analysis). If we fail at recovering a single component when all others are revealed, we must fail at solving exact recovery all at once, thus

$$\mathbb{P}\{\hat{\Omega}_{\mathrm{map}}(G) \neq \Omega\} \geq \mathbb{P}\{\exists u \in [n] : \hat{X}_{u,\mathrm{map}}(G, X_{\sim u}) \neq X_u\}. \tag{27}$$

This lower bound may appear to be loose at first, as recovering the entire communities from the graph G seems much more difficult than classifying each vertex by having all others revealed (we call the latter component-MAP). We however show that is tight in the regime

considered. In any case, studying when this lower bound is not vanishing always provides a necessary condition for exact recovery.

Let $E_u := \{\hat{X}_{u,\mathrm{map}}(G, X_{\sim u}) \neq X_u\}$. If the events E_u were independent, we could write $\mathbb{P}\{\cup_u E_u\} = 1 - \mathbb{P}\{\cap_u E_u^c\} = 1 - (1 - \mathbb{P}\{E_1\})^n \geq 1 - e^{-n\mathbb{P}\{E_1\}}$ and if $\mathbb{P}\{E_1\} = \omega(1/n)$, this would drive $\mathbb{P}\{\cup_u E_u\}$, and thus P_e, to 1. The events E_u are not independent, but their dependencies are weak enough such that previous reasoning still applies, and P_e is driven to 1 when $\mathbb{P}\{E_1\} = \omega(1/n)$.

Formally, one can handle the dependencies with different approaches. We describe here an approach via the second moment method. Recall the following basic inequality.

Lemma 17 *If Z is a random variable taking values in $\mathbb{Z}_+ = \{0, 1, 2, \dots\}$, then*

$$\mathbb{P}\{Z = 0\} \leq \frac{\mathbb{V}\mathrm{ar} Z}{(\mathbb{E}Z)^2}.$$

We apply this inequality to

$$Z = \sum_{u \in [n]} \mathbb{1}(\hat{X}_{u,\mathrm{map}}(G, X_{\sim u}) \neq X_u),$$

which counts the number of components where component-MAP fails. Note that the right hand side of (27) corresponds to $\mathbb{P}\{Z \geq 1\}$ as desired. Our goal is to show that $\frac{\mathbb{V}\mathrm{ar} Z}{(\mathbb{E}Z)^2}$ stays strictly below 1 in the limit, or equivalently, $\frac{\mathbb{E}Z^2}{(\mathbb{E}Z)^2}$ stays strictly below 2 in the limit. In fact, the latter tends to 1 in the converse of Theorem 14.

Note that $Z = \sum_{u \in [n]} Z_u$ where $Z_u := \mathbb{1}(\hat{X}_{u,\mathrm{map}}(G, X_{\sim u}) \neq X_u)$ are binary random variables with $\mathbb{E}Z_u = \mathbb{E}Z_v$ for all u, v. Hence,[6]

$$\mathbb{E}Z = n\mathbb{P}\{Z_1 = 1\} \tag{28}$$

$$\mathbb{E}Z^2 = \sum_{u,v \in [n]} \mathbb{E}(Z_u Z_v) = \sum_{u,v \in [n]} \mathbb{P}\{Z_u = Z_v = 1\} \tag{29}$$

$$= n\mathbb{P}\{Z_1 = 1\} + n(n-1)\mathbb{P}\{Z_1 = 1\}\mathbb{P}\{Z_2 = 1 | Z_1 = 1\} \tag{30}$$

and $\frac{\mathbb{E}Z^2}{(\mathbb{E}Z)^2}$ tends to 1 if

$$\frac{n\mathbb{P}\{Z_1 = 1\} + n(n-1)\mathbb{P}\{Z_1 = 1\}\mathbb{P}\{Z_2 = 1 | Z_1 = 1\}}{n^2\mathbb{P}\{Z_1 = 1\}^2} = 1 + o(1) \tag{31}$$

or

$$\frac{1}{n\mathbb{P}\{Z_1 = 1\}} + \frac{\mathbb{P}\{Z_2 = 1 | Z_1 = 1\}}{\mathbb{P}\{Z_1 = 1\}} = 1 + o(1). \tag{32}$$

This takes place if $n\mathbb{P}\{Z_1 = 1\}$ diverges and

$$\frac{\mathbb{P}\{Z_2 = 1 | Z_1 = 1\}}{\mathbb{P}\{Z_2 = 1\}} = 1 + o(1), \tag{33}$$

6. One should normally distinguish whether vertices 1 and 2 are in the same community or not, but this has no effect on the final result.

i.e., if E_1, E_2 are asymptotically independent.

The asymptotic independence takes place due to the regime that we consider for the block model in the theorem. To given a related example, in the context of the Erdős-Rényi model $ER(n,p)$, if W_1 is 1 when vertex u is isolated and 0 otherwise, then $\mathbb{P}\{W_1 = 1|W_2 = 1\} = (1-p)^{n-2}$ and $\mathbb{P}\{W_1 = 1\} = (1-p)^{n-1}$, and thus $\frac{\mathbb{P}\{W_1=1|W_2=1\}}{\mathbb{P}\{W_1=1\}} = (1-p)^{-1}$ tends to 1 as long as p tends to 0. That is, the property of a vetex being isolated is asymptotically independent as long as the edge probability is vanishing. A similar outcomes takes place for the property of MAP-component failing when edge probabilities are vanishing in the block model.

The location of the threshold is then dictated by requirement that $n\mathbb{P}\{Z_1 = 1\}$ diverges, and this where the CH-divergence threshold emerges from a moderate deviation analysis. We next summarize what we obtained with the above reasoning, and then specialized to the regime of Theorem 14.

Theorem 18 *Let $(X,G) \sim \mathrm{SBM}(n,p,W)$ and $Z_u := \mathbb{1}(\hat{X}_{u,\mathrm{map}}(G, X_{\sim u}) \neq X_u)$, $u \in [n]$. If p, W are such that E_1 and E_2 are asymptotically independent, then exact recovery is not solvable if*

$$\mathbb{P}\{\hat{X}_{u,\mathrm{map}}(G, X_{\sim u}) \neq X_u\} = \omega(1/n). \tag{34}$$

The next lemma gives the behavior of $\mathbb{P}\{Z_1 = 1\}$ in the logarithmic degree regime.

Lemma 19 *Abbe and Sandon (2015b) Consider the hypothesis test where $H = i$ has prior probability p_i for $i \in [k]$, and where observable Y is distributed $\mathrm{Bin}(np, W_i)$ under hypothesis $H = i$. Then the probability of error $P_e(p,W)$ of MAP decoding for this test satisfies $\frac{1}{k-1}\mathrm{Over}(n,p,W) \leq P_e(p,W) \leq \mathrm{Over}(n,p,W)$ where*

$$\mathrm{Over}(n,p,W) = \sum_{i<j} \sum_{z \in \mathbb{Z}_+^k} \min(\mathbb{P}\{\mathrm{Bin}(np,W_i) = z\}p_i, \mathbb{P}\{\mathrm{Bin}(np,W_j) = z\}p_j),$$

and for a symmetric $Q \in \mathbb{R}_+^{k\times k}$,

$$\mathrm{Over}(n,p,\log(n)Q/n) = n^{-I_+(p,Q)-O(\log\log(n)/\log n)}, \tag{35}$$

where $I_+(p,Q) = \min_{i<j} D_+((\mathrm{diag}(p)Q)_i, (\mathrm{diag}(p)Q)_j)$.

Corollary 20 *Let $(X,G) \sim \mathrm{SBM}(n,p,W)$ where p is constant and $W = Q\frac{\ln n}{n}$. Then*

$$\mathbb{P}\{\hat{X}_{u,\mathrm{map}}(G, X_{\sim u}) \neq X_u\} = n^{-I_+(p,Q)+o(1)}. \tag{36}$$

A robust extension of this Lemma is proved in Abbe and Sandon (2015b) that allows for a slight perturbation of the binomial distributions. We next explain why E_1 and E_2 are asymptotically independent.

Recall that $Z_1 = \mathbb{1}(\hat{X}_{1,\mathrm{map}}(G, X_{\sim 1}) \neq X_1)$, and E_1 is the event that $Z_1 = 1$, i.e., that G and X take values g and x such that[7]

$$\mathrm{argmax}_{i\in[k]} \mathbb{P}\{X_1 = i | G = g, X_{\sim 1} = x_{\sim 1}\} \neq x_1. \tag{37}$$

7. Formally, argmax is a set, and we are asking that this set is not the singleton $\{x_1\}$. It could be that this set is not that singleton but contains x_1, in which case breaking ties may still make component-MAP succeed by luck; however this gives a probability of error of at least 1/2, and thus already fails exact recovery. This is why we declare an error in case of ties.

Let $x_{\sim 1} \in [k]^{n-1}$ and $\omega(x_{\sim 1}) :_i = |\Omega_i(x_{\sim 1})|$. We have

$$\begin{aligned}
&\mathbb{P}\{X_1 = x_1 | G = g, X_{\sim 1} = x_{\sim 1}\} && (38)\\
&\propto \mathbb{P}\{G = g | X_{\sim 1} = x_{\sim 1}, X_1 = x_1\} \cdot \mathbb{P}\{X_{\sim 1} = x_{\sim 1}, X_1 = x_1\} && (39)\\
&\propto \mathbb{P}\{G = g | X_{\sim 1} = x_{\sim 1}, X_1 = x_1\}\mathbb{P}\{X_1 = x_1\} && (40)\\
&= p(x_1) \prod_{1\leq i<j\leq k} W_{i,j}^{N_{ij}(x_{\sim 1},g)}(1-W_{i,j})^{N_{ij}^c(x_{\sim 1},g)} && (41)\\
&\quad \cdot \prod_{1\leq i\leq k} W_{i,x_1}^{N_i^{(1)}(x_{\sim 1},g)}(1-W_{i,x_1})^{\omega_i(x_{\sim 1})-N_i^{(1)}(x_{\sim 1},g)} && (42)\\
&\propto p(x_1) \prod_{1\leq i\leq k} W_{i,x_1}^{N_i^{(1)}(x_{\sim 1},g)}(1-W_{i,x_1})^{\omega_i(x_{\sim 1})-N_i^{(1)}(x_{\sim 1},g)} && (43)
\end{aligned}$$

where $N_i^{(1)}(x_{\sim 1}, g)$ is the number of neighbors that vertex 1 has in community i. We denote by $N^{(1)}$ the random vector valued in $\mathbb{Z}_+^k$ whose i-th component is $N_i^{(1)}(X_{\sim 1}, G)$, and call $N^{(1)}$ the *degree profile* of vertex 1. As just shown, $(N^{(1)}, |\Omega(X_{\sim 1})|)$ is a sufficient statistics for component-MAP. We thus have to resolve an hypothesis test with k hypotheses, where $|\Omega(X_{\sim 1})|$ contains the sizes of the k communities with $(n-1)$ vertices (irrespective of the hypothesis), and under hypothesis $X_1 = x_1$ which has prior p_{x_1}, the observable $N^{(1)}$ has distribution proportional to (43), i.e., i.i.d. components that are Binomial$(\omega_i(x_{\sim 1}), W_{i,x_1})$.

Consider now the case of two symmetric (assortative) communities to simplify the discussion. By symmetry, it is sufficient to show that

$$\frac{\mathbb{P}\{E_1 | E_2, X_1 = 1\}}{\mathbb{P}\{E_1 | X_1 = 1\}} = 1 + o(1). \tag{44}$$

Further, $|\Omega_1(X_{\sim 1})| \sim \mathrm{Bin}(n-1, 1/2)$ is a sufficient statistics for the number of vertices in each community. By standard concentration, $|\Omega_1(X_{\sim 1})| \in [n/2 - \sqrt{n}\log n, n/2 + \sqrt{n}\log n]$ with probability $1 - O(n^{-\frac{1}{2}\log n})$. Instead, from Lemma 19, we have that $\mathbb{P}\{Z_1 = 1 | |\Omega(X_{\sim 1})| \in [n/2 - \sqrt{n}\log n, n/2 + \sqrt{n}\log n]\}$ decays only polynomially to 0, thus it is sufficient to show that

$$\frac{\mathbb{P}\{E_1 | E_2, X_1 = 1, |\Omega(X_{\sim 1})| \in [n/2 - \sqrt{n}\log n, n/2 + \sqrt{n}\log n]\}}{\mathbb{P}\{E_1 | X_1 = 1, |\Omega(X_{\sim 1})| \in [n/2 - \sqrt{n}\log n, n/2 + \sqrt{n}\log n]\}} = 1 - o(1). \tag{45}$$

Recall that the error event E_1 depends only on $(N^{(1)}, |\Omega(X_{\sim 1})|)$, and $N^{(1)}$ contains two components, $N_1^{(1)}, N_2^{(1)}$, which are the number of edges that vertex 1 has in each of the two communities. It remains to show that the effect of knowing E_2 does not affect the numerator by much. This intuitively follows from the fact that for communities of constrained size, this conditioning gives only information about edge $(1,2)$ in the graph (by Markovian property of the model), which creates negligible dependencies. Showing this with a formal argument requires further technical expansions.

3.2.2 Achievability: graph-splitting and two-round algorithms

Two-rounds algorithms have proved to be powerful in the context of exact recovery. The general idea consists in using a first algorithm to obtain a good but not necessarily exact

clustering, solving a joint assignment of all vertices, and then to switch to a local algorithm that "cleans up" the good clustering into an exact one by reclassifying each vertex. This approach has a few advantages:

- If the clustering of the first round is accurate enough, the second round becomes approximately the genie-aided hypothesis test discussed in previous section, and the approach is built in to achieve the threshold;
- if the clustering of the first round is efficient, then the overall method is efficient since the second round only performs computations for each single node separately and has thus linear complexity.

Some difficulties need to be overome for this program to be carried out:

- One needs to obtain a good clustering in the first round, which is typically non-trivial;
- One needs to be able to analyze the probability of success of the second round, as the graph is no longer independent of the obtained clusters.

To resolve the latter point, we rely in Abbe et al. (2016) a technique which we call "graph-splitting" and which takes again advantage of the sparsity of the graph.

Definition 21 (Graph-splitting) *Let g be an n-vertex graph and $\gamma \in [0,1]$. The graph-splitting of g with split-probability γ produces two random graphs G_1, G_2 on the same vertex set as G. The graph G_1 is obtained by sampling each edge of g independently with probability γ, and $G_2 = g \setminus G_1$ (i.e., G_2 contains the edges from g that have not been subsampled in G_1).*

Graph splitting is convenient in part due to the following fact.

Lemma 22 *Let $(X,G) \sim \mathrm{SBM}(n,p,\log nQ/n)$, (G_1,G_2) be a graph splitting of G with parameters γ and $(X,\tilde{G}_2) \sim \mathrm{SBM}(n,p,(1-\gamma)\log nQ/n)$ with $\tilde{G}_2$ independent of G_1. Let $\hat{X} = \hat{X}(G_1)$ be valued in $[k]^n$ such that $\mathbb{P}\{A(X,\hat{X}) \geq 1-o(n)\} = 1-o(1)$. For any $v \in [n]$, $d \in \mathbb{Z}_+^k$,*

$$\mathbb{P}\{D_v(\hat{X},G_2) = d\} \leq (1+o(1))\mathbb{P}\{D_v(\hat{X},\tilde{G}_2) = d\} + n^{-\omega(1)}, \tag{46}$$

where $D_v(\hat{X},G_2)$ is the degree profile of vertex v, i.e., the k-dimensional vector counting the number of neighbors of vertex v in each community using the clustered graph $(\hat{X},G_2)$.

The meaning of this lemma is as follows. We can consider G_1 and G_2 to be approximately independent, and export the output of an algorithm run on G_1 to the graph G_2 without worrying about dependencies to proceed with component-MAP. Further, if γ is to chosen as $\gamma = \tau(n)/\log(n)$ where $\tau(n) = o(\log(n))$, then G_1 is distributed as $\mathrm{SBM}(n,p,\tau(n)Q/n)$ and G_2 remains approximately as $\mathrm{SBM}(n,p,\log nQ/n)$. This means that from our original SBM graph, we produce essentially 'for free' a preliminary graph G_1 with $\tau(n)$ expected degrees that can be used to get a preliminary clustering, and we can then improve that clustering on the graph G_2 which has still logarithmic expected degree.

Our goal is to obtain on G_1 a clustering that is almost exact, i.e., with only a vanishing fraction of misclassified vertices. If this can be achieved for some $\tau(n) = o(\log(n))$, then a

robust version of the genie-aided hypothesis test described in Section 3.2.1 can be run to re-classify each node successfully when $I_+(p,Q) > 1$. Luckily, as we shall see in Section 5, almost exact recovery can be solved with the mere requirement that $\tau(n) = \omega(1)$ (i.e., $\tau(n)$ diverges). In particular, setting $\tau(n) = \log\log(n)$ does the job. We next describe more formally the previous reasoning.

Theorem 23 *Assume that almost exact recovery is solvable in $\mathrm{SBM}(n,p,\omega(1)Q/n)$. Then exact recovery is solvable in $\mathrm{SBM}(n,p,\log nQ/n)$ if*

$$I_+(p,Q) > 1. \tag{47}$$

To see this, let $(X,G) \sim \mathrm{SBM}(n,p,\tau(n)Q/n)$, and (G_1,G_2) be a graph splitting of G with parameters $\gamma = \log\log n/\log n$. Let $(X,\tilde{G}_2) \sim \mathrm{SBM}(n,p,(1-\gamma)\tau(n)Q/n)$ with $\tilde{G}_2$ independent of G_1 (note that the same X appears twice). Let $\hat{X} = \hat{X}(G_1)$ be valued in $[k]^n$ such that $\mathbb{P}\{A(X,\hat{X}) \geq 1-o(1)\} = 1-o(1)$; note that such an $\hat{X}$ exists from the Theorem's hypothesis. Since $A(X,\hat{X}) = 1-o(1)$ with high probability, $(G_2,\hat{X})$ are functions of G and using a union bound, we have

$$\begin{aligned}
\mathbb{P}\{\hat{\Omega}_{\mathrm{map}}(G) \neq \Omega\} &\leq \mathbb{P}\{\hat{\Omega}_{\mathrm{map}}(G) \neq \Omega | A(X,\hat{X}) = 1-o(1)\} + o(1) &(48)\\
&\leq \mathbb{P}\{\hat{\Omega}_{\mathrm{map}}(G_2,\hat{X}) \neq \Omega | A(X,\hat{X}) = 1-o(1)\} + o(1) &(49)\\
&\leq n\mathbb{P}\{X_{1,\mathrm{map}}(G_2,\hat{X}_{\sim 1}) \neq X_1 | A(X,\hat{X}) = 1-o(1)\} + o(1). &(50)
\end{aligned}$$

We next replace G_2 by $\tilde{G}_2$. Note that $\tilde{G}_2$ has already the same marginal as G_2, the only issue is that G_2 is not independent from G_1 since the two graphs are disjoint, and since $\hat{X}$ is derived from G_2, some dependencies are carried along with G_1. However, $\tilde{G}_2$ and G_2 are 'essentially independent' as stated in Lemma 22, because the probability that $\tilde{G}_2$ samples an edge that is already present in G_1 is $O(\log^2 n/n^2)$, and the expected degrees in each graph is $O(\log n)$. This takes us to

$$\mathbb{P}\{\hat{\Omega}_{\mathrm{map}}(G) \neq \Omega\} \leq n\mathbb{P}\{X_{1,\mathrm{map}}(\tilde{G}_2,\hat{X}_{\sim 1}) \neq X_1 | A(X,\hat{X}) = 1-o(1)\}(1+o(1)) + o(1). \tag{51}$$

We can now replace $\hat{X}_{\sim 1}$ with $X_{\sim 1}$ to the expense that we may blow up this the probability by a factor $n^{o(1)}$ since $A(X,\hat{X}) = 1-o(1)$, using again the fact that expected degrees are logarithmic. Thus we have

$$\mathbb{P}\{\hat{\Omega}_{\mathrm{map}}(G) \neq \Omega\} \leq n^{1+o(1)}\mathbb{P}\{X_{1,\mathrm{map}}(\tilde{G}_2,X_{\sim 1}) \neq X_1 | A(X,\hat{X}) = 1-o(1)\} + o(1) \tag{52}$$

and the conditioning on $A(X,\hat{X}) = 1-o(1)$ can now be removed due to independence, so that

$$\mathbb{P}\{\hat{\Omega}_{\mathrm{map}}(G) \neq \Omega\} \leq n^{1+o(1)}\mathbb{P}\{X_{1,\mathrm{map}}(\tilde{G}_2,X_{\sim 1}) \neq X_1\} + o(1). \tag{53}$$

The last step consists in closing the loop and replacing $\tilde{G}_2$ by G, since $1-\gamma = 1-o(1)$, which uses the same type of argument as for the replacement of G_2 by $\tilde{G}_2$, with a blow up that is at most $n^{o(1)}$. As a result,

$$\mathbb{P}\{\hat{\Omega}_{\mathrm{map}}(G) \neq \Omega\} \leq n^{1+o(1)}\mathbb{P}\{X_{1,\mathrm{map}}(G,X_{\sim 1}) \neq X_1\} + o(1), \tag{54}$$

and if

$$\mathbb{P}\{X_{1,\text{map}}(G, X_{\sim 1}) \neq X_1\} = n^{-1-\varepsilon} \tag{55}$$

for $\varepsilon > 0$, then $\mathbb{P}\{\hat{\Omega}_{\text{map}}(G) \neq \Omega\}$ is vanishing as stated in the theorem.

Therefore, in view of Theorem 23, the achievability part of Theorem 14 reduces to the following result.

Theorem 24 *Abbe and Sandon (2015b) Almost exact recovery is solvable in* $\text{SBM}(n, p, \omega(1)Q/n)$, *and efficiently so.*

This follows from Theorem 50 from Abbe and Sandon (2015b) using the Sphere-comparison algorithm discussed in Section 5. Note that to prove that almost exact recovery is solvable in this regime without worrying about efficiency, the Typicality Sampling Algorithm discussed in Section 4.6.1 is already sufficient.

In conclusion, in the regime of Theorem 14, exact recovery follows from solving almost exact recovery on an SBM with degrees that grow sub-logarithmically, using graph-splitting and a clean-up round. The behavior of the component-MAP error (i.e., the probability of misclassifying a single node when others have been revealed) pings down the behavior of the threshold: if this probability is $\omega(1/n)$, exact recovery is not possible, and if it is $o(1/n)$, exact recovery is possible. Decoding for the latter is then resolved by obtaining the exponent of the component-MAP error, which brings the CH-divergence in.

3.3 Local to global amplification

Previous two sections give a lower bound and an upper bound on the probability that MAP fails at recovering the entire clusters, in terms of the probability that MAP fails at recovering a single vertex when others are revealed. Denoting by P_{global} and P_{local} these two probability of errors, we essentially[8] have

$$1 - \frac{1}{nP_{\text{local}}} + o(1) \leq P_{\text{global}} \leq nP_{\text{local}} + o(1). \tag{56}$$

This implies that P_{global} has a threshold phenomena as P_{local} varies:

$$P_{\text{global}} \to \begin{cases} 0 & \text{if } P_{\text{local}} \ll 1/n, \\ 1 & \text{if } P_{\text{local}} \gg 1/n. \end{cases} \tag{57}$$

Moreover, deriving this relies mainly on the regime of the model, rather than the specific structure of the SBM. In particular, it mainly relies on the exchangeability of the model (i.e., vertex labels have no relevance) and the fact that the vertex degrees do not grow rapidly. This suggests that this 'local to global' phenomenon takes place in a more general class of models. The expression of the threshold for exact recovery in $\text{SBM}(n, p, \log nQ/n)$ as a function of the parameters p, Q is instead specific to the model, and relies on the CH-divergence in the case of the SBM, but the moderate deviation analysis of P_{local} for other models may reveal a different functional or f-divergence.

8. The upper bound discussed in Section 3.2.2 gives $n^{1+o(1)}P_{\text{local}} + o(1)$, but the analysis can be tighten to yield a factor n instead of $n^{1+o(1)}$.

The local to global approach has also an important implication at the computational level. The achievability proof described in previous section gives directly an algorithm: use graph-splitting to produce two graphs; solve almost exact recovery on the first graph and improve locally the latter with the second graph. Since the second round is by construction efficient (it corresponds to n parallel local computations), it is sufficient to solve almost exact recovery efficiently (in the regime of diverging degrees) to obtain for free an efficient algorithm for exact recovery down to the threshold. This thus gives a computational reduction. In fact, the process can be iterated to further reduce almost exact recovery to a weaker recovery requirements, until a 'bottle-neck' problem is attained.

3.4 Semidefinite programming and spectral methods

The two-round procedure discussed in Section 3.2.2 has the advantage to only require almost exact recovery to be efficiently solved. As it can only be easier to solve almost exact rather than exact recovery, this approach may be beneficial compared to solving efficiently exact recovery in 'one shot.' The approach can also be handy in real applications, improving the communities with a clean-up phase. Nonetheless, it is also possible to achieve exact recovery in 'one shot' without relying on two-rounds. We provide here some examples of methods.

Semi-definite programming (SDP). We present here the SDP developed in Abbe et al. (2016) for the symmetric SBM with two communities and balanced clusters. SDPs were also used in various works on the SBM such as B. Hajek (2014); Guédon and Vershynin (2016); Amini and Levina (2014); Bandeira (2015); Montanari and Sen (2016); Javanmard et al. (2016). The idea is to approximate MAP decoding. Assume for the purpose of this section that we work with the symmetric SBM with two balanced clusters that are drawn uniformly at random. In this case, MAP decoding looks for a balanced partition of the vertices into two clusters such that the number of crossing edges is minimized (when the connection probability inside clusters A is less than the connection probability across clusters B, otherwise maximized). This is seen by writing the a posteriori distribution as

$$\mathbb{P}\{X = x | G = g\} \propto \mathbb{P}\{G = g | X = x\} \cdot \mathbb{1}(x \text{ is balanced}), \tag{58}$$

$$\propto A^{N_{in}}(1-A)^{\frac{n^2}{4}-N_{in}} B^{N_{out}}(1-B)^{\frac{n^2}{4}-N_{out}} \cdot \mathbb{1}(x \text{ is balanced}) \tag{59}$$

$$\propto \left(\frac{B(1-A)}{A(1-B)}\right)^{N_{out}} \cdot \mathbb{1}(x \text{ is balanced}) \tag{60}$$

where N_{in} is the number of edges that G has inside the clusters defined by x, and N_{out} is the number of crossing edges. If $A > B$, then $\frac{B(1-A)}{A(1-B)} < 1$ and MAP looks for a balanced partition that has the least number of crossing edges, i.e., a min-bisection. In the worst-case model, min-bisection is NP-hard, and approximations leave a polylogarithmic integrality gap Krauthgamer and Feige (2006). However, Theorem 14 tells us that it is still be possible to recover the min-bisection efficiently for the typical instances of the SBM, without any gap to the information-theoretic threshold.

We express now the min-bisection as a quadratic optimization problem, using $\{+1, -1\}$ variables to label the two communities. More precisely, define

$$\hat{x}_{\text{map}}(g) = \text{argmax}_{\substack{x \in \{+1,-1\}^n \\ x^t 1^n = 0}} x^t A(g) x \tag{61}$$

where $A(g)$ is the adjacency matrix of the graph g, i.e., $A(g)_{ij} = 1$ if there is an edge between vertices i and j, and $A(g)_{ij} = 0$ otherwise. The above maximizes the number of edges inside the clusters, minus the number of edges across the clusters; since the total number of edges is invariant from the clustering, this is equivalent to min-bisection.

Solving (61) is hard because of the integer constraint $x \in \{+1, -1\}^n$. A first possible relaxation is to replace this constraint with an Euclidean constraint on real vectors, turning (61) into an eigenvector problem, which is the idea behind spectral methods discussed next. The idea of SDPs is instead to lift the variables to change the quadratic optimization into a linear optimization (as for max-cut Goemans and Williamson (1995)), albeit with additional constraints. Namely, since $\mathrm{tr}(AB) = \mathrm{tr}(BA)$ for any matrices of matching dimensions, we have

$$x^t A(g) x = \mathrm{tr}(x^t A(g) x) = \mathrm{tr}(A(g) x x^t), \tag{62}$$

hence defining $X := xx^t$, we can write (61) as

$$\hat{X}_{\mathrm{map}}(g) = \mathrm{argmax}_{\substack{X \succeq 0 \\ X_{ii}=1, \forall i \in [n] \\ \mathrm{rank} X = 1 \\ X 1^n = 0}} \mathrm{tr}(A(g) X). \tag{63}$$

Note that the first three constraints on X force X to take the form xx^t for a vector $x \in \{+1, -1\}^n$, as desired, and the last constraint gives the balance requirement. The advantage of (63) is that the objective function is now linear in the lifted variable X. The constraint $\mathrm{rank} X = 1$ is responsible now for keeping the optimization hard. We hence simply remove that constraint to obtain our SDP relaxation:

$$\hat{X}_{sdp}(g) = \mathrm{argmax}_{\substack{X \succeq 0 \\ X_{ii}=1, \forall i \in [n] \\ X 1^n = 0}} \mathrm{tr}(A(g) X). \tag{64}$$

A possible approach to handle the constraint $X1^n = 0$ is to replace the adjacency matrix $A(g)$ by the matrix $B(g)$ such that $B(g)_{ij} = 1$ if there is an edge between vertices i and j, and $B(g)_{ij} = -1$ otherwise. Using $-T$ for a large T instead of -1 for non-edges would force the clusters to be balanced, and it turns out that -1 is already sufficient for our purpose. This gives another SDP:

$$\hat{X}_{SDP}(g) = \mathrm{argmax}_{\substack{X \succeq 0 \\ X_{ii}=1, \forall i \in [n]}} \mathrm{tr}(B(g) X). \tag{65}$$

The dual of this SDP is given by

$$\min_{\substack{Y_{ij}=0 \forall 1 \le i \ne j \le n \\ Y \succeq B(g)}} \mathrm{tr}(Y). \tag{66}$$

Since the dual minimization gives an upper-bound on the primal maximization, a solution is optimal if it makes the dual minima match the primal maxima. The Ansatz here consists in taking $Y = 2(D_{in} - D_{out}) + I_n$ as a candidate for the diagonal matrix Y, which gives the primal maxima. It we thus have $Y \succeq B(g)$, this is a feasible solution for the dual, and we obtain a dual certificate. The following is shown in Abbe et al. (2016) based on this reasoning.

Definition 25 *Define the SBM Laplacian for G drawn under the symmetric SBM with two communities by*

$$L_{\mathrm{SBM}} = D(G_{in}) - D(G_{out}) - A(G), \tag{67}$$

where $D(G_{in})$ ($D(G_{out})$) are the degree matrices of the subgraphs of G containing only the edges inside (respectively across) the clusters, and $A(G)$ is the adjacency matrix of G.

Theorem 26 *The SDP solves exact recovery in the symmetric SBM with 2 communities if $2L_{\mathrm{SBM}} + 11^t + I_n \succeq 0$.*

This condition is satisfies with high probability all the way down to the exact recovery threshold. In Abbe et al. (2016), it is showsn that this condition holds in a regime that does not exactly match the threshold, off roughly by a factor of 2 for large degrees. This gap is closed in B. Hajek (2014); Bandeira (2015), which show that SDPs achieve the exact recovery threshold in the symmetric case. Some results for unbalanced communities were also obtained in Perry and Wein (2015), although it is still open to achieve the general CH threshold with SDPs. Many other works have studied SDPs for the stochastic block model, we refer to Amini and Levina (2014); Abbe et al. (2016); Bandeira (2015); B. Hajek (2014); Montanari and Sen (2016); Perry and Wein (2015) for further references. In particular, Montanari and Sen (2016) shows that SDPs allow to approach the threshold for weak recovery in the two-community SSBM arbitrarily close when the expected degrees diverge.

Spectral methods. Consider again the symmetric SBM with 2 balanced communities. Recall that MAP maximizes

$$\max_{\substack{x \in \{+1,-1\}^n \\ x^t 1^n = 0}} x^t A(g) x. \tag{68}$$

The general idea behind spectral methods is to relax the integral constraint to an Euclidean constraint on real valued vectors. This lead to looking for a maximizer of

$$\max_{\substack{x \in \mathbb{R}^n : \|x\|_2^2 = n \\ x^t 1^n = 0}} x^t A(g) x. \tag{69}$$

Without the constraint $x^t 1^n = 0$, the above maximization gives precisely the eigenvector corresponding to the largest eigenvalue of $A(g)$. Note that $A(g)1^n$ is the vector containing the degrees of each node in g, and when g is an instance of the symmetric SBM, this concentrates to the same value for each vertex, and 1^n is close to an eigenvector of $A(g)$. Since $A(g)$ is real and symmetric, this suggests that the constraint $x^t 1^n = 0$ leads the maximization (69) to focus on the eigenspace orthogonal to the first eigenvector, and thus to the eigenvector corresponding to the second largest eigenvalue. Thus one can take the second largest eigenvector and round it (assigning positive and negative components to different communities) to obtain an efficient algorithm.

Equivalently, one can write the MAP estimator as a maximizer of

$$\max_{\substack{x \in \{+1,-1\}^n \\ x^t 1^n = 0}} \sum_{1 \le i < j \le n} A_{ij}(g)(x_i - x_j)^2 \tag{70}$$

since the above minimizes the size of the cut between two balanced clusters. From simple algebraic manipulations, this is equivalent to looking for maximizers of

$$\max_{\substack{x\in\{+1,-1\}^n \\ x^t 1^n=0}} x^t L(g) x, \tag{71}$$

where $L(g)$ is the classical Laplacian of the graph, i.e.,

$$L(g) = D(g) - A(g), \tag{72}$$

and $D(g)$ is the degree matrix of the graph. With this approach 1^n is precisely an eigenvector of $L(g)$ with eigenvalue 0, and the relaxation to a real valued vector leads directly to the second eigenvector of $L(g)$, which can be rounded (positive or negative) to determine the communities.

The challenge with such 'basic' spectral methods is that, as the graph becomes sparser, the fluctuations in the node degrees become more important, and this can disrupt the second largest eigenvector from concentrating on the communities (it may concentrate instead on large degree nodes). To analyze this, one may express the adjacency matrix as a perturbation of its expected value, i.e.,

$$A(G) = \mathbb{E}A(G) + (A(G) - \mathbb{E}A(G)). \tag{73}$$

When indexing the first $n/2$ rows and columns to be in the same community, the expected adjacency matrix takes the following block structure

$$\mathbb{E}A(G) = \begin{pmatrix} A^{n/2\times n/2} & B^{n/2\times n/2} \\ B^{n/2\times n/2} & A^{n/2\times n/2} \end{pmatrix}, \tag{74}$$

where $A^{n/2\times n/2}$ is the $n/2\times n/2$ matrix with all entries equal to A. As expected, $\mathbb{E}A(G)$ has two eigenvalues, the expected degree $(A+B)/2$ with the constant eigenvector, and $(A-B)/2$ with the eigenvector taking the same constant with opposite signs on each community. The spectral methods described above succeeds in recovering the true communities if the noise $Z = A(G) - \mathbb{E}A(G)$ does not disrupt the first two eigenvectors from keeping their rank. See also Section 2.6. Theorems of random matrix theory allow to analyze this type of perturbations (see Vu (2014); Bordenave et al. (2015)), most commonly when the noise is independent rather than for the specific noise occurring here, but a direct application does typically not suffice to achieve the exact recovery threshold.

For exact recovery, one can use preprocessing steps to still succeed down to the threshold using the adjacency or Laplacian matrices, in particular by trimming the high degree nodes to regularize the graph spectra Feige and Ofek (2005). We refer to the papers of Vu Vu (2014) and in particular Proutiere et al. Yun and Proutiere (2014) for spectral methods achieving the exact recovery threshold. For the weak recovery problem discussed in next section, such tricks do not suffice to achieve the threshold, and one has to rely on other types of spectral operators as discussed in Section 4 with nonbacktracking operators.[9]

9. Similarly for SDPs which likely do not achieve the weak recovery threshold Moitra et al. (2016); Montanari and Sen (2016).

Note also that for k clusters, the expected adjacency matrix has rank k, and one typically has to take the k largest eigenvectors (corresponding the k largest eigenvalues), form n vectors of dimensional k by stacking each component of the k largest vectors into a vector (typically rescaled by $\sqrt{n}$), and run k-means clustering to generalize the 'rounding' step and produce k clusters.

We refer to Ng et al. (2001); Spielman and Teng (2007); Kannan et al. (2000); von Luxburg (2007) for further details on Laplacian spectral methods and k-means, and Vu (2014); Yun and Proutiere (2014); Yun and Proutiere (2015) for applications to exact recovery in the SBM.

3.5 Extensions

In this section, we demonstrate how the tools developed in previous section allow for fairly straightforward generalizations to other type of models.

3.5.1 Edge-labels, overlaps, bi-clustering

Labelled edges. Consider the labelled stochastic block model, where edges have labels attached to them, such as to model intensities of similarity. We assume that the labels belong to $\mathcal{Y} = \mathcal{Y}_+ \cup \{0\}$, where $\mathcal{Y}_+$ is a measurable set of labels (e.g., $(0,1]$), and 0 represents the special symbol corresponding to no edge. As for $\mathrm{SBM}(n,p,W)$, we define $\mathrm{LSBM}(n,p,\mu)$ where each vertex label X_u is drawn i.i.d. under p on $[k]$, $\mu(\cdot|x,x')$ is a probability measure on $\mathcal{Y}$ for all $x,x' \in [k]$, such that for $u<v$ in $[n]$, S a measurable set of $\mathcal{Y}$,

$$\mathbb{P}\{E_{uv}(G) \in S | X_u = x_u, X_v = x_v\} = \mu(S|x_u,x_v). \tag{75}$$

As for the unlabelled case, the symbol 0 will typically have probability $1-o(1)$ for any community pair, i.e., $\mu(\cdot|x,x')$ has an atom at 0 for all $x,x' \in [k]$, while μ_+, the measure restricted to $\mathcal{Y}_+$, may be arbitrary but of measure $o(1)$.

We now explain how the genie-aided converse and the graph-splitting techniques allow to obtain the fundamental limit for exact recovery in this model, without much additional effort. Consider first the case where $\mathcal{Y}_+$ is finite, and let L be the cardinality of $\mathcal{Y}_+$, and $\mu(0|x,x') = 1 - c_{x,x'}\log n/n$ for some constant $c_{x,x'}$ for all $x,x' \in [k]$. Hence $\mu(\mathcal{Y}_+|x,x') = c_{x,x'}\log n/n$.

For the achievability part, use a graph-splitting technique of, say, $\gamma = \log\log n/\log n$. On the first graph, merge the non-zero labels to a special symbol 1, i.e., collapse the model to $\mathrm{SBM}(n,p,W^\star)$ where $W^\star_{x,x'} = \sum_{y\in\mathcal{Y}_+}\mu(y|x,x')$ by assigning all non-zero labels to 1. Using our result on almost exact recovery (see Theorem 50), we can still solve almost exact recovery for this model as the expect degrees are diverging. Now, use the obtained clustering on the second graph to locally improve it. We have a seemingly different genie-aided hypothesis test than for the classical SBM, as we have k communities but also L labels on the edges. However, since the genie-aided test reveals all other community labels than the current vertex being classified, we can simply view the different labels on the edges as sub-communities, with a total of kL virtual communities. The distribution for each hypothesis is still essentially a multivariate Binomial, where hypothesis $i \in [k]$ has $\mathrm{Bin}(np_j, W(\ell|i,j))$ neighbors in augmented community $(j,\ell) \in [k]\times[L]$. Denoting by W_ℓ the matrix whose (i,j)-entry is $W(\ell|i,j)$, we thus have from the local to global results of

Section 3.3 that the threshold is given by

$$\min_{\substack{i,i'\in[k],l,l'\in[L]\\ i\neq i',l\neq l'}} D((PW_l)_i,(PW_{l'})_{i'}). \tag{76}$$

Further, this is efficiently achievable since almost exact recovery can be solved with the algorithm discussed in the classical setting, and since the new hypothesis test remains linear in n for finite k and L.

For non finite labels, the achievability part can be treated similarly using a quantization of the labels. This gives a continuous extension of the CH-divergence, and shows that strictly above this threshold, exact recovery is efficiently solvable (although the complexity may increase with the gap to capacity shrinking). The converse and the behavior at the threshold require a few more technical steps.

Several papers have investigated the labelled SBM with labels, we refer in particular to Heimlicher et al. (2012); Xu et al. (2014); Jog and Loh (2015); Yun and Proutiere (2015). A special case of labelled block model with further applications to synchronization, correlation clustering and objet alignment problems has been defied as the censored block model in Abbe et al. (2014a,b), and was further studied in Chin et al. (2015); Saade et al. (2015); Chen et al. (2014); Chen and Goldsmith (2014). This model captures a setting in which the edges carry information about the similarity of the nodes, whereas non-edges carry zero information (as opposed to the SBM where non-edges carry a little bit of information).

Overlapping communities. Consider the following model that accounts for overlapping communities, which we call the overlapping stochastic block model (OSBM).

Definition 27 *Let $n,t\in\mathbb{Z}_+$, $f:\{0,1\}^t\times\{0,1\}^t\to[0,1]$ symmetric, and p a probability distribution on $\{0,1\}^t$. A random graph with distribution OSBM(n,p,f) is generated on the vertex set $[n]$ by drawing independently for each $v\in[n]$ the vector-labels (or user profiles) $X(v)$ under p, and by drawing independently for each $u,v\in[n]$, $u<v$, an edge between u and v with probability $f(X(u),X(v))$.*

Example 1 *One may consider $f(x,y)=\theta_g(x,y)$, where x_i encodes whether a node is in community i or not, and*

$$\theta_g(x,y)=g(\langle x,y\rangle), \tag{77}$$

where $\langle x,y\rangle=\sum_{i=1}^t x_iy_i$ counts the number of common communities between the labels x and y, and $g:\{0,1,\dots,t\}\to[0,1]$ is a function that maps the overlap score into probabilities (g is typically increasing).

We can represent the OSBM as a SBM with $k=2^t$ communities, where each community represents a possible profile in $\{0,1\}^t$. For example, two overlapping communities can be modelled by assigning nodes with a single attribute $(1,0)$ and $(0,1)$ to each of the disjoint communities and nodes with both attributes $(1,1)$ to the overlap community, while nodes having none of the attributes, i.e., $(0,0)$, may be assigned to the null community.

Assume now that we identify community $i\in[k]$ with the profile corresponding to the binary expansion of $i-1$. The prior and connectivity matrix of the corresponding SBM are then given by

$$p_i=p(b(i)) \tag{78}$$
$$W_{i,j}=f(b(i),b(j)), \tag{79}$$

where $b(i)$ is the binary expansion of $i-1$, and

$$\text{OSBM}(n,p,f)\overset{(d)}{=}\text{SBM}(n,p,W). \tag{80}$$

We can then use the results of previous sections to obtain exact recovery in the OSBM.

Corollary 28 *Exact recovery is solvable for the OSBM if the conditions of Theorem 14 apply to the SBM(n,p,W) with p and W as defined in* (78), (79).

This approach treats intersections of communities as sub-communities, and proceeds in extracting these in the case where all overlaps are of linear size. When the parameters are such that the original communities can be identified from the sub-communities, this allows to reconstruct the original communities. If the patching is not identifiable, nothing can be done to improve on what the above corollary provides. However, this approach does not seem very practical for large number of communities or for small overlaps, where one would like to have an approach that provides a soft membership of each vertex to different communities (such as a probability distribution). This connects also to the mixed-membership model Airoldi et al. (2008), and to the case of SBMs with continuous vertex labels, for which exact recovery is typically not the right metric. The problems are fairly open in such contexts, as further discussed in Section 8, with partial progress in Kaufmann et al. (2015).

Previous result can also applied to understand what level of granularity can be obtained in extracting hierarchical communities. A simple example is the case of two communities, where one of the two communities is further divided into two sub-communities, leading to connectivity matrices that encode such a nested structure. A particular case concerns the model with a single planted community, for which detailed results have been obtained in Montanari (2015); Hajek et al. (2015b,a) both for weak and exact recovery.

Bipartite communities. Another important application concerns bipartite graphs, where communities can take place on both sides of the graph. This is also called bi-clustering, and happens for example in recommendation systems, where the two sides separate users and items (such as movies), internet with users and webpages, topic modelling with documents and words, and many other examples.

From a block model point of view, such bipartite models are simply SBMs where some of the Q_{ij}'s are equal to 0. Consider for example the problem of finding botnets, where the left nodes represent the users separated in two communities, A and B, corresponding to human and robots, and where the right nodes represent the webpages separated in two communities, 1 and 2, corresponding to normal and infected pages. We can use an SBM with 4 communities such that $W_{1,1}=W_{1,2}=W_{2,2}=W_{A,A}=W_{A,B}=W_{B,B}=0$.

On can then use Theorem 14 to obtain the fundamental limit for extracting communities, or the extension of Section 3.5.2 to extract a specific community (e.g., the robots in previous example). Note that the establishment of the fundamental limit allows to quantify the fact that treating the data in the bipartite model strictly improves on collapsing the data into a single clustering problem. The collapsing part refers to building a simple graph out of the bipartite graph Zhou et al. (2007), keeping only the right or left nodes (but not both), and connecting them based on how much common neighbors the vertices have on the other side.

Hypergraphs. Another extension is concerned with hypergraphs, where the observations are not pairwise interactions but triplets or k-tuples. This takes place for example in collaboration networks, such as citation networks, where a publication represents an hyperedge on multiple authors. One can easily extend the SBM in such settings, giving a different probability for each type of hyperedges (i.e., hyperedges that are fully in one community, or that have different proportions of vertices in different communities). The fundamental limit for exact recovery in this model does not follow directly from Theorem 14, but the techniques described in Sections 3.2.1 and 3.2.2 apply, and in particular, the approach described for labeled edges above. We also refer to Angelini et al. (2015) for spectral algorithms in this context.

3.5.2 Subset of communities

Before delving into the partial recovery of the communities, one may ask whether it possible to exactly recover *subsets* of the communities, or only a specific community, while not necessarily being able to recover all others. This question is answered in Abbe and Sandon (2015b) as follows.

To determine which communities can be recovered, partition the community profiles into the largest collection of disjoint subsets such that the CH-divergence among these subsets is at least 1 (where the CH-divergence between two subsets is the minimum of the CH-divergence between any two elements in these subsets). We refer to this as the *finest partition* of the communities. Note that this gives the set of connected components in the graph where each community is a vertex, and two communities are adjacent if and only if they have a CH-divergence less than 1. Figure 3 illustrates this partition. The theorem below shows that this is indeed the most granular partition that can be recovered about the communities, in particular, it characterizes the information-theoretic and computational threshold for exact recovery in this setting.

Theorem 29 *Abbe and Sandon (2015b) Let Q be a $k \times k$ matrix with nonzero entries, $p \in (0,1)^k$ with $\sum p = 1$. Exact recovery is information-theoretically solvable in* $\mathrm{SBM}(n,p,\log(n)Q/n)$ *for a partition $[k] = \sqcup_{s=1}^{t} A_s$ if and only if for all i and j in different subsets of the partition,*

$$D_+((PQ)_i,(PQ)_j) \geq 1, \tag{81}$$

Moreover, exact recovery for a partition is efficiently solvable whenever it is information-theoretically solvable.

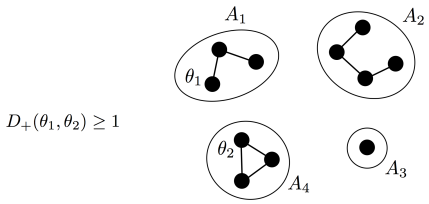


Figure 3: Finest partition: To determine which communities can be recovered in $\mathrm{SBM}(n,p,\log(n)Q/n)$, embed each community with its community profile $\theta_i = (\mathrm{diag}\, pQ)_i$ in $\mathbb{R}_+^k$ and find the partition of $\theta_1,\ldots,\theta_k$ into the largest number of subsets that are at CH-divergence at least 1 from each other.

4. Weak Recovery (a.k.a. Detection)

The focus on weak recovery, also called detection, initiated[10] with Coja-Oghlan (2010); Decelle et al. (2011). Note that weak recovery is typically investigated in SBMs where vertices have constant expected degree, as otherwise the problem can trivially be resolved by exploiting the degree variations.

4.1 Fundamental limit and KS threshold

The following conjecture was stated in Decelle et al. (2011) based on deep but non-rigorous statistical physics arguments, and is responsible in part for the resurged interest on the SBM:

Conjecture 30 *Decelle et al. (2011); Mossel et al. (2015) Let (X,G) be drawn from* $\mathrm{SSBM}(n,k,a/n,b/n)$*, i.e., the symmetric SBM with k communities, probability a/n inside the communities and b/n across. Define* $\mathrm{SNR} = \frac{(a-b)^2}{k(a+(k-1)b)}$*. Then,*

(i) For any $k \geq 2$, it is possible to solve weak recovery efficiently if and only if $\mathrm{SNR} > 1$ *(the Kesten-Stigum (KS) threshold);*

(ii) If[11] *$k \geq 4$, it is possible to solve weak recovery information-theoretically (i.e., not necessarily in polynomial time in n) for some* SNR *strictly below 1.*[12]

It was proved in Massoulié (2014); Mossel et al. (2014a) that the KS threshold can be achieved efficiently for $k = 2$, with an alternative proof later given in Bordenave et al. (2015), and Mossel et al. (2015) shows that it is impossible to detect below the KS threshold for $k = 2$. Further, Deshpande et al. (2015) extends the results for $k = 2$ to the case

10. The earlier work Reichardt and Leone (2008) also considers detection in the SBM.
11. The conjecture states that $k = 5$ is necessary when imposing the constraint that $a > b$, but $k = 4$ is enough in general.
12. Decelle et al. (2011) made in fact a more precise conjecture, stating that there is a second transition below the KS threshold for information-theoretic methods when $k \geq 4$, whereas there is a single threshold when $k = 3$.

where a and b diverge while maintaining the SNR finite. So weak recovery is closed for $k = 2$ in SSBM. It was also shown in Bordenave et al. (2015) that for SBMs with multiple communities satisfying a certain asymmetry condition (i.e., the requirement that μ_k is a simple eigenvalue in Theorem 5 of Bordenave et al. (2015)), the KS threshold can be achieved efficiently. The asymmetry requirement of Bordenave et al. (2015) does not allow to resolve Conjecture 30 for $k \geq 3$.

Concerning crossing the KS threshold with information theory, a few papers have studied bounds and information-computation tradeoffs for SBMs with a growing number of communities Y. Chen (2014), two unbalanced[13] communities Neeman and Netrapalli (2014), and a single community Montanari (2015). These do not apply to Conjecture 1 part (ii). Both positive parts of Conjecture 1 have been proved in Abbe and Sandon (2015) (with a simplified algorithm presented in Abbe and Sandon (2016b), a simplified bound in Abbe and Sandon (2016a), and a full version in Abbe and Sandon (2017)). Papers Banks and Moore (2016); Banks et al. (2016) also provide an upper-bound on the information-theoretic threshold, which crosses the KS threshold for $k = 5$ rather than $k = 4$ in the symmetric case, but with in addition a lower-bound that matches the scaling of the upper-bound for large k.

Note that the terminology 'KS threshold' comes from the reconstruction problem on trees Kesten and Stigum (1966); Evans et al. (2000); Mossel and Peres (2003); Mézard and Montanari (2006). In the binary case, a transmitter broadcasts a uniform bit to some relays, which themselves forward the received bits to other relays, etc. The goal is to reconstruct the root bit from the leaf bits as the depth of the tree diverges. In particular, for two communities, Mossel et al. (2015) makes a reduction between failing in the reconstruction problem in the tree setting and failing in detection in the SBM. This is discussed in more details in next section. The fact that the reconstruction problem on tree also gives the positive behavior for efficient algorithm requires a more involved argument discussed in Section 4.5.1.

Achieving the KS threshold raises an interesting challenge for community detection algorithms, as standard clustering methods fail to achieve the threshold Krzakala et al. (2013). This includes spectral methods based on the adjacency matrix or standard Laplacians, as well as SDPs. For standard spectral methods, a first issue is that the fluctuations in the node degrees produce high-degree nodes that disrupt the eigenvectors from concentrating on the clusters Krzakala et al. (2013).[14] A classical trick is to trim such high-degree nodes Coja-Oghlan (2010); Vu (2014); Guédon and Vershynin (2016); Chin et al. (2015), throwing away some information, but this does not suffice to achieve the KS threshold. SDPs are a natural alternative, but they also stumble[15] before the KS threshold Guédon and Vershynin (2016); Montanari and Sen (2016), focusing on the most likely rather than typical clusterings. As shown in Bordenave et al. (2015); Abbe and Sandon (2015), a linearized BP algorithm which corresponds to a spectral algorithm on a generalized non-backtracking operator provide instead a solution to the positive part to Conjecture 1 part (i).

13. Detailed results were recently obtained in Caltagirone et al. (2016) for unbalanced communities with diverging degrees.
14. This issue is further enhanced on real networks where degree variations are large.
15. The recent results of Moitra et al. (2016) on robustness to monotone adversaries suggest that SDPs can in fact not achieve the KS threshold.

4.2 Impossibility below KS for $k = 2$ and reconstruction on trees

Theorem 31 *Mossel et al. (2015) For $k = 2$, weak recovery is not solvable if* SNR ≤ 1 *(i.e., $(a-b)^2 \leq 2(a+b)$).*

This result is obtained by a reduction to the problem of reconstruction on trees, which is next discussed; we refer to Mossel and Peres (2003) for a survey on this problem. An addition result is also obtained in Mossel et al. (2015), showing that when SNR ≤ 1, the symmetric SBM with two communities is in fact contiguous to the Erdős-Rénti model with edge probability $(a+b)/(2n)$, i.e, distinguishability is not solvable in this case. Contiguity is further discussed in Section 4.6.1.

Reconstruction on trees. The problem consists in broadcasting a bit from the root of a tree down to its leaves, and trying to guess back this bit from the leaves at large depth. Consider first the case of a deterministic tree with fixed degree $c+1$, i.e., each vertex has exactly c descendants (note that the root has degree c). Assume that on each branch of the tree the incoming bit is flipped with probability $\varepsilon \in [0,1]$, and that each branch acts independently. Let $X^{(t)}$ be the bits received at depth t in this tree, with $X^{(0)}$ being the root bit, assumed to be drawn uniformly at random in $\{0,1\}$.

We now define successful detection in this context. Note that $\mathbb{E}(X^{(0)}|X^{(t)})$ is a random variable that gives the probability that $X^{(0)} = 1$ given the leaf-bits, as a function of the leaf-bits $X^{(t)}$. If this probability is equal to $1/2$, then the leaf-bits provide no useful information about the root, and we are interested in understanding whether this takes place or not in the limit of large t.

Definition 32 *Detection (usually called reconstruction) in the tree model is solvable if* $\lim_{t\to\infty} \mathbb{E}|\mathbb{E}(X^{(0)}|X^{(t)}) - 1/2| > 0$. *Equivalently, detection is solvable if* $\lim_{t\to\infty} I(X^{(0)}; X^{(t)}) > 0$, *where I is the mutual information.*

Note that the above limits exist due to monotonicity arguments. The results for detection this model are as follows.

Theorem 33 *In the tree model with constant degree c and flip probability ε,*

- *Kesten and Stigum (1966) detection is solvable if $c(1-2\varepsilon)^2 > 1$,*
- *Bleher et al. (1995); Evans et al. (2000) detection is not solvable*[16] *if $c(1-2\varepsilon)^2 \leq 1$.*

Thus detection in the tree model is solvable if and only if $c(1-2\varepsilon)^2 > 1$, which gives rise to the so-called Kesten-Stigum (KS) threshold in this tree context. Note that Mossel and Peres (2003) further shows that the KS threshold is sharp for "census reconstruction," i.e., deciding about the root-bit by taking majority on the leaf-bits, which is shown to still hold in models such as the multicolor Potts model where the KS threshold is no longer sharp for reconstruction.

To see the two parts of Theorem 33, note that the number 0-bits minus 1-bits at generation t, i.e, $\sum_{i\in[c^t]}(-1)^{X_i^{(t)}}$ is a random variable with expectation of order $c^t(1-2\varepsilon)^t$

16. The proof from Evans et al. (2000) appeared first in 1996.

with a sign depending on the value of the root-bit, and with variance of order c^t. Thus the signal-to-noise ratio (i.e., the ratio between the expectation and the standard deviation of this statistic) is $(\sqrt{c}(1-2\varepsilon))^t$, and $\sqrt{c}(1-2\varepsilon) > 1$ allows to make reliable inference about the root-bit as t diverges. In the binary case and with the flip model considered for the noise, it turns out that the mutual information is sub-additive among leaves Evans et al. (2000), i.e., $I(X^{(0)}; X^{(t)}) \leq \sum_{i=1}^{c^t} I(X^{(0)}; X_i^{(t)}) = c^t I(X^{(0)}; X_1^{(t)})$ (this is however not true in general for binary non-symmetric noise or for non-binary labels). The channel between $X^{(0)}$ and a single leaf-bit $X_1^{(t)}$ corresponds to the addition of t Bernoulli(ε) random variables, and it is easy to check that its mutual information scales as $(1-2\varepsilon)^{2t}$, which shows that $I(X^{(0)}; X^{(t)})$ is upper bounded by $c^t(1-2\varepsilon)^{2t}$. Hence, if $c(1-2\varepsilon)^2$ is less than 1, the information of the root-bit is lost.

We will soon turn to the connection between the reconstruction on tree problem and weak recovery in the SBM. It is easy to guess that the tree for us will not be a fixed degree tree, but the local neighborhood of an SBM vertex, which is a Gaton-Watson tree with Poisson offspring. We first state the above results for Galton-Walton trees.

Definition 34 *A Galton-Walton tree with offspring distribution μ on $\mathbb{Z}_+$ is a rooted tree where the number of descendants from each vertex is independently drawn under the distribution μ. We denote by $T^{(t)} \sim GW(\mu)$ a Galton-Walton tree with offspring μ and t generations of descendants.*

Note that detection in the context of a random tree is defined by requiring that

$$\lim_{t\to\infty} \mathbb{E}|\mathbb{E}(X^{(0)}|X^{(t)}, T^{(t)}) - 1/2| > 0,$$

where $X^{(t)}$ are the variables at generation t obtained from broadcasting the root-bit as in the previous case. In Evans et al. (2000), it is shown that the threshold $c(1-2\varepsilon)^2 > 0$ is necessary and sufficient for detection for a large class of offspring distributions, where c is the expectation of μ, such as the Poisson(c) distribution that is of interest to us. Another important extension is the 'robust reconstruction' problem Janson and Mossel (2004), where the leaves are not revealed exactly but in a noisy fashion. It was shown in Janson and Mossel (2004) that for very noisy leaves, the KS threshold is also tight.

The connection with the SBM comes from the fact that if one picks a vertex v in the SBM graph, its neighborhood at small enough depth behaves likes a Galton-Watson tree of offspring Poisson($(a+b)/2$), and the labelling on the vertices behaves like the broadcasting process discussed above with a flip probability of $b/(a+b)$. Note that the latter parameter is precisely the probability that two vertices have different labels given that there is an edge between them. More formally, if the depth is $t \leq (1/2-\delta)\log(n)/\log(a+b)/2)$ for some $\delta > 0$, then the true distribution and the above one have a vanishing total variation when n diverges. This depth requirement can be understood from the fact the expected number of leaves in that case is in expectation $n^{1/2-\delta}$, and by the birthday paradox, no collision will likely occur between two vertices neighborhoods if $\delta > 0$.

To establish Theorem 31, it is sufficient to argue that, if it impossible to detect a single vertex when a genie reveals all the leaves at such a depth, it must be impossible to detect. In fact, consider $\mathbb{P}\{X_u = x_u | G = g, X_v = x_v\}$, the posterior distribution given the graph

and an arbitrary vertex revealed (here u and v are arbitrary and chosen before the graph is drawn). With high probability, these vertices will not be at small graph-distance of each other, and one can open a small neighborhood around u of dept, say, $\log\log(n)$. Now reveal not only the value of X_v but in fact all the values at the boundary of this neighborhood. This is an easier problem since the neighborhood is a tree with high probability and since there is approximately a Markov relationship between these boundary vertices and the original X_v (note that 'approximate' is used here since there is a negligible effect of non-edges to handle). We are now back to the broadcasting problem on tree discussed above, and the requirement $c(1-2\varepsilon)^2 \leq 0$ gives a sufficient condition for detection to fail. Since $c = (a+b)/2$ and $\varepsilon = b/(a+b)$, this gives $(a-b)^2 \leq 2(a+b)$.

The reduction extends to more than two communities (i.e., non-binary labels broadcasted on trees) and to asymmetrical communities, but the tightness of the KS bound is no longer present in these cases. For two asymmetrical communities, the result still extends if the communities are roughly symmetrical, using Borgs et al. (2006) and Mossel et al. (2013); pri. For more than three symmetric communities, new gap phenomena take place as discussed in Section 4.6.

4.3 Achieving KS for $k = 2$

Theorem 35 *Massoulié (2014); Mossel et al. (2014a) For $k = 2$, weak recovery is efficiently solvable if* SNR > 1 *(i.e., $(a-b)^2 > 2(a+b)$).*

The first paper Massoulié (2014) is based[17] on a spectral method from the matrix of self-avoiding walks (entry (i,j) counts the number of self-avoiding walks of moderate size between vertices i and j) Massoulié (2014), the second on counting weighted non-backtracking walks between vertices Mossel et al. (2014a). The first method has a complexity of $O(n^{1+\varepsilon})$, $\varepsilon > 0$, while the second method affords a lesser complexity of $O(n\log^2 n)$ but with a large constant (see discussion in Mossel et al. (2014a)). These papers appeared in 2014 and were the first to achieve the KS threshold for two communities.

Later in 2015, Bordenave et al. (2015) obtains an alternative proof on a spectral method with the matrix of non-backtracking walks between directed edges. The paper gives a detailed analysis of the spectrum of the nonbacktracking operator and allows going beyond the SBM with 2 communities, requiring a certain condition in the SBM parameters to obtain a result for detection (the precise conditions are the uniformity of p and the requirement that μ_k is a simple eigenvalue of M in Theorem 5 of Bordenave et al. (2015)), falling short of proving the positive part of Conjecture 1.(i) for $k \geq 3$ due to technical reasons (the second eigenvalue in this case has multiplicity at least 2). In Abbe and Sandon (2015), another variant based on higher order nonbacktracking power iterations is shown to achieve the KS threshold in the general SBM.

The nonbacktracking operator was first proposed for the SBM in Krzakala et al. (2013), also described as a linearization of BP, with formal results obtained in Bordenave et al. (2015). This gives a new approach to community detection, and a strong case for nonbacktracking operators. Linearizations of BP of similar nature were already used in Janson and Mossel (2004) in the context of robust reconstruction on trees. As the nonbacktracking

17. Related ideas relying on shortest paths were also considered in Bhattacharyya and Bickel (2014).

matrix is not normal, Saade et al. (2014) also introduced an alternative approach based on the Bethe Hessian operator.

We next discuss the algorithm of Bordenave et al. (2015) for 2 symmetric communities. We define first the nonbacktracking matrix of a graph.

Definition 36 *[The nonbacktracking (NB) matrix.] Hashimoto (1989) Let $G = (V, E)$ be a simple graph and let $\vec{E}$ be the set of oriented edges obtained by doubling each edge of E into two directed edge. The non-backtracking matrix B is a $|\vec{E}| \times |\vec{E}|$ matrix indexed by the elements of $\vec{E}$ such that, for $e = (e_1, e_2), f = (f_1, f_2) \in \vec{E}$,*

$$B_{e,f} = \mathbb{1}(e_2 = f_1)\mathbb{1}(e_1 \neq f_2), \tag{82}$$

i.e., entry (e, f) of B is 1 if e and f follow each other without creating a loop, and 0 otherwise.

The non-backtracking matrix can be used to count efficiently non-backtracking walks in a graph. Recall that a walk in a graph is a sequence of adjacent vertices whereas a non-backtracking walk is a walk that does not repeat a vertex within 2 steps. Counting walks of a given length can simply be done by taking powers of the adjacency matrix. The non-backtracking matrix allows for a similar approach to count nonbacktracking walks between edges. To obtain the number of non-backtracking walks of length $k \geq 2$ starting at a directed edge e and ending in a directed edge f, one simply needs to take entry (e, f) of the power matrix B^{k-1}. Note also that to count paths, i.e., walks that do not repeat any vertex, no such efficient method is known and the count is #P-complete.

The nonbacktracking matrix B of a graph was introduced by Hashimoto Hashimoto (1989) to study the Ihara zeta function, with the identity $\det(I - zB) = \frac{1}{\zeta(z)}$, where ζ is the Ihara zeta function of the graph. In particular, the poles of the Ihara zeta function are the reciprocals of the eigenvalues of B. Studying the spectrum of a graph thus implies properties on the location of the Ihara zeta function. The matrix is further used to define the graph Riemann hypothesis Horton et al. (2006), and studying its spectrum for random graphs such as the block model allows for generalizations of notions of Ramanujan graphs and Friedman's Theorem Friedman (2003) to non-regular cases, see also Bordenave et al. (2015). The operator that we study is a natural extension of the classical nonbacktracking operator of Hashimoto, where we prohibit not only standard backtracks but also finite cycles.

We now describe the spectral algorithm based on the nonbacktracking matrix to detect communities in the symmetric SBM with two communities.

Nonbacktracking eigenvector extraction algorithm Krzakala et al. (2013); Bordenave et al. (2015).
Input: An n-vertex graph g and a parameter $\tau \in \mathbb{R}$.
(1) Construct the nonbacktracking matrix B of the graph g.
(2) Extract the eigenvector ξ_2 corresponding to the second largest eigenvalue of B.
(3) Assign vertex v to the first community if $\sum_{e:e_2=v} \xi_2(e) > \tau/\sqrt{n}$ and to the second community otherwise.

It is shown in Bordenave et al. (2015) that the exists a $\tau \in \mathbb{R}$ such that above algorithm solves detection if $(a-b)^2 > 2(a+b)$, i.e., down to the KS threshold. For more than two communities, the above algorithm needs to be modified and its proof of detection currently applies to some cases of SBMs as previously discussed (balanced communities and no multiplicity of eigenvalues).

A quick intuition on why the nonbacktracking matrix is more amenable for community detection than the adjacency matrix is obtained by taking powers of these matrices. In the case of the adjacency matrix, powers are counting walks from a vertex to another, and these get multiplied around high-degree vertices since the walk can come in and out of such vertices in multiple ways. Instead, by construction of the nonbacktracking matrix, taking powers forces a directed edge to leave to another directed edge that does not backtrack, preventing such amplifications around high-degree vertices. So the nonbacktracking gives a way to mitigate the degree-variations and to avoid localized eigenvector (recall discussion in Section 2.6), arguably more efficiently than trimming which removes information from the graph. This property is reflected in the spectrum of the nonbacktracking matrix, which has for largest eigenvalue (in magnitude) λ_1 (which is real positive). Then the question if the second largest eigenvalue λ_2 appears before $\sqrt{\lambda_1}$, i.e.,

$$\sqrt{\lambda_1} < |\lambda_2| \leq \lambda_1, \tag{83}$$

weak recovery can be solved by using the eigenvector corresponding to λ_2 to obtain the non-trivial separation Bordenave et al. (2015).

Extracting the second eigenvector of the nonbacktracking matrix directly may not be the most efficient way to proceed, specially as the graph gets denser. A power iteration method is a natural implementation, likely to be used by softwares, but this requires additional proofs as discussed next. The approach of Mossel et al. (2014a) based on the count of weighted nonbacktracking walks between vertices provides another practical alternative.

4.4 Achieving KS for general k

The next result proves the positive parts of Conjecture 1.

Theorem 37 *Abbe and Sandon (2015) (part 1 presented in Abbe and Sandon (2016b) and part 2 presented in Abbe and Sandon (2016a).)*

1. *For any $k \geq 2$, weak recovery is solvable in $O(n \log n)$ if* SNR > 1 *with the approximate acyclic belief propagation (ABP) algorithm;*

2. *For any $k \geq 4$, weak recovery is information-theoretically solvable for some* SNR *strictly below* 1 *with the typicality sampling (TS) algorithm.*

We describe next the two algorithms used in previous theorem. In brief, ABP is a belief propagation algorithm where the update rules are linearized around the uniform prior and where the feedback on short cycles is mitigated; TS is a non-efficient algorithm that samples uniformly at random a clustering having typical clusters' volumes and cuts. The fact that BP with a random initialization can achieve the KS threshold for arbitrary k was conjectured in the original paper Decelle et al. (2011), but handling random initialization and cycles with

BP is a classical challenge. A linearized version is more manageable to analyze, although the effect of cycles remains a difficulty to overcome.

The simplest linearized version of BP is to repeatedly update beliefs about a vertex's community based on its neighbor's suspected communities while avoiding backtrack. However, this only works ideally if the graph is a tree. The correct response to a cycle would be to discount information reaching the vertex along either branch of the cycle to compensate for the redundancy of the two branches. Due to computational issues we simply prevent information from cycling around constant size cycles. We also add steps where a multiple of the beliefs in the previous step are subtracted from the beliefs in the current step to prevent the beliefs from settling into an equilibrium where vertices' communities are sytematically misrepresented in ways that add credibility to each other.

Approximate message passing algorithms have also been developed in other contexts, such as in Donoho et al. (2009) for compressed sensing with approximate message passing (AMP) and state evolution. This approach applies to dense graphs whereas the approximation of ABP applies to the sparse regime. We refer to Section 6.4 for discussions on how the output of ABP can be fed into standard BP in order to achieve optimal accuracy for partial recovery.

The approach of ABP is also related to Mossel et al. (2014a); Bordenave et al. (2015), while diverging in several parts. Some technical expansions are similar to those carried in Mossel et al. (2014a), such as the weighted sums over nonbacktracking walks and the SAW decomposition in Mossel et al. (2014a), which are similar to the compensated nonbacktracking walk counts and Shard decomposition of Abbe and Sandon (2015). The approach of Abbe and Sandon (2015) is however developed to cope with the general SBM model, in particular the compensation of dominant eigenvalues due to the linearization, which is particularly delicate. The algorithm complexity of Abbe and Sandon (2015) is also slightly reduced by a logarithmic factor compared to Mossel et al. (2014a).

As seen below, ABP can also be interpreted as a power iteration method on a generalized nonbacktracking operator, where the random initialization of the beliefs in ABP corresponds to the random vector to which the power iteration is applied. This formalizes the connection described in Krzakala et al. (2013) and makes ABP closely related to Bordenave et al. (2015) that proceeds with eigenvector extraction rather than power iteration. This distinction requires particular care at the proof level. The approach of ABP also differs from Bordenave et al. (2015) in that it relies on a generalization of the nonbacktracking matrix Hashimoto (1989) with higher order nonbacktracks (see Definition 38 below), and relies on different proof techniques to cope with the setting of Conjecture 1. The current proof requires the backtrack order r to be a constant but not necessarily 2. While we believe that $r = 2$ may suffice for the sole purpose of achieving the KS threshold, we also suspect that larger backtracks may be necessary for networks with more small cycles, such as many of those that occur in practice.

We next describe the message passing implementation of ABP with a simplified version ABP* that applies to the general SBM (with constant expected degree but not necessarily symmetric; see Section 4.5 for its performance in the general SBM). We then define the generalized nonbacktracking matrix and the spectral counter-part of ABP*.

ABP*. Abbe and Sandon (2016b)
Input: a graph G and parameters $m, r \in \mathbb{Z}_+$.

1. For each adjacent v and v' in G, randomly draw $y^{(1)}_{v,v'}$ from a Normal distribution. Assign $y^{(t)}_{v,v'}$ to 0 for $t < 1$.

2. For each $1 < t \le m$, set

$$z^{(t-1)}_{v,v'} = y^{(t-1)}_{v,v'} - \frac{1}{2|E(G)|} \sum_{(v'',v''')\in E(G)} y^{(t-1)}_{v'',v'''}$$

for all adjacent v and v'. For each adjacent v, v' in G that are not part of a cycle of length r or less, set

$$y^{(t)}_{v,v'} = \sum_{v'':(v',v'')\in E(G), v''\neq v} z^{(t-1)}_{v',v''},$$

and for the other adjacent v, v' in G, let v''' be the other vertex in the cycle that is adjacent to v, the length of the cycle be r', and set

$$y^{(t)}_{v,v'} = \sum_{v'':(v',v'')\in E(G), v''\neq v} z^{(t-1)}_{v',v''} - \sum_{v'':(v,v'')\in E(G), v''\neq v', v''\neq v'''} z^{(t-r')}_{v,v''}$$

unless $t = r'$, in which case set $y^{(t)}_{v,v'} = \sum_{v'':(v',v'')\in E(G), v''\neq v} z^{(t-1)}_{v',v''} - z^{(1)}_{v''',v}$.

3. Set $y'_v = \sum_{v':(v',v)\in E(G)} y^{(m)}_{v,v'}$ for all $v \in G$. Return $(\{v : y'_v > 0\}, \{v : y'_v \le 0\})$.

Remarks:

1. In the $r = 2$ case, one does not need to find cycles and one can exit step 2 after the second line. As mentioned above, we rely on a less compact version of the algorithm to prove the theorem, but expect that the above also succeeds at detection as long as $m > 2\ln(n)/\ln(\mathrm{SNR}) + \omega(1)$.

2. What the algorithm does if (v, v') is in multiple cycles of length r or less is unspecified above, as there is no such edge with probability $1 - o(1)$ in the sparse SBM. This can be modified for more general settings. The simplest such modification is to apply this adjustment independently for each such cycle, setting

$$y^{(t)}_{v,v'} = \sum_{\substack{v'':(v',v'')\in E(G)\\ v''\neq v}} z^{(t-1)}_{v',v''} - \sum_{r'=1}^{r} \sum_{v''':(v,v''')\in E(G)} C^{(r')}_{v''',v,v'} \sum_{\substack{v'':(v,v'')\in E(G)\\ v''\neq v', v''\neq v'''}} z^{(t-r')}_{v,v''},$$

where $C^{(r')}_{v''',v,v'}$ denotes the number of length r' cycles that contain v''', v, v' as consecutive vertices, substituting $z^{(1)}_{v''',v}$ for $\sum_{v'':(v,v'')\in E(G), v''\neq v', v''\neq v'''} z^{(t-r')}_{v,v''}$ when $r' = t$. This does not exactly count r-nonbacktracking walks, but it gives a good enough approximation.

3. The purpose of setting $z^{(t-1)}_{v,v'} = y^{(t-1)}_{v,v'} - \frac{1}{2|E(G)|}\sum_{(v'',v''')\in E(G)} y^{(t-1)}_{v'',v'''}$ is to ensure that the average value of the $y^{(t)}$ is approximately 0, and thus that the eventual division of the vertices into two sets is roughly even. There is an alternate way of doing this in which we simply let $z^{(t-1)}_{v,v'} = y^{(t-1)}_{v,v'}$ and then compensate for any bias of $y^{(t)}$ towards positive or negative values at the end. More specifically, we define Y to be the $n \times m$ matrix such that for all t and v, $Y_{v,t} = \sum_{v':(v',v)\in E(G)} y^{(t)}_{v,v'}$, and M to be the $m \times m$ matrix such that $M_{i,i} = 1$ and $M_{i,i+1} = -\lambda_1$ for all i, and all other entries of M are equal to 0. Then we set $y' = YM^{m'}e_m$, where $e_m \in \mathbb{R}^m$ denotes the unit vector with 1 in the m-th entry, and m' is a suitable integer.

4. This algorithm is intended to classify vertices with an accuracy nontrivially better than that attained by guessing randomly. However, it is relatively easy to convert this to an algorithm that classifies vertices with optimal accuracy. Once one has reasonable initial guesses of which communities the vertices are in, one can then use full belief propagation to improve this to an optimal classification. See further details in Section 6.4.

In order to prove that ABP solves detection, a few modifications are made relative to the vanilla version described above. The main differences are as follows. First, at the end we assign vertices to sets with probabilities that scale linearly with their entries in y' instead of simply assigning them based on the signs of their entries. This allows us to use the fact that the average values of y'_v for v in different communities differ to prove that vertices from different communities have different probabilities of being assigned to the first set. Second, we remove a small fraction of the edges from the graph at random at the beginning of the algorithm. Then we define y''_v to be the sum of $y'_{v'}$ over all v' connected to v by paths of a suitable length with removed edges at their ends in order to eliminate some dependency issues. Also, instead of just compensating for PQ's dominant eigenvalue, we also compensate for some of its smaller eigenvalues. We refer to Abbe and Sandon (2015) for the full description of the official ABP algorithm. Note that while it is easier to prove that the ABP algorithm works, the ABP* algorithm should work at least as well in practice.

We now define the generalized nonbacktracking matrix and the spectral implementation of ABP.

Definition 38 *[The r-nonbacktracking (r-NB) matrix.] Let $G = (V, E)$ be a simple graph and let $\vec{E}_r$ be the set of directed paths of length $r-1$ obtained on E. The r-nonbacktracking matrix $B^{(r)}$ is a $|\vec{E}_r| \times |\vec{E}_r|$ matrix indexed by the elements of $\vec{E}_r$ such that, for $e = (e_1, \dots, e_{r-1}), f = (f_1, \dots, f_{r-1}) \in \vec{E}_r$,*

$$B^{(r)}_{e,f} = \prod_{i=1}^{r-1} \mathbb{1}((e_{i+1})_2 = (f_i)_1)\mathbb{1}((e_1)_1 \neq (f_{r-1})_2), \tag{84}$$

i.e., entry (e, f) of $B^{(r)}$ is 1 if f extends e by one edge (i.e., the last $r-1$ edges of e agree with the first $r-1$ edges of f) without creating a loop, and 0 otherwise.

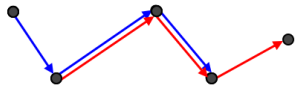


Figure 4: Two paths of length 3 that contribute to an entry of 1 in $B^{(4)}$.

Remark 39 *Note that $B^{(2)} = B$ is the classical nonbacktracking matrix from Definition 36. As for $r = 2$, we have that $((B^{(r)})^{k-1})_{e,f}$ counts the number of r-nonbacktracking walks of length k from e to f.*

r-nonbacktracking power iteration algorithm. Abbe and Sandon (2015)
Input: a graph G and parameters $m, m' \in \mathbb{Z}_+$; denote by d the average degree of the graph.
(1) Draw $y^{(1)}$ of dimension $|\vec{E}_r|$ with i.i.d. Normal components.
(2) For each $1 < t \leq m$, let $y^{(t)} = B^{(r)}y^{(t-1)}$.
(3) Change $y^{(m)}$ to $(B^{(r)} - dI)^{m'}y^{(m-m')}$.
(4) For each v, set $y'_v = \sum_{v':(v',v)\in E(G)} y^{(m)}_{v,v'}$ and return $(\{v : y'_v > 0\}, \{v : y'_v \leq 0\})$.

Parameters should be chosen as discussed for ABP*.

4.5 Weak recovery in the general SBM

Given parameters p and Q in the general model SBM$(n, p, Q/n)$, let P be the diagonal matrix such that $P_{i,i} = p_i$ for each $i \in [k]$. Also, let $\lambda_1, ..., \lambda_h$ be the distinct eigenvalues of PQ in order of nonincreasing magnitude.

Definition 40 *Define the signal-to-noise ratio of* SBM$(n, p, Q/n)$ *by*

$$\text{SNR} = \lambda_2^2/\lambda_1.$$

In the k community symmetric case where vertices in the same community are connected with probability a/n and vertices in different communities are connected with probability b/n, we have $\text{SNR} = (\frac{a-b}{k})^2/(\frac{a+(k-1)b}{k}) = (a-b)^2/(k(a+(k-1)b))$, which is the quantity in Conjecture 1.

Theorem 41 *Bordenave et al. (2015) Let $G \sim$ SBM$(n, p, Q/n)$ such that $p = (1/k, \dots, 1/k)$ and such that PQ has an eigenvector with corresponding eigenvalue in $(\sqrt{\lambda_1}, \lambda_1)$ of single multiplicity. If* SNR > 1*, then weak recovery is efficiently solvable.*

Theorem 42 *Abbe and Sandon (2015); Abbe and Sandon (2017) Let $G \sim$ SBM$(n, p, Q/n)$ for p, Q arbitrary. If* SNR > 1*, then weak recovery is efficiently solvable.*

For the symmetric SBM, the above reduces to part (1) of Theorem 37, which proves the first positive part of Conjecture 1. Note that Decelle et al. (2011) extends Conjecture 1 to the general setting, where max-detection is claimed to be efficiently solvable if and only if $\lambda_2^2 > \lambda_1$. This statement is however not true, as discussed in Section 2.3. An

example is obtained for example by taking an SSBM with two symmetric communities of intra-connectivity $3/n$, small enough extra-connectivity, and breaking one community into 4 identical sub-communities. Then max-detection is not solvable if $\lambda_2^2 > \lambda_1$, but it is solvable for our notion of definition (weak recovery). If one uses that definition, then we also conjecture that such a statement holds, i.e., detection is efficiently solvable if and only if $\lambda_2^2 > \lambda_1$ (besides for the case where λ_1 has multiplicity more than one, for which it is sufficient to have $\lambda_1 > 1$, and always focusing on the case of constant expected degrees).

The full version of ABP is described in Abbe and Sandon (2015), but as for the symmetric case, the version ABP* described in previous section applies to the general setting, replacing d with the largest eigenvalue of PQ. Theorem 42 provides general condition for solving efficiently detection in the SBM with linear size communities. We also conjecture that this is a tight condition, i.e., if SNR < 1, then efficient detection is not solvable. However, establishing formally such a converse argument seems out of reach at the moment: as we shall see in next section, except for a few possible cases with low values of k (e.g., symmetric SBMs with $k = 2, 3$), it is possible to detect information-theoretically when SNR < 1, and thus one cannot get a converse for efficient algorithms by considering all algorithms (requiring significant headways in complexity theory that are likely go beyond the scope of SBMs). On the other hand, Decelle et al. (2011) provides non-formal arguments based on statistical physics arguments that such a converse hold. It would be interesting to further connect the computational barriers occurring here with those from other problems such as planted clique Alon et al. (1998), or as in Berthet and Rigollet (2013).

Finally, the extension to models discussed in Section 3.5 can also be understood in the lens of weak recovery. The case of edge labels or hyperedges need a separate treatment, since the reductions described in Section 3.5 are specific to exact recovery. The converse for weak recovery in the labelled SBM is covered in Heimlicher et al. (2012). The bipartite case can instead be treated as a special case of the threshold $\lambda_2^2/\lambda_1 > 1$ when the matrix Q has 0 entries.

Concerning the separation of specific communities, the results from Abbe and Sandon (2015) imply that communities i and j can be separated if there exists an eigenvector w of PQ such that $w_i \neq w_j$ and the eigenvalue of PQ corresponding to w has a magnitude of at least $\sqrt{\lambda_1}$.

4.5.1 Proof technique: approximate acyclic belief propagation (ABP)

For simplicity, consider first the two community symmetric case where vertices in the same community are adjacent with probability a/n and vertices in different communities are adjacent with probability b/n. We describe the ABP algorithm that implies Theorem 42; more specifically, if SNR > 1, then there exist $r \in \mathbb{Z}^+$, $c > 0$, and $m : \mathbb{Z}^+ \to \mathbb{Z}^+$ such that ABP$(G, m(n), r, c, (\lambda_1, ..., \lambda_h))$ solves detection and runs in $O(n \log n)$ time.

Consider determining the community of v using belief propagation, assuming some preliminary guesses about the vertices t edges away from it, and assuming that the subgraph of G induced by the vertices within t edges of v is a tree. For any vertex v' such that $d(v, v') < t$, let $C_{v'}$ be the set of the children of v'. If we believe based on either our prior knowledge or propagation of beliefs up to these vertices that v'' is in community 1 with probability $\frac{1}{2} + \frac{1}{2}\epsilon_{v''}$ for each $v'' \in C_{v'}$, then the algorithm will conclude that v' is in

community 1 with a probability of

$$\frac{\prod_{v'' \in C_{v'}} (\frac{a+b}{2} + \frac{a-b}{2}\epsilon_{v''})}{\prod_{v'' \in C_{v'}} (\frac{a+b}{2} + \frac{a-b}{2}\epsilon_{v''}) + \prod_{v'' \in C_{v'}} (\frac{a+b}{2} - \frac{a-b}{2}\epsilon_{v''})}.$$

If all of the $\epsilon_{v''}$ are close to 0, then this is approximately equal to

$$\frac{1 + \sum_{v'' \in C_{v'}} \frac{a-b}{a+b}\epsilon_{v''}}{2 + \sum_{v'' \in C_{v'}} \frac{a-b}{a+b}\epsilon_{v''} + \sum_{v'' \in C_{v'}} -\frac{a-b}{a+b}\epsilon_{v''}} = \frac{1}{2} + \frac{a-b}{a+b} \sum_{v'' \in C_{v'}} \frac{1}{2}\epsilon_{v''}.$$

That means that the belief propagation algorithm will ultimately assign an average probability of approximately $\frac{1}{2} + \frac{1}{2}(\frac{a-b}{a+b})^t \sum_{v'': d(v,v'')=t} \epsilon_{v''}$ to the possibility that v is in community 1. If there exists ϵ such that $E_{v'' \in \Omega_1}[\epsilon_{v''}] = \epsilon$ and $E_{v'' \in \Omega_2}[\epsilon_{v''}] = -\epsilon$ (recall that $\Omega_i = \{v : \sigma_v = i\}$), then on average we would expect to assign a probability of approximately $\frac{1}{2} + \frac{1}{2}\left(\frac{(a-b)^2}{2(a+b)}\right)^t \epsilon$ to v being in its actual community, which is enhanced as t increases when SNR > 1. Note that since the variance in the probability assigned to the possibility that v is in its actual community will also grow as $\left(\frac{(a-b)^2}{2(a+b)}\right)^t$, the chance that this will assign a probability of greater than $1/2$ to v being in its actual community will be $\frac{1}{2} + \Theta\left(\left(\frac{(a-b)^2}{2(a+b)}\right)^{t/2}\right)$.

Equivalently, given a vertex v and a small t, the expected number of vertices that are t edges away from v is approximately $(\frac{a+b}{2})^t$, and the expected number of these vertices in the same community as v is approximately $(\frac{a-b}{2})^t$ greater than the expected number of these vertices in the other community. So, if we had some way to independently determine which community a vertex is in with an accuracy of $\frac{1}{2} + \epsilon$ for small ϵ, we could guess that each vertex is in the community that we think that the majority of the vertices t steps away from it are in to determine its community with an accuracy of roughly $\frac{1}{2} + \left(\frac{(a-b)^2}{2(a+b)}\right)^{t/2} \epsilon$.

One idea for the initial estimate is to simply guess the vertices' communities at random, in the expectation that the fractions of the vertices from the two communities assigned to a community will differ by $\theta(1/\sqrt{n})$ by the central limit theorem. Unfortunately, for any t large enough that $\left(\frac{(a-b)^2}{2(a+b)}\right)^{t/2} > \sqrt{n}$, we have that $\left(\frac{(a+b)}{2}\right)^t > n$ which means that our approximation breaks down before t gets large enough to detect communities. In fact, t would have to be so large that not only would neighborhoods not be tree like, but vertices would have to be exhausted.

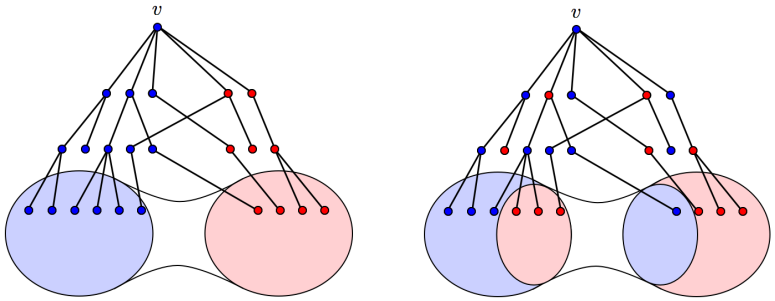


Figure 5: The left figure shows the neighborhood of vertex v pulled from the SBM graph at depth $c \log_{\lambda_1} n$, $c < 1/2$, which is a tree with high probability. If one had an educated guess about each vertex's label, of good enough accuracy, then it would be possible to amplify that guess by considering only such small neighborhoods (deciding with the majority at the leaves). However, we do not have such an educated guess and thus initialize our labels at random, obtaining a small advantage of roughly $\sqrt{n}$ vertices by luck (i.e., the central limit theorem), in either an agreement or disagreement form. This is illustrated in agreement form in the right figure. We next attempt to amplify that lucky guess by exploiting the information of the SBM graph. Unfortunately, the graph is too sparse to let us amplify that guess by considering tree like or even loopy neighborhoods; the vertices would have to be exhausted. This takes us to considering nonbacktracking walks.

One way to handle this would be to stop counting vertices that are t edges away from v, and instead count each vertex a number of times equal to the number of length t paths from v to it.[18] Unfortunately, finding all length t paths starting at v can be done efficiently only for values of t that are smaller than what is needed to amplify a random guess to the extent needed here. We could instead calculate the number of length t walks from v to each vertex more quickly, but this count would probably be dominated by walks that go to a high degree vertex and then leave and return to it repeatedly, which would throw the calculations off. On the other hand, most reasonably short nonbacktracking walks are likely to be paths, so counting each vertex a number of times equal to the number of nonbacktracking walks of length t from v to it seems like a reasonable modification. That said, it is still possible that there is a vertex that is in cycles such that most nonbacktracking walks simply leave and return to it many times. In order to mitigate this, we use r-nonbacktracking walks, walks in which no vertex reoccurs within r steps of a previous occurrence, such that walks cannot return to any vertex more than t/r times.

Unfortunately, this algorithm would not work because the original guesses will inevitably be biased towards one community or the other. So, most of the vertices will have more r-nonbacktracking walks of length t from them to vertices that were suspected of being in that community than the other. One way to deal with this bias would be to subtract the

18. This type of approach is considered in Bhattacharyya and Bickel (2014).

average number of r-nonbacktracking walks to vertices in each set from each vertex's counts. Unfortunately, that will tend to undercompensate for the bias when applied to high degree vertices and overcompensate for it when applied to low degree vertices. So, we modify the algorithm that counts the difference between the number of r-nonbacktracking walks leading to vertices in the two sets to subtract off the average at every step in order to prevent a bias from building up.

One of the features of our approach is that it extends fairly naturally to the general SBM. Despite the potential presence of more than 2 communities, we still only assign one value to each vertex, and output a partition of the graph's vertices into two sets in the expectation that different communities will have different fractions of their vertices in the second set. One complication is that the method of preventing the results from being biased towards one comunity does not work as well in the general case. The problem is, by only assigning one value to each vertex, we compress our beliefs onto one dimension. That means that the algorithm cannot detect biases orthogonal to that dimension, and thus cannot subtract them off. We then cancel out the bias by subtracting multiples of the counts of the numbers of r-nonbacktracking walks of some shorter length that will also have been affected by it.

4.6 Crossing KS and the information-computation gap

4.6.1 Information-theoretic threshold

We discuss in this section SBM regimes where detection can be solved information-theoretically. As stated in Conjecture 1 and proved in Theorem 42, the information-computation gap—defined as the gap between the KS and IT thresholds—takes place when the number of communities k is larger than 4. We provide an information-theoretic (IT) bound for SSBM$(n, k, a/n, b/n)$ that confirms this, showing further that the gap grows fast with the number of communities in some regimes.

The information-theoretic bound described below is obtained by using a non-efficient algorithm that samples uniformly at random a clustering that is typical, i.e., that has the right proportions of edges inside and across the clusters. Note that to capture the exact information-theoretic threshold, one would have to rely on tighter estimates on the posterior distribution of the clsuters given the graph. A possibility is to estimate the limit of the normalized mutual information between the clusters and the graph, i.e., $\frac{1}{n} I(X; G)$, as done in Deshpande et al. (2015) for the regime of finite SNR with diverging degrees[19]—see Section 6.2. Recent results also made significant headways for the finite degree regime in the disassortative case Coja-Oghlan et al. (2016). Another possibility is to estimate the limiting total variation or KL-divergence between the graph distribution in the SBM vs. Erdős-Rényi model of matching expected degree. The limiting total variation is positive if and only if an hypothesis test can distinguish between the two models with a chance better than half. The easy implication of this is that if the total variation is vanishing, the weak recovery is not solvable (otherwise we would detect virtual clusters in the Erdős-Rényi model). This used in Banks and Moore (2016) to obtain a lower-bound on the information-theoretic threshold, using a contiguity argument, see further details at the end of this section.

19. Similar results were also obtained recently in a more general context in Caltagirone et al. (2016); Lelarge and Miolane (2016).

To obtain our information-theoretic upper-bound, we rely on the following sampling algorithm:

Typicality Sampling Algorithm. Given an n-vertex graph G and $\delta > 0$, the algorithm draws $\hat{\sigma}_{\rm typ}(G)$ uniformly at random in

$$\begin{aligned} T_\delta(G) =& \{x \in \text{Balanced}(n,k) : \\ & \sum_{i=1}^{k} |\{G_{u,v} : (u,v) \in \binom{[n]}{2} \text{ s.t. } x_u = i, x_v = i\}| \geq \frac{an}{2k}(1-\delta), \\ & \sum_{i,j\in[k], i<j} |\{G_{u,v} : (u,v) \in \binom{[n]}{2} \text{ s.t. } x_u = i, x_v = j\}| \leq \frac{bn(k-1)}{2k}(1+\delta)\}, \end{aligned}$$

where the above assumes that $a > b$; flip the above two inequalities in the case $a < b$.

The bound that is obtained below is claimed to be tight at the extremal regimes of a and b. For $b = 0$, $\text{SSBM}(n, k, a/n, 0)$ is simply a patching of disjoint Erdős-Rényi random graph, and thus the information-theoretic threshold corresponds to the giant component threshold, i.e., $a > k$, achieved by separating the giants. This breaks down for b positive, however small, but we expect that the bound derived below remains tight in the scaling of small b. For $a = 0$, the problem corresponds to planted coloring, which is already challenging Alon and Kahale (1997). The bound obtained below gives in this case that detection is information-theoretically solvable if $b > ck \ln k + o_k(1)$, $c \in [1, 2]$. This scaling is further shown to be tight in Banks and Moore (2016), which also provides a simple upper-bound that scales as $k \ln k$ for $a = 0$. Overall, the bound below shows that the KS threshold gives a much more restrictive regime than what is possible information-theoretically, as the latter reads $b > k(k-1)$ for $a = 0$.

Theorem 43 *Let $d := \frac{a+(k-1)b}{k}$, assume $d > 1$, and let $\tau = \tau_d$ be the unique solution in $(0,1)$ of $\tau e^{-\tau} = de^{-d}$, i.e., $\tau = \sum_{j=1}^{+\infty} \frac{j^{j-1}}{j!}(de^{-d})^j$. The Typicality Sampling Algorithm detects*[20] *communities in $\text{SSBM}(n, k, a/n, b/n)$ if*

$$\frac{a\ln a + (k-1)b\ln b}{k} - \frac{a+(k-1)b}{k}\ln\frac{a+(k-1)b}{k} \tag{85}$$

$$> \min\left(\frac{1-\tau}{1-\tau k/(a+(k-1)b)}2\ln(k), 2\ln(k) - 2\ln(2)e^{-a/k}(1-(1-e^{-b/k})^{k-1})\right). \tag{86}$$

This bound strictly improves on the KS threshold for $k \geq 4$:

Corollary 44 *Conjecture 1 part (ii) holds.*

See Abbe and Sandon (2017) for a numerical example. Note that (86) simplifies to

$$\frac{1}{2\ln k}\left(\frac{a\ln a + (k-1)b\ln b}{k} - d\ln d\right) > \frac{1-\tau}{1-\tau/d} =: f(\tau, d), \tag{87}$$

20. Setting $\delta > 0$ small enough gives the existence of $\varepsilon > 0$ for detection.

and since $f(\tau, d) < 1$ when $d > 1$ (which is needed for the presence of the giant), detection is already solvable in $\text{SBM}(n, k, a, b)$ if

$$\frac{1}{2\ln k}\left(\frac{a\ln a + (k-1)b\ln b}{k} - d\ln d\right) > 1. \tag{88}$$

The above bound[21] corresponds to the regime where there is no bad clustering that is typical with high probability. However, the above bound is not tight in the extreme regime of $b = 0$, since it reads $a > 2k$ as opposed to $a > k$, and it only crosses the KS threshold at $k = 5$. Before explaining how to obtain tight interpolations, we provide further insight on the bound of Theorem 43.

Defining $a_k(b)$ as the unique solution of

$$\frac{1}{2\ln k}\left(\frac{a\ln a + (k-1)b\ln b}{k} - d\ln d\right) \tag{89}$$

$$= \min\left(f(\tau, d), 1 - \frac{e^{-a/k}(1-(1-e^{-b/k})^{k-1})\ln(2)}{\ln(k)}\right) \tag{90}$$

and simplifying the bound in Theorem 43 gives the following.

Corollary 45 *Detection is solvable*

$$\text{in} \quad \text{SBM}(n,k,0,b) \quad \text{if} \quad b > \frac{2k\ln k}{(k-1)\ln\frac{k}{k-1}} f(\tau, b(k-1)/k), \tag{91}$$

$$\text{in} \quad \text{SBM}(n,k,a,b) \quad \text{if} \quad a > a_k(b), \quad \text{where } a_k(0) = k. \tag{92}$$

Remark 46 *Note that (92) approaches the optimal bound given by the presence of the giant at $b = 0$, and we further conjecture that $a_k(b)$ gives the correct first order approximation of the information-theoretic bound for small b.*

Remark 47 *Note that the k-colorability threshold for Erdős-Rényi graphs grows as $2k\ln k$ Achlioptas and Naor (2005). This may be used to obtain an information-theoretic bound, which would however be looser than the one obtained above.*

It is possible to see that this gives also the correct scaling in k for $a = 0$, i.e., that for $b < (1-\varepsilon)k\ln(k) + o_k(1)$, $\varepsilon > 0$, detection is information-theoretically impossible. To see this, consider $v \in G$, $b = (1-\epsilon)k\ln(k)$, and assume that we know the communities of all vertices more than $r = \ln(\ln(n))$ edges away from v. For each vertex r edges away from v, there will be approximately k^ϵ communities that it has no neighbors in. Then vertices $r-1$ edges away from v have approximately $k^\epsilon \ln(k)$ neighbors that are potentially in each community, with approximately $\ln(k)$ fewer neighbors suspected of being in its community than in the average other community. At that point, the noise has mostly drowned out the signal and our confidence that we know anything about the vertices' communities continues to degrade with each successive step towards v.

21. The analog of this bound in the unbalanced case already provides examples to crossing KS for two communities, such as with $p = (1/10, 9/10)$ and $Q = (0, 81; 81, 72)$.

A different approach is developed in Banks and Moore (2016) to prove that the scaling in k is in fact optimal, obtaining both upper and lower bounds on the information-theoretic threshold that match in the regime of large k when $(a-b)/d = O(1)$. In terms of the expected degree, the threshold reads as follows.

Theorem 48 *Banks and Moore (2016); Banks et al. (2016) When $(a-b)/d = O(1)$, the critical value of d satisfies $d = \Theta\left(\frac{d^2 k \log k}{(a-b)^2}\right)$, i.e., $\mathrm{SNR} = \Theta(\log(k)/k)$.*

The upper-bound in Banks and Moore (2016) corresponds essentially to (88), the regime in which the first moment bound is vanishing. The lower-bound is based on a contiguity argument and second moment estimates from Achlioptas and Naor (2005). The idea is to compare the distribution of graphs drawn from the SBM, i.e.,

$$\mu_{\mathrm{SBM}}(g) := \sum_{x \in [k]^n} \mathbb{P}\{G = g | X = x\} \mathbb{P}\{X = x\} \tag{93}$$

with the distribution of graphs drawn from the Erdős-Rényi model with matching expected degree, call it μ_{ER}. If one can show that

$$\|\mu_{\mathrm{SBM}} - \mu_{\mathrm{ER}}\|_1 \to 0, \tag{94}$$

then upon observing a graph drawn from either of the two models, say with probability half for each, it is impossible to decide from which ensemble the graph is drawn with probability asymptotically greater than half. Thus it is not possible to solve weak recovery (otherwise one would detect clusters in the Erdős-Rényi model). A sufficient condition to imply (94) is to show that $\mu_{\mathrm{SBM}} \trianglelefteq \mu_{\mathrm{ER}}$, i.e., if for any sequence of event E_n such that $\mu_{\mathrm{ER}}(E_n) \to 0$, it must be that $\mu_{\mathrm{SBM}} \to 0$. In particular, μ_{SBM} and μ_{ER} are called contiguous if $\mu_{\mathrm{SBM}} \trianglelefteq \mu_{\mathrm{ER}}$ and $\mu_{\mathrm{ER}} \trianglelefteq \mu_{\mathrm{SBM}}$, but only the first of these conditions is needed here. Further, this is implied from Cauchy-Schwarz if the ratio function

$$\rho(G) := \mu_{\mathrm{SBM}}(G)/\mu_{\mathrm{ER}}(G)$$

has a bounded second moment, i.e., $\mathbb{E}_{G \sim \mathrm{ER}} \rho^2(G) = O(1)$, which is shown in Banks and Moore (2016) (see also Moore (2017) for more details).

4.7 Nature of the gap

The nature of such gap phenomena can be understood from different perspectives. One interpretation comes from the behavior of belief propagation (or the cavity method).

Above the Kesten-Stigum threshold, the uniform fixed point is unstable and BP does not get attracted to it and reaches on most initialization a non-trivial solution. In particular, the ABP algorithm discussed in Section 4.5.1, which starts with a random initialization giving a mere bias of order $\sqrt{n}$ vertices towards the true partition (due to the Central Limit Theorem), is enough to make linearized BP reach a non-trivial fixed point. Below the information-theoretic threshold, the non-trivial fixed points are no longer present, and BP settles in a solution that represents a noisy-clustering, i.e., one that would also take place in the Erős-Rényi model due to the model fluctuations. In the gap region, non-trivial

fixed points are still present, but the trivial fixed points are locally stable and attracts most initializations. One could try multiple initializations until a non-trivial fixed point is reached, using for example the graph-splitting technique discussed in Section 3 to test such solutions. However, it is believed that an exponential number of initializations is needed to reach such a good solution. See Moore (2017) for further discussions.

This connects to the energy landscape of the possible clusterings: in this gap region, the non-trivial fixed-points have a very small basin of attraction, and they can only attract an exponentially small fraction of initializations. To connect to the results from Section 3 and the two-rounds procedure, there too the picture is related the energy landscape. Above the CH threshold, an almost exact solution having $n - o(n)$ correctly labeled vertices can be converted to an exact solution by degree-profiling. This is essentially saying that BP at depth 1, i.e., computing the likelihood of a vertex based on its direct neighbors, allows to reach the global maxima of the likelihood function with such a strong initialization. In other words, the BP view, or more precisely understanding how accurate our initial beliefs need to be in order to amplify these to non-trivial levels based on neighbords at a given depth, is related to the landscape of the objective functions.

The gap phenomenon also admits a local manifestation in the context of ABP, having to do with the approximation discussed in Section 4.5.1, where the non-linear terms behave differently from $k = 3$ to $k = 4$ due to the loss of a diminishing return property. Understanding better such gap phenomena is an active research area.

4.7.1 Proof technique for crossing KS

We explain in this section how to obtain the bound in Theorem 43. A first question is to estimate the likelihood that a bad clustering, i.e., one that has an overlap close to $1/k$ with the true clustering, belongs to the typical set. As clusters sampled from the TS algorithm are balanced, a bad clustering must split each cluster roughly into k balanced subgroups that belong to each community, see Figure 6. It is thus unlikely to keep the right proportions of edges inside and across the clusters, but depending on the exponent of this rare event, and since there are exponentially many bad clusterings, there may exist one bad clustering that looks typical.

As illustrated in Figure 6, the number of edges that are contained in the clusters of a bad clustering is roughly distributed as the sum of two Binomial random variables,

$$E_{\mathrm{in}} \dot{\sim} \mathrm{Bin}\left(\frac{n^2}{2k^2}, \frac{a}{n}\right) + \mathrm{Bin}\left(\frac{(k-1)n^2}{2k^2}, \frac{b}{n}\right),$$

where we use $\dot{\sim}$ to emphasize that this is an approximation that ignores the fact that the clustering is not exactly bad and exactly balanced. Note that the expectation of the above distribution is $\frac{n}{2k}\frac{a+(k-1)b}{k}$. In contrast, the true clustering would have a distribution given by $\mathrm{Bin}(\frac{n^2}{2k}, \frac{a}{n})$, which would give an expectation of $\frac{an}{2k}$. In turn, the number of edges that are crossing the clusters of a bad clustering is roughly distributed as

$$E_{\mathrm{out}} \dot{\sim} \mathrm{Bin}\left(\frac{n^2(k-1)}{2k^2}, \frac{a}{n}\right) + \mathrm{Bin}\left(\frac{n^2(k-1)^2}{2k^2}, \frac{b}{n}\right),$$

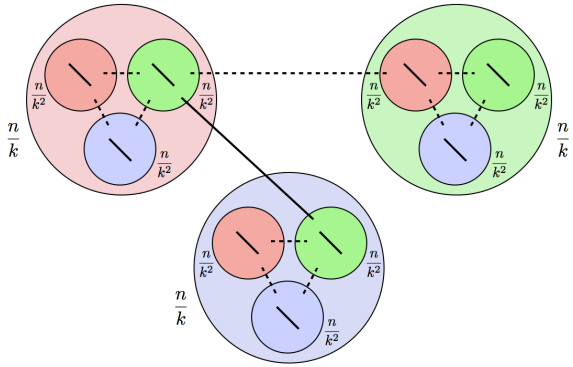


Figure 6: A bad clustering roughly splits each community equally among the k communities. Each pair of nodes connects with probability a/n among vertices of same communities (i.e., same color groups, plain line connections), and b/n across communities (i.e., different color groups, dashed line connections). Only some connections are displayed in the Figure to ease the visualization.

which has an expectation of $\frac{n(k-1)}{2k}\frac{a+(k-1)b}{k}$. In contrast, the true clustering would have the above replaced by $\mathrm{Bin}(\frac{n^2(k-1)}{2k}, \frac{b}{n})$, and an expectation of $\frac{bn(k-1)}{2k}$.

Thus, we need to estimate the rare event that the Binomial sum deviates from its expectations. While there is a large list of bounds on Binomial tail events, the number of trials here is quadratic in n and the success bias decays linearly in n, which require particular care to ensure tight bounds. We derive these in Abbe and Sandon (2015), obtaining that $\mathbb{P}\{x_{\mathrm{bad}} \in T_\delta(G) | x_{\mathrm{bad}} \in B_\epsilon\}$ behaves when ε, δ are arbitrarily small as

$$\exp\left(-\frac{n}{k}A\right)$$

where $A := \frac{a+b(k-1)}{2}\ln\frac{k}{a+(k-1)b} + \frac{a}{2}\ln a + \frac{b(k-1)}{2}\ln b$. One can then use the fact that $|T_\delta(G)| \geq 1$ with high probability, since the planted clustering is typical with high probability, and using a union bound and the fact that there are at most k^n bad clusterings:

$$\begin{aligned} P\{\hat{X}(G) \in B_\epsilon\} &= E_G \frac{|T_\delta(G) \cap B_\epsilon|}{|T_\delta(G)|} && (95)\\ &\leq E_G |T_\delta(G) \cap B_\epsilon| + o(1) && (96)\\ &\leq k^n \cdot \mathbb{P}\{x_{\mathrm{bad}} \in T_\delta(G) | x_{\mathrm{bad}} \in B_\epsilon\} + o(1). \end{aligned}$$

Checking when the above upper-bound vanishes already gives a regime that crosses the KS threshold when $k \geq 5$, and scales properly in k when $a = 0$. However, it does not interpolate the correct behavior of the information-theoretic bound in the extreme regime of $b = 0$ and does not cross at $k = 4$. In fact, for $b = 0$, the union bound requires $a > 2k$ to imply no bad typical clustering with high probability, whereas as soon as $a > k$, an algorithm that simply separates the two giants in $\mathrm{SBM}(n, k, a, 0)$ and assigns communities uniformly at random for the other vertices solves detection. Thus when $a \in (k, 2k]$, the union bound is loose. To remediate to this, we next take into account the topology of the SBM graph to tighten our bound on $|T_\delta(G)|$.

Since the algorithm samples a typical clustering, we only need the number of bad and typical clusterings to be small compared to the total number of typical clusterings, in expectation. Namely, we can get a tighter bound on the probability of error of the TS algorithm by obtaining a tighter bound on the typical set size than simply 1, i.e., estimating (95) without relying on the loose bound from (96). We proceed here with three level of refinements to bound the typical set size. In each level, we construct a random labelling of the vertices that maintain the planted labelling a typical one, and then use entropic estimates to count the number of such typical labellings.

First we exploit the large fraction of nodes that are in tree-like components outside of the giant. Conditioned on being on a tree, the SBM labels are distributed as in a broadcasting problem on a Galton-Watson tree—see Section 4.2. Specifically, for a uniformly drawn root node X, each edge in the tree acts as a k-ary symmetric channel. Thus, labelling the nodes in the trees according to the above distribution and freezing the giant to the correct labels leads to a typical clustering with high probability. The resulting bound matches the giant component bound at $b = 0$, but is unlikely to scale properly for small b. To improve on this, we next take into account the vertices in the giant that belong to planted trees, and follow the same program as above, except that the root node (in the giant) is now frozen to the correct label rather than being uniformly drawn. This gives a bound that we claim is tight at the first order approximation when b is small. Finally, we also take into account vertices that are not saturated, i.e., whose neighbors do not cover all communities and who can thus be swapped without affecting typicality. The final bound allows to cross at $k = 4$.

5. Almost Exact Recovery

5.1 Regimes

Almost exact recovery, also called weak consistency in the statistics literature, or strong recovery, has been investigated in various papers such as Yun and Proutiere (2014); Amini and Levina (2014); Gao et al. (2015); Mossel et al. (2014b); Yun and Proutiere (2014); Abbe and Sandon (2015b).

In the symmetric case, necessary and sufficient conditions have been identified.

Theorem 49 *Almost exact recovery is solvable in* $\mathrm{SSBM}(n, k, a_n/n, b_n/n)$ *if and only if*

$$\frac{(a_n - b_n)^2}{k(a_n + (k-1)b_n)} = \omega(1). \qquad (97)$$

This result appeared in several papers. A first appearance is from Yun and Proutiere (2014) where it results from the case of non-adaptive random samples, also from Mossel et al. (2014b) for $k = 2$, and from Abbe and Sandon (2015b) for $k \geq 2$. For the general

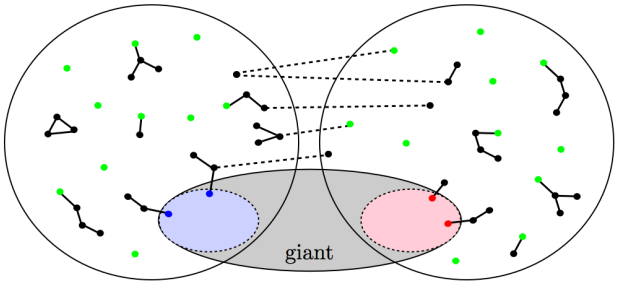


Figure 7: Illustration of the topology of SBM(n,k,a,b) for $k=2$. A giant component covering the two communities takes place when $d=\frac{a+(k-1)b}{k}>1$; a linear fraction of vertices belong to isolated trees (including isolated vertices), and a linear fraction of vertices in the giant are on planted trees. The following is used to estimate the size of the typical set in Abbe and Sandon (2017). For isolated trees, sample a bit uniformly at random for a vertex (green vertices) and propagate the bit according to the symmetric channel with flip probability $b/(a+(k-1)b)$ (plain edges do not flip whereas dashed edges flip). For planted trees, do the same but freeze the root bit to its true value.

SBM(n,p,W), a natural extension of the above statement would be to require

$$\frac{n\lambda_2(\mathrm{diag}(p)W)^2}{\lambda_1(\mathrm{diag}(p)W)}=\omega(1). \tag{98}$$

This is shown to imply almost exact recovery with additional requirements on the SBM in Abbe and Sandon (2015b), but it is unlikely to give a necessary and sufficient condition in the general case. It remains thus open to characterize in great generality when almost exact recovery is solvable or not.

5.2 Algorithms and proof techniques

The following result gives an achievability result that applies to the general SBM in the regime where $W=\omega(1)Q$, which is particularly important for the results on exact recovery discussed in Section 3.

Theorem 50 *Abbe and Sandon (2015b) For any $k\in\mathbb{Z}$, $p\in(0,1)^k$ with $|p|=1$, and symmetric matrix Q with no two rows equal, there exist $\epsilon(c)=O(1/\ln(c))$ such that for all sufficiently large c, it is possible to detect communities in SBM$(n,p,cQ/n)$ with accuracy $1-e^{-\Omega(c)}$ and complexity $O_n(n^{1+\epsilon(c)})$. In particular, almost exact recovery is solvable efficiently in SBM$(n,p,\omega(1)Q/n)$.*

Note that the exponential scaling in c above is optimal. The optimal constant in the exponent is obtained in Gao et al. (2015) for symmetric SBMs, and the optimal expression beyond the exponent is obtained in Deshpande et al. (2015); Mossel and Xu (2015) for a specific regime. We do not cover in details the algorithms for almost exact recovery, as these can be seen as a byproduct of the algorithms discussed for weak recovery in previous section, which all achieve almost exact recovery when the signal-to-noise ratio diverges. On the other hand, weak recovery requires additional sophistication that are not necessary for almost exact recovery.

A simpler yet efficient and general algorithm that allows for almost exact recovery, and used in Theorem 50 above, is the Sphere-comparison Algorithm described next. The idea is to compare neighborhoods of vertices at a given depth in the graph, to decide whether two vertices are in the same community or not. This algorithm has desirable features for real data implementations, as it seems to handle well certain types of degree variations and cliques. To ease the presentation of the algorithm's idea, we consider the symmetric case SSBM$(n,k,a/n,b/n)$ and let $d:=(a+(k-1)b)/k$ be the average degree.

Definition 51 *For any vertex v, let $N_{r[G]}(v)$ be the set of all vertices with shortest path in G to v of length r. We often drop the subscript G if the graph in question is the original SBM.*

For an arbitrary vertex v and reasonably small r, there will be typically about d^r vertices in $N_r(v)$, and about $(\frac{a-b}{k})^r$ more of them will be in v's community than in each other community. Of course, this only holds when $r<\log n/\log d$ because there are not enough vertices in the graph otherwise. The obvious way to try to determine whether or not two vertices v and v' are in the same community is to guess that they are in the same community if $|N_r(v)\cap N_r(v')|>d^{2r}/n$ and different communities otherwise. Unfortunately, whether or not a vertex is in $N_r(v)$ is not independent of whether or not it is in $N_r(v')$, which compromises this plan. Instead, we propose to rely again on a *graph-splitting* step: Randomly assign every edge in G to some set E with a fixed probability c and then count the number of edges in E that connect $N_{r[G\backslash E]}$ and $N_{r'[G\backslash E]}$. Formally:

Definition 52 *For any $v,v'\in G$, $r,r'\in\mathbb{Z}$, and subset of G's edges E, let $N_{r,r'[E]}(v\cdot v')$ be the number of pairs (v_1,v_2) such that $v_1\in N_{r[G\backslash E]}(v)$, $v_2\in N_{r'[G\backslash E]}(v')$, and $(v_1,v_2)\in E$.*

Note that E and $G\backslash E$ are disjoint. However, G is sparse enough that the two graphs can be treated as independent for the reasons discussed in Section 3.2.2. Thus, given v,r, and denoting by $\lambda_1=(a+(k-1)b)/k$ and $\lambda_2=(a-b)/k$ the two eigenvalues of PQ in the symmetric case, the expected number of intra-community neighbors at depth r from v is approximately $\frac{1}{k}(\lambda_1^r+(k-1)\lambda_2^r)$, whereas the expected number of extra-community neighbors at depth r from v is approximately $\frac{1}{k}(\lambda_1^r-\lambda_2^r)$ for each of the other $(k-1)$ communities. All of these are scaled by $1-c$ if we do the computations in $G\backslash E$. Using now the emulated independence between E and $G\backslash E$, and assuming v and v' to be in the same community, the expected number of edges in E connecting $N_{r[G\backslash E]}(v)$ to $N_{r'[G\backslash E]}(v')$ is approximately given by the inner product

$$u^t(cPQ)u,$$

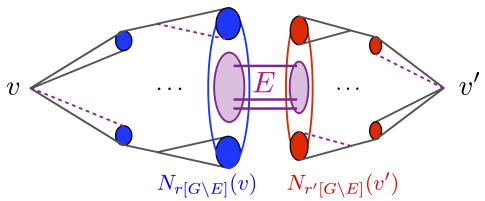


Figure 8: Sphere comparison: The algorithm takes a graph-splitting of the graph with a constant probability, and decides whether two vertices are in the same community or not based on the number of crossing edges (in the first graph of the graph-split) between the two neighborhoods' spheres at a given depth of each vertices (in the second graph of the graph-split). A careful choice of r, r' allows to reduce the complexity of the algorithm, but in general, $r = r' = \frac{3}{4}\log n/\log d$ suffices for the algorithm to succeed (where d is the average degree).

where

$$u = \frac{1}{k}(\lambda_1^r + (k-1)\lambda_2^r, \lambda_1^r - \lambda_2^r, \ldots, \lambda_1^r - \lambda_2^r)$$

and (PQ) is the matrix with $a/2$ on the diagonal and $b/2$ elsewhere. When v and v' are in different communities, the inner product is instead between u and a permutation of u that moves the first component. After simplifications, this gives

$$N_{r,r'[E]}(v \cdot v') \approx \frac{c(1-c)^{r+r'}}{n}\left[d^{r+r'+1} + \left(\frac{a-b}{k}\right)^{r+r'+1}(k\delta_{\sigma_v,\sigma_{v'}} - 1)\right] \tag{99}$$

where $\delta_{\sigma_v,\sigma_{v'}}$ is 1 if v and v' are in the same community and 0 otherwise, and where $\approx$ means that the right hand side is with high probability the dominant term of the left hand side. In order for $N_{r,r'[E]}(v \cdot v')$ to depend on the relative communities of v and v', it must be that $c(1-c)^{r+r'}|\frac{a-b}{k}|^{r+r'+1}k$ is large enough, i.e., more than n, so $r + r'$ needs to be at least $\log n/\log|\frac{a-b}{k}|$. This can then be used as a basis of the algorithm to decide whether pairs of vertices are in the same community or not, and thus to recover communities. As we shall see in Section 7.1, this approach can also be adapted to work without knowledge of the model parameters.

We conclude this section by noting that one can also study more specific almost exact recovery requirements, allowing for a specified number of misclassified vertices $s(n)$. This is investigated in Yun and Proutiere (2015) when $s(n)$ is moderately small (at most logarithmic), with an extension of Theorem 50 that applies to this more general setting. The case where $s(n)$ is linear, i.e., a constant fraction of errors, is more challenging and discussed in the next section.

6. Partial Recovery

Recall that partial recovery refers to the a fraction of misclassified vertices that is constant, whereas previous section investigates a fraction of misclassified vertices that is vanishing.

6.1 Regimes

In the symmetric SSBM$(n, k, a/n, b/n)$, the regime for partial recovery takes place when the following notion of SNR is finite:

$$\text{SNR} := \frac{(a-b)^2}{k(a+(k-1)b)} = O(1). \tag{100}$$

This regime takes place under two circumstances:

A. If a, b are constant, i.e., the constant degree regime,

B. If a, b are functions of n that diverge such that the numerator and denominator in SNR scale proportionally.

Our main goal is to identify the optimal tradeoff between SNR and the fraction of misclassified vertices, or between SNR and the MMSE or entropy of the clusters. The latter has in particular application to the compression of graphs Abbe (2016). We first mention some bounds.

Upper bounds on the fraction of incorrectly recovered vertices were demonstrated, among others, in Abbe and Sandon (2015b); Yun and Proutiere (2014); Chin et al. (2015). As detailed in Theorem 50, Abbe and Sandon (2015b) provides a bound of the type $C\exp(-c\text{SNR})$ for the general SBM. A refined bound that applies to the general SBM with arbitrary connectivity matrix $W = Q/n$ is also provided in Abbe and Sandon (2015b). In Yun and Proutiere (2014), a spectral algorithm is shown to reach an upper bounded of $C\exp\{-\text{SNR}/2\}$ for the two symmetric case, and in a suitable asymptotic sense. An upper bound of the form $C\exp(-\text{SNR}/4.1)$—again for a spectral algorithm—was obtained earlier in Chin et al. (2015). Further, Gao et al. (2015) also establishes minimax optimal rate of $C\exp\{-\text{SNR}/2\}$ in the case of large SNR and for certain types of SBMs, further handling a growing number of communities (to the expense of looser bounds).

It was shown in Mossel et al. (2013) that for the $k = 2$ symmetric case, when the SNR is sufficiently large, the optimal fraction of nodes that can be recovered is determined by the broadcasting problem on tree Evans et al. (2000) and achieved by a variant of belief propagation. That is, the probability of recovering the bit correctly from the leaves at large depth allows to determine the fraction of nodes that can be correctly labeled in the SBM in this regime. It remains open to establish such a result at arbitrary finite SNR.

We next describe a result that gives for the two-symmetric SBM the exact expression of the optimal tradeoffs between SNR and the MMSE (or the mutual information) of the clusters in the B regime at all finite SNR, and a tight approximation in the A regime for large degrees.

6.2 Distortion-SNR tradeoff

For $(X,G) \sim \mathrm{SSBM}(n,2,p_n,q_n)$, the mutual information of the SBM is $I(X;G)$, where

$$I(X;G) = H(G) - H(G|X) = H(X) - H(X|G),$$

and H denotes the entropy. We next introduce the normalized MMSE of the SBM:

$$\mathsf{MMSE}_n(\mathrm{SNR}) \equiv \frac{1}{n(n-1)}\mathbb{E}\Big\{\big\|XX^{\mathsf{T}} - \mathbb{E}\{XX^{\mathsf{T}}|G\}\big\|_F^2\Big\}. \tag{101}$$

$$= \min_{\widehat{x}_{12}:\mathcal{G}_n\to\mathbb{R}} \mathbb{E}\big\{\big[X_1X_2 - \widehat{x}_{12}(G)\big]^2\big\}. \tag{102}$$

To state our result that provides a single-letter characterization of the per-vertex MMSE (or mutual information), we need to introduce the *effective Gaussian scalar channel*. Namely, define the Gaussian channel

$$Y_0 = Y_0(\gamma) = \sqrt{\gamma}\, X_0 + Z_0, \tag{103}$$

where $X_0 \sim \mathsf{Unif}(\{+1,-1\})$ independent[22] of $Z_0 \sim \mathsf{N}(0,1)$. We denote by $\mathsf{mmse}(\gamma)$ and $\mathsf{I}(\gamma)$ the corresponding minimum mean square error and mutual information:

$$\mathsf{I}(\gamma) = \mathbb{E}\,\log\Big\{\frac{\mathrm{d}p_{Y|X}(Y_0(\gamma)|X_0)}{\mathrm{d}p_Y(Y_0(\gamma))}\Big\}, \tag{104}$$

$$\mathsf{mmse}(\gamma) = \mathbb{E}\big\{(X_0 - \mathbb{E}\{X_0|Y_0(\gamma)\})^2\big\}. \tag{105}$$

Note that these quantities can be written explicitly as Gaussian integrals of elementary functions:

$$\mathsf{I}(\gamma) = \gamma - \mathbb{E}\log\cosh\big(\gamma + \sqrt{\gamma}\,Z_0\big), \tag{106}$$

$$\mathsf{mmse}(\gamma) = 1 - \mathbb{E}\big\{\tanh(\gamma + \sqrt{\gamma}\,Z_0)^2\big\}. \tag{107}$$

We are now in position to state the result.

Theorem 53 *Deshpande et al. (2015) For any $\lambda > 0$, let $\gamma_* = \gamma_*(\lambda)$ be the largest non-negative solution of the equation*

$$\gamma = \lambda\big(1 - \mathsf{mmse}(\gamma)\big) \tag{108}$$

and

$$\Psi(\gamma,\lambda) = \frac{\lambda}{4} + \frac{\gamma^2}{4\lambda} - \frac{\gamma}{2} + \mathsf{I}(\gamma). \tag{109}$$

Let $(X,G) \sim \mathrm{SSBM}(n,2,p_n,q_n)$ and define[23] *$\mathrm{SNR} := n\,(p_n-q_n)^2/(2(p_n+q_n)(1-(p_n+q_n)/2))$. Assume that, as $n\to\infty$, (i) $\mathrm{SNR}\to\lambda$ and (ii) $n(p_n+q_n)/2(1-(p_n+q_n)/2)\to\infty$.*

22. Throughout the paper, we will generally denote scalar equivalents of vector/matrix quantities with the 0 subscript

23. Note that this is asymptotically the same notion of SNR as defined earlier when p_n, q_n vanish.

Then,

$$\lim_{n\to\infty} \mathsf{MMSE}_n(\mathrm{SNR}) = 1 - \frac{\gamma_*(\lambda)^2}{\lambda^2} \tag{110}$$

$$\lim_{n\to\infty} \frac{1}{n} I(X;G) = \Psi(\gamma_*(\lambda),\lambda). \tag{111}$$

Further, this implies $\lim_{n\to\infty}\mathsf{MMSE}_n(\mathrm{SNR}) = 1$ for $\lambda \le 1$ (i.e., no meaningful detection) and $\lim_{n\to\infty}\mathsf{MMSE}_n(\mathrm{SNR}) < 1$ for $\lambda > 1$ (detection achieved).

Corollary 54 *Deshpande et al. (2015) When $p_n = a/n, q_n = b/n$, where a,b are bounded as n diverges, there exists an absolute constant C such that*

$$\limsup_{n\to\infty}\Big|\frac{1}{n}I(X;G) - \Psi(\gamma_*(\lambda),\lambda)\Big| \le \frac{C\lambda^{3/2}}{\sqrt{a+b}}. \tag{112}$$

Here λ, $\psi(\gamma,\lambda)$ and $\gamma_*(\lambda)$ are as in Theorem 53.

A few remarks about previous theorem and corollary:

- Theorem 53 shows that the normalized MMSE (or mutual information) is non-trivial if and only if $\lambda > 1$. This extends the results on weak recovery Massoulié (2014); Mossel et al. (2015) discussed in Section 4 from the A to the B regime for finite SNR, closing weak recovery in the SSBM with two communities;

- The result also gives upper and lower bound for the optimal agreement. Let

  $$\mathsf{Overlap}_n(\mathrm{SNR}) = \frac{1}{n}\sup_{\widehat{\boldsymbol{s}}:\mathcal{G}_n\to\{+1,-1\}^n}\mathbb{E}\big\{|\langle X,\widehat{\boldsymbol{s}}(G)\rangle|\big\}.$$

  Then,

  $$1 - \mathsf{MMSE}_n(\mathrm{SNR}) + O(n^{-1}) \le \mathsf{Overlap}_n(\mathrm{SNR}) \tag{113}$$

  $$\le \sqrt{1 - \mathsf{MMSE}_n(\mathrm{SNR})} + O(n^{-1/2}). \tag{114}$$

- In Mossel and Xu (2015), tight expressions similar to those obtained in Theorem 53 for the MMSE are obtained for the optimal expected agreement with additional scaling requirements. Namely, it is shown that for $\mathrm{SSBM}(n,2,a/n,b/n)$ with $a = b + \mu\sqrt{b}$ and $b = o(\log n)$, the least fraction of misclassified vertices is in expectation given by $Q(\sqrt{v^*})$ where v^* is the unique fixed point of the equation $v = \frac{\mu^2}{4}\mathbb{E}\tanh(v + v\sqrt{Z})$, Z is normal distributed, and Q is the Q-function for the normal distribution. Similar results were also reported in Zhang et al. (2016) for the overlap metric, and Lesieur et al. (2015) for the MMSE.

- Note that Theorem 53 requires merely diverging degrees (arbitrarily slowly), in contrast to results from random matrix theory such as Baik et al. (2005) that would require poly-logarithmic degrees to extract communities from the spiked Wigner model.

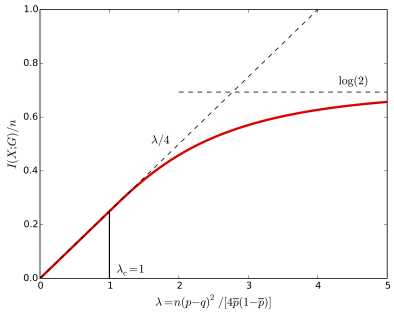


Figure 9: Asymptotic mutual information per vertex of the symmetric stochastic block model with two communities, as a function of the signal-to-noise ratio λ. The dashed lines are simple upper bounds: $\lim_{n\to\infty} I(X;G)/n \le \lambda/4$ and $I(X;G)/n \le \log 2$.

6.3 Proof technique and spiked Wigner model

Theorem 53 gives an exact expression for the normalized MMSE and mutual information in terms of an effective Gaussian noise channel. The reason for the Gaussian distribution to emerge is that the proof of the result shows as a side result that in the regime of the theorem, the SBM model is equivalent to a spiked Wigner model given by

$$Y = \sqrt{\lambda/n} X X^t + Z$$

where Z is a Wigner random matrix (i.e., symmetric with i.i.d. Normal entries), and where we recall that λ corresponds to the limit of SNR.

In other words, the per-dimension mutual information turns out to be *universal* across multiple noise models and does not depend on the microscopic details of the noise distribution, but only on the first two moments, i.e., the SNR in our case. The formal statement of the equivalence is as follows:

Theorem 55 (Equivalence of SBM and Gaussian models) *Let $I(X;G)$ be the mutual information of* SSBM$(n,2,p_n,q_n)$ *with* SNR $\to \lambda$ *and $n(p_n+q_n)/2(1-(p_n+q_n)/2) \to \infty$, and $I(X;Y)$ be the mutual information for spiked Wigner model $Y = \sqrt{\lambda/n} X X^t + Z$. Then, there is a constant C independent of n such that*

$$\frac{1}{n}\big|I(X;G) - I(X;Y)\big| \le C\left(\frac{\lambda^{3/2}}{\sqrt{n(p_n+q_n)/2(1-(p_n+q_n)/2)}} + |\mathrm{SNR}-\lambda|\right). \qquad (115)$$

To obtain the limiting expression for the normalized mutual information in Theorem 53, notice first that for $Y(\lambda) = \sqrt{\lambda/n} X X^t + Z$,

$$\frac{1}{n} I(X;Y(0)) = 0 \qquad \frac{1}{n} I(X;Y(\infty)) = \log(2).$$

Next, (i) use the fundamental theorem of calculus to express these boundary conditions as an integral of the derivative of the mutual information, (ii) the I-MMSE identity Guo et al. (2005) to express this derivative in terms of the MMSE, (iii) upper-bound the MMSE with error with the specific estimate obtained from the AMP algorithm Donoho et al. (2009), (iv) evaluate the asymptotic performance of the AMP estimate using the density evolution technique Bayati and Montanari (2011); Deshpande and Montanari (2014), and (v) notice that this matches the original value of $\log(2)$ in the limit of n tending to infinty:

$$\log(2) \overset{(i)}{=} \frac{1}{n}\int_0^\infty \frac{\partial}{\partial\lambda} I(XX^t; Y(\lambda))\, d\lambda \qquad (116)$$

$$\overset{(ii)}{=} \frac{1}{4n^2}\int_0^\infty \mathrm{MMSE}(XX^t|Y(\lambda))\, d\lambda \qquad (117)$$

$$\overset{(iii)}{\le} \frac{1}{4n^2}\int_0^\infty \mathbb{E}(XX^t - \hat{x}_{\mathrm{AMP},\lambda}(\infty)\hat{x}^t_{\mathrm{AMP},\lambda}(\infty))^2\, d\lambda \qquad (118)$$

$$\overset{(iv)}{=} \Psi(\gamma_*(\infty),\infty) - \Psi(\gamma_*(0),0) + o_n(1) \qquad (119)$$

$$\overset{(v)}{=} \log(2) + o_n(1). \qquad (120)$$

This implies that (iii) is in fact an equality asymptotically, and using monotonicity and continuity properties of the integrant, the identify must hold for all SNR as stated in the theorem. The only caveat not discussed here is the fact that AMP needs an initialization that is not fully symmetric to converge to the right solution, which causes the insertion in the proof of a noisy observation on the true labels X at the channel output to break the symmetry for AMP (removing then this information by taking a limit).

6.4 Optimal detection for constant degrees

Obtaining the expression for the optimal agreement at finite SNR when the degrees are constant remains an open problem (see also Sections 5 and 8). The problem is settled for high enough SNR in Mossel et al. (2013), with the following expression relying on reconstruction error for the broadcasting on tree problem.

Define the optimal agreement fraction as

$$P_{G_n}(a,b) := \frac{1}{2} + \sup_f \mathbb{E}\Big|\frac{1}{n}\sum_v \mathbb{1}(f(v,G_n) = X_v) - \frac{1}{2}\Big|. \qquad (121)$$

Note that the above expression takes into account the symmetry of the problem and can also been interpreted as a normalized agreement or probability. Let $P_G(a,b) := \limsup_g P_{G_n}(a,b)$. Define now the counter-part for the broadcasting problem on tree: Back to the notation of Section 4.2, define $T^{(t)}$ as the Galton-Watson tree with Poisson$((a+b)/2)$ offspring, flip probability $b/(a+b)$ and depth t, and define the optimal inference probability of the root as

$$P_T(a,b) := \frac{1}{2} + \lim_{t\to\infty} \mathbb{E}\big|\mathbb{E}(X^{(0)}|X^{(t)}) - 1/2\big|. \qquad (122)$$

The reduction from Mossel et al. (2015) discussed in Section 4.2 allows to deduce that $P_G(a,b) \le P_T(a,b)$, and this is conjectured to be an equality, as shown for large enough SNR:

Theorem 56 *Mossel et al. (2013) There exists C large enough such that if* $\mathrm{SNR} > C$ *then* $P_G(a,b) = P_T(a,b)$*, and this normalized agreement is efficiently achievable.*

The theorem in Mossel et al. (2013) has a weak requirement of $\mathrm{SNR} > C\log(a+b)$, but later developments allow for the improved version stated above. Note that $P_T(a,b)$ gives only an implicit expression for the optimal fraction, though it admits a variational representation due to Mézard and Montanari (2006).

In Decelle et al. (2011), it is conjectured that BP gives the optimal agreement at all SNR. However, the problem with BP is the classical one, it is hard to analyze it in the context of loopy graphs with a random initialization. Another strategy here is to proceed with a two-round procedure, which is used to establish the above results in Mossel et al. (2013) for two communities, with a variant discussed in Abbe and Sandon (2016b) for multiple communities. The idea is to use a simpler algorithm to obtain a non-trivial reconstruction when $\mathrm{SNR} > 1$, see Section 4, and to then improve the accuracy using full BP at shorter depth. To show that the accuracy achieved is optimal, one has to also show that a noisy version of the reconstruction on tree problem Janson and Mossel (2004), where leaves do not have exact labels but noisy labels, leads to the same probability of error at the root. This is expected to take place for two communities at all SNR above the KS threshold, and it was shown in Mossel et al. (2013) for the case of large enough SNR. This type of claim is not expected to hold for general k. For more than two communities, one needs to convert first the output of the algorithm discussed in Section 4.5.1, which gives two sets that correlated with their communities, into a nontrivial assignment of a belief to each vertex; this is discussed in Abbe and Sandon (2017). Then one uses these beliefs as starting probabilities for a belief propagation algorithm of depth $\ln(n)/3\ln(\lambda_1)$, which runs now on a tree-like graph.

7. Learning the SBM

In this section we investigate the problem of estimating the SBM parameters by observing a one shot realization of the graph. We consider first the case where degrees are diverging, where estimation can be obtained as a side result of universal almost exact recovery, and the case of constant degrees, where estimation can be performed without being able to recover the clusters but only above the weak recovery threshold.

7.1 Diverging degree regime

For diverging degrees, one can estimate the parameters by solving first universal almost exact recovery, and proceeding then to basic estimates on the clusters' cuts and volumes. This requires solving an harder problem potentially, but turns out to be solvable as shown next:

Theorem 57 *Abbe and Sandon (2015c) Given $\delta > 0$ and for any $k \in \mathbb{Z}$, $p \in (0,1)^k$ with $\sum p_i = 1$ and $0 < \delta \leq \min p_i$, and any symmetric matrix Q with no two rows equal such that every entry in Q^k is strictly positive (in other words, Q such that there is a nonzero probability of a path between vertices in any two communities in a graph drawn from SBM$(n,p,Q/n)$), there exist $\epsilon(c) = O(1/\ln(c))$ such that for all sufficiently large α, the Agnostic-sphere-comparison algorithm detects communities in graphs drawn from SBM$(n,p,\alpha Q/n)$ with accuracy at least $1-e^{-\Omega(\alpha)}$ in $O_n(n^{1+\epsilon(\alpha)})$ time.*

Note that the knowledge on δ in this theorem can be removed if $\alpha = \omega(1)$. We then obtain:

Corollary 58 *Abbe and Sandon (2015c) The number of communities k, the community prior p and the connectivity matrix Q can be consistently estimated in quasi-linear time in SBM$(n,p,\omega(1)Q/n)$.*

Recall that in Section 5 we discussed the Sphere-comparison algorithm, where the neighborhoods at depth r and r' from two vertices are compared in order to decide whether the vertices are in the same community or not. The key statistics was the number of crossing edges (in the background graph of the graph-split) between these two neighborhoods:

$$N_{r,r'[E]}(v\cdot v') \approx \frac{c(1-c)^{r+r'}}{n}\left[d^{r+r'+1} + \left(\frac{a-b}{k}\right)^{r+r'+1}(k\delta_{\sigma_v,\sigma_{v'}}-1)\right] \tag{123}$$

where $\delta_{\sigma_v,\sigma_{v'}}$ is 1 if v and v' are in the same community and 0 otherwise. A difficulty is that for a specific pair of vertices, the $d^{r+r'+1}$ term will be multiplied by a random factor dependent on the degrees of v, v', and the nearby vertices. So, in order to stop the variation in the $d^{r+r'+1}$ term from drowning out the $\left(\frac{a-b}{k}\right)^{r+r'+1}(k\delta_{\sigma_v,\sigma_{v'}}-1)$ term, it is necessary to cancel out the dominant term. This motivates the introduction in Abbe and Sandon (2015c) of the following **sign-invariant statistics**:

$$\begin{aligned} &I_{r,r'[E]}(v\cdot v') := N_{r+2,r'[E]}(v\cdot v')\cdot N_{r,r'[E]}(v\cdot v') - N^2_{r+1,r'[E]}(v\cdot v') \\ &\approx \frac{c^2(1-c)^{2r+2r'+2}}{n^2}\cdot\left(d-\frac{a-b}{k}\right)^2\cdot d^{r+r'+1}\left(\frac{a-b}{k}\right)^{r+r'+1}(k\delta_{\sigma_v,\sigma_{v'}}-1) \end{aligned}$$

In particular, for $r+r'$ odd, $I_{r,r'[E]}(v\cdot v')$ will tend to be positive if v and v' are in the same community and negative otherwise, irrespective of the specific values of a,b,k. That suggests the following agnostic algorithm for partial recovery, which requires knowledge of $\delta < 1/k$ in the constant degree regime (i.e., an upper bound on the number communities), but not in the regime where a,b scale with n.

Note that for symmetric SBMs, SDPs Abbe et al. (2016); B. Hajek (2014); Hajek et al. (2015c); Bandeira (2015) can be used to recover the communities without knowledge of the parameters, and thus to learn the parameters in the symmetric case. A different line of work has also studied the problem of estimating 'graphons' Choi et al. (2012); Airoldi et al. (2013); Olhede and Wolfe (2014) via block models, assuming regularity conditions on the graphon, such as piecewise Lipschitz, to obtain estimation guarantees. In addition, Borgs et al. (2015) considers private graphon estimation in the logarithmic degree regime, and obtains a non-efficient procedure to estimate 'graphons' in an appropriate version of the L_2 norm. More recently, Borgs et al. (2015) extends the type of results from Wolfe and Olhede (2013) to a much more general family of 'graphons' and to sparser regimes (though still with diverging degrees) with efficient methods (based on degrees) and non-efficient methods (based on least square and least cut norm).

Agnostic-sphere-comparison. Assume knowledge of $\delta > 0$ such that $\min_{i\in[k]} p_i \geq \delta$ and let d be the average degree of the graph:

1. Set $r = r' = \frac{3}{4}\log n/\log d$ and put each of the graph's edges in E with probability $1/10$.
2. Set $k_{\max} = 1/\delta$ and select $k_{\max}\ln(4k_{\max})$ random vertices, $v_1, ..., v_{k_{\max}\ln(4k_{\max})}$.
3. Compute $I_{r,r'[E]}(v_i \cdot v_j)$ for each i and j. If there is a possible assignment of these vertices to communities such that $I_{r,r'[E]}(v_i \cdot v_j) > 0$ if and only if v_i and v_j are in the same community, then randomly select one vertex from each apparent community, $v[1], v[2], ...v[k']$. Otherwise, fail.
4. For every v' in the graph, guess that v' is in the same community as the $v[i]$ that maximizes the value of $I_{r,r'[E]}(v[i] \cdot v')$.

7.2 Constant degree regime

In the case of the constant degree regime, it is not possible to recover the clusters (let alone without knowing the parameters), and thus estimation has to be done differently. The first paper that shows how to estimate the parameter in this regime tightly is Mossel et al. (2015), which is based on approximating cycle counts by nonbacktracking walks. An alternative method based on expectation-maximization using the Bethe free energy is also proposed in Decelle et al. (2011) (without a rigorous analysis).

Theorem 59 *Mossel et al. (2015) Let $G \sim \mathrm{SSBM}(n, 2, a/n, b/n)$ such that $(a-b)^2 > 2(a+b)$, and let C_m be the number of m-cycles in G, $\hat{d}_n = 2|E(G)|/n$ be the average degree in G and $\hat{f}_n = (2m_n C_{m_n} - \hat{d}_n^{m_n})^{1/m_n}$ where $m_n = \lfloor \log^{1/4}(n) \rfloor$. Then $\hat{d}_n + \hat{f}_n$ and $\hat{d}_n - \hat{f}_n$ are consistent estimators for a and b respectively. Further, there is a polynomial time estimator to calculate $\hat{d}_n$ and $\hat{f}_n$.*

This theorem is extended in Abbe and Sandon (2015) for the symmetric SBM with k clusters, where k is also estimated. The first step needed is the following estimate.

Lemma 60 *Let C_m be the number of m-cycles in $\mathrm{SBM}(n, p, Q/n)$. If $m = o(\log\log(n))$, then*

$$\mathbb{E}C_m \sim \mathbb{V}\mathrm{ar}C_m \sim \frac{1}{2m}\mathrm{tr}(\mathrm{diag}(p)Q)^m. \tag{124}$$

To see this lemma, note that there is a cycle on a given selection of m vertices with probability

$$\sum_{x_1,...,x_m\in[k]} \frac{Q_{x_1,x_2}}{n} \cdot \frac{Q_{x_2,x_3}}{n} \cdot \ldots \cdot \frac{Q_{x_m,x_1}}{n} \cdot p_{x_1} \cdot \ldots \cdot p_{x_m} = \mathrm{tr}(\mathrm{diag}(p)Q/n)^m. \tag{125}$$

Since there are $\sim n^m/2m$ such selections, the first moment follows. The second moment follows from the fact that overlapping cycles do not contribute to the second moment. See

Mossel et al. (2015) for proof details for the 2-SSBM and Abbe and Sandon (2015) for the general SBM.

Hence, one can estimate $\frac{1}{2m}\mathrm{tr}(\mathrm{diag}(p)Q)^m$ for slowly growing m. In the symmetric SBM, this gives enough degrees of freedom to estimate the three parameters a, b, k. Theorem 59 uses for example the average degree ($m = 1$) and slowly growing cycles to obtain a system of equation that allows to solve for a, b. This extends easily to all symmetric SBMs, and the efficient part follows from the fact that for slowly growing m, the cycle counts coincides with the nonbacktracking walk counts with high probability Mossel et al. (2015). Note that Theorem 59 provides a tight condition for the estimation problem, i.e., Mossel et al. (2015) also shows that when $(a-b)^2 \leq 2(a+b)$ (which we recall is equivalent to the requirement for impossibility of weak recovery) the SBM is contiguous to the Erdős-Rényi model with edge probability $(a+b)/(2n)$.

However, for the general SBM, the problem is more delicate and one has to first stabilize the cycle count statistics to extract the eigenvalues of PQ, and use detection methods to further peal down the parameters p and Q. Deciding which parameters can or cannot be learned in the general SBM seems to be a non-trivial problem. This is also expected to come into play in the estimation of graphons Choi et al. (2012); Airoldi et al. (2013); Borgs et al. (2015).

8. Open Problems

The establishment of fundamental limits for community detection in the SBM have appeared in the recent years. There is therefore a long list of open problems and directions to pursue, both related to the SBM and to its extensions. We provide here a partial list:

- *Exact recovery for sub-linear communities.* Theorems 14 and 29 give a fairly comprehensive result for exact recovery in the case of linear-size communities, i.e., when the entries of p and its dimension k do not scale with n. If $k = o(\log(n))$, and the communities remain reasonably balanced, most of the developed techniques extend. However new phenomena seem to take place beyond this regime, with again gaps between information and computational thresholds. In Y. Chen (2014), some of this is captured by looking at coarse regimes of the parameters. It would be interesting to pursue sub-linear communities in the lens of phase transitions and information-computation gaps.
- *Partial recovery.* What is the fundamental tradeoff between the SNR and the distortion (MMSE or agreement) for partial recovery in the constant degree regime? As a preliminary result, one may attempt to show that $I(X;G)/n$ admits a limit in the constant degree regime. This is proved in Abbe and Montanari (2015) for two symmetric disassortative communities,[24] but the assortative case remains open. Establishing the expression for the optimal tradeoff in partial recovery is unknown for $k \geq 3$ (and only known for large enough SNR for $k = 2$ Mossel et al. (2013)).
- *The information-computation gap:*

24. Limiting expressions have recently been obtained for disassortative communities in Coja-Oghlan et al. (2016).

- Can we locate the exact information-theoretic threshold for weak recovery when $k \geq 3$? Recent results and precise conjectures were recently obtained in Caltagirone et al. (2016), for the regime of finite SNR with diverging degrees discussed in Section 6.2.
- Can we strengthen the evidences that the KS threshold is the computational threshold? In the general sparse SBM, this corresponds to the following conjecture:

  Conjecture 61 *Let $k \in \mathbb{Z}_+$, $p \in (0,1)^k$ be a probability distribution, Q be a $k \times k$ symmetric matrix with nonnegative entries. If $\lambda_2^2 < \lambda_1$, then there is no polynomial time algorithm that can solve weak recovery in G drawn from* $\mathrm{SBM}(n, p, Q/n)$.

- *Learning the general sparse SBM.* Under what condition can we learn the parameters in $\mathrm{SBM}(n, p, Q/n)$ efficiently or information-theoretically?
- *Scaling laws:* What is the optimal scaling/exponents of the probability of error for the various recovery requirements? How large need the graph be, i.e., what is the scaling in n, so that the probability of error in the discussed results[25] is below a given threshold?
- *Beyond the SBM:*
  - How do previous results and open problems generalize to the extensions of SBMs with labels, degree-corrections, overlaps, etc. beyond cases discussed in Section 3.5? In the related line of work for graphons Choi et al. (2012); Airoldi et al. (2013); Borgs et al. (2015), are there fundamental limits in learning the model or recovering the vertex parameters up to a given distortion? It was shown in Moitra et al. (2016); Makarychev et al. (2015) that monotone adversaries can interestingly shift the threshold for weak recovery; what is the threshold for such adversarial models and variants for adding loops?
  - Can we establish fundamental limits and algorithms achieving the limits for other unsupervised machine learning problems, such as topic modelling, ranking, Gaussian mixture clustering, low-rank matrix recovery (see Deshpande and Montanari (2014) for sparse PCA) or the graphical channels discussed in Section 1.2?
- *Semi-supervised extensions:* How do the fundamental limits change in a semi-supervised setting,[26] i.e., when some of the vertex labels are revealed, exactly or probabilistically?
- *Dynamical extensions:* In some cases, the network may be dynamical and one may observe different time instances of the network. How does one integrate such dynamics to understand community detection?[27]

25. Recent work Young et al. (2016) has investigated finite size information-theoretic analysis for detection.
26. Partial results and experiments were obtained for a semi-supervised model Zhang et al. (2014). Another setting with side-information is considered in Clauset et al. (2004) with metadata available at the network vertices.
27. Partial results were recently obtained in Ghasemian et al. (2016).

Acknowledgements

I would like to thank my collaborators from the main papers discussed in the manuscript, in particular, A. Bandeira, Y. Deshpande, G. Hall, A. Montanari, C. Sandon, the many colleagues with whom I had conservations on the topic, in particular, E. Airoldi, C. Bordenave, F. Krzakala, M. Lelarge, L. Massoulié, C. Moore, E. Mossel, A. Sly, V. Vu, L. Zdeborová, as well as the various colleagues, students and anonymous reviewers who gave useful comments on the earlier drafts.

References

E. Mossel, Private communications, 2017.

E. Abbe. Graph compression: The effect of clusters. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1–8, Sept 2016. doi: 10.1109/ALLERTON.2016.7852203.

E. Abbe and A. Montanari. Conditional random fields, planted constraint satisfaction, and entropy concentration. *Theory of Computing*, 11(17):413–443, 2015. doi: 10.4086/toc.2015.v011a017. URL `http://www.theoryofcomputing.org/articles/v011a017`.

E. Abbe and C. Sandon. Community detection in general stochastic block models: fundamental limits and efficient recovery algorithms. *arXiv:1503.00609.*, March 2015a.

E. Abbe and C. Sandon. Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 670–688, 2015b. doi: 10.1109/FOCS.2015.47. URL `http://dx.doi.org/10.1109/FOCS.2015.47`.

E. Abbe and C. Sandon. Recovering communities in the general stochastic block model without knowing the parameters. In C. Cortes, N.D. Lawrence, D.D. Lee, M. Sugiyama, R. Garnett, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NIPS) 28*, pages 676–684. Curran Associates, Inc., 2015c.

E. Abbe and C. Sandon. Detection in the stochastic block model with multiple clusters: proof of the achievability conjectures, acyclic BP, and the information-computation gap. *ArXiv e-prints 1512.09080*, December 2015.

E. Abbe and C. Sandon. Crossing the ks threshold in the stochastic block model with information theory. *In the proc. of ISIT*, 2016a.

E. Abbe and C. Sandon. Proof of the achievability conjectures in the general stochastic block model. *To Appear in Communications on Pure and Applied Mathematics*, 2017.

E. Abbe, A. S. Bandeira, and G. Hall. Exact Recovery in the Stochastic Block Model. *ArXiv e-prints*, May 2014.

E. Abbe, A.S. Bandeira, A. Bracher, and A. Singer. Decoding binary node labels from censored edge measurements: Phase transition and efficient recovery. *Network Science and Engineering, IEEE Transactions on*, 1(1):10–22, Jan 2014a. ISSN 2327-4697. doi: 10.1109/TNSE.2014.2368716.

E. Abbe, A.S. Bandeira, A. Bracher, and A. Singer. Linear inverse problems on Erdös-Rényi graphs: Information-theoretic limits and efficient recovery. In *Information Theory (ISIT), 2014 IEEE International Symposium on*, pages 1251–1255, June 2014b. doi: 10.1109/ISIT.2014.6875033.

E. Abbe, A.S. Bandeira, and G. Hall. Exact recovery in the stochastic block model. *Information Theory, IEEE Transactions on*, 62(1):471–487, Jan 2016. ISSN 0018-9448. doi: 10.1109/TIT.2015.2490670.

Emmanuel Abbe and Colin Sandon. Achieving the ks threshold in the general stochastic block model with linearized acyclic belief propagation. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1334–1342. Curran Associates, Inc., 2016b.

D. Achlioptas, A. Naor, and Y. Peres. Rigorous Location of Phase Transitions in Hard Optimization Problems. *Nature*, 435:759–764, 2005.

Dimitris Achlioptas and Assaf Naor. The two possible values of the chromatic number of a random graph. *Annals of Mathematics*, 162(3):1335–1351, 2005. ISSN 0003486X. URL http://www.jstor.org/stable/20159944.

L. Adamic and N. Glance. The political blogosphere and the 2004 u.s. election: Divided they blog. In *Proceedings of the 3rd International Workshop on Link Discovery*, LinkKDD '05, pages 36–43, New York, NY, USA, 2005. ISBN 1-59593-215-1. doi: 10.1145/1134271.1134277. URL http://doi.acm.org/10.1145/1134271.1134277.

N. Agarwal, A. S. Bandeira, K. Koiliaris, and A. Kolla. Multisection in the Stochastic Block Model using Semidefinite Programming. *ArXiv e-prints*, July 2015.

E. Airoldi, T. Costa, and S. Chan. Stochastic blockmodel approximation of a graphon: Theory and consistent estimation. *arXiv:1311.1731*, 2013.

E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.*, 9:1981–2014, June 2008. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1390681.1442798.

David J. Aldous. Representations for partially exchangeable arrays of random variables. *Journal of Multivariate Analysis*, 11(4):581 – 598, 1981. ISSN 0047-259X. doi: http://dx.doi.org/10.1016/0047-259X(81)90099-3. URL http://www.sciencedirect.com/science/article/pii/0047259X81900993.

Noga Alon and Nabil Kahale. A spectral technique for coloring random 3-colorable graphs. *SIAM Journal on Computing*, 26(6):1733–1748, 1997. doi: 10.1137/S0097539794270248. URL http://dx.doi.org/10.1137/S0097539794270248.

Noga Alon, Michael Krivelevich, and Benny Sudakov. Finding a large hidden clique in a random graph. *Random Structures and Algorithms*, 13(3-4):457–466, 1998. ISSN 1098-2418.

A. Amini and E. Levina. On semidefinite relaxations for the block model. *arXiv:1406.5647*, June 2014.

M. C. Angelini, F. Caltagirone, F. Krzakala, and L. Zdeborova. Spectral detection on sparse hypergraphs. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 66–73, Sept 2015. doi: 10.1109/ALLERTON.2015.7446987.

A. Asadi, E. Abbe, and S. Verdú. Compressing data on graphs with clusters. *In proc. of ISIT, Aachen*, 2017.

J. Xu B. Hajek, Y. Wu. Achieving exact cluster recovery threshold via semidefinite programming. *arXiv:1412.6156*, November 2014.

Jinho Baik, Grard Ben Arous, and Sandrine Pch. Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices. *Ann. Probab.*, 33(5):1643–1697, 09 2005. doi: 10.1214/009117905000000233. URL http://dx.doi.org/10.1214/009117905000000233.

Brian Ball, Brian Karrer, and M. E. J. Newman. An efficient and principled method for detecting communities in networks. *Phys. Rev. E*, 84:036103, Sep 2011a. doi: 10.1103/PhysRevE.84.036103. URL http://link.aps.org/doi/10.1103/PhysRevE.84.036103.

Brian Ball, Brian Karrer, and Mark E. J. Newman. Efficient and principled method for detecting communities in networks. *Phys. Rev. E*, 84(3):036103, 2011b.

A. S. Bandeira. Random laplacian matrices and convex relaxations. *arXiv:1504.03987*, 2015.

J. Banks and C. Moore. Information-theoretic thresholds for community detection in sparse networks. *ArXiv e-prints*, January 2016.

Jess Banks, Cristopher Moore, Joe Neeman, and Praneeth Netrapalli. Information-theoretic thresholds for community detection in sparse networks. *Proc. of COLT*, 2016.

Mohsen Bayati and Andrea Montanari. The dynamics of message passing on dense graphs, with applications to compressed sensing. *Information Theory, IEEE Transactions on*, 57 (2):764–785, 2011.

Q. Berthet and P. Rigollet. Optimal detection of sparse principal components in high dimension. *Ann. Statist.*, 41(4):1780–1815, 08 2013. doi: 10.1214/13-AOS1127. URL http://dx.doi.org/10.1214/13-AOS1127.

Q. Berthet, P. Rigollet, and P. Srivastava. Exact recovery in the Ising blockmodel. *ArXiv e-prints*, December 2016.

S. Bhattacharyya and P. J. Bickel. Community Detection in Networks using Graph Distance. *ArXiv e-prints*, January 2014.

Peter J. Bickel and Aiyou Chen. A nonparametric view of network models and newman-girvan and other modularities. *Proceedings of the National Academy of Sciences*, 106 (50):21068–21073, 2009. doi: 10.1073/pnas.0907096106. URL http://www.pnas.org/content/106/50/21068.abstract.

P. M. Bleher, J. Ruiz, and V. A. Zagrebnov. On the purity of the limiting gibbs state for the ising model on the bethe lattice. *Journal of Statistical Physics*, 79(1):473–482, 1995. doi: 10.1007/BF02179399. URL http://dx.doi.org/10.1007/BF02179399.

Béla Bollobás, Svante Janson, and Oliver Riordan. The phase transition in inhomogeneous random graphs. *Random Struct. Algorithms*, 31(1):3–122, August 2007. ISSN 1042-9832. doi: 10.1002/rsa.v31:1. URL http://dx.doi.org/10.1002/rsa.v31:1.

R.B. Boppana. Eigenvalues and graph bisection: An average-case analysis. *In 28th Annual Symposium on Foundations of Computer Science*, pages 280–285, 1987.

Charles Bordenave, Marc Lelarge, and Laurent Massoulie. Non-backtracking spectrum of random graphs: Community detection and non-regular ramanujan graphs. In *Proceedings of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, FOCS '15, pages 1347–1357, Washington, DC, USA, 2015. IEEE Computer Society. ISBN 978-1-4673-8191-8. doi: 10.1109/FOCS.2015.86. URL http://dx.doi.org/10.1109/FOCS.2015.86.

C. Borgs, J. Chayes, E. Mossel, and S. Roch. The Kesten-Stigum Reconstruction Bound Is Tight for Roughly Symmetric Binary Channels. *ArXiv Mathematics e-prints*, April 2006.

C. Borgs, J.T. Chayes, L. Lovasz, V.T. Sos, and K. Vesztergombi. Convergent sequences of dense graphs i: Subgraph frequencies, metric properties and testing. *Advances in Mathematics*, 219(6):1801 – 1851, 2008. ISSN 0001-8708. doi: http://dx.doi.org/10.1016/j.aim.2008.07.008. URL http://www.sciencedirect.com/science/article/pii/S0001870808002053.

C. Borgs, J. T. Chayes, H. Cohn, and Y. Zhao. An L^p theory of sparse graph convergence I: limits, sparse random graph models, and power law distributions. *ArXiv e-prints*, January 2014.

C. Borgs, J. T. Chayes, H. Cohn, and S. Ganguly. Consistent nonparametric estimation for heavy-tailed sparse graphs. *ArXiv e-prints*, August 2015.

Christian Borgs, Jennifer Chayes, and Adam Smith. Private graphon estimation for sparse graphs. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1369–1377. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5828-private-graphon-estimation-for-sparse-graphs.pdf.

T.N. Bui, S. Chaudhuri, F.T. Leighton, and M. Sipser. Graph bisection algorithms with good average case behavior. *Combinatorica*, 7(2):171–191, 1987. ISSN 0209-9683. doi: 10.1007/BF02579448. URL http://dx.doi.org/10.1007/BF02579448.

I. Cabreros, E. Abbe, and A. Tsirigos. Detecting Community Structures in Hi-C Genomic Data. *Conference on Information Science and Systems, Princeton University. ArXiv e-prints 1509.05121*, September 2015.

Francesco Caltagirone, Marc Lelarge, and Léo Miolane. Recovering asymmetric communities in the stochastic block model. *Allerton*, 2016.

J. Chen and B. Yuan. Detecting functional modules in the yeast protein?protein interaction network. *Bioinformatics*, 22(18):2283–2290, 2006. doi: 10.1093/bioinformatics/btl370.

Y. Chen and A. J. Goldsmith. Information recovery from pairwise measurements. *In Proc. ISIT, Honolulu.*, 2014.

Y. Chen, Q.-X. Huang, and L. Guibas. Near-optimal joint object matching via convex relaxation. *Available Online: arXiv:1402.1473 [cs.LG]*, 2014.

P. Chin, A. Rao, and V. Vu. Stochastic block model and community detection in the sparse graphs: A spectral algorithm with optimal rate of recovery. *arXiv:1501.05021*, January 2015.

D. S. Choi, P. J. Wolfe, and E. M. Airoldi. Stochastic blockmodels with a growing number of classes. *Biometrika*, pages 1–12, 2012. doi: 10.1093/biomet/asr053.

Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70:066111, Dec 2004. doi: 10.1103/PhysRevE.70.066111. URL http://link.aps.org/doi/10.1103/PhysRevE.70.066111.

M.S. Cline, M. Smoot, E. Cerami, A. Kuchinsky, N. Landys, C. Workman, R. Christmas, I. Avila-Campilo, M. Creech, B. Gross, K. Hanspers, R. Isserlin, R. Kelley, S. Killcoyne, S. Lotia, S. Maere, J. Morris, K. Ono, V. Pavlovic, A.R. Pico, A. Vailaya, P. Wang, A. Adler, B.R. Conklin, L. Hood, M. Kuiper, C. Sander, I. Schmulevich, B. Schwikowski, G. J. Warner, T. Ideker, and G.D. Bader. Integration of biological networks and gene expression data using cytoscape. *Nature Protocols*, 2(10):2366–2382, September 2007. ISSN 1754-2189. doi: 10.1038/nprot.2007.324. URL http://dx.doi.org/10.1038/nprot.2007.324.

A. Coja-Oghlan. Graph partitioning via adaptive spectral techniques. *Comb. Probab. Comput.*, 19(2):227–284, March 2010. ISSN 0963-5483. doi: 10.1017/S0963548309990514. URL http://dx.doi.org/10.1017/S0963548309990514.

A. Coja-Oghlan, F. Krzakala, W. Perkins, and L. Zdeborova. Information-theoretic thresholds from the cavity method. *ArXiv e-prints*, November 2016.

A. Condon and R. M. Karp. Algorithms for graph partitioning on the planted partition model. *Lecture Notes in Computer Science*, 1671:221–232, 1999.

I. Csiszár. Eine informationstheoretische ungleichung und ihre anwendung auf den beweis der ergodizitat von markoffschen ketten. *Magyar. Tud. Akad. Mat. Kutató Int. Közl*, 8:85–108, 1963.

A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Phys. Rev. E*, 84: 066106, December 2011.

Y. Deshpande, E. Abbe, and A. Montanari. Asymptotic mutual information for the two-groups stochastic block model. *arXiv:1507.08685*, 2015.

Yash Deshpande and Andrea Montanari. Information-theoretically optimal sparse pca. In *Information Theory (ISIT), 2014 IEEE International Symposium on*, pages 2197–2201. IEEE, 2014.

P. Diaconis and S. Janson. Graph limits and exchangeable random graphs. *ArXiv e-prints*, December 2007.

David L. Donoho, Arian Maleki, and Andrea Montanari. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences*, 106(45):18914–18919, 2009. doi: 10.1073/pnas.0909892106. URL http://www.pnas.org/content/106/45/18914.abstract.

M.E. Dyer and A.M. Frieze. The solution of some random NP-hard problems in polynomial expected time. *Journal of Algorithms*, 10(4):451 – 489, 1989. ISSN 0196-6774. doi: 10.1016/0196-6774(89)90001-1. URL http://www.sciencedirect.com/science/article/pii/0196677489900011.

P. Erdös and A Rényi. On the evolution of random graphs. In *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, pages 17–61, 1960.

W. Evans, C. Kenyon, Y. Peres, and L. J. Schulman. Broadcasting on trees and the Ising model. *Ann. Appl. Probab.*, 10:410–433, 2000.

Uriel Feige and Eran Ofek. Spectral techniques applied to sparse random graphs. *Random Structures & Algorithms*, 27(2):251–275, 2005. ISSN 1098-2418. doi: 10.1002/rsa.20089. URL http://dx.doi.org/10.1002/rsa.20089.

S. Fortunato. Community detection in graphs. *Physics Reports*, 486 (3-5):75–174, 2010.

J. Friedman. A proof of alon's second eigenvalue conjecture. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing*, STOC '03, pages 720–724, New York, NY, USA, 2003. ACM.

C. Gao, Z. Ma, A. Y. Zhang, and H. H. Zhou. Achieving Optimal Misclassification Proportion in Stochastic Block Model. *ArXiv e-prints*, May 2015.

A. Ghasemian, P. Zhang, A. Clauset, C. Moore, and L. Peel. Detectability Thresholds and Optimal Algorithms for Community Structure in Dynamic Networks. *Physical Review X*, 6(3):031005, July 2016. doi: 10.1103/PhysRevX.6.031005.

M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002. doi: 10.1073/pnas.122653799. URL http://www.pnas.org/content/99/12/7821.abstract.

M. X. Goemans and D. P. Williamson. Improved apprximation algorithms for maximum cut and satisfiability problems using semidefine programming. *Journal of the Association for Computing Machinery*, 42:1115–1145, 1995.

A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi. A survey of statistical network models. *Foundations and Trends in Machine Learning*, 2(2):129–233, 2010.

P. K. Gopalan and D. M. Blei. Efficient discovery of overlapping communities in massive networks. *Proceedings of the National Academy of Sciences*, 2013.

Olivier Guédon and Roman Vershynin. Community detection in sparse networks via grothendieck's inequality. *Probability Theory and Related Fields*, 165(3):1025–1049, 2016. ISSN 1432-2064. doi: 10.1007/s00440-015-0659-z. URL http://dx.doi.org/10.1007/s00440-015-0659-z.

Dongning Guo, Shlomo Shamai, and Sergio Verdú. Mutual information and minimum mean-square error in gaussian channels. *Information Theory, IEEE Transactions on*, 51 (4):1261–1282, 2005.

B. Hajek, Y. Wu, and J. Xu. Information Limits for Recovering a Hidden Community. *ArXiv e-prints*, September 2015a.

B. Hajek, Y. Wu, and J. Xu. Recovering a Hidden Community Beyond the Spectral Limit in $O(|E|\log^*|V|)$ Time. *ArXiv e-prints*, 2015b.

B. Hajek, Y. Wu, and J. Xu. Achieving Exact Cluster Recovery Threshold via Semidefinite Programming: Extensions. *ArXiv e-prints*, February 2015c.

K.-I. Hashimoto. Zeta functions of finite graphs and representations of p-adic groups. *In Automorphic forms and geometry of arithmetic varieties. Adv. Stud. Pure Math.*, 15: 211–280, 1989.

S. Heimlicher, M. Lelarge, and L. Massoulié. Community detection in the labelled stochastic block model. *arXiv:1209.2910*, 2012.

P. W. Holland, K. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983. URL http://d.wanfangdata.com.cn/NSTLQK_10.1016-0378-8733(83)90021-7.aspx.

D. Hoover. *relations on Probability Spaces and Arrays of Random Variables*. Preprint, Institute for Advanced Study, Princeton., 1979.

M.D. Horton, H.M. Stark, and A.A. Terras. What are zeta functions of graphs and what are they good for? *Contemporary Mathematics, Quantum Graphs and Their Applications*, pages 415:173–190, 2006.

Svante Janson and Elchanan Mossel. Robust reconstruction on trees is determined by the second eigenvalue. *Ann. Probab.*, 32(3B):2630–2649, 07 2004. doi: 10.1214/009117904000000153. URL http://dx.doi.org/10.1214/009117904000000153.

A. Javanmard, A. Montanari, and F. Ricci-Tersenghi. Performance of a community detection algorithm based on semidefinite programming. *ArXiv e-prints*, March 2016.

D. Jiang, C. Tang, and A. Zhang. Cluster analysis for gene expression data: a survey. *Knowledge and Data Engineering, IEEE Transactions on*, 16(11):1370–1386, Nov 2004. ISSN 1041-4347. doi: 10.1109/TKDE.2004.68.

Jiashun Jin. Fast community detection by score. *Ann. Statist.*, 43(1):57–89, 02 2015. doi: 10.1214/14-AOS1265. URL http://dx.doi.org/10.1214/14-AOS1265.

V. Jog and P.-L. Loh. Information-theoretic bounds for exact recovery in weighted stochastic block models using the Renyi divergence. *ArXiv e-prints*, September 2015.

A. Joseph and B. Yu. Impact of regularization on Spectral Clustering. *ArXiv e-prints*, December 2013.

R. Kannan, S. Vempala, and A. Veta. On clusterings-good, bad and spectral. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 367–377, 2000. doi: 10.1109/SFCS.2000.892125.

B. Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Phys. Rev. E*, 83:016107, Jan 2011. doi: 10.1103/PhysRevE.83.016107. URL http://link.aps.org/doi/10.1103/PhysRevE.83.016107.

E. Kaufmann, T. Bonald, and M. Lelarge. A Spectral Algorithm with Additive Clustering for the Recovery of Overlapping Communities in Networks. *ArXiv e-prints*, June 2015.

Tatsuro Kawamoto and Yoshiyuki Kabashima. Limitations in the spectral method for graph partitioning: Detectability threshold and localization of eigenvectors. *Phys. Rev. E*, 91 (6):062803, 2015.

H. Kesten and B. P. Stigum. A limit theorem for multidimensional galton-watson processes. *Ann. Math. Statist.*, 37(5):1211–1223, 10 1966. doi: 10.1214/aoms/1177699266. URL http://dx.doi.org/10.1214/aoms/1177699266.

Robert Krauthgamer and Uriel Feige. A polylogarithmic approximation of the minimum bisection. *SIAM Review*, 48(1):99–130, 2006. doi: 10.1137/050640904. URL http://dx.doi.org/10.1137/050640904.

Florent Krzakala, Cristopher Moore, Elchanan Mossel, Joe Neeman, Allan Sly, Lenka Zdeborova, and Pan Zhang. Spectral redemption in clustering sparse networks. *Proceedings of the National Academy of Sciences*, 110(52):20935–20940, 2013. doi: 10.1073/pnas.1312486110. URL http://www.pnas.org/content/110/52/20935.abstract.

Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Trawling the web for emerging cyber-communities. *Comput. Netw.*, 31(11-16):1481–1493, May 1999. ISSN 1389-1286. doi: 10.1016/S1389-1286(99)00040-7. URL http://dx.doi.org/10.1016/S1389-1286(99)00040-7.

J. Lafferty. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. pages 282–289. Morgan Kaufmann, 2001.

C. M. Le, E. Levina, and R. Vershynin. Sparse random graphs: regularization and concentration of the Laplacian. *ArXiv e-prints*, February 2015.

Marc Lelarge and Léo Miolane. Fundamental limits of symmetric low-rank matrix estimation. *arXiv preprint arXiv:1611.03888*, 2016.

T. Lesieur, F. Krzakala, and L. Zdeborová. MMSE of probabilistic low-rank matrix estimation: Universality with respect to the output channel. *ArXiv e-prints*, July 2015.

G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, January 2003. ISSN 1089-7801. doi: 10.1109/MIC.2003.1167344. URL http://dx.doi.org/10.1109/MIC.2003.1167344.

L. Lovász. *Large Networks and Graph Limits*. American Mathematical Society colloquium publications. American Mathematical Society, 2012. ISBN 9780821890851. URL http://books.google.com/books?id=FsFqHLid8sAC.

L. Lovász and B. Szegedy. Limits of dense graph sequences. *Journal of Combinatorial Theory, Series B*, 96(6):933 – 957, 2006. ISSN 0095-8956. doi: http://dx.doi.org/10.1016/j.jctb.2006.05.002. URL http://www.sciencedirect.com/science/article/pii/S0095895606000517.

K. Makarychev, Y. Makarychev, and A. Vijayaraghavan. Learning Communities in the Presence of Errors. *ArXiv e-prints*, November 2015.

E.M. Marcotte, M. Pellegrini, H.-L. Ng, D.W. Rice, T.O. Yeates, and D. Eisenberg. Detecting protein function and protein-protein interactions from genome sequences. *Science*, 285(5428):751–753, 1999. doi: 10.1126/science.285.5428.751.

L. Massoulié. Community detection thresholds and the weak Ramanujan property. In *STOC 2014: 46th Annual Symposium on the Theory of Computing*, pages 1–10, New York, United States, June 2014. URL https://hal.archives-ouvertes.fr/hal-00969235.

F. McSherry. Spectral partitioning of random graphs. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 529–537, 2001. doi: 10.1109/SFCS.2001.959929.

M. Mézard, G. Parisi, and R. Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297:812–815, 2003.

Marc Mézard and Andrea Montanari. Reconstruction on trees and spin glass transition. *Journal of Statistical Physics*, 124(6):1317–1350, 2006. ISSN 0022-4715. doi: 10.1007/s10955-006-9162-3. URL http://dx.doi.org/10.1007/s10955-006-9162-3.

Ankur Moitra, William Perry, and Alexander S Wein. How robust are reconstruction thresholds for community detection? In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, pages 828–841. ACM, 2016.

A. Montanari. Finding one community in a sparse graph. *arXiv:1502.05680*, 2015.

Andrea Montanari and Subhabrata Sen. Semidefinite programs on sparse random graphs and their application to community detection. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2016, pages 814–827, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4132-5. doi: 10.1145/2897518.2897548. URL http://doi.acm.org/10.1145/2897518.2897548.

C. Moore. The Computer Science and Physics of Community Detection: Landscapes, Phase Transitions, and Hardness. *ArXiv e-prints*, February 2017.

E. Mossel and Y. Peres. Information flow on trees. *Ann. Appl. Probab.*, 13:817–844, 2003.

E. Mossel and J. Xu. Density Evolution in the Degree-correlated Stochastic Block Model. *ArXiv e-prints*, September 2015.

E. Mossel, J. Neeman, and A. Sly. Belief propagation, robust reconstruction, and optimal recovery of block models. *Arxiv:arXiv:1309.1380*, 2013.

E. Mossel, J. Neeman, and A. Sly. A proof of the block model threshold conjecture. *Available online at arXiv:1311.4115 [math.PR]*, January 2014a.

E. Mossel, J. Neeman, and A. Sly. Consistency thresholds for binary symmetric block models. *Arxiv:arXiv:1407.1591. In proc. of STOC15.*, July 2014b.

Elchanan Mossel, Joe Neeman, and Allan Sly. Reconstruction and estimation in the planted partition model. *Probability Theory and Related Fields*, 162(3):431–461, 2015. ISSN 1432-2064. doi: 10.1007/s00440-014-0576-6. URL http://dx.doi.org/10.1007/s00440-014-0576-6.

Raj Rao Nadakuditi and M. E. J. Newman. Graph spectra and the detectability of community structure in networks. *Phys. Rev. Lett.*, 108:188701, May 2012. doi: 10.1103/PhysRevLett.108.188701. URL http://link.aps.org/doi/10.1103/PhysRevLett.108.188701.

J. Neeman and P. Netrapalli. Non-reconstructability in the stochastic block model. *Available at arXiv:1404.6304*, 2014.

M. Newman. *Networks: an introduction*. Oxford University Press, Oxford, 2010.

M. E. J. Newman. Communities, modules and large-scale structure in networks. *Nature Physics*, 8(1):25–31, dec 2011. ISSN 1745-2473. doi: 10.1038/nphys2162. URL http://dx.doi.org/10.1038/nphys2162.

M. E. J. Newman, D. J. Watts, and S. H. Strogatz. Random graph models of social networks. *Proc. Natl. Acad. Sci. USA*, 99:2566–2572.

Mark EJ Newman and Tiago P Peixoto. Generalized communities in networks. *Phys. Rev. Lett.*, 115(8):088701, 2015.

Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, pages 849–856. MIT Press, 2001.

Sofia C. Olhede and Patrick J. Wolfe. Network histograms and universality of blockmodel approximation. *Proceedings of the National Academy of Sciences*, 111(41):14722–14727, 2014. doi: 10.1073/pnas.1400374111.

G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 2005.

Tiago P Peixoto. Model selection and hypothesis testing for large-scale network models with overlapping groups. *Phys. Rev. X*, 5(1):011033, 2015.

A. Perry and A. S. Wein. A semidefinite program for unbalanced multisection in the stochastic block model. *ArXiv e-prints*, July 2015.

Jörg Reichardt and Michele Leone. (Un)detectable cluster structure in sparse networks. *Phys. Rev. Lett.*, 101(7):078701, 2008.

T. Richardson and R. Urbanke. An introduction to the analysis of iterative coding systems. In *Codes, Systems, and Graphical Models*, IMA Volume in Mathematics and Its Applications, pages 1–37. Springer, 2001.

A. Saade, F. Krzakala, and L. Zdeborová. Spectral Clustering of Graphs with the Bethe Hessian. *ArXiv e-prints*, June 2014.

A. Saade, F. Krzakala, M. Lelarge, and L. Zdeborová. Spectral detection in the censored block model. *arXiv:1502.00163*, January 2015.

S. Sahebi and W. Cohen. Community-based recommendations: a solution to the cold start problem. In *Workshop on Recommender Systems and the Social Web (RSWEB), held in conjunction with ACM RecSys11*, October 2011. URL http://d-scholarship.pitt.edu/13328/.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York, NY, USA, 2014. ISBN 1107057132, 9781107057135.

C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, july, october 1948. URL http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf.

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 1997.

T. A. B. Snijders and K. Nowicki. Estimation and Prediction for Stochastic Blockmodels for Graphs with Latent Block Structure. *Journal of Classification*, 14(1):75–100, January 1997. ISSN 0176-4268. doi: 10.1007/s003579900004. URL http://dx.doi.org/10.1007/s003579900004.

T. Sorlie, C.M. Perou, R. Tibshirani, T. Aas, S. Geisler, H. Johnsen, T. Hastie, Mi.B. Eisen, M. van de Rijn, S.S. Jeffrey, T. Thorsen, H. Quist, J.C. Matese, P.O. Brown, D. Botstein, P.E. Lonning, and A. Borresen-Dale. Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. 98(19):10869–10874, 2001. doi: 10.1073/pnas.191367098.

Daniel A. Spielman and Shang-Hua Teng. Spectral partitioning works: Planar graphs and finite element meshes. *Linear Algebra and its Applications*, 421(2):284 – 305, 2007. ISSN 0024-3795. doi: http://dx.doi.org/10.1016/j.laa.2006.07.020. URL `http://www.sciencedirect.com/science/article/pii/S0024379506003454`.

E. Szemerédi. Regular partitions of graphs. *Problemes combinatoires et theorie des graphes (Colloq. Internat. CNRS, Univ. Orsay, Orsay, 1976)*, 1976.

Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4): 395–416, 2007. ISSN 1573-1375. doi: 10.1007/s11222-007-9033-z. URL `http://dx.doi.org/10.1007/s11222-007-9033-z`.

V. Vu. A simple svd algorithm for finding hidden partitions. *Available at arXiv:1404.3918. To appear in CPC*, April 2014.

Van H. Vu. Spectral norm of random matrices. *Combinatorica*, 27(6):721–736, November 2007. ISSN 0209-9683. doi: 10.1007/s00493-007-2190-z. URL `http://dx.doi.org/10.1007/s00493-007-2190-z`.

P. J. Wolfe and S. C. Olhede. Nonparametric graphon estimation. *ArXiv e-prints*, September 2013.

R. Wu, J. Xu, R. Srikant, L. Massoulié, M. Lelarge, and B. Hajek. Clustering and Inference From Pairwise Comparisons. *ArXiv e-prints*, February 2015.

J. Xu, M. Lelarge, and L. Massoulie. Edge label inference in generalized stochastic block models: from spectral theory to impossibility results. *Proceedings of COLT 2014*, 2014.

J. Xu Y. Chen. Statistical-computational tradeoffs in planted problems and submatrix localization with a growing number of clusters and submatrices. *arXiv:1402.1267*, February 2014.

Jean-Gabriel Young, Patrick Desrosiers, Laurent Hébert-Dufresne, Edward Laurence, and Louis J Dubé. Finite size analysis of the detectability limit of the stochastic block model. *arXiv:1701.00062*, 2016.

S. Yun and A. Proutiere. Accurate community detection in the stochastic block model via spectral algorithms. *arXiv:1412.7335*, December 2014.

S.-Y. Yun and A. Proutiere. Community Detection via Random and Adaptive Sampling. *ArXiv e-prints 1402.3072. In proc. COLT14*, February 2014.

S.-Y. Yun and A. Proutiere. Optimal Cluster Recovery in the Labeled Stochastic Block Model. *ArXiv e-prints*, October 2015.

Pan Zhang, Cristopher Moore, and Lenka Zdeborová. Phase transitions in semisupervised clustering of sparse networks. *Phys. Rev. E*, 90:052802, Nov 2014. doi: 10.1103/PhysRevE.90.052802. URL `http://link.aps.org/doi/10.1103/PhysRevE.90.052802`.

Pan Zhang, Cristopher Moore, and M. E. J. Newman. Community detection in networks with unequal groups. *Phys. Rev. E*, 93:012303, Jan 2016. doi: 10.1103/PhysRevE.93.012303. URL `http://link.aps.org/doi/10.1103/PhysRevE.93.012303`.

Tao Zhou, Jie Ren, Matús Medo, and Yi-Cheng Zhang. Bipartite network projection and personal recommendation. *Phys. Rev. E*, 76:046115, Oct 2007. doi: 10.1103/PhysRevE.76.046115. URL `http://link.aps.org/doi/10.1103/PhysRevE.76.046115`.

Journal of Machine Learning Research 18 (2018) 1-42 Submitted 11/16; Revised 12/17; Published 4/18

On b-bit Min-wise Hashing for Large-scale Regression and Classification with Sparse Data

Rajen D. Shah r.shah@statslab.cam.ac.uk
Statistical Laboratory
University of Cambridge
Cambridge, CB3 0WB, UK

Nicolai Meinshausen meinshausen@stat.math.ethz.ch
Seminar für Statistik
ETH Zürich
8092 Zürich, Switzerland

Editor: Sanjiv Kumar

Abstract

Large-scale regression problems where both the number of variables, p, and the number of observations, n, may be large and in the order of millions or more, are becoming increasingly more common. Typically the data are sparse: only a fraction of a percent of the entries in the design matrix are non-zero. Nevertheless, often the only computationally feasible approach is to perform dimension reduction to obtain a new design matrix with far fewer columns and then work with this compressed data.

b-bit min-wise hashing (Li and König, 2011; Li et al., 2011) is a promising dimension reduction scheme for sparse matrices which produces a set of random features such that regression on the resulting design matrix approximates a kernel regression with the resemblance kernel. In this work, we derive bounds on the prediction error of such regressions. For both linear and logistic models, we show that the average prediction error vanishes asymptotically as long as $q\|\boldsymbol{\beta}^*\|_2^2/n \to 0$, where q is the average number of non-zero entries in each row of the design matrix and $\boldsymbol{\beta}^*$ is the coefficient of the linear predictor.

We also show that ordinary least squares or ridge regression applied to the reduced data can in fact allow us fit more flexible models. We obtain non-asymptotic prediction error bounds for interaction models and for models where an unknown row normalisation must be applied in order for the signal to be linear in the predictors.

Keywords: large-scale data, min-wise hashing, resemblance kernel, ridge regression, sparse data.

1. Introduction

The modern field of high-dimensional statistics has now developed a powerful range of methods to deal with data sets where the number of variables p may greatly exceed the number of variables n (see Bühlmann and van de Geer (2011) for an overview of recent advances). The prototypical example of microarray data, where p may be in the tens of thousands but n is typically not more than a few hundred, has motivated much of this development. Yet not all modern data sets come in this sort of shape and size. The emerging area of 'large-scale data' or the more vaguely defined 'Big Data' is a response to the increasing prevalence of computationally challenging data sets as arise in text analysis or web-scale prediction tasks, to give two examples. Here both n and p can run into the millions or more, particularly if interactions are considered. In these 'large p, large n' regression scenarios, one can imagine situations where ordinary least squares (OLS) has a competitive performance for prediction, but the sheer size of the data renders it infeasible for computational rather than statistical reasons.

An important feature of many large-scale data sets is that they are sparse: the overwhelming majority of entries in the design matrices are exactly zero. This is not to be confused with signal sparsity, a common assumption in the high-dimensional context. Indeed, when the design matrix is sparse, having only a few variables that contribute to the response would make the expected response values of all observations with no non-zero entries for the important variables exactly the same; one expects that such a property would not be possessed by many data sets. However, similarly to the way in which many high-dimensional techniques exploit sparsity to improve statistical efficiency, one might hope that sparsity in the data could be leveraged to yield both computational and statistical improvements, and indeed we demonstrate in this work that this can be achieved.

Kernel machines are an important class of machine learning methods for which such large-scale data poses particularly serious computational challenges. For example, standard implementations of kernel ridge regression would have computational complexity $O(n^3)$ and a storage cost of $O(n^2)$ when p is considered fixed; a large p will increase these computational costs depending on the kernel to be used. There has therefore been a great deal of work on approximating kernel machines by first randomly mapping the $n \times p$ design matrix $\mathbf{X}$ to a $n \times d$ matrix $\mathbf{S}$ with $d \ll p$ such that dot products between rows of $\mathbf{S}$ approximate the kernel evaluated on the corresponding rows of $\mathbf{X}$. Then a regular ridge regression on $\mathbf{S}$ will resemble a kernel ridge regression on $\mathbf{X}$, for example.

A remarkably effective way of forming $\mathbf{S}$ that is applicable when the design matrix is sparse and binary, is b-bit min-wise hashing (Li and König, 2011; Li et al., 2011) which is based on an earlier technique called min-wise hashing (Broder et al., 1998; Cohen et al., 2001; Datar and Muthukrishnan, 2002). Here $\mathbf{S}$ is constructed such that the dot product between any two rows of $\mathbf{S}$, $\mathbf{s}_i^T\mathbf{s}_j$, can approximate the *resemblance* or *Jaccard similarity* or between the corresponding rows of $\mathbf{X}$, defined as $|\mathbf{z}_i \cap \mathbf{z}_j|/|\mathbf{z}_i \cup \mathbf{z}_j|$ where $\mathbf{z}_i = \{k : X_{ik} \neq 0\}$.

The empirical performance of regression and classification procedures following b-bit min-wise hashing (Li et al., 2011, 2013) is particularly impressive. Existing theory on b-bit min-wise hashing (Li and König, 2011) has focused on the variance and bias in the approximation of the kernel. However, there remain significant gaps in our theoretical understanding of this important procedure when used to approximate a kernel machine:

(a) What sorts of regression models is the resemblance kernel well-suited for and how does sparsity of the design matrix play a role?

(b) What is the loss in prediction accuracy due to the approximation provided by b-bit min-wise hashing for different sorts of regression procedures?

(c) What is the overall prediction error incurred by different regression methods following b-bit min-wise hashing in different regression models?

An answer to (c) would be the ultimate goal here, and it would appear that in order to tackle this one must first solve (a) and (b). In this paper, we take a very different approach and aim to answer (c) directly: rather than considering what sorts of functions lie in the reproducing kernel Hilbert space (RKHS) associated with the resemblance kernel and have low RKHS norms, we look at the sorts of signals that can be approximated well by linear combinations of columns of the matrix $\mathbf{S}$ constructed by b-bit min-wise hashing. In this way, we use the random feature expansions provided by b-bit min-wise hashing to understand the predictive properties of the resemblance kernel.

1.1 Our contributions and organisation of the paper

In this paper we derive finite-sample bounds on the expected risk of linear and logistic regression following dimension reduction through b-bit min-wise hashing under various different models. Our results show that the method, and hence also the resemblance kernel, are particularly suited to sparse data.

We describe the b-bit min-wise hashing algorithm in Section 2 and also discuss in greater details the connection to the resemblance kernel. We also introduce a generalisation of b-bit min-wise hashing applicable to sparse data with real-valued entries motivated by our theory. Perhaps the simplest sorts of signals that we could hope to be able to fit well are linear signals of the form $\mathbf{X}\boldsymbol{\beta}^*$. In Section 3 we first consider how well a linear combination of columns of $\mathbf{S}$ can approximate such a signal. We then study a much larger class of signals defined by first scaling the rows of $\mathbf{X}$ in different ways depending on their sparsity and then forming a linear signal from a scaled version of $\mathbf{X}$. Some form of row normalisation is often performed on the original data as a pre-processing step, but the optimal normalisation to use is seldom known; our theory shows how b-bit min-wise hashing, and hence also the resemblance kernel, is able to automatically discover an appropriate scaling in several settings.

In Section 4.1 we study the performance of ordinary least squares, ridge regression and ℓ_2-penalised logistic regression using the reduced design matrix it creates. Our results are applicable to both linear signals and nonlinear signals of the sort described above. In the former setting, we show that the expected mean-squared prediction error is bounded by a small constant times $\sqrt{q/n}\,\|\boldsymbol{\beta}^*\|_2$, where q is the average number nonzero entries in the rows of $\mathbf{X}$ and $\boldsymbol{\beta}^*$ is the coefficient vector. We present similar results for logistic regression.

In Section 5 we study another form of nonlinear signal that can be approximated by the b-bit minwise hashing and the resemblance kernel: we show that interaction models in the original data can also be captured by main effects regression on the compressed data. Variable importance measures are discussed in Section 6. We conclude with a discussion in Section 7. The appendix contains all proofs, an additional result concerning the implications of our approximation error bound for properties of the RKHS of the resemblance kernel, and an empirical study validating our bounds.

1.2 Related work

There has been very little work in understanding properties of the resemblance kernel. One of the few pieces of work in this direction is Bouchard et al. (2013), who show that the kernel matrix with entries given by the Jaccard similarity between different elements of the

power set of $\{1,\ldots,p\}$ minus the empty set is positive definite. It follows that the RKHS of the resemblance kernel contains every real-valued function on p-dimensional binary vectors (see Section B). However, this result is not informative for understanding which sorts of regression models a kernel ridge regression will perform well for, a question which we provide some answers to through our study of b-bit min-wise hashing.

Approximating kernel methods using random feature expansions was pioneered by Rahimi and Recht (2007) who used random Fourier features to approximate translation invariant kernels such as the Gaussian kernel. Sutherland and Schneider (2015) provides bounds on the approximation of the corresponding kernel as well as bounds on the distance between the predictions from regression on the random features and kernel ridge regression in terms of distances between the true kernel and its approximation. Le et al. (2013) introduce a scheme related to random Fourier features that further improves the computational efficiency. Rahimi and Recht (2008) consider more general random feature expansions and study how well they can approximate functions in a family determined by the distribution of feature expansions in terms of a certain form of function norm defined on the family. Rahimi and Recht (2009) provides prediction error bounds for a method that minimises the empirical risk of a weighted sum of random feature expansions where weights are constrained in ℓ_∞-norm. Bach (2017) studies how well random feature expansions can approximate elements of their corresponding RKHS in terms of the eigenvalues of the associated kernel integral operator. The Nyström method (Williams and Seeger, 2001) is related and aiming at a computationally efficient low-rank approximation to the full kernel matrix; see (Bach, 2013) and (Rudi et al., 2015) for approximation guarantees.

A distinguishing feature of our work is that bounds are obtained not in terms of the norm of the RKHS of the resemblance kernel, which would be difficult to interpret, but in terms of quantities derived directly from the different models considered (we look at linear models with unknown row scaling and at nonlinear interaction models). We could divide the analysis into two parts: (i) first we could try to understand the predictive accuracy when using exact kernel regression with the resemblance kernel for such true regression functions and then (ii) in a second step understand how much predictive accuracy we lose by using b-bit minwise hashing as an approximation to using exact kernel regression with the resemblance kernel. Instead of making these two separate steps, we study here directly how well b-bit minwise hashing performs for these model classes.

Properties of b-bit min-wise hashing related to similarity search are studied in Li and König (2011). Theory concerning its use for large-scale learning is presented in Li et al. (2011) which quantifies the mean and variance of entries in the Gram matrix $\mathbf{S}\mathbf{S}^T$ and its relationship to the resemblance kernel as well as providing comparisons with random projections and *Vowpal Wabbit*. Random feature expansions for other types of kernels are developed in Shi et al. (2009); Weinberger et al. (2009); Vedaldi and Zisserman (2012); Kar and Karnick (2012); Li (2014); Pennington et al. (2015).

More generally, there is a huge variety of dimension reduction schemes across the statistics and computer science literature. Performing principal component analysis (Jolliffe, 1986) (PCA) and retaining only the first d components is one of the most popular methods. One drawback however in the large-scale data setting is that computing the principal components can be computationally demanding. The method of random projections, motivated by the celebrated Johnson–Lindenstrauss lemma (Johnson and Lindenstrauss, 1984), offers

dimension reduction at a low computational cost. In this scheme, $\mathbf{X}$ is mapped to $\mathbf{XA}$, where $\mathbf{A}$ is a $p \times d$ matrix typically with i.i.d. random entries. Efficient implementations are discussed in Achlioptas (2001); Li et al. (2006) and some numerical results on random projections and a wider literature review are in Fradkin and Madigan (2003); Vempala (2005). The software package *Vowpal Wabbit* (Langford et al., 2007) is a popular learning system for large-scale data sets that uses sparse random projections.

A separate line of work has considered pre-multiplying $\mathbf{X}$ with a random matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ to produce a reduced matrix $\mathbf{AX} \in \mathbb{R}^{m \times p}$, known as a *sketch*. Though the dimension p is not reduced, when n is large, performing OLS on the sketched matrix may be possible despite the computational infeasibility of applying least squares directly to $\mathbf{X}$. A number of works have studied properties sketched least squares (see Boutsidis and Drineas (2009); Drineas et al. (2011); Mahoney (2011); Pilanci and Wainwright (2015) and references therein) whilst Pilanci and Wainwright (2014) propose an iterative variant of this scheme. Yang et al. (2017) considers sketching ideas in the context of kernel ridge regression.

2. b-bit min-wise hashing

Given a sparse design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, the aim of dimension reduction is to map this to a compressed matrix $\mathbf{S} \in \mathbb{R}^{n \times d}$, in a way that is computationally efficient and such that the relevant information in $\mathbf{X}$ is preserved in $\mathbf{S}$. Section 2.2 describes the mapping to $\mathbf{S}$ under b-bit min-wise hashing for binary data, as proposed in Li and König (2011) and Li et al. (2011). The construction may seem unintuitive at first sight, but we will try to shed light on why the scheme works for linear and interaction models throughout the manuscript.

2.1 Notation

Given a matrix $\mathbf{U}$, we will write $\mathbf{u}_i$ and $\mathbf{U}_j$ for the ith row and jth column respectively, where both are to be regarded as column vectors. The ijth entry will be denoted U_{ij}. A vector of 1's will be denoted $\mathbf{1}$.

When the parentheses following probability and expectation signs, $\mathbb{P}$ and $\mathbb{E}$, enclose multiple potential sources of randomness, we will sometimes add subscripts to indicate what is being considered as random. For example, if U and V are random variables, we may write $\mathbb{E}_U(U|V)$ for the conditional expectation of U given V, and $\mathbb{E}_{U,V}(U+V)$ for the expected value of $U+V$.

2.2 Construction of S with b-bit min-wise hashing and binary variables

The compressed matrix $\mathbf{S}$ generated by b-bit min-wise hashing consists of blocks of size 2^b, where we may choose the number of blocks L. Each block is created using a random permutation and the blocks of columns form a collection of L i.i.d. random matrices.

There are three steps to the construction.

Step 1: Generate a random permutation of the set $\{1, \ldots, p\}$, π_l, and permute the columns of $\mathbf{X}$ according to this permutation.

Step 2: Search along each row of the permuted design matrix (in order of increasing column index) and record in the vector $\mathbf{H}_l \in \mathbb{N}^n$ the indices of the variables (indexed as in the original order) with the first non-zero value or the vector $\mathbf{M}_l \in \mathbb{N}^n$ the indices of the variables (indexed as in the permuted order) with the first non-zero value.

Step 3: Form $\mathbf{S}_l \in \{0,1\}^{n \times 2^b}$ with ith row given by the last b bits of the binary representation of the ith entry of $\mathbf{M}_l$. For example, when $b = 1$, all odd numbers in $\mathbf{M}_l$ map to the vector $(0,1)$, whereas all even numbers map to $(1,0)$.

This construction is illustrated for a toy example in Table 1.

$\mathbf{X} = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{pmatrix} \cdot & 1 & \cdot & 1 \\ \cdot & \cdot & 1 & 1 \\ 1 & \cdot & 1 & \cdot \\ \cdot & 1 & 1 & \cdot \\ 1 & 1 & \cdot & \cdot \end{pmatrix} \end{array} \overset{\pi_l = 2314}{\mapsto} \begin{array}{c} \begin{matrix} 3 & 1 & 2 & 4 \end{matrix} \\ \begin{pmatrix} \cdot & \cdot & \mathbf{1} & 1 \\ \mathbf{1} & \cdot & \cdot & 1 \\ \mathbf{1} & 1 & \cdot & \cdot \\ \mathbf{1} & \cdot & 1 & \cdot \\ \cdot & \mathbf{1} & 1 & \cdot \end{pmatrix} \end{array}$	$\mathbf{H}_l = \begin{pmatrix} 2 \\ 3 \\ 3 \\ 3 \\ 1 \end{pmatrix}, \mathbf{M}_l = \begin{pmatrix} 3 \\ 1 \\ 1 \\ 1 \\ 2 \end{pmatrix}$	$\mathbf{S}_l = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$
Step 1: non-zero indices whose variable indices will appear in $\mathbf{H}_l$ in Step 2 are in bold.	*Step 2.*	*Step 3.*

Table 1: Steps 1–3 applied to a toy example with $b = 2$. Dots represent zeroes.

We can think of each column of $\mathbf{S}_l$ as representing different categories for the observations. The matrix $\mathbf{S}_l$ itself codes for the assignment of the different rows of $\mathbf{X}$ to the different categories. Different blocks $\mathbf{S}_l$ then represent different random categorisations. Identical rows will always be assigned the same categories and the more different the rows are, the less likely they are to be assigned the same category. The notion of difference here is that of *resemblance*; see Section 2.4

Note that one would not necessarily follow the above steps when implementing b-bit min-wise hashing. In practice, one would not store the entire matrix of signs nor all the random permutations. In an implementation, hash functions (Carter and Wegman, 1979) would be used to create the matrix $\mathbf{S}$ deterministically, though it is beyond the scope of this paper to go into the details; see Li et al. (2013) for more information and further computational improvements. With this approach, $\mathbf{S}$ would be created row-by-row, and only a single observation from $\mathbf{X}$ would need to be kept in memory at any one time. Furthermore, many rows could be created in parallel. Other ideas such as one-permutation hashing (Li et al., 2012) can also be used to speed up the pre-processing step.

2.3 Continuous data and additional randomisation

For continuous data, we introduce a modification where we replace the map extracting the last b bits by L random maps in the following way. Fix b and let $\mathbf{\Psi} \in \{1, \ldots, 2^b\}^{p \times L}$ be a random matrix with independent entries each having the uniform distribution on the set $\{1, \ldots, 2^b\}$. We then create $\mathbf{S}$ by modifying the previous Step 3 to the following.

Step 3: Form $\mathbf{S}_l \in \{0,1\}^{n \times 2^b}$ with ith row all zero except component $\Psi_{H_{il}l}$ takes the value 1.

Step 4: If $\mathbf{X}$ is not binary, multiply the ith row of $\mathbf{S}_l$ by $X_{iH_{il}}$.

This generalisation is motivated by our theoretical results on how well the column space of $\mathbf{S}$ can capture different sorts of signals (see Section 3.1).

Let $\mathbf{z}_i = \{k : X_{ik} \neq 0\}$ be the set of variable indices whose entries have non-zero values for the ith observation. Performing the steps above for all $l = 1, \ldots, L$, we get $n \times L$ matrices $\mathbf{H}$, and $\mathbf{M}$ given by

$$H_{il} = \arg\min_{k \in \mathbf{z}_i} \pi_l(k), \tag{1}$$

$$M_{il} = \min_{k \in \mathbf{z}_i} \pi_l(k) = \pi_l(H_{il}), \tag{2}$$

The matrix $\mathbf{S}$ is a binary $n \times 2^b L$ matrix. With a slight abuse of notation, we will denote by $\mathbf{S}_{ilc}$ the cth entry in the lth block of $\mathbf{S}$:

$$S_{ilc} := S_{i(c+(l-1)2^b)} = X_{iH_{il}} \mathbb{1}_{\{\Psi_{H_{il}l} = c\}}, \qquad \text{for } c = 1, \ldots, 2^b. \tag{3}$$

If not stated otherwise, we will work with this second randomised variation of b-bit min-wise hashing from now on. We emphasise that we do not make the claim this version is to be preferred over the original proposal of Li and König (2011) and Li et al. (2011) when data is binary. We simply introduce the additional randomisation here to simplify the analysis. We note that the two versions are essentially identical for all practical purposes when b is not too large.

2.4 The resemblance kernel

We now briefly describe the connection between b-bit min-wise hashing and the resemblance kernel alluded to earlier. This is not needed for the rest of the paper, though it provides some intuition for the scheme. A more detailed analysis from this perspective is carried out by Li et al. (2011) and we refer the reader to Hofmann et al. (2008) for a review of kernel methods and the kernel trick.

Suppose $\mathbf{X}$ is binary. Consider the normalised Gram matrix of the compressed design $\mathbf{S}$ from (randomised) b-bit min-wise hashing, $\mathbf{S}\mathbf{S}^T/L$. The expected value of the ijth component may be calculated as follows.

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\mathbf{s}_i^T \mathbf{s}_j / L) &= \frac{1}{L} \sum_{l=1}^{L} \sum_{c=1}^{2^b} \mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\mathbb{1}_{\{\Psi_{H_{il}l} = c\}} \mathbb{1}_{\{\Psi_{H_{jl}l} = c\}}) \\
&= \mathbb{P}(\Psi_{H_{il}l} = \Psi_{H_{jl}l}) \\
&= \mathbb{P}(\Psi_{H_{il}l} = \Psi_{H_{jl}l} | H_{il} = H_{jl}) \mathbb{P}(H_{il} = H_{jl}) \\
&\quad + \mathbb{P}(\Psi_{H_{il}l} = \Psi_{H_{jl}l} | H_{il} \neq H_{jl}) \{1 - \mathbb{P}(H_{il} = H_{jl})\} \\
&= \frac{|\mathbf{z}_i \cap \mathbf{z}_j|}{|\mathbf{z}_i \cup \mathbf{z}_j|} (1 - 2^{-b}) + 2^{-b}.
\end{aligned}$$

Thus the ijth entry is an average of L i.i.d. random variables with expectation a constant plus a constant times the resemblance between the ith and jth rows of $\mathbf{X}$. If an intercept term is included when regressing on $\mathbf{S}$, the additive constant plays no part, and the scaling would be absorbed into the scaling of the regression coefficients. We also note that when $\mathbf{X}$ is continuous, the resulting kernel is similar to the the CoRE kernels of Li (2014).

Now as the resemblance kernel is positive definite, the theory surrounding the kernel trick tells us that any ℓ_2-regularised regression on $\mathbf{S}$ is effectively approximating a regularised regression on transformed data $\phi(\mathbf{x}_i)$ where $\phi : \{0,1\}^p \to \mathcal{H}$ and $\mathcal{H}$ is a high-dimensional inner product space (the feature space). This space may be taken to be a reproducing kernel Hilbert space (RKHS), and then ϕ and $\mathcal{H}$ are uniquely defined.

Although this is encouraging, the kernel trick does not guarantee that regression on $\mathbf{S}$ will necessarily have good predictive properties for models of interest. To gain a better understanding, we must study the regularisation properties of the resemblance kernel itself: what characterises those elements of the associated RKHS $\mathcal{H}$ that have low norm and thus will be penalised less?

A direct analysis of the RKHS corresponding to the resemblance kernel in those terms seems challenging. We take a different approach and explicitly construct regression coefficients for $\mathbf{S}$ that approximate signals of interest. By showing that particular signals can be approximated well, we are indirectly discovering elements of $\mathcal{H}$ with low RKHS norm (see also Section B for more details).

3. Approximation error

In this section, we present results that bound the expected prediction error when performing regression on the reduced design matrix $\mathbf{S}$ in the contexts of the linear and logistic regression models. Note that throughout the rest of the manuscript, by b-bit min-wise hashing we are referring to the randomised variant described in Section 2.3. Let q_i be the number of non-zero entries in the ith row of $\mathbf{X}$, and let $\delta_i = q_i/p$ be the row sparsity. We will assume that the signal we wish to approximate for the ith observation takes the form

$$\kappa(\delta_i) \mathbf{x}_i^T \boldsymbol{\beta}^*. \tag{4}$$

Here $\boldsymbol{\beta}^* \in \mathbb{R}^p$ is an unknown vector of coefficients and the function κ allows the ith linear predictor to be scaled in a way which depends on the number of non-zero entries in the ith row of $\mathbf{X}$. Some normalisations of special interest include:

(a) $\kappa(\delta)$ constant. This yields standard linear or logistic regression models.

(b) $\kappa(\delta) \propto \delta^{-1/2}$. In text analysis with a bag of words representation of documents, rows of $\mathbf{X}$ are often scaled to have the same ℓ_2-norm to help balance situations when documents vary greatly in length (Banerjee et al., 2005). When $\mathbf{X}$ is binary, this is exactly achieved by taking $\kappa(\delta) = p^{-1/2}\delta^{-1/2}$, so $\kappa(\delta_i) = q_i^{-1/2}$.

(c) $\kappa(\delta) \propto \delta^{-1}$. This leads to a ℓ_1-norm scaling as opposed to the ℓ_2-norm scaling mentioned above.

Throughout we will assume that $\mathbf{X} \in [-1,1]^{n \times p}$, so the entries in $\mathbf{X}$ are bounded. This covers the important case of binary design but also allows for real-valued entries.

The first step in obtaining our prediction error results is to construct a vector $\mathbf{b}^*$ such that $\mathbf{s}_i^T \mathbf{b}^*$ is close to $\kappa(\delta_i) \mathbf{x}_i^T \boldsymbol{\beta}^*$ on average.

3.1 Un-scaled signals

We will first consider un-scaled signals where $\kappa(\delta)$ in (4) is a constant. Non-constant row-scaling is treated in more detail in the Section 3.2. To begin with we will assume that $q_i = q \geq 1$ for all $i = 1, \ldots, n$, a restriction which simplifies the results but highlights some interesting properties of b-bit min-wise hashing. Unequal row sparsity is treated in detail in the appendix in Section A.4 but a sketch of the results are given just below Theorem 1.

To simplify notation, we first introduce the following norm for $\boldsymbol{\beta} \in \mathbb{R}^p$,

$$\|\boldsymbol{\beta}\|_b^2 := \|\boldsymbol{\beta}\|_2^2 + (2^b - 2)\sum_{k=1}^{p} \frac{\|\mathbf{X}_k\|_2^2}{n}\beta_k^2. \tag{5}$$

For $b = 1$, we have of course that $\|\boldsymbol{\beta}\|_b^2 = 2\|\boldsymbol{\beta}\|_2^2$. For larger values of b, the norm is influenced more heavily by the second term which can be seen to be the weighted version of the ℓ_2-norm, where the weight of each variable is proportional to its squared ℓ_2-norm. We will first discuss how well the original signal can be approximated with the column space of the matrix $\mathbf{S}$ generated by the b-bit min-wise hashing operation.

Theorem 1 *Let $\mathbf{S}$ be the matrix generated by b-bit min-wise hashing. Then there exists a vector $\mathbf{b}^* \in \mathbb{R}^{2^b L}$ with the following properties.*

(i) The approximation is unbiased: $\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\mathbf{S}\mathbf{b}^*) = \mathbf{X}\boldsymbol{\beta}^*$.

(ii) The norm is bounded by

$$\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\|\mathbf{b}^*\|_2^2) \leq \frac{(2-\delta)q}{L(1-2^{-b})}\|\boldsymbol{\beta}^*\|_2^2.$$

(iii) The approximation error is bounded by

$$\frac{1}{n}\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\|\mathbf{S}\mathbf{b}^* - \mathbf{X}\boldsymbol{\beta}^*\|_2^2) \leq \frac{(2-\delta)q}{2^b L(1-2^{-b})}\|\boldsymbol{\beta}^*\|_b^2.$$

Specifically, for $b = 1$, $\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\|\mathbf{S}\mathbf{b}^* - \mathbf{X}\boldsymbol{\beta}^*\|_2^2)/n \leq (2-\delta)q\|\boldsymbol{\beta}^*\|_2^2/L$.

A form of the approximation error (iii) and the norm bound (ii) continue to be valid in the non-equal sparsity case under a mild restriction on the size of L, where we get instead of (iii) the bound

$$\frac{1}{n}\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\|\mathbf{S}\mathbf{b}^* - \mathbf{X}\boldsymbol{\beta}^*\|_2^2) \leq \frac{6\bar{q}}{2^b L(1-2^{-b})}\|\boldsymbol{\beta}^*\|_b^2,$$

where $\bar{q}$ is the the average of the q_i; see Theorem 12 in the appendix for details.

The results above show that the signal $\mathbf{X}\boldsymbol{\beta}^*$ can be well approximated by a linear combination of the columns in the matrix $\mathbf{S}$ if we generate a sufficiently large number of permutations L, especially for sparse data matrices. Another useful property of $\mathbf{b}^*$ here, aside from the approximation accuracy it delivers, is given in (ii): on average, $\|\mathbf{b}^*\|_2^2$ is small when L is large. This proves to be useful when studying the application of ridge regression.

This result has interesting implications for the resemblance kernel and its RKHS $\mathcal{H}$. In particular, it shows that if we constrain the input space to contain those vectors with

sparsity q, linear functions $f_{\boldsymbol{\beta}}$ defined by coefficients $\boldsymbol{\beta} \in \mathbb{R}^p$ with $\sum_j \beta_j = 0$ have RKHS norm satisfying $\|f_{\boldsymbol{\beta}}\|_{\mathcal{H}}^2 \leq (2-\delta)q\|\boldsymbol{\beta}\|_2^2$. As these properties of the RKHS are not directly used in any subsequent results, we defer formal presentation of these facts to Section B in the appendix.

Whilst the bound on the expectation of $\|\mathbf{b}^*\|_2^2$ is almost constant as b changes, the approximation error bound (iii) does vary with b. Consider the case where $\mathbf{X}$ is binary and let $\gamma_k = \|\mathbf{X}_k\|_2^2/n$ be the column sparsity. Typically one would expect $\|\boldsymbol{\beta}^*\|_2^2$ to be significantly larger than $\sum_{k=1}^p \gamma_k \beta_k^{*2}$ and thus increasing b by 1 almost halves the approximation error when b is small.

A proof of Theorem 1 is given in Section A of the appendix; here we briefly sketch some of the main ideas. Note that

$$\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\mathbf{S}\mathbf{b}^*) = \sum_{l=1}^{L} \mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}\Big(\sum_{c=1}^{2^b} \mathbf{S}_{lc} b_{lc}^*\Big). \tag{6}$$

We construct $\mathbf{b}^*$ with the following two properties: each of the L blocks of $\mathbf{b}^*$ are i.i.d. with the lth block only depending on π_l and $\boldsymbol{\Psi}_l$; and each of the L summands in (6) equals $\mathbf{X}\boldsymbol{\beta}^*/L$. With each of the L summands being unbiased in this way, we see that the approximation error is controlled by the variance of the sum; this variance scales as $1/L$ since the summands are i.i.d.

At first sight it may seem surprising that it is possible to exhibit a $\mathbf{b}^*$ with each block having the unbiasedness property discussed above. However, the following construction gives an indication of the possibilities. Using our convention that the cth component of the lth block of $\mathbf{b}^*$ is indexed as $b_{lc}^* := b_{c+(l-1)2^b}^*$, consider taking

$$b_{lc}^* = \frac{q}{L}\sum_{k=1}^{p} \beta_k^* \frac{\mathbb{1}_{\{\Psi_{lk}=c\}} - 2^{-b}}{1-2^{-b}}. \tag{7}$$

Then writing $\boldsymbol{\psi} = \boldsymbol{\Psi}_1$, $\pi = \pi_1$, $H_i = H_{i1}$ we have

$$\begin{aligned}\frac{L}{q}\mathbb{E}_{\pi,\boldsymbol{\psi}}\Big(\sum_{c=1}^{2^b} S_{lc} b_{1c}^*\Big) &= \mathbb{E}_{\pi,\boldsymbol{\psi}}\Big(\sum_{c=1}^{2^b}\sum_{j=1}^{p} X_{ij}\mathbb{1}_{\{H_i=j,\psi_j=c\}}\sum_{k=1}^{p}\beta_k^*\frac{\mathbb{1}_{\{\psi_k=c\}}-2^{-b}}{1-2^{-b}}\Big)\\ &= \mathbb{E}_{\pi,\boldsymbol{\psi}}\Big(\sum_{j=1}^{p} X_{ij}\mathbb{1}_{\{H_i=j\}}\sum_{k=1}^{p}\beta_k^*\frac{\mathbb{1}_{\{\psi_k=\psi_j\}}-2^{-b}}{1-2^{-b}}\Big).\end{aligned} \tag{8}$$

Now since $\mathbb{E}_{\boldsymbol{\psi}}\{(\mathbb{1}_{\{\psi_k=\psi_j\}} - 2^{-b})/(1-2^{-b})\} = \mathbb{1}_{\{k=j\}}$ we see the above display equals

$$q\sum_{k=1}^{p} X_{ik}\beta_k^* \mathbb{P}_\pi(H_i = k) = \mathbf{X}\boldsymbol{\beta}^*.$$

The final line uses the fact that for k with $X_{ik} \neq 0$, $\mathbb{P}_\pi(H_i = k)$ is the reciprocal of the number of non-zero entries in the ith row of $\mathbf{X}$; with our simplifying assumption of equal row sparsity, this is precisely $1/q$. Note one could scale the rows of $\mathbf{S}$ according to the number of non-zeroes in each row to achieve unbiasedness in the case of unequal row

sparsity. However as shown in Section A.4, it turns out that by incurring some bias one can still keep the approximation error low even in this situation without having to perform any sort of scaling.

The form of $\mathbf{b}^*$ used in the proof of Theorem 1 differs slightly from that in (7) by introducing a random weight multiplying each coefficient that decays as $\pi_l(k)$ increases. This reduces the variance and yields the approximation error in (iii) that has a factor q rather than the factor of p which would be obtained from (7).

3.2 Row-scaled signals

We now turn to the more general setting with unequal row sparsity and signal given by (4). We consider the family of scaling functions $\delta \mapsto (\delta_{\min}/\delta)^a$ where $\delta_{\min} = \min_i \delta_i$, for $1/2 \le a \le 1$. Including $\delta_{\min}$ in the scaling functions means that were the row sparsity to be equal, the approximation error here would be of the same form as that considered in Theorems 1. We could alternatively replace $\delta_{\min}$ with the average of the δ_i for the same effect, but using $\delta_{\min}$ helps to simplify the results. Writing $q_{\min} = \min_i q_i$, we have the following results.

Theorem 2 *Let $L \ge 5$ and assume $\delta_{\min} \le 1/2$ if $a = 1/2$, and $L > 2/(2a-1)$ if $a > 1/2$. Then there exists $\mathbf{b}^* \in \mathbb{R}^L$ depending on a such that the approximation error satisfies*

$$\frac{1}{n}\sum_{i=1}^n \mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}[\{(\delta_{\min}/\delta_i)^a \mathbf{x}_i^T\boldsymbol{\beta}^* - \mathbf{s}_i^T\mathbf{b}^*\}^2] \le \begin{cases} \dfrac{q_{\min}}{2^b L(1-2^{-b})}\|\boldsymbol{\beta}^*\|_b^2 \log\{4\log(L)/\delta_{\min}\} & \text{if } a = 1/2, \\ \dfrac{q_{\min}}{2^b L(1-2^{-b})}\|\boldsymbol{\beta}^*\|_b^2 \dfrac{1}{2a-1}[\log\{2(2a-1)L\}]^{2a-1} & \text{if } 1/2 < a \le 1, \end{cases}$$

and the norm of $\mathbf{b}^*$ is bounded in expectation by

$$\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\|\mathbf{b}^*\|_2^2) \le \begin{cases} \dfrac{q_{\min}\log\{4\log(L)/\delta_{\min}\}}{L(1-2^{-b})}\|\boldsymbol{\beta}^*\|_2^2 & \text{if } a = 1/2, \\ \dfrac{1}{2a-1}\dfrac{q_{\min}[\log\{2(2a-1)L\}]^{2a-1}}{L(1-2^{-b})}\|\boldsymbol{\beta}^*\|_2^2 & \text{if } 1/2 < a \le 1. \end{cases}$$

The min-wise hashing based dimension reduction scheme appears to be well-suited to approximating signals scaled by a power of the sparsity, with the approximation error only incurring a further multiplicative term involving $\log(L)$ compared to the results of Theorem 1.

We now briefly outline how we construct coefficient vectors $\mathbf{b}^*$ achieving the bounds above. Consider the following refinement of (7):

$$b_{lc}^* = \frac{1}{L}\sum_{k=1}^p \beta_k^* \frac{\mathbb{1}_{\{\Psi_{lk}=c\}} - 2^{-b}}{1-2^{-b}} w_{\pi_l(k)},$$

where $\mathbf{w} \in \mathbb{R}^p$ is a vector of non-negative weights. Arguing as in (8) but replacing $q\beta_k^*$ with $\beta_k^* w_{\pi(k)}$ we arrive at

$$L\mathbb{E}_{\pi,\psi}\Big(\sum_{c=1}^{2^b} S_{lc} b_{1c}^*\Big) = \sum_{k=1}^p X_{ik}\beta_k^* \mathbb{E}_\pi(\mathbb{1}_{\{H_i=k\}} w_{\pi(k)}).$$

Recall that writing $M_i = M_{i1}$, $M_i = \pi(H_i)$, the position of the first non-zero entry in row i under permutation π. Note that H_i and M_i are independent. Now for large p, M_i behaves roughly like a geometric random variable with parameter δ_i. Thus for k with $X_{ik} \ne 0$,

$$\mathbb{E}_\pi(\mathbb{1}_{\{H_i=k\}} w_{\pi(k)}) = \mathbb{E}_\pi(\mathbb{1}_{\{H_i=k\}} w_{M_i}) \approx \frac{1}{p\delta_i}\sum_{\ell=1}^p w_\ell \delta_i(1-\delta_i)^{\ell-1} = \frac{1}{p}\sum_{\ell=1}^p w_\ell(1-\delta_i)^{\ell-1}.$$

If $w_{\ell+1} = p(-1)^\ell \kappa^{(\ell)}(1)/\ell!$ we see that the RHS resembles a Taylor series of $\kappa(\delta_i)$ about 1. In this way we can approximate a large family of row-scaled signals.

4. Prediction error

The approximation error results in the three previous sections allow us to derive bounds on the prediction errors for linear and logistic regression models with potentially row-scaled data. Here we will present results under the assumption of q non-zero entries per row and also where the scaling function κ is proportional to the square-root function

$$\kappa_0(\delta) = \sqrt{\delta_{\min}/\delta}. \tag{9}$$

However, all of the approximation error results can be extended to results on prediction error via general theorems on prediction error we present in Section D. In particular, Theorem 12 can be used to show that versions of the equal row sparsity results hold more generally with q replaced by the average number of non-zeroes per row $\bar{q}$ provided L is not excessively large.

4.1 Linear regression models

Assume we have the following approximately linear model:

$$Y_i = \alpha^* + \kappa(\delta_i)\mathbf{x}_i^T\boldsymbol{\beta}^* + \varepsilon_i, \qquad 1 = 1, \ldots, n. \tag{10}$$

Here α^* is the intercept and $\mathbf{x}_i \in [-1,1]^p$. We assume that the random noise $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ satisfies $\mathbb{E}(\varepsilon_i) = 0$, $\mathbb{E}(\varepsilon_i^2) = \sigma^2$ and $\mathrm{Cov}(\varepsilon_i, \varepsilon_j) = 0$.

Our results here give bounds on a mean-squared prediction error (MSPE) of the form

$$\mathrm{MSPE}((\hat{\alpha}, \hat{\mathbf{b}})) := \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{\boldsymbol{\varepsilon},\boldsymbol{\pi},\boldsymbol{\Psi}}\{(\alpha^* + \kappa(\delta_i)\mathbf{x}_i^T\boldsymbol{\beta}^* - \hat{\alpha} - \mathbf{S}\hat{\mathbf{b}})^2\} \tag{11}$$

where $\hat{\alpha}$ and $\hat{\mathbf{b}}$ are the estimated intercept and regression coefficients arising from regression on $\mathbf{S}$. Note we consider a denoising-type error: the error on the data used to fit the regression coefficients. Bounds on the prediction error at new observations would require conditions on the distribution of observations and we have avoided making any such assumptions for the results here.

4.1.1 Ordinary least squares

Perhaps the simplest way to estimate the linear model is to apply a least squares estimator,

$$(\hat{\alpha}, \hat{\mathbf{b}}) := \underset{(\alpha,\mathbf{b})\in\mathbb{R}\times\mathbb{R}^{2^b L}}{\arg\min} \|\mathbf{Y} - \alpha\mathbf{1} - \mathbf{S}\mathbf{b}\|_2^2, \tag{12}$$

to the matrix $\mathbf{S}$. We have the following theorem.

Theorem 3 *Let $(\hat{\alpha}, \hat{\mathbf{b}})$ be the least squares estimator (12). We have the bound*

$$\mathrm{MSPE}((\hat{\alpha}, \hat{\mathbf{b}})) \;\le\; \frac{C}{2^b L(1-2^{-b})}\,\|\boldsymbol{\beta}^*\|_b^2 + 2^b L\frac{\sigma^2}{n}.$$

For equal row sparsity δ we have $C = (2-\delta)q$. For unequal row sparsity, when $\kappa = \kappa_0$ as in (9), the result holds for $C = q_{\min}\log\{4\log(L)/\delta_{\min}\}$.

An optimal choice L_b^* of L will balance the approximation error and variance contributions (first and second term on the right hand side respectively). In the equal row sparsity we arrive at

$$L_b^* = \frac{\sqrt{(2-\delta)qn}}{2^b\sqrt{1-2^{-b}}}\|\boldsymbol{\beta}^*\|_b$$

which yields an optimal MSPE of the order $\sigma\sqrt{q/n}\|\boldsymbol{\beta}^*\|_b$. If we ignore log terms the rate is analogous in the case of uneven row-sparsity. The slow rate in n seems unavoidable if we do not make stronger conditions on the design. Indeed, a similar error rate is obtained in Theorem 21 of Maillard and Munos (2012) and in Kaban (2014) for OLS following dimension reduction by random projections. More precisely: projecting K times with a random projection, followed by an OLS estimation is shown in Kaban (2014) to lead to a bound on MSPE of

$$\frac{1}{K}\,\|\boldsymbol{\beta}^*\|_\kappa^2 + K\frac{\sigma^2}{n}, \tag{13}$$

where the norm $\|\cdot\|_\kappa$ depends on the eigenvalue structure of the design matrix. In contrast the bound we have above for min-wise hashing depends in contrast on the sparsity q through the constant C. The bound (13) is otherwise structurally identical to the bound for b-bit min-wise hashing above, and the role of the number L of projections is now taken by the number K of random projections. The optimal values of K and L are both of order $\sqrt{n}$, leading to the same convergence rate of the risk as $n\to\infty$.

To better understand the implications of Theorem 3, it is helpful to fix the size of the signal so that $\|\mathbf{X}\boldsymbol{\beta}^*\|_2^2/n = 1$, and look at whether we can show consistency of the method as both $p, n\to\infty$. If the signal is spread out and all variables have the same sparsity, $\|\boldsymbol{\beta}^*\|_b$ will be of order $\sqrt{p/q}$ and the MSPE will vanish when $p/n\to 0$, which excludes the high-dimensional setting.

However, now assume that the signal is concentrated on a fixed set of variables. The norm $\|\boldsymbol{\beta}^*\|_b$ is then constant as p increases and all that is required for consistency is $q/n\to 0$ (or $q_{\min}/n\to 0$ for the more general case of uneven row-sparsity).

An interesting scenario is one of increasing variable sparseness. In many applications, the more predictor variables are added the sparser they tend to become. In text analysis,

the first block of predictor variables might encode the presence of individual words. The next block might code for bigrams and the following, higher order N-grams. With this design, predictor variables in each successive block become sparser than the previous. It is then interesting to consider how much the MSPE can increase if we add a block with many sparse variables which contain no additional signal contribution. The result above indicates that the MSPE only increases as $\sqrt{q}$. Adding a block of several million (sparse) bigrams might thus have the same statistical effect as adding several thousand (denser) unigrams (individual words).

We now comment the optimal choice of L and computational complexity. If we assume fixed $\|\boldsymbol{\beta}^*\|_2$ and $n = O(q)$, which is all that would be required to keep the prediction error bounded asymptotically, then the optimal dimension of the min-wise projection scales as $L_b^* = O(q)$, considering b fixed here. This dimension will in general be a substantial reduction over the original dimension of the data, p, and would result in a correspondingly large reduction in the computational cost of regression. Indeed, ridge regression or the LAR algorithm (Efron et al., 2004) applied to $\mathbf{X}$ would have complexity $O(q^2 p)$, and one would expect that the Lasso (Tibshirani, 1996) would have similar computational cost. In contrast, OLS applied to $\mathbf{S}$ would only require $O(q^3)$ operations, an improvement of q/p. The discussion above considered an optimal choice of $L \approx L_b^*$. Even if we cannot afford to work with the optimal dimension L_b^* for computational reasons, the bound will still be useful for smaller values of L. The guarantee on prediction accuracy could not be obtained if, for example, simply a random subset of L predictors were chosen and the remaining ones discarded.

The dependence of the bound on b is also interesting: a minimum value occurs for $b = 1$. However, this would imply a larger value of L_b^*. Note the memory requirement for storing $\mathbf{S}$ would be $O(nL_b^* b)$ as b bits would be required to store the locations of each of the nL_b^* nonzeroes. We see that with a constraint on nbL or on the number of permutations L, larger values of b are more favourable, particularly with high sparsity, as this would tend to make $\|\boldsymbol{\beta}^*\|_b$ not much larger than $\|\boldsymbol{\beta}^*\|_2$. A different perspective on the optimal choice of b based on the variance of inner products of rows of $\mathbf{S}$ is taken in Li and König (2011), with similar conclusions.

4.1.2 Ridge regression

Instead of using a least-squares estimator on the transformed data matrix $\mathbf{S}$ we can also apply ridge regression (Hoerl and Kennard, 1970). For a given $\lambda > 0$, the regression coefficients are found by

$$(\hat{\alpha}_\lambda, \hat{\mathbf{b}}_\lambda) := \underset{(\alpha,\mathbf{b})\in\mathbb{R}\times\mathbb{R}^L}{\arg\min} \|\mathbf{Y} - \hat{\alpha}\mathbf{1} - \mathbf{S}\mathbf{b}\|_2^2 \;\text{ such that }\; \|\mathbf{b}\|_2^2 \le \lambda, \tag{14}$$

The theorem below gives a bound on the MSPE of $(\hat{\alpha}, \hat{\mathbf{b}}_\lambda)$.

Theorem 4 *There exist regularisation parameters λ depending on $\boldsymbol{\beta}^*$ and $\mathbf{S}$ such that*

$$\mathrm{MSPE}((\hat{\alpha}_\lambda, \hat{\mathbf{b}}_\lambda)) \;\le\; \sigma\sqrt{\frac{2C}{(1-2^{-b})n}}\|\boldsymbol{\beta}^*\|_2 + \frac{C}{2^b L(1-2^{-b})}\|\boldsymbol{\beta}^*\|_b^2 + \frac{\sigma^2}{n}.$$

Here the value of C is defined as in Theorem 3 by $C = (2-\delta)q$ for equal row sparsity δ and $C = q_{\min}\log\{4\log(L)/\delta_{\min}\}$ for $\kappa = \kappa_0$ and unequal row-sparsity.

The ridge regression result for large L is similar to that for OLS with an optimal L_b^*, though there is a small difference: the leading terms are $\sigma\|\boldsymbol{\beta}^*\|_2\sqrt{q/n}$ and $\sigma\|\boldsymbol{\beta}^*\|_b\sqrt{q/n}$ respectively. Ridge regression takes advantage of the fact that not only do we have a $\mathbf{b}^*$ such that $\mathbf{S}\mathbf{b}^*$ and $\mathbf{X}\boldsymbol{\beta}^*$ are close, we also know that there is a $\mathbf{b}^*$ with this property that has low ℓ_2-norm. Our bound on the expected squared ℓ_2-norm of $\mathbf{b}^*$ ((ii) in Theorem 1) does not depend much on b. In contrast, OLS only makes use of the approximation error result, (iii) in Theorem 1.

Note that when L is large, regardless of the value of b, ridge regression on $\mathbf{S}$ approximates a kernel ridge regression using the resemblance kernel (see Section 2.4). The MSPE of a kernel ridge regression with the resemblance kernel should of course not depend on b, and this observation largely agrees with our result.

Another key difference between ridge regression and OLS here is the following: achieving a good prediction error with OLS hinges on a careful choice of L. In contrast, with ridge regression, L can (and should) be chosen very large, from a purely statistical point of view. However, the constraint on the ℓ_2-norm of $\hat{\mathbf{b}}$ needs to be chosen carefully with ridge regression, typically by cross-validation. In practice, the number L of dimensions can be chosen as large as possible according to the available computational budget.

4.2 Logistic regression

We give an analogous result to Theorem 4 for classification problems under logistic loss. Let $\mathbf{x}_i \in [-1,1]^p$ and let $\mathbf{Y} \in \{0,1\}^n$ be an associated vector of class labels. We assume the model

$$Y_i \sim \text{Bernoulli}(p_i); \qquad \log\left(\frac{p_i}{1-p_i}\right) = \kappa(\delta_i)\mathbf{x}_i^T\boldsymbol{\beta}^*, \tag{15}$$

with the Y_i independent for $i = 1,\ldots,n$. Note that we have omitted the separate intercept term for simplicity.

Here we consider a linear classifier constructed by ℓ_2-constrained logistic regression. One can obtain a similar result for unconstrained logistic regression based on Lemma 6.6 of Bühlmann and van de Geer (2011), but we do not pursue this further here. Define

$$\hat{\mathbf{b}}_\lambda = \arg\min_{\mathbf{b}} \frac{1}{n}\sum_{i=1}^n \left[-Y_i\mathbf{s}_i^T\mathbf{b} + \log\{1+\exp(\mathbf{s}_i^T\mathbf{b})\}\right] \quad \text{such that} \quad \|\mathbf{b}\|_2^2 \le \lambda. \tag{16}$$

Let $\mathcal{E}(\hat{\mathbf{b}}_\lambda)$ denote the excess risk of $\hat{\mathbf{b}}_\lambda$ under logistic loss, so

$$\mathcal{E}(\hat{\mathbf{b}}_\lambda) = \frac{1}{n}\sum_{i=1}^n \left[-p_i\mathbf{s}_i^T\hat{\mathbf{b}}_\lambda + \log\{1+\exp(\mathbf{s}_i^T\hat{\mathbf{b}}_\lambda)\}\right] - \frac{1}{n}\sum_{i=1}^n \left[-p_i\kappa(\delta_i)\mathbf{x}_i^T\boldsymbol{\beta}^* + \log\{1+\exp(\kappa(\delta_i)\mathbf{x}_i^T\boldsymbol{\beta}^*)\}\right]. \tag{17}$$

We can now state the analogous result to Theorem 4.

Theorem 5 *Define $\tilde{p} \in \mathbb{R}$ by*

$$\tilde{p} := \frac{1}{n}\sum_{i=1}^n p_i(1-p_i) \le \frac{1}{2}. \tag{18}$$

Then we have that there exists a λ depending $\boldsymbol{\beta}^*$ and $\mathbf{S}$ such that

$$\mathbb{E}_{\mathbf{Y},\boldsymbol{\pi},\boldsymbol{\Psi}}\{\mathcal{E}(\hat{\mathbf{b}}_\lambda)\} \le \sqrt{\frac{2\tilde{p}C}{(1-2^{-b})n}}\|\boldsymbol{\beta}^*\|_2 + \frac{C}{2^{b+2}L(1-2^{-b})}\|\boldsymbol{\beta}^*\|_b^2.$$

Here the value of C is defined as in Theorem 3 by $C = (2-\delta)q$ for equal row sparsity δ and $C = q_{\min}\log\{4\log(L)/\delta_{\min}\}$ for $\kappa = \kappa_0$ and unequal row-sparsity.

The result illustrates that the usefulness of b-bit min-wise hashing is not limited to regression problems. In fact, most applications of are classification problems (Li and König, 2011) and our analysis of b-bit min-wise hashing here gives a theoretical explanation for its performance in these cases.

5. Interaction models

One of the compelling aspects of regression and classification with b-bit min-wise hashing is the fact that a particular form of interactions between variables can be fitted. This does not require any change in the procedure other than a possible increase in L. To be clear, in order to capture interactions with b-bit min-wise hashing, just as in the main effects case, we create a reduced matrix $\mathbf{S}$ and then fit a main effects model to $\mathbf{S}$. The dimension of the compressed data, 2^bL, can still be substantially smaller than the $O(p^2)$ number of coefficients that would need to be estimated if the interactions were modelled in the conventional way, and so the resulting computational advantage can be very large.

Note that in situations where the number of original predictors, p, may be manageable, including interactions explicitly can quickly become computationally infeasible. For example, if we start with, 10^5 variables, the two-way interactions number more than a billion. For larger values of p, even methods such as Random Forest (Breiman, 2001) or Rule Ensembles (Friedman and Popescu, 2008) would suffer similar computational problems.

We now describe a type of interaction model that can be fitted with b-bit min-wise hashing. Let $\mathbf{f}^* \in \mathbb{R}^n$ be given by

$$f_i^* = \sum_{k=1}^p X_{ik}\theta_k^{*,(1)} + \sum_{k,k_1=1}^p X_{ik}\mathbb{1}_{\{X_{ik_1}=0\}}\Theta_{k,k_1}^{*,(2)}, \quad i = 1,\ldots,n, \tag{19}$$

where $\boldsymbol{\theta}^{*,(1)} \in \mathbb{R}^p$ is a vector of coefficients for the main effects terms, and $\boldsymbol{\Theta}^{*,(2)} \in \mathbb{R}^{p\times p}$ is a matrix of coefficients for interactions whose diagonal entries are zero. As elsewhere in the paper, throughout this section we will assume that $\mathbf{X} \in [-1,1]^{n\times p}$. Note that if $\mathbf{X}$ were a binary matrix, then (19) parametrises (in fact over-parametrises) all linear combinations of bivariate functions of predictors; that is all possible two-way interactions are included in the model.

In general, the interaction model includes the tensor product of the set of original variables with the columns of an $n \times p$ matrix with ikth entry $\mathbb{1}_{\{X_{ik}=0\}}$. The value zero is thus given a special status and the model seems particularly appropriate in the sparse design setting we are considering here.

5.1 Approximation error

We will assume that the number of non-zero entries in each row of $\mathbf{X}$ is $q \geq 1$. However, we believe our proof techniques can be extended to the unequal sparsity and unknown row scaling scenario dealt with in Section 3.2. Furthermore, for technical reasons, we assume here that $p \geq 3$.

Let $\mathbf{\Theta}^*$ collect together $\boldsymbol{\theta}^{*,(1)}$ and $\mathbf{\Theta}^{*,(2)}$ and define the following norms analogously to (5):

$$\|\mathbf{\Theta}^*\| := \|\boldsymbol{\theta}^{*,(1)}\|_2 + \left(2(2-\delta)q \sum_{k,k_1,k_2} \left|\Theta^{*,(2)}_{kk_1}\Theta^{*,(2)}_{kk_2}\right|\right)^{1/2}, \tag{20}$$

$$\|\mathbf{\Theta}^*\|_b := \|\boldsymbol{\theta}^{*,(1)}\|_b + \left\{2(2-\delta)q\left(\sum_{k,k_1,k_2}\left|\Theta^{*,(2)}_{kk_1}\Theta^{*,(2)}_{kk_2}\right| + \delta(2^b-2)\sum_{k,k_1,k_2}\frac{\|\mathbf{X}_k\|_2^2}{n}\left|\Theta^{*,(2)}_{kk_1}\Theta^{*,(2)}_{kk_2}\right|\right)\right\}^{1/2}. \tag{21}$$

Theorem 6 *Suppose we have exactly q non-zero entries in each row of $\mathbf{X}$. Then there exists a vector $\mathbf{b}^* \in \mathbb{R}^{2^b L}$ with the following properties:*

(i) The approximation is unbiased, $\mathbb{E}_{\boldsymbol{\pi},\mathbf{\Psi}}(\mathbf{S}\mathbf{b}^*) = \mathbf{f}^*$.

(ii) The ℓ_2-norm is bounded by

$$\mathbb{E}_{\boldsymbol{\pi},\mathbf{\Psi}}(\|\mathbf{b}^*\|_2^2) \leq \frac{(2-\delta)q}{L(1-2^{-b})}\|\mathbf{\Theta}^*\|^2.$$

(iii) The approximation error is bounded by

$$\mathbb{E}_{\boldsymbol{\pi},\mathbf{\Psi}}(\|\mathbf{S}\mathbf{b}^* - \mathbf{f}^*\|_2^2)/n \leq \frac{(2-\delta)q}{2^b L(1-2^{-b})}\|\mathbf{\Theta}^*\|_b^2.$$

The bound on the approximation error in (iii) is most suited to situations where there are a fixed number of interaction terms, so

$$\sum_{k,k_1,k_2}\left|\Theta^{*,(2)}_{kk_1}\Theta^{*,(2)}_{kk_2}\right| = O(1). \tag{22}$$

Then we see that the contribution of the interaction terms to the bound on the approximation error is of order q^2. On the other hand, if we are considering a growing number of many small interaction terms, much tighter bounds than that given by (iii) can be obtained. The bounds above show in particular that the form of function given by (19) lies in the RKHS of the resemblance kernel and its RKHS norm is upper bounded by $(2-\delta)q\|\mathbf{\Theta}^*\|^2$; further details are given in the appendix Section B.

The results for interaction models corresponding to Theorems 3, 4 and 5 now follow.

5.2 Prediction error

We now present results for linear and logistic regression models where the signal involves interactions.

5.2.1 Linear regression models

Assume the model (10) and define the MSPE by (11) but in both cases with $\mathbf{X}\boldsymbol{\beta}^*$ now replaced by $\mathbf{f}^*$ (19). As in the previous section, we will assume that $\mathbf{X}$ has q non-zero entries in each row. When OLS estimation is used, we have the following result.

Theorem 7 *Let $(\hat{\alpha}, \hat{\mathbf{b}})$ be the least squares estimator (12). Then*

$$\text{MSPE}((\hat{\alpha},\hat{\mathbf{b}})) \;\leq\; \frac{(2-\delta)q}{2^b L(1-2^{-b})}\|\mathbf{\Theta}^*\|_b^2 + 2^b L\frac{\sigma^2}{n}.$$

To interpret the result, consider a situation where there are a fixed number of interaction and main effects of fixed size, so in particular (22) holds. Then treating b as fixed, the optimal L, $L^* = O(\sqrt{q^2 n/\sigma})$. If n, q and p increase by collecting new data and adding uninformative variables, then in order for the MSPE to vanish asymptotically, we require $q^2/n \to 0$. Compare this to the corresponding requirement of OLS applied to $\mathbf{X}$, that $p^2/n \to 0$. Particularly in situations of increasing variable sparseness, as discussed in Section 4.1.1, this can amount to a large statistical advantage.

The computational gains can be equally great. If, for example, $n \approx q^2$, then $L^* = O(q^2)$. If ridge regression were applied to $\mathbf{X}$ augmented by $O(p^2)$ interaction terms, the number of operations required would be $O(p^2q^4)$; OLS using $\mathbf{S}$ has complexity $O(q^6)$. If instead $n \approx p^2$, then regression with explicitly coded interaction terms would have complexity $O(p^6)$, whilst with the compressed data this would be reduced to $O(p^4q^2)$.

As in the main effects case, the ridge regression result is similar.

Theorem 8 *Let the ridge regression estimator be given by* (14)*. There exists λ depending on $\mathbf{f}^*$ and $\mathbf{S}$ such that we have*

$$\text{MSPE}((\hat{\alpha},\hat{\mathbf{b}})) \;\leq \sigma\sqrt{\frac{(2-\delta)q}{n(1-2^{-b})}}\|\mathbf{\Theta}^*\| + \frac{(2-\delta)q}{2^b L(1-2^{-b})}\|\mathbf{\Theta}^*\|_b^2 + \frac{\sigma^2}{n}.$$

Similarly to Theorem 4 the result here suggests choosing a large L is always better from a statistical point of view. However, for computational reasons, it may not be possible to take L much larger than L^*.

5.2.2 Logistic regression

Here we assume the model (15) and define the excess risk by (17), but in both cases with $\mathbf{X}\boldsymbol{\beta}^*$ replaced by $\mathbf{f}^*$.

Theorem 9 *Define $\tilde{p} \in \mathbb{R}$ as in* (18) *and the ℓ_2-penalised logistic regression estimator as in* (16)*. Then we have that there exists λ such that*

$$\mathbb{E}_{\mathbf{Y},\boldsymbol{\pi},\mathbf{\Psi}}\{\mathcal{E}(\hat{\mathbf{b}}_\lambda)\} \leq \sigma\sqrt{\frac{\tilde{p}(2-\delta)q}{n(1-2^{-b})}}\|\mathbf{\Theta}^*\| + \frac{(2-\delta)q}{2^{b+2}L(1-2^{-b})}\|\mathbf{\Theta}^*\|_b^2.$$

One could continue to look at higher-order interaction models by adding three-way interactions in (19) and adapting (20) and (21) in suitable ways. However, being able to show that two-way interaction models can be fitted with b-bit min-wise hashing may well be sufficient for most applications.

6. Extensions

We now describe some extensions to the methodology.

6.1 Variable importance

Typically prediction, rather than model selection, is the primary goal in large-scale applications with sparse data, one reason for this being that we cannot expect a very small subset of variables to approximate the signal well when the design matrix is sparse. Nevertheless, it is often illuminating to study the influence of specific variables or look for the variables that have the largest influence on predictions. Indeed, such study is often undertaken following applications of Random Forest (Breiman, 2001), where several variable importance measures allow practitioners to better interpret the fits produced.

We now describe how importance measures can be obtained for b-bit min-wise hashing as described in Section 2.3. Let $\hat{f} : \mathbb{R}^p \to \mathbb{R}$ be the regression function created following regression on b-bit min-wise hashed data, and let $\hat{f}_i := \hat{f}(\mathbf{x}_i)$. Furthermore, for $k = 1, \ldots, p$, let $\hat{f}^{(-k)} := \hat{f}(\mathbf{x}_i^{(-k)})$, where $\mathbf{x}_i^{(-k)}$ is equal to $\mathbf{x}_i$ but with kth component set to zero.

The vector $\hat{\mathbf{f}} - \hat{\mathbf{f}}^{(-k)}$ is the difference in predictions obtained when fitting to $\mathbf{X}$, and those obtained when fitting to $\mathbf{X}$ with the kth column set to zero. When the underlying model in $\mathbf{X}$ contains only main effects (10) and no structural error is present, we might expect that

$$\hat{\mathbf{f}} - \hat{\mathbf{f}}^{(-k)} \approx \boldsymbol{\beta}_k^* \mathbf{X}_k.$$

To obtain a measure of variable importance, one could look at the ℓ_2-norm of $\hat{\mathbf{f}} - \hat{\mathbf{f}}^{(-k)}$, for example (Breiman, 2001).

The difference in predictions can be computed relatively easily by considering the $n \times 2^b L$ matrix $\tilde{\mathbf{S}}$ with entries given by $\tilde{S}_{ilc} = \tilde{S}_{i(c+(l-1)2^b} = X_{i\tilde{H}_{il}} \mathbb{1}_{\{\Psi_{\tilde{H}_{il}l} = c\}}$, where

$$\tilde{H}_{il} := \underset{k \in \mathbf{z}_i \setminus H_{il}}{\arg\min}\, \pi_l(k).$$

Thus $\tilde{H}_{il}$ is the variable index in $\mathbf{z}_i$ whose value under permutation π_l is second smallest among $\{\pi_l(k) : k \in \mathbf{z}_i\}$. If $\mathbf{z}_i \setminus H_{il} = \emptyset$, we simply set $\tilde{S}_{il} = 0$. Then

$$\hat{f}_i - \hat{f}_i^{(-k)} = \sum_{l=1}^{L} \mathbb{1}_{\{H_{il}=k\}} \sum_{c=1}^{2^b} (S_{ilc} - \tilde{S}_{ilc}) \hat{b}_{lc}. \tag{23}$$

Note that we only need to store the $n \times L$ matrix $\mathbf{H}$ and $n \times 2^b L$ matrices $\mathbf{S}$ and $\tilde{\mathbf{S}}$ to compute the variable importance for all variables; moreover the latter matrices only have at most nL non-zero entries each.

Interaction effects are not directly visible, but do manifest themselves in the form of a higher variability among $\{\hat{f}_i - \hat{f}_i^{(-k)} : \mathbf{x}_i \approx \mathbf{x}\}$, for any given value of $\mathbf{x}$, if variable k is involved in an interaction term. In principle, one could attempt to detect this increased variability, but further investigation of this is beyond the scope of the current work.

6.2 Other fitting procedures

Here we have only considered OLS, ridge regression and ℓ_2-penalised logistic regression as prediction methods after reducing the design matrix. However, it is also conceivable that other fitting procedures could be suitable. In particular, it would be interesting to look at matching pursuit, boosting and the Lasso, for which results in (Tropp, 2004; Bühlmann, 2006; Van De Geer, 2008) could be leveraged. Matching pursuit would have the computational advantage that the entire $\mathbf{S}$ matrix would not need to be held in memory. Instead, one could create the columns during the fitting process. Such an approach may be useful for problems where the dimension of the hashing-matrix, $2^b L$, needs to be very large to achieve a desired predictive accuracy.

7. Discussion

In this paper we have derived approximation error bounds for b-bit min-wise hashing. We were able to show that not only does b-bit min-wise hashing take advantage of sparsity in the design matrix computationally, it is also able to exploit this for improved statistical performance. In particular, the MSPE of regression following dimension reduction by b-bit min-wise hashing is of the form $\sqrt{q/n}\|\boldsymbol{\beta}^*\|_2$ if the data follow a linear model with coefficient vector $\boldsymbol{\beta}^*$ and q is the average number of non-zero variables for an observation. The linear model can then be well-approximated by the low-dimensional b-bit min-wise hashed data if the norm of $\|\boldsymbol{\beta}^*\|_2$ is low, as occurs, for example if the signal is approximately replicated in distinct blocks of variables.

In addition, we have shown that more complicated models such as interaction models can be fitted by a regression on the hashed data matrix that contains only main effects. Though a larger dimension L of the hashed data may be required than when approximating a main effects model, no further changes are needed to the procedure.

These bounds also reveal some of the predictive properties of the resemblance kernel, and provide an insight into the sorts of regression functions that have small norm in its associated RKHS. More generally, we believe that random feature expansions may well be useful as a theoretical tool to understand properties of otherwise intractable kernels. We expect to see more extensions and applications b-bit min-wise hashing and other random feature expansions, both as computational and theoretical tools, in the future.

Acknowledgments

The first author was supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1 and an EPSRC programme grant.

Appendix A. Approximation error results

In this section we prove results on the approximation error presented in the main text (Theorems 1, 2 and 6) as well as an additional result on the approximation error of linear signals when row sparsity is not necessarily equal (Theorem 12).

A.1 Preliminary results

We will let q_i be the number of non-zeroes in the ith row of $\mathbf{X}$ and define $\delta_i = q_i/p$. We will assume that $q_i \geq 1$ for all i. For the proofs of results on approximation error in settings with just main effects, we will make use of the following lemma. This lemma formalises the ideas of the discussion at the end of Section 3.2, that the elements of $\mathbf{M}$ behave rather like geometric random variables.

Lemma 10 *There exist random functions $\{g_l(k)\}_{l=1,\ldots,L,\, k=1,\ldots,p}$ defined on the same probability space as the permutations $\boldsymbol{\pi}$ with the following properties:*

(i) The random variables $\{g_1(k)\}_{k=1,\ldots,p}, \ldots, \{g_L(k)\}_{k=1,\ldots,p}$ are i.i.d. and are independent of $\boldsymbol{\Psi}$.

(ii) The rank of $g_l(k)$ among $g_l(1), \ldots, g_l(p)$ taken in increasing order is $\pi_l(k)$.

(iii) Marginally $g_l(k) \sim Geo(p^{-1})$.

(iv) $G_{il} := \min_{k \in \mathbf{z}_i} g_l(k) = g_l(H_{il}) \sim Geo(\delta_i)$.

(v) $\mathbf{G}$ and $\mathbf{H}$ are independent.

Proof First consider generating permutations $\boldsymbol{\pi}$ in the following way. Let $m \in \mathbb{N}$ and let $\sigma_1^{(m)}, \ldots, \sigma_L^{(m)}$ be L i.i.d. random permutations of $\{1, \ldots, mp\}$. For $k = 1, \ldots, p$, let

$$g_l^{(m)}(k) = \min_{a=0,\ldots,m-1} \sigma_l^{(m)}(k + ap).$$

Note that the $g_l^{(m)}(k)$ are all distinct and any ordering of them is equally likely so they define a random permutation of $\{1, \ldots, p\}$. Furthermore, for $j = 1, \ldots, mp - m + 1$,

$$\mathbb{P}(g_l^{(m)}(k) = j) = \binom{mp - j}{m - 1} \Big/ \binom{mp}{m} = \frac{1}{p}\left(1 - \frac{1 - m^{-1}}{p - m^{-1}}\right) \cdots \left(1 - \frac{1 - m^{-1}}{p - (j-1)m^{-1}}\right).$$

Thus

$$\mathbb{P}(g_l^{(m)}(k) = j) \to \frac{1}{p}\left(1 - \frac{1}{p}\right)^{j-1}$$

as $m \to \infty$ for $j = 1, 2, \ldots$. Similarly $G_{il}^{(m)} := \min_{k \in \mathbf{z}_i} g_l^{(m)}(k)$ has $\mathbb{P}(G_{il}^{(m)} = j) \to \delta_i(1 - \delta_i)^{j-1}$ as $m \to \infty$. Note that $\mathbf{G}^{(m)}$ and $\mathbf{H}$ are independent. Thus

$$\{g_l^{(m)}(k)\}_{l=1,\ldots,L, k=1,\ldots,p} \xrightarrow{d} \{g_l(k)\}_{l=1,\ldots,L, k=1,\ldots,p}$$

as $m \to \infty$ with the random variables $g_l(k)$ having the properties given in the statement of the lemma. ■

In the proofs which follow, we will consider the permutations as having been generated as described by Lemma 10. We will let $\pi = \pi_1$, $M_i = M_{i1}$, $g = g_1$, $G_1 = G_{i1}$, $H_i = H_{i1}$ and $\boldsymbol{\psi} = \boldsymbol{\Psi}_1$. Let $C = 2^b$, $\nu = 2^{-b}$.

The next lemma introduces the general form of $\mathbf{b}^*$ that we will use for the main effects results. It also establishes results on the mean and variance of the approximation and gives a bound on $\mathbb{E}(\|\mathbf{b}^*\|_2^2)$; these will form the basis of the theorems to follow.

Lemma 11 *For a given sequence of weights $\{w_j\}_{j=1}^{\infty}$, let $\tilde{\mathbf{b}}^* \in \mathbb{R}^{LC}$ be given by*

$$\tilde{b}^*_{lc} = \frac{1}{L}\sum_{k=1}^{p} \beta^*_k \frac{\mathbb{1}_{\{\Psi_{lk} = c\}} - \nu}{1 - \nu} w_{g_l(k)}$$

and let $\mathbf{b}^* = \mathbb{E}(\tilde{\mathbf{b}}^* | \boldsymbol{\pi})$. We have the following.

(i)

$$\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\mathbf{s}_i^T \mathbf{b}^*) = \frac{1}{p}\mathbf{x}_i^T \boldsymbol{\beta}^* \sum_{\ell=1}^{\infty} (1 - \delta_i)^{\ell - 1} w_\ell.$$

(ii)

$$\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\|\mathbf{b}^*\|_2^2) \leq \frac{1}{pL(1-\nu)} \|\boldsymbol{\beta}^*\|_2^2 \sum_{\ell=1}^{\infty} w_\ell^2. \tag{24}$$

(iii)

$$\mathrm{Var}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\mathbf{s}_i^T \mathbf{b}^*) \leq \frac{1}{pL(1-\nu)} \left(\nu \|\boldsymbol{\beta}^*\|_2^2 + (1 - 2\nu) \sum_{k=1}^{p} X_{ik}^2 \beta_k^{*2}\right) \sum_{\ell=1}^{\infty} w_\ell^2. \tag{25}$$

Proof First note that

$$\mathbb{E}\left(\frac{\mathbb{1}_{\{\psi_k = \psi_j\}} - \nu}{1 - \nu} \middle| \psi_j\right) = \begin{cases} 1 & \text{if } k = j \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

$$\mathbb{E}\left(\frac{\mathbb{1}_{\{\psi_k = \psi_j\}} - \nu}{1 - \nu} \frac{\mathbb{1}_{\{\psi_\ell = \psi_j\}} - \nu}{1 - \nu} \middle| \psi_j\right) = \begin{cases} 1 & \text{if } k = \ell = j \\ 0 & \text{if } k \neq \ell \\ \frac{\nu}{1-\nu} & \text{otherwise.} \end{cases} \tag{27}$$

For (i), we have

$$\begin{aligned} \mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\mathbf{s}_i^T \mathbf{b}^*) &= \mathbb{E}_{g,\psi}\left(\sum_{c=1}^{C}\sum_{j=1}^{p} X_{ij} \mathbb{1}_{\{H_i = j\}} \mathbb{1}_{\{\psi_j = c\}} \sum_{k=1}^{p} \beta_k^* \frac{\mathbb{1}_{\{\psi_k = c\}} - \nu}{1 - \nu} w_{g(k)}\right) \\ &= \mathbb{E}_g\left(\sum_{k=1}^{p} X_{ik} \mathbb{1}_{\{H_i = k\}} \beta_k^* w_{g(k)}\right) \\ &= \frac{1}{q_i}\sum_{k=1}^{p} X_{ik} \beta_k^* \mathbb{E}(w_{G_i}), \end{aligned}$$

where to arrive at the second line we used (26).

Turning to (ii), note that each component of $\mathbf{b}^*$ has mean zero and so

$$\mathbb{E}(b^{*2}_{lc}) = \mathrm{Var}(b^*_{lc}) = \mathrm{Var}\{\mathbb{E}(\tilde{b}^*_{lc} | \boldsymbol{\pi})\} \leq \mathrm{Var}(\tilde{b}^*_{lc}).$$

Now we have

$$\mathbb{E}_{g_1,\ldots,g_L,\boldsymbol{\Psi}}\|\tilde{\mathbf{b}}^*\|_2^2 = \frac{1}{L}\sum_{c=1}^{C}\sum_{k,\ell}\beta_k^*\beta_\ell^*\mathbb{E}\Big(\frac{\mathbb{1}_{\{\psi_k=c\}}-\nu}{1-\nu}\frac{\mathbb{1}_{\{\psi_\ell=c\}}-\nu}{1-\nu}\Big)\mathbb{E}(w_{g(k)}w_{g(\ell)})$$

Using (27), we get

$$\mathbb{E}_{g_1,\ldots,g_L,\boldsymbol{\Psi}}\|\tilde{\mathbf{b}}^*\|_2^2 = \frac{1}{L(1-\nu)}\sum_k \beta_k^{*2}\mathbb{E}(w_{g(k)}^2) \le \frac{1}{pL(1-\nu)}\|\boldsymbol{\beta}^*\|_2^2\sum_{\ell=1}^{\infty}w_\ell^2.$$

For (iii) we argue as follows.

$$\begin{aligned}\mathrm{Var}(\mathbf{s}_i^T\mathbf{b}^*) &\le \mathrm{Var}(\mathbf{s}_i^T\tilde{\mathbf{b}}^*)\\ &\le \frac{1}{L}\mathbb{E}_{g,\psi}\Big(X_{iH_i}^2\sum_{k,\ell}\beta_k^*\beta_\ell^*\frac{\mathbb{1}_{\{\psi_k=\psi_{H_i}\}}-\nu}{1-\nu}\frac{\mathbb{1}_{\{\psi_\ell=\psi_{H_i}\}}-\nu}{1-\nu}w_{g(k)}w_{g(\ell)}\Big)\end{aligned}$$

Using (27) and the fact that $\mathbf{X}\in[-1,1]^{n\times p}$, we have

$$\begin{aligned}\mathrm{Var}(\mathbf{s}_i^T\mathbf{b}^*) &\le \frac{1}{L}\mathbb{E}\Big\{X_{iH_i}^2\Big(\frac{\nu}{1-\nu}\sum_{k=1}^{p}(\beta_k^*)^2w_{g(k)}^2+\frac{1-2\nu}{1-\nu}(\beta_{H_i}^*)^2w_{G_i}^2\Big)\Big\} \qquad (28)\\ &\le \frac{1}{L(1-\nu)}\Big\{\nu\sum_{k=1}^{p}\beta_k^{*2}\mathbb{E}(w_{g(k)}^2)+\frac{1-2\nu}{q_i}\mathbb{E}(w_{G_i}^2)\sum_{k=1}^{p}X_{ik}^2\beta_k^{*2}\Big\}.\end{aligned}$$

The result then follows as

$$\sum_{\ell=1}^{\infty}w_\ell^2 \ge \mathbb{E}(w_{g(k)}^2) = \frac{1}{p}\sum_{\ell=1}^{\infty}w_\ell^2\Big(1-\frac{1}{p}\Big)^{\ell-1} \ge \frac{\delta_i}{q_i}\sum_{\ell=1}^{\infty}w_\ell^2(1-\delta_i)^{\ell-1} = \frac{\mathbb{E}(w_{G_i}^2)}{q_i}.$$

■

A.2 Proof of Theorem 1

We use a $\mathbf{b}^*$ and $\tilde{\mathbf{b}}^*$ as in Lemma 11 but here we choose the weights w_ℓ so as to minimise $\sum_{\ell=1}^{\infty}w_\ell^2$ (a term which features in our upper bounds on the variance and $\mathbb{E}(\|\mathbf{b}^*\|_2^2)$) subject to the unbiasedness constraint (i). The unbiasedness constraint amounts to

$$\sum_{\ell=1}^{\infty}(1-\delta)^{\ell-1}w_\ell = p.$$

Performing the minimisation with this constraint yields

$$w_\ell = p\frac{(1-\delta)^{\ell-1}}{\sum_{\ell=1}^{\infty}(1-\delta)^{2\ell-2}}.$$

With this choice we have

$$\sum_{\ell=1}^{\infty}w_\ell^2 = p^2\Big(\sum_{\ell=1}^{\infty}(1-\delta)^{2\ell-2}\Big)^{-1} = p^2\{1-(1-\delta)^2\} = (2-\delta)qp.$$

Substituting into (24) and (25) then yields the result.

A.3 Proof of Theorem 2

We use a $\mathbf{b}^*$ and $\tilde{\mathbf{b}}^*$ as in Lemma 11 but here we take

$$w_{\ell+1} = p(-1)^\ell\frac{\kappa^{(\ell)}(1)}{\ell!}\{\mathbb{1}_{\{\ell\le\lfloor m\rfloor\}}+(m-\lfloor m\rfloor)\mathbb{1}_{\{\ell=\lceil m\rceil\}}\}$$

where $m>0$ is a parameter to be chosen. Thus the weights correspond to coefficients from a truncated Taylor series expansion of κ about 1. We have

$$\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}[\{(\delta_{\min}/\delta_i)^a\mathbf{x}_i^T\boldsymbol{\beta}^*-\mathbf{s}_i^T\mathbf{b}^*\}^2] = \{(\delta_{\min}/\delta_i)^a\mathbf{x}_i^T\boldsymbol{\beta}^*-\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\mathbf{s}_i^T\mathbf{b}^*)\}^2+\mathrm{Var}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\mathbf{s}_i^T\mathbf{b}^*).$$

We first bound the variance term by bounding the squared sum of the sequence of weights. To this end, we note that by Lemma 20

$$\frac{\delta_{\min}^{-2a}}{p^2}\sum_{\ell=1}^{\infty}w_\ell^2 \le 1+a^2+a^2e^{2a}\Big(\sum_{\ell=2}^{\lfloor m\rfloor}\frac{1}{\ell^{2(1-a)}}+\frac{m-\lfloor m\rfloor}{\lceil m\rceil^{2(1-a)}}\Big).$$

Now

$$\begin{aligned}\sum_{\ell=2}^{\lfloor m\rfloor}\frac{1}{\ell^{2(1-a)}}+\frac{m-\lfloor m\rfloor}{\lceil m\rceil^{2(1-a)}} &\le \int_1^m\frac{1}{\ell^{2(a-1)}}d\ell\\ &= \begin{cases}\frac{m^{2a-1}-1}{2a-1} & \text{if } a\ne 1/2\\ \log(m) & \text{if } a=1/2.\end{cases}\end{aligned}$$

Let

$$\tau_a(m) = \begin{cases}e\log(me^{5/e})/4 & \text{if } a=1/2,\\ a^2e^{2a}m^{2a-1}/(2a-1) & \text{if } 1/2<a\le 1.\end{cases}$$

Then

$$\sum_{\ell=1}^{\infty}w_\ell^2 \le p^2\delta_{\min}^{2a}\tau_a(m). \qquad (29)$$

The variance is then at most

$$\delta_{\min}^{2a}\tau_a(m)\frac{p}{L(1-\nu)}\Big(\nu\|\boldsymbol{\beta}^*\|_2^2+(1-2\nu)\sum_{k=1}^{p}X_{ik}^2\beta_k^{*2}\Big).$$

Turning now to the bias term, note first that by (i) of Lemma 11, this is equal to

$$(\mathbf{x}_i^T\boldsymbol{\beta}^*)^2\Big\{(\delta_{\min}/\delta_i)^a-\frac{1}{p}\sum_{\ell=1}^{\infty}(1-\delta_i)^{\ell-1}w_\ell\Big\}^2. \qquad (30)$$

We see this is bounded above by

$$\delta_{\min}^{2a}(\mathbf{x}_i^T\boldsymbol{\beta}^*)^2\Big\{ae^a\Big(\sum_{\ell=\lceil m\rceil}^{\infty}(1-\delta_i)^\ell\frac{1}{\ell^{1-a}}\Big)\Big\}^2.$$

Now

$$\sum_{\ell=\lceil m\rceil}^{\infty}(1-\delta_i)^{\ell}\frac{1}{\ell^{1-a}} \le \frac{e^{-\delta_i m}}{m^{1-a}\delta_i}.$$

By the Cauchy–Schwarz inequality (assuming $X_{ij}\in[-1,1]$)

$$\frac{(\mathbf{x}_i^T\boldsymbol{\beta}^*)^2}{\delta_i} = \frac{1}{\delta_i}\Big(\sum_{k\in\mathbf{z}_i} X_{ik}\beta_k^*\Big)^2 \le p\sum_{k=1}^{p} X_{ik}^2\beta_k^{*2} \le p\|\boldsymbol{\beta}^*\|_2^2.$$

Thus the squared bias is at most

$$\frac{p}{1-\nu}\frac{a^2e^{2a}}{m^{1-2a}}\max_{i=1,\ldots,n}\Big(\frac{e^{-2\delta_i m}}{m\delta_i}\Big)\Big(\nu\|\boldsymbol{\beta}^*\|_2^2 + (1-2\nu)\sum_{k=1}^{p}X_{ik}^2\beta_k^{*2}\Big).$$

Therefore the MSE (now averaging over the observations) is bounded by the minimum over $m>0$ of

$$\frac{p}{L(1-2^{-b})}\delta_{\min}^{2a}\Big\{\tau_a(m) + \frac{a^2e^{2a}}{m^{1-2a}}\max_{i=1,\ldots,n}\Big(\frac{e^{-2\delta_i m}}{m\delta_i}\Big)\Big\}\|\boldsymbol{\beta}^*\|_b^2.$$

For $a=1/2$, we set $m=\log(L)/\{2\delta_{\min}\}$. This yields

$$\begin{aligned}\min_{m>0}\Big\{\tau_{1/2}(m) + \frac{Le}{4}\max_{i=1,\ldots,n}\Big(\frac{e^{-2\delta_i m}}{m\delta_i}\Big)\Big\} &\le \frac{e}{4}\Big\{\log\Big(\frac{\log(L)e^{5/e}}{2\delta_{\min}}\Big) + \frac{2}{\log(L)}\Big\}\\ &\le \log\{4\log(L)/\delta_{\min}\}\end{aligned}$$

provided $L\ge 10$ and $\delta_{\min}\le 1/2$. Finally the bound for $a>1/2$ comes from setting

$$m = \frac{1}{2}\log\{2(2a-1)L\}/\delta_{\min}$$

which gives

$$\begin{aligned}\min_{m>0}\Big\{\tau_a(m) + \frac{La^2e^{2a}}{m^{1-2a}}\max_{i=1,\ldots,n}\Big(\frac{e^{-2\delta_i m}}{m\delta_i}\Big)\Big\} &\le \frac{\delta_{\min}^{1-2a}a^2e^{2a}}{2^{2a-1}(2a-1)}[\log\{2(2a-1)L\}]^{2a-2}\log\{2(2a-1)eL\}\\ &\le \frac{4\delta_{\min}^{1-2a}}{1-2a}[\log\{2(2a-1)L\}]^{2a-1}\end{aligned}$$

for $L\ge 2/(1-2a)$. Using the bounds on τ_a with these choices of m and (29), we obtain the bounds on $\mathbb{E}(\|\mathbf{b}^*\|_2^2)$ by substituting into (24).

A.4 Unequal row sparsity and constant row-scaling

Here we prove results indicated after the presentation of Theorem 1 in Section 3.1. When the scaling function is simply the constant 1, the spread of the δ_i becomes more critical in determining how well the signal can be approximated. Define

$$\bar\delta = \frac{1}{n}\sum_{i=1}^{n}\delta_i,$$

$$\mathcal{V}(\boldsymbol{\delta}) = \frac{1}{\|\mathbf{X}\boldsymbol{\beta}^*\|_2^2}\sum_{i=1}^{n}(\mathbf{x}_i^T\boldsymbol{\beta}^*)^2(\delta_i-\bar\delta)^2.$$

Theorem 12 *Suppose*

$$2^bL(1-2^{-b}) \le \frac{p(2\bar\delta)^3\|\boldsymbol{\beta}^*\|_b^2}{\|\mathbf{X}\boldsymbol{\beta}^*\|_2^2\mathcal{V}(\boldsymbol{\delta})/n}. \tag{31}$$

Then there exists $\mathbf{b}^*\in\mathbb{R}^L$ *such that the approximation error satisfies*

$$\frac{1}{n}\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}\{\|\mathbf{X}\boldsymbol{\beta}^* - \mathbf{S}\mathbf{b}^*\|_2^2\} \le \frac{6p\bar\delta}{2^bL(1-2^{-b})}\|\boldsymbol{\beta}^*\|_b^2, \tag{32}$$

and

$$\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\|\mathbf{b}^*\|_2^2) \le \frac{2\bar q}{L(1-2^{-b})}\|\boldsymbol{\beta}^*\|_2^2. \tag{33}$$

Provided 2^bL is not too large, we recover essentially the same approximation error bound as Theorem 1 up to a constant factor, but with the row sparsity replaced by the average row sparsity $\bar\delta$. In the simple situation where the entries of $\mathbf{X}$ are realisations of i.i.d. Bernoulli random variables with probability δ, we would have $\bar\delta\approx\delta$, $\|\mathbf{X}\boldsymbol{\beta}^*\|_2^2/n\approx\delta\|\boldsymbol{\beta}^*\|_2^2$ and $\mathcal{V}(\boldsymbol{\delta})\approx\delta/p$. Substituting these values into the requirement on 2^bL shows that the condition reduces to $2^bL\le 8p^2\delta\{1+(2^b-2)\delta\}$. Note that typically one would choose 2^bL of the order $\bar\delta p$. More generally, provided $\mathcal{V}(\boldsymbol{\delta})$ and $\|\mathbf{X}\boldsymbol{\beta}^*\|_2^2/\|\boldsymbol{\beta}^*\|_2^2$ are small, we can expect that the bound of Theorem 1 will hold true, up to a constant factor.

Proof of Theorem 12

We use a $\mathbf{b}^*$ and $\tilde{\mathbf{b}}^*$ as in Lemma 11 taking

$$w_\ell = p(1-\bar\delta)^{\ell-1}\mathbb{1}_{\{\ell\le m\}}\frac{\bar\delta(2-\bar\delta)}{1-(1-\bar\delta)^{2m}}.$$

where $m\in\mathbb{N}$ is a parameter to be chosen. This gives

$$\frac{1}{p^2}\sum_{\ell=1}^{\infty}w_\ell^2 = \frac{\bar\delta(2-\bar\delta)}{1-(1-\bar\delta)^{2m}},$$

which gives us a bound on the variance term.

Lemma 11 (i) gives the expression for the bias term. To bound this, first note that

$$\frac{1}{p}\sum_{\ell=1}^{m}(1-\bar\delta)^{\ell-1}w_\ell = 1.$$

Next

$$\begin{aligned}\Big[\sum_{\ell=1}^{m}(1-\bar\delta)^{\ell-1}\{(1-\bar\delta)^{\ell-1}-(1-\delta_i)^{\ell-1}\}\Big]^2 &= (\delta_i-\bar\delta)^2\Big[\sum_{\ell=1}^{m}(1-\bar\delta)^{\ell-1}\sum_{k=0}^{\ell-2}(1-\bar\delta)^k(1-\delta_i)^{\ell-2-k}\Big]^2\\ &\le (\delta_i-\bar\delta)^2\Big(\sum_{\ell=1}^{m}(1-\bar\delta)^{\ell-1}(\ell-1)\Big)^2\\ &= \min\Big\{\frac{m(m-1)}{2},\frac{1}{\bar\delta^2}\Big\}(\delta_i-\bar\delta)^2.\end{aligned}$$

Also note that as

$$(1-\bar{\delta})^{2m} \leq 1 - 2m\bar{\delta} + m(2m-1)\bar{\delta}^2$$

we have

$$\begin{aligned}\frac{\bar{\delta}(2-\bar{\delta})}{1-(1-\bar{\delta})^{2m}} &\leq \frac{2}{2m - m(2m-1)\bar{\delta}}\mathbb{1}_{\{m \leq 1/(2\bar{\delta})\}} + \frac{2}{1/\bar{\delta} - (1/\bar{\delta}-1)/2}\mathbb{1}_{\{m > 1/(2\bar{\delta})\}} \\ &\leq \max\left(\frac{2}{m+1/2}, \frac{4\bar{\delta}}{1+\bar{\delta}}\right)\end{aligned}$$

and for $m \leq 1/(2\bar{\delta}) + 1/2$,

$$\frac{m(m-1)}{2}\max\left(\frac{2}{m+1/2}, \frac{4\bar{\delta}}{1+\bar{\delta}}\right) \leq (m-1/2)\mathbb{1}_{\{1 < m \leq 1/(2\bar{\delta})+1/2\}}.$$

Thus the overall approximation error is bounded above by the minimum over $m = 1, 2, \ldots, \lfloor 1/(2\bar{\delta}) + 1/2 \rfloor$ of

$$\mathbb{1}_{\{m>1\}}(m-1/2)^2\frac{1}{n}\|\mathbf{X}\boldsymbol{\beta}^*\|_2^2\mathcal{V}(\boldsymbol{\delta}) + \max\left(\frac{2}{m+1/2}, 4\bar{\delta}\right)\frac{p}{2^b L(1-2^{-b})}\|\boldsymbol{\beta}^*\|_b^2,$$

which in turn is bounded by the minimum over $m \in [0, 1/(2\bar{\delta})]$ of

$$m^2\frac{1}{n}\|\mathbf{X}\boldsymbol{\beta}^*\|_2^2\mathcal{V}(\boldsymbol{\delta}) + \frac{2}{m}\frac{p}{2^b L(1-2^{-b})}\|\boldsymbol{\beta}^*\|_b^2. \tag{34}$$

Optimising over $m > 0$ in the above then gives

$$m = \min\left\{\left(\frac{p\|\boldsymbol{\beta}^*\|_b^2}{2^b L(1-2^{-b})\|\mathbf{X}\boldsymbol{\beta}^*\|_2^2\mathcal{V}(\boldsymbol{\delta})/n}\right)^{1/3}, \frac{1}{2\bar{\delta}}\right\}.$$

The condition on L (31) ensures that the minimum is achieved at $1/(2\bar{\delta})$. Substituting this value of m into (34) then gives (32). For (33) we note that

$$\sum_{\ell=1}^{\infty} w_\ell^2 \leq 2p^2\bar{\delta};$$

the result follows using Lemma 11 (ii).

A.5 Proof of Theorem 6

We let $\mathbf{b}^* = \mathbf{b}^{*,(1)} + \mathbf{b}^{*,(2)}$ where $\mathbf{b}^{*,(1)}$ is chosen in line with Theorem 1. Explicitly, let $\mathbf{b}^{*,(1)} = \mathbb{E}(\tilde{\mathbf{b}}^*|\boldsymbol{\pi})$ where

$$\tilde{b}_{lc}^* = \frac{p}{L}\sum_{k=1}^{p}\theta_k^{*,(1)}\frac{\mathbb{1}_{\{\Psi_{lk}=c\}} - \nu}{1-\nu}\frac{(1-\delta)^{g_l(k)-1}}{\sum_{\ell=1}^{\infty}(1-\delta)^{2\ell-2}}.$$

We construct $\mathbf{b}^{*,(2)}$ to approximate the interactions as follows. Let

$$b_{lc}^{*,(2)} = \frac{pq}{L}\sum_{k=1}^{p}\frac{\mathbb{1}_{\{\Psi_{lk}=c\}} - \nu}{1-\nu}\sum_{k_1=1}^{p}\Theta_{kk_1}^{*,(2)}\mathbb{1}_{\{\pi_l(k_1)<\pi_l(k)\}}w_{\pi_l(k)},$$

where $\mathbf{w} \in \mathbb{R}^p$ is a vector of weights to be chosen such that

$$\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\mathbf{s}_i^T\mathbf{b}^{*,(2)}) = \sum_{k,k_1} X_{ik}\mathbb{1}_{\{X_{ik_1}=0\}}\Theta_{kk_1}^{*,(2)}. \tag{35}$$

We compute

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\mathbf{s}_i^T\mathbf{b}^{*,(2)}) &= \frac{pq}{L}\sum_{l=1}^{L}\sum_{c=1}^{C}\mathbb{E}_{\pi_l,\boldsymbol{\Psi}_l}\left(S_{ilc}\sum_{k=1}^{p}\frac{\mathbb{1}_{\{\Psi_{kl}=c\}}-\nu}{1-\nu}\sum_{k_1=1}^{p}\Theta_{kk_1}^{*,(2)}\mathbb{1}_{\{\pi_l(k_1)<\pi_l(k)\}}w_{\pi_l(k)}\right) \\ &= pq\mathbb{E}_{\pi,\boldsymbol{\psi}}\left(\sum_{c=1}^{C}\sum_{j=1}^{p}X_{ij}\mathbb{1}_{\{H_i=j\}}\mathbb{1}_{\{\psi_j=c\}}\sum_{k=1}^{p}\frac{\mathbb{1}_{\{\psi_k=c\}}-\nu}{1-\nu}\sum_{k_1=1}^{p}\Theta_{kk_1}^{*,(2)}\mathbb{1}_{\{\pi(k_1)<\pi(k)\}}w_{\pi(k)}\right) \\ &= pq\mathbb{E}_{\pi}\left(\sum_{k=1}^{p}X_{ik}\mathbb{1}_{\{H_i=k\}}\sum_{k_1=1}^{p}\Theta_{kk_1}^{*,(2)}\mathbb{1}_{\{\pi(k_1)<\pi(k)\}}\sum_{\ell=2}^{p}w_\ell\mathbb{1}_{\{\pi(k)=\ell\}}\right).\end{aligned}$$

where in the final line we have appealed to (26). Now observe that for $k \in \mathbf{z}_i$,

$$\mathbb{1}_{\{H_i=k\}}\mathbb{1}_{\{\pi(k_1)<\pi(k)\}}\mathbb{1}_{\{\pi(k)=\ell\}} = \mathbb{1}_{\{X_{ik_1}=0\}}\mathbb{1}_{\{H_i=k\}}\mathbb{1}_{\{M_i=\ell,\,\pi(k_1)<\ell\}},$$

and $\mathbb{1}_{\{H_i=k\}}$ and $\mathbb{1}_{\{M_i=\ell,\,\pi(k_1)<\ell\}}$ are independent. Thus we have

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}((\mathbf{S}\mathbf{b}^{*,(2)})_i) &= \sum_{k,k_1}X_{ik}\mathbb{1}_{\{X_{ik_1}=0\}}\Theta_{kk_1}^{*,(2)}\sum_{\ell=1}^{p}p\mathbb{P}_\pi(M_i=\ell,\pi(k_1)<\ell)w_\ell \\ &= \sum_{k,k_1}X_{ik}\mathbb{1}_{\{X_{ik_1}=0\}}\Theta_{kk_1}^{*,(2)}\sum_{\ell=2}^{p}(\ell-1)\mathbb{P}_\pi(M_i=\ell|\,\pi(k_1)<\ell)w_\ell \\ &= \sum_{k,k_1}X_{ik}\mathbb{1}_{\{X_{ik_1}=0\}}\Theta_{kk_1}^{*,(2)}\sum_{\ell=2}^{p}(\ell-1)\frac{\binom{p-\ell}{q-1}}{\binom{p-1}{q}}w_\ell.\end{aligned}$$

Thus if we choose $\mathbf{w}$ such that

$$\sum_{\ell=2}^{p}(\ell-1)\frac{\binom{p-\ell}{q-1}}{\binom{p-1}{q}}w_\ell = 1, \tag{36}$$

property (35) will be satisfied.

Next we compute

$$\begin{aligned}
\mathbb{E}(\|\mathbf{b}^{*,(2)}\|_2^2) &\leq \frac{p^2q^2}{L(1-\nu)}\sum_{k=1}^{p}\mathbb{E}\Big\{\Big(\sum_{k_1=1}^{p}\Theta^{*,(2)}_{kk_1}\mathbb{1}_{\{\pi(k_1)<\pi(k)\}}\Big)^2 w^2_{\pi(k)}\Big\} \\
&= \frac{p^2q^2}{L(1-\nu)}\sum_{k=1}^{p}\sum_{\ell=1}^{p} w_\ell^2\Big(\sum_{k_1}(\Theta^{*,(2)}_{kk_1})^2\mathbb{P}(\pi(k)=\ell,\pi(k_1)<\ell) \\
&\qquad + \sum_{k_1\neq k_2}\Theta^{*,(2)}_{kk_1}\Theta^{*,(2)}_{kk_2}\mathbb{P}(\pi(k)=\ell,\pi(k_1)<\ell,\pi(k_2)<\ell)\Big) \\
&= \frac{pq^2}{L(1-\nu)}\sum_{k=1}^{p}\sum_{\ell=2}^{p} w_\ell^2\Big(\frac{\ell-1}{p-1}\sum_{k_1}(\Theta^{*,(2)}_{kk_1})^2 + \frac{(\ell-1)(\ell-2)}{(p-1)(p-2)}\sum_{k_1\neq k_2}\Theta^{*,(2)}_{kk_1}\Theta^{*,(2)}_{kk_2}\Big) \\
&\leq \frac{pq^2}{(p-1)L(1-\nu)}\sum_{k,k_1,k_2}\Big|\Theta^{*,(2)}_{kk_1}\Theta^{*,(2)}_{kk_2}\Big|\sum_{\ell=2}^{p}(\ell-1)w_\ell^2. \qquad (37)
\end{aligned}$$

Choosing

$$w_\ell = \frac{\binom{p-\ell}{q-1}\Big/\binom{p-1}{q}}{\sum_{\ell'=2}^{p}(\ell'-1)\left\{\binom{p-\ell'}{q-1}\Big/\binom{p-1}{q}\right\}^2} \qquad (38)$$

minimises (37) subject to (36) to give

$$\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\|\mathbf{b}^{*,(2)}\|_2^2) \leq \frac{pq^2}{(p-1)L(1-\nu)}\sum_{k,k_1,k_2}\Big|\Theta^{*,(2)}_{kk_1}\Theta^{*,(2)}_{kk_2}\Big|\left\{\sum_{\ell=1}^{p-1}\ell\left(\frac{\binom{p-1-\ell}{q-1}}{\binom{p-1}{q}}\right)^2\right\}^{-1}.$$

Finally, Lemma 19 bounds the right-most term from above to yield

$$\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\|\mathbf{b}^{*,(2)}\|_2^2) \leq \frac{2\{(2-\delta)q\}^2}{L(1-\nu)}\sum_{k,k_1,k_2}\Big|\Theta^{*,(2)}_{kk_1}\Theta^{*,(2)}_{kk_2}\Big|. \qquad (39)$$

Now we turn to the mean-squared error. Observe that $\mathbf{s}_i^T\mathbf{b}^*$ is a sum of L independent random variables, each having the same distribution as

$$\sum_{c=1}^{C}S_{i1c}b^*_{1c} = \sum_{c=1}^{C}S_{i1c}(b^{*,(1)}_{1c}+b^{*,(2)}_{1c}).$$

Thus

$$\begin{aligned}
\mathrm{Var}(\mathbf{s}_i^T\mathbf{b}^*) &\leq \frac{1}{L}\mathbb{E}\Big(\sum_{c=1}^{C}S_{i1c}(b^{*,(1)}_{1c}+b^{*,(2)}_{1c})\Big)^2 \\
&\leq \frac{1}{L}\left[\left\{\mathbb{E}\Big(\sum_{c=1}^{C}S_{i1c}\mathbf{b}^{*,(1)}_{1c}\Big)^2\right\}^{1/2} + \left\{\mathbb{E}\Big(\sum_{c=1}^{C}S_{i1c}\mathbf{b}^{*,(2)}_{1c}\Big)^2\right\}^{1/2}\right]^2,
\end{aligned}$$

where we have used the Cauchy–Schwarz inequality in the final line. Now using the fact that $\|\mathbf{X}\|_\infty \leq 1$, and following the argument that leads to (28), we arrive at

$$\begin{aligned}
\mathbb{E}\Big(\sum_{c=1}^{C}S_{i1c}\mathbf{b}^{*,(2)}_{1c}\Big)^2 = p^2q^2\mathbb{E}\Big\{&\frac{\nu}{1-\nu}\sum_{k=1}^{p}\Big(\sum_{k_1=1}^{p}\Theta^{*,(2)}_{kk_1}\mathbb{1}_{\{\pi(k_1)<\pi(k)\}}w_{\pi(k)}\Big)^2 \\
&+ \frac{1-2\nu}{1-\nu}X^2_{iH_i}\Big(\sum_{k_1=1}^{p}\Theta^{*,(2)}_{H_ik_1}\mathbb{1}_{\{\pi(k_1)<M_i\}}w_{M_i}\Big)^2\Big\}. \qquad (40)
\end{aligned}$$

We have

$$\begin{aligned}
&\mathbb{E}\Big\{X^2_{iH_i}\Big(\sum_{k_1=1}^{p}\Theta^{*,(2)}_{H_ik_1}\mathbb{1}_{\{\pi(k_1)<M_i\}}w_{M_i}\Big)^2\Big\} = \frac{1}{q}\sum_{k=1}^{p}\sum_{\ell=1}^{p}X^2_{ik}\mathbb{E}\Big\{\Big(\sum_{k_1}\Theta^{*,(2)}_{k,k_1}\mathbb{1}_{\{\pi(k_1)<\ell\}}w_\ell\Big)^2\mathbb{1}_{\{M_i=\ell\}}\Big\} \\
&= \sum_{k=1}^{p}X^2_{ik}\sum_{\ell=2}^{p}w_\ell^2\Big(\sum_{k_1}(\Theta^{*,(2)}_{kk_1})^2\mathbb{P}(M_i=\ell,\pi(k_1)<\ell) + \sum_{k_1\neq k_2}\Theta^{*,(2)}_{kk_1}\Theta^{*,(2)}_{kk_2}\mathbb{P}(M_i=\ell,\pi(k_1)<\ell,\pi(k_2)<\ell)\Big) \\
&= \sum_{k=1}^{p}X^2_{ik}\sum_{\ell=2}^{p}w_\ell^2\Big(\frac{\ell-1}{p-1}\frac{\binom{p-\ell}{q-1}}{\binom{p-1}{q}}\sum_{k_1}(\Theta^{*,(2)}_{kk_1})^2 + \frac{(\ell-1)(\ell-2)}{(p-1)(p-2)}\frac{\binom{p-\ell}{q-1}}{\binom{p-2}{q}}\sum_{k_1\neq k_2}\Theta^{*,(2)}_{kk_1}\Theta^{*,(2)}_{kk_2}\Big). \qquad (41)
\end{aligned}$$

Now

$$\frac{\binom{p-\ell}{q-1}}{\binom{p-1}{q}} \leq \frac{q}{p-1} \qquad \text{and} \qquad \frac{\ell-2}{p-2}\frac{\binom{p-\ell}{q-1}}{\binom{p-2}{q}} \leq \frac{q}{p-1}.$$

Thus by Lemma 19 the quantity in (41) is at most

$$\frac{2(2-\delta)^2\delta}{p^2}\sum_{k,k_1,k_2}^{p}\Big|X^2_{ik}\Theta^{*,(2)}_{kk_1}\Theta^{*,(2)}_{kk_2}\Big|.$$

Returning to (40) and using the argument leading to (37) therefore gives us

$$\mathbb{E}\Big(\sum_{c=1}^{C}S_{i1c}\mathbf{b}^{*,(2)}_{1c}\Big)^2 \leq 2(2-\delta)^2q^2\Big(\frac{\nu}{1-\nu}\sum_{k,k_1,k_2}\Big|\Theta^{*,(2)}_{kk_1}\Theta^{*,(2)}_{kk_2}\Big| + \delta\frac{1-2\nu}{1-\nu}\sum_{k,k_1,k_2}\Big|X^2_{ik}\Theta^{*,(2)}_{kk_1}\Theta^{*,(2)}_{kk_2}\Big|\Big),$$

which then gives part (iii) of the result.

Appendix B. Implications for the RKHS of the resemblance kernel

We first observe the following result that is an immediate consequence of Bouchard et al. (2013).

Proposition 13 *Consider the resemblance kernel with input space $\mathcal{X} = \{0,1\}^p$ and let $\mathcal{H}$ be the corresponding RKHS. Then $\mathcal{H}$ contains every function $f : \mathcal{X} \to \mathbb{R}$.*

Proof Let $\mathbf{X} \in \mathbb{R}^{|\mathcal{X}|\times p}$ be the matrix with each row a different element of $\mathcal{X}$ and let $\mathbf{K} \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}$ be the matrix with $K_{\mathbf{x}\mathbf{x}'} = k(\mathbf{x}, \mathbf{x}')$ where k is the resemblance kernel. Bouchard et al. (2013) shows that $\mathbf{K}$ is positive definite. Given $f : \mathcal{X} \to \mathbb{R}$, let $\mathbf{f} \in \mathbb{R}^{|\mathcal{X}|}$ be the vector of function evaluations so $f_x = f(x)$. Let $\alpha = \mathbf{K}^{-1}\mathbf{f}$. Then

$$f(\cdot) = \sum_{x\in\mathcal{X}} \alpha_x k(\cdot, x)$$

so $f \in \mathcal{H}$. ■

The following corollary of Theorem 6 derives properties of the RKHS associated with the resemblance kernel from our approximation error bounds.

Corollary 14 *Let $\mathcal{H}$ be the RKHS of the resemblance kernel k when the input space $\mathcal{X} \subset \{0,1\}^p$ is constrained such that every element has q non-zeroes. Suppose $p \geq 3$. For $\boldsymbol{\theta}^{(1)} \in \mathbb{R}^p$, $\boldsymbol{\Theta}^{(2)} \in \mathbb{R}^{p\times p}$ and $\boldsymbol{\Theta} = (\boldsymbol{\theta}^{(1)}, \boldsymbol{\Theta}^{(2)})$, define $f_{\boldsymbol{\Theta}} : \mathcal{X} \to \mathbb{R}$ by*

$$f_{\boldsymbol{\Theta}}(\mathbf{x}) = \sum_{k=1}^{p} x_k \theta_k^{(1)} + \sum_{k=1}^{p}\sum_{j=1}^{p} x_k(1-x_j)\Theta_{k,j}^{(2)}.$$

Suppose $\boldsymbol{\Theta}$ is such that $f_{\boldsymbol{\Theta}}$ is centred so $\sum_{\mathbf{x}\in\mathcal{X}} f_{\boldsymbol{\Theta}}(\mathbf{x}) = 0$ Then $f_{\boldsymbol{\Theta}} \in \mathcal{H}$ and $\|f_{\boldsymbol{\Theta}}\|_{\mathcal{H}}^2 \leq (2-\delta)q\|\boldsymbol{\Theta}\|^2$. In particular if $\boldsymbol{\Theta}^{(2)} = \mathbf{0}$ then $\|f_{\boldsymbol{\theta}^{(1)}}\|_{\mathcal{H}}^2 \leq (2-\delta)q\|\boldsymbol{\theta}^{(1)}\|_2^2$.

Proof Let $\mathbf{K} \in \mathbb{R} \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}$ be the matrix with $K_{xx'} = k(x, x')$. We will make use of the fact that $\mathbf{K}$ is positive definite (Bouchard et al., 2013). Suppose $\mathbf{X} \in \{0,1\}^{|\mathcal{X}|\times p}$ has as each row a different element of $\mathcal{X}$. For $L \in \mathbb{N}$, let $\mathbf{S}_L$ be the matrix formed from 1-bit min-wise hashing applied to $\mathbf{X}$ and let $\mathbf{K}_L = 2\mathbf{S}_L\mathbf{S}_L^T/L - \mathbf{J}$ where $\mathbf{J}$ is a $|\mathcal{X}| \times |\mathcal{X}|$ matrix of 1's. Given $\boldsymbol{\Theta}$, let $\mathbf{b}_L^*$ be as in the proof of Theorem 6 (see Section A.5) constructed using the permutations and $\boldsymbol{\Psi}$ matrix corresponding to $\mathbf{S}_L$.

Let $k_L : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be the random kernel associated with $\mathbf{K}_L$, that is $k(x, x') = K_{L,xx'}$; further let $\mathcal{H}_L$ be the associated RKHS. Let $\tilde{\mathbf{b}}_L$ be a centred version of $\mathbf{b}_L^*$ so $\tilde{\mathbf{b}}_L = \mathbf{b}_L^* - \bar{\mathbf{b}}^*{}_L$. Observe that $\|\tilde{\mathbf{b}}_L\|_2^2 \leq \|\mathbf{b}_L^*\|_2^2$. Let $f_L : \mathcal{X} \to \mathbb{R}$ be given by $f_L(x) = (S_L\tilde{\mathbf{b}}_L)_x$. Then $f_L \in \mathcal{H}_L$ and $\|f_L\|_{\mathcal{H}_L}^2 = L\|\tilde{\mathbf{b}}_L\|_2^2/2$.

Note that the construction of $\mathbf{b}_L^*$ ensures that each component block is i.i.d. Thus as $L \to \infty$, we have that almost surely

$$L\|\tilde{\mathbf{b}}_L\| \leq L\|\mathbf{b}_L^*\|_2^2 \to L^2\mathbb{E}\|(b_{L,1}^*, b_{L,2}^*)^T\|_2^2 \leq 2(2-\delta)q\|\boldsymbol{\Theta}\|^2$$

(note that the expression on the right hand side of the limit does not in fact depend on L). Also $\mathbf{K}_L \to \mathbf{K}$ almost surely by the strong law of large numbers (see Section 2.4).

Now observe that as $\mathbf{f}$ is centred, $\|\mathbf{S}_L\tilde{\mathbf{b}}_L - \mathbf{f}\|_2^2 \leq \|\mathbf{S}_L\mathbf{b}_L^* - \mathbf{f}\|_2^2$. Thus from Theorem 6 (iii) we have that $\mathbf{S}_L\tilde{\mathbf{b}}_L \to \mathbf{f}$ in probability where $\mathbf{f} \in \mathbb{R}^{|\mathcal{X}|}$ has components $f_x = f_{\boldsymbol{\Theta}}(x)$. Therefore there exists a subsequence L_j along which $\mathbf{S}_{L_j}\tilde{\mathbf{b}}_{L_j} \to \mathbf{f}$ almost surely. Thus, there exists a realisation of the random elements above such that simultaneously $\mathbf{K}_{L_j} \to \mathbf{K}$, $f_{L_j}(x) \to f(x)$ as $j \to \infty$ and $\lim_{j\to\infty} \|f_{L_j}\|_{\mathcal{H}_{L_j}}^2 \leq (2-\delta)q\|\boldsymbol{\Theta}\|^2$. In particular we have that $\|f_{L_j}\|_{\mathcal{H}_{L_j}}$ is bounded for all j. Applying Lemma 21 then gives the result. ■

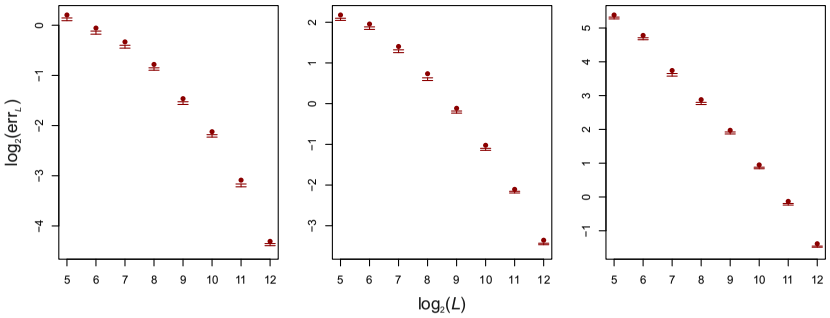


Figure 1: Plot of $\log_2(\text{err}_L)$ against $\log_2(L)$ for $q = 500, 1000, 5000$ going from left to right. The bars give the first and third quartiles of $\log_2(\text{err}_L)$ over the 100 simulations, and the circles give maximum values.

Appendix C. Empirical verification of Theorem 1

In order to assess whether the scaling in L provided by (iii) of Theorem 1 is in line with what is observed in practice, we looked at several numerical experiments. We generated design matrices $\mathbf{X} \in \{0,1\}^{n\times p}$ with different levels of sparsity $q \in \{500, 1000, 5000\}$ and $(n, p) = (10^4, 10^5)$. Different $\mathbf{S}$ matrices were constructed for each of the three $\mathbf{X}$ matrices with $\log_2 L \in \{5, 6, \ldots, 12\}$. We then generated 100 vectors of coefficients $\boldsymbol{\beta}^* \in \mathbb{R}^p$ with $\|\boldsymbol{\beta}^*\|_2 = 1$ for each setting and examined

$$\text{err}_L := \min_{\mathbf{b}\in\mathbb{R}^L} \|\mathbf{X}\boldsymbol{\beta}^* - \mathbf{S}\mathbf{b}\|_2^2/n. \tag{42}$$

Plots of $\log_2(\text{err}_L)$ against $\log_2(L)$ are given in Figure 1. Given the scaling in L suggested by Theorem 1 (iii), we would expect the points to lie on a straight line with slope -1.. We see this is indeed approximately the case for lager L and q.

Note that Theorem 1 does not make the claim that the $\mathbf{b}^*$ given is optimal in the sense of (42). Indeed it also satisfies unbiasedness and has a low ℓ_2-norm in expectation: properties not necessarily satisfied by the minimiser of (42). Moreover, the bound must encompass a worst case in terms of $\mathbf{X}$ and the direction of $\boldsymbol{\beta}^*$; tighter bounds may be used at the expense of a more complicated dependence on the precise form of $\mathbf{X}$ and $\boldsymbol{\beta}^*$. However, we see from the empirical study that the scaling in L provided by the result approximately parallels that corresponding to the minimiser of (42).

The details of the simulation study are as follows. The design $\mathbf{X}$ was generated randomly with the first $p/100$ columns containing $q/10$ 1's and the remaining columns containing $9q/10$ 1's. This mimics the setting of increasing variable sparsity described in Section 4.1. The vector of coefficients $\boldsymbol{\beta}^*$ had its first $p/100$ entries generated independently with an Exp(1) distribution and the remaining entries were set to 0; $\boldsymbol{\beta}^*$ was then scaled to have ℓ_2-norm 1.

Appendix D. Prediction error results

Here we prove results for the prediction error under linear and logistic regression models. We denote the signal to be estimated by $\mathbf{f}^*$ and assume the existence of a $\mathbf{b}^* \in \mathbb{R}^{2^b L}$ with

$$\frac{1}{n}\mathbb{E}(\|\mathbf{f}^* - \mathbf{S}\mathbf{b}^*\|_2^2) \le c_1/L$$
$$\mathbb{E}(\|\mathbf{b}^*\|_2^2) \le c_2/L.$$

Explicit constructions for such coefficient vectors are provided in the previous section. Using the results here in conjunction with the approximation error results proved in Section A yield Theorems 3–9: for example, substituting (iii) of Theorem 1 immediately gives Theorem 3.

D.1 Linear regression

We assume the model

$$\mathbf{Y} = \alpha^*\mathbf{1} + \mathbf{f}^* + \boldsymbol{\varepsilon}, \tag{43}$$

where $\mathrm{Var}(\boldsymbol{\varepsilon}) = \sigma^2\mathbf{I}$ and our goal is to estimate $\mathbf{f}^*$.

Theorem 15 *Let $(\hat{\alpha}, \hat{\mathbf{b}})$ be the least squares estimator (12). Then*

$$\mathrm{MSPE}((\hat{\alpha}, \hat{\mathbf{b}})) \;\le\; \frac{c_1}{L} + \frac{\sigma^2\{(2^b-1)L+1\}}{n}.$$

Proof Let us write

$$\mathbf{Y} = \alpha^*\mathbf{1} + \mathbf{f}^* + \boldsymbol{\varepsilon} = \alpha^*\mathbf{1} + \mathbf{S}\mathbf{b}^* + \boldsymbol{\Delta} + \boldsymbol{\varepsilon},$$

so $\boldsymbol{\Delta}$ is the approximation error of $\mathbf{S}\mathbf{b}^*$. Then we have

$$\mathrm{MSPE}((\hat{\alpha}, \hat{\mathbf{b}})) = \frac{1}{n}\mathbb{E}_{\boldsymbol{\varepsilon},\boldsymbol{\pi},\boldsymbol{\Psi}}(\|\alpha^*\mathbf{1} + \mathbf{f}^* - \hat{\alpha}\mathbf{1} - \mathbf{S}\hat{\mathbf{b}}\|_2^2).$$

Now let $\check{\mathbf{S}} = (\mathbf{1}\ \mathbf{S})$, and $\mathbf{P}_{\check{\mathbf{S}}}$ be the projection on to the column space of $\check{\mathbf{S}}$ (so $\mathbf{P}_{\check{\mathbf{S}}} = \check{\mathbf{S}}\check{\mathbf{S}}^+$, where $\check{\mathbf{S}}^+$ denotes the Moore–Penrose pseudoinverse of $\check{\mathbf{S}}$). We have the following decomposition.

$$\begin{aligned}\alpha^*\mathbf{1} + \mathbf{f}^* - \hat{\alpha}\mathbf{1} - \mathbf{S}\hat{\mathbf{b}} &= \alpha^*\mathbf{1} + \mathbf{f}^* - \mathbf{P}_{\check{\mathbf{S}}}\mathbf{Y} \\ &= \alpha^*\mathbf{1} + \mathbf{S}\mathbf{b}^* + \boldsymbol{\Delta} - \mathbf{P}_{\check{\mathbf{S}}}(\alpha^*\mathbf{1} + \mathbf{S}\mathbf{b}^* + \boldsymbol{\Delta} + \boldsymbol{\varepsilon}) \\ &= (\mathbf{I} - \mathbf{P}_{\check{\mathbf{S}}})\boldsymbol{\Delta} - \mathbf{P}_{\check{\mathbf{S}}}\boldsymbol{\varepsilon}.\end{aligned}$$

Hence

$$\begin{aligned}\mathrm{MSPE}((\hat{\alpha}, \hat{\mathbf{b}})) &= \frac{1}{n}\mathbb{E}_{\boldsymbol{\varepsilon},\boldsymbol{\pi},\boldsymbol{\Psi}}(\|(\mathbf{I} - \mathbf{P}_{\check{\mathbf{S}}})\boldsymbol{\Delta} - \mathbf{P}_{\check{\mathbf{S}}}\boldsymbol{\varepsilon}\|_2^2) \\ &= \frac{1}{n}\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\|(\mathbf{I} - \mathbf{P}_{\check{\mathbf{S}}})\boldsymbol{\Delta}\|_2^2) + \frac{1}{n}\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}\{\mathbb{E}_{\boldsymbol{\varepsilon}}(\|\mathbf{P}_{\check{\mathbf{S}}}\boldsymbol{\varepsilon}\|_2^2 \,|\, \boldsymbol{\pi}, \boldsymbol{\Psi})\} \\ &\le \frac{1}{n}\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\|\boldsymbol{\Delta}\|_2^2) + \frac{\sigma^2\{(2^b-1)L+1\}}{n} \qquad (44) \\ &\le \frac{c_1}{L} + \frac{\sigma^2\{(2^b-1)L+1\}}{n},\end{aligned}$$

where in for (44) we have used the fact that $\mathrm{rank}(\check{\mathbf{S}}) \le (2^b-1)L + 1$ as each the L blocks sums to a vector of 1's ■

Theorem 16 *There exists λ depending on $\mathbf{f}^*$ and $\mathbf{S}$ such that defining*

$$(\hat{\alpha}, \hat{\mathbf{b}}) := \operatorname*{arg\,min}_{(\alpha,\mathbf{b})\in\mathbb{R}\times\mathbb{R}^L} \|\mathbf{Y} - \hat{\alpha}\mathbf{1} - \mathbf{S}\mathbf{b}\|_2^2 \;\; \textit{such that} \;\; \|\mathbf{b}\|_2^2 \le \lambda,$$

we have

$$\mathrm{MSPE}((\hat{\alpha}, \hat{\mathbf{b}})) \;\le\; \sigma\sqrt{\frac{c_2}{n}} + \frac{c_1}{L} + \frac{\sigma^2}{n}.$$

Proof We will take $\lambda = \|\mathbf{b}^*\|_2^2$. Let a bar over any vector $\mathbf{v}$ denote the average of the components of $\mathbf{v}$, so $\bar{\mathbf{v}} = \sum_j v_j$. Note that $\hat{\alpha} = \overline{\mathbf{Y} - \mathbf{S}\hat{\mathbf{b}}}$, and define $\hat{a}^* = \overline{\mathbf{Y} - \mathbf{S}\mathbf{b}^*}$. By our choice of λ, we have that

$$\|\mathbf{Y} - \hat{\alpha}\mathbf{1} - \mathbf{S}\hat{\mathbf{b}}\|_2^2 \le \|\mathbf{Y} - \hat{a}^*\mathbf{1} - \mathbf{S}\mathbf{b}^*\|_2^2.$$

Noting that for any $\mathbf{v}, \mathbf{u} \in \mathbb{R}^n$, $\mathbf{v}^T(\mathbf{u} - \bar{\mathbf{u}}\mathbf{1}) = (\mathbf{v} - \bar{\mathbf{v}}\mathbf{1})^T\mathbf{u}$, rearranging the inequality above we get

$$\|\alpha^*\mathbf{1} + \mathbf{f}^* - \hat{\alpha}\mathbf{1} - \mathbf{S}\hat{\mathbf{b}}\|_2^2 \le 2(\boldsymbol{\varepsilon} - \bar{\boldsymbol{\varepsilon}}\mathbf{1})^T\mathbf{S}(\hat{\mathbf{b}} - \mathbf{b}^*) + \|\alpha^*\mathbf{1} + \mathbf{f}^* - \hat{a}^*\mathbf{1} - \mathbf{S}\mathbf{b}^*\|_2^2. \tag{45}$$

Now observe that

$$\begin{aligned}\|\alpha^*\mathbf{1} + \mathbf{f}^* - \hat{a}^*\mathbf{1} - \mathbf{S}\mathbf{b}^*\|_2^2 &= \|\mathbf{f}^* - \overline{\mathbf{f}^*}\mathbf{1} - (\mathbf{S}\mathbf{b}^* - \overline{\mathbf{S}\mathbf{b}^*}\mathbf{1})\|_2^2 + n\bar{\boldsymbol{\varepsilon}}^2 \\ &\le \|\mathbf{f}^* - \mathbf{S}\mathbf{b}^*\|_2^2 + n\bar{\boldsymbol{\varepsilon}}^2. \qquad (46)\end{aligned}$$

As $\mathbf{b}^*$ is independent of $\boldsymbol{\varepsilon}$, taking expectations of (45) yields

$$\mathrm{MSPE}(\hat{\mathbf{b}}) = \frac{2}{n}\mathbb{E}\{(\boldsymbol{\varepsilon} - \bar{\boldsymbol{\varepsilon}}\mathbf{1})^T\mathbf{S}\hat{\mathbf{b}}\} + \frac{1}{n}\mathbb{E}(\|\mathbf{f}^* - \mathbf{S}\mathbf{b}^*\|_2^2) + \frac{\sigma^2}{n}. \tag{47}$$

Now using the fact that $\|\hat{\mathbf{b}}\|_2 \le \|\mathbf{b}^*\|_2$ and applying the Cauchy–Schwarz inequality we have

$$\mathbb{E}_{\boldsymbol{\varepsilon},\boldsymbol{\pi},\boldsymbol{\Psi}}\{(\boldsymbol{\varepsilon} - \bar{\boldsymbol{\varepsilon}}\mathbf{1})^T\mathbf{S}\hat{\mathbf{b}}\} \le \sqrt{\mathbb{E}_{\boldsymbol{\varepsilon},\boldsymbol{\pi},\boldsymbol{\Psi}}\{\|\mathbf{S}^T(\boldsymbol{\varepsilon} - \bar{\boldsymbol{\varepsilon}}\mathbf{1})\|_2^2\}}\sqrt{\mathbb{E}(\|\mathbf{b}^*\|_2^2)}.$$

But

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\varepsilon}}(\|\mathbf{S}^T(\boldsymbol{\varepsilon} - \bar{\boldsymbol{\varepsilon}}\mathbf{1})\|_2^2|\boldsymbol{\pi}, \boldsymbol{\Psi}) &= \mathbb{E}_{\boldsymbol{\varepsilon}}[\mathrm{Tr}\{(\boldsymbol{\varepsilon} - \bar{\boldsymbol{\varepsilon}}\mathbf{1})^T\mathbf{S}\mathbf{S}^T(\boldsymbol{\varepsilon} - \bar{\boldsymbol{\varepsilon}}\mathbf{1})\}|\boldsymbol{\pi}, \boldsymbol{\Psi}] \\ &= \mathbb{E}_{\boldsymbol{\varepsilon}}[\mathrm{Tr}\{(\boldsymbol{\varepsilon} - \bar{\boldsymbol{\varepsilon}}\mathbf{1})(\boldsymbol{\varepsilon} - \bar{\boldsymbol{\varepsilon}}\mathbf{1})^T\mathbf{S}\mathbf{S}^T\}|\boldsymbol{\pi}, \boldsymbol{\Psi}] \\ &= \mathrm{Tr}[\mathbb{E}_{\boldsymbol{\varepsilon}}\{(\boldsymbol{\varepsilon} - \bar{\boldsymbol{\varepsilon}}\mathbf{1})(\boldsymbol{\varepsilon} - \bar{\boldsymbol{\varepsilon}}\mathbf{1})^T\}\mathbf{S}\mathbf{S}^T] \\ &= \sigma^2\|(\mathbf{I} - n^{-1}\mathbf{1}\mathbf{1}^T)\mathbf{S}\|_F^2 \le \sigma^2\|\mathbf{S}\|_F^2 \le \sigma^2 nL,\end{aligned}$$

whence

$$\mathbb{E}_{\boldsymbol{\varepsilon},\boldsymbol{\pi},\boldsymbol{\Psi}}\{(\boldsymbol{\varepsilon} - \bar{\boldsymbol{\varepsilon}}\mathbf{1})^T\mathbf{S}\hat{\mathbf{b}}\} \le \sigma\sqrt{c_2 n}. \tag{48}$$

Substituting in to (47) then gives the result. ■

D.2 Logistic regression

We give an analogous result to Theorem 4 for classification problems under logistic loss. Let $\mathbf{X} \in [-1,1]^{n\times p}$ be the design matrix of predictor variables and let $\mathbf{Y} \in \{0,1\}^n$ be an associated vector of class labels. We assume the model

$$Y_i \sim \text{Bernoulli}(p_i); \qquad \log\left(\frac{p_i}{1-p_i}\right) = f_i,$$

with the Y_i independent for $1 \le i \le n$. Define

$$\hat{\mathbf{b}}_\lambda = \arg\min_{\mathbf{b}} \frac{1}{n}\sum_{i=1}^n \left[-Y_i \mathbf{s}_i^T\mathbf{b} + \log\{1+\exp(\mathbf{s}_i^T\mathbf{b})\}\right] \text{ such that } \|\mathbf{b}\|_2^2 \le \lambda.$$

Let $\mathcal{E}(\hat{\mathbf{b}}_\lambda)$ denote the excess risk of $\hat{\mathbf{b}}_\lambda$ under logistic loss, so

$$\mathcal{E}(\hat{\mathbf{b}}_\lambda) = \frac{1}{n}\sum_{i=1}^n \left[-p_i \mathbf{s}_i^T\hat{\mathbf{b}}_\lambda + \log\{1+\exp(\mathbf{s}_i^T\hat{\mathbf{b}}_\lambda)\}\right] - \frac{1}{n}\sum_{i=1}^n \left[-p_i f_i + \log\{1+\exp(f_i)\}\right].$$

Theorem 17 *Let $\tilde{p} \in \mathbb{R}$ be given by (18). Then we have that there exists λ such that*

$$\mathbb{E}_{\mathbf{Y},\boldsymbol{\pi},\boldsymbol{\Psi}}\{\mathcal{E}(\hat{\mathbf{b}}_\lambda)\} \le \frac{c_1}{4L} + \sqrt{\tilde{p}c_2/n}.$$

Proof We take $\lambda = \|\mathbf{b}^*\|_2^2$. By the definition of $\hat{\mathbf{b}}$ (dropping the subscript λ), we have

$$\frac{1}{n}\sum_{i=1}^n \left[-Y_i \mathbf{s}_i^T\hat{\mathbf{b}} + \log\{1+\exp(\mathbf{s}_i^T\hat{\mathbf{b}})\}\right] \le \frac{1}{n}\sum_{i=1}^n \left[-Y_i \mathbf{s}_i^T\mathbf{b}^* + \log\{1+\exp(\mathbf{s}_i^T\mathbf{b}^*)\}\right].$$

Using this, analogously to (45) we get,

$$\mathcal{E}(\hat{\mathbf{b}}) \le \frac{1}{n}\sum_{i=1}^n (Y_i - p_i)\{\mathbf{S}(\hat{\mathbf{b}} - \mathbf{b}^*)\}_i + \mathcal{E}(\mathbf{b}^*).$$

Let $\boldsymbol{\varepsilon} := \mathbf{Y} - \mathbf{p}$ be the residual vector. Since $\mathbf{b}^*$ is independent of $\boldsymbol{\varepsilon}$, after taking expectations we arrive at

$$\mathbb{E}_{\boldsymbol{\varepsilon},\boldsymbol{\pi},\boldsymbol{\Psi}}\{\mathcal{E}(\hat{\mathbf{b}})\} \le \frac{1}{n}\mathbb{E}_{\boldsymbol{\varepsilon},\boldsymbol{\pi},\boldsymbol{\Psi}}(\boldsymbol{\varepsilon}^T\mathbf{S}\hat{\mathbf{b}}) + \mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}\{\mathcal{E}(\mathbf{b}^*)\}.$$

Write $h(a) = \log(1+e^a)$. By the mean value theorem, we have

$$\begin{aligned}|\mathcal{E}(\mathbf{b}^*)| &= \frac{1}{n}\sum_{i=1}^n |h(\mathbf{s}_i^T\mathbf{b}^*) - h(f_i) - (\mathbf{s}_i^T\mathbf{b}^* - f_i)h'(f_i)| \\ &\le \frac{1}{n}\sup_{a\in\mathbb{R}} h''(a)\|\mathbf{f}^* - \mathbf{S}\mathbf{b}^*\|_2^2 \le \frac{c_1}{4L}.\end{aligned}$$

The same argument that leads to (48) gives

$$\frac{1}{n}\mathbb{E}_{\boldsymbol{\varepsilon},\boldsymbol{\pi},\boldsymbol{\Psi}}(\boldsymbol{\varepsilon}^T\mathbf{S}\hat{\mathbf{b}}) \le \frac{1}{n}\sqrt{\mathbb{E}_{\boldsymbol{\varepsilon},\boldsymbol{\pi},\boldsymbol{\Psi}}(\|\mathbf{S}^T\boldsymbol{\varepsilon}\|_2^2)}\sqrt{c_2/L} \le \sqrt{\tilde{p}c_2/n}.$$

Collecting together the various inequalities, we get the required result. ■

Appendix E. Technical lemmas

In this section we collect all technical lemmas used by the results presented earlier.

Lemma 18 *Let $(a_i)_{i=1}^\infty$ and $(b_i)_{i=1}^\infty$ be two sequences of non-negative, non-increasing, real numbers such that that there is some $i^* \in \mathbb{N}$ for which*

$$\begin{aligned}a_i &\le b_i \quad \text{for all } i \le i^*, \\ a_i &\ge b_i \quad \text{for all } i > i^*.\end{aligned}$$

(i) If

$$\sum_{i=1}^\infty a_i = \sum_{i=1}^\infty b_i < \infty,$$

and $m \ge 1$, then

$$\sum_{i=1}^\infty a_i^m \le \sum_{i=1}^\infty b_i^m.$$

(ii) If $(c_i)_{i=1}^\infty$ is a sequence of non-negative, non-decreasing real numbers and

$$\sum_{i=1}^\infty b_i \le \sum_{i=1}^\infty a_i < \infty, \ \sum_{i=1}^\infty c_i a_i, \ \sum_{i=1}^\infty c_i b_i < \infty,$$

then

$$\sum_{i=1}^\infty c_i a_i \ge \sum_{i=1}^\infty c_i b_i.$$

Proof Note that the sequence $(b_i)_{i=1}^\infty$ majorises $(a_i)_{i=1}^\infty$ (see page 191 of Steele (2004)). Result (i) follows from applying Schur's majorisation inequality (Steele (2004); page 201) with the convex function $x \mapsto x^m$ on $[0,\infty)$.

For (ii) we argue,

$$\sum_{i=1}^{i^*} c_i(b_i - a_i) \le c_{i^*}\sum_{i=1}^{i^*}(b_i - a_i) \le c_{i^*}\sum_{i>i^*}(a_i - b_i) \le \sum_{i>i^*} c_i(a_i - b_i).$$

■

Lemma 19 *Let $q, p \in \mathbb{N}$ with $q \ge 1$, $p \ge \max\{q, 3\}$. We have*

$$\sum_{\ell=1}^{p-1} \ell\left(\frac{\binom{p-1-\ell}{q-1}}{\binom{p-1}{q}}\right)^2 \ge \frac{1}{2(2-q/p)^2}\frac{p^2}{(p-1)^2}.$$

Proof Let the sequences $(a_\ell)_{\ell=1}^\infty$ and $(b_\ell)_{\ell=1}^\infty$ be defined by

$$a_\ell = \begin{cases} \left(\frac{\binom{p-1-\ell}{q-1}}{\binom{p-1}{q}}\right)^2 & \text{if } 1 \le \ell \le p-1 \\ 0 & \text{otherwise,} \end{cases}$$

$$b_\ell = \begin{cases} \left(\frac{q}{p-1}\right)^2 & \text{if } \ell \le \left\lfloor \frac{(p-1)^2}{\{2(p-1)-q\}q} \right\rfloor \\ \frac{q}{2(p-1)-q} - \left(\frac{q}{p-1}\right)^2 \left\lfloor \frac{(p-1)^2}{\{2(p-1)-q\}q} \right\rfloor & \text{if } \ell = \left\lfloor \frac{(p-1)^2}{\{2(p-1)-q\}q} \right\rfloor + 1 \\ 0 & \text{otherwise.} \end{cases}$$

Let the sequence $(c_\ell)_{\ell=1}^\infty$ be defined by $c_\ell = \ell$. Note the sequences $(a_\ell)_{\ell=1}^\infty$, $(b_\ell)_{\ell=1}^\infty$ and $(c_\ell)_{\ell=1}^\infty$ satisfy the hypotheses of Lemma 18. Thus

$$\sum_{\ell=1}^{p-1} \ell a_\ell \ge \sum_{\ell=1}^{p-1} \ell b_\ell,$$

and

$$\begin{aligned} \sum_{\ell=1}^{p-1} \ell b_\ell = \frac{1}{2}\left(\frac{q}{p-1}\right)^2 \left(\left\lfloor \frac{(p-1)^2}{\{2(p-1)-q\}q} \right\rfloor + 1\right) \left\lfloor \frac{(p-1)^2}{\{2(p-1)-q\}q} \right\rfloor \\ + \left(\frac{q}{p-1}\right)^2 \left(\frac{(p-1)^2}{\{2(p-1)-q\}q} - \left\lfloor \frac{(p-1)^2}{\{2(p-1)-q\}q} \right\rfloor\right) \left(\left\lfloor \frac{(p-1)^2}{\{2(p-1)-q\}q} \right\rfloor + 1\right). \end{aligned}$$

Letting $x = (p-1)^2/[\{2(p-1)-q\}q]$, we have

$$\begin{aligned} \sum_{\ell=1}^{p-1} \ell b_\ell &= \frac{1}{2}(\lfloor x \rfloor + 1)\lfloor x \rfloor + (x - \lfloor x \rfloor)(\lfloor x \rfloor + 1) \\ &= \frac{1}{2}x(x+1) - \frac{1}{2}\{(x - \lfloor x \rfloor)\lfloor x \rfloor + (x - \lfloor x \rfloor)(x+1)\} + (x - \lfloor x \rfloor)(\lfloor x \rfloor + 1). \end{aligned}$$

Since $1 \ge 1/2 + (x - \lfloor x \rfloor)/2$, we see that

$$(x - \lfloor x \rfloor)(\lfloor x \rfloor + 1) \ge \frac{1}{2}(x - \lfloor x \rfloor)(x + 1 + \lfloor x \rfloor),$$

so

$$\begin{aligned} \sum_{\ell=1}^{p-1} \ell b_\ell &\ge \frac{1}{2}x(x+1) \\ &= \frac{1}{2}\left(\frac{(p-1)^2}{\{2(p-1)-q\}q} + 1\right)\frac{q}{2(p-1)-q} \\ &= \frac{1}{2(p-1)}\frac{p + \{2 - q/(p-1)\}q - 1}{\{2 - q/(p-1)\}^2} \\ &\ge \frac{1}{2(p-1)}\frac{p+q}{\{2 - q/(p-1)\}^2} \\ &\ge \frac{1}{2(2-q/p)^2}\frac{p^2}{(p-1)^2}. \end{aligned}$$

■

Lemma 20 *Let $\kappa(\delta) = \delta^{-a}$ where $a \in [0,1]$. For $\ell \ge 2$,*

$$\left|\frac{\kappa^{(\ell)}(1)}{\ell!}\right| \le a e^a \frac{1}{\ell^{1-a}}.$$

Proof

$$\begin{aligned} \left|\frac{\kappa^{(\ell)}(1)}{\ell!}\right| &= \frac{a(a+1)\cdots(a+\ell-1)}{1\cdot 2\cdots \ell} \\ &= \frac{a}{\ell}\frac{a+1}{1}\frac{a+2}{2}\cdots\frac{a+\ell-1}{\ell-1}. \end{aligned}$$

By Jensen's inequality

$$\begin{aligned} \frac{1}{\ell-1}\left\{\log\left(\frac{a+1}{1}\right) + \log\left(\frac{a+2}{2}\right) + \cdots + \log\left(\frac{a+\ell-1}{\ell-1}\right)\right\} \\ \le \log\left(1 + \frac{a\{1+\log(\ell-1)\}}{\ell-1}\right), \end{aligned}$$

and

$$\left(1 + \frac{a\{1+\log(\ell-1)\}}{\ell-1}\right)^{\ell-1} \le \exp[a\{1+\log(\ell-1)\}].$$

Thus

$$\left|\frac{\kappa^{(\ell)}(1)}{\ell!}\right| \le a e^a \frac{(\ell-1)^a}{\ell} \le a e^a \frac{1}{\ell^{1-a}}.$$

■

Lemma 21 *Suppose we have a sequence of positive definite kernels $\{k_L\}_{L=1}^\infty$ on a finite input space $\mathcal{X}$. For $L \in \mathbb{N}$, let $\mathbf{K}_L \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}$ be the matrix with $K_{L,xx'} = k_L(x,x')$. Suppose that $\mathbf{K}_L \to \mathbf{K}$ where $\mathbf{K}$ is positive definite and corresponds to kernel k. Let the RKHS's associated with k_L and k be $\mathcal{H}_L$ and $\mathcal{H}$ respectively. Suppose $f_L \in \mathcal{H}_L$ satisfies $|f_L(x) - f(x)| \to 0$ for some $f : \mathcal{X} \to \mathbb{R}$ and all $x \in \mathcal{X}$, and $\|f_L\|_{\mathcal{H}_L} < C$ for some $C > 0$. Then $f \in \mathcal{H}$ and $\|f_L\|_{\mathcal{H}_L} \to \|f\|_{\mathcal{H}}$ as $L \to \infty$.*

Proof Since $\mathcal{X}$ is finite, for each L there exists $\boldsymbol{\alpha}_L \in \mathbb{R}^{|\mathcal{X}|}$ with $(f_L(x))_{x\in\mathcal{X}} = \mathbf{K}_L\boldsymbol{\alpha}_L$. Writing $\mathbf{f} = (f(x))_{x\in\mathcal{X}}$, we have $\mathbf{K}_L\boldsymbol{\alpha}_L \to \mathbf{f}$. Now $\mathbf{f} = \mathbf{K}\boldsymbol{\alpha}$ where $\boldsymbol{\alpha} = \mathbf{K}^{-1}\mathbf{f}$ showing that $f \in \mathcal{H}$.

It remains to show that $\boldsymbol{\alpha}_L^T\mathbf{K}_L\boldsymbol{\alpha}_L \to \boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha}$. Note that $\mathbf{K}_L$ is positive definite for L sufficiently large, so the fact that $\boldsymbol{\alpha}_L^T\mathbf{K}_L\boldsymbol{\alpha}_L < C$ ensures the $\boldsymbol{\alpha}_L$ are bounded. Now suppose, for a contradiction, that there exists $\epsilon > 0$ and a subsequence L_j with

$$|\boldsymbol{\alpha}_{L_j}^T\mathbf{K}_{L_j}\boldsymbol{\alpha}_{L_j} - \boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha}| > \epsilon. \tag{49}$$

Then as the $\boldsymbol{\alpha}_{L_j}$ are bounded, there exists a further subsequence $L_{j_m} = l_m$ such that $\boldsymbol{\alpha}_{l_m} \to \boldsymbol{\alpha}_*$ as $m \to \infty$. But then since the fact that $\mathbf{K}_L \to \mathbf{K}$ implies the maximal eigenvalues of the $\mathbf{K}_L$ are bounded, $\boldsymbol{\alpha}_{l_m}^T \mathbf{K}_{l_m} \boldsymbol{\alpha}_{l_m} \to \boldsymbol{\alpha}_*^T K_{l_m} \boldsymbol{\alpha}_*$ as $m \to \infty$. But then $\boldsymbol{\alpha}_{l_m}^T \mathbf{K}_{l_m} \boldsymbol{\alpha}_{l_m} \to \boldsymbol{\alpha}_*^T \mathbf{K} \boldsymbol{\alpha}_*$, contradicting (49). ■

References

D. Achlioptas. Database-friendly random projections. In *Proceedings of the twentieth ACM symposium on principles of database systems*, pages 274–281. ACM, 2001.

F. Bach. Sharp analysis of low-rank kernel matrix approximations. In *Conference on Learning Theory*, pages 185–209, 2013.

F. Bach. On the equivalence between kernel quadrature rules and random feature expansions. *Journal of Machine Learning Research*, 18:1–38, 2017.

A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6:1345–1382, sep 2005.

M. Bouchard, A.-L. Jousselme, and P.-E. Dor. A proof for the positive definiteness of the jaccard index matrix. *International Journal of Approximate Reasoning*, 54(5):615 – 626, 2013.

C. Boutsidis and P. Drineas. Random projections for the nonnegative least-squares problem. *Linear Algebra and its Applications*, 431:760–771, 2009.

L. Breiman. Random Forests. *Machine Learning*, 45:5–32, 2001.

A. Broder, M. Charikar, A. Frieze, and M. Mitzenmacher. Min-wise independent permutations. In *Proceedings of the thirtieth annual ACM symposium on theory of computing*, pages 327–336. ACM, 1998.

P. Bühlmann. Boosting for high-dimensional linear models. *Annals of Statistics*, 34:559–583, 2006.

P. Bühlmann and S. van de Geer. *Statistics for high-dimensional data*. Springer, 2011.

J.L. Carter and M.N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18:143–154, 1979.

E. Cohen, M. Datar, S. Fujiwara, A. Gionis, P. Indyk, R. Motwani, J. Ullman, and C. Yang. Finding interesting associations without support pruning. *IEEE Transactions on Knowledge and Data Engineering*, 13:64–78, 2001.

M. Datar and S. Muthukrishnan. Estimating rarity and similarity over data stream windows. *Lecture Notes in Computer Science*, 2461:323, 2002.

P. Drineas, M.W. Michael W Mahoney, S. Muthukrishnan, and T. Sarlós. Faster least squares approximation. *Numerische Mathematik*, 117:219–249, 2011.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–451, 2004.

D. Fradkin and D. Madigan. Experiments with random projections for machine learning. In *Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining*, pages 517–522. ACM, 2003.

J. Friedman and B. Popescu. Predictive learning via rule ensembles. *Annals of Applied Statistics*, 2:916–954, 2008.

A.E. Hoerl and R.W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, pages 55–67, 1970.

T. Hofmann, B. Schölkopf, and A. Smola. Kernel methods in machine learning. *Annals of Statistics*, pages 1171–1220, 2008.

W.B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26(189-206):1, 1984.

I.T. Jolliffe. Principal components in regression analysis. In *Principal component analysis*, pages 129–155. Springer, 1986.

A. Kaban. New bounds on compressive linear least squares regression. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pages 448–456, 2014.

P. Kar and H. Karnick. Random feature maps for dot product kernels. In Neil D. Lawrence and Mark A. Girolami, editors, *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22, pages 583–591, 2012.

J. Langford, L. Li, and A. Strehl. Vowpal wabbit online learning project, 2007.

Q. Le, T. Sarlós, and A. Smola. Fastfood-computing hilbert space expansions in loglinear time. In *Proceedings of the 30th International Conference on Machine Learning*, pages 244–252, 2013.

P. Li. Core kernels. *arXiv preprint arXiv:1404.6216*, 2014.

P. Li and A.C. König. Theory and applications of b-bit minwise hashing. *Communications of the ACM*, 54:101–109, 2011.

P. Li, T. Hastie, and K. Church. Very sparse random projections. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 287–296. ACM, 2006.

P. Li, A. Shrivastava, J. Moore, and A. König. Hashing algorithms for large-scale learning. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2011.

P. Li, A. Owen, and C.-H. Zhang. One permutation hashing. In *Advances in Neural Information Processing Systems*, pages 3122–3130, 2012.

P. Li, A. Shrivastava, and A. König. b-bit minwise hashing in practice. In *Proceedings of the 5th Asia-Pacific Symposium on Internetware*, page 13. ACM, 2013.

Michael W Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends® in Machine Learning*, 3(2):123–224, 2011.

O. Maillard and R. Munos. Linear regression with random projections. *Journal of Machine Learning Research*, 13:2735–2772, 2012.

J. Pennington, F. Yu, and S. Kumar. Spherical random features for polynomial kernels. In *Advances in Neural Information Processing Systems*, pages 1846–1854, 2015.

M. Pilanci and M.J. Wainwright. Iterative hessian sketch: Fast and accurate solution approximation for constrained least-squares. *arXiv preprint arXiv:1411.0347*, 2014.

M. Pilanci and M.J. Wainwright. Randomized sketches of convex programs with sharp guarantees. *Information Theory, IEEE Transactions on*, 61(9):5096–5115, 2015.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, pages 1177–1184, 2007.

A. Rahimi and B. Recht. Uniform approximation of functions with random bases. In *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, pages 555–561. IEEE, 2008.

A. Rahimi and B. Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in neural information processing systems*, pages 1313–1320, 2009.

A. Rudi, R. Camoriano, and L. Rosasco. Less is more: Nyström computational regularization. In *Advances in Neural Information Processing Systems*, pages 1657–1665, 2015.

Q. Shi, J. Petterson, G. Dror, J. Langford, A. Smola, and S. Vishwanathan. Hash kernels for structured data. *Journal of Machine Learning Research*, 10:2615–2637, 2009.

M. Steele. *The Cauchy–Schwarz Master Class*. Cambridge University Press, 2004.

D. J. Sutherland and J. Schneider. On the error of random fourier features. In *UAI*, 2015.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

J.A. Tropp. Greed is good: algorithmic results for sparse approximation. *Information Theory, IEEE Transactions on*, 50:2231–2242, 2004.

S.A. Van De Geer. High-dimensional generalized linear models and the lasso. *Annals of Statistics*, 36:614–645, 2008.

A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(3):480–492, 2012.

S. Vempala. *The random projection method*, volume 65. American Mathematical Society, 2005.

K. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg. Feature hashing for large scale multitask learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1113–1120. ACM, 2009.

C.K.I. Williams and M. Seeger. Using the nyström method to speed up kernel machines. In *Advances in neural information processing systems*, pages 682–688, 2001.

Y. Yang, M. Pilanci, and M.J. Wainwright. Randomized sketches for kernels: Fast and optimal non-parametric regression. *Annals of Statistics*, 25:991–1023, 2017.

Journal of Machine Learning Research 18 (2018) 1-52 Submitted 2/17; Revised 11/17; Published 4/18

Efficient Learning with a Family of Nonconvex Regularizers by Redistributing Nonconvexity

Quanming Yao QYAOAA@CSE.UST.HK
James T. Kwok JAMESK@CSE.UST.HK
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Hong Kong

Editor: Mark Schmidt

Abstract

The use of convex regularizers allows for easy optimization, though they often produce biased estimation and inferior prediction performance. Recently, nonconvex regularizers have attracted a lot of attention and outperformed convex ones. However, the resultant optimization problem is much harder. In this paper, a popular subclass of ℓ_1-based nonconvex sparsity-inducing and low-rank regularizers is considered. This includes nonconvex variants of lasso, sparse group lasso, tree-structured lasso, nuclear norm and total variation regularizers. We propose to move the nonconvexity from the regularizer to the loss. The nonconvex regularizer is then transformed to a familiar convex one, while the resultant loss function can still be guaranteed to be smooth. Learning with the convexified regularizer can be performed by existing efficient algorithms originally designed for convex regularizers (such as the proximal algorithm, Frank-Wolfe algorithm, alternating direction method of multipliers and stochastic gradient descent). This is further extended to consider cases where the convexified regularizer does not have a closed-form proximal step, and when the loss function is nonconvex nonsmooth. Extensive experiments on a variety of machine learning application scenarios show that optimizing the transformed problem is much faster than running the state-of-the-art on the original problem.

Keywords: Nonconvex optimization, Nonconvex regularization, Proximal algorithm, Frank-Wolfe algorithm, Matrix completion

1. Introduction

Regularized risk minimization is fundamental to machine learning. It admits a tradeoff between the empirical loss and regularization as:

$$\min_x F(x) \equiv f(x) + g(x), \tag{1}$$

where x is the model parameter, f is the loss and g is the regularizer. The choice of regularizers is important and application-specific, and is often the crux to obtain good prediction performance. Popular examples include the sparsity-inducing regularizers, which have been commonly used in image processing (Beck and Teboulle, 2009; Mairal et al., 2009; Jenatton et al., 2011) and high-dimensional feature selection (Tibshirani et al., 2005; Jacob et al., 2009; Liu and Ye, 2010); and the low-rank regularizers, which have obtained good empirical performance on various matrix and tensor learning tasks such as collaborative filtering (Candès and Recht, 2009; Mazumder et al., 2010) and visual data analysis (Liu et al., 2013; Lu et al., 2014).

Most of these regularizers are convex. Well-known examples include the ℓ_1-regularizer for sparse coding (Donoho, 2006), and the nuclear norm regularizer in low-rank matrix learning (Candès and Recht, 2009). Besides having nice theoretical guarantees, convex regularizers also allow easy optimization. Popular optimization algorithms in machine learning include the proximal algorithm (Parikh and Boyd, 2013), Frank-Wolfe (FW) algorithm (Jaggi, 2013), alternating direction method of multipliers (ADMM) (Boyd et al., 2011), stochastic gradient descent (SGD) and its variants (Bottou, 1998; Xiao and Zhang, 2014). Many of these are efficient, scalable, and have sound convergence properties.

However, convex regularizers often lead to biased estimation. For example, in sparse coding, the solution obtained by the ℓ_1-regularizer is often not as sparse and accurate (Zhang, 2010b). In low-rank matrix learning, the estimated rank obtained with the nuclear norm regularizer is often much higher (Mazumder et al., 2010). To alleviate this problem, a number of nonconvex regularizers, which are variants of the convex ℓ_1-norm, have been recently proposed. Examples include the Geman penalty (GP) (Geman and Yang, 1995), log-sum penalty (LSP) (Candès et al., 2008), minimax concave plus (MCP) penalty (Zhang, 2010a), Laplace penalty (Trzasko and Manduca, 2009), and smoothly clipped absolute deviation (SCAD) penalty (Fan and Li, 2001). As can be seen from Figure 1, they are all (i) nonsmooth at zero, which encourage a sparse solution; and (ii) concave, which place a smaller penalty than the ℓ_1-regularizer on features with large magnitudes. Empirically, these nonconvex regularizers usually outperform convex regularizers.

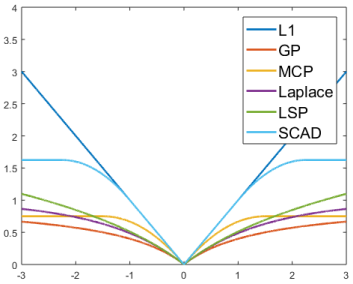


Figure 1: Plot of various nonconvex variants of the convex ℓ_1-regularizer.

In this paper, we consider a popular subclass of ℓ_1-based nonconvex sparsity-inducing and low-rank regularizers such as nonconvex variants of the lasso, sparse group lasso, tree-structured lasso, nuclear norm and total variation regularizers. Even with a convex loss, the resulting nonconvex problem (1) is much harder to optimize. One can use general-purpose nonconvex optimization solvers such as the concave-convex procedure (Yuille and Rangarajan, 2002). Unfortunately, the subproblem in each iteration can be as expensive as the original problem, and the concave-convex procedure is thus often slow in practice (Gong et al., 2013; Zhong and Kwok, 2014).

Recently, the proximal algorithm has been extended for nonconvex problems. Examples include nonconvex inexact proximal splitting (NIPS) (Sra, 2012), inertial proximal algorithm for non-convex optimization (IPiano) (Ochs et al., 2014), unified treatment of accelerated gradient (UAG) (Ghadimi and Lan, 2016), general iterative shrinkage and thresholding (GIST) (Gong et al.,

2013), inertial forward-backward (IFB) algorithm (Bot et al., 2016), and nonmonotone accelerated proximal gradient (nmAPG) (Li and Lin, 2015) algorithm. Specifically, NIPS, IPiano and UAG allow f in (1) to be Lipschitz smooth (and possibly nonconvex) but g has to be convex; while GIST, IFB and nmAPG further allow g to be nonconvex. The current state-of-the-art is nmAPG. Nevertheless, efficient computation of the underlying proximal operator is only possible for simple nonconvex regularizers. When the regularizer is complicated, such as the nonconvex versions of the fused lasso and overlapping group lasso regularizers (Zhong and Kwok, 2014), the corresponding proximal step has to be solved numerically and is again expensive. Another approach is by using the proximal average (Zhong and Kwok, 2014), which computes and averages the proximal step of each underlying regularizer. As the proximal step is only approximate, its convergence is usually slower than typical applications of the proximal algorithm (Yu, 2013; Li and Lin, 2015).

When f is smooth, there are endeavors to extend other algorithms from convex to nonconvex optimization. For the global consensus problem, standard ADMM converges only when g is convex (Hong et al., 2016). When g is nonconvex, convergence of ADMM is only established for problems of the form $\min_{x,y} f(x) + g(y) : y = Ax$, where matrix A has full row rank (Li and Pong, 2015). The convergence of ADMM in more general cases is an open issue. More recently, the stochastic variance reduced gradient (SVRG) algorithm (Xiao and Zhang, 2014), which is a SGD variant with reduced variance in the gradient estimates, has also been extended for problems with nonconvex f, but the regularizer g is still required to be convex (Reddi et al., 2016a; Zhu and Hazan, 2016).

Sometimes, it is desirable to have a nonsmooth loss f. For example, the absolute loss is more robust to outliers than the square loss, and has been popularly used in applications such as image denoising (Yan, 2013), robust dictionary learning (Zhao et al., 2011) and robust PCA (Candès et al., 2011). The resulting optimization problem becomes more challenging. When both f and g are convex, ADMM is often the main optimization tool for problem (1) (He and Yuan, 2012). However, when either f or g is nonconvex, ADMM no longer guarantees convergence. Besides, the loss f may also be nonconvex as this is more robust to outliers and can obtain better performance (e,g., ℓ_0-norm (Yan, 2013) and capped-ℓ_1 norm (Sun et al., 2013)). However, when f is nonsmooth and nonconvex, none of the above-mentioned algorithms (i.e., proximal algorithms, FW algorithms, ADMM, and SVRG) can be used. As a last resort, one can use more general nonconvex optimization approaches such as convex concave programming (CCCP) (Yuille and Rangarajan, 2002), which is slow in general.

In this paper, we first consider the case where the loss function f is smooth (possibly nonconvex) and the regularizer g is nonconvex. We propose to handle nonconvex regularizers by reusing the abundant repository of efficient convex algorithms originally designed for convex regularizers. Motivated by the fact that recent proximal algorithms (Gong et al., 2013; Li and Lin, 2015; Zhong and Kwok, 2014) all rely on the smoothness of f and a simple closed-form proximal step for g, the key is to shift nonconvexity associated with the nonconvex regularizer to the loss function. The nonconvex regularizer is then transformed to a familiar convex regularizer, while the transformed loss function is still smooth. To illustrate the practical usefulness of this convexification scheme, we show how it can be used with popular optimization algorithms in machine learning. For example, for the proximal algorithm, the resultant proximal step can be much easier after transformation. Specifically, for the nonconvex tree-structured lasso and nonconvex sparse group lasso, we show that the corresponding proximal steps have closed-form solutions on the transformed problems, but not on the original ones. For the nonconvex total variation problem, though there is no closed-form solution for the proximal step before and after the transformation, we show that the proximal step

is still cheaper and easier for optimization after the transformation. To allow further speedup, we propose a proximal algorithm variant that allows the use of inexact proximal steps with convex g when it has no closed-form proximal step solution. For the FW algorithm, we consider its application to nonconvex low-rank matrix learning problems, and propose a variant with guaranteed convergence to a critical point of the nonconvex problem. For SVRG in stochastic optimization and ADMM in consensus optimization, we show that these algorithms have convergence guarantees on the transformed problems but not on the original ones.

We further consider the case where f is both nonconvex and nonsmooth (and g is nonconvex). We demonstrate that problem (1) can be similarly transformed to an equivalent problem with a smooth loss and convex regularizer. However, as the proximal step with the transformed regularizer has to be solved numerically and exact proximal step is required, usage with the proximal algorithm may not be efficient. We show that this can be alleviated by the proposed inexact proximal algorithm. Finally, in the experiments, we demonstrate the above-mentioned advantages of optimizing the transformed problems instead of the original ones on various tasks, and show that running algorithms on the transformed problems can be much faster than running the state-of-the-art on the original problems.

The rest of the paper is organized as follows. Section 2 provides a review on the related works. The main idea for problem transformation is presented in Section 3, and its usage with various algorithms are discussed in Section 4. Experimental results are shown in Section 5, and the last section gives some concluding remarks. All the proofs are in Appendix A. Note that this paper extends a shorter version published in the International Conference of Machine Learning (Yao and Kwok, 2016).

Notation: We denote vectors and matrices by lowercase and uppercase boldface letters, respectively. For a vector $x \in \mathbb{R}^d$, $\|x\|_2 = (\sum_{i=1}^d |x_i|^2)^{1/2}$ is its ℓ_2-norm, $\mathrm{Diag}(x)$ returns a diagonal matrix $X \in \mathbb{R}^{d\times d}$ with $X_{ii} = x_i$. For a matrix $X \in \mathbb{R}^{m\times n}$ (where $m \le n$ without loss of generality), its nuclear norm is $\|X\|_* = \sum_{i=1}^m \sigma_i(X)$, where $\sigma_i(X)$'s are the singular values of X, and its Frobenius norm is $\|X\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n X_{ij}^2}$, and $\|X\|_\infty = \max_{i,j} |X_{ij}|$. For a square matrix X, $X \in \mathcal{S}_+$ indicates it is positive semidefinite. For two matrices X and Y, $\langle X, Y\rangle = \sum_{i,j} X_{ij}Y_{ij}$. For a smooth function f, $\nabla f(x)$ is its gradient at x. For a convex but nonsmooth f, $\partial f(x) = \{u : f(y) \ge f(x) + \langle u, y - x\rangle\}$ is its subdifferential at x, and $g \in \partial f(x)$ is a subgradient.

Given a function (e.g., f), a bar on top (e.g., $\bar{f}$) indicates the function is smooth but not necessarily convex, and a breve on top (e.g., $\breve{f}$) means that it is convex but not necessarily smooth.

2. Related Works

In this section, we review some popular algorithms for solving (1). Here, f is assumed to be Lipschitz smooth.

2.1 Convex-Concave Procedure (CCCP)

The convex-concave procedure (CCCP) (Yuille and Rangarajan, 2002; Lu, 2012) is a popular and general solver for (1). It assumes that F can be decomposed as a difference of convex (DC) functions (Hiriart-Urruty, 1985), i.e., $F(x) = \breve{F}(x) + \hat{F}(x)$ where $\breve{F}$ is convex and $\hat{F}$ is concave. In each

CCCP iteration, $\hat{F}$ is linearized at x_t, and x_{t+1} is generated as

$$x_{t+1} = \arg\min_x \breve{F}(x) + \hat{F}(x_t) - (x - x_t)^\top s_t, \tag{2}$$

where $s_t \in \partial(-\hat{F}(x_t))$ is a subgradient. This is a convex problem and can be easier than directly minimizing F.

However, CCCP is expensive as (2) needs to be exactly solved. Sequential convex programming (SCP) (Lu, 2012) improves its efficiency when F is of the form in (1). It assumes that f is L-Lipschitz smooth (possibly nonconvex); while g can be nonconvex, but admits a DC decomposition as $g(x) = \breve{\zeta}(x) + \hat{\zeta}(x)$ where $\breve{\zeta}$ is convex and $\hat{\zeta}$ is concave. It then generates x_{t+1} as

$$\begin{aligned} x_{t+1} &= \arg\min_x f(x_t) + (x - x_t)^\top \nabla f(x_t) + \frac{L}{2}\|x - x_t\|_2^2 + \breve{\zeta}(x) + \hat{\zeta}(x_t) - (x - x_t)^\top s_t \\ &= \arg\min_x \frac{1}{2}\|x - x_t - s_t + \frac{1}{L}\nabla f(x_t)\|_2^2 + \breve{\zeta}(x), \end{aligned} \tag{3}$$

where $s_t \in \partial\left(-\hat{\zeta}(x_t)\right)$. When $\breve{\zeta}$ is a simple function, (3) has a closed-form solution, and SCP can be faster than CCCP. For example, when $\breve{\zeta}(x) = \|x\|_1$, (3) is the proximal step of the ℓ_1-norm and has a closed-form solution (Tibshirani, 1996). However, convergence of SCP is still slow in general (Gong et al., 2013; Zhong and Kwok, 2014; Li and Lin, 2015).

2.2 Proximal Algorithm

The proximal algorithm (Parikh and Boyd, 2013) has been popularly used for optimization problems of the form in (1). Let f be convex and L-Lipschitz smooth, and g is convex. The proximal algorithm generates iterates $\{x_t\}$ as

$$\begin{aligned} x_{t+1} &= \arg\min_x f(x_t) + (x - x_t)^\top \nabla f(x_t) + \frac{L}{2}\|x - x_t\|_2^2 + g(x) \\ &= \text{prox}_{\frac{1}{L}g}\left(x_t - \frac{1}{L}\nabla f(x_t)\right), \end{aligned}$$

where $\text{prox}_g(z) \equiv \arg\min_x \frac{1}{2}\|x - z\|_2^2 + g(x)$ is the proximal step, The proximal algorithm converges at a rate of $O(1/T)$. This can be further accelerated to $O(1/T^2)$ by modifying the generation of $\{x_t\}$ as (Beck, 2009; Nesterov, 2013):

$$\begin{aligned} y_t &= x_t + \frac{\alpha_{t-1} - 1}{\alpha_t}(x_t - x_{t-1}), \\ x_{t+1} &= \text{prox}_{\frac{1}{L}g}\left(y_t - \frac{1}{L}\nabla f(y_t)\right), \end{aligned}$$

where $\alpha_0 = \alpha_1 = 1$ and $\alpha_{t+1} = \frac{1}{2}(\sqrt{4\alpha_t^2 + 1} + 1)$.

Recently, the proximal algorithm has been extended to nonconvex optimization. In particular, NIPS (Sra, 2012), IPiano (Ochs et al., 2014) and UAG (Ghadimi and Lan, 2016) allow f to be nonconvex, while g is still required to be convex. GIST (Gong et al., 2013), IFB (Bot et al., 2016) and nmAPG (Li and Lin, 2015) further remove this restriction and allow g to be nonconvex. It is desirable that the proximal step has a closed-form solution. This is true for many convex regularizers such as the lasso regularier (Tibshirani, 1996), tree-structured lasso regularizer (Liu and Ye, 2010; Jenatton et al., 2011) and sparse group lasso regularizer (Jacob et al., 2009). However, when g is nonconvex, a closed-form solution only exists when g is simple (e.g., nonconvex lasso regularizer (Gong et al., 2013)), but not for the more general cases (e.g., nonconvex tree-structured lasso regularizer (Zhong and Kwok, 2014)).

On the other hand, Zhong and Kwok (2014) used the proximal average (Bauschke et al., 2008) to handle complicated g's of the form $g(x) = \sum_{i=1}^K \mu_i g_i(x)$, where each g_i has a simple proximal step. The iterates are generated as

$$x_{t+1} = \sum_{i=1}^K \mu_i \cdot \text{prox}_{\frac{\mu_i}{L}g_i}\left(x_t - \frac{1}{L}\nabla f(x_t)\right) / \sum_{i=1}^K \mu_i.$$

Each of the constituent proximal steps $\text{prox}_{\frac{\mu_i}{L}g_i}(\cdot)$ can be computed inexpensively, and the per-iteration complexity is low. However, it only approximates $\text{prox}_g(z)$ and consequently also the original problem (1). Empirically, its convergence can be slow.

2.3 Frank-Wolfe (FW) Algorithm

The FW algorithm (Frank and Wolfe, 1956) is used for solving optimization problems of the form

$$\min_x f(x) \ : \ x \in \mathcal{C}, \tag{4}$$

where f is Lipschitz-smooth and convex, and $\mathcal{C}$ is a compact convex set. Recently, it has been popularly used in machine learning (Jaggi, 2013). In each iteration, the FW algorithm generates the next iterate x_{t+1} as

$$s_t = \arg\min_{s \in \mathcal{C}} s^\top \nabla f(x_t), \tag{5}$$

$$\gamma_t = \arg\min_{\gamma \in [0,1]} f((1-\gamma)x_t + \gamma s_t), \tag{6}$$

$$x_{t+1} = (1 - \gamma_t)x_t + \gamma_t s_t. \tag{7}$$

Here, (5) is a linear subproblem which can often be easily solved; (6) performs line search, and the next iterate x_{t+1} is generated from a convex combination of x_t and s_t in (7). The FW algorithm has a convergence rate of $O(1/T)$ (Jaggi, 2013).

In this paper, we will focus on using the FW algorithm to learn a low-rank matrix $X \in \mathbb{R}^{m \times n}$. Without loss of generality, we assume that $m \leq n$. The nuclear norm $\|X\|_*$ of X is the tightest convex envelope of rank(X), and is often used as a low-rank regularizer (Candès and Recht, 2009). The low-rank matrix learning problem can be written as

$$\min_X f(X) + \mu\|X\|_*, \tag{8}$$

where f is the loss. For example, in matrix completion (Candès and Recht, 2009),

$$f(X) = \frac{1}{2}\|\mathcal{P}_\Omega(X - O)\|_F^2, \tag{9}$$

where O is the observed incomplete matrix, $\Omega \in \{0,1\}^{m \times n}$ contains indices to the observed entries in O, and $[P_\Omega(A)]_{ij} = A_{ij}$ if $\Omega_{ij} = 1$, and 0 otherwise.

The FW algorithm for this nuclear norm regularized problem is shown in Algorithm 1 (Zhang et al., 2012). Let the iterate at the tth iteration be X_t. As in (5), the following linear subproblem has to be solved (Jaggi, 2013):

$$\min_{S:\|S\|_*\leq 1} \langle S, \nabla f(X_t)\rangle. \tag{10}$$

This can be obtained from the rank-one SVD of $\nabla f(X_t)$ (step 3). Similar to (6), line search is performed at step 4. As a rank-one matrix is added into X_t in each iteration, it is convenient to write X_t as

$$\sum_{i=1}^{t} u_i v_i^\top = U_t V_t^\top, \tag{11}$$

where $U_t = [u_1, \ldots, u_t]$ and $V_t = [v_1, \ldots, v_t]$. The FW algorithm has a convergence rate of $O(1/T)$ (Jaggi, 2013). To make it empirically faster, Algorithm 1 also performs optimization at step 6 (Laue, 2012; Zhang et al., 2012). Substituting $\|X\|_* = \min_{X=UV^\top} \frac{1}{2}\left(\|U\|_F^2 + \|V\|_F^2\right)$ (Srebro et al., 2004) into (8), we have the following local optimization problem:

$$\min_{U,V} f(UV^\top) + \frac{\mu}{2}(\|U\|_F^2 + \|V\|_F^2). \tag{12}$$

This can be solved by standard solvers such as L-BFGS (Nocedal and Wright, 2006).

Algorithm 1 Frank-Wolfe algorithm for problem (8) with f convex (Zhang et al., 2012).

1: $U_1 = [\,]$ and $V_1 = [\,]$;
2: **for** $t = 1 \ldots T$ **do**
3: $[u_t, s_t, v_t] = \text{rank1SVD}(\nabla f(X_t))$;
4: $[\alpha_t, \beta_t] = \arg\min_{\alpha\geq 0, \beta\geq 0} f(\alpha X_t + \beta u_t v_t^\top) + \mu(\alpha\|X_t\|_* + \beta)$;
5: $\bar{U}_t = \left[\sqrt{\alpha_t}U_t; \sqrt{\beta_t}u_t\right]$ and $\bar{V}_t = \left[\sqrt{\alpha_t}V_t; \sqrt{\beta_t}v_t\right]$;
6: obtain $[U_{t+1}, V_{t+1}]$ from (12), using $\bar{U}_t$ and $\bar{V}_t$ for warm-start; // $X_{t+1} = U_{t+1}V_{t+1}^\top$
7: **end for**
8: **return** U_{T+1} and V_{T+1}.

2.4 Alternating Direction Method of Multipliers (ADMM)

ADMM is a simple but powerful algorithm first introduced in the 1970s (Glowinski and Marroco, 1975). Recently, it has been popularly used in diverse fields such as machine learning, data mining and image processing (Boyd et al., 2011). It can be used to solve optimization problems of the form

$$\min_{x,y}\ f(x) + g(y)\ :\ Ax + By = c, \tag{13}$$

where f, g are convex functions, and A, B (resp. c) are constant matrices (resp. vector) of appropriate sizes. Consider the augmented Lagrangian $\mathcal{L}(x, y, u) = f(x) + g(y) + u^\top(Ax + By - c) + \frac{\tau}{2}\|Ax + By - c\|_2^2$, where u is the vector of Lagrangian multipliers, and $\tau > 0$ is a penalty parameter. At the tth iteration of ADMM, the values of x, y and u are updated as

$$x_{t+1} = \arg\min_x \mathcal{L}(x, y_t, u_t), \tag{14}$$

$$y_{t+1} = \arg\min_y \mathcal{L}(x_{t+1}, y, u_t), \tag{15}$$

$$u_{t+1} = u_t + \tau(Ax_{t+1} + By_{t+1} - c). \tag{16}$$

By minimizing $L(x, y, u_k)$ w.r.t. x and y in an alternating manner ((14) and (15)), ADMM can more easily decompose the optimization problem when f, g are separable.

In this paper, we will focus a special case of (13), namely, the consensus optimization problem:

$$\min_{y,x^1,\ldots,x^M} \sum_{i=1}^{M} f_i(x^i) + g(y)\quad :\quad x^1 = \cdots = x^M = y. \tag{17}$$

Here, each f_i is Lipschitz-smooth, x^i is the variable in the local objective f_i, and y is the global consensus variable. This type of problems is often encountered in machine learning, signal processing and wireless communication (Bertsekas and Tsitsiklis, 1989; Boyd et al., 2011). For example, in regularized risk minimization, y is the model parameter, f_i is the regularized risk functional defined on the ith data subset, and g is the regularizer. The augmented Lagrangian for (17) is

$$\mathcal{L}(y, x^1, \ldots, x^M, u^1, \ldots, u^M) = g(y) + \sum_{i=1}^{M} f_i\left(x^i\right) + (u^i)^\top(x^i - y) + \frac{\tau}{2}\|x^i - y\|_2^2,$$

where u^i is the dual variable for the constraint $x^i = y$. Substituting into (14)-(16), we have

$$x_{t+1}^i = \arg\min_{x^i} f_i(x^i) + (u_t^i)^\top(x^i - y_t) + \frac{\tau}{2}\|x^i - y_t\|_2^2,\quad i = 1, \ldots, M, \tag{18}$$

$$y_{t+1} = \arg\min_y \frac{1}{2}\left\|y - \sum_{i=1}^{M}\left(x_t^i + \frac{1}{\tau}u_t^i\right)\right\|_2^2 + \frac{1}{\tau}g(y) = \text{prox}_{\frac{1}{\tau}g}\left(\sum_{i=1}^{M} x_t^i + \frac{1}{\tau}u_t^i\right), \tag{19}$$

$$u_{t+1} = u_t + \tau\left(x_{t+1}^i + y_{t+1}\right),\quad i = 1, \ldots, M.$$

When f_i is smooth and g is convex, ADMM converges to a critical point of (17) (Hong et al., 2016). However, when g is nonconvex, its convergence is still an open issue.

3. Shifting Nonconvexity from Regularizer to Loss

In recent years, a number of nonconvex regularizers have been proposed. Examples include the Geman penalty (GP) (Geman and Yang, 1995), log-sum penalty (LSP) (Candès et al., 2008) and Laplace penalty (Trzasko and Manduca, 2009). In general, learning with nonconvex regularizers is much more difficult than learning with convex regularizers. In this section, we show how to move the nonconvex component from the nonconvex regularizers to the loss function. Existing algorithms can then be reused to learn with the convexified regularizers.

First, we make the following standard assumptions on (1).

A1. F is bounded from below and $\lim_{\|x\|_2\to\infty} F(x) = \infty$;

A2. f is L-Lipschitz smooth (i.e., $\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2$), but possibly nonconvex.

Let κ be a function that is concave, non-decreasing, ρ-Lipschitz smooth with κ' non-differentiable at finite points, and $\kappa(0) = 0$. All nonconvex regularizers in Table 1 satisfy the requirements on κ.

In this paper, we consider g of the following forms:

	$\kappa(\alpha)$	$\kappa'(\alpha)$	κ_0	ρ
GP (Geman and Yang, 1995)	$\frac{\beta\alpha}{\theta+\alpha}$	$\frac{\beta\theta}{(\theta+\alpha)^2}$	$\frac{\beta}{\theta}$	$\frac{2\beta}{\theta^2}$
LSP (Candès et al., 2008)	$\beta\log(1+\frac{\alpha}{\theta})$	$\frac{\beta}{\theta+\alpha}$	$\frac{\beta}{\theta}$	$\frac{\beta}{\theta^2}$
MCP (Zhang, 2010a)	$\begin{cases}\beta\alpha-\frac{\alpha^2}{2\theta} & \alpha\le\beta\theta\\ \frac{1}{2}\theta\beta^2 & \alpha>\beta\theta\end{cases}$	$\begin{cases}\beta-\frac{\alpha}{\theta} & \alpha\le\beta\theta\\ 0 & \alpha>\beta\theta\end{cases}$	β	$\frac{1}{\theta}$
Laplace (Trzasko and Manduca, 2009)	$\beta(1-\exp(-\frac{\alpha}{\theta}))$	$\frac{\beta}{\theta}\exp\left(-\frac{\alpha}{\theta}\right)$	$\frac{\beta}{\theta}$	$\frac{\beta}{\theta^2}$
SCAD (Fan and Li, 2001)	$\begin{cases}\beta\alpha & \alpha\le\beta\\ \frac{-\alpha^2+2\theta\beta\alpha-\beta^2}{2(\theta-1)} & \beta<\alpha\le\theta\beta\\ \frac{\beta^2(1+\theta)}{2} & \alpha>\theta\beta\end{cases}$	$\begin{cases}\beta & \alpha\le\beta\\ \frac{-\alpha+\theta\beta}{\theta-1} & \beta<\alpha\le\theta\beta\\ 0 & \alpha>\theta\beta\end{cases}$	β	$\frac{1}{\theta-1}$

Table 1: Example nonconvex regularizers. Here, $\kappa_0 \equiv \kappa'(0)$ and $\beta > 0$. For SCAD, $\theta > 2$, whereas for others, $\theta > 0$.

C1. $g(x) = \sum_{i=1}^K \mu_i g_i(x)$, where $\mu_i \ge 0$,

$$g_i(x) = \kappa(\|A_i x\|_2), \tag{20}$$

and A_i is a matrix. When κ is the identity function, $g(x)$ reduces to the convex regularizer $\sum_{i=1}^K \mu_i \|A_i x\|_2$. By using different A_i's, g becomes various structured sparsity regularizers such as the group lasso (Jacob et al., 2009), fused lasso (Tibshirani et al., 2005), and graphical lasso (Jacob et al., 2009).

C2. $g(X) = \mu \sum_{i=1}^m \kappa(\sigma_i(X))$, where X is a matrix and $\mu \ge 0$. When κ is the identity function, g reduces to the nuclear norm.

First, consider g in **C1**. Rewrite each nonconvex g_i in (20) as

$$g_i(x) = \bar{g}_i(x) + \kappa_0\|A_i x\|_2, \tag{21}$$

where $\kappa_0 = \kappa'(0)$, and $\bar{g}_i(x) = \kappa(\|A_i x\|_2) - \kappa_0\|A_i x\|_2$. Obviously, $\kappa_0\|A_i x\|_2$ is convex but nonsmooth. The following shows that $\bar{g}_i$, though nonconvex, is concave and Lipschitz smooth.

Proposition 1 *$\kappa(\|z\|_2) - \kappa_0\|z\|_2$, where $z \in \mathbb{R}^d$, is concave and 2ρ-Lipschitz smooth.*

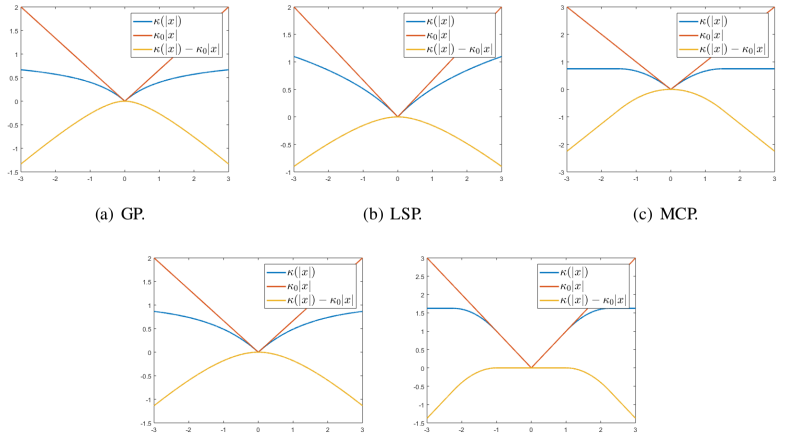
Corollary 2 *$\bar{g}_i$ is concave and Lipschitz smooth with modulus $\bar{L}_i = 2\rho\|A_i\|_F$.*

Corollary 3 *$g(x)$ can be decomposed as $\bar{g}(x) + \breve{g}(x)$, where $\bar{g}(x) \equiv \sum_{i=1}^K \mu_i \bar{g}_i(x)$ is concave and Lipschitz-smooth, while $\breve{g}(x) \equiv \kappa_0 \sum_{i=1}^K \mu_i \|A_i x\|_2$ is convex but nonsmooth.*

Remark 4 *When $A_i = Diag(e_i)$, where e_i is the unit vector for dimension i, $\|A_i x\|_2 = |x_i|$ and*

$$g(x) = \sum_{i=1}^d \mu_i \kappa(\|A_i x\|_2) = \sum_{i=1}^d \mu_i \kappa(|x_i|). \tag{22}$$

Using Corollary 3, g can be decomposed as $\bar{g}(x) + \breve{g}(x)$, where $\bar{g}(x) \equiv \sum_{i=1}^d \mu_i(\kappa(|x_i|) - \kappa_0|x_i|)$ is concave and 2ρ-Lipschitz smooth, while $\breve{g}(x) \equiv \kappa_0 \sum_{i=1}^d \mu_i|x_i|$ is convex and nonsmooth. When $d = 1$ and $\mu_1 = 1$, an illustration of $g(x) = \kappa(|x|)$, $\bar{g}(x) = \kappa(|x|) - \kappa_0|x|$ and $\breve{g}(x) = \kappa_0|x|$ for the various nonconvex regularizers is shown in Figure 2. When κ is the identity function and $\mu_1 = \cdots = \mu_m = \mu$, g in (22) reduces to the lasso regularizer $\mu\|x\|_1$.



(a) GP. (b) LSP. (c) MCP.

(d) Laplace. (e) SCAD.

Figure 2: Decompositions of the regularizers in Table 1 ($\beta = 1$ for all regularizers; $\theta = 2.25$ for SCAD and 1.5 for others).

Using Corollary 3, problem (1) can then be rewritten as

$$\min_x \bar{f}(x) + \breve{g}(x), \tag{23}$$

where $\bar{f}(x) \equiv f(x) + \bar{g}(x)$. Note that $\bar{f}$ (which can be viewed as an augmented loss) is Lipschitz smooth while $\breve{g}$ (viewed as a convexified regularizer) is convex but possibly nonsmooth. In other words, nonconvexity is shifted from the regularizer g to the loss f, while ensuring that the augmented loss is smooth.

When X is a matrix, similar to Corollary 3, the following Proposition holds for g in **C2**.

Proposition 5 *Any g in **C2** can be decomposed as $\bar{g}(X) + \breve{g}(X)$, where*

$$\bar{g}(X) \equiv \mu \sum_{i=1}^m \kappa(\sigma_i(X)) - \mu\kappa_0\|X\|_* \tag{24}$$

is concave and 2ρ-Lipschitz smooth, and $\breve{g}(X) \equiv \kappa_0\|X\|_*$ is convex and nonsmooth.

Since $\bar{g}$ is concave and $\breve{g}$ is convex, the nonconvex regularizer $g = \breve{g} - (-\bar{g})$ can be viewed as a difference of convex functions (DC) (Hiriart-Urruty, 1985). Lu (2012); Gong et al. (2013); Zhong and Kwok (2014) also relied on DC decompositions of the nonconvex regularizer. However, they do not utilize this in the computational procedures, while we use the DC decomposition to simplify the regularizers. As will be seen, though the DC decomposition of a nonconvex function is not unique in general, the particular one proposed here is crucial for efficient optimization.

4. Example Use Cases

In this section, we provide concrete examples to show how the proposed convexification scheme can be used with various optimization algorithms. An overview is summarized in Table 2.

	section	advantages
proximal algorithm	4.1, 4.6	cheaper proximal step
FW algorithm	4.2	cheaper linear subproblem
(consensus) ADMM	4.3	cheaper proximal step; provide convergence guarantee
SVRG	4.4	cheaper proximal step; provide convergence guarantee
mOWL-QN	4.5	simpler analysis; capture curvature information

Table 2: Using the proposed convexification scheme with various algorithms.

4.1 Proximal Algorithms

In this section, we provide example applications on using the proximal algorithm for nonconvex structured sparse learning. The proximal algorithm has been commonly used for learning with convex regularizers (Parikh and Boyd, 2013). With a nonconvex regularizer, the underlying proximal step becomes much more challenging. Gong et al. (2013); Li and Lin (2015) and Bot et al. (2016) extended proximal algorithm to simple nonconvex g, but cannot handle more complicated nonconvex regularizers such as the tree-structured lasso regularizer (Liu and Ye, 2010; Schmidt et al., 2011), sparse group lasso regularizer (Jacob et al., 2009) and total variation regularizer (Nikolova, 2004). Using the proximal average (Bauschke et al., 2008), Zhong and Kwok (2014) can handle nonconvex regularizers of the form $g = \sum_{i=1}^{K} \mu_i g_i$, where each g_i is simple. However, the solutions obtained are only approximate. General nonconvex optimization techniques such as the concave-convex procedure (CCCP) (Yuille and Rangarajan, 2002) or its variant sequential convex programming (SCP) (Lu, 2012) can also be used, though they are slow in general (Gong et al., 2013; Zhong and Kwok, 2014).

Using the proposed transformation, one only needs to solve the proximal step of a standard convex regularizer instead of that of a nonconvex regularizer. This allows reuse of existing solutions for the proximal step and is much less expensive. As proximal algorithms have the same convergence guarantee for convex and nonconvex f (Gong et al., 2013; Li and Lin, 2015; Yao et al., 2017), solving the transformed problem can be much faster. The following gives some specific examples.

4.1.1 NONCONVEX SPARSE GROUP LASSO

In sparse group lasso, the feature vector x is divided into groups. Let $\mathcal{G}_j$ be the set containing dimensions of x that are in group j, and $\left[x_{\mathcal{G}_j}\right]_i = x_i$ if $i \in \mathcal{G}_j$ and 0 otherwise. Given training samples $\{(a_1, y_1), \ldots, (a_N, y_N)\}$, (convex) sparse group lasso is formulated as (Jacob et al., 2009):

$$\min_x \sum_{i=1}^{N} \ell(y_i, a_i^\top x) + \lambda \|x\|_1 + \sum_{j=1}^{K} \mu_j \|x_{\mathcal{G}_j}\|_2, \tag{25}$$

where ℓ is a smooth loss, and K is the number of (non-overlapping) groups.

For the nonconvex extension, the regularizer becomes

$$g(x) = \lambda \sum_{i=1}^{d} \kappa(|x_i|) + \sum_{j=1}^{K} \mu_j \kappa(\|x_{\mathcal{G}_j}\|_2). \tag{26}$$

Using Corollary 3 and Remark 4, the convexified regularizer is $\breve{g}(x) = \kappa_0(\lambda\|x\|_1 + \sum_{j=1}^{K} \mu_j \|x_{\mathcal{G}_j}\|_2)$. Its proximal step can be easily computed by the algorithm in (Yuan et al., 2011). Specifically, the proximal operator of $\breve{g}$ can be obtained by computing $\text{prox}_{\mu_j\|\cdot\|_2}(\text{prox}_{\lambda\|\cdot\|_1}(x_{\mathcal{G}_j}))$ for each group separately. This can then be used with any proximal algorithm that can handle nonconvex objectives (as $\bar{f}$ is nonconvex). In particular, we will adopt the state-of-the-art nonmontonic APG (nmAPG) algorithm (Li and Lin, 2015) (shown in Algorithm 2). On the other hand, note that nmAPG cannot be directly used with the nonconvex regularizer g in (26), as the corresponding proximal step has no inexpensive closed-form solution.

Algorithm 2 Nonmonotonic APG (nmAPG) (Li and Lin, 2015).

Initialize $z_1 = x_1 = x_0$, $\alpha_0 = 0$, $\alpha_1 = 1$, $\eta \in [0, 1)$, $c_1 = F(x_1)$, $q_1 = 1$, and stepsize $\tau > \bar{L}$, $\delta \in (0, \tau - \bar{L})$;
for $t = 1, \ldots, T$ **do**
  $y_t = x_t + \frac{\alpha_{t-1}}{\alpha_t}(z_t - x_t) + \frac{\alpha_{t-1}-1}{\alpha_t}(x_t - x_{t-1})$;
  $z_{t+1} = \text{prox}_{\frac{1}{\tau}\breve{g}}(y_t - \frac{1}{\tau}\nabla \bar{f}(y_t))$;
  if $F(z_{t+1}) \le c_t - \frac{\delta}{2}\|z_{t+1} - y_t\|_2^2$ **then**
    $x_{t+1} = z_{t+1}$;
  else
    $v_{t+1} = \text{prox}_{\frac{1}{\tau}\breve{g}}(x_t - \frac{1}{\tau}\nabla \bar{f}(x_t))$;
    $x_{t+1} = \begin{cases} z_{t+1} & F(z_{t+1}) \le F(v_{t+1}) \\ v_{t+1} & \text{otherwise} \end{cases}$;
  end if
  $\alpha_{t+1} = \frac{1}{2}(\sqrt{4\alpha_t^2 + 1} + 1)$;
  $q_{t+1} = \eta q_t + 1$;
  $c_{t+1} = \frac{\eta q_t c_t + F(x_{t+1})}{q_{t+1}}$;
end for
return x_{T+1};

4.1.2 NONCONVEX TREE-STRUCTURED GROUP LASSO

In (convex) tree-structured group lasso (Liu and Ye, 2010; Jenatton et al., 2011), the dimensions in x are organized as nodes in a tree, and each group corresponds to a subtree. The regularizer is of the form $\sum_{j=1}^{K} \lambda_j \|x_{\mathcal{G}_j}\|_2$. Interested readers are referred to (Liu and Ye, 2010) for details.

For the nonconvex extension, $g(x)$ becomes $\sum_{j=1}^{K} \lambda_j \kappa(\|x_{\mathcal{G}_j}\|_2)$. Again, there is no closed-form solution of its proximal step. On the other hand, the convexified regularizer is $\breve{g}(x) \equiv \kappa_0 \sum_{j=1}^{K} \lambda_j \|x_{\mathcal{G}_j}\|_2$. As shown in (Liu and Ye, 2010), its proximal step can be computed efficiently by processing all the groups once in some appropriate order.

4.1.3 NONCONVEX TOTAL VARIATION (TV) REGULARIZER

In an image, nearby pixels are usually strongly correlated. The TV regularizer captures such behavior by assuming that changes between nearby pixels are small. Given an image $X \in \mathbb{R}^{m \times n}$, the TV regularizer is defined as $\text{TV}(X) = \|D_v X\|_1 + \|X D_h\|_1$ (Nikolova, 2004), $D_v = \begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix} \in \mathbb{R}^{(m-1)\times m}$ and $D_h = \begin{bmatrix} -1 & & \\ 1 & \ddots & \\ & \ddots & -1 \\ & & 1 \end{bmatrix} \in \mathbb{R}^{n \times (n-1)}$ are the horizontal and vertical partial derivative operators, respectively. Thus, it is popular on image processing problems, such as image denoising and deconvolution (Nikolova, 2004; Beck and Teboulle, 2009).

As in previous sections, the nonconvex extension of TV regularizer can be defined as

$$\sum_{i=1}^{m-1} \sum_{j=1}^{m} \kappa\left(\left|[D_v X]_{ij}\right|\right) + \sum_{i=1}^{n} \sum_{j=1}^{n-1} \kappa\left(\left|[X D_h]_{ij}\right|\right). \tag{27}$$

Again, it is not clear how its proximal step can be efficiently computed. Instead, with the proposed transformation, the transformed problem is

$$\min_X \bar{f}(X) + \mu\kappa_0 \text{TV}(X),$$

where μ is the regularization parameter, $\bar{f}(X) = f(X) + \mu \sum_{i=1}^{m-1} \sum_{j=1}^{m} (\kappa(|[D_v X]_{ij}|) - \kappa_0 |[D_v X]_{ij}|) + \mu \sum_{i=1}^{n} \sum_{j=1}^{n-1} (\kappa(|[X D_h]_{ij}|) - \kappa_0 |[X D_h]_{ij}|)$ is concave and Lipschitz smooth. One then only needs to compute the proximal step of the standard TV regularizer.

Unlike the proximal steps in Sections 4.1.1 and 4.1.2, the proximal step of the TV regularizer has no closed-form solution and needs to be solved iteratively. In this case, Schmidt et al. (2011) showed that using inexact proximal steps can make proximal algorithms faster, but they only considered the situation where both f and g are convex. In the following, we extend nmAPG (Algorithm 2), which can be used with nonconvex objectives, to allow for inexact proximal steps (steps 5 and 9 of Algorithm 3). However, Lemma 2 of (Li and Lin, 2015), which is key to the convergence of nmAPG, no longer holds because of the inexact proximal step. To fix this problem, in step 6 of Algorithm 3, we use $F(X_t)$ instead of c_t in Algorithm 2. We also drop the comparison of $F(Z_{t+1})$ and $F(V_{t+1})$ (originally in step 9 of Algorithm 2).

Inexactness of the proximal step can be controlled as follows. Let $P = X - \frac{1}{\tau}\nabla \bar{f}(X)$, and $h(X) \equiv \frac{1}{2}\|X - P\|_F^2 + \frac{1}{\tau}\breve{g}(X)$ be the objective in $\text{prox}_{\frac{1}{\tau}\breve{g}}(P)$. As $\breve{g}(X) = \kappa_0 \text{TV}(X)$ is convex,

Algorithm 3 Inexact nmAPG.

1: Initialize $\tilde{Z}_1 = X_1 = X_0$, $\alpha_0 = 0$, $\alpha_1 = 1$ and stepsize $\tau > \bar{L}$, $\delta \in (0, \tau - \bar{L})$;
2: **for** $t = 1, \ldots, T$ **do**
3: choose tolerance ϵ_t;
4: $Y_t = X_t + \frac{\alpha_{t-1}}{\alpha_t}(Z_t - X_t) + \frac{\alpha_{t-1}-1}{\alpha_t}(X_t - X_{t-1})$;
5: $\tilde{Z}_{t+1}$ = approximate $\text{prox}_{\frac{1}{\tau}\breve{g}}(Y_t - \frac{1}{\tau}\nabla \bar{f}(Y_t))$, with inexactness $\vartheta_{t+1} \le \epsilon_t$;
6: **if** $F(\tilde{Z}_{t+1}) \le F(X_t) - \frac{\delta}{2}\|\tilde{Z}_{t+1} - Y_t\|_F^2$ **then**
7: $X_{t+1} = \tilde{Z}_{t+1}$;
8: **else**
9: X_{t+1} = approximate $\text{prox}_{\frac{1}{\tau}\breve{g}}(X_t - \frac{1}{\tau}\nabla \bar{f}(X_t))$, with inexactness $\vartheta_{t+1} \le \epsilon_t$;
10: **end if**
11: $\alpha_{t+1} = \frac{1}{2}(\sqrt{4\alpha_t^2 + 1} + 1)$;
12: **end for**
13: **return** X_{T+1};

h is also convex. Let $\tilde{X}$ be an inexact solution of this proximal step. The inexactness $h(\tilde{X}) - h(\text{prox}_{\frac{1}{\tau}\breve{g}}(P))$ is upper-bounded by the duality gap $\vartheta \equiv h(\tilde{X}) - \mathcal{D}(\tilde{W})$, where $\mathcal{D}$ is the dual of h, and $\tilde{W}$ is the corresponding dual variable. In steps 5 and 9 of Algorithm 3, we solve the proximal step until the duality gap ϑ_{t+1} is smaller than a given threshold ϵ_t. The following Theorem shows convergence of Algorithm 3.

Theorem 6 *Let $\sum_{t=1}^{\infty} \epsilon_t < \infty$. The sequence $\{X_t\}$ generated from Algorithm 3 has at least one limit point, and every limit point is also a critical point of* (1).

If the proximal step is exact, $\|V_t - \text{prox}_{\frac{1}{\tau}\breve{g}}(V_t - \frac{1}{\tau}\nabla \bar{f}(V_t))\|_F^2$ can be used to measure how far V_t is from a critical point (Gong et al., 2013; Ghadimi and Lan, 2016). In Algorithm 3, the proximal step is inexact, and X_{t+1} is an inexact solution to $\text{prox}_{\frac{1}{\tau}\breve{g}}(V_t - \frac{1}{\tau}\nabla \bar{f}(V_t))$, where $V_t = Y_t$ if step 7 is executed, and $V_t = X_t$ if step 9 is executed. As X_{t+1} converges to a critical point of (1), we propose using $d_t \equiv \|X_{t+1} - V_t\|_F^2$ to measure how far X_{t+1} is from a critical point. The following Proposition shows a $O(1/T)$ convergence rate on $\min_{t=1,\ldots,T} d_t$.

Proposition 7 *(i) $\lim_{t\to\infty} d_t = 0$; and (ii) $\min_{t=1,\ldots,T} d_t$ converges to zero at a rate of $O(1/T)$.*

Note that the (exact) nmAPG in Algorithm 2 cannot handle the nonconvex g in (27) efficiently, as the corresponding proximal step has no closed-form solution but has to be solved exactly. Even the proposed inexact nmAPG (Algorithm 3) cannot be directly used with nonconvex g. As the dual of the nonconvex proximal step is difficult to derive and the optimal duality gap is nonzero in general, the proximal step's inexactness cannot be easily controlled.

Remark 8 *As mentioned in Section 3, the proposed decomposition of the nonconvex regularizer g can be regarded as a DC decomposition, which is not unique in general. For example, in Section 4.1.1, we might try to add a quadratic term to convexify the nonconvex sparse group lasso regularizer. Specifically, we can decompose $g(x)$ in (26) as $\breve{\varsigma}(x) + \hat{\varsigma}(x)$, where*

$$\breve{\varsigma}(x) = \lambda \sum_{i=1}^{d} \left(\kappa(|x_i|) + \frac{\rho}{2}x_i^2\right) + \sum_{j=1}^{K} \mu_j \left(\kappa(\|x_{\mathcal{G}_j}\|_2) + \frac{\rho}{2}\|x_{\mathcal{G}_j}\|_2^2\right), \tag{28}$$

and $\hat{\varsigma}(x) = -\frac{\rho}{2}\sum_{j=1}^{K}(\mu_j+\lambda)\|x_{\mathcal{G}_j}\|_2^2$. *It can be easily shown that* $\hat{\varsigma}$ *is concave, and the following Proposition 9 shows that* $\check{\varsigma}$ *is convex. Thus,* F *can be transformed as* $F(x) = \bar{f}(x) + \check{\varsigma}(x)$, *where* $\bar{f}(x) = f(x) + \hat{\varsigma}(x)$ *is Lipschitz-smooth, and* $\check{\varsigma}$ *is convex but nonsmooth. However, the proximal step associated with* $\check{\varsigma}$ *has no simple closed-form solution.*

Proposition 9 $\kappa(\|\cdot\|) + \frac{\rho}{2}\|\cdot\|^2$, *where* $\|\cdot\|$ *is a norm, is convex.*

Similarly, in Section 4.1.2, we can also add a quadratic term to convexify the nonconvex tree-structured group lasso regularizer as $\sum_{j=1}^{K}\lambda_j(\kappa(\|x_{\mathcal{G}_j}\|_2)+\frac{\rho}{2}\|x_{\mathcal{G}_j}\|_2^2)$. *The corresponding proximal step is*

$$\min_x \frac{1}{2}\|x-z\|_2^2 + \sum_{j=1}^{K}\lambda_j\left(\kappa\left(\|x_{\mathcal{G}_j}\|_2\right) + \frac{\rho}{2}\|x_{\mathcal{G}_j}\|_2^2\right). \tag{29}$$

However, inexpensive closed-form solution on the proximal step is only known for tree-structured group lasso regularizers of the form $\sum_{j=1}^{K}\lambda_j\|x_{\mathcal{G}_j}\|_2$ *(Jenatton et al., 2011; Liu and Ye, 2010). Thus, (29) has to be iteratively solved (e.g., using ADMM), and is slow.*

In Section 4.1.3, by adding quadratic terms to the nonconvex TV regularizer, it becomes $\sum_{i=1}^{m-1}\sum_{j=1}^{m}(\kappa(|[D_vX]_{ij}|)+\frac{\rho}{2}[D_vX]_{ij}^2)+\sum_{i=1}^{n}\sum_{j=1}^{n-1}(\kappa(|[XD_h]_{ij}|)+\frac{\rho}{2}[XD_h]_{ij}^2)$, *where* $\rho>0$ *is a constant. The corresponding proximal step is*

$$\min_X \frac{1}{2}\|X-Z\|_F^2+\mu\sum_{i=1}^{m-1}\sum_{j=1}^{m}\Big(\kappa\left(|[D_vX]_{ij}|\right)+\frac{\rho}{2}[D_vX]_{ij}^2\Big)+\mu\sum_{i=1}^{n}\sum_{j=1}^{n-1}\Big(\kappa\left(|[XD_h]_{ij}|\right)+\frac{\rho}{2}[XD_h]_{ij}^2\Big),$$

which is difficult to solve. Morevoer, unlike the proposed convexification scheme, the dual of the above is difficult to derive.

4.2 Frank-Wolfe Algorithm

In this section, we use the Frank-Wolfe algorithm to learn a low-rank matrix $X \in \mathbb{R}^{m\times n}$ for matrix completion (Section 2.3). The nuclear norm regularizer in (8) may over-penalize top singular values. Recently, there is growing interest to replace this with nonconvex regularizers (Lu et al., 2014, 2015; Yao et al., 2015). Hence, instead of (8), we consider

$$\min_X f(X) + \mu\sum_{i=1}^{m}\kappa(\sigma_i(X)). \tag{30}$$

When κ is the identity function, (30) reduces to (8). Note that the FW algorithm cannot be directly used on (8), as its linear subproblem in (10) then becomes $\min_{S:\sum_{i=1}^{m}\kappa(\sigma_i(S))\le 1}\langle S, \nabla f(X_t)\rangle$, which is difficult to solve.

Using Proposition 5, problem (30) is transformed into

$$\min_X \bar{f}(X) + \bar{\mu}\|X\|_*, \tag{31}$$

where

$$\bar{f}(X) = f(X) + \bar{g}(X),\ \ \bar{g}(X) = \mu\sum_{i=1}^{m}(\kappa(\sigma_i(X)) - \kappa_0\sigma_i(X)), \tag{32}$$

and $\bar{\mu} = \mu\kappa_0$. Although this only involves the standard nuclear norm regularizer, Algorithm 1 still cannot be used as $\bar{f}$ in (32) is no longer convex. A FW variant allowing nonconvex $\bar{f}$ is proposed in (Bredies et al., 2009). However, condition 1 in (Bredies et al., 2009) requires g to satisfy $\lim_{\|X\|_F\to\infty}\frac{g(X)}{\|X\|_F} = \infty$. This condition does not hold with $g(X) = \|X\|_*$ in (31), as

$$\frac{\|X\|_*}{\|X\|_F} = \sqrt{\frac{(\sum_{i=1}^{m}\sigma_i)^2}{\sum_{i=1}^{m}\sigma_i^2}} \le \sqrt{\frac{m\sum_{i=1}^{m}\sigma_i^2}{\sum_{i=1}^{m}\sigma_i^2}} = \sqrt{m} < \infty.$$

In the following, we propose a nonconvex FW variant (Algorithm 4) for the transformed problem (31). It is similar to the original FW Algorithm 1, but with three important modifications. First, $\bar{g}(X)$ in (32) depends on the singular values of X, which cannot be directly obtained from the $UV^\top$ factorization in (11). Instead, we use the low-rank factorization

$$X = UBV^\top, \tag{33}$$

where $U \in \mathbb{R}^{m\times k}$, $V \in \mathbb{R}^{n\times k}$ are orthogonal and $B \in \mathcal{S}_+^{k\times k}$ is positive semidefinite.

Algorithm 4 Frank-Wolfe algorithm for solving the nonconvex problem (31).

1: $U_1 = [\,]$, $B_1 = [\,]$ and $V_1 = [\,]$;
2: **for** $t = 1\dots T$ **do**
3: $[u_t, s_t, v_t] = \text{rank1SVD}(\nabla\bar{f}(X_t))$;
4: obtain α_t and β_t from (36);
5: $[\bar{U}_t, \bar{B}_t, \bar{V}_t] = \text{warmstart}(U_t, u_t, V_t, v_t, B_t, \alpha_t, \beta_t)$;
6: obtain $[U_{t+1}, B_{t+1}, V_{t+1}]$ from (37), using $\bar{U}_t$, $\bar{B}_t$ and $\bar{V}_t$ for warm-start;
// $X_{t+1} = U_{t+1}B_{t+1}V_{t+1}^\top$
7: **end for**
8: **return** U_{T+1}, B_{T+1} and V_{T+1}.

The second problem is that line search in Algorithm 1 is inefficient in general when operated on a nonconvex $\bar{f}$. Specifically, step 4 in Algorithm 1 then becomes

$$[\alpha_t, \beta_t] = \arg\min_{\alpha\ge 0,\beta\ge 0}\bar{f}(\alpha X_t + \beta u_t v_t^\top) + \bar{\mu}(\alpha\|X_t\|_* + \beta). \tag{34}$$

To solve (34), we have to compute $\frac{\partial\bar{f}(S)}{\partial\alpha}$ and $\frac{\partial\bar{f}(S)}{\partial\beta}$, where $S = \alpha X_t + \beta u_t v_t^\top$. As shown in Proposition 10, this requires the SVD of S and can be expensive.

Proposition 10 *Let the SVD of* S *be* $U_S\mathit{Diag}([\sigma_1(S),\dots,\sigma_m(S)])V_S^\top$. *Then*

$$\frac{\partial\bar{f}(S)}{\partial\alpha} = \alpha\langle X_t, \nabla\bar{f}(S)\rangle, \quad \textit{and} \quad \frac{\partial\bar{f}(S)}{\partial\beta} = \beta u_t^\top\nabla\bar{f}(S)v_t,$$

where $\nabla\bar{f}(S) = \nabla f(S) + \mu U_S\mathit{Diag}(w)V_S^\top$, *and* $w = [\kappa'(\sigma_i(S)) - \kappa_0] \in \mathbb{R}^m$.

Alternatively, as S is a rank-one update of X_t, one can perform incremental update on SVD, which takes $O((m+n)t^2)$ time (Golub and Van Loan, 2012). However, every time α, β are changed, this incremental SVD has to be recomputed, and is thus inefficient.

To alleviate this problem, we approximate $\bar{f}(S)$ by an upper bound as

$$\begin{aligned}\bar{f}(S) &= \bar{f}(X_t + (\alpha-1)X_t + \beta u_t v_t^\top) \\ &\leq \bar{f}(X_t) + \langle (\alpha-1)X_t + \beta u_t v_t^\top, \nabla \bar{f}(X_t)\rangle + \frac{\bar{L}}{2}\|(\alpha-1)X_t + \beta u_t v_t^\top\|_F^2. \end{aligned} \quad (35)$$

As (u_t, v_t) is obtained from the rank-1 SVD of $\nabla \bar{f}(X_t)$, we have $\|u_t v_t^\top\|_F = 1$ and $u_t^\top \nabla \bar{f}(X_t) v_t = s_t$. Moreover, $X_t = U_t B_t V_t^\top$, and so $\|X_t\|_F = \|B_t\|_F$ and $\|X_t\|_* = \mathrm{Tr}\,(B_t)$. Substituting these and the upper bound (35) into (34), we obtain a simple quadratic program:

$$\begin{aligned}\min_{\alpha\geq 0, \beta\geq 0} \quad & \frac{(\alpha-1)^2\bar{L}}{2}\|B_t\|_F^2 + (\alpha-1)\beta\bar{L}(u_t^\top U_t)B_t(V_t^\top v_t) + \frac{\beta^2\bar{L}}{2} + \beta s_t \\ & + \alpha\langle B_t, U_t^\top \nabla \bar{f}(X_t) V_t\rangle + \bar{\mu}(\alpha\|B_t\|_* + \beta). \end{aligned} \quad (36)$$

Note that the objective in (36) is convex, as the RHS in (35) is convex and the last term from (34) is affine. Using the following Corollary 11, $\langle B_t, U_t^\top \nabla \bar{f}(X_t) V_t\rangle$ in (36) can be obtained as

$$\langle B_t, U_t^\top \nabla \bar{f}(X_t) V_t\rangle = \langle B_t, U_t^\top \nabla f(X_t) V_t\rangle + \bar{\mu}\sum_{i=1}^{t}\sigma_i(B_t)(\kappa'(\sigma_i(B_t)) - \kappa_0).$$

Corollary 11 *For X in (33), let the SVD of B be $U_B Diag([\sigma_1(B), \ldots, \sigma_k(B)])V_B^\top$. Then, $\nabla \bar{f}(X) = \nabla f(X) + \bar{\mu}(UU_B)Diag(w)(VV_B)^\top$, where $w = [\kappa'(\sigma_i(B)) - \kappa_0] \in \mathbb{R}^k$.*

Instead of requiring SVD on X_t, it only requires SVD on B_t (which is of size $t \times t$ at the tth iteration of Algorithm 4). As the target matrix is supposed to be low-rank, $t \ll m$. Hence, all the coefficients in (36) can be obtained in $O((m+n)t^2 + \|\Omega\|_1 t)$ time. Besides, (36) is a quadratic program with only two variables, and thus can be very efficiently solved.

The third modification is that with $\bar{f}$ instead of f, (12) can no longer be used for local optimization, as $\bar{g}(X)$ in (32) depends on the singular values of X. On the other hand, with the decomposition of X in (33) and Proposition 12 below, (31) can be rewritten as

$$\begin{aligned}\min_{U,B,V} \quad & f(UBV^\top) + \bar{g}(B) + \bar{\mu}\,\mathrm{Tr}\,(B) && (37)\\ \text{s.t.} \quad & U^\top U = I, V^\top V = I, B \in \mathcal{S}_+. && (38)\end{aligned}$$

This can be efficiently solved using matrix optimization techniques on the Grassmann manifold (Ngo and Saad, 2012).

Proposition 12 *For orthogonal matrices U and V, $\bar{g}(UBV^\top) = \bar{g}(B)$.*

In Algorithm 4, step 5 is used to warm-start (37), and the procedure is shown in Algorithm 5. It expresses $X_t = \alpha_t U_{t-1} B_{t-1} V_{t-1}^\top + \beta_t u_t v_t^\top$ obtained in step 4 to the form $U_t B_t V_t^\top$ so that the orthogonal constraints on U_t, V_t in (38) are satisfied.

Existing analysis for the FW algorithm cannot be used on this nonconvex problem. The following Theorem shows convergence of Algorithm 4 to a critical point of (8).

Theorem 13 *If* (8) *has a rank-r critical point, then Algorithm 4 converges to a critical point of* (8) *after r iterations.*

Algorithm 5 warmstart$(U_t, u_t, V_t, v_t, B_t, \alpha_t, \beta_t)$.

1: $[\bar{U}_t, R_{\bar{U}_t}] = \mathrm{QR}([U_t, u_t])$; // QR denotes the QR factorization
2: $[\bar{V}_t, R_{\bar{V}_t}] = \mathrm{QR}([V_t, v_t])$;
3: $\bar{B}_t = R_{\bar{U}_t}\begin{bmatrix}\alpha_t B_t & 0 \\ 0 & \beta_t\end{bmatrix} R_{\bar{V}_t}^\top$;
4: **return** $\bar{U}_t$, $\bar{B}_t$ and $\bar{V}_t$;

As in Remark 8, an alternative convexification approach is to decompose the regularizer in (30) as $\breve{\varsigma}(X) + \hat{\varsigma}(X)$, where $\hat{\varsigma}(X) = -\frac{\rho}{2}\sum_{i=1}^m \sigma_i^2(X)$ and $\breve{\varsigma}(X) = \sum_{i=1}^m \kappa(\sigma_i(X)) + \frac{\rho}{2}\sigma_i^2(X)$. The corresponding linear subproblem in (5) becomes $\min_{S:\breve{\varsigma}(S)\leq 1}\langle S, \nabla f(X_t)\rangle$, which is difficult to solve. On the other hand, with the proposed procedure, the subproblem associated with the transformed problem (31) can be easily solved via rank-1 SVD (Jaggi, 2013).

4.3 Alternating Direction Method of Multipliers (ADMM)

In this section, we consider using ADMM on the consensus optimization problem (17). When all the f_i's and g are convex, ADMM has a convergence rate of $O(1/T)$ (He and Yuan, 2012). Recently, ADMM has been extended to problems where g is convex but f_i's are nonconvex (Hong et al., 2016). However, when g is nonconvex, such as when a nonconvex regularizer is used in regularized risk minimization, the convergence of ADMM is still an open reseach problem.

Using the proposed transformation, we can decompose a nonconvex g as $\bar{g} + \breve{g}$, where $\bar{g}$ is concave and Lipschitz-smooth, while $\breve{g}$ is convex but possibly nonsmooth. Problem (17) can then be rewritten as

$$\min_{y, x^1, \ldots, x^M} \sum_{i=1}^M \bar{f}_i(x^i) + \breve{g}(y) \quad : \quad x^1 = \cdots = x^M = y, \quad (39)$$

where $\bar{f}_i(x) = f_i(x) + \frac{1}{M}\bar{g}(x)$. Let p^i be the dual variable for the constraint $x^i = y$. The augmented Lagrangian for (39) is

$$\mathcal{L}\left(y, x^1, \ldots, x^M, p^1, \ldots, p^M\right) = \breve{g}(y) + \sum_{i=1}^M \bar{f}_i(x^i) + (p^i)^\top(x^i - y) + \frac{\tau}{2}\|x^i - y\|_2^2.$$

Using (14) and (15), we have the following update equations at iteration t:

$$\begin{aligned} x_{t+1}^i &= \arg\min_{x^i} \bar{f}_i(x^i) + (p_t^i)^\top(x^i - y_t) + \frac{\tau}{2}\|x^i - y_t\|_2^2, \quad i = 1, \ldots, M, \\ y_{t+1} &= \arg\min_y \frac{1}{2}\left\|y - \sum_{i=1}^M\left(x_t^i + \frac{1}{\tau}p_t^i\right)\right\|_2^2 + \frac{1}{\tau}\breve{g}(y) = \mathrm{prox}_{\frac{1}{\tau}\breve{g}}\left(\sum_{i=1}^M x_t^i + \frac{1}{\tau}p_t^i\right). \end{aligned} \quad (40)$$

As in previous sections, the proximal step in (40), which is associated with the convex $\breve{g}$, is usually easier to compute than the proximal step associated with the original nonconvex g. Moreover, since $\breve{g}$ is convex, convergence results in Theorem 2.4 of (Hong et al., 2016) can now be applied. Specifically, the sequence $\{y_t, \{x_t^i\}\}$ generated by the ADMM procedure is bounded and all its limit points are critical points of (39).

As in Remark 8, the alternative convexification approach based on adding a quadratic regularizer (Proposition 9) does not help. For example, when g is the nonconvex tree-structured lasso regularizer, after adding a quadratic regularizer $\frac{\rho}{2}\|y_{\mathcal{G}_j}\|_2^2$, the y_t update in (19) becomes

$$y_{t+1} = \arg\min_y \frac{1}{2}\left\| y - \sum_{i=1}^{M}\left(x_t^i + \frac{1}{\tau}p_t^i\right)\right\|_2^2 + \frac{1}{\tau}\sum_{j=1}^{K}\lambda_j\left(\kappa\left(\|y_{\mathcal{G}_j}\|_2\right) + \frac{\rho}{2}\|y_{\mathcal{G}_j}\|_2^2\right).$$

which is still difficult to solve.

4.4 Stochastic Variance Reduced Gradient

Variance reduction methods have been commonly used to speed up the often slow convergence of SGD. Examples include the stochastic variance reduced gradient (SVRG) and its proximal extension Prox-SVRG (Xiao and Zhang, 2014). They can be used for the following optimization problem

$$\min_x \sum_{i=1}^{N}\ell(y_i, a_i^\top x) + g(x), \tag{41}$$

where $\{(a_1, y_1), \ldots, (a_N, y_N)\}$ are the training samples, ℓ is a smooth convex loss function, and g is a convex regularizer. Recently, Prox-SVRG is also extended for nonconvex objectives. Reddi et al. (2016a) and Zhu and Hazan (2016) considered smooth nonconvex ℓ but without g. This is further extended to the case of smooth ℓ and convex nonsmooth g in (Reddi et al., 2016b). However, convergence is still unknown for the more general case where the regularizer g is also nonconvex.

Using the proposed transformation, (41) can be rewritten as

$$\min_x \sum_{i=1}^{N}\left(\ell(y_i, a_i^\top x) + \frac{1}{N}\bar{g}(x)\right) + \breve{g}(x),$$

where $\ell + \frac{1}{N}\bar{g}$ is smooth and $\breve{g}$ is convex. The convergence results of Theorem 1 of (Reddi et al., 2016b) can now be applied, which shows that SVRG generates a bounded sequence and all its limit points are critical points of (41).

As in Remark 8, adding a quadratic term to convexify the nonconvex regularizer does not make the corresponding proximal step easier, and so does not help.

4.5 With OWL-QN

In this section, we consider OWL-QN (Andrew and Gao, 2007) and its variant mOWL-QN (Gong and Ye, 2015b), which are efficient algorithms for the ℓ_1-regularization problem

$$\min_x f(x) + \mu\|x\|_1. \tag{42}$$

Recently, Gong and Ye (2015a) proposed a nonconvex generalization for (42), in which the standard ℓ_1 regularizer is replaced by the nonconvex $g(x) = \mu\sum_{i=1}^{d}\kappa(|x_i|)$:

$$\min_x f(x) + \mu\sum_{i=1}^{d}\kappa(|x_i|). \tag{43}$$

Gong and Ye (2015a) proposed a sophisticated algorithm (HONOR) which involves a combination of quasi-Newton and gradient descent steps. Though the algorithm is similar to OWL-QN and mOWL-QN, the convergence analysis in (Gong and Ye, 2015b) cannot be directly applied as the regularizer is nonconvex. Instead, a non-trivial extension was developed in (Gong and Ye, 2015a).

Here, by convexifying the nonconvex regularizer, (43) can be rewritten as

$$\min_x \bar{f}(x) + \mu\kappa_0\|x\|_1, \tag{44}$$

where $\bar{f}(x) = f(x) + \bar{g}(x)$, and $\bar{g}(x) = \mu\sum_{i=1}^{d}(\kappa(|x_i|) - \kappa_0|x_i|)$. It is easy to see that the analysis in (Gong and Ye, 2015b) can be extended to handle smooth but nonconvex $\bar{f}$. As a result, Theorem 1 in (Gong and Ye, 2015b) can still be applied. Thus, mOWL-QN is guaranteed to generate a bounded sequence and its limit points are critical points of (42).

As in previous subsections, adding a quadratic term to convexify the nonconvex regularizer does not help. The mOWL-QN can only work with the ℓ_1-regularizer, but not with the modified regularizer $\breve{\varsigma}(x) = \frac{\rho\mu}{2}\|x\|_2^2 + \mu\sum_{i=1}^{d}\kappa(|x_i|)$.

Problem (43) can be solved by either (i) directly using HONOR, or (ii) using mOWL-QN on the transformed problem (44). We believe that the latter approach is computationally more efficient. In (43), the Hessian depends on both terms in the objective, as the second-order derivative of κ is not zero in general. However, HONOR constructs the approximate Hessian using only information from f, and thus ignores the curvature information due to $\sum_{i=1}^{d}\kappa(|x_i|)$. On the other hand, the Hessian in (44) depends only on $\bar{f}$, as the Hessian due to $\|x\|_1$ is zero (Andrew and Gao, 2007), and mOWL-QN now extracts Hessian from $\bar{f}$. Hence, optimizing (44) with mOWL-QN is potentially faster, as all the second-order information is utilized. This will be verified empirically in Section 5.4.

4.6 Nonsmooth and Nonconvex Loss

In many applications, besides having nonconvex regularizers, the loss function may also be nonconvex and nonsmooth. In this section, we consider using the nonconvex functions in Figure 1 as the loss function. Thus, neither f nor g in (1) is convex, smooth. The optimization problem becomes even harder, and many existing algorithms cannot be used. In particular, the proximal algorithm requires f in (1) to be smooth (possibly nonconvex) (Gong et al., 2013; Li and Lin, 2015; Bot et al., 2016). The FW algorithm requires f in (4) to be smooth and convex (Jaggi, 2013). For the ADMM, it allows f in the consensus problem to be smooth, but g has to be convex (Hong et al., 2016). For problems of the form $\min_{x,z} f(y) + g(y) : y = Ax$, ADMM requires A to have full row-rank (Li and Pong, 2015). As will be seen, it is not satisfied for problems considered in this section. CCCP (Yuille and Rangarajan, 2002) and smoothing (Chen, 2012) are more general and can still be used, but are usually very slow.

In this section, we consider two application examples, and show how they can be efficiently solved with the proposed transformation.

4.6.1 TOTAL VARIATION IMAGE DENOISING

Using the (convex) ℓ_1 loss and (convex) TV regularizer introduced in Section 4.1.3, consider the following optimization problem:

$$\min_X \|Y - X\|_1 + \mu\mathrm{TV}(X), \tag{45}$$

where $Y \in \mathbb{R}^{m\times n}$ is a given corrupted image, and X is the target image to be recovered. The use of nonconvex loss and regularizer often produce better performance (Yan, 2013). Thus, we consider the following nonconvex extension:

$$\min_X \sum_{i=1}^{m}\sum_{j=1}^{n} \kappa\left(\left|[Y-X]_{ij}\right|\right) + \mu \sum_{i=1}^{m-1}\sum_{j=1}^{m} \kappa\left(\left|[D_vX]_{ij}\right|\right) + \mu \sum_{i=1}^{n}\sum_{j=1}^{n-1} \kappa\left(\left|[XD_h]_{ij}\right|\right), \quad (46)$$

where both the loss and regularizer are nonconvex and nonsmooth. As discussed above, this can be solved by CCCP and smoothing. However, as will be experimentally demonstrated in Section 5.5, the convergence is slow.

Using the proposed transformation on both the loss and regularizer, problem (46) can be transformed to the following problem:

$$\min_X \bar{f}(X) + \kappa_0\|X-Y\|_1 + \kappa_0\mu\text{TV}(X), \quad (47)$$

where

$$\begin{aligned}\bar{f}(X) =& \sum_{i=1}^{m}\sum_{j=1}^{n} \kappa\left(\left|[Y-X]_{ij}\right|\right) - \kappa_0\|Y-X\|_1 \\ &+ \mu\left[\sum_{i=1}^{m-1}\sum_{j=1}^{m} \kappa\left(\left|[D_vX]_{ij}\right|\right) - \kappa_0\|D_vX\|_1 + \sum_{i=1}^{n}\sum_{j=1}^{n-1} \kappa\left(\left|[XD_h]_{ij}\right|\right) - \kappa_0\|XD_h\|_1\right]\end{aligned}$$

is smooth and nonconvex. As (47) is not a consensus problem, the method in (Hong et al., 2016) cannot be used. To use the ADMM algorithm in (Li and Pong, 2015), extra variables and constraints $Z_v = D_vX$ and $Z_h = XD_h$ have to be imposed. However, the full row-rank condition in (Li and Pong, 2015) does not hold.

In this section, we consider the proximal algorithm. Given some Z, the proximal step in (47) is

$$\arg\min_X \frac{1}{2}\|X-Z\|_F^2 + \frac{1}{\tau}\left(\|X-Y\|_1 + \mu\text{TV}(X)\right), \quad (48)$$

where τ is the stepsize. Though this has no closed-form solution, $\|X-Y\|_1 + \mu\text{TV}(X)$ in (48) is convex and one can monitor inexactness of the proximal step via the duality gap. Thus, we can use the proposed inexact nmAPG algorithm in Algorithm 3 for (47). It can be shown that the dual of (48) is

$$\begin{aligned}\min_{W,P,Q}\quad & \frac{1}{2\tau}\|W + \mu D_v^\top P + \mu Q D_h^\top\|_F^2 - \langle Z, W\rangle - \mu\langle D_vZ, P\rangle - \mu\langle ZD_h, Q\rangle + \langle Y, W\rangle \\ \text{s.t.}\quad & \|W\|_\infty \le 1, \|P\|_\infty \le 1 \text{ and } \|Q\|_\infty \le 1,\end{aligned} \quad (49)$$

and the primal variable can be recovered as $X = Z - \frac{1}{\tau}(W + \mu D_v^\top P + \mu Q D_h^\top)$. By substituting the obtained X into (48) and $\{W, P, Q\}$ into (49), the duality gap can be computed in $O(mn)$ time. As (49) is a smooth and convex problem, both accelerated gradient descent (Nesterov, 2013) and L-BFGS (Nocedal and Wright, 2006) can be applied. Algorithm 3 is then guaranteed to converge to a critical point of (46) (Theorem 6 and Proposition 7).

Note that it is more advantageous to transform both the loss and regularizer in (47). If only the regularizer in (46) is transformed, we obtain

$$\bar{f}_{\text{TV}}(X) + \sum_{i=1}^{m}\sum_{j=1}^{n} \kappa\left(\left|[Y-X]_{ij}\right|\right) + \kappa_0\mu\text{TV}(X), \quad (50)$$

where

$$\bar{f}_{\text{TV}}(X) = \mu\left[\sum_{i=1}^{m-1}\sum_{j=1}^{m} \kappa\left(\left|[D_vX]_{ij}\right|\right) - \kappa_0\|D_vX\|_1 + \sum_{i=1}^{n}\sum_{j=1}^{n-1} \kappa\left(\left|[XD_h]_{ij}\right|\right) - \kappa_0\|XD_h\|_1\right]$$

is nonconvex. The corresponding proximal step for (50) is

$$\arg\min_X \frac{1}{2}\|X-Z\|_F^2 + \frac{1}{\tau}\left(\sum_{i=1}^{m}\sum_{j=1}^{n} \kappa\left(\left|[Y-X]_{ij}\right|\right) + \kappa_0\mu\text{TV}(X)\right). \quad (51)$$

While the proximal steps in both (48) and (51) have no closed-form solution, working with (48) is more efficient. As (48) is convex, its dual can be efficiently solved with methods such as accelerated gradient descent and L-BFGS. In contrast, (51) is nonconvex, its duality gap is nonzero, and so can only be solved in the primal with slower methods like CCCP and smoothing. Besides, one can only use the more expensive nmAPG (Algorithm 2) but not the proposed inexact proximal algorithm.

One may also consider simultaneously transforming both the loss and regularizer using the decomposition discussed in Remark 8. However, it is not helpful here. By adding and subtracting a quadratic term, the objective in (46) can be decomposed as $\breve{\varsigma}(X) + \hat{\varsigma}(X)$, where

$$\begin{aligned}\hat{\varsigma}(X) =& -\sum_{i=1}^{m}\sum_{j=1}^{n}\frac{\rho}{2}[Y-X]_{ij}^2 - \frac{\rho\mu}{2}\sum_{i=1}^{m-1}\sum_{j=1}^{m}[D_vX]_{ij}^2 - \frac{\rho\mu}{2}\sum_{i=1}^{n}\sum_{j=1}^{n-1}[XD_h]_{ij}^2, \\ \breve{\varsigma}(X) =& \sum_{i=1}^{m}\sum_{j=1}^{n}\left(\kappa(|\,[Y-X]_{ij}\,|) + \frac{\rho}{2}[Y-X]_{ij}^2\right) \\ &+ \mu\sum_{i=1}^{m-1}\sum_{j=1}^{m}\left(\kappa(|[D_vX]_{ij}|) + \frac{\rho}{2}[D_vX]_{ij}^2\right) + \mu\sum_{i=1}^{n}\sum_{j=1}^{n-1}\left(\kappa(|[XD_h]_{ij}|) + \frac{\rho}{2}[XD_h]_{ij}^2\right).\end{aligned}$$

We need to solve the proximal step associated with $\breve{\varsigma}(X)$, which is difficult.

4.6.2 ROBUST SPARSE CODING

The second application is robust sparse coding, which has been popularly used in face recognition (Yang et al., 2011), image analysis (Lu et al., 2013) and background modeling (Zhao et al., 2011). Given an observed signal $y \in \mathbb{R}^m$, the goal is to seek a robust sparse representation $x \in \mathbb{R}^d$ of y based on the dictionary $D \in \mathbb{R}^{m\times d}$ (which is assumed to be fixed here). Mathematically, it is formulated as the following optimization problem:

$$\min_x \|y - Dx\|_1 + \mu\|x\|_1.$$

Its nonconvex extension is:

$$\min_x \sum_{j=1}^m \kappa(|[y-Dx]_j|) + \mu \sum_{i=1}^d \kappa(|x_i|). \tag{52}$$

Using the proposed transformation, problem (52) becomes

$$\min_x \bar{f}(x) + \kappa_0 \|y - Dx\|_1 + \mu\kappa_0 \|x\|_1, \tag{53}$$

where

$$\bar{f}(x) = \mu \sum_{j=1}^d \kappa(|x_j|) - \kappa_0 \mu \|x\|_1 + \sum_{j=1}^m \kappa(|[y-Dx]_j|) - \kappa_0 \|y - Dx\|_1$$

is smooth and nonconvex. Again, we use the inexact nmAPG algorithm in Algorithm 3. The proximal step for (53) is

$$\arg\min_x \frac{1}{2}\|x - z\|_2^2 + \frac{1}{\tau}(\|y - Dx\|_1 + \mu\|x\|_1), \tag{54}$$

where τ is the stepsize and z is given. As in Section 4.6.1, $\|y - Dx\|_1 + \mu\|x\|_1$ in (54) is convex, and one can monitor inexactness of the proximal step by the duality gap. The dual of (54) is

$$\min_{p,q} \frac{1}{2\tau}\|D^\top p + \mu q\|_2^2 - p^\top Dz - \mu q^\top z \ : \ \|p\|_\infty \le 1, \|q\|_\infty \le 1. \tag{55}$$

As in (49), this can be solved with L-BFGS or accelerated gradient descent. The primal variable can be recovered as $x = z - \frac{1}{\tau}(D^\top p + \mu q)$, and the duality gap can be checked in $O(md)$ time.

If only the regularizer is transformed, we obtain

$$\min_x \sum_{j=1}^m \kappa(|[y-Dx]_j|) + \bar{f}_{\text{RSC}}(x) + \kappa_0\mu\|x\|_1, \tag{56}$$

where $\bar{f}_{\text{RSC}}(x) = \mu\sum_{j=1}^d \kappa(|x_j|) - \kappa_0\mu\|x\|_1$. The corresponding proximal step is

$$\arg\min_x \frac{1}{2}\|x - z\|_2^2 + \sum_{j=1}^m \kappa(|[y-Dx]_j|) + \kappa_0\mu\|x\|_1, \tag{57}$$

which still involve the nonconvex function κ. As in Section 4.6.1, (55) is easier to solve than (57).

As in previous sections, adding a quadratic term to convexify the loss and regularizer is not helpful. The objective in (52) will then be decomposed as $\breve{\varsigma}(X) + \hat{\varsigma}(X)$, where

$$\hat{\varsigma}(X) = -\frac{\rho}{2}\sum_{j=1}^m [y-Dx]_j^2 - \frac{\rho\mu}{2}\sum_{i=1}^d x_i^2,$$

$$\breve{\varsigma}(X) = \sum_{j=1}^m \left(\kappa(|[y-Dx]_j|) + \frac{\rho}{2}[y-Dx]_j^2\right) + \mu\sum_{i=1}^d \left(\kappa(|x_i|) + \frac{\rho}{2}x_i^2\right),$$

and the proximal step associated with $\breve{\varsigma}(X)$ is again difficult to solve.

5. Experiments

In this section, we perform experiments on using the proposed procedure with (i) proximal algorithms (Sections 5.1 and 5.2); (ii) Frank-Wolfe algorithm (Section 5.3); (iii) comparision with HONOR (Section 5.4) and (vi) image denoising (Section 5.5). Experiments are performed on a PC with Intel i7 CPU and 32GB memory. All algorithms are implemented in Matlab.

5.1 Nonconvex Sparse Group Lasso

In this section, we perform experiments on the nonconvex sparse group lasso model in Section 4.1.1. For simplicity, assume that $\mu_1 = \cdots = \mu_K = \mu$. Using the square loss, (25) becomes

$$\min_x \frac{1}{2}\|y - A^\top x\|_2^2 + \lambda\sum_{i=1}^d \kappa\left(|x_i|\right) + \mu\sum_{j=1}^K \kappa\left(\|x_{\mathcal{G}_j}\|_2\right), \tag{58}$$

where $A = [a_1, \ldots, a_N]$. In this experiment, we use the LSP regularizer in Table 1 (with $\theta = 0.5$) as $\kappa(\cdot)$. The synthetic data set is generated as follows. We set $d = 200{,}000$. The ground-truth parameter $\bar{x} \in \mathbb{R}^d$ is divided into 200 non-overlapping groups: $\{1, \ldots, 1000\}$, $\{1001, \ldots, 2000\}$, $\ldots$, $\{199001, \ldots, 200000\}$ (Figure 3). We randomly set 87.5% of the groups to zero. In each nonzero group, we randomly set 50% of its features to zero, and generate the nonzero features from the standard normal distribution $\mathcal{N}(0, 1)$. The whole data set has $400{,}000$ samples, and entries of the input $A \in \mathbb{R}^{200{,}000\times 400{,}000}$ is a sparse matrix with 0.01% nonzero elements which are generated from $\mathcal{N}(0, 1)$. The ground-truth output is $\bar{y} = A^\top\bar{x}$. This is then corrupted by random Gaussian noise ϵ in $\mathcal{N}(0, 0.05)$ to produce $y = \bar{y} + \epsilon$.

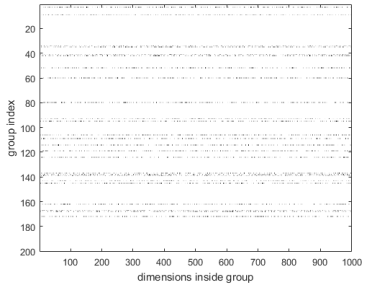


Figure 3: An example ground-truth parameter $\bar{x} \in \mathbb{R}^{200000}$. It is reshaped as a 200×1000 matrix, with each row representing a group.

The proposed algorithm will be called N2C (Nonconvex-to-Convex). The proximal step of the convexified regularizer $\breve{g}(x) = \kappa_0(\lambda\|x\|_1 + \sum_{j=1}^K \mu_j\|x_{\mathcal{G}_j}\|_2)$ is obtained using the algorithm in (Yuan et al., 2011). The nmAPG algorithm (Algorithm 2) in (Li and Lin, 2015) is used for optimization. This will be compared with the following state-of-the-art algorithms:

1. SCP: Sequential convex programming (Lu, 2012), in which the LSP regularizer is decomposed as in (28).

2. GIST (Gong et al., 2013): Since the nonconvex regularizer is not separable, the associated proximal operator has no closed-form solution. Instead, we use SCP (with warm-start) to solve it numerically.

3. GD-PAN (Zhong and Kwok, 2014): It performs gradient descent with the proximal average (Bauschke et al., 2008) of the nonconvex regularizers. Closed-form solutions for the proximal operators of each individual regularizer are obtained separately, and then averaged.

4. nmAPG with the original nonconvex regularizer: As in GIST, the proximal step is solved numerically by SCP.

5. As a baseline, we also compare with the FISTA (Beck, 2009) algorithm, which solves the convex sparse group lasso model (with κ removed from (58)).

We do not compare with the concave-convex procedure (Yuille and Rangarajan, 2002), which has been shown to be slow (Gong et al., 2013; Zhong and Kwok, 2014).

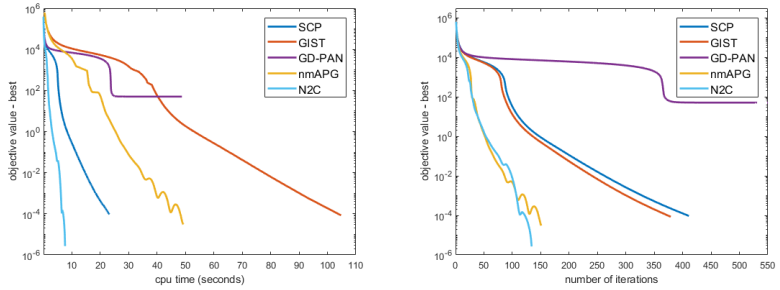
We use 50% of the data for training, another 25% as validation set to tune λ, μ in (58), and the rest for testing. The stepsize is fixed at $\tau = \sigma_1(A^\top A)$. For performance evaluation, we use the (i) testing root-mean-squared error (RMSE) on the predictions; (ii) absolute error between the obtained parameter $\hat{x}$ and the corresponding ground-truth $\bar{x}$: ABS $= \|\hat{x} - \bar{x}\|_1/d$; and (iii) CPU time. To reduce statistical variability, the experimental results are averaged over 5 repetitions.

Results are shown in Table 3. As can be seen, all the nonconvex models obtain better errors (RMSE and ABS) than the convex FISTA. As for the training speed, N2C is the fastest. SCP, GIST, nmAPG and N2C all solve the original problem (1), and have the same recovery performance. GD-PAN solves an approximate problem in each iteration, and its error is slightly worse than the other nonconvex algorithms on this data set.

	non-accelerated			accelerated		convex
	SCP	GIST	GD-PAN	nmAPG	N2C	FISTA
RMSE	**48.9±0.1**	**48.9±0.1**	49.2±0.1	**48.9±0.1**	**48.9±0.1**	64.0±0.1
ABS	**3.1±0.1**	**3.1±0.1**	5.5±0.1	**3.1±0.1**	**3.1±0.1**	13.0±0.1
CPU time(sec)	23.2±7.5	105.6±24.6	33.7±11.7	43.0±3.5	**7.9±1.2**	5.1±0.8

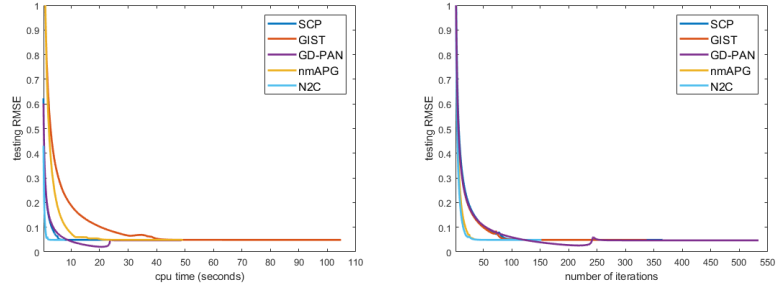
Table 3: Results on nonconvex sparse group lasso. RMSE and ABS are scaled by 10^{-3}, and the CPU time is in seconds. The best and comparable results (according to the paired t-test with 95% confidence) are highlighted.

Figure 4(a) shows convergence of the objective with time for a typical run. SCP, GIST, nmAPG and N2C all converge towards the same objective value. GD-PAN can only approximate the original problem. Thus, it converges to an objective value which is larger than the others. Figure 4(b) shows the convergence with number of iterations. As can be seen, N2C and nmAPG, which are based on the same state-of-the-art proximal algorithm (Algorithm 2), require nearly the same number of iterations for convergence. However, as N2C has an inexpensive closed-form solution for its proximal step, it is much faster when measured in terms of time (Figure 4(a)). Figure 5 shows convergence of the testing RMSE. Its behaviour is similar to those observed in Figure 4. Overall, N2C, which uses acceleration and inexpensive proximal step, is the fastest.



(a) Objective vs CPU time (seconds). (b) Objective vs number of iterations.

Figure 4: Convergence of objective on nonconvex sparse group lasso. In the ordinate, "best" refers to the smallest objective value obtained among the various methods. Note that FISTA is not shown as its (convex) objective is different from the others.



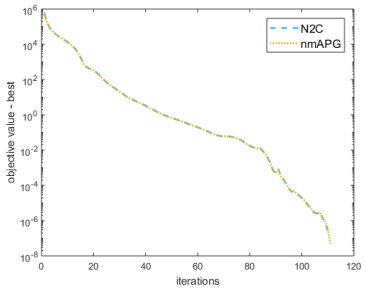
(a) Testing RMSE vs CPU time (seconds). (b) Testing RMSE vs number of iterations.

Figure 5: Convergence of the testing RMSE on nonconvex sparse group lasso.

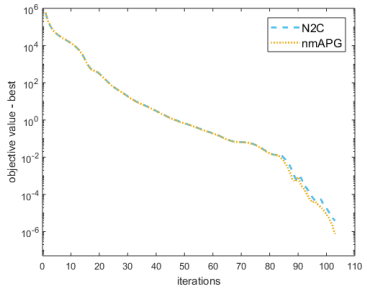
5.1.1 IMPACT OF NONCONVEXITY OF THE LOSS ON NMAPG

Recall that N2C uses nmAPG as the underlying proximal algorithm solver. Thus, in the above experiment, we have compared the performance of using nmAPG on (i) the transformed problem, which has a convex regularizer and a more nonconvex loss; and (ii) the original problem, which has a nonconvex regularizer and a less nonconvex loss). In the following, we will provide more empirical evidence that the increased nonconvexity of the transformed loss does not harm convergence of nmAPG. Instead of tuning the regularization parameter μ in (58) using the validation set, we vary μ in $\{0.01, 0.1, 1\}$. A larger μ makes the regularizer g more nonconvex, and more nonconvexity will be transferred to $\bar{f}$ by the proposed transformation. The other parts of the experimental setup are the same. Figure 6 compares the convergence behavior of N2C and nmAPG w.r.t. the number

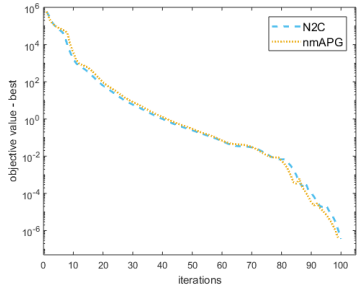
of iterations. As can be seen, they are almost identical, which agrees with Figure 4(b). Hence, nonconvexity of the loss have little effect on the empirical performance of nmAPG.



(a) $\mu = 0.01$.



(b) $\mu = 0.1$.



(c) $\mu = 1$.

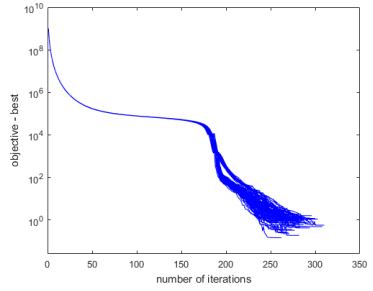
Figure 6: Convergence of N2C and nmAPG on nonconvex sparse group lasso with various μ.

5.1.2 NONCONVEX OPTIMIZATION

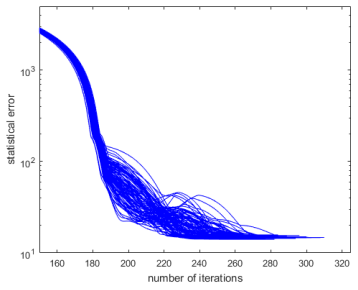
As discussed in (Ge et al., 2015), there are two main theoretical issues in optimizing nonconvex functions. The first issue is that it may be hard to find the global minimum. However, it has been shown that for composite minimization problems with nonconvex regularizer and the loss function satisfying the restricted strong convexity condition (of which nonconvex lasso is such an example), *all stationary points* can have nearly the same statistical error[1] (Loh and Wainwright, 2015). Note that with the proposed transformation, the transformed optimization problem has the same stationary points as the original problem, and so all stationary points still have nearly the same statistical error. To verify this, we experiment on the nonconvex sparse group lasso with 100 different initializations. Figure 7 shows the convergence behaviour and the obtained statistical error from the N2C algorithm. As can be seen from Figure 7(a), the differences in the final objective values obtained from different

1. Let $\bar{x}$ be the ground-truth predictor and x be an arbitrary stationary point. The statistical error is defined as $\|x - \bar{x}\|_2$.

initialization are small. From Figure 7(b), we can also see that the obtained solutions have nearly the same statistical error.



(a) objective vs number of iterations.



(b) statistical error vs number of iterations.

Figure 7: Objective / statistical error on nonconvenx sparse group lasso ($\mu = 1$). Here, N2C algorithm is used, and the "best" in Figure 7(a) is the lowest objective among the 100 random runs.

The second issue is that even finding a local minimum can be hard, as the optimizer may get trapped in saddle points. However, recent papers (Ge et al., 2015, 2017, 2016; Lee et al., 2016, 2017) have shown that batch gradient descent and noisy stochastic gradient descent almost never converge to saddle points, and can converge to a local minimizer. We expect that a similar result also holds for proximal gradient descent, but will leave this as future work.

5.2 Nonconvex Tree-Structured Group Lasso

In this section, we perform experiments on the nonconvex tree-structured group lasso model in Section 4.1.2. Following (Liu and Ye, 2010), we use the face data set *JAFFE*[2], which contains 213 256×256 images with seven facial expressions: anger, disgust, fear, happy, neutral, sadness and surprise. Their tree structure, which is based on pixel neighborhoods, is also used here. The number of groups K is 341.

Since our goal is only to demonstrate usefulness of the proposed convexification scheme, we focus on the binary classification problem "anger vs not-anger" (with 30 anger images and 183 non-anger images). The logistic loss is used, which is more appropriate for classification. Given training samples $\{(a_1, y_1), \ldots, (a_N, y_N)\}$, the optimization problem is then

$$\min_x \sum_{i=1}^{N} w_i \log\left(1 + \exp(-y_i \cdot a_i^\top x)\right) + \mu \sum_{i=1}^{K} \lambda_i \kappa\left(\|x_{\mathcal{G}_i}\|_2\right), \tag{59}$$

where $\kappa(\cdot)$ is the LSP regularizer (with $\theta = 0.5$), w_i's are weights (set to be the reciprocal of the size of sample i's class) used to alleviate class imbalance, and $\lambda_i = 1/\sqrt{\|\mathcal{G}_i\|_1}$ as in (Liu and Ye,

2. http://www.kasrl.org/jaffe.html

2010). We use 60% of the data for training, 20% as validation set to tune μ, and the rest for testing. For the proposed N2C algorithm, the proximal step of the convexified regularizer is obtained as in (Liu and Ye, 2010).

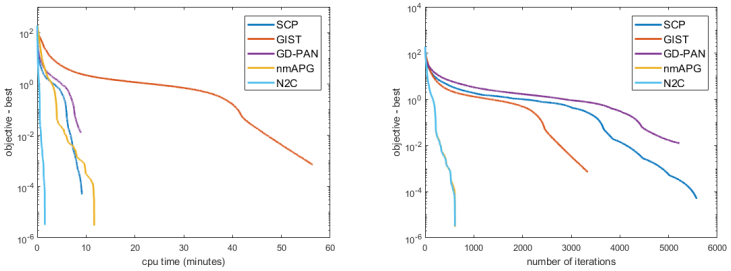
As in Section 5.1, we compare the proposed N2C with SCP, GIST, GD-PAN, nmAPG, and FISTA. The stepsize η is obtained by line search. For performance evaluation, we use (i) the testing accuracy; (ii) solution sparsity (i.e., percentage of nonzero elements); and (iii) CPU time. To reduce statistical variability, the experimental results are averaged over 5 repetitions.

Results are shown in Table 4. As can be seen, all nonconvex models have similar testing accuracies, and they again outperform the convex model. Moreover, solutions from the nonconvex models are sparser. Overall, N2C is the fastest and has the sparsest solution.

	non-accelerated			accelerated		convex
	SCP	GIST	GD-PAN	nmAPG	N2C	FISTA
testing accuracy (%)	**99.5±1.0**	**99.5±1.0**	**99.5±1.0**	**99.5±1.0**	**99.5±1.0**	96.7±1.3
sparsity (%)	**9.8±0.9**	11.6±3.6	**9.7±0.9**	**9.6±0.9**	**9.6±0.8**	24.1±0.8
CPU time (min)	10.3±1.5	68.0±17.2	12.0±2.4	11.7±1.3	**1.7±0.2**	0.5±0.1

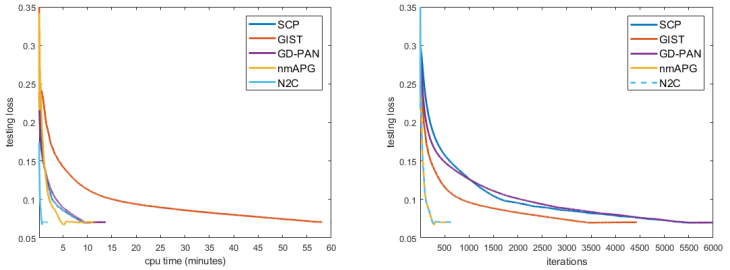
Table 4: Results on tree-structured group lasso. The best and comparable results (according to the paired t-test with 95% confidence) are highlighted.

Figure 8 shows convergence of the algorithms versus CPU time and number of iterations. The observations are similar to those in Figure 8, and N2C is the fastest. GIST is the slowest, as it does not utilize acceleration and its proximal step is solved numerically which is expensive. GD-PAN converges to a less optimal solution due to its use of approximation. Moreover, as in Section 5.1, nmAPG and N2C show similar convergence behavior w.r.t. the number of iterations (Figure 8(b)), but N2C is much faster w.r.t. time (Figure 8(a)). Convergence of the testing loss is shown in Figure 9.



(a) Objective vs CPU time (minutes). (b) Objective vs number of iterations.

Figure 8: Convergence of objective on nonconvex tree-structured group lasso. Note that the curves for nmAPG and N2C overlap in Figure 8(b).



(a) Testing loss vs CPU time (minutes). (b) Testing loss vs number of iterations.

Figure 9: Convergence of testing loss on nonconvex tree-structured group lasso.

As in Section 5.1.1, Figure 10 shows the convergence of N2C and nmAPG w.r.t. the number of iterations when μ is varied in $\{0.01, 0.1, 1\}$. Again, their behavior are almost identical. Hence, the increased nonconvexity of the transformed loss does not harm convergence of nmAPG.

5.3 Nonconvex Low-Rank Matrix Completion

In this section, we perform experiments on nonconvex low-rank matrix completion (Section 4.2), with the square loss in (30). The LSP regularizer is used, with $\theta = \sqrt{\mu}$ as in (Yao et al., 2015). We use the data sets MovieLens, Netflix and Yahoo, which have been commonly used for evaluating matrix completion (Mazumder et al., 2010; Wen et al., 2012; Hsieh and Olsen, 2014). The MovieLens and Netflix data sets contain ratings $\{1, 2, \ldots, 5\}$ assigned by various users on movies, while the Yahoo data set contains ratings $\{10, 20, \ldots, 100\}$ on music. Following (Yao et al., 2015), we normalize the ratings to zero mean and unit variance.

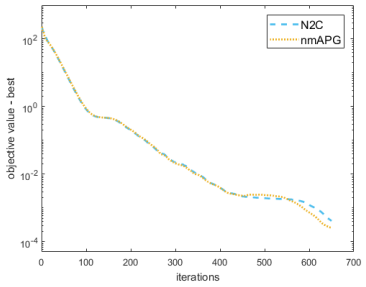
		#users	#movies	#ratings
MovieLens	100K	943	1,682	100,000
	1M	6,040	3,449	999,714
	10M	69,878	10,677	10,000,054
Netflix		480,189	17,770	100,480,507
Yahoo		249,012	296,111	62,551,438

Table 5: Recommendation data sets used in the experiments.

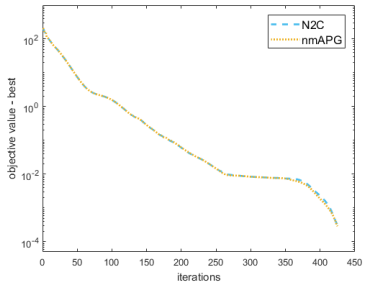
5.3.1 MOVIELENS

The proposed FW procedure (Algorithm 4), denoted N2C-FW, is compared with the following algorithms:

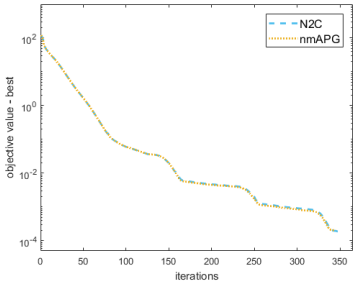
1. FaNCL (Yao et al., 2015): This is a recent nonconvex matrix regularization algorithm. It is based on the proximal algorithm using efficient approximate SVD and automatic thresholding of singular values.



(a) $\mu = 0.01$.



(b) $\mu = 0.1$.



(c) $\mu = 1$.

Figure 10: Convergence of N2C and nmAPG on tree-structured group lasso with various μ.

2. LMaFit (Wen et al., 2012): It factorizes X as a product of low-rank matrices $U \in \mathbb{R}^{m\times k}$ and $V \in \mathbb{R}^{n\times k}$. The nonconvex objective $\frac{1}{2}\|P_\Omega(UV^\top - O)\|_F^2$ is then minimized by alternating minimization on U and V using gradient descent.

3. Active subspace selection (denoted "active") (Hsieh and Olsen, 2014): This solves the (convex) nuclear norm regularized problem (with κ being the identity function in (8)) by using the active row/column subspaces to reduce the optimization problem size.

We do not compare with IRNN (Lu et al., 2014) and GPG (Lu et al., 2015), which have been shown to be much slower than FaNCL (Yao et al., 2015).

Following (Yao et al., 2015), we use 50% of the ratings for training, 25% for validation and the rest for testing. For performance evaluation, we use (i) the testing RMSE; and (ii) the recovered rank. To reduce statistical variability, the experimental results are averaged over 5 repetitions.

Results are shown in Table 6. As can be seen, the nonconvex models (N2C-FW, FaNCL and LMaFit) achieve lower RMSEs than the convex model (active subspace selection), with N2C-FW having the smallest RMSE. Moreover, the convex model needs a much higher rank than the nonconvex models, which agrees with previous observations in (Mazumder et al., 2010; Yao et al., 2015). Thus, its running time is also much longer than the others. Figure 11 compares convergence of N2C and FaNCL w.r.t. the objective in (30). The objectives of LMaFit and active subspace selection are different from N2C-FW, and thus are not shown. As can be seen, though FaNCL uses singular value thresholding to truncate the SVD, it does not control the rank as directly as N2C-FW and so is still slower. Figures 12 compares convergence of the testing RMSE on all algorithms. As the recovered matrix ranks for the nonconvex models are very low ($2-9$ in Table 6), N2C-FW is much faster than the others as it starts from a rank-one matrix and only increases its rank by one in each iteration.

		RMSE	rank	CPU time(sec)
100K	N2C-FW	**0.855±0.004**	2	**0.8±0.1**
	FaNCL	0.857±0.003	2	2.6±0.5
	LMaFit	0.867±0.004	2	1.8±0.2
	(convex) active	0.875±0.002	52	9.4±0.3
1M	N2C-FW	**0.785±0.001**	5	**19.9±0.5**
	FaNCL	0.786±0.001	5	53.6±6.5
	LMaFit	0.812±0.002	5	45.1±2.7
	(convex) active	0.811±0.001	106	124.6±1.3
10M	N2C-FW	**0.778±0.001**	9	313.0±2.2
	FaNCL	0.779±0.001	9	615.7±6.0
	LMaFit	0.797±0.001	9	**264.9±3.9**
	(convex) active	0.808±0.001	137	904.8±30.2

Table 6: Results on the MovieLens data sets. The best results (according to the paired t-test with 95% confidence) are highlighted.

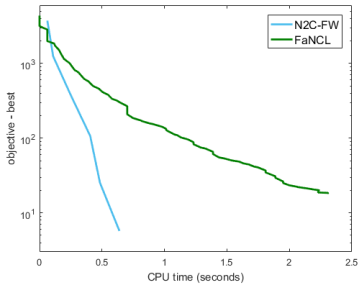
5.3.2 NETFLIX AND YAHOO

Next, we perform experiments on two very large recommendation data sets, Netflix and Yahoo (Table 5). We randomly use 50% of the observed ratings for training, 25% for validation and the rest for testing. As active subspace selection has been shown to be slower and inferior to the others (Table 6), it is not compared here. Each experiment is repeated five times. Results are shown in Table 7, and a more detailed convergence comparison with CPU time is shown in Figures 13 and 14. Again, N2C is much faster than FaNCL, and has the lowest testing RMSE.

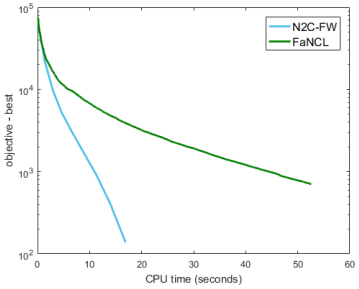
5.4 Comparison with HONOR

In this section, we experimentally compare the proposed method with HONOR (Section 4.5) on the model in (43), using the logistic loss and LSP regularizer. Following (Gong and Ye, 2015a), we fix $\mu = 1$ in (43), and θ in the LSP regularizer to 0.01μ. Experiments are performed on three large data sets,[3] kdd2010a, kdd2010b and url (Table 8). Both kdd2010a and kdd2010b are educational data sets, and the task is to predict students' successful attempts to answer concepts related to algebra.

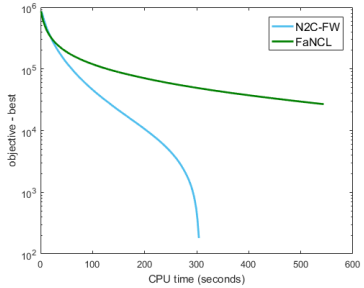
3. https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html



(a) MovieLens-100K.



(b) MovieLens-1M.



(c) MovieLens-10M.

Figure 11: Convergence of objective vs CPU time on the MovieLens data sets for nonconvex low-rank matrix completion.

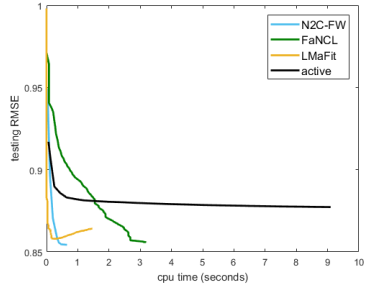
The url data set contains a collection of websites, and the task is to predict whether a particular website is malicious. We compare

1. running HONOR directly on (43). The threshold of the hybrid step in HONOR is set to 10^{-10}, which yields the best empirical performance in (Gong and Ye, 2015a);

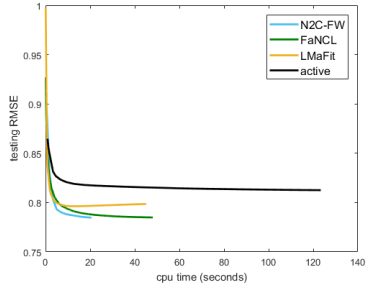
2. running mOWL-QN (Gong and Ye, 2015b) on the transformed problem (44).

To reduce statistical variability, the experimental results are averaged over 5 repetitions.

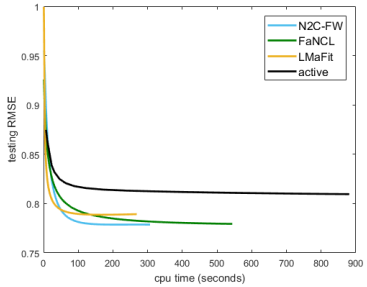
Figure 15 shows convergence of the objective (which is the same in (43) and (44)) with CPU time. As can be seen, mOWL-QN converges faster than HONOR. This validates our claim that the curvature information of the nonconvex regularizer helps.



(a) MovieLens-100K.



(b) MovieLens-1M.



(c) MovieLens-10M.

Figure 12: Convergence of testing RMSE vs CPU time on the MovieLens data sets for nonconvex low-rank matrix completion.

5.5 Image Denoising

In this section, we perform experiments on total variation image denoising with nonconvex loss and nonconvex regularizer (as introduced in Section 4.6.1). The LSP function (with $\theta = 1$) is used as κ in (46) on both the loss and regularizer. Eight popular images[4] are used (Figure 16). They are then corrupted by pepper-and-salt noise, with 10% of the pixels randomly set to 0 or 255 with equal probabilities.

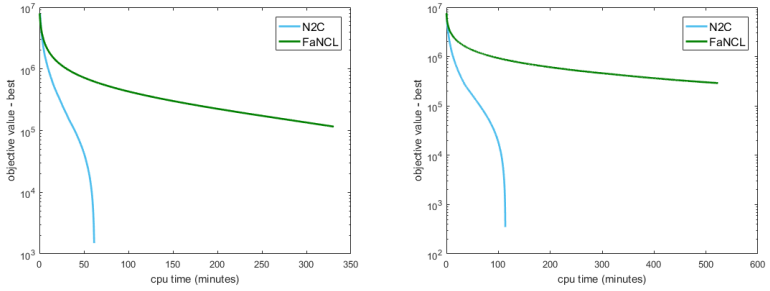
For performance evaluation, we use the RMSE $= \sqrt{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(X_{ij} - \bar{X}_{ij})^2}$, where $\bar{X} \in \mathbb{R}^{m\times n}$ is the clean image, and $X \in \mathbb{R}^{m\times n}$ is the recovered image. To tune μ, we pick the value that leads to the smallest RMSE on the first four images (boat, couple, fprint, hill). Denoising performance is then reported on the remaining images (house, lena, man, peppers).

The following algorithms will be compared:

4. http://www.cs.tut.fi/~foi/GCF-BM3D/

		RMSE	rank	CPU time(min)
Netflix	N2C-FW	**0.792±0.001**	13	**64.9±1.8**
	FaNCL	0.793±0.001	13	322.1±7.3
	LMaFit	0.807±0.001	15	98.2±1.8
Yahoo	N2C-FW	**0.643±0.001**	9	**111.3±13.5**
	FaNCL	0.650±0.001	9	444.7±92.4
	LMaFit	0.666±0.001	12	179.3±37.5

Table 7: Results on the Netflix and Yahoo data sets. The best results (according to the paired t-test with 95% confidence) are highlighted.



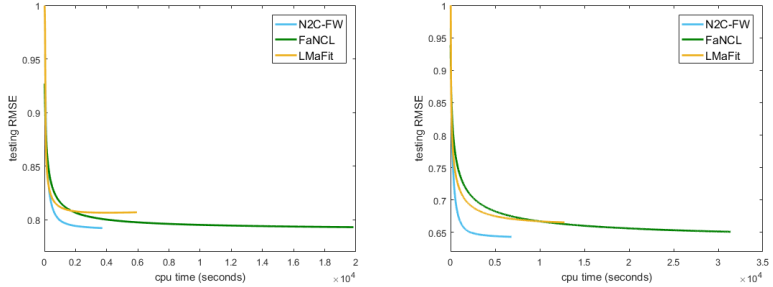
(a) Netflix. (b) Yahoo.

Figure 13: Convergence of objective vs CPU time on the Netflix and Yahoo data sets for nonconvex low-rank matrix completion.

1. CCCP (Yuille and Rangarajan, 2002): Proposition 9 is used to construct the DC decomposition for κ (Details are in Appendix B.1);
2. Smoothing (Chen, 2012): The nonsmooth κ is smoothed, and then gradient descent is used (Details are in Appendix B.2);
3. nmAPG (Li and Lin, 2015): This optimizes (50) with Algorithm 2, and the exact proximal step is solved numerically using CCCP;
4. inexact-nmAPG: This optimizes (47) with Algorithm 3 (with $\epsilon_t = 0.95^t$), and the inexact proximal step is solved numerically using L-BFGS.
5. As a baseline, we also compare with ADMM (Boyd et al., 2011) with the convex formulation.

To reduce statistical variability, the experimental results are averaged over 5 repetitions.

The RMSE results are shown in Table 9. As can be seen, the (convex) ADMM formulation leads to the highest RMSE, while CCCP, smoothing, nmAPG and inexact-nmAPG have the same RMSE which is lower than that of ADMM. This agrees with previous observations that nonconvex formulations can yield better performance than the convex ones. Timing results are shown in



(a) Netflix. (b) Yahoo.

Figure 14: Convergence of testing RMSE vs CPU time on the Netflix and Yahoo data sets for nonconvex low-rank matrix completion.

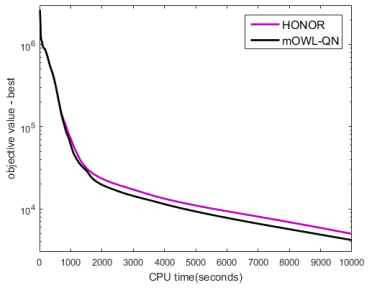
	kdd2010a	kdd2010b	url
number of samples	510,302	748,401	2,396,130
number of features	20,216,830	29,890,095	3,231,961

Table 8: Data sets used in the comparison with HONOR.

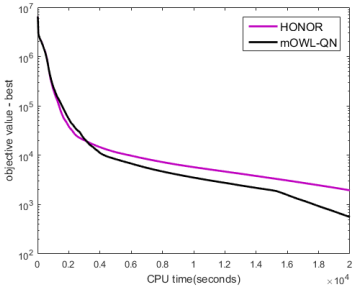
Table 10 and Figure 17. As can be seen, smoothing has low iteration complexity but suffers from slow convergence. CCCP and nmAPG both need to exactly solve a subproblem, and thus are also slow. The inexact-nmAPG algorithm does not guarantee the objective value to be monotonically decreasing as iteration proceeds. As the inexactness is initially large, there is an initial spike in the objective. However, inexact-nmAPG then quickly converges, and is much faster than all the baselines.

	house	lena	man	peppers
CCCP	**0.0205±0.0010**	**0.0174±0.0005**	**0.0223±0.0002**	**0.0207±0.0009**
smoothing	**0.0205±0.0011**	**0.0174±0.0005**	**0.0223±0.0002**	**0.0207±0.0009**
nmAPG	**0.0205±0.0010**	**0.0174±0.0005**	**0.0223±0.0002**	**0.0207±0.0009**
inexact-nmAPG	**0.0205±0.0010**	**0.0174±0.0005**	**0.0223±0.0002**	**0.0207±0.0009**
(convex) ADMM	0.0223±0.0011	0.0193±0.0005	0.0242±0.0002	0.0229±0.0008

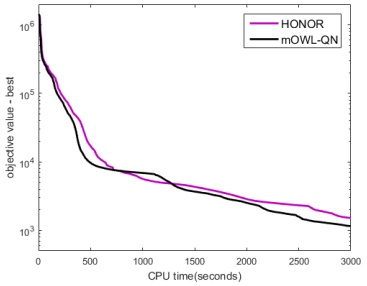
Table 9: RMSE for image denoising. The best RMSE's (according to the paired t-test with 95% confidence) are highlighted.



(a) kdd2010a.



(b) kdd2010b.



(c) url.

Figure 15: Convergence of the objective vs CPU time for HONOR and mOWL-QN.

(a) boat. (b) couple. (c) fprint. (d) hill. (e) house. (f) lena. (g) man. (h) peppers.

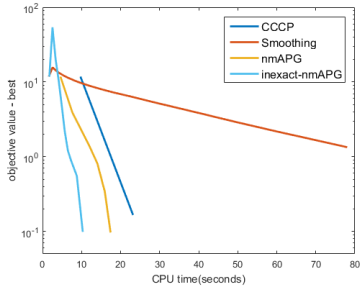
Figure 16: Samples images used in the denoising experiment. Top: Clean images; Bottom: Noisy images.

6. Conclusion

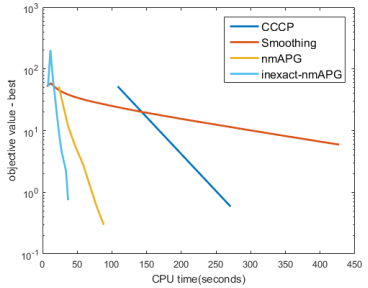
In this paper, we proposed a novel approach to learning with nonconvex regularizers that are variants of the convex ℓ_1-norm. By moving the nonconvexity associated with the nonconvex regularizer

	house	lena	man	peppers
CCCP	21.0±2.3	270.0±13.0	325.3±17.4	14.5±1.2
smoothing	75.5±2.0	433.1±4.8	437.7±6.8	61.9±1.7
nmAPG	19.4±2.3	91.4±7.3	104.4±2.7	16.1±1.8
inexact-nmAPG	**10.3±1.1**	**37.9±5.0**	**43.0±7.6**	**8.1±0.2**
(convex) ADMM	3.0±0.1	42.8±1.1	46.9±1.0	2.2±0.1

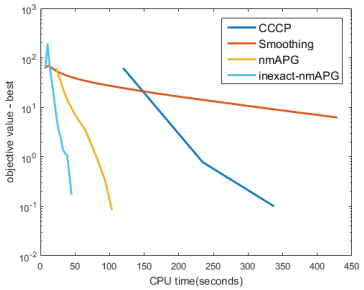
Table 10: CPU time (seconds) for image denoising. The shortest CPU time (according to the paired t-test with 95% confidence) are highlighted.



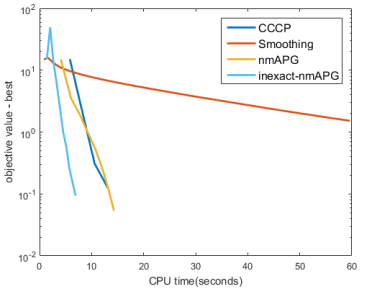
(a) house.



(b) lena.



(c) man.



(d) peppers.

Figure 17: CPU time (seconds) vs objective value on different images.

to the loss, the nonconvex regularizer is convexified to become a familiar convex regularizer while the augmented loss is still Lipschitz smooth. This allows one to reuse efficient algorithms originally designed for convex regularizers on the transformed problem. To illustrate usages with the proposed transformation, we plug it into many popular optimization algorithms. First, we consider the proximal algorithm, and showed that while the proximal step is expensive on the

original problem, it becomes much easier on the transformed problem. We further propose an inexact proximal algorithm, which allows inexact update of proximal step when it does not have a closed-form solution. Second, we combine the proposed convexification scheme with the Frank-Wolfe algorithm on learning low-rank matrices, and showed that its crucial linear programming step becomes cheaper and more easily solvable. As no convergence results exist on this nonconvex problem, we designed a novel Frank-Wolfe algorithm based on the proposed transformation and with convergence guarantee. Third, when using with ADMM and SVRG, we showed that the existing convergence results can be applied on the transformed problem but not on the original one. We further extend the proposed transformation to handle nonconvex and nonsmooth loss functions, and illustrate its benefits on the total variation model and robust sparse coding. Finally, we demonstrate the empirical advantages of working with the transformed problems on various tasks with both synthetic and real-world data sets. Experimental results show that better performance can be obtained with nonconvex regularizers, and algorithms on the transformed problems run much faster than the state-of-the-art on the original problems.

Acknowledgments

This research was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region (Grant 614513).

Appendix A. Proofs

A.1 Proposition 1

Proof First, we introduce a few Lemmas.

Lemma 14 (Golub and Van Loan (2012)) *For $x \neq 0$, the gradient of the ℓ_2-norm is $\nabla_{x_i}\|x\|_2 = x_i/\|x\|_2$.*

Let $h(z) = \kappa(\|z\|_2) - \kappa_0\|z\|_2$.

Lemma 15

$$\nabla_{z_i} h(z) = \begin{cases} \frac{\kappa'(\|z\|_2)-\kappa_0}{\|z\|_2} z_i & z \neq 0 \\ 0 & \textit{otherwise} \end{cases}. \tag{60}$$

Proof For $z \neq 0$, $\|z\|_2$ is differentiable (Lemma 14), and we obtain the first part of (60). For $z = 0$, let $\bar{h}_i(z) = \frac{\kappa'(\|z\|_2)-\kappa_0}{\|z\|_2} z_i$. Consider any Δ with $\|\Delta\|_2 = 1$.

$$\begin{aligned} \lim_{\alpha\to 0^+} \bar{h}_i(0+\alpha\Delta) &= \lim_{\alpha\to 0^+} \frac{\kappa'(\|\alpha\Delta\|_2) - \kappa_0}{\|\alpha\Delta\|_2}\alpha\Delta_i \\ &= \lim_{\alpha\to 0^+} (\kappa'(\alpha) - \kappa_0)\Delta_i = 0, \end{aligned}$$

as $\lim_{\alpha\to 0^+} \kappa'(\alpha) - \kappa_0 = 0$. Thus, $h(z)$ is smooth at $z = 0$, and we obtain the second part of (60). ■

Lemma 16 (Eriksson et al. (2004)) *Let $f : \mathbb{R} \to \mathbb{R}$ be a differentiable function. If its derivative f' is bounded, then f is Lipschitz-continuous with constant c where c is equal to the maximum value of $|f'|$.*

Lemma 17 (Eriksson et al. (2004)) *If a continuous function $f : \mathbb{R} \to \mathbb{R}$ is L_1-Lipschitz continuous in $[a, b]$ and L_2-Lipschitz continuous in $[b, c]$ (where $-\infty \leq a < b < c \leq \infty$), then it is $\max(L_1, L_2)$-Lipschitz continuous in $[a, c]$.*

Lemma 18 *Let z be an arbitrary vector, and e_i be the unit vector with only its ith dimension equal to 1. Define $\hat{h}_i(\gamma) = \frac{\kappa'(\|z+e_i\gamma\|_2)-\kappa_0}{\|z+e_i\gamma\|_2}(z_i + \gamma)$. Then, $\hat{h}$ is 2ρ-Lipschitz continuous.*

Proof Let the finite non-differentiable points of κ' be $\{\hat{\alpha}_1, \ldots, \hat{\alpha}_k\}$, where $\hat{\alpha}_1 < \cdots < \hat{\alpha}_k$. We partition $(-\infty, \infty)$ into intervals $(-\infty, \hat{\alpha}_1] \cup [\hat{\alpha}_1, \hat{\alpha}_2] \cup \cdots \cup [\hat{\alpha}_k, \infty)$, such that κ'' exists in each interval. Let $w = z + e_i\gamma$. For any interval,

$$\hat{h}_i'(\gamma) = \frac{\kappa''(\|w\|_2)}{\|w\|_2}(z_i+\gamma)^2 + \left(1 - \frac{(z_i+\gamma)^2}{\|w\|_2^2}\right)\frac{\kappa'(\|w\|_2) - \kappa_0}{\|w\|_2}. \tag{61}$$

Let $\phi(\alpha) = \kappa'(\alpha) - \kappa_0$, where $\alpha \geq 0$. Note that $\phi(0) = 0$. Moreover, $\phi(\alpha)$ is ρ-Lipschitz continuous as κ is ρ-Lipschitz smooth. Thus,

$$|\phi(\alpha) - \phi(0)| = |\kappa'(\alpha) - \kappa_0| \leq \rho\alpha,$$

and so

$$\left|\kappa'(\|w\|_2) - \kappa_0\right| \leq \rho\|w\|_2. \tag{62}$$

Note that $(z_i + \gamma)^2 \leq \|w\|_2^2$, (61) can be rewritten as

$$\begin{aligned} \left|\hat{h}_i'(\gamma)\right| &\leq \left|\frac{\kappa''(\|w\|_2)}{\|w\|_2}(z_i+\gamma)^2\right| + \left|\left(1 - \frac{(z_i+\gamma)^2}{\|w\|_2^2}\right)\frac{\kappa'(\|w\|_2) - \kappa_0}{\|w\|_2}\right| \\ &\leq \left|\kappa''(\|w\|_2)\right| + \left|\frac{\kappa'(\|w\|_2) - \kappa_0}{\|w\|_2}\right| \leq 2\rho, \end{aligned}$$

where the last inequality is due to that κ is ρ-Lipschitz smooth and (62). Thus, $|\hat{h}_i'(\gamma)| \leq 2\rho$, and by Lemma 16, we have $\hat{h}_i(\gamma)$ is 2ρ-Lipschitz continuous on any interval. Obviously $\hat{h}_i$ is continuous, and we conclude that $\hat{h}_i$ is also 2ρ-Lipschitz continuous by Lemma 17. ■

From Lemma 18, $\hat{h}_i$ is 2ρ-Lipschitz continuous. Thus, ∇h is 2ρ-Lipschitz continuous in each of its dimensions. For any $x, y \in \mathbb{R}^d$,

$$\begin{aligned} \|\nabla h(x) - \nabla h(y)\|_2^2 &= \sum_{i=1}^d [\nabla_{x_i} h(x) - \nabla_{y_i} h(y)]^2 \\ &\leq 4\rho^2 \sum_{i=1}^d (x_i - y_i)^2 = 4\rho^2\|x - y\|_2^2, \end{aligned}$$

and hence h is 2ρ-Lipschitz smooth. Finally, we will show that $h(z)$ is also concave.

Lemma 19 (Boyd and Vandenberghe (2004)) *$\phi(x) = \pi(q(x))$ is concave if π is concave, non-increasing and q is convex.*

Let $\pi(\alpha) = \kappa(\alpha) - \kappa_0\alpha$, where $\alpha \geq 0$. Note that π is concave. Moreover, $\pi(0) = 0$ and $\pi'(\alpha) \leq 0$. Thus, $\pi(\alpha)$ is non-increasing on $\alpha \geq 0$. Next, let $q(z) = \|z\|_2$. Then, $h(z) \equiv \kappa(\|z\|_2) - \kappa_0\|z\|_2 = \pi(q(z))$. As q is convex, $h(z)$ is concave from Lemma 19. ■

A.2 Corollary 2

Proof From Proposition 1 and the definition of $\bar{g}_i$, we can see that $\bar{g}_i$ is concave. Then, for any x, y,

$$\|\nabla h(A_i x) - \nabla h(A_i y)\|_2^2 \leq 4\rho^2\|A_i x - A_i y\|_2^2 \leq 4\rho^2\|A_i\|_F^2\|x - y\|_2^2.$$

Thus, $\bar{g}_i$ is $2\rho\|A_i\|_F$-Lipschitz smooth. ■

A.3 Corollary 3

Proof It is easy to see that $\breve{g}(x) = \kappa_0 \sum_{i=1}^K \mu_i\|A_i x\|_2$ is convex but not smooth. Using Corollary 2, as each $\bar{g}_i$ is concave and Lipschitz-smooth, $\bar{g}$ is also concave and Lipschitz-smooth. ■

A.4 Proposition 5

Proof First, we introduce a few Lemmas.

Definition 20 (Bertsekas (1999)) *A function $f : \mathbb{R}^m \to \mathbb{R}$ is* absolute symmetric *if $f([x_1; \ldots; x_m]) = f\left([|x_{\pi(1)}|; \ldots; |x_{\pi(m)}|]\right)$ for any permutation π.*

Lemma 21 (Lewis and Sendov (2005)) *Let $\sigma(X) = [\sigma_1(X); \ldots; \sigma_m(X)]$ be the vector containing singular values of X. For an absolute symmetric function $f : \mathbb{R}^m \to \mathbb{R}$, $\phi(X) \equiv f(\sigma(X))$ is concave on X if and only if f is concave.*

From the definition of $\bar{g}$ in (24),

$$\bar{g}(X) = \bar{\mu}\sum_{i=1}^m(\kappa(\sigma_i(X)) - \kappa_0\|X\|_*) = \bar{\mu}\sum_{i=1}^m(\kappa(\sigma_i(X)) - \kappa_0\sigma_i(X)).$$

Let

$$h(x) = \bar{\mu}\sum_{i=1}^m(\kappa(|x_i|) - \kappa_0|x_i|). \tag{63}$$

Obviously, h is absolute symmetric. From Remark 4, h is concave. Thus, $\bar{g}$ is also concave by Lemma 21.

Lemma 22 (Lewis and Sendov (2005)) *Let the SVD of X be $U\text{Diag}(\sigma(X))V^\top$, where $\sigma(X) = [\sigma_1(X); \ldots; \sigma_m(X)]$, $f : \mathbb{R}^m \to \mathbb{R}$ be smooth and absolute symmetric, and $\phi(X) \equiv f(\sigma(X))$. We have*

1. *$\nabla\phi(X) = U\text{Diag}(\nabla f(\sigma(X)))V^\top$; and*
2. *If f is L-Lipschitz smooth, then ϕ is also L-Lipschitz smooth.*

From Remark 4, h in (63) is 2ρ-Lipschitz smooth. Hence, from Lemma 22, $\bar{g}(X)$ is also 2ρ-Lipschitz smooth and $\nabla\bar{g}(X) = U\text{Diag}(\nabla h(\sigma(X)))V^\top$. ■

A.5 Proposition 9

Proof First, we introduce the following Lemma.

Lemma 23 (Boyd and Vandenberghe (2004)) *$\phi(x) = \pi(q(x))$ is convex if π is convex, non-decreasing and q is convex.*

Let $\pi(\alpha) = \kappa(\alpha) + \frac{\rho}{2}\alpha^2$ where $\alpha \geq 0$. As κ is ρ-Lipschitz smooth, $\kappa'(\beta) - \kappa'(\alpha) \leq \rho(\alpha - \beta)$. Thus, $\pi'(\alpha) - \pi'(\beta) = \kappa'(\alpha) + \rho\alpha - \kappa'(\beta) - \rho\beta \geq 0$, i.e., π is convex. Besides, $\pi'(0) = \kappa'(0) \geq 0$. Thus, $\pi'(\alpha) \geq 0$ and π is also non-decreasing. Let $q(x) = \|x\|$ which is obviously convex as $\|\cdot\|$ is a norm, we can express $\phi(x) = \pi(q(x)) = \kappa(\|x\|_x) + \frac{\rho}{2}\|x\|^2$. Finally, $\phi(x)$ is also convex due to Lemma 23. ■

A.6 Theorem 6

Proof First, we introduce a few Lemmas.

Lemma 24 *Let $\tilde{X}$ be an inexact solution of the proximal step $\min_Z h(Z)$, where $h(Z) = \frac{1}{2}\|Z - (X - \frac{1}{\tau}\nabla\bar{f}(X))\|_F^2 + \frac{1}{\tau}\breve{g}(Z)$. Let $\hat{X} = \arg\min_Z h(Z)$. If $h(\tilde{X}) - h(\hat{X}) \leq \epsilon$, then*

$$F(\tilde{X}) \leq F(X) - \frac{\tau - \bar{L}}{2}\|\tilde{X} - X\|_F^2 + \tau\epsilon.$$

Proof Let $\phi(Z) = \langle Z - X, \nabla f(X)\rangle + \frac{\tau}{2}\|Z - X\|_F^2 + \breve{g}(Z)$. We have

$$\hat{X} = \arg\min_Z h(Z) = \arg\min_Z \phi(Z), \tag{64}$$

$$\phi(Z) = \tau h(Z) - \frac{1}{\tau}\|\nabla\bar{f}(X)\|_F^2. \tag{65}$$

From (64), we have

$$\phi(\hat{X}) = \langle\hat{X} - X, \nabla f(X)\rangle + \frac{\tau}{2}\|\hat{X} - X\|_F^2 + \breve{g}(\hat{X}) \leq \breve{g}(X). \tag{66}$$

As $h(\tilde{X}) - h(\hat{X}) \leq \epsilon$, from (65) (note that $\|\nabla\bar{f}(X)\|_F^2$ is a constant), we have

$$\phi(\tilde{X}) - \phi(\hat{X}) = \tau(h(\tilde{X}) - h(\hat{X})) \leq \tau\epsilon.$$

Then with (66), we have $\phi(\tilde{X}) \le \tau\epsilon + \phi(\hat{X}) \le \breve{g}(X) + \tau\epsilon$, i.e.,

$$\langle \tilde{X} - X, \nabla f(X)\rangle + \frac{\tau}{2}\|\tilde{X} - X\|_F^2 + \breve{g}(\tilde{X}) \le \breve{g}(X) + \tau\epsilon. \tag{67}$$

As $\bar{f}$ is $\bar{L}$-Lipschitz smooth,

$$\bar{f}(\tilde{X}) \le \bar{f}(X) + \langle \tilde{X} - X, \nabla f(X)\rangle + \frac{\bar{L}}{2}\|\tilde{X} - X\|_F^2.$$

Combining with (67), we obtain

$$\bar{f}(\tilde{X}) + \frac{\tau}{2}\|\tilde{X} - X\|_2^2 + \breve{g}(\tilde{X}) \le \bar{f}(X) + \frac{\bar{L}}{2}\|\tilde{X} - X\|_F^2 + \breve{g}(X) + \tau\epsilon.$$

Thus, $F(\tilde{X}) \le F(X) - \frac{\tau-\bar{L}}{2}\|\tilde{X} - X\|_F^2 + \tau\epsilon$. ■

If step 6 in Algorithm 3 is satisfied, $X_{t+1} = \tilde{Z}_{t+1}$, and

$$F(X_{t+1}) \le F(X_t) - \frac{\delta}{2}\|X_{t+1} - Y_t\|_F^2. \tag{68}$$

Otherwise, step 9 is executed, and from Lemma 24, we have

$$F(X_{t+1}) \le F(X_t) - \frac{\tau - \bar{L}}{2}\|X_{t+1} - X_t\|_F^2 + \tau\epsilon_t. \tag{69}$$

Partition $\Omega(T) = \{1, 2, \ldots, T\}$ into $\Omega_1(T)$ and $\Omega_2(T)$, such that step 7 is performed if $t \in \Omega_1(T)$; and execute step 9 otherwise. Combining (68) and (69), we have

$$\begin{aligned}
&F(X_1) - F(X_{T+1})\\
&\ge \frac{\delta}{2}\sum_{t\in\Omega_1(T)}\|X_{t+1} - Y_t\|_F^2 + \frac{\tau-\bar{L}}{2}\sum_{t\in\Omega_2(T)}\left(\|X_{t+1} - X_t\|_F^2 - \tau\epsilon_t\right),\\
&\ge \frac{\delta}{2}\sum_{t\in\Omega_1(T)}\|X_{t+1} - Y_t\|_F^2 + \frac{\tau-\bar{L}}{2}\sum_{t\in\Omega_2(T)}\|X_{t+1} - X_t\|_F^2 - \frac{(\tau-\bar{L})\tau}{2}\sum_{t\in\Omega_2(T)}\epsilon_t\\
&\ge \frac{\delta}{2}\sum_{t\in\Omega_1(T)}\|X_{t+1} - Y_t\|_F^2 + \frac{\tau-\bar{L}}{2}\sum_{t\in\Omega_2(T)}\|X_{t+1} - X_t\|_F^2 - \frac{(\tau-\bar{L})\tau}{2}\sum_{t=1}^{\infty}\epsilon_t\\
&\ge \frac{\delta}{2}\sum_{t\in\Omega_1(T)}\|X_{t+1} - Y_t\|_F^2 - c_1 + \frac{\tau-\bar{L}}{2}\sum_{t\in\Omega_2(T)}\|X_{t+1} - X_t\|_F^2,
\end{aligned} \tag{70}$$

where $c_1 = \frac{(\tau-\bar{L})\tau}{2}\sum_{t=1}^{\infty}\epsilon_t < \infty$ and $c_1 \ge 0$. From (70), we have

$$\begin{aligned}
F(X_1) - \inf_X F(X) + c_1 &\ge F(X_1) - \lim_{T\to\infty} F(X_{T+1}) + c_1\\
&\ge \lim_{T\to\infty}\frac{\delta}{2}\sum_{t\in\Omega_1(T)}\|X_{t+1} - Y_t\|_F^2 + \frac{\tau-\bar{L}}{2}\sum_{t\in\Omega_2(T)}\|X_{t+1} - X_t\|_F^2\\
&\equiv c_2.
\end{aligned} \tag{71}$$

From Assumption A1, $c_2 \le F(X_1) - \inf_X F(X) + c_1 < \infty$. Thus, $c_2 \ge 0$ is a finite constant. Let $\Omega_1^\infty = \lim_{T\to\infty}\Omega_1(T)$, and $\Omega_2^\infty = \lim_{T\to\infty}\Omega_2(T)$. Consider the three cases:

1. $|\Omega_1^\infty|$ is finite, and $|\Omega_2^\infty|$ is infinite. As $|\Omega_2^\infty| = \infty$ and $\lim_{\|X\|_F\to\infty} F(X) = \infty$ from Assumption A1 and (71), we have

$$\lim_{t\in\Omega_2^\infty, t\to\infty}\|X_{t+1} - X_t\|_F^2 = 0.$$

Thus, there exists a limit point such that $X_* = \lim_{t_j\in\Omega_2^\infty, t_j\to\infty} X_{t_j}$ for a subsequence $\{X_{t_j}\}$ of $\{X_t\}$. Since $\lim_{t_j\to\infty}\epsilon_{t_j} = 0$, then

$$\lim_{t_j\in\Omega_2^\infty, t_j\to\infty} X_{t_j+1} = \lim_{t_j\in\Omega_2^\infty, t_j\to\infty}\mathrm{prox}_{\frac{1}{\tau}\breve{g}}(X_{t_j} - \frac{1}{\tau}\nabla\bar{f}(X_{t_j})).$$

As a result,

$$0 \in \lim_{t_j\in\Omega_2^\infty, t_j\to\infty}\frac{1}{\tau}\nabla\bar{f}(X_{t_j}) + (X_{t_j+1} - X_{t_j}) + \frac{1}{\tau}\partial\breve{g}(X_{t_j+1}).$$

Since both $\lim_{t_j\in\Omega_2^\infty, t_j\to\infty} X_{t_j} = \lim_{t_j\in\Omega_2^\infty, t_j\to\infty} X_{t_j+1} = X_*$, we then have $\nabla\bar{f}(X_*) + \partial\breve{g}(X_*) \ni 0$, and X_* is a critical point of (1).

2. $|\Omega_1^\infty|$ is infinite, and $|\Omega_2^\infty|$ is finite. As Ω_1^∞ is infinite and $\lim_{\|X\|_F\to\infty} F(X) = \infty$ from Assumption A1 and (71), we have

$$\lim_{t_j\in\Omega_1^\infty, t_j\to\infty}\|X_{t_j+1} - Y_{t_j}\|_F^2 = 0$$

for a subsequence $\{X_{t_j}\}$ of $\{X_t\}$. Thus, there exist a limit point such that

$$X_* = \lim_{t_j\in\Omega_1^\infty, t_j\to\infty} X_{t_j+1} = \lim_{t_j\in\Omega_1^\infty, t_j\to\infty} Y_{t_j}. \tag{72}$$

As $\lim_{t_j\to\infty}\epsilon_{t_j} = 0$, we have

$$0 \in \lim_{t_j\in\Omega_1^\infty, t_j\to\infty}\frac{1}{\tau}\nabla\bar{f}(Y_{t_j}) + (X_{t_j+1} - Y_{t_j}) + \frac{1}{\tau}\partial\breve{g}(X_{t_j+1}).$$

From (72), we have $\nabla\bar{f}(X_*) + \partial\breve{g}(X_*) \ni 0$ and X_* is a critical point of (1).

3. Both $|\Omega_1^\infty|$ and $|\Omega_2^\infty|$ are infinite. From the above two cases, we can see that once $|\Omega_1^\infty|$ or $|\Omega_2^\infty|$ is infinite, then $\{X_t\}$ is bounded, and any limit point of $\{X_t\}$ is also a critical point. In the third case, both $|\Omega_1^\infty|$ and $|\Omega_2^\infty|$ are infinite. Thus, any limit point of $\{X_t\}$ is also a critical point of (1).

As a result, $\{X_t\}$ is bounded and its limits points are critical points of (1). ■

A.7 Proposition 7

Proof From (71), we have

$$\frac{\delta}{2}\sum_{t_1\in\Omega_1(T)}\|X_{t_1+1} - Y_{t_1}\|_F^2 + \frac{\tau-\bar{L}}{2}\sum_{t_2\in\Omega_2(T)}\|X_{t_2+1} - X_{t_2}\|_F^2 < c_2, \tag{73}$$

where $c_2 \in (0, \infty)$ is a positive constant. Let $c_3 = \min(\frac{\delta}{2}, \frac{\tau - \bar{L}}{2})$. From the definition of V_t, (73) can be rewritten as

$$c_3 \sum_{t=1}^{T} \|X_{t+1} - V_t\|_F^2 \leq \frac{\delta}{2} \sum_{t_1 \in \Omega_1(T)} \|X_{t_1+1} - Y_{t_1}\|_F^2 + \frac{\tau - \bar{L}}{2} \sum_{t_2 \in \Omega_2(T)} \|X_{t_2+1} - X_{t_2}\|_F^2 \leq c_2.$$

Since c_2 is finite, thus $\lim_{t\to\infty} d_t \equiv \|X_{t+1} - V_t\|_F^2 = 0$. Besides, we have

$$\min_{t=1,\ldots,T} \sum_{t=1}^{T} \|X_{t+1} - V_t\|_F^2 \leq \frac{1}{T} \sum_{t=1}^{T} \|X_{t+1} - V_t\|_F^2 \leq \frac{c_2}{c_3 T}.$$

■

A.8 Proposition 10

Proof Note from (32) that $\nabla \bar{f}(S) = \nabla f(S) + \nabla \bar{g}(S)$. Using the matrix chain rule, since $S = \alpha X_t + \beta u_t v_t^\top$ and $\frac{\partial S}{\partial \alpha} = X_t$, then

$$\frac{\partial \bar{f}(S)}{\partial \alpha} = \left\langle \nabla \bar{f}(S), \frac{\partial S}{\partial \alpha} \right\rangle = \alpha \langle X_t, \nabla \bar{f}(S) \rangle.$$

Similarly, since $\frac{\partial S}{\partial \beta} = u_t v_t^\top$,

$$\frac{\partial \bar{f}(S)}{\partial \beta} = \left\langle \nabla \bar{f}(S), \frac{\partial S}{\partial \beta} \right\rangle = \beta \left\langle u_t v_t^\top, \nabla \bar{f}(S) \right\rangle = \beta \left(u_t^\top \nabla \bar{f}(S) v_t \right).$$

As $\bar{g}(S) = \mu \sum_{i=1}^{m} \kappa(\sigma_i(S)) - \mu \kappa_0 \sigma_i(S)$, using Lemma 22, $\nabla \bar{f}(X) = \nabla f(S) + \mu U_S \text{Diag}(w) V_S^\top$ and $w_i = \kappa'(\sigma_i(S)) - \kappa_0$. ■

A.9 Corollary 11

Proof Note that the SVD of X is $(UU_B)\text{Diag}([\sigma_1(B), \ldots, \sigma_k(B)])(VV_B)^\top$. Using Lemma 22,

$$\nabla \bar{f}(X) = \nabla f(X) + \nabla \bar{g}(X) = \nabla f(X) + \mu (UU_B) \text{Diag}(w) (VV_B)^\top,$$

where $w \in \mathbb{R}^k$ with $w_i = \kappa'(\sigma_i(B)) - \kappa_0$. ■

A.10 Proposition 12

Proof As $\bar{g}(X)$ is defined on the singular values of the input matrix X, we only need to show that $UBV^\top$ and B have the same singular values. Let the SVD of B be $U_B \text{Diag}(\sigma(B)) V_B^\top$, where $\sigma(B) = [\sigma_1(B), \ldots, \sigma_m(B)]$. As U and V are orthogonal, it is easy to see that $(UU_B)\,\text{Diag}(\sigma(B))\,(V_B V)^\top$ is the SVD of X. Thus, the Proposition holds. ■

A.11 Theorem 13

Proof We first introduce two Propositions.

Proposition 25 (Mishra et al. (2013)) *For a square matrix X, let $sym(X) = \frac{1}{2}(X + X^\top)$. The first-order optimality conditions for* (37) *are*

$$\begin{aligned} \nabla \bar{f}(X) VB - U\, sym(U^\top \nabla \bar{f}(X) VB) &= 0, \\ (\nabla \bar{f}(X))^\top UB - V\, sym(V^\top \nabla \bar{f}(X) UB) &= 0, \\ sym(U^\top \nabla \bar{f}(X) V) + \bar{\mu} I &= 0. \end{aligned}$$

Proposition 26 *If* (31) *has a critical point with rank r, choose the sizes of matrices U, V and B be $m \times r$, $n \times r$ and $r \times r$, respectively. Then, any critical point of* (37) *is also a critical point of* (31).

Proof The subdifferential of the nuclear norm can be obtained as (Watson, 1992)

$$\partial \|X\|_* = \{UV^\top + W : U^\top W = 0, WV = 0, \|W\|_\infty \leq 1\}, \tag{74}$$

where $X = UBV^\top$. Let $\hat{X} = \hat{U}\hat{B}\hat{V}^\top$ be a critical point of (37). We have $\text{sym}(\hat{U}^\top \nabla \bar{f}(\hat{X}) \hat{V}) + \bar{\mu} I = 0$ due to Proposition 25. From the property of the matrix norm, we have

$$\lambda = \|\,\text{sym}(\hat{U}^\top \nabla \bar{f}(\hat{X}) \hat{V})\|_\infty \leq \|\hat{U}^\top \nabla \bar{f}(\hat{X}) \hat{V}\|_\infty \leq \|\nabla \bar{f}(\hat{X})\|_\infty.$$

The equality holds only when $\nabla \bar{f}(\hat{X}) = -\bar{\mu} \hat{U} \hat{V}^\top - \bar{\mu} \hat{U}_\perp \hat{\Sigma}_\perp \hat{V}_\perp^\top$, where $\hat{U}_\perp$ and $\hat{V}_\perp$ are orthogonal matrices with $\hat{U}^\top \hat{U}_\perp = 0$ and $\hat{V}^\top \hat{V}_\perp = 0$, and $\hat{\Sigma}_\perp$ is a diagonal matrix with positive elements $[\Sigma_\perp]_{ii} \leq 1$. Combining this with (74), we have

$$\nabla \bar{f}(\hat{X}) \in -\bar{\mu} \partial \|\hat{X}\|_*. \tag{75}$$

Then, for (31), if X_* is a critical point, we have

$$\nabla \bar{f}(X_*) \in -\bar{\mu} \partial \|X_*\|_*. \tag{76}$$

Comparing (75) and (76), the difference is on the ranks of $\hat{X}$ and X_*. As (31) has a critical point with rank-r, $\hat{X}$ is also a critical point of (31). ■

In Algorithm 4, the sizes of U, V and B are selected as $m \times t$, $n \times t$, and $t \times t$, respectively. If (31) has a critical point with rank r, then as iteration goes and $t = r$, from Proposition 26, Algorithm 4 will return a critical point of (31). ■

Appendix B. Details in Section 5.5

B.1 CCCP

Using Proposition 9, we can decompose $\kappa(|x|) = \hat{\varsigma}(x) + \check{\varsigma}(x)$, where $\hat{\varsigma}(x) = -\frac{\rho}{2} x^2$ is convex and $\check{\varsigma}(x) = \kappa(|x|) + \frac{\rho}{2} x^2$ is concave. Apply the above decomposition on κ in (46), and we have the

following DC decomposition:

$$\breve{F}(X) = \sum_{i=1}^{m}\sum_{j=1}^{n} \breve{\varsigma}\left([Y-X]_{ij}\right) + \mu \sum_{i=1}^{m-1}\sum_{j=1}^{m} \breve{\varsigma}\left([D_v X]_{ij}\right) + \mu \sum_{i=1}^{n}\sum_{j=1}^{n-1} \breve{\varsigma}\left([XD_h]_{ij}\right),$$

$$\hat{F}(X) = \sum_{i=1}^{m}\sum_{j=1}^{n} \hat{\varsigma}\left([Y-X]_{ij}\right) + \mu \sum_{i=1}^{m-1}\sum_{j=1}^{m} \hat{\varsigma}\left([D_v X]_{ij}\right) + \mu \sum_{i=1}^{n}\sum_{j=1}^{n-1} \hat{\varsigma}\left([XD_h]_{ij}\right).$$

The CCCP procedures at Section 2.1 can then be applied.

B.2 Smoothing

As the LSP is used as κ, a smoothed version of it can be obtained as $\tilde{\kappa}_\lambda(x) = \beta \log\left(1 + \frac{h_\lambda(x)}{\theta}\right)$, where $h_\lambda(x) = \begin{cases} |x| & \text{if } |x| \geq \lambda \\ \frac{x^2}{2\lambda} + \frac{\lambda}{2} & \text{otherwise} \end{cases}$. Thus, (46) is smoothed as

$$\breve{F}_\lambda(X) = \sum_{i=1}^{m}\sum_{j=1}^{n} \tilde{\kappa}_\lambda\left([Y-X]_{ij}\right) + \mu \sum_{i=1}^{m-1}\sum_{j=1}^{m} \tilde{\kappa}_\lambda\left([D_v X]_{ij}\right) + \mu \sum_{i=1}^{n}\sum_{j=1}^{n-1} \tilde{\kappa}_\lambda\left([XD_h]_{ij}\right).$$

Gradient descent is then used for optimization (Chen, 2012). Specifically, we need to minimize a sequence of subproblems $\{\breve{F}_{\lambda_1}(X), \breve{F}_{\lambda_2}(X), \dots\}$ with $\lambda_i = \lambda_0 \nu^i$, and using X from $\breve{F}_{\lambda_{i-1}}(X)$ to warm start $\breve{F}_{\lambda_i}(X)$. In the experiment, we set $\lambda_0 = 0.1$ and $\nu = 0.95$.

References

G. Andrew and J. Gao. Scalable training of ℓ_1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine learning*, pages 33–40, 2007.

H. Bauschke, R. Goebel, Y. Lucet, and X. Wang. The proximal average: Basic theory. *SIAM Journal on Optimization*, 19(2):766–785, 2008.

A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Transactions on Image Processing*, 18(11):2419–2434, 2009.

M. Beck, A.and Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

D.P. Bertsekas and J.N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, 1989.

R.I. Bot, E. Robert Csetnek, and S.C. László. An inertial forward-backward algorithm for the minimization of the sum of two nonconvex functions. *EURO Journal on Computational Optimization*, 4(1):3–25, 2016.

L. Bottou. Online learning and stochastic approximations. *On-line Learning in Neural Networks*, 17(9):142–174, 1998.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

K. Bredies, D.A. Lorenz, and P. Maass. A generalized conditional gradient method and its connection to an iterative shrinkage method. *Computational Optimization and Applications*, 42 (2):173–193, 2009.

E.J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

E.J. Candès, M.B. Wakin, and S. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications*, 14(5-6):877–905, 2008.

E.J. Candès, X. Li, Yi Ma, and J. Wright. Robust principal component analysis. *Journal of the ACM*, 58(3):1–37, 2011.

X. Chen. Smoothing methods for nonsmooth, nonconvex minimization. *Mathematical Programming*, 134(1):71–99, 2012.

D.L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

K. Eriksson, F. Estep, and C. Johnson. *Applied Mathematics: Body and Soul: Volume 1: Derivatives and Geometry in IR3*. Springer-Verlag, 2004.

J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics*, 3(1-2): 95–110, 1956.

R. Ge, F. Huang, C. Jin, and Y. Yuan. Escaping from saddle points - online stochastic gradient for tensor decomposition. In *International Conference on Learning Theory*, pages 797–842, 2015.

R. Ge, J. Lee, and T. Ma. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems*, pages 2973–2981, 2016.

R. Ge, C. Jin, and Y. Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

D. Geman and C. Yang. Nonlinear image recovery with half-quadratic regularization. *IEEE Transactions on Image Processing*, 4(7):932–946, 1995.

S. Ghadimi and G. Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming*, 156(1-2):59–99, 2016.

R. Glowinski and A. Marroco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, 9(2):41–76, 1975.

G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 2012.

P. Gong and J. Ye. HONOR: Hybrid Optimization for NOn-convex Regularized problems. In *Advances in Neural Information Processing Systems*, pages 415–423, 2015a.

P. Gong and J. Ye. A modified orthant-wise limited memory quasi-Newton method with convergence analysis. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 276–284, 2015b.

P. Gong, C. Zhang, Z. Lu, J. Huang, and J. Ye. A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems. In *Proceedings of the 30th International Conference on Machine Learning*, pages 37–45, 2013.

B. He and X. Yuan. On the $o(1/n)$ convergence rate of the douglas-rachford alternating direction method. *SIAM Journal on Numerical Analysis*, 50(2):700–709, 2012.

J.B. Hiriart-Urruty. Generalized differentiability, duality and optimization for problems dealing with differences of convex functions. *Convexity and Duality in Optimization*, pages 37–70, 1985.

M. Hong, Z.-Q. Luo, and M. Razaviyayn. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization*, 26(1):337–364, 2016.

C.-J. Hsieh and P. Olsen. Nuclear norm minimization via active subspace selection. In *Proceedings of the 31st International Conference on Machine Learning*, pages 575–583, 2014.

L. Jacob, G. Obozinski, and J.-P. Vert. Group lasso with overlap and graph lasso. In *Proceedings of the 26th International Conference on Machine Learning*, pages 433–440, 2009.

M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning*, pages 427–435, 2013.

R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for hierarchical sparse coding. *Journal of Machine Learning Research*, 12:2297–2334, 2011.

S. Laue. A hybrid algorithm for convex semidefinite optimization. In *Proceedings of the 29th International Conference on Machine Learning*, pages 177–184, 2012.

J. Lee, M. Simchowitz, M. Jordan, and B. Recht. Gradient descent only converges to minimizers. In *Proceedings of the 29th Conference on Learning Theory*, pages 1246–1257, 2016.

J. Lee, I. Panageas, G. Piliouras, M. Simchowitz, M.I. Jordan, and B. Recht. First-order methods almost always avoid saddle points. arxiv preprint, University of Southern California, 2017.

A.S. Lewis and H.S. Sendov. Nonsmooth analysis of singular values. *Set-Valued Analysis*, 13(3): 243–264, 2005.

G. Li and T.K. Pong. Global convergence of splitting methods for nonconvex composite optimization. *SIAM Journal on Optimization*, 25(4):2434–2460, 2015.

H. Li and Z. Lin. Accelerated proximal gradient methods for nonconvex programming. In *Advances in Neural Information Processing Systems*, pages 379–387, 2015.

J. Liu and J. Ye. Moreau-Yosida regularization for grouped tree structure learning. In *Advances in Neural Information Processing Systems*, pages 1459–1467, 2010.

J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208–220, 2013.

P. Loh and M. Wainwright. Regularized M-estimators with nonconvexity: Statistical and algorithmic theory for local optima. *Journal of Machine Learning Research*, 16:559–616, 2015.

C. Lu, J. Shi, and J. Jia. Online robust dictionary learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 415–422, 2013.

C. Lu, J. Tang, S. Yan, and Z. Lin. Generalized nonconvex nonsmooth low-rank minimization. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 4130–4137, 2014.

C. Lu, C. Zhu, C. Xu, S. Yan, and Z. Lin. Generalized singular value thresholding. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 1805–1811, 2015.

Z. Lu. Sequential convex programming methods for a class of structured nonlinear programming. Preprint arXiv:1210.3039, 2012.

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th International Conference on Machine Learning*, pages 689–696, 2009.

R. Mazumder, T. Hastie, and R. Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11:2287–2322, 2010.

B. Mishra, G. Meyer, F. Bach, and R. Sepulchre. Low-rank optimization with trace norm penalty. *SIAM Journal on Optimization*, 23(4):2124–2149, 2013.

Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

T. Ngo and Y. Saad. Scaled gradients on Grassmann manifolds for matrix completion. In *Advances in Neural Information Processing Systems*, pages 1412–1420, 2012.

M. Nikolova. A variational approach to remove outliers and impulse noise. *Journal of Mathematical Imaging and Vision*, 20(1-2):99–120, 2004.

J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, 2006.

P. Ochs, Y. Chen, T. Brox, and T. Pock. iPiano: Inertial proximal algorithm for nonconvex optimization. *SIAM Journal on Imaging Sciences*, 7(2):1388–1419, 2014.

N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2013.

S.J. Reddi, A. Hefny, S. Sra, B. Póczos, and A.J. Smola. Stochastic variance reduction for nonconvex optimization. In *Proceedings of the 33nd International Conference on Machine Learning*, pages 314–323, 2016a.

S.J. Reddi, S. Sra, B. Poczos, and A. Smola. Fast stochastic methods for nonsmooth nonconvex optimization. In *Advances in Neural Information Processing Systems*, pages 1145–1153, 2016b.

M. Schmidt, N.L. Roux, and F. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. In *Advances in Neural Information Processing Systems*, pages 1458–1466, 2011.

S. Sra. Scalable nonconvex inexact proximal splitting. In *Advances in Neural Information Processing Systems*, pages 530–538, 2012.

N. Srebro, J. Rennie, and T.S. Jaakkola. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems*, pages 1329–1336, 2004.

Q. Sun, S. Xiang, and J. Ye. Robust principal component analysis via capped norms. In *Proceedings of the 19th International Conference on Knowledge Discovery and Data Mining*, pages 311–319, 2013.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, 73(3):273–282, 1996.

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B*, 67(1):91–108, 2005.

J. Trzasko and A. Manduca. Highly undersampled magnetic resonance image reconstruction via homotopic-minimization. *IEEE Transactions on Medical Imaging*, 28(1):106–121, 2009.

G.A. Watson. Characterization of the subdifferential of some matrix norms. *Linear Algebra and its Applications*, 170:33–45, 1992.

Z. Wen, W. Yin, and Y. Zhang. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation*, 4(4): 333–361, 2012.

L. Xiao and T. Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

M. Yan. Restoration of images corrupted by impulse noise and mixed gaussian impulse noise using blind inpainting. *SIAM Journal on Imaging Sciences*, 6(3):1227–1245, 2013.

M. Yang, L. Zhang, J. Yang, and D. Zhang. Robust sparse coding for face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 625–632, 2011.

Q. Yao and J.T. Kwok. Efficient learning with a family of nonconvex regularizers by redistributing nonconvexity. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 2645–2654, 2016.

Q. Yao, J.T. Kwok, and W. Zhong. Fast low-rank matrix learning with nonconvex regularization. In *Proceedings of IEEE International Conference on Data Mining*, pages 539–548, 2015.

Q. Yao, J. Kwok, F. Gao, W. Chen, and T.-Y. Liu. Efficient inexact proximal gradient algorithm for nonconvex problems. In *International Joint Conferences on Artifical Intelligence*, pages 3308–3314, 2017.

Y. Yu. Better approximation and faster algorithm using the proximal average. In *Advances in Neural Information Processing Systems*, pages 458–466, 2013.

L. Yuan, J. Liu, and J. Ye. Efficient methods for overlapping group lasso. In *Advances in Neural Information Processing Systems*, pages 352–360, 2011.

A.L. Yuille and A. Rangarajan. The concave-convex procedure (CCCP). In *Advances in Neural Information Processing Systems*, pages 1033–1040, 2002.

C.H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *Annals of Statistics*, 38(2):894–942, 2010a.

T. Zhang. Analysis of multi-stage convex relaxation for sparse regularization. *Journal of Machine Learning Research*, 11:1081–1107, 2010b.

X. Zhang, D. Schuurmans, and Y.-L. Yu. Accelerated training for matrix-norm regularization: A boosting approach. In *Advances in Neural Information Processing Systems*, pages 2906–2914, 2012.

C. Zhao, X. Wang, and W.-K. Cham. Background subtraction via robust dictionary learning. *EURASIP Journal on Image and Video Processing*, 2011(1):1–12, 2011.

W. Zhong and J.T. Kwok. Gradient descent with proximal average for nonconvex and composite regularization. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 2206–2212, 2014.

Z.A. Zhu and E. Hazan. Variance reduction for faster non-convex optimization. In *Proceedings of the 33nd International Conference on Machine Learning*, pages 699–707, 2016.

Journal of Machine Learning Research 18 (2018) 1-47 Submitted 7/17; Revised 1/18; Published 4/18

Mode-Seeking Clustering and Density Ridge Estimation via Direct Estimation of Density-Derivative-Ratios

Hiroaki Sasaki HSASAKI@IS.NAIST.JP
Graduate School of Information Science
Nara Institute of Science and Technology
Nara, Japan

Takafumi Kanamori KANAMORI@C.TITECH.AC.JP
Department of Mathematical and Computing Science
Tokyo Institute of Technology
Tokyo, Japan
Center for Advanced Intelligence Project
RIKEN
Tokyo, Japan

Aapo Hyvärinen A.HYVARINEN@UCL.AC.UK
Gatsby Computational Neuroscience Unit
University College London
London, United Kingdom
Department of Computer Science
University of Helsinki
Helsinki, Finland
Canadian Institute for Advanced Research

Gang Niu GANG@MS.K.U-TOKYO.AC.JP
Graduate School of Frontier Sciences
The University of Tokyo
Chiba, Japan
Center for Advanced Intelligence Project
RIKEN
Tokyo, Japan

Masashi Sugiyama SUGI@K.U-TOKYO.AC.JP
Center for Advanced Intelligence Project
RIKEN
Tokyo, Japan
Graduate School of Frontier Sciences
The University of Tokyo
Chiba, Japan

Editor: Miguel Á. Carreira-Perpiñán

Abstract

Modes and ridges of the probability density function behind observed data are useful geometric features. Mode-seeking clustering assigns cluster labels by associating data samples with the nearest modes, and estimation of density ridges enables us to find lower-dimensional structures hidden in data. A key technical challenge both in mode-seeking clustering and density ridge estimation is accurate estimation of the ratios of the first- and second-order density derivatives to the density. A naive approach takes a three-step approach of first estimating the data density, then computing its derivatives, and finally taking their ratios. However, this three-step approach can be unreliable because a good density estimator does not necessarily mean a good density derivative estimator, and division by the estimated density could significantly magnify the estimation error. To cope with these problems, we propose a novel estimator for the *density-derivative-ratios*. The proposed estimator does not involve density estimation, but rather *directly* approximates the ratios of density derivatives of any order. Moreover, we establish a convergence rate of the proposed estimator. Based on the proposed estimator, novel methods both for mode-seeking clustering and density ridge estimation are developed, and the respective convergence rates to the mode and ridge of the underlying density are also established. Finally, we experimentally demonstrate that the developed methods significantly outperform existing methods, particularly for relatively high-dimensional data.

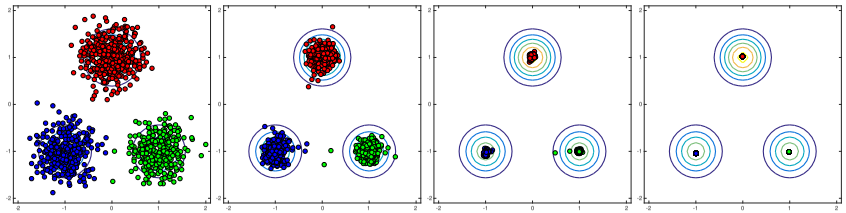
Keywords: Density Derivative, Geometric Feature, Mode-Seeking Clustering, Density Ridge Estimation

1. Introduction

Characterizing the probability density function underlying observed data is a fundamental problem in machine learning. One approach is to consider geometric properties of the density such as modes and ridges. Estimation of such geometric properties is a challenging task, yet offers a variety of applications (Wasserman, 2018).

The *modes* (i.e., local maxima) of probability density functions have received much attention over the years. A motivation of estimating the modes classically appeared in the seminal work on kernel density estimation (Parzen, 1962). More recently, the modes of density functions for random curves have been used in functional data analysis (Gasser et al., 1998). Furthermore, in supervised learning, modal regression associates input variables with the modes of the conditional density function of the output variable, and enables us to simultaneously capture multiple functional relationships between the input and output (Sager and Thisted, 1982; Carreira-Perpiñán, 2000, 2001; Einbeck and Tutz, 2006; Chen et al., 2016a; Sasaki et al., 2016). One of the most natural applications is clustering. *Mean shift clustering* (MS) makes use of the modes of the estimated density function (Fukunaga and Hostetler, 1975; Cheng, 1995; Comaniciu and Meer, 2002): MS initially regards all data samples as candidates for cluster centers, and then iteratively updates them toward the nearest modes of the estimated density by gradient ascent (Fig.1). Finally, the data samples which converge to the same mode are assigned the same cluster label. Unlike standard clustering methods such as k-means clustering (MacQueen, 1967) and mixture-model-based clustering (Melnykov and Maitra, 2010), the notable advantage is that the number of clusters is automatically determined according to the number of detected modes. MS has been applied to a wide range of tasks such as image segmentation (Comaniciu and Meer, 2002; Tao et al., 2007; Wang et al., 2004) and object tracking (Collins, 2003; Comaniciu et al., 2000). (See also a recent review article by Carreira-Perpiñán (2015))

A *ridge* of the probability density function generalizes the notion of the mode. The density ridge is a lower-dimensional hidden structure of the data (Fig.2), and the zero-dimensional ridge can be interpreted as the mode (Genovese et al., 2014). Application of density ridge estimation can be found in a variety of fields such as filamentary structure estimation in cosmology (Chen et al., 2016c), extraction of curvilinear structures (e.g., blood vessels in the eyes) in medical imaging (You et al., 2011), and shape analysis in computer vision (Su et al., 2013) (See Pulkkinen (2015) for more



(a) Iteration=0 (b) Iteration=1 (c) Iteration=3 (d) Iteration=10

Figure 1: Illustration of a mode-seeking process. The contour plot indicates the probability density function that generates the data samples.

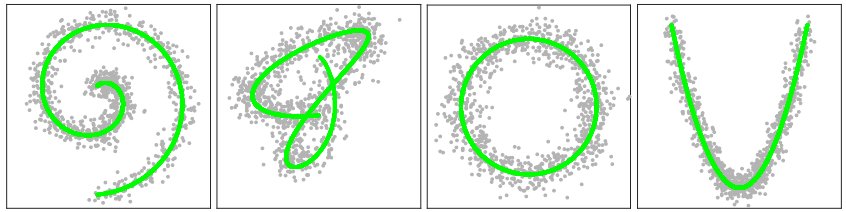


Figure 2: Examples of the density ridges hidden in data. Gray dot points and green curves indicate data samples and density ridges, respectively.

applications). Density ridge estimation is closely related to manifold estimation. When data is assumed to be generated on a lower-dimensional manifold with additive Gaussian noise, density ridge estimation offers a way to circumvent the difficulty of manifold estimation: Genovese et al. (2014) theoretically proved that the density ridges capture the essential properties of such manifolds and estimating the density ridge is substantially easier than estimating the manifold. A practical algorithm called *subspace constrained mean shift* (SCMS) by Ozertem and Erdogmus (2011). SCMS is an extension to MS, but a projected gradient ascent method is performed to find density ridges instead of the gradient ascent method in MS; the gradient vector of the estimated density is projected to the subspace which is orthogonal to the ridge. Such a subspace can be obtained by applying principal component analysis to an estimate of the Hessian matrix of the log-density, which is composed of the ratios of the first- and second-order density derivatives to the density. Along the projected gradient vector, SCMS updates data points toward the ridge of the estimated density until convergence.

For MS, the technical challenge is accurate estimation of the derivatives of the probability density function. To derive practical methods, MS takes a two-step approach, firstly estimating the probability density function and then computing its derivatives (Comaniciu and Meer, 2002, Section 2).[1] However, this approach can be unreliable because a good density estimator does not

1. As reviewed in Section 3.2, practical methods themselves do not perform initial density estimation.

necessarily imply a good density derivative estimator in many practical situations. For example, small random fluctuations in a density estimate can create fake modes and may produce large errors in density-derivative estimation, even if the density estimate is fairly good in terms of density estimation (Genovese et al., 2016, Fig.1). Therefore, testing methods have been proposed to investigate whether the estimated modes are real modes from the underlying data density or fake modes due to the random fluctuations (Godtliebsen et al., 2002; Duong et al., 2008; Genovese et al., 2016). For SCMS, it is even more challenging to estimate the ratios of density derivatives to the density, but SCMS also naively estimates the ratios by adding one more step to the two-step approach in MS: the computed density derivatives are divided by the estimated density. However, such a division could strongly magnify estimation error.

To cope with these problems, we propose a novel estimator of the ratios of density derivatives to the density. In stark contrast with the approaches in MS and SCMS, the key idea is to *directly* estimate the ratios without going through density estimation. Moreover, we theoretically analyze the proposed estimator and establish a convergence rate. The direct approach has been adopted and proved to be useful both empirically and theoretically when estimating the ratio of two probability density functions (Sugiyama et al., 2008; Nguyen et al., 2008; Kanamori et al., 2009, 2012; Sugiyama et al., 2012; Kpotufe, 2017). Here, we follow the direct approach in the context of a different problem and derive an estimator in a substantially different way. Previously, a direct estimator has been proposed for the log-density derivatives (Beran, 1976; Cox, 1985), which are the ratios of first-order density derivatives to the density. On the other hand, the proposed estimator in this paper approximates the ratio of the derivatives of any order to the density, and thus generalizes the previous estimator.

The proposed estimator is first applied to mode-seeking clustering. We derive an update rule for mode-seeking based on a fixed-point algorithm, while inheriting the advantage of MS: the proposed clustering method also does not require the number of clusters to be specified in advance. This is advantageous because clustering is an unsupervised learning problem and tuning the number of clusters is not straightforward in general. Next, based on the mode-seeking clustering, we propose a novel method for density ridge estimation. For both methods, we prove the consistency of the mode and ridge estimators, and establish the convergence rates. Finally, we experimentally demonstrate that our proposed methods outperform MS and SCMS, particularly for high(er)-dimensional data.

This paper is organized as follows: In Section 2, we propose a novel estimator for the ratio of the derivatives of any order to the density, and establish a non-parametric convergence rate. The proposed estimator is applied to develop novel methods for mode-seeking clustering and density ridge estimation in Sections 3 and 4 respectively, and both methods are theoretically analyzed. Section 5 experimentally investigates the performance of the proposed methods for mode-seeking clustering and density ridge estimation. Section 6 concludes this paper. Preliminary results of this paper were presented at ECML/PKDD 2014 (Sasaki et al., 2014) and AISTATS 2017 (Sasaki et al., 2017). However, in addition to combining the results in those conference papers, we have added new theoretical analysis of the proposed estimator, mode-seeking clustering and density ridge estimation methods. From a theoretical stand point, we further improved upon the methods appeared in the conference papers, and performed more experiments in this paper.

2. Direct Estimation of Density-Derivative-Ratios

This section proposes a novel estimator of the ratios of density derivatives to the density and performs theoretical analysis.

2.1 Problem Formulation

Suppose that n i.i.d. samples, which were drawn from a probability distribution on $\mathbb{R}^D$ with density $p(\boldsymbol{x})$, are available:

$$\mathcal{D} := \{\boldsymbol{x}_i = (x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(D)})^\top\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x}).$$

Here, our goal is to estimate the ratio of the $|\boldsymbol{j}|$-th order partial derivative of $p(\boldsymbol{x})$ to $p(\boldsymbol{x})$ from $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^n$,

$$\frac{\partial_{\boldsymbol{j}} p(\boldsymbol{x})}{p(\boldsymbol{x})}, \tag{1}$$

where $\partial_{\boldsymbol{j}} = \frac{\partial^{|\boldsymbol{j}|}}{\partial^{j_1} x^{(1)} \partial^{j_2} x^{(2)} \ldots \partial^{j_D} x^{(D)}}$, $\boldsymbol{j} = (j_1, j_2, \ldots, j_D)^\top$ and $|\boldsymbol{j}| = j_1 + j_2 + \cdots + j_D$ for non-negative integers $j_i = 0, 1, \ldots, |\boldsymbol{j}|$. For instance, when $|\boldsymbol{j}| = 1$ (or $|\boldsymbol{j}| = 2$), $\partial_{\boldsymbol{j}} p(\boldsymbol{x})/p(\boldsymbol{x})$ is a single element of $\nabla p(\boldsymbol{x})/p(\boldsymbol{x})$ (or of $\nabla\nabla p(\boldsymbol{x})/p(\boldsymbol{x})$).

2.2 Least-Squares Density-Derivative-Ratios

Our main idea is to directly fit a model $r_{\boldsymbol{j}}(\boldsymbol{x})$ to $\partial_{\boldsymbol{j}} p(\boldsymbol{x})/p(\boldsymbol{x})$ under the squared-loss:

$$\begin{aligned} J_{\boldsymbol{j}}(r_{\boldsymbol{j}}) &:= \int \left\{ r_{\boldsymbol{j}}(\boldsymbol{x}) - \frac{\partial_{\boldsymbol{j}} p(\boldsymbol{x})}{p(\boldsymbol{x})} \right\}^2 p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\ &= \int \{r_{\boldsymbol{j}}(\boldsymbol{x})\}^2 p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} - 2 \int r_{\boldsymbol{j}}(\boldsymbol{x}) \partial_{\boldsymbol{j}} p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} + \int \left\{ \frac{\partial_{\boldsymbol{j}} p(\boldsymbol{x})}{p(\boldsymbol{x})} \right\}^2 p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}. \end{aligned} \tag{2}$$

The first term on the right-hand side of (2) can be naively estimated from samples and the third term is ignorable, but it seems challenging to estimate the second term because it includes the derivative of the unknown density. However, as in Sasaki et al. (2015), repeatedly applying *integration by parts* allows us to transform the second term as

$$\int r_{\boldsymbol{j}}(\boldsymbol{x}) \left\{ \partial_{\boldsymbol{j}} p(\boldsymbol{x}) \right\} \mathrm{d}\boldsymbol{x} = (-1)^{|\boldsymbol{j}|} \int \left\{ \partial_{\boldsymbol{j}} r_{\boldsymbol{j}}(\boldsymbol{x}) \right\} p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \tag{3}$$

where we assumed that as $|x^{(j)}| \to \infty$ for all j, the product of $\partial_{\boldsymbol{j}_1} r_{\boldsymbol{j}}(\boldsymbol{x})$ and $\partial_{\boldsymbol{j}_2} p(\boldsymbol{x})$ approaches zero for any pairs of $\boldsymbol{j}_1$ and $\boldsymbol{j}_2$ satisfying $|\boldsymbol{j}_1| + |\boldsymbol{j}_2| = |\boldsymbol{j}| - 1$ for $|\boldsymbol{j}_1|, |\boldsymbol{j}_2| = 0, 1, \ldots, |\boldsymbol{j}| - 1$. As a result, the right-hand side of (3) can be easily estimated from samples. Then, an empirical version of (2) is given by

$$\widehat{J}_{\boldsymbol{j}}(r_{\boldsymbol{j}}) := \frac{1}{n} \sum_{i=1}^n \left\{ r_{\boldsymbol{j}}(\boldsymbol{x}_i)^2 - 2(-1)^{|\boldsymbol{j}|} \partial_{\boldsymbol{j}} r_{\boldsymbol{j}}(\boldsymbol{x}_i) \right\} + \text{const.} \tag{4}$$

After adding the regularizer $R(r_{\boldsymbol{j}})$, the estimator is defined as the minimizer of

$$\widehat{r}_{\boldsymbol{j}} := \operatorname*{argmin}_{r_{\boldsymbol{j}}} \left[\widehat{J}_{\boldsymbol{j}}(r_{\boldsymbol{j}}) + \lambda_{\boldsymbol{j}} R(r_{\boldsymbol{j}}) \right], \tag{5}$$

where $\lambda_{\boldsymbol{j}}$ is the regularization parameter.

We call this method the *least-squares density-derivative ratios* (LSDDR). Note that when $|\boldsymbol{j}| = 1$, $J_{\boldsymbol{j}}$ is called the Fisher divergence and has been used for parameter estimation of unnnormalized statistical models (Hyvärinen, 2005), density estimation with the computationally intractable partition function (Sriperumbudur et al., 2017), and direct estimation of log-density derivatives (Beran, 1976; Cox, 1985; Sasaki et al., 2014). Therefore, LSDDR can be regarded as a generalization of such methods to higher-order derivatives.

2.3 Theoretical Analysis of LSDDR

Next, we theoretically analyze LSDDR.

2.3.1 PRELIMINARIES AND NOTATIONS

For a D-dimensional vector $\boldsymbol{x} \in \mathbb{R}^D$, the norm is defined by $\|\boldsymbol{x}\| := \sqrt{\sum_{j=1}^D (x^{(j)})^2}$. For a domain $\mathcal{X} (\subseteq \mathbb{R}^D)$, $C(\mathcal{X})$ denotes the space of all continuous functions on $\mathcal{X}$. Furthermore, we define the L^p space of functions f on $\mathcal{X}$: For $1 \leq p \leq \infty$, $L^p(\mathcal{X}) := \{f : \|f\|_p < \infty\}$ where $\|\cdot\|_p$ is the L^p norm defined by $\|f\|_p := \left(\int |f(\boldsymbol{x})|^p \mathrm{d}\boldsymbol{x} \right)^{\frac{1}{p}}$ with the Lebesgue measure for $1 \leq p < \infty$ and $\|f\|_\infty := \operatorname{ess\,sup}_{\boldsymbol{x} \in \mathcal{X}} |f(\boldsymbol{x})|$. For $f \in L^1(\mathbb{R}^D)$, the *Fourier transform* is defined as

$$f^\wedge(\boldsymbol{\omega}) := \frac{1}{(2\pi)^{D/2}} \int f(\boldsymbol{x}) e^{-i\boldsymbol{\omega}^\top \boldsymbol{x}} \mathrm{d}\boldsymbol{x},$$

where i denotes the imaginary unit.

Let $\mathcal{H}$ be a reproducing kernel Hilbert space (RKHS) over $\mathcal{X}$ uniquely associated with the reproducing kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. The norm and inner product on $\mathcal{H}$ are denoted by $\|\cdot\|_{\mathcal{H}}$ and $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, respectively. k is a real-valued, symmetric and positive definite function and has the reproducing property: For all $\boldsymbol{x} \in \mathcal{X}$ and $f \in \mathcal{H}$, $\langle f, k(\cdot, \boldsymbol{x}) \rangle_{\mathcal{H}} = f(\boldsymbol{x})$. An example of reproducing kernels is the *Gaussian kernel*, $k(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|^2}{2\sigma^2} \right)$ where $\sigma > 0$ is the width parameter. Another example is the *Matérn kernel*, $k(\boldsymbol{x}, \boldsymbol{y}) = \psi(\boldsymbol{x} - \boldsymbol{y}) = \frac{2^{1-s}}{\Gamma(s)} \|\boldsymbol{x} - \boldsymbol{y}\|^{s - D/2} \mathfrak{K}_{D/2-s}(\|\boldsymbol{x} - \boldsymbol{y}\|)$, whose corresponding RKHS $\mathcal{H}$ coincides with the Sobolev space H_2^s with the smoothness parameter $s > D/2$ (Wendland, 2004, Chapter 10):

$$\mathcal{H} = H_2^s := \left\{ f \in L^2(\mathbb{R}^D) \cap C(\mathbb{R}^D) : \int (1 + \|\boldsymbol{\omega}\|^2)^s |f^\wedge(\boldsymbol{\omega})|^2 \mathrm{d}\boldsymbol{\omega} < \infty \right\}.$$

$\Gamma(\cdot)$ denotes the Gamma function, and $\mathfrak{K}_v(\cdot)$ is the modified Bessel function of the second kind of order v.

2.3.2 THE CONVERGENCE RATE OF LSDDR

Here, we derive a rate of convergence for LSDDR under the RKHS norm. To this end, we assume that the true density-derivative-ratio is contained in $\mathcal{H}$:

$$r_{\boldsymbol{j}}^*(\boldsymbol{x}) := \frac{\partial_{\boldsymbol{j}} p(\boldsymbol{x})}{p(\boldsymbol{x})} \in \mathcal{H}.$$

Furthermore, we restrict the search space of $r_{\boldsymbol{j}}$ to $\mathcal{H}$ and express LSDDR with $R(r_{\boldsymbol{j}}) = \|r_{\boldsymbol{j}}\|_{\mathcal{H}}^2$ as

$$\widehat{r}_{\boldsymbol{j}} = \operatorname*{argmin}_{r_{\boldsymbol{j}} \in \mathcal{H}} \left[\widehat{J}_{\boldsymbol{j}}(r_{\boldsymbol{j}}) + \lambda_{\boldsymbol{j}} \|r_{\boldsymbol{j}}\|_{\mathcal{H}}^2 \right]. \tag{6}$$

To establish a convergence rate under the RKHS norm, we make the following assumptions as in Sriperumbudur et al. (2013):

(A) $\mathcal{X}$ is compact.

(B) k is $2|\boldsymbol{j}|$ continuously differentiable.

(C) The following equation holds:

$$\int_{\mathcal{X}} k(\cdot, \boldsymbol{x}) \partial_{\boldsymbol{j}} p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = (-1)^{|\boldsymbol{j}|} \int_{\mathcal{X}} \partial_{\boldsymbol{j}} k(\cdot, \boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}.$$

(D) For all $\boldsymbol{j}$, there exists $\epsilon \geq 1$ subject to

$$\left(\int_{\mathcal{X}} \|k(\cdot, \boldsymbol{x})\|_{\mathcal{H}}^{2\epsilon} p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \right)^{\frac{1}{2\epsilon}} < \infty \quad \text{and} \quad \left(\int_{\mathcal{X}} \|\partial_{\boldsymbol{j}} k(\cdot, \boldsymbol{x})\|_{\mathcal{H}}^{\epsilon} p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \right)^{\frac{1}{\epsilon}} < \infty.$$

Assumption **(A)** makes $\mathcal{H}$ separable (Steinwart and Christmann, 2008, Lemma 4.33) and the separability of $\mathcal{H}$ is required to apply Proposition A.2 in Sriperumbudur et al. (2013). Assumption **(B)** ensures that arbitrary functions in $\mathcal{H}$ are $2|\boldsymbol{j}|$ continuously differentiable (Steinwart and Christmann, 2008, Corollary 4.36). Assumption **(C)** holds under mild assumptions of k and p as in (3). From Assumption **(D)**, $J_{\boldsymbol{j}}(r_{\boldsymbol{j}}) < \infty$ when $\epsilon = 1$. Then, the following theorem establishes the convergence rate under the RKHS norm:

Theorem 1 *Let*

$$C := \int_{\mathcal{X}} k(\cdot, \boldsymbol{x}) \otimes k(\cdot, \boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x},$$

where $\otimes$ denotes the tensor product, be an operator on $\mathcal{H}$. If there exists $\gamma > 0$ such that $r_{\boldsymbol{j}}^*$ is in the range of C^γ (i.e., $r_{\boldsymbol{j}}^* \in \mathcal{R}(C^\gamma)$), then

$$\|\widehat{r}_{\boldsymbol{j}} - r_{\boldsymbol{j}}^*\|_{\mathcal{H}} = O_{\mathrm{P}}\left(n^{-\min\left\{\frac{1}{4}, \frac{\gamma}{2(\gamma+1)}\right\}} \right),$$

with $\epsilon = 2$ and $\lambda_{\boldsymbol{j}} = O\left(n^{-\max\left\{\frac{1}{4}, \frac{1}{2(\gamma+1)}\right\}} \right)$ as $n \to \infty$.

The proof is given in Appendix A. We followed the proof techniques in Sriperumbudur et al. (2013), but adopted them to a different problem: Sriperumbudur et al. (2013) proposed and analyzed a non-parametric estimator for log-densities with the intractable partition functions based on the Fisher divergence, which is a special case of $J_{\boldsymbol{j}}$ at $|\boldsymbol{j}| = 1$. The range space assumption $r_{\boldsymbol{j}}^* \in \mathcal{R}(C^\gamma)$ is closely related to the smoothness of $r_{\boldsymbol{j}}^*$ (Sriperumbudur et al., 2013, Section 4.2): Larger γ implies that $r_{\boldsymbol{j}}^*$ is smoother. As seen in Sections 3.3.3 and 4.3.2, Theorem 1 is particularly useful in the analysis of our mode-seeking clustering and density ridge estimation methods.

Remark 2 *By following Sriperumbudur et al. (2017, Section 4.2), Theorem 1 has some connection to the minimax theory (Tsybakov, 2009) under Sobolev spaces where for any $\alpha > s \geq 0$, the minimax rate is given by*

$$\inf_{\widehat{r}_{\boldsymbol{j},n}} \sup_{r_{\boldsymbol{j}}^* \in H_2^\alpha} \|\widehat{r}_{\boldsymbol{j},n} - r_{\boldsymbol{j}}^*\|_{H_2^s} \asymp n^{-\frac{\alpha - s}{2(\alpha - s) + D}}.$$

$\inf$ *is taken over possible estimators $\widehat{r}_{\boldsymbol{j},n}$, and $a_n \asymp b_n$ means that a_n / b_n has lower- and upper-bounds away from zero and infinity, respectively. To establish a connection to Sobolev spaces, suppose that the Matérn kernel is employed whose corresponding RKHS is a Sobolev space $\mathcal{H} = H_2^s$ with the smoothness parameter $s > D/2$. As proved in Appendix B, when the true density belongs to $L^1(\mathbb{R}^D)$ (i.e., $p \in L^1(\mathbb{R}^D)$), $r_{\boldsymbol{j}}^* \in \mathcal{R}(C^\gamma)$ for $\gamma \geq 1$ implies that $r_{\boldsymbol{j}}^* \in H_2^{\frac{3D}{2} - \frac{1}{2} + \epsilon}$ for arbitrarily small $\epsilon > 0$. Then, the convergence rate $n^{-\frac{1}{4}}$ is minimax optimal under $\mathcal{H} = H_2^{D - \frac{1}{2} + \epsilon}$. Furthermore, this result implies that the dimension effect is veiled through the relative smoothness between two Sobolev spaces ($H_2^{\frac{3D}{2} - \frac{1}{2} + \epsilon}$ and $H_2^{D - \frac{1}{2} + \epsilon}$), and therefore the rate in Theorem 1 is independent of data dimension D. Details are provided in Appendix B.*

2.4 Practical Implementation of LSDDR

Here, we describe practical implementation of LSDDR.

- *A practical version of LSDDR*: The representer theorem (Zhou, 2008, Theorem 2) states that the estimator $\widehat{r}_{\boldsymbol{j}}$ should take the following form:

$$\widehat{r}_{\boldsymbol{j}}(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_{\boldsymbol{j}}^{(i)} k(\boldsymbol{x}, \boldsymbol{x}_i) + \beta_{\boldsymbol{j}}^{(i)} \partial_{\boldsymbol{j}}' k(\boldsymbol{x}, \boldsymbol{x}') \Big|_{\boldsymbol{x}' = \boldsymbol{x}_i} = \sum_{i=1}^{2n} \theta_{\boldsymbol{j}}^{(i)} \psi_{\boldsymbol{j}}^{(i)}(\boldsymbol{x}) = \boldsymbol{\theta}_{\boldsymbol{j}}^\top \boldsymbol{\psi}_{\boldsymbol{j}}(\boldsymbol{x}), \tag{7}$$

  where $\partial_{\boldsymbol{j}}'$ denotes the partial derivative with respect to $\boldsymbol{x}'$,

$$\theta_{\boldsymbol{j}}^{(i)} := \begin{cases} \alpha_{\boldsymbol{j}}^{(i)} & i = 1, \ldots, n, \\ \beta_{\boldsymbol{j}}^{(i-n)} & i = n+1, \ldots, 2n, \end{cases} \qquad \psi_{\boldsymbol{j}}^{(i)}(\boldsymbol{x}) := \begin{cases} k(\boldsymbol{x}, \boldsymbol{x}_i) & i = 1, \ldots, n, \\ \partial_{\boldsymbol{j}}' k(\boldsymbol{x}, \boldsymbol{x}') \Big|_{\boldsymbol{x}' = \boldsymbol{x}_{i-n}} & i = n+1, \ldots, 2n. \end{cases}$$

  To estimate $\boldsymbol{\theta}_{\boldsymbol{j}}$, we substitute (7) into $\widehat{J}_{\boldsymbol{j}}$ in (4). Then, when $R(r_{\boldsymbol{j}}) = \boldsymbol{\theta}_{\boldsymbol{j}}^\top \boldsymbol{\theta}_{\boldsymbol{j}}$, the optimal solution of $\boldsymbol{\theta}_{\boldsymbol{j}}$ can be computed analytically as

$$\widehat{\boldsymbol{\theta}}_{\boldsymbol{j}} := \operatorname*{argmin}_{\boldsymbol{\theta}_{\boldsymbol{j}}} \left[\boldsymbol{\theta}_{\boldsymbol{j}}^\top \widehat{\boldsymbol{G}}_{\boldsymbol{j}} \boldsymbol{\theta}_{\boldsymbol{j}} - 2(-1)^{|\boldsymbol{j}|} \boldsymbol{\theta}_{\boldsymbol{j}}^\top \widehat{\boldsymbol{h}}_{\boldsymbol{j}} + \lambda_{\boldsymbol{j}} \boldsymbol{\theta}_{\boldsymbol{j}}^\top \boldsymbol{\theta}_{\boldsymbol{j}} \right] = (-1)^{|\boldsymbol{j}|} \left(\widehat{\boldsymbol{G}}_{\boldsymbol{j}} + \lambda_{\boldsymbol{j}} \mathbf{I}_{2n} \right)^{-1} \widehat{\boldsymbol{h}}_{\boldsymbol{j}},$$

where $\mathbf{I}_{2n}$ denotes the $2n$ by $2n$ identity matrix,

$$\widehat{\boldsymbol{G}}_{\boldsymbol{j}} := \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\psi}_{\boldsymbol{j}}(\boldsymbol{x}_i)\boldsymbol{\psi}_{\boldsymbol{j}}(\boldsymbol{x}_i)^\top \qquad \text{and} \qquad \widehat{\boldsymbol{h}}_{\boldsymbol{j}} := \frac{1}{n}\sum_{i=1}^{n} \partial_{\boldsymbol{j}}\boldsymbol{\psi}_{\boldsymbol{j}}(\boldsymbol{x}_i).$$

Finally, a practical version of LSDDR is given by

$$\widehat{r}_{\boldsymbol{j}}(\boldsymbol{x}) := \widehat{\boldsymbol{\theta}}_{\boldsymbol{j}}^\top \boldsymbol{\psi}_{\boldsymbol{j}}(\boldsymbol{x}) = \sum_{i=1}^{n} \widehat{\alpha}_{\boldsymbol{j}}^{(i)} k(\boldsymbol{x}, \boldsymbol{x}_i) + \widehat{\beta}_{\boldsymbol{j}}^{(i)} \partial'_{\boldsymbol{j}} k(\boldsymbol{x}, \boldsymbol{x}')\Big|_{\boldsymbol{x}'=\boldsymbol{x}_i}.$$

- *Model selection by cross-validation*: Model selection is a crucial problem in LSDDR. As in standard model selection methods for kernel density estimation (Bowman, 1984; Sheather, 2004), we take a least-squares approach based on (2), and optimize the model parameters (parameters in $k(\cdot,\cdot)$ and the regularization parameter $\lambda_{\boldsymbol{j}}$) by cross-validation as follows:

  1. Divide the samples $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^n$ into T disjoint subsets $\{\mathcal{D}_t\}_{t=1}^T$.
  2. Obtain the estimator $\widehat{r}_{\boldsymbol{j}}^{(t)}(\boldsymbol{x})$ from $\mathcal{D} \setminus \mathcal{D}_t$ (i.e., $\mathcal{D}$ without $\mathcal{D}_t$), and then compute $\widehat{J}_{\boldsymbol{j}}$ from the hold-out samples as

  $$\mathrm{CV}(t) := \frac{1}{|\mathcal{D}_t|}\sum_{\boldsymbol{x}\in\mathcal{D}_t}\left[\left\{\widehat{r}_{\boldsymbol{j}}^{(t)}(\boldsymbol{x})\right\}^2 - 2(-1)^{|\boldsymbol{j}|}\partial_{\boldsymbol{j}}\widehat{r}_{\boldsymbol{j}}^{(t)}(\boldsymbol{x})\right],$$

  where $|\mathcal{D}_t|$ denotes the number of elements in $\mathcal{D}_t$.
  3. Choose the model that minimizes $\frac{1}{T}\sum_{t=1}^T \mathrm{CV}(t)$.

2.5 Notation

In the rest of this paper, we consider LSDDR only for $|\boldsymbol{j}| = 1$ and $|\boldsymbol{j}| = 2$. Therefore, we use more specific notations as follows:

- (Sections 3 and 4) For $|\boldsymbol{j}| = 1$, a first order density-derivative-ratio corresponds to a first order derivative of the log-density, and we express the true derivative as

$$g_j(\boldsymbol{x}) := \frac{\partial_j p(\boldsymbol{x})}{p(\boldsymbol{x})} = \partial_j \log p(\boldsymbol{x}),$$

where $\partial_j := \frac{\partial}{\partial x^{(j)}}$. Then, LSDDR to $g_j(\boldsymbol{x})$ is denoted by

$$\widehat{g}_j(\boldsymbol{x}) := \sum_{i=1}^{2n} \widehat{\theta}_j^{(i)} \psi_j^{(i)}(\boldsymbol{x}) = \sum_{i=1}^{n} \widehat{\alpha}_j^{(i)} k(\boldsymbol{x}, \boldsymbol{x}_i) + \widehat{\beta}_j^{(i)} \partial'_j k(\boldsymbol{x}, \boldsymbol{x}')\Big|_{\boldsymbol{x}'=\boldsymbol{x}_i},$$

where ∂'_j denotes the partial derivative with respect to the j-th coordinate in $\boldsymbol{x}'$, and the subscript j of $\widehat{\theta}_j^{(i)}$ is simplified from $\boldsymbol{j}$ because only one element in $\boldsymbol{j}$ is one and the others are zeros when $|\boldsymbol{j}| = 1$.

- (Section 4) For $|\boldsymbol{j}| = 2$, we express a true second order density-derivative-ratio by $[\boldsymbol{H}(\boldsymbol{x})]_{ij} := \frac{\partial_i\partial_j p(\boldsymbol{x})}{p(\boldsymbol{x})}$ where $[\boldsymbol{H}(\boldsymbol{x})]_{ij}$ denotes the (i,j)-th element of the matrix $\boldsymbol{H}(\boldsymbol{x})$. LSDDR to $[\boldsymbol{H}(\boldsymbol{x})]_{ij}$ is denoted by $[\widehat{\boldsymbol{H}}(\boldsymbol{x})]_{ij}$.

3. Application to Mode-Seeking Clustering

This section applies LSDDR to mode-seeking clustering.

3.1 Problem Formulation for Clustering

Suppose that we are given a collection of data samples $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^n$. The goal of clustering is to assign a cluster label $c_i \in \{1, \ldots, c\}$ to each data sample $\boldsymbol{x}_i$, where c denotes the number of clusters, and is *unknown*.

3.2 Brief Review of Mean Shift Clustering

Mean shift clustering (MS) (Fukunaga and Hostetler, 1975; Cheng, 1995; Comaniciu and Meer, 2002) is a popular clustering method, and has been applied in a wide-range of fields such as image segmentation (Comaniciu and Meer, 2002; Tao et al., 2007; Wang et al., 2004) and object tracking (Collins, 2003; Comaniciu et al., 2000) (see a recent review article by Carreira-Perpiñán (2015)). MS initially regards all data samples as candidates of cluster centers, and updates them toward the nearest modes of the estimated density by gradient ascent. Finally, the same cluster label is assigned to the data samples which converge to the same mode. Unlike standard clustering methods such as *k-means clustering* (MacQueen, 1967), MS automatically determines the number of clusters according to the number of detected modes.

To update data samples, the technical challenge is to accurately estimate the gradient of $p(\boldsymbol{x})$. MS takes a two-step approach: The first step performs kernel density estimation (KDE) as

$$\widehat{p}_{\mathrm{KDE}}(\boldsymbol{x}) := \frac{1}{Z_{n,h}}\sum_{i=1}^{n} K_{\mathrm{KDE}}\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2h^2}\right),$$

where K_{KDE} is a kernel function for KDE, $Z_{n,h}$ is the normalizing constant, and h denotes the bandwidth parameter. Then, the second step computes the partial derivatives of $\widehat{p}_{\mathrm{KDE}}(\boldsymbol{x})$ as

$$\begin{aligned}\partial_j \widehat{p}_{\mathrm{KDE}}(\boldsymbol{x}) &= \frac{1}{h^2 Z_{n,h}}\sum_{i=1}^{n}(x_i^{(j)} - x^{(j)}) G_{\mathrm{KDE}}\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2h^2}\right) \\ &= \frac{1}{h^2 Z_{n,h}}\left\{\sum_{i=1}^{n} G_{\mathrm{KDE}}\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2h^2}\right)\right\}\left\{\frac{\sum_{i=1}^n x_i^{(j)} G_{\mathrm{KDE}}\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2h^2}\right)}{\sum_{i=1}^n G_{\mathrm{KDE}}\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2h^2}\right)} - x^{(j)}\right\},\end{aligned}$$

where $G_{\mathrm{KDE}}(t) = -\frac{\mathrm{d}}{\mathrm{d}t}K_{\mathrm{KDE}}(t)$.

By denoting the τ-th update of a data sample by $\boldsymbol{z}_k^\tau = (z_k^{(\tau,1)}, z_k^{(\tau,2)}, \ldots, z_k^{(\tau,D)})^\top$ where $\boldsymbol{z}_k^0 = \boldsymbol{x}_k$, setting $\partial_j \widehat{p}_{\mathrm{KDE}}(\boldsymbol{x}) = 0$ yields the following fixed-point iteration formula:

$$z_k^{(\tau+1,j)} = \frac{\sum_{i=1}^n x_i^{(j)} G_{\mathrm{KDE}}\left(\frac{\|\boldsymbol{z}_k^\tau-\boldsymbol{x}_i\|^2}{2h^2}\right)}{\sum_{i=1}^n G_{\mathrm{KDE}}\left(\frac{\|\boldsymbol{z}_k^\tau-\boldsymbol{x}_i\|^2}{2h^2}\right)}. \tag{8}$$

Simple calculation shows that (8) can be equivalently expressed as

$$\boldsymbol{z}_k^{\tau+1} = \boldsymbol{z}_k^{\tau} + \frac{h^2 Z_{n,h}}{\sum_{i=1}^n G_{\mathrm{KDE}}\left(\frac{\|\boldsymbol{z}_k^{\tau}-\boldsymbol{x}_i\|^2}{2h^2}\right)} \nabla \widehat{p}_{\mathrm{KDE}}(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{z}_k^{\tau}} = \boldsymbol{z}_k^{\tau} + \widehat{\boldsymbol{m}}_{\mathrm{KDE}}(\boldsymbol{z}_k^{\tau}), \tag{9}$$

where ∇ denotes the vector differential operator with respect to $\boldsymbol{x}$, and $\widehat{\boldsymbol{m}}_{\mathrm{KDE}}(\boldsymbol{z}) = (\widehat{m}^{(1)}_{\mathrm{KDE}}(\boldsymbol{z}), \widehat{m}^{(2)}_{\mathrm{KDE}}(\boldsymbol{z}), \ldots, \widehat{m}^{(D)}_{\mathrm{KDE}}(\boldsymbol{z}))^\top$ is called the *mean shift vector* and defined by

$$\widehat{\boldsymbol{m}}_{\mathrm{KDE}}(\boldsymbol{z}) = \frac{h^2 Z_{n,h}}{\sum_{i=1}^n G_{\mathrm{KDE}}\left(\frac{\|\boldsymbol{z}-\boldsymbol{x}_i\|^2}{2h^2}\right)} \nabla \widehat{p}_{\mathrm{KDE}}(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{z}}. \tag{10}$$

Eq.(9) indicates that MS performs gradient ascent. To speed up MS, acceleration strategies were also developed in Carreira-Perpiñán (2006).

Properties of MS have been theoretically well-investigated (Cheng, 1995; Fashing and Tomasi, 2005; Ghassabeh, 2013; Arias-Castro et al., 2016). For instance, a sequence $\{\boldsymbol{z}_k^{\tau}, \tau = 0, 1, 2, \ldots\}$ generated by MS converges to a mode of $\widehat{p}_{\mathrm{KDE}}(\boldsymbol{x})$ as τ goes infinity (Comaniciu and Meer, 2002; Li et al., 2007; Ghassabeh, 2013); Carreira-Perpiñán (2007) showed that the algorithm of MS is equivalent to the EM algorithm (Dempster et al., 1977) when $K_{\mathrm{KDE}}(t) = \exp(-t)$; Furthermore, Fashing and Tomasi (2005) proved that MS performs a bound optimization. Although MS has good theoretical properties, the two-step approach in gradient estimation seems practically inappropriate because a good-density estimator does not necessarily mean a good-density gradient estimator. A more appropriate way would be to directly estimate the gradient. Following this idea, we apply LSDDR to mode-seeking clustering.

3.3 Least-Squares Log-Density Gradient Clustering

Here, LSDDR is employed to develop a novel mode-seeking clustering method because LSDDR is an estimator of a single element in the log-density gradient when $|\boldsymbol{j}| = 1$. The proposed clustering method is called the *least-squares log-density gradient clustering* (LSLDGC).

3.3.1 FIXED-POINT ITERATION

First, when we estimate the j-th element in $\boldsymbol{g}(\boldsymbol{x}) = \nabla \log p(\boldsymbol{x})$, the form of the kernel function is restricted as

$$k(\boldsymbol{x}, \boldsymbol{x}_i) = \phi\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right),$$

where σ_j denotes a bandwidth parameter, and ϕ is a non-negative, monotonically non-increasing, convex and differentiable function. For example, when $\phi(t) = \exp(-t)$, $k(\boldsymbol{x}, \boldsymbol{x}_i)$ is the Gaussian kernel. Under the restriction, LSDDR can be rewritten as

$$\widehat{g}_j(\boldsymbol{x}) = \sum_{i=1}^n \left[\widehat{\alpha}_j^{(i)} \phi\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right) + \widetilde{\beta}_j^{(i)} \frac{x_i^{(j)} - x^{(j)}}{\sigma_j^2} \varphi\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right)\right], \tag{11}$$

where $\widetilde{\beta}_j^{(i)} = -\widehat{\beta}_j^{(i)}$ and $\varphi(t) = -\frac{\mathrm{d}}{\mathrm{d}t}\phi(t)$.

For our mode-seeking clustering method, we derive a fixed-point iteration similarly to MS. When $\sum_{i=1}^n \widetilde{\beta}_j^{(i)} \varphi\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right) \neq 0$, (11) can be expanded as

$$\begin{aligned}\widehat{g}_j(\boldsymbol{x}) &= \sum_{i=1}^n \left[\widehat{\alpha}_j^{(i)} \phi\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right) + \frac{\widetilde{\beta}_j^{(i)} x_i^{(j)}}{\sigma_j^2} \varphi\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right)\right] - \frac{x^{(j)}}{\sigma_j^2} \sum_{i=1}^n \widetilde{\beta}_j^{(i)} \varphi\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{\sigma_j^2}\right) \\ &= \frac{1}{\sigma_j^2} \sum_{i=1}^n \widetilde{\beta}_j^{(i)} \varphi\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right) \left[\frac{\sum_{i=1}^n \left[\sigma_j^2 \widehat{\alpha}_j^{(i)} \phi\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right) + \widetilde{\beta}_j^{(i)} x_i^{(j)} \varphi\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right)\right]}{\sum_{i=1}^n \widetilde{\beta}_j^{(i)} \varphi\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right)} - x^{(j)}\right].\end{aligned}$$

As in MS, setting $\widehat{g}_j(\boldsymbol{x}) = 0$ yields the following update formula:

$$z_k^{(\tau+1,j)} = \frac{\sum_{i=1}^n \left[\sigma_j^2 \widehat{\alpha}_j^{(i)} \phi\left(\frac{\|\boldsymbol{z}_k^{\tau}-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right) + \widetilde{\beta}_j^{(i)} x_i^{(j)} \varphi\left(\frac{\|\boldsymbol{z}_k^{\tau}-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right)\right]}{\sum_{i=1}^n \widetilde{\beta}_j^{(i)} \varphi\left(\frac{\|\boldsymbol{z}_k^{\tau}-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right)}, \tag{12}$$

where $\boldsymbol{z}_k^{\tau}$ denotes the τ-th update of a data sample initialized by $\boldsymbol{x}_k$. Eq.(12) can be also equivalently expressed as

$$z^{(\tau+1,j)} = z^{(\tau,j)} + \frac{\sigma_j^2}{\sum_{i=1}^n \widetilde{\beta}_j^{(i)} \varphi\left(\frac{\|\boldsymbol{z}^{\tau}-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right)} \widehat{g}_j(\boldsymbol{z}^{\tau}) = z^{(\tau,j)} + \widehat{m}^{(j)}(\boldsymbol{z}^{\tau}), \tag{13}$$

where

$$\widehat{m}^{(j)}(\boldsymbol{z}) := \frac{\sigma_j^2}{\sum_{i=1}^n \widetilde{\beta}_j^{(i)} \varphi\left(\frac{\|\boldsymbol{z}-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right)} \widehat{g}_j(\boldsymbol{z}). \tag{14}$$

When $\widehat{\alpha}_j^{(i)} = 0$ and $\widetilde{\beta}_j^{(i)} = 1/n$, (12) is reduced to the MS update formula (8). Thus, LSLDGC includes MS as a special case.

The form of (12) motivates us to develop a coordinate-wise update rule. From $j = 1$ to $j = D$, we iteratively update one coordinate at a time by simply modifying (12) as

$$z_k^{(\tau+1,j)} = \frac{\sum_{i=1}^n \left[\sigma_j^2 \widehat{\alpha}_j^{(i)} \phi\left(\frac{\|\tilde{\boldsymbol{z}}_k^{\tau}-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right) + \widetilde{\beta}_j^{(i)} x_i^{(j)} \varphi\left(\frac{\|\tilde{\boldsymbol{z}}_k^{\tau}-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right)\right]}{\sum_{i=1}^n \widetilde{\beta}_j^{(i)} \varphi\left(\frac{\|\tilde{\boldsymbol{z}}_k^{\tau}-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right)}, \tag{15}$$

where

$$\tilde{\boldsymbol{z}}_k^{\tau} = (z_k^{(\tau+1,1)}, \ldots, z_k^{(\tau+1,j-1)}, z_k^{(\tau,j)}, z_k^{(\tau,j+1)}, \ldots, z_k^{(\tau,D)})^\top.$$

Note that the $(j-1)$-th and j-th elements in $\tilde{\boldsymbol{z}}_k^{\tau}$ are different in terms of τ. As shown below, this coordinate-wise update rule has a nice theoretical property.

3.3.2 SUFFICIENT CONDITIONS FOR MONOTONIC HILL-CLIMBING

LSLDGC updates data samples towards the modes like hill-climbing. Here, we show sufficient conditions for monotonic hill-climbing, i.e., LSLDGC makes data samples never climbing-down. The challenge in this analysis is that unlike MS, we cannot know the estimated density, and thus it is not straightforward to investigate this property for LSLDGC. To overcome this challenge, we employ *path integral*[2] (Strang, 1991): For the vector field $\boldsymbol{g}(\boldsymbol{x}) = \nabla \log p(\boldsymbol{x})$ and a differentiable curve $\boldsymbol{\gamma}(t)$, $t \in [0, s]$ connecting $\boldsymbol{x}$ and $\boldsymbol{y}$, i.e., $\boldsymbol{\gamma}(0) = \boldsymbol{y}$, $\boldsymbol{\gamma}(s) = \boldsymbol{x}$, the standard formula of path integral is given by

$$D_{\boldsymbol{g}}[\boldsymbol{x}|\boldsymbol{y}] := \int_0^s \mathrm{I} < \boldsymbol{g}(\boldsymbol{\gamma}(t)), \dot{\boldsymbol{\gamma}}(t) > \mathrm{d}t = \log p(\boldsymbol{x}) - \log p(\boldsymbol{y}), \tag{16}$$

where $\dot{\boldsymbol{\gamma}}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\gamma}(t)$ and $\mathrm{I} < \cdot, \cdot >$ denotes the inner product. The notable property of path integral is that the integral is independent of any choice of a path, and determined only by the two points, $\boldsymbol{y}$ and $\boldsymbol{x}$, as shown in the most right-hand side of (16). In this analysis, we use the following path along with one coordinate at a time repeatedly:

$$\begin{aligned}\boldsymbol{y} = (y^{(1)}, y^{(2)}, y^{(3)}, \ldots, y^{(D)}) &\to (x^{(1)}, y^{(2)}, y^{(3)}, \ldots, y^{(D)}) \to (x^{(1)}, x^{(2)}, y^{(3)}, \ldots, y^{(D)}) \\ &\to (x^{(1)}, x^{(2)}, x^{(3)}, \ldots, y^{(D)}) \to \cdots \to (x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(D)}) = \boldsymbol{x}. \end{aligned} \tag{17}$$

By substituting our gradient estimate $\widehat{\boldsymbol{g}}(\boldsymbol{x})$ into the middle part of (16) under the path (17),

$$\widehat{D}_{\widehat{\boldsymbol{g}}}[\boldsymbol{x}|\boldsymbol{y}] := \int_0^s \mathrm{I} < \widehat{\boldsymbol{g}}(\boldsymbol{\gamma}(t)), \dot{\boldsymbol{\gamma}}(t) > \mathrm{d}t = \sum_{j=1}^{D} \int_{y^{(j)}}^{x^{(j)}} \widehat{g}_j(x^{(1)}, x^{(2)}, \ldots, z^{(j)}, \ldots, y^{(D)}) \mathrm{d}z^{(j)}. \tag{18}$$

From (16), $\widehat{D}_{\widehat{\boldsymbol{g}}}[\boldsymbol{x}|\boldsymbol{y}]$ can be regarded as an estimator of $\log p(\boldsymbol{x}) - \log p(\boldsymbol{y})$ when we fix the curve that connects $\boldsymbol{x}$ and $\boldsymbol{y}$. Thus, $\widehat{D}_{\widehat{\boldsymbol{g}}}[\boldsymbol{z}_k^{\tau+1}|\boldsymbol{z}_k^{\tau}] \geq 0$ for all τ implies that the data samples updated by LSLDGC never climb down. The following theorem provides some sufficient conditions:

Theorem 3 *Suppose that ϕ is a non-negative, monotonically non-increasing, convex and differentiable function. Then, if $\widehat{\alpha}_j^{(i)} = 0$ and $\widetilde{\beta}_j^{(i)} \geq 0$, under the coordinate-wise update rule* (15) *and path* (17)*,*

$$\widehat{D}_{\widehat{\boldsymbol{g}}}[\boldsymbol{z}_k^{\tau+1}|\boldsymbol{z}_k^{\tau}] \geq 0.$$

The proof is deferred to Appendix C.

Remark 4 *Theorem 3 shows sufficient conditions that LSLDGC with the coordinate-wise update rule* (15) *makes data samples monotonically hill-climb towards the modes. However, without satisfying the conditions, we empirically observed that most of data samples monotonically converge to modes. Therefore, we conjecture that some milder conditions exist, and do not apply all sufficient conditions in practice. Practical implementation is described in Section 3.4.*

2. Path integral is also called *line integral*.

Remark 5 *For another update rule* (12)*, sufficient conditions for monotonic hill-climbing were not established as in Theorem 3. However, Theorem 7 implies that accurate mode-seeking is possible for both update rules as long as $\widehat{D}_{\widehat{\boldsymbol{g}}}[\boldsymbol{z}_k^{\tau+1}|\boldsymbol{z}_k^{\tau}]$ is kept non-negative for all τ. Therefore, in practice, whenever $\widehat{D}_{\widehat{\boldsymbol{g}}}[\boldsymbol{z}_k^{\tau+1}|\boldsymbol{z}_k^{\tau}]$ is negative, we perform standard gradient ascent. The details are given in Section 3.4.*

Remark 6 *Sufficient conditions for monotonic hill-climbing have been established in MS (Comaniciu and Meer, 2002; Li et al., 2007; Ghassabeh, 2013). The main difference is that we obtain the difference of two log-density estimates from a gradient estimate, while previous work directly begins with density estimation based on KDE. Thus, the proof is substantially different.*

Theorem 3 holds under the path (17). However, the following theorem states that as n increases, $\widehat{D}_{\widehat{\boldsymbol{g}}}[\boldsymbol{x}|\boldsymbol{y}]$ approaches $D_{\boldsymbol{g}}[\boldsymbol{x}|\boldsymbol{y}]$, which is independent of the choice of a path:

Theorem 7 *Suppose that both $\boldsymbol{g}$ and $\widehat{\boldsymbol{g}}$ are finite on the path* (17) *and the assumptions in Theorem 1 hold. Then, for arbitrary $\boldsymbol{x}$ and $\boldsymbol{y}$,*

$$\left| D_{\boldsymbol{g}}[\boldsymbol{x}|\boldsymbol{y}] - \widehat{D}_{\widehat{\boldsymbol{g}}}[\boldsymbol{x}|\boldsymbol{y}] \right| \leq \|\boldsymbol{g} - \widehat{\boldsymbol{g}}\|_\infty \|\boldsymbol{x} - \boldsymbol{y}\|_1 \leq O_{\mathrm{P}}\left(n^{-\min\left\{\frac{1}{4}, \frac{\gamma}{2(\gamma+1)}\right\}} \right),$$

where $\|\cdot\|_1$ denotes the ℓ_1 norm.

The proof is given in Appendix D.

Remark 8 *Theorem 7 shows*

$$|D_{\boldsymbol{g}}[\boldsymbol{z}_k^{\tau}|\boldsymbol{z}_k^{\tau+1}] - \widehat{D}_{\widehat{\boldsymbol{g}}}[\boldsymbol{z}_k^{\tau}|\boldsymbol{z}_k^{\tau+1}]| \leq \|\boldsymbol{g} - \widehat{\boldsymbol{g}}\|_\infty \|\boldsymbol{z}_k^{\tau} - \boldsymbol{z}_k^{\tau+1}\|_1. \tag{19}$$

From (19)*, the non-negativity of $\widehat{D}_{\widehat{\boldsymbol{g}}}[\boldsymbol{z}_k^{\tau}|\boldsymbol{z}_k^{\tau+1}]$ implies that $D_{\boldsymbol{g}}[\boldsymbol{z}_k^{\tau}|\boldsymbol{z}_k^{\tau+1}]$ is also non-negative when n is sufficiently large. Thus, Theorem 7 ensures that accurate mode-seeking is possible by both update rules* (12) *and* (15)*.*

3.3.3 THE CONVERGENCE RATE TO THE TRUE MODE SET

First, we define the set of the true mode points as

$$\mathcal{M} := \{\boldsymbol{\mu} \ : \ \boldsymbol{g}(\boldsymbol{\mu}) = \boldsymbol{0}, \nabla \boldsymbol{g}(\boldsymbol{\mu}) \prec \boldsymbol{O}\}, \tag{20}$$

where $\nabla \boldsymbol{g}(\boldsymbol{\mu})$ is the Hessian matrix of the log-density at a mode point $\boldsymbol{\mu}$, and $\nabla \boldsymbol{g}(\boldsymbol{\mu}) \prec \boldsymbol{O}$ means that $\nabla \boldsymbol{g}(\boldsymbol{\mu})$ is (strictly) negative definite. The set of the estimated mode points is also denoted by $\widehat{\mathcal{M}}$. Our goal is to establish the convergence rate between $\mathcal{M}$ and $\widehat{\mathcal{M}}$ under the Hausdorff distance:

$$\mathrm{Haus}(\mathcal{A}, \mathcal{B}) := \max\left(\sup_{\boldsymbol{x}\in\mathcal{A}} \inf_{\boldsymbol{y}\in\mathcal{B}} \|\boldsymbol{x} - \boldsymbol{y}\|, \sup_{\boldsymbol{y}\in\mathcal{B}} \inf_{\boldsymbol{x}\in\mathcal{A}} \|\boldsymbol{x} - \boldsymbol{y}\| \right), \tag{21}$$

where $\mathcal{A}$ and $\mathcal{B}$ denote two sets.

The following theorem establishes the convergence rate of $\mathrm{Haus}(\widehat{\mathcal{M}}, \mathcal{M})$.

Theorem 9 *Suppose that the assumptions in Theorem 1 hold. Further assume that each mode point $\boldsymbol{\mu} \in \mathcal{M}$ is approximated by a unique estimated mode point $\widehat{\boldsymbol{\mu}} \in \widehat{\mathcal{M}}$. Then, with high probability,*

$$\mathrm{Haus}(\widehat{\mathcal{M}}, \mathcal{M}) = O_{\mathrm{P}}\left(n^{-\min\left\{\frac{1}{4}, \frac{\gamma}{2(\gamma+1)}\right\}}\right). \tag{22}$$

The proof can be seen in Appendix E.

Remark 10 *Chen et al. (2016b, Theorem 1) established the following convergence rate based on KDE: With the asymptotically optimal bandwidth $h = O\left(n^{-\frac{1}{D+6}}\right)$,*

$$\mathit{Haus}(\widehat{\mathcal{M}}_{KDE}, \mathcal{M}) = O_{\mathrm{P}}\left(n^{-\frac{2}{D+6}}\right), \tag{23}$$

where $\widehat{\mathcal{M}}_{KDE}$ denotes the set of mode points based on KDE. Eq.(23) shows that the convergence rate of $\mathit{Haus}(\widehat{\mathcal{M}}_{KDE}, \mathcal{M})$ depends on data dimension D, although direct comparison to our result is not straightforward due to the different assumptions in both analyses.

3.4 Practical Implementation of LSLDGC

Here, we describe details of practical implementation of LSLDGC.

- *Sufficient conditions in Theorem 3*: The conditions, $\widehat{\alpha}_j^{(i)} = 0$ and $\widetilde{\beta}_j^{(i)} \left(= -\widehat{\beta}_j^{(i)}\right) \geq 0$, ensure that $\widehat{D}_{\widehat{\boldsymbol{g}}}[\boldsymbol{z}_k^{\tau+1} | \boldsymbol{z}_k^{\tau}] \geq 0$. Here, we set $\widehat{\alpha}_j^{(i)} = 0$ for all i and j, and the coordinate-wise update rule (15) is simplified as

$$z_k^{(\tau+1,j)} = \frac{\sum_{i=1}^n \widetilde{\beta}_j^{(i)} x_i^{(j)} \varphi\left(\frac{\|\tilde{\boldsymbol{z}}_k^{\tau} - \boldsymbol{x}_i\|^2}{2\sigma_j^2}\right)}{\sum_{i=1}^n \widetilde{\beta}_j^{(i)} \varphi\left(\frac{\|\tilde{\boldsymbol{z}}_k^{\tau} - \boldsymbol{x}_i\|^2}{2\sigma_j^2}\right)}.$$

  The same simplification is applied to the update rule (12) as well. This significantly reduces the computational costs in LSDDR because $\alpha_j^{(i)}$ do not need to be estimated. On the other hand, to satisfy $\widetilde{\beta}_j^{(i)} \geq 0$, we have to solve a constrained optimization problem, which tends to be time-consuming. Therefore, the unconstrained optimization problem is solved as in Section 3.4, but as a remedy we perform gradient ascent whenever $\widehat{D}_{\widehat{\boldsymbol{g}}}[\boldsymbol{z}_k^{\tau+1} | \boldsymbol{z}_k^{\tau}] < 0$. Details of the gradient ascent are given below.

- *Stability in the mode-seeking process*: The derivation of (12) indicates that the mode-seeking (hill-climbing) process in LSLDGC can be unstable when $f_j(\boldsymbol{z}_k^{\tau}) := \sum_{i=1}^n \widetilde{\beta}_j^{(i)} \varphi\left(\frac{\|\boldsymbol{z}_k^{\tau} - \boldsymbol{x}_i\|}{2\sigma_j^2}\right)$ is close to zero. To cope with this problem, we simply perform gradient ascent when $f_j(\boldsymbol{z}_k^{\tau})$ is close to zero.

- *Gradient ascent*: Whenever $\widehat{D}_{\widehat{\boldsymbol{g}}}[\boldsymbol{z}_k^{\tau+1} | \boldsymbol{z}_k^{\tau}] < 0$ or $\exists j$, $f_j(\boldsymbol{z}_k^{\tau}) \approx 0$, we perform the following gradient ascent:

$$\boldsymbol{z}_k^{\tau+1} = \boldsymbol{z}_k^{\tau} + \eta \widehat{\boldsymbol{g}}(\boldsymbol{z}_k^{\tau}), \tag{24}$$

  where the step size parameter η is selected so that $\widehat{D}_{\widehat{\boldsymbol{g}}}[\boldsymbol{z}_k^{\tau} + \eta \widehat{\boldsymbol{g}}(\boldsymbol{z}_k^{\tau}) | \boldsymbol{z}_k^{\tau}]$ is maximized.

- *Choice of the kernel function*: Throughout the paper, we use the Gaussian kernel:

$$k(\boldsymbol{x}, \boldsymbol{x}_i) = \phi\left(\frac{\|\boldsymbol{x} - \boldsymbol{x}_i\|^2}{2\sigma_j^2}\right) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}_i\|^2}{2\sigma_j^2}\right).$$

  The Gaussian kernel satisfies the conditions of ϕ in Theorem 3, and is a universal kernel associated with which RKHS covers a wide range of functions (Micchelli et al., 2006).

- *Decreasing the computation costs*: After the simplification above, LSDDR requires to compute the inverse of a $2n$ by $2n$ matrix, which is computationally costly to large n. To decrease the computation costs, we reduce the number of center points as $\phi\left(\frac{\|\boldsymbol{x} - \boldsymbol{c}_i\|^2}{2\sigma_j^2}\right)$ and $\varphi\left(\frac{\|\boldsymbol{x} - \boldsymbol{c}_i\|^2}{2\sigma_j^2}\right)$ where $\{\boldsymbol{c}_i\}_{i=1}^b$ is a randomly chosen subset of $\{\boldsymbol{x}_i\}_{i=1}^n$. As a result, the coefficients can be represented as $\widetilde{\boldsymbol{\beta}}_j = (\widetilde{\beta}_j^{(1)}, \widetilde{\beta}_j^{(2)}, \ldots, \widetilde{\beta}_j^{(b)})^{\top}$. Appendix F shows that this significantly decreases the computation cost without scarifying clustering performance. In this paper, we fix the number of centers at $b = \min(n, 100)$ as long as we do not specify it.

The mode-seeking algorithm in LSLDGC is summarized in Figs.3 and 4.[3]

4. Application to Density Ridge Estimation

This section applies LSDDR to density ridge estimation and develops a novel method.

4.1 Problem Formulation for Density Ridge Estimation

For a positive integer d such that $d < D$, the goal is to estimate from a collection of data samples $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^n$ the d-dimensional *density ridge*, which is defined as a collection of points satisfying

$$\mathcal{R} := \{\boldsymbol{x} \in \mathbb{R}^D \mid \|\mathbf{V}(\boldsymbol{x})\mathbf{V}(\boldsymbol{x})^{\top} \boldsymbol{g}(\boldsymbol{x})\| = 0, \eta_{d+1}(\boldsymbol{x}) < 0\}, \tag{25}$$

where $\boldsymbol{g}(\boldsymbol{x}) = \nabla \log p(\boldsymbol{x})$, $\mathbf{V}(\boldsymbol{x}) = (\boldsymbol{v}_{d+1}, \ldots, \boldsymbol{v}_D)$, and $\boldsymbol{v}_i$ is the eigenvector associated with the eigenvalue $\eta_i(\boldsymbol{x})$ of the Hessian matrix of the logarithm of the probability density function, $\nabla\nabla \log p(\boldsymbol{x})$. We assume that the eigenvalues are sorted in descending order such that $\eta_1(\boldsymbol{x}) \geq \eta_2(\boldsymbol{x}) \geq \cdots \geq \eta_D(\boldsymbol{x})$.

Here, we defined the density ridge in terms of the logarithm of the probability density function because our practical algorithm is proposed based on the logarithm. While the density ridge has been previously defined without the logarithm (Eberly, 1996; Ozertem and Erdogmus, 2011; Genovese et al., 2014; Chen et al., 2015b), both definitions offer the same density ridge.

4.2 Brief Review of Subspace Constrained Mean Shift

A practical algorithm for density ridge estimation called *subspace constrained mean shift* (SCMS) was proposed by Ozertem and Erdogmus (2011). SCMS extends MS: SCMS performs projected gradient ascent on the subspace orthogonal to the density ridge, while MS updates data points by

3. A MATLAB package of LSLDGC is available at `https://sites.google.com/site/hworksites/home/software/lsldg`.

Input: $\{\boldsymbol{x}_i\}_{i=1}^n$.

$\left\{\{\widetilde{\boldsymbol{\beta}}_j\}_{j=1}^D, \{\boldsymbol{c}_i\}_{i=1}^b\right\} \leftarrow \text{LSDDR1}(\{\boldsymbol{x}_i\}_{i=1}^n)$;

for $k = 1$ to n **do**
  $\tau \leftarrow 0$;
  $\boldsymbol{z}_k^\tau \leftarrow \boldsymbol{x}_k$;
  repeat
    $\{\boldsymbol{z}_k^{\tau+1}, \{f_j\}_{j=1}^D\} \leftarrow \text{ModeSeeking}(\{\widetilde{\boldsymbol{\beta}}_j\}_{j=1}^D, \{\boldsymbol{c}_i\}_{i=1}^b, \boldsymbol{z}_k^\tau)$
    $\widehat{D} \leftarrow \widehat{D}_{\widehat{\boldsymbol{g}}}[\boldsymbol{z}_k^{\tau+1}|\boldsymbol{z}_k^\tau]$;

    if $\widehat{D} < 0$ or $\exists j,\ |f_j| \approx 0$ **then**
      $\boldsymbol{z}_k^{\tau+1} \leftarrow \boldsymbol{z}_k^\tau + \eta\widehat{\boldsymbol{g}}(\boldsymbol{z}_k^\tau)$;
      $\widehat{D} \leftarrow \widehat{D}_{\widehat{\boldsymbol{g}}}[\boldsymbol{z}_k^{\tau+1}|\boldsymbol{z}_k^\tau]$;
    end if
    $\tau \leftarrow \tau + 1$;
  until
  $\boldsymbol{z}_k \leftarrow \boldsymbol{z}_k^\tau$;
end for

Outputs: $\{\boldsymbol{z}_i\}_{i=1}^n$.

Figure 3: The mode-seeking algorithm in LSLDGC. LSDDR1($\{\boldsymbol{x}_i\}_{i=1}^n$) denotes the LSDDR estimator for the first-order density-derivative-ratios from data samples $\{\boldsymbol{x}_i\}_{i=1}^n$, and $\{\widetilde{\boldsymbol{\beta}}_j\}_{j=1}^D$ and $\{\boldsymbol{c}_i\}_{i=1}^b$ are the coefficients and (sub-sampled) centers, respectively. ModeSeeking($\{\widetilde{\boldsymbol{\beta}}_j\}_{j=1}^D, \{\boldsymbol{c}_i\}_{i=1}^b, \boldsymbol{z}_k^\tau$) is a single step mode-seeking process whose details are given in Fig.4. The update of $\boldsymbol{z}_k^\tau$ terminates when either $\widehat{D}$ or $\|\boldsymbol{z}_k^{\tau+1} - \boldsymbol{z}_k^\tau\|$ is less than a small positive constant.

Input: $\{\widetilde{\boldsymbol{\beta}}_j\}_{j=1}^D, \{\boldsymbol{c}_i\}_{i=1}^b, \boldsymbol{z}_k^\tau$

$\boldsymbol{z}_k^{\tau+1} \leftarrow \boldsymbol{z}_k^\tau + \widehat{\boldsymbol{m}}(\boldsymbol{z}_k^\tau)$;

$\{f_j\}_{j=1}^D \leftarrow \left\{\widetilde{\boldsymbol{\beta}}_j^\top \boldsymbol{\varphi}_j(\boldsymbol{z}_k^\tau)\right\}_{j=1}^D$;

Outputs: $\boldsymbol{z}_k^{\tau+1}, \{f_j\}_{j=1}^D$.

Input: $\{\widetilde{\boldsymbol{\beta}}_j\}_{j=1}^D, \{\boldsymbol{c}_i\}_{i=1}^b, \boldsymbol{z}_k^\tau$
$\tilde{\boldsymbol{z}} \leftarrow \boldsymbol{z}_k^\tau$;
for $j \in \{1, \ldots, D\}$ **do**
  $\tilde{z}^{(j)} \leftarrow z_k^{(\tau,j)} + m^{(j)}(\tilde{\boldsymbol{z}})$;
  $f_j \leftarrow \widetilde{\boldsymbol{\beta}}_j^\top \boldsymbol{\varphi}_j(\tilde{\boldsymbol{z}})$;
end for
$\boldsymbol{z}_k^{\tau+1} \leftarrow \tilde{\boldsymbol{z}}$;
Outputs: $\boldsymbol{z}_k^{\tau+1}, \{f_j\}_{j=1}^D$.

Figure 4: Two mode-seeking algorithms in LSLDGC. The left figure uses the update rule (12), while the right one is based on the coordinate-wise update rule (15). $\boldsymbol{\varphi}_j(\boldsymbol{z}) = (\varphi_j^{(1)}(\boldsymbol{z}), \varphi_j^{(2)}(\boldsymbol{z}), \ldots, \varphi_j^{(b)}(\boldsymbol{z}))^\top$ where $\varphi_j^{(i)}(\boldsymbol{z}) = \varphi\left(\frac{\|\boldsymbol{z}-\boldsymbol{c}_i\|^2}{2\sigma_j^2}\right)$.

gradient ascent. SCMS obtains such a subspace as the span of the eigenvectors of the negative Hessian matrix of the log-density, which is called the *inverse local-covariance matrix* (Ozertem and Erdogmus, 2011):

$$\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}) := -\nabla\nabla \log p(\boldsymbol{x}) = -\frac{\nabla\nabla p(\boldsymbol{x})}{p(\boldsymbol{x})} + \frac{\nabla p(\boldsymbol{x})\nabla p(\boldsymbol{x})^\top}{p(\boldsymbol{x})^2} = -\boldsymbol{H}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{g}(\boldsymbol{x})^\top. \quad (26)$$

An advantage of employing the log-density is discussed in the context of manifold estimation in Genovese et al. (2014): Theorem 7 in Genovese et al. (2014) states that when D-dimensional data is assumed to be generated on a d-dimensional manifold with D-dimensional Gaussian noise, the density ridge is close to the lower-dimensional manifold in the sense of the Hausdorff distance, and thus can be a surrogate for the manifold. This surrogate property holds in an $O(1)$ neighborhood of the manifold for the log-density, while the theorem holds in an $O(\sigma_n)$ neighborhood of the manifold for the (non-log) density, where σ_n is the standard deviation of the Gaussian noise. Furthermore, when $p(\boldsymbol{x})$ is Gaussian, (26) reduces to the inverse of the covariance matrix. This allows us to intuitively understand that SCMS finds the subspace by PCA to the non-stationary covariance matrix at a location $\boldsymbol{x}$ around the ridge.

In practice, SCMS substitutes $\widehat{p}_{\text{KDE}}(\boldsymbol{x})$ into (26):

$$\widehat{\boldsymbol{\Sigma}}_{\text{KDE}}^{-1}(\boldsymbol{x}) := -\frac{\nabla\nabla\widehat{p}_{\text{KDE}}(\boldsymbol{x})}{\widehat{p}_{\text{KDE}}(\boldsymbol{x})} + \frac{\nabla\widehat{p}_{\text{KDE}}(\boldsymbol{x})\nabla\widehat{p}_{\text{KDE}}(\boldsymbol{x})^\top}{\widehat{p}_{\text{KDE}}(\boldsymbol{x})^2}.$$

Then, SCMS obtains the orthogonal projector to the subspace as $\widehat{\boldsymbol{L}}_{\text{KDE}}(\boldsymbol{x}) = \widehat{\mathbf{V}}_{\text{KDE}}(\boldsymbol{x})\widehat{\mathbf{V}}_{\text{KDE}}(\boldsymbol{x})^\top$, where $\widehat{\mathbf{V}}_{\text{KDE}}(\boldsymbol{x}) \in \mathbb{R}^{D\times(D-d)}$ consists of the $D-d$ eigenvectors associated with the $D-d$ largest eigenvalues of $\widehat{\boldsymbol{\Sigma}}_{\text{KDE}}^{-1}(\boldsymbol{x})$. Then, the update rule of SCMS is given by

$$\boldsymbol{z}^{\tau+1} = \boldsymbol{z}^\tau + \widehat{\boldsymbol{L}}_{\text{KDE}}(\boldsymbol{z}^\tau)\widehat{\boldsymbol{m}}_{\text{KDE}}(\boldsymbol{z}^\tau), \quad (27)$$

where $\boldsymbol{z}^\tau$ denotes the τ-th update of an arbitrarily initialized point and $\widehat{\boldsymbol{m}}_{\text{KDE}}(\boldsymbol{x})$ is the mean shift vector defined in (10). Eq.(27) is repeatedly applied until convergence. The monotonic hill-climbing property for SCMS is proved in Ghassabeh et al. (2013).

One of the key challenges in SCMS is to accurately estimate $\boldsymbol{\Sigma}^{-1}(\boldsymbol{x})$ in (26). SCMS takes a three-step approach, i.e., estimate $p(\boldsymbol{x})$ by KDE, compute its derivatives, and plug them into $\boldsymbol{\Sigma}^{-1}(\boldsymbol{x})$. However, this approach can perform poorly because of the same reason as MS, i.e., a good density estimator does not necessarily mean a good density derivative estimator. In addition, division by the estimated density could further magnify the estimation error for density derivatives. To cope with this problem, we employ LSDDR for direct estimation of density-derivative-ratios in $\boldsymbol{\Sigma}^{-1}(\boldsymbol{x})$ without going through density estimation and division, and propose a novel method for density ridge estimation.

4.3 Least-Squares Density Ridge Finder

Based on LSDDR, we develop a novel density ridge finder called the *least-squares density ridge finder* (LSDRF), which extends LSLDGC for density ridge estimation.

Input: $\{\boldsymbol{x}_i\}_{i=1}^n, \{\boldsymbol{y}_k\}_{k=1}^{n'}$.

$\left\{\{\widetilde{\boldsymbol{\beta}}_j\}_{j=1}^D, \{\boldsymbol{c}_i\}_{i=1}^b\right\} \leftarrow \mathrm{LSDDR1}(\{\boldsymbol{x}_i\}_{i=1}^n)$;
$\left\{\{\widetilde{\boldsymbol{\theta}}_j\}_{j=1}^{D(D+1)/2}, \{\boldsymbol{c}'_i\}_{i=1}^b\right\} \leftarrow \mathrm{LSDDR2}(\{\boldsymbol{x}_i\}_{i=1}^n)$;

for $k = 1$ to n' **do**
  $\tau \leftarrow 0$;
  $\boldsymbol{z}_k^\tau \leftarrow \boldsymbol{y}_k$;
  repeat
    $\{\widehat{\boldsymbol{g}}(\boldsymbol{z}_k^\tau), \widehat{\boldsymbol{m}}(\boldsymbol{z}_k^\tau)\} \leftarrow \mathrm{ComputeGrad}(\{\widetilde{\boldsymbol{\beta}}_j\}_{j=1}^D, \{\boldsymbol{c}_i\}_{i=1}^b, \boldsymbol{z}_k^\tau)$;
    $\widehat{\boldsymbol{L}}(\boldsymbol{z}_k^\tau) \leftarrow \mathrm{ComputeProjector}(\widehat{\boldsymbol{g}}(\boldsymbol{z}_k^\tau), \{\widehat{\boldsymbol{\theta}}_j\}_{j=1}^{D(D+1)/2}, \{\boldsymbol{c}'_i\}_{i=1}^b, \boldsymbol{z}_k^\tau)$;

    $\boldsymbol{z}_k^{\tau+1} \leftarrow \boldsymbol{z}_k^\tau + \widehat{\boldsymbol{L}}(\boldsymbol{z}_k^\tau)\widehat{\boldsymbol{m}}(\boldsymbol{z}_k^\tau)$;
    $\{f_j\}_{j=1}^D \leftarrow \left\{\widetilde{\boldsymbol{\beta}}_j^\top \boldsymbol{\varphi}_j(\boldsymbol{z}_k^\tau)\right\}_{j=1}^D$;

    $\widehat{D} \leftarrow \widehat{D}_{\widehat{\boldsymbol{g}}}[\boldsymbol{z}_k^{\tau+1} | \boldsymbol{z}_k^\tau]$;

    if $\widehat{D} < 0$ or $\exists j,\ |f_j| \approx 0$ **then**
      $\boldsymbol{z}_k^{\tau+1} \leftarrow \boldsymbol{z}_k^\tau + \eta\widehat{\boldsymbol{L}}(\boldsymbol{z}_k^\tau)\widehat{\boldsymbol{g}}(\boldsymbol{z}_k^\tau)$;
      $\widehat{D} \leftarrow \widehat{D}_{\widehat{\boldsymbol{g}}}[\boldsymbol{z}_k^{\tau+1} | \boldsymbol{z}_k^\tau]$;
    end if
    $\tau \leftarrow \tau + 1$;
  until
  $\boldsymbol{z}_k \leftarrow \boldsymbol{z}_k^\tau$;
end for

Outputs: $\{\boldsymbol{z}_k\}_{k=1}^{n'}$.

Figure 5: The algorithm of LSDRF. $\mathrm{LSDDR2}(\{\boldsymbol{x}_i\}_{i=1}^n)$ denotes the LSDDR estimator for the second-order density-derivative-ratios, and $\{\widehat{\boldsymbol{\theta}}_j\}_{j=1}^{D(D+1)/2}$ are the corresponding coefficient vectors. $\{\boldsymbol{y}_k\}_{k=1}^{n'}$ are initial points to approximate the density ridge. $\mathrm{ComputeGrad}(\{\widetilde{\boldsymbol{\beta}}_j\}_{j=1}^D, \{\boldsymbol{c}_i\}_{i=1}^b, \boldsymbol{z}_k^\tau)$ computes the estimated log-density gradient $\widehat{\boldsymbol{g}}(\boldsymbol{z}_k^\tau)$ and $\widehat{\boldsymbol{m}}(\boldsymbol{z}_k^\tau)$ in (14), while $\mathrm{ComputeProjector}(\widehat{\boldsymbol{g}}(\boldsymbol{z}_k^\tau), \{\widehat{\boldsymbol{\theta}}_j\}_{j=1}^{D(D+1)/2}, \{\boldsymbol{c}'_i\}_{i=1}^b, \boldsymbol{z}_k^\tau)$ computes the subspace projector $\widehat{\boldsymbol{L}}(\boldsymbol{z}_k^\tau)$. The update of $\boldsymbol{z}_k^\tau$ terminates when either $\widehat{D}$ or $\|\boldsymbol{z}_k^{\tau+1} - \boldsymbol{z}_k^\tau\|$ is less than a small positive constant. The other notations follow Figs.3 and 4.

4.3.1 ALGORITHM OF LSDRF

The algorithm of LSDRF essentially follows the same line as SCMS, which performs projected gradient ascent. By employing LSDDR, we obtain an estimate of $\boldsymbol{\Sigma}^{-1}(\boldsymbol{x})$ as

$$\widehat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{x}) := -\widehat{\boldsymbol{H}}(\boldsymbol{x}) + \widehat{\boldsymbol{g}}(\boldsymbol{x})\widehat{\boldsymbol{g}}^\top(\boldsymbol{x}), \tag{28}$$

where we recall that $\widehat{g}_j(\boldsymbol{x})$ and $[\widehat{\boldsymbol{H}}(\boldsymbol{x})]_{ij}$ are LSDDR to $\partial_j p(\boldsymbol{x})/p(\boldsymbol{x})$ and $\partial_i\partial_j p(\boldsymbol{x})/p(\boldsymbol{x})$, respectively. Then, we obtain the orthogonal projector to the subspace as $\widehat{\boldsymbol{L}}(\boldsymbol{x}) = \widehat{\mathbf{V}}(\boldsymbol{x})\widehat{\mathbf{V}}^\top(\boldsymbol{x})$ where $\widehat{\mathbf{V}}(\boldsymbol{x})$ consists of the $D-d$ eigenvectors associated with the $D-d$ largest eigenvalues of $\widehat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{x})$. By replacing $\widehat{\boldsymbol{L}}_{\mathrm{KDE}}(\boldsymbol{x})$ and $\widehat{\boldsymbol{m}}_{\mathrm{KDE}}(\boldsymbol{x})$ in (27) with $\widehat{\boldsymbol{L}}(\boldsymbol{x})$ and $\widehat{\boldsymbol{m}}(\boldsymbol{x})$ respectively, the following update rule for LSDRF is obtained by

$$\boldsymbol{z}^{\tau+1} = \boldsymbol{z}^\tau + \widehat{\boldsymbol{L}}(\boldsymbol{z}^\tau)\widehat{\boldsymbol{m}}(\boldsymbol{z}^\tau), \tag{29}$$

where $\widehat{\boldsymbol{m}}(\boldsymbol{x}) = (\widehat{m}^{(1)}(\boldsymbol{x}), \widehat{m}^{(2)}(\boldsymbol{x}), \ldots, \widehat{m}^{(D)}(\boldsymbol{x}))$ is used in LSLDGC for mode-seeking whose definition is given in (14).

The implementation techniques of LSLDGC in Section 3.4 are inherited, but LSDRF performs projected gradient ascent instead of the gradient ascent: Whenever $\widehat{D}_{\widehat{\boldsymbol{g}}}[\boldsymbol{z}^{\tau+1} | \boldsymbol{z}^\tau] < 0$ or $\exists j, f_j(\boldsymbol{z}^{(\tau)}) \approx 0$, we perform the projected gradient ascent as

$$\boldsymbol{z}^{\tau+1} = \boldsymbol{z}^\tau + \eta\widehat{\boldsymbol{L}}(\boldsymbol{z}^\tau)\widehat{\boldsymbol{g}}(\boldsymbol{z}^\tau). \tag{30}$$

The step size parameter η is selected so that $\widehat{D}_{\widehat{\boldsymbol{g}}}[\boldsymbol{z}^\tau + \eta\widehat{\boldsymbol{L}}(\boldsymbol{z}^\tau)\widehat{\boldsymbol{g}}(\boldsymbol{z}^\tau) | \boldsymbol{z}^\tau]$ is maximized. The algorithm of LSDRF is summarized in Fig.5.[4] The algorithm is essentially the same as LSLDGC based on the update rule (12) (Figs. 3 and 4), where we only replace (13) and (24) in LSLDGC with (29) and (30) in LSDRF, respectively. Unlike clustering, for density ridge estimation, the starting points $\boldsymbol{z}^{\tau=0}$ are arbitrary, but in this paper, we set them at data samples $\boldsymbol{x}_i$ because data samples are fairly good starting points.

4.3.2 THE CONVERGENCE RATE TO THE TRUE RIDGE

Here, we establish the convergence rate to understand how the estimated ridge approaches to the true ridge as n increases. Based on LSDDR, the estimated ridge is defined as

$$\widehat{\mathcal{R}} := \{\boldsymbol{x} \in \mathbb{R}^D \mid \|\widehat{\mathbf{V}}(\boldsymbol{x})\widehat{\mathbf{V}}(\boldsymbol{x})^\top \widehat{\boldsymbol{g}}(\boldsymbol{x})\| = 0, \widehat{\eta}_{d+1}(\boldsymbol{x}) < 0\},$$

where $\widehat{\eta}_i(\boldsymbol{x})$ denotes the i-th largest eigenvalue of $-\widehat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{x})$.

In our analysis, we make the following assumptions:

(A0) Kernel boundedness: $k(\boldsymbol{x}, \boldsymbol{x}')$ and $\partial_j\partial'_j k(\boldsymbol{x}, \boldsymbol{x}')$ for all j are uniformly bounded, where ∂'_j denotes the partial derivative with respect to the j-th coordinate in $\boldsymbol{x}'$.

(A1) Differentiability and boundedness: Let $B_D(\boldsymbol{x}, \delta)$ be the D-dimensional ball of radius $\delta > 0$ centered at $\boldsymbol{x}$ and let $\mathcal{R} \oplus \delta := \cup_{\boldsymbol{x} \in \mathcal{R}} B_D(\boldsymbol{x}, \delta)$. For all $\boldsymbol{x} \in \mathcal{R} \oplus \delta$, the $|\boldsymbol{j}|$-th order derivatives of $\log p(\boldsymbol{x})$ for $|\boldsymbol{j}| = 0, 1, 2, 3$ exist and are bounded.

4. A MATLAB package of LSDRF is available at `https://sites.google.com/site/hworksites/home/software/lsdrf`.

(A2) Eigengap: Assume that there exists $\kappa > 0$ and δ such that for all $\boldsymbol{x} \in \mathcal{R} \oplus \delta$, $\eta_{d+1}(\boldsymbol{x}) < -\kappa$ and $\eta_d(\boldsymbol{x}) - \eta_{d+1}(\boldsymbol{x}) > \kappa$, where $\eta_i(\boldsymbol{x})$ denotes the i-th eigenvalue of $\nabla\nabla \log p(\boldsymbol{x})$.

(A3) Path smoothness: For each $\boldsymbol{x} \in \mathcal{R} \oplus \delta$,

$$\|\boldsymbol{L}^{\perp}(\boldsymbol{x})\boldsymbol{g}(\boldsymbol{x})\| \cdot \|\boldsymbol{\Sigma}^{-1\prime}(\boldsymbol{x})\|_{\max} < \frac{\kappa^2}{2D^{3/2}},$$

where $\boldsymbol{L}^{\perp}(\boldsymbol{x}) := \mathbf{I}_D - \boldsymbol{V}(\boldsymbol{x})\boldsymbol{V}(\boldsymbol{x})^{\top}$, $\boldsymbol{\Sigma}^{-1\prime}(\boldsymbol{x}) := \nabla \mathrm{vec}\left(\boldsymbol{\Sigma}^{-1}(\boldsymbol{x})\right)$, $\mathrm{vec}(\cdot)$ denotes vectorization of matrices by concatenating the columns, and $\|\boldsymbol{A}\|_{\max} := \max_{i,j} |[\boldsymbol{A}]_{ij}|$. The (i,j)-th element in $\nabla \mathrm{vec}\left(\boldsymbol{\Sigma}^{-1}(\boldsymbol{x})\right)$ $(\in \mathbb{R}^{D^2 \times D})$ is given by $\partial_j [\mathrm{vec}\left(\boldsymbol{\Sigma}^{-1}(\boldsymbol{x})\right)]_i$.

Assumptions (A2) and (A3) are a straightforward modification of the assumptions in Genovese et al. (2014) from the (non-log) density to the log-density. Assumption (A2) indicates that the density ridge has a sharp and curvilinear shape in the subspace orthogonal to the ridge. Assumption (A3) indicates that $\|\boldsymbol{L}^{\perp}(\boldsymbol{x})\boldsymbol{g}(\boldsymbol{x})\|$ and $\|\boldsymbol{\Sigma}^{-1\prime}(\boldsymbol{x})\|_{\max}$ are both bounded. Since $\boldsymbol{L}^{\perp}(\boldsymbol{x})$ is orthogonal to $\boldsymbol{V}(\boldsymbol{x})\boldsymbol{V}(\boldsymbol{x})^{\top}$ for all $\boldsymbol{x}$, the boundedness of $\|\boldsymbol{L}^{\perp}(\boldsymbol{x})\boldsymbol{g}(\boldsymbol{x})\|$ implies that the gradient $\boldsymbol{g}(\boldsymbol{x})$ is not too steep in the orthogonal subspace. The boundedness of $\|\boldsymbol{\Sigma}^{-1\prime}(\boldsymbol{x})\|_{\max}$ means that the third-order derivative is bounded and thus the subspace direction does not abruptly change, which implies that the (projected) gradient ascent path cannot be too wiggly (Genovese et al., 2014, Section 2.2). Note that Assumptions (A1)-(A3) are only valid in the neighborhood around the ridge.

Let

$$\epsilon' := \max_j \|g_j(\boldsymbol{x}) - \widehat{g}_j(\boldsymbol{x})\|_{\infty}, \qquad \epsilon'' := \max_{ij} \|[\boldsymbol{\Sigma}^{-1}(\boldsymbol{x})]_{ij} - [\widehat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{x})]_{ij}\|_{\infty},$$

$$\epsilon''' := \max_{ij} \|[\boldsymbol{\Sigma}^{-1\prime}(\boldsymbol{x})]_{ij} - [\widehat{\boldsymbol{\Sigma}}^{-1\prime}(\boldsymbol{x})]_{ij}\|_{\infty}.$$

To establish the convergence rate, we rely on two lemmas. The first lemma is a simple modification of Theorem 4 in Genovese et al. (2014) to the log-density from the (non-log) density, and we use it without proof. The lemma states that if ϵ', ϵ'' and ϵ''' are sufficiently small, then the true and estimated ridges are close to each other:

Lemma 11 *Suppose that (A1)-(A3) hold. Let $\psi := \max\{\epsilon', \epsilon''\}$ and $\Psi := \max\{\epsilon', \epsilon'', \epsilon'''\}$. When Ψ is sufficiently small, the following statements hold:*

(i) Conditions (A2) and (A3) hold for $\widehat{\boldsymbol{g}}$, $\widehat{\boldsymbol{\Sigma}}^{-1}$ and $\widehat{\boldsymbol{\Sigma}}^{-1\prime}$.

(ii) $\mathrm{Haus}(\mathcal{R}, \widehat{\mathcal{R}})$ *is bounded as*

$$\mathrm{Haus}(\mathcal{R}, \widehat{\mathcal{R}}) = O(\psi). \tag{31}$$

The next lemma characterizes the convergence rates of ϵ', ϵ'' and ϵ''' when we employ LSDDR:

Lemma 12 *Suppose that the assumptions in Theorem 1 and (A0) hold. When LSDDR is applied for density-derivative-ratio estimation,*

$$\epsilon' = O_{\mathrm{P}}\left(n^{-\min\left\{\frac{1}{4}, \frac{\gamma}{2(\gamma+1)}\right\}}\right), \tag{32}$$

$$\epsilon'' = O_{\mathrm{P}}\left(n^{-\min\left\{\frac{1}{4}, \frac{\gamma}{2(\gamma+1)}\right\}}\right), \tag{33}$$

$$\epsilon''' = O_{\mathrm{P}}\left(n^{-\min\left\{\frac{1}{4}, \frac{\gamma}{2(\gamma+1)}\right\}}\right). \tag{34}$$

The proof is given in Appendix G.

Combining Lemma 11 with Lemma 12 yields the following theorem:

Theorem 13 *Suppose that the assumptions in Theorem 1 and (A0)-(A3) hold. Then,*

$$\mathrm{Haus}(\mathcal{R}, \widehat{\mathcal{R}}) = O_{\mathrm{P}}\left(n^{-\min\left\{\frac{1}{4}, \frac{\gamma}{2(\gamma+1)}\right\}}\right). \tag{35}$$

Proof Lemma 12 ensures that $\psi = O_{\mathrm{P}}\left(n^{-\min\left\{\frac{1}{4}, \frac{\gamma}{2(\gamma+1)}\right\}}\right)$. This completes the proof from (31). ■

Remark 14 *Genovese et al. (2014, Eq.(1)) established the following convergence rate based on KDE:*

$$\mathrm{Haus}(\mathcal{R}, \widehat{\mathcal{R}}_{KDE}) = O_{\mathrm{P}}\left(\left(\frac{\log n}{n}\right)^{\frac{2}{D+8}}\right), \tag{36}$$

where $\widehat{\mathcal{R}}_{KDE}$ denotes the estimated ridge by KDE. Comparison to our result is difficult, but the main difference is that the rate in (36) *explicitly depends on data dimension D.*

5. Numerical Illustration on Mode-Seeking Clustering and Density Ridge Estimation

This section experimentally illustrates the performance of the proposed methods for mode-seeking clustering and density ridge estimation on a variety of datasets.

5.1 Illustration on Clustering

First, we illustrate the performance of LSLDGC both on artificial and benchmark datasets.

5.1.1 ARTIFICIAL DATASETS: LSLDGC VS MS

Here, we compare the performance of LSLDGC to MS with two different bandwidth selection methods:

- **LSLDGC**: LSLDGC based on the update rule (12). The width parameter σ_j in the Gaussian kernel and regularization parameter λ_j were selected by cross-validation as in Section 2.4. We selected ten candidates of σ_j and λ_j from $c_\sigma \times \sigma_{\mathrm{med}}^{(j)}$ $(0.5 \leq c_\sigma \leq 5)$ and 10^m $(-3 \leq m \leq 0)$, respectively where $\sigma_{\mathrm{med}}^{(j)}$ is the median value of $|x_i^{(j)} - x_k^{(j)}|$ with respect to i and k.
- **LSLDGC$_{\mathbf{CW}}$**: LSLDGC based on the coordinate-wise update rule (15). The same cross-validation was performed as above.
- **MS$_{\mathbf{LS}}$**: The bandwidth parameter h was cross-validated based on the standard *integrated squared error*. We selected ten candidates of h from $10^l \times h_{\mathrm{med}}$ $(-1.5 \leq l \leq 0)$ where h_{med} is the median value of $|x_i^{(j)} - x_k^{(j)}|$ with respect to i, j and k.

- **$\mathrm{MS}_{\mathrm{NR}}$**: The bandwidth parameter h was determined by

$$\bar{S}_n \left(\frac{4}{D+4}\right)^{\frac{1}{D+6}} n^{-\frac{1}{d+6}},$$

where $\bar{S}_n = \frac{1}{nD}\sum_{j=1}^{D}\sum_{i=1}^{n}(x_i^{(j)} - \bar{x}^{(j)})^2$ and $\bar{x}^{(j)} = \frac{1}{n}\sum_{i=1}^{n} x_i^{(j)}$. This bandwidth parameter was used in Chen et al. (2016b) and a slight modification of the normal reference rule (Silverman, 1986).

First, we generated three kinds of two-dimensional data as follows:

(a) **Three Gaussian blobs** (Fig.6(a)): Each data sample was drawn from a mixture of three Gaussians with means $(0,1)^\top$, $(-1,-1)^\top$ and $(1,-1)^\top$, and covariance matrices $0.1\mathbf{I}_2$. The mixing coefficients were $0.4, 0.3, 0.3$, respectively.

(b) **Two curves** (Fig.6(d)): Two curves are generated as $(x^{(1)}, x^{(2)}) = (\cos(\pi t^{(1)}), \sin(\pi t^{(1)}))^\top$ and $(x^{(1)}, x^{(2)}) = (-\cos(\pi t^{(2)}) + 1, -\sin(\pi t^{(2)}))^\top$ where $t^{(1)}$ and $t^{(2)}$ are independently drawn from the Gaussian density with mean 0.5 and standard deviation 0.15. Then, Gaussian noise with covariance matrix $0.1\mathbf{I}_2$ was added to these curves. The numbers of data samples for both curves were approximately same.

(c) **Two curves & a Gaussian blob** (Fig.6(g)): Data samples from the Gaussian density with mean 0 and standard deviation 0.1 were added to the two curves similarly generated as in (b). The number of samples for the two curves was same, and for the Gaussian blob, we set the number at $n/3$ approximately.

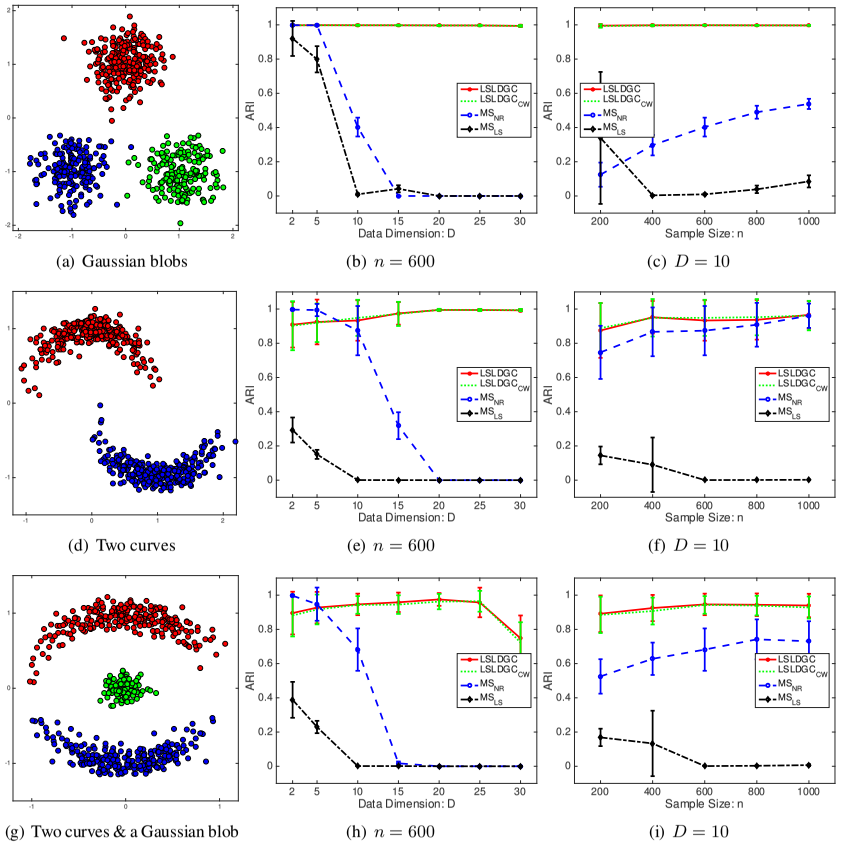
When higher-dimensional data were generated, we simply appended Gaussian variables with mean 0 and standard deviation 0.1 to the two-dimensional data. Clustering performance was measured by the adjusted Rand index (ARI) (Hubert and Arabie, 1985): ARI takes a value less than or equal to one, a larger value indicates a better clustering result, and when a clustering result is perfect, the ARI value equals to one.

Fig.6(b,e,h) clearly indicates the advantage of our clustering methods over MS: Both LSLDGC and $\mathrm{LSLDGC}_{\mathrm{CW}}$ significantly outperform $\mathrm{MS}_{\mathrm{LS}}$ and $\mathrm{MS}_{\mathrm{NR}}$ particularly for higher-dimensional data. When the dimensionality of data is low, $\mathrm{MS}_{\mathrm{NR}}$ performs well to all kinds of datasets. However, the ARI values of both $\mathrm{MS}_{\mathrm{LS}}$ and $\mathrm{MS}_{\mathrm{NR}}$ quickly approach zero as the dimensionality of data increases. These unsatisfactory results seem to be due to the fact that the bandwidth selection in KDE is more difficult for high(er)-dimensional data. Thus, our direct approach would be more suitable particularly for high(er)-dimensional data.

Both LSLDGC and $\mathrm{LSLDGC}_{\mathrm{CW}}$ keep the ARI values high on a wide range of sample sizes (Fig.6(c,f,i)). The performance of $\mathrm{MS}_{\mathrm{NR}}$ is improved as n increases. However, $\mathrm{MS}_{\mathrm{LS}}$ performs rather worse for large(r) datasets. The least-squares cross-validation often suggests small bandwidth parameters for large(r) datasets, which make the estimated density unsmooth. Thus, the estimated density can include a lot of spurious modes with small peaks even if it was good in terms of density estimation. This also supports that our direct estimation is a more appropriate approach.

5.1.2 BENCHMARK DATASETS

Next, we investigate the performance of LSLDGC over the following benchmark datasets:



(a) Gaussian blobs (b) $n = 600$ (c) $D = 10$

(d) Two curves (e) $n = 600$ (f) $D = 10$

(g) Two curves & a Gaussian blob (h) $n = 600$ (i) $D = 10$

Figure 6: Clustering performance on artificial data. Each point and error bar denote the average and standard deviation of ARI over 50 runs, respectively.

Table 1: The average and standard deviation of ARI values over 50 runs. A larger value means a better result. Numbers in the parentheses are standard deviations. The best and comparable methods judged by the unpaired t-test at the significance level 1% are described in boldface.

Banknote $(D, n, c) = (4, 100, 2)$					
LSLDGC	$\text{LSLDGC}_{\text{CW}}$	MS_{LS}	MS_{NR}	SC	KM
0.165(0.059)	**0.169(0.055)**	0.036(0.014)	**0.167(0.147)**	0.054(0.064)	0.039(0.051)

Accelerometry $(D, n, c) = (5, 300, 3)$					
LSLDGC	$\text{LSLDGC}_{\text{CW}}$	MS_{LS}	MS_{NR}	SC	KM
0.628(0.058)	**0.624(0.065)**	0.029(0.007)	0.500(0.041)	0.226(0.271)	0.499(0.023)

Olive oil $(D, n, c) = (8, 200, 9)$					
LSLDGC	$\text{LSLDGC}_{\text{CW}}$	MS_{LS}	MS_{NR}	SC	KM
0.717(0.081)	**0.728(0.062)**	0.020(0.019)	**0.756(0.078)**	0.552(0.060)	0.618(0.063)

Vowel $(D, n, c) = (10, 110, 11)$					
LSLDGC	$\text{LSLDGC}_{\text{CW}}$	MS_{LS}	MS_{NR}	SC	KM
0.147(0.037)	0.139(0.032)	0.017(0.010)	0.133(0.026)	0.145(0.027)	**0.180(0.027)**

Sat-image $(D, n, c) = (36, 120, 6)$					
LSLDGC	$\text{LSLDGC}_{\text{CW}}$	MS_{LS}	MS_{NR}	SC	KM
0.427(0.072)	**0.422(0.073)**	0.000(0.000)	0.343(0.063)	**0.418(0.056)**	**0.434(0.052)**

Speech $(D, n, c) = (50, 400, 2)$					
LSLDGC	$\text{LSLDGC}_{\text{CW}}$	MS_{LS}	MS_{NR}	SC	KM
0.146(0.063)	**0.147(0.054)**	0.000(0.000)	0.000(0.000)	0.004(0.004)	0.002(0.004)

- *Banknote* ($D = 4, n = 100, \text{ and } c = 2$) (Bache and Lichman, 2013)[5]: This dataset consists of four-dimensional features from 400 by 400 images for genuine and forged banknote-like specimens. The features were extracted by wavelet transformation. We randomly chose 50 samples from each of the two classes.

- *Accelerometry* ($D = 5, n = 300, \text{ and } c = 3$)[6]: The ALKAN dataset contains 3-axis (i.e., x-, y-, and z-axes) accelerometric data. During the data collection, subjects were instructed to perform walking, running, and standing up. After segmenting each data stream into windows, five orientation-invariant-features were computed from each window (Sugiyama et al., 2014). We randomly chose 100 samples from each of the three classes.

5. https://archive.ics.uci.edu/ml/datasets/banknote+authentication#
6. http://alkan.mns.kyutech.ac.jp/web/data.html

- *Olive oil* ($D = 8, n = 200, \text{ and } c = 9$) (Forina et al., 1983). This dataset was obtained from the R software.[7] The dataset includes eight chemical measurements on different specimen of olive oil produced in nine regions in Italy. We randomly chose 200 samples.

- *Vowel* ($D = 10, n = 110, \text{ and } c = 11$) (Turney, 1993; Bache and Lichman, 2013)[8]: This consists utterance data for eleven vowels of British English. Each utterance is expressed by a ten-dimensional vector. We randomly chose 10 samples from each of the eleven classes.

- *Sat-image* ($D = 36, n = 120, \text{ and } c = 6$) (Bache and Lichman, 2013)[9]: The dataset contains the multi-spectral values of pixels in 3×3 neighborhoods in a satellite image with six classes. We randomly chose 20 samples from each of the six classes.

- *Speech* ($D = 50, n = 400, \text{ and } c = 2$). An in-house speech dataset (Sugiyama et al., 2014), which contains short utterance samples recorded from 2 male subjects speaking in French with sampling rate 44.1kHz. 50-dimensional line spectral frequencies vectors (Kain and Macon, 1998) were computed from each utterance sample. We randomly chose 200 samples from each of the two classes.

As preprocessing, each data sample was standardized by the sample mean and standard deviation in coordinate-wise manner. For comparison, we applied *k-means clustering* (KM) (MacQueen, 1967) and *spectral clustering* (SC) (Ng et al., 2001; Shi and Malik, 2000) to the same datasets. Since KM and SC require to input the number of clusters, we set it at the correct number.

As seen in the illustration on artificial data, when the dimensionality of data is low, the performance of LSLDGC, $\text{LSLDGC}_{\text{CW}}$ and MS_{NR} is comparable, but LSLDGC and $\text{LSLDGC}_{\text{CW}}$ significantly work better than MS_{NR} to higher-dimensional datasets (sat-image and speech datasets). KM and SC have prior information about the number of clusters. Nonetheless, the performance of LSLDGC and $\text{LSLDGC}_{\text{CW}}$ are often better than KM and SC.

From the results of both the artificial and benchmark datasets, we conclude that LSLDGC and $\text{LSLDGC}_{\text{CW}}$ are advantageous to relatively high-dimensional data.

5.2 Illustration on Density Ridge Estimation

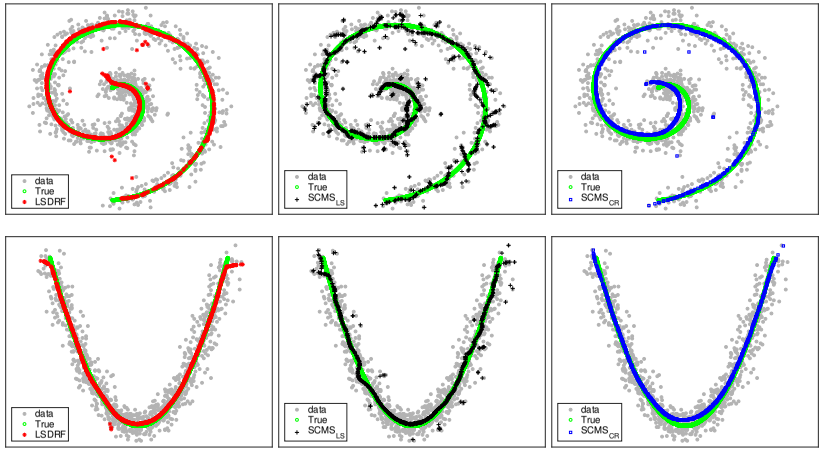
Next, we illustrate the performance of LSDRF, and compare LSDRF with SCMS both on artificial and standard benchmark datasets.

5.2.1 ARTIFICIAL DATA: LSDRF VS SCMS

The performance of LSDRF is compared to SCMS with two different bandwidth selection methods:

- **LSDRF**: When estimating $g_j(\boldsymbol{x})$, we selected ten candidates of the width parameter in the Gaussian kernel and the regularization parameter from $10^l \times \sigma_{\text{med}}^{(j)}$ ($-0.3 \leq l \leq 1$) and 10^m ($-4 \leq m \leq 0$), respectively. When estimating $[\boldsymbol{H}(\boldsymbol{x})]_{ij}$, ten candidates of the width parameter in the Gaussian kernel were selected from $10^l \times \sqrt{\sigma_{\text{med}}^{(i)} \sigma_{\text{med}}^{(j)}}$ ($-0.3 \leq l \leq 1$). For the regularization parameter, we used the same candidates as in $g_j(\boldsymbol{x})$.

7. https://artax.karlin.mff.cuni.cz/r-help/library/pdfCluster/html/oliveoil.html
8. https://archive.ics.uci.edu/ml/datasets/Connectionist+Bench+(Vowel+Recognition+-+Deterding+Data)
9. https://archive.ics.uci.edu/ml/datasets/Statlog+(Landsat+Satellite)



(a) LSDRF (b) SCMS_{LS} (c) SCMS_{CR}

Figure 7: Comparison of the two estimated ridges by LSDRF, SCMS_{LS} and SCMS_{CR}.

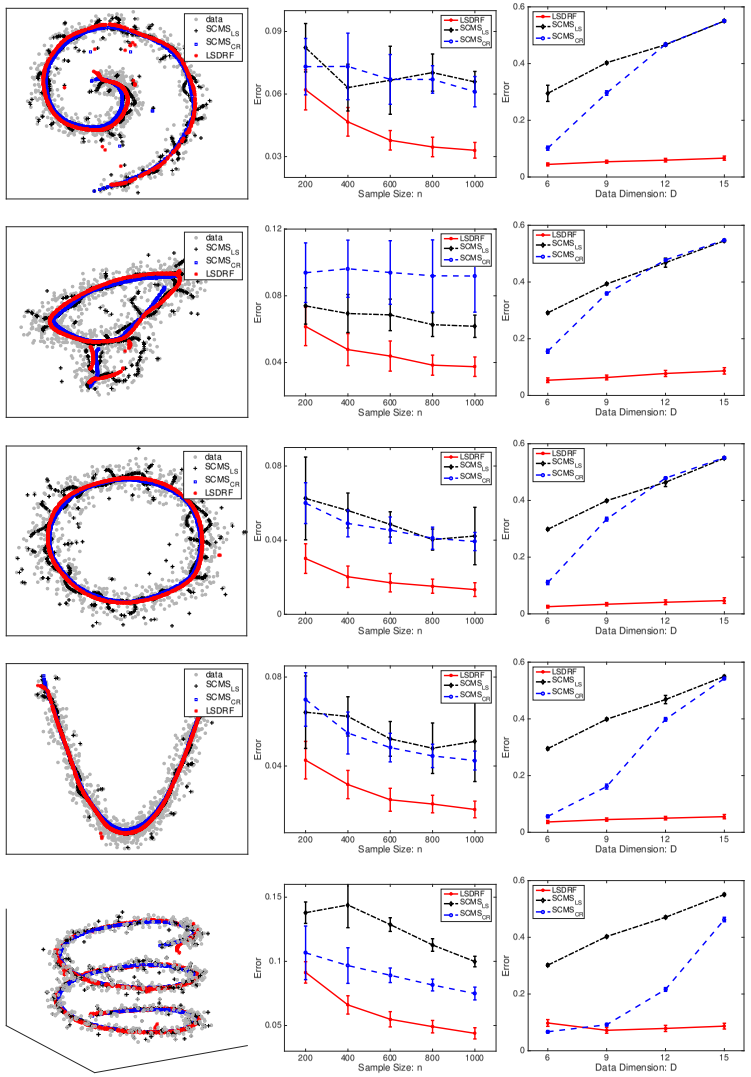
- **SCMS_{LS}**: The bandwidth parameter h was cross-validated based on the standard *integrated squared error*. We selected ten candidates of h from $10^l \times h_{\text{med}}$ $(-1.5 \leq l \leq 0)$ where h_{med} is the median value of $|x_i^{(j)} - x_k^{(j)}|$ with respect to i, j and k.
- **SCMS_{CR}**: The bandwidth parameter h was cross-validated based on the coverage risk proposed in Chen et al. (2015a). As suggested in Chen et al. (2015a), we selected ten candidates of h from $10^l \times h_{\text{NR}}$ $(-1 \leq l \leq 0)$ where h_{NR} is the bandwidth based on the normal reference rule (Silverman, 1986).

We investigate the performance of these methods on a variety of simulated datasets.[10] The i-th observation of data was generated according to $x_i^{(j)} = f^{(j)}(t_i) + n_i^{(j)}$, where t_i was taken from some range at regular intervals, $f^{(j)}(\cdot)$ denotes some fixed function, and $n_i^{(j)}$ was the Gaussian noise with mean 0 and standard deviation 0.15. Higher-dimensional data were created by appending the Gaussian variables with mean 0 and standard deviation 0.15. The estimation error was measured by

$$\text{Error} = \frac{1}{n}\sum_{i=1}^{n} \min_l \|\widehat{\boldsymbol{y}}_i - \boldsymbol{f}(t_l)\|, \tag{37}$$

where $\boldsymbol{f}(\cdot) = (f^{(1)}(\cdot), f^{(2)}(\cdot), \ldots, f^{(D)}(\cdot))^\top$ and $\widehat{\boldsymbol{y}}_i$ denotes an estimate of the density ridge point from $\boldsymbol{x}_i$.

10. Most of the datasets are generated using a MATLAB package made by Jakob Verbeek, which is available at `http://lear.inrialpes.fr/people/verbeek/code/kseg_soft.tar.gz`.



(a) Density ridge estimates (b) Error against sample size (c) Error against data dimension

Figure 8: Performance of ridge estimation on artificial data. Each point and error bar denote the average and standard deviation of ARI over 50 runs, respectively. For (c), $n = 1000$. Errors for ridge estimation are computed according to (37).

The estimated ridges are visualized in Fig.7. SCMS_{LS} provides a broken and non-smooth ridge estimate because the selected bandwidth by the least-squares cross-validation is small for density ridge estimation as in mode-seeking clustering. In contrast, the ridges estimated by LSDRF and SCMS_{CR} are smooth. However, SCMS_{CR} gives a biased estimate around highly curved region in the true ridge (e.g., the centers of the spiral and quadratic curve in Fig.7), while the bias in LSDDR seems smaller. This implies that LSDRF more accurately estimates density ridges. The accuracy of LSDRF is quantified on a variety of artificial datasets in Fig 8. LSDRF produces smaller errors particularly when the sample size is large (Fig 8(b)). In addition, as in mode-seeking clustering, the performance of LSDRF is even better when the dimensionality of data is higher (Fig. 8(c)). This implies that our direct approach is useful for high(er)-dimensional data.

5.2.2 DENSITY RIDGE ESTIMATION ON REAL-WORLD DATASETS

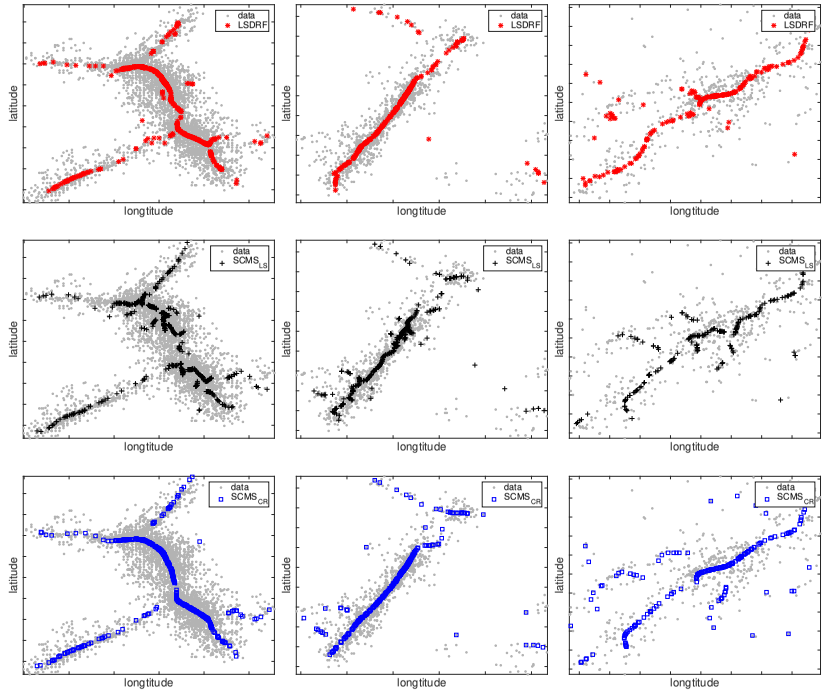
Next, we apply LSDRF to real-world datasets. As in Pulkkinen (2015), we employed the following two datasets:

- *New Madrid earthquake* dataset: This seismological dataset was downloaded from the Center for Earthquake Research and Information.[11] The dataset contains positional information for earthquakes around the New Madrid seismic zone from 1974 to 2016, providing $11,131$ samples. The three regions in Figs.9(a,b,c) were extracted according to (a) $(-90.2, -89.25)$, (b) $(-92.5, -92.15)$ and (c) $(-85.5, -83.5)$ degrees for the latitude range, respectively. For the longitude range, (a) $(36, 36.8)$, (b) $(35.2, 35.4)$ and (c) $(34.5, 36.5)$ degrees were selected. The total numbers of the original data samples and reduced data samples in each region were (a) $(N, n) = (5902, 500)$, (b) $(N, n) = (1548, 300)$ and (c) $(N, n) = (594, 200)$.

- *Shapley galaxy* dataset: This dataset was downloaded from the Center for Astrostatistics at Pennsylvania State University.[12] The dataset contains information about the three-dimensional sky angles and recession velocity of $4,215$ galaxies. As done in Pulkkinen (2015), we transformed the data samples into the three-dimensional Cartesian coordinates based on the fact that the recession velocity is proportional to the radial distance (Drinkwater et al., 2004). The three regions in Figs 10(a,b,c) were extracted according to a velocity range: (a) $(6000, 20000)$ km/s, (b) $(1500, 6000)$ km/s and (c) $(6000, 10500)$ km/s, respectively. The total numbers of the original data samples and reduced data samples in each region were (a) $(N, n) = (2849, 500)$, (b) $(N, n) = (595, 200)$ and (c) $(N, n) = (351, 150)$.

In each dataset, we focused on three regions containing prominent features, and standardized data samples in each region by subtracting the mean value and dividing by standard deviation in a dimension-wise manner. Here, the standardized data samples are collectively denoted by $\widetilde{\mathcal{D}} = \{\widetilde{\boldsymbol{x}}_i\}_{i=1}^N$. Before applying density ridge estimation methods, we performed preprocessing to remove clutter noises: KDE was applied to the dataset of each region, and then the data samples $\widetilde{\boldsymbol{x}}_i$ in each region were removed when $\frac{\widehat{p}_{\text{KDE}}(\widetilde{\boldsymbol{x}}_i)}{\max_j[\widehat{p}_{\text{KDE}}(\widetilde{\boldsymbol{x}}_j)]} < 10^{-3}$. After noise removal, we randomly chose n data samples from each region, and applied the three density ridge estimation methods to the sub-sampled data. The sub-sampled data are collectively expressed by $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^n$. For performance comparison, we computed the logarithm of $\widehat{p}_{\text{KDE}}$ on the estimated density ridges, which

11. http://www.memphis.edu/ceri/seismic/
12. http://astrostatistics.psu.edu/datasets/Shapley_galaxy.html



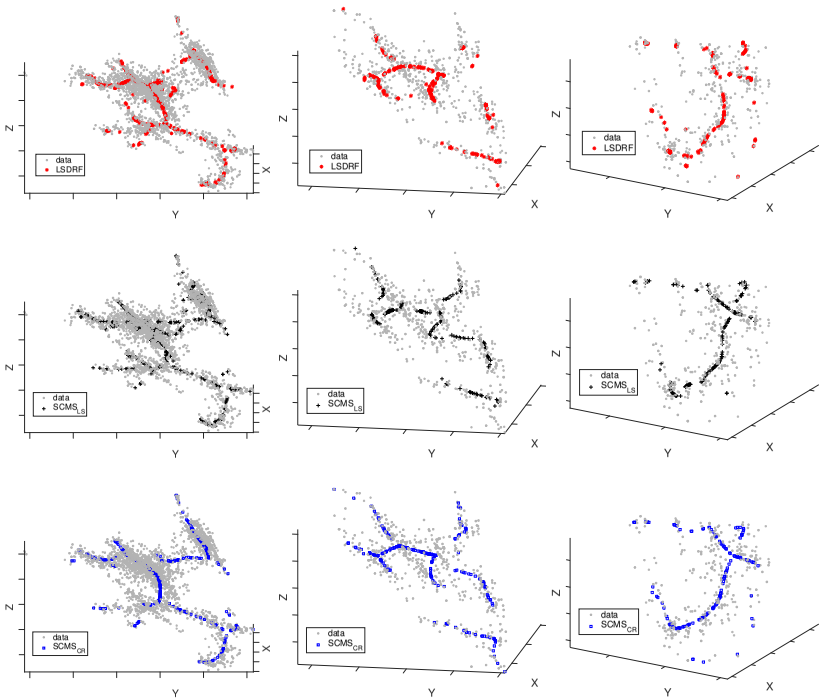
(a) Madrid 1 (b) Madrid 2 (c) Madrid 3

Figure 9: Density ridge estimation to three regions in the New Madrid earthquake dataset. The three regions (a,b,c) were extracted according to a range of latitude and longitude. The first, second and third rows correspond to results from LSDRF, SCMS_{LS} and SCMS_{CR}, respectively.

is given by

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n} \log \widehat{p}_{\text{KDE}}(\widehat{\boldsymbol{y}}_i),$$

where the centers of the kernel function in $\widehat{p}_{\text{KDE}}$ were set at the original data samples $\widetilde{\boldsymbol{x}}_i$ in each region, and $\widehat{\boldsymbol{y}}_i$ denotes an estimated density ridge point from $\boldsymbol{x}_i$. If $\mathcal{L}$ is larger, the performance can be interpreted to be better because ridges are defined on relatively high density areas. Unlike the



(a) Shapley 1 (b) Shapley 2 (c) Shapley 3

Figure 10: Density ridge estimation to three regions in the Shapley galaxy dataset. The three regions were extracted according to a range of recession velocity. The first, second and third rows correspond to results from LSDRF, $\mathrm{SCMS}_{\mathrm{LS}}$ and $\mathrm{SCMS}_{\mathrm{CR}}$, respectively.

last illustration, for $\mathrm{SCMS}_{\mathrm{LS}}$, we employed the following adaptive-bandwidth Gaussian kernel:

$$\frac{1}{(2\pi h_i^2)^{D/2}} \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2h_i^2}\right),$$

where h_i denotes the bandwidth parameter. We restricted h_i at the m-nearest neighbor Euclidean distance from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ $(i \neq j)$, and performed cross-validation with respect to m whose candidates were 128, 64, 32, 16, 8 and 4. For $\mathrm{SCMS}_{\mathrm{LS}}$, the ten candidates of the bandwidth parameter were selected from $10^l \times h_{\mathrm{NR}}$ $(-0.3 \leq l \leq 0)$. For LSDRF, we employed all data samples $\{\boldsymbol{x}_i\}_{i=1}^n$ as the centers of the Gaussian kernel, and used the median value of $\mathrm{CV}(t)$ in Section 2.4 instead of the mean value in cross-validation.

Table 2: The average and standard deviation of the performance measure $\mathcal{L}$ over 50 runs. Madrid 1, 2 and 3 (or Shapley 1, 2 and 3) correspond to the three regions in Fig.9 (or Fig.10). A larger value means a better result. Numbers in the parentheses are standard deviations. The best and comparable methods judged by the unpaired t-test at the significance level 1% are described in boldface.

New Madrid earthquake			
	LSDRF	$\mathrm{SCMS}_{\mathrm{LS}}$	$\mathrm{SCMS}_{\mathrm{CR}}$
Madrid 1	**-0.511(0.101)**	-0.610(0.072)	-0.571(0.075)
Madrid 2	**0.001(0.175)**	**0.029(0.065)**	-0.076(0.075)
Madrid 3	**-1.173(0.132)**	-1.238(0.086)	-1.238(0.098)

Shapley galaxy			
	LSDRF	$\mathrm{SCMS}_{\mathrm{LS}}$	$\mathrm{SCMS}_{\mathrm{CR}}$
Shapley 1	**0.188(0.093)**	0.094(0.073)	0.063(0.121)
Shapley 2	**-1.120(0.145)**	-1.220(0.097)	-1.462(0.223)
Shapley 3	**-1.295(0.114)**	-1.544(0.076)	-1.581(0.091)

Ridges estimated by LSDRF are smooth and seem to qualitatively well-match the ridges in the underlying data, and $\mathrm{SCMS}_{\mathrm{CR}}$ and $\mathrm{SCMS}_{\mathrm{LS}}$ also perform fairly good (Figs.9 and 10). Table 2 is quantitative comparison by $\mathcal{L}$, showing that LSDRF compares favorably with both $\mathrm{SCMS}_{\mathrm{CR}}$ and $\mathrm{SCMS}_{\mathrm{LS}}$.

6. Conclusion

In this paper, we proposed a novel estimator of the ratios of the density derivatives to the density. In stark contrast with the approaches in mean shift clustering and subspace constrained mean shift, our approach is to *directly* estimate the *density-derivative-ratios* without going through density estimation and computing the ratios. The proposed estimator was theoretically investigated, and the convergence rate was established. We applied the proposed estimator to mode-seeking clustering and density ridge estimation, and developed practical methods. Moreover, theoretical analysis were also performed to these methods , and the convergence rates to the mode and ridge of the true density were established. Our experimental illustration demonstrated that the proposed methods for mode-seeking clustering and density ridge estimation outperformed existing methods particularly for high(er)-dimensional data.

This paper focused only on mode-seeking clustering and density ridge estimation. The proposed estimator can be useful or extended for other problems. For instance, making use of the global mode (the global maximum) of a conditional density enables us to develop a regression method robust against outliers (Yao et al., 2012). Non-parametric estimation of the mode is also needed in functional data analysis (Gasser et al., 1998). In future, we explore novel applications of the proposed estimator.

Acknowledgments

The authors are grateful to Dr. Matthew James Holland for his helpful comments on an earlier version of this paper. Takafumi Kanamori was supported by KAKENHI 16K00044, 15H03636, and 15H01678. Aapo Hyvärinen was supported by the Academy of Finland. Gang Niu was supported by CREST JPMJCR1403. Masashi Sugiyama was supported by the International Research Center for Neurointelligence (WPI-IRCN) at The University of Tokyo Institutes for Advanced Study.

Appendix A. Proof of Theorem 1

Proof We first derive the following two lemmas by modifying the proof techniques in Sriperumbudur et al. (2013):

Lemma 15 *With $\epsilon = 1$ in Assumption* **(D)***, the following statements hold:*

(i) For J_j with the regularizer,

$$J_j^\lambda(r_j) := J_j(r_j) + \lambda_j \|r_j\|_{\mathcal{H}}^2,$$

the minimizer of J_j^λ is given by

$$r_j^\lambda := \operatorname*{argmin}_{r_j \in \mathcal{H}} J_j^\lambda(r_j) = (C + \lambda_j I)^{-1} \xi_j = (C + \lambda_j I)^{-1} C r_j^*,$$

where $C = \int_{\mathcal{X}} k(\cdot, \boldsymbol{x}) \otimes k(\cdot, \boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}$, $\otimes$ is the tensor product, and

$$\xi_j := (-1)^{|j|} \int_{\mathcal{X}} \partial_j k(\cdot, \boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}.$$

(ii) $\widehat{J}_j(r_j)$ can be equivalently expressed as

$$\widehat{J}_j(r_j) = 1 < r_j - r_j^*, \widehat{C}(r_j - r_j^*) >_{\mathcal{H}},$$

where

$$\widehat{C} := \frac{1}{n} \sum_{i=1}^n k(\cdot, \boldsymbol{x}_i) \otimes k(\cdot, \boldsymbol{x}_i) \quad \textit{and} \quad \widehat{\xi}_j := \frac{(-1)^{|j|}}{n} \sum_{i=1}^n \partial_j k(\cdot, \boldsymbol{x}_i).$$

Then, $\widehat{r}_j$ is given by

$$\widehat{r}_j = \operatorname*{argmin}_{r_j \in \mathcal{H}} \left[\widehat{J}_j(r_j) + \lambda_j \|r_j\|_{\mathcal{H}}^2 \right] = (\widehat{C} + \lambda_j I)^{-1} \widehat{\xi}_j.$$

Lemma 16 *With $\epsilon = 2$ in Assumption* **(D)***,*

$$\|\widehat{\xi}_j - \widehat{C} r_j^*\|_{\mathcal{H}} = O_{\mathrm{P}}\left(n^{-1/2}\right). \tag{38}$$

The proofs of these lemmas can be seen in Appendices A.1 and A.2, respectively.

Next, we make use of the proof of Theorem 5 in Sriperumbudur et al. (2013) to prove Theorem 1. From Lemma 15,

$$\begin{aligned}
\widehat{r}_j - r_j^\lambda &= (\widehat{C} + \lambda_j I)^{-1} \widehat{\xi}_j - r_j^\lambda \\
&= (\widehat{C} + \lambda_j I)^{-1} \left\{ \widehat{\xi}_j - \widehat{C} r_j^\lambda - \lambda_j r_j^\lambda \right\} \\
&= (\widehat{C} + \lambda_j I)^{-1} (\widehat{\xi}_j - \widehat{C} r_j^*) + (\widehat{C} + \lambda_j I)^{-1} (C - \widehat{C})(r_j^\lambda - r_j^*),
\end{aligned}$$

where we used $\lambda_j r_j^\lambda = C(r_j^* - r_j^\lambda)$ from Lemma 15(i). Therefore,

$$\begin{aligned}
\|\widehat{r}_j - r_j^*\|_{\mathcal{H}} &\leq \|\widehat{r}_j - r_j^\lambda\|_{\mathcal{H}} + \|r_j^\lambda - r_j^*\|_{\mathcal{H}}. \\
&\leq \|(\widehat{C} + \lambda_j I)^{-1}\| (\|\widehat{\xi}_j - \widehat{C} r_j^*\|_{\mathcal{H}} + \|C - \widehat{C}\| \mathcal{A}_0(\lambda_j)) + \mathcal{A}_0(\lambda_j),
\end{aligned}$$

where $\mathcal{A}_0(\lambda_j) = \|r_j^\lambda - r_j^*\|_{\mathcal{H}}$. It can be shown that $\|(\widehat{C} + \lambda_j I)^{-1}\| \leq 1/\lambda_j$ for sufficiently small λ_j. Thus, Lemma 16 shows that the first term can be bounded by $O_{\mathrm{P}}\left(\frac{1}{\lambda_j \sqrt{n}}\right)$. In addition, with the proof techniques in Fukumizu et al. (2007, Lemma 5), $\|C - \widehat{C}\| \leq \|C - \widehat{C}\|_{\mathrm{HS}} = O_{\mathrm{P}}(n^{-1/2})$ with $\epsilon = 2$ where $\|\cdot\|_{\mathrm{HS}}$ denotes the Hilbert-Schmidt norm. Thus, the second term is of the order $O_{\mathrm{P}}\left(\frac{\mathcal{A}_0(\lambda_j)}{\sqrt{n}\lambda_j}\right)$. From these results,

$$\|\widehat{r}_j - r_j^*\|_{\mathcal{H}} \leq O_{\mathrm{P}}\left(\frac{1}{\lambda_j \sqrt{n}}\right) + O_{\mathrm{P}}\left(\frac{\mathcal{A}_0(\lambda_j)}{\sqrt{n}\lambda_j}\right) + \mathcal{A}_0(\lambda_j). \tag{39}$$

Propostion A.2 in Sriperumbudur et al. (2013) states that if $r_j^* \in \mathcal{R}(C^\gamma)$ and C is a bounded and self-adjoint compact operator on a separable $\mathcal{H}$, the following inequality holds:

$$\mathcal{A}_0(\lambda_j) \leq \max(1, \|C\|^{\gamma - 1}) \lambda_j^{\min(1,\gamma)} \|C^{-\gamma} r_j^*\|_{\mathcal{H}}. \tag{40}$$

It can be easily verified that C is a self-adjoint operator. Assumption (D) with $\epsilon = 2$ ensures that C is a Hilbert-Schmidt operator and therefore compact because it is bounded in terms of the Hilbert-Schmidt norm. Thus, applying (40) to (39) completes the proof when choosing $\lambda_j = O\left(n^{-\max\left\{\frac{1}{4}, \frac{1}{2(\gamma+1)}\right\}}\right)$ as $n \to \infty$. ■

A.1 Proof of Lemma 15

Proof *(i)* From the definition of J_j,

$$\begin{aligned}
J_j(r_j) &= \int_{\mathcal{X}} \left\{ r_j(\boldsymbol{x}) - r_j^*(\boldsymbol{x}) \right\}^2 p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\
&= \int_{\mathcal{X}} 1 < r_j - r_j^*, k(\cdot, \boldsymbol{x}) >_{\mathcal{H}}^2 p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\
&= \int_{\mathcal{X}} 1 < r_j - r_j^*, (k(\cdot, \boldsymbol{x}) \otimes k(\cdot, \boldsymbol{x}))(r_j - r_j^*) >_{\mathcal{H}} p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\
&= \int_{\mathcal{X}} 1 < r_j - r_j^*, C_{\mathrm{x}}(r_j - r_j^*) >_{\mathcal{H}} p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\
&= 1 < r_j - r_j^*, C(r_j - r_j^*) >_{\mathcal{H}},
\end{aligned}$$

where $C_{\mathrm{x}} := k(\cdot, \boldsymbol{x}) \otimes k(\cdot, \boldsymbol{x})$. Expanding the right-hand side above transforms $J_{\boldsymbol{j}}^{\lambda}$ as

$$\begin{aligned} J_{\boldsymbol{j}}^{\lambda}(r_{\boldsymbol{j}}) &= 1 < r_{\boldsymbol{j}}, C r_{\boldsymbol{j}} >_{\mathcal{H}} - 2 1 < r_{\boldsymbol{j}}, C r_{\boldsymbol{j}}^* >_{\mathcal{H}} + 1 < r_{\boldsymbol{j}}^*, C r_{\boldsymbol{j}}^* >_{\mathcal{H}} + \lambda_{\boldsymbol{j}} 1 < r_{\boldsymbol{j}}, r_{\boldsymbol{j}} >_{\mathcal{H}} \\ &= 1 < r_{\boldsymbol{j}}, (C + \lambda_{\boldsymbol{j}} I) r_{\boldsymbol{j}} >_{\mathcal{H}} - 2 1 < r_{\boldsymbol{j}}, C r_{\boldsymbol{j}}^* >_{\mathcal{H}} + 1 < r_{\boldsymbol{j}}^*, C r_{\boldsymbol{j}}^* >_{\mathcal{H}} . \end{aligned} \tag{41}$$

For the second term in (41), we compute

$$\begin{aligned} 1 < r_{\boldsymbol{j}}, C r_{\boldsymbol{j}}^* >_{\mathcal{H}} &= 1 < r_{\boldsymbol{j}}, \int_{\mathcal{X}} k(\cdot, \boldsymbol{x}) r_{\boldsymbol{j}}^*(\boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} >_{\mathcal{H}} \\ &= 1 < r_{\boldsymbol{j}}, \int_{\mathcal{X}} k(\cdot, \boldsymbol{x}) \partial_{\boldsymbol{j}} p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} >_{\mathcal{H}} \\ &= 1 < r_{\boldsymbol{j}}, (-1)^{|\boldsymbol{j}|} \int_{\mathcal{X}} \partial_{\boldsymbol{j}} k(\cdot, \boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} >_{\mathcal{H}} \\ &= 1 < r_{\boldsymbol{j}}, \xi_{\boldsymbol{j}} >_{\mathcal{H}}, \end{aligned} \tag{42}$$

where we applied Assumption **(C)**, and

$$\xi_{\boldsymbol{j}} = (-1)^{|\boldsymbol{j}|} \int_{\mathcal{X}} \partial_{\boldsymbol{j}} k(\cdot, \boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}.$$

Comparing the left-hand side with the right-hand side at the last line in (42) gives

$$C r_{\boldsymbol{j}}^* = \xi_{\boldsymbol{j}}. \tag{43}$$

Eq.(43) is valid because (42) holds for arbitrary $r_{\boldsymbol{j}} \in \mathcal{H}$.

Simple calculation after substituting (42) into (41) provides

$$J_{\boldsymbol{j}}^{\lambda}(r_{\boldsymbol{j}}) = \|(C + \lambda_{\boldsymbol{j}} I)^{1/2} r_{\boldsymbol{j}} - (C + \lambda_{\boldsymbol{j}} I)^{-1/2} \xi_{\boldsymbol{j}}\|_{\mathcal{H}}^2 - 1 < \xi_{\boldsymbol{j}}, (C + \lambda_{\boldsymbol{j}} I)^{-1} \xi_{\boldsymbol{j}} >_{\mathcal{H}} + 1 < r_{\boldsymbol{j}}^*, C r_{\boldsymbol{j}}^* >_{\mathcal{H}},$$

Since the second and third terms in the right-hand side above do not include $r_{\boldsymbol{j}}$, the minimizer of $J_{\boldsymbol{j}}^{\lambda}(r_{\boldsymbol{j}})$ is given by $r_{\boldsymbol{j}}^{\lambda} = (C + \lambda_{\boldsymbol{j}} I)^{-1} \xi_{\boldsymbol{j}} = (C + \lambda_{\boldsymbol{j}} I)^{-1} C r_{\boldsymbol{j}}^*$ where (43) was applied.

(ii) It follows from (i) by substituting C and $\xi_{\boldsymbol{j}}$ with $\widehat{C}$ and $\widehat{\xi}_{\boldsymbol{j}}$, respectively. ■

A.2 Proof of Lemma 16

Proof We first compute the expectation of $\|\widehat{\xi}_{\boldsymbol{j}} - \widehat{C} r_{\boldsymbol{j}}^*\|_{\mathcal{H}}^2$ as

$$E\|\widehat{\xi}_{\boldsymbol{j}} - \widehat{C} r_{\boldsymbol{j}}^*\|_{\mathcal{H}}^2 = \frac{n-1}{n} \|\xi_{\boldsymbol{j}} - C r_{\boldsymbol{j}}^*\|_{\mathcal{H}}^2 + \frac{1}{n} \int_{\mathcal{X}} \|(-1)^{|\boldsymbol{j}|} \partial_{\boldsymbol{j}} k(\cdot, \boldsymbol{x}) + C_{\mathrm{x}} r_{\boldsymbol{j}}^*\|_{\mathcal{H}}^2 p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \tag{44}$$

where $C_{\mathrm{x}} = k(\cdot, \boldsymbol{x}) \otimes k(\cdot, \boldsymbol{x})$. Eq.(43) indicates that the first term in the right-hand side of (44) vanishes, i.e., $\|\xi_{\boldsymbol{j}} - C r_{\boldsymbol{j}}^*\|_{\mathcal{H}} = 0$. From

$$\|(-1)^{|\boldsymbol{j}|} \partial_{\boldsymbol{j}} k(\cdot, \boldsymbol{x}) + C_{\mathrm{x}} r_{\boldsymbol{j}}^*\|_{\mathcal{H}}^2 \leq 2 \|\partial_{\boldsymbol{j}} k(\cdot, \boldsymbol{x})\|_{\mathcal{H}}^2 + 2 \|C_{\mathrm{x}}\|_{\mathrm{HS}}^2 \|r_{\boldsymbol{j}}^*\|_{\mathcal{H}}^2,$$

Assumption (D) with $\epsilon = 2$ ensures that the second term in the right-hand side of (44) is finite. Thus, applying the Chebyshev's inequality proves the lemma because $E(\widehat{\xi}_{\boldsymbol{j}} - \widehat{C} r_{\boldsymbol{j}}^*) = \xi_{\boldsymbol{j}} - C r_{\boldsymbol{j}}^* = 0$ from (43). ■

Appendix B. Connection to the Minimax Theory

This appendix provides details for the connections to the minimax theory discussed in the remark after Theorem 1. First, we introduce the following results:

- By the minimax theory (Tsybakov, 2009), Eq.(10) in Sriperumbudur et al. (2017) shows the minimax rate: For any $\alpha > \delta \geq 0$,

$$\inf_{\widehat{r}_{\boldsymbol{j},n}} \sup_{r_{\boldsymbol{j}}^* \in H_2^{\alpha}} \|\widehat{r}_{\boldsymbol{j},n} - r_{\boldsymbol{j}}^*\|_{H_2^{\delta}} \asymp n^{-\frac{\alpha-\delta}{2(\alpha-\delta)+D}}. \tag{45}$$

- The following proposition provides necessary conditions for $r_{\boldsymbol{j}}^* \in R(C)$:

Proposition 17 *Suppose that $\psi, \phi \in C(\mathbb{R}^D) \cap L^1(\mathbb{R}^D)$ are real-valued, shift-invariant and positive definite kernel functions. Let $\mathcal{H}$ and $\mathcal{G}$ be RKHSs associated with $\psi(\boldsymbol{x} - \boldsymbol{y})$ and $\phi(\boldsymbol{x} - \boldsymbol{y})$, respectively. For $2 \leq r \leq \infty$, assume that the followings hold,*

$$p \in L^{\frac{r}{r-1}}(\mathbb{R}^D), \quad \left\| \frac{\phi^{\wedge}}{\psi^{\wedge}} \right\|_{\infty} < \infty \quad \textit{and} \quad \left\| \frac{\psi^{\wedge 2}}{\phi^{\wedge}} \right\|_{\frac{r}{r-2}} < \infty.$$

Then, $r_{\boldsymbol{j}}^* \in R(C)$ implies that $r_{\boldsymbol{j}}^* \in \mathcal{G} \subset \mathcal{H}$, where $k(\boldsymbol{x}, \boldsymbol{y}) = \psi(\boldsymbol{x} - \boldsymbol{y})$ in the operator C.

The proof of Proposition 17 is deferred to Section B.1. The conditions are necessary ones for $r_{\boldsymbol{j}}^* \in R(C^{\beta})$ with $\beta > 1$ as well because $R(C^{\beta_1}) \subset R(C^{\beta_2})$ for $0 < \beta_1 < \beta_2 < \infty$ (Sriperumbudur et al., 2017, Section 4.2 and Appendix B.3).

Recall that when the *Matérn kernel*, $k(\boldsymbol{x}, \boldsymbol{y}) = \psi(\boldsymbol{x} - \boldsymbol{y}) = \frac{2^{1-s}}{\Gamma(s)} \|\boldsymbol{x} - \boldsymbol{y}\|^{s - D/2} \mathfrak{K}_{D/2-s}(\|\boldsymbol{x} - \boldsymbol{y}\|)$, is employed, the corresponding RKHS $\mathcal{H}$ is the Sobolev space H_2^s with $s > D/2$ (Wendland, 2004, Chapter 10):

$$\mathcal{H} = H_2^s := \left\{ f \in L^2(\mathbb{R}^D) \cap C(\mathbb{R}^D) : \int (1 + \|\boldsymbol{\omega}\|^2)^s |f^{\wedge}(\boldsymbol{\omega})|^2 \mathrm{d}\boldsymbol{\omega} < \infty \right\}.$$

Theorem 6.13 in Wendland (2004) gives the Fourier transform of ψ as

$$\psi^{\wedge}(\boldsymbol{\omega}) = (1 + \|\boldsymbol{\omega}\|^2)^{-s}.$$

When $p \in L^1(\mathbb{R}^D)$, applying Proposition 17 ensures that $r_{\boldsymbol{j}}^* \in R(C)$ implies $r_{\boldsymbol{j}}^* \in H_2^{s'} \subset H_2^s$ with $\frac{D}{2} < s \leq s' < 2s + \frac{1}{2} - \frac{D}{2}$. Thus, $r_{\boldsymbol{j}}^* \in H_2^{2s + \frac{1}{2} - \frac{D}{2} - \epsilon}$ for arbitrarily small $\epsilon > 0$. Then, if we chose $\mathcal{H} = H_2^{D - \frac{1}{2} + \epsilon}$, the rate $n^{-\frac{1}{4}}$ in Theorem 1 is minimax optimal (Set $\alpha = 2s + \frac{1}{2} - \frac{D}{2} - \epsilon$ and $\delta = s$ in (45), equate the exponent in the right-hand side of (45) with $-\frac{1}{4}$, and solve it with respect to s). Similar discussion is possible when $p \in L^2(\mathbb{R}^D)$: The rate is minimax optimal under the choice of $\mathcal{H} = H_2^{\frac{D}{2} + \epsilon}$.

B.1 Proof of Proposition 17

Here, we modify the proof of Proposition 8 in Sriperumbudur et al. (2017).

Proof To characterize RKHSs induced by shift-invariant kernels, we employ the following lemma:

Lemma 18 (Theorem 10.12 in Wendland (2004)) *Let $\psi(\boldsymbol{x}-\boldsymbol{y})$ be a real-valued, symmetric and positive definite kernel. When $\psi \in C(\mathbb{R}^D) \cap L^1(\mathbb{R}^D)$, it induces the following Hilbert space,*

$$\mathcal{H} := \left\{ f \in C(\mathbb{R}^D) \cap L^2(\mathbb{R}^D) : \frac{f^\wedge}{\sqrt{\psi^\wedge}} \in L^2(\mathbb{R}^D) \right\},$$

with the reproducing kernel $\psi(\boldsymbol{x}-\boldsymbol{y})$ and inner product,

$$\langle f, g \rangle_{\mathcal{H}} := \frac{1}{(2\pi)^{D/2}} \int \frac{f^\wedge(\boldsymbol{\omega})\overline{g^\wedge(\boldsymbol{\omega})}}{\psi^\wedge(\boldsymbol{\omega})} \mathrm{d}\boldsymbol{\omega}.$$

$\overline{g^\wedge(\boldsymbol{\omega})}$ above denotes the complex conjugate of $g^\wedge(\boldsymbol{\omega})$. In particular, every f in $\mathcal{H}$ can be recovered from its Fourier transform $f^\wedge \in L^1(\mathbb{R}^D) \cap L^2(\mathbb{R}^D)$ as

$$f(\boldsymbol{x}) = \frac{1}{(2\pi)^{D/2}} \int f^\wedge(\boldsymbol{\omega}) e^{i\boldsymbol{x}^\top \boldsymbol{\omega}} \mathrm{d}\boldsymbol{\omega}. \tag{46}$$

Let us express an RKHS $\mathcal{G}$ induced by another real-valued, symmetric and positive definite kernel $\phi(\boldsymbol{x}-\boldsymbol{y})$. We first show that $\mathcal{G} \subset \mathcal{H}$ if $\left\| \frac{\phi^\wedge(\boldsymbol{\omega})}{\psi^\wedge(\boldsymbol{\omega})} \right\|_\infty < \infty$. From Lemma 18, for $g \in \mathcal{G}$, the norm in $\mathcal{H}$ is computed as

$$\|g\|_{\mathcal{H}}^2 = \frac{1}{(2\pi)^{D/2}} \int \frac{|g^\wedge(\boldsymbol{\omega})|^2}{\psi^\wedge(\boldsymbol{\omega})} \mathrm{d}\boldsymbol{\omega} = \frac{1}{(2\pi)^{D/2}} \int \frac{|g^\wedge(\boldsymbol{\omega})|^2 \phi^\wedge(\boldsymbol{\omega})}{\phi^\wedge(\boldsymbol{\omega})\psi^\wedge(\boldsymbol{\omega})} \mathrm{d}\boldsymbol{\omega} \leq \|g\|_{\mathcal{G}}^2 \left\| \frac{\phi^\wedge(\boldsymbol{\omega})}{\psi^\wedge(\boldsymbol{\omega})} \right\|_\infty < \infty.$$

Thus, $g \in \mathcal{H}$, which indicates that $\mathcal{G} \subset \mathcal{H}$.

Next, we show that $r_{\boldsymbol{j}}^* \in R(C)$ indicates $r_{\boldsymbol{j}}^* \in \mathcal{G}$. Since $r_{\boldsymbol{j}}^* \in R(C)$, there exists $f \in \mathcal{H}$ such that $r_{\boldsymbol{j}}^* = Cf$, i.e.,

$$\begin{aligned}
r_{\boldsymbol{j}}^*(\boldsymbol{y}) &= \int k(\boldsymbol{x}, \boldsymbol{y}) f(\boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\
&= \int \psi(\boldsymbol{x}-\boldsymbol{y}) f(\boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\
&= \int \left[\frac{1}{(2\pi)^{D/2}} \int \psi^\wedge(\boldsymbol{\omega}) e^{i(\boldsymbol{x}-\boldsymbol{y})^\top \boldsymbol{\omega}} \mathrm{d}\boldsymbol{\omega} \right] f(\boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\
&= \int \left[\frac{1}{(2\pi)^{D/2}} \int f(\boldsymbol{x}) p(\boldsymbol{x}) e^{i\boldsymbol{x}^\top \boldsymbol{\omega}} \mathrm{d}\boldsymbol{x} \right] \psi^\wedge(\boldsymbol{\omega}) e^{-i\boldsymbol{y}^\top \boldsymbol{\omega}} \mathrm{d}\boldsymbol{\omega} \\
&= \int (f^\wedge * p^\wedge)(-\boldsymbol{\omega}) \psi^\wedge(\boldsymbol{\omega}) e^{-i\boldsymbol{y}^\top \boldsymbol{\omega}} \mathrm{d}\boldsymbol{\omega},
\end{aligned} \tag{47}$$

where we applied (46) to $\psi(\boldsymbol{x}-\boldsymbol{y})$ on the third line and Fubini's theorem on the fourth line, and $*$ denotes the convolution such that

$$(f * p)(\boldsymbol{x}) := \int f(\boldsymbol{y}) p(\boldsymbol{x}-\boldsymbol{y}) \mathrm{d}\boldsymbol{y}.$$

Eq.(47) indicates that the Fourier transform of $r_{\boldsymbol{j}}^*$ is given by

$$r_{\boldsymbol{j}}^{*\wedge}(\boldsymbol{\omega}) = (f^\wedge * p^\wedge)(-\boldsymbol{\omega}) \psi^\wedge(\boldsymbol{\omega}).$$

Computing the norm of $r_{\boldsymbol{j}}^*$ in $\mathcal{G}$ yields

$$\|r_{\boldsymbol{j}}^*\|_{\mathcal{G}}^2 = \int |(f^\wedge * p^\wedge)(-\boldsymbol{\omega})|^2 \frac{\psi^\wedge(\boldsymbol{\omega})^2}{\phi^\wedge(\boldsymbol{\omega})} \mathrm{d}\boldsymbol{\omega} \leq \|(f^\wedge * p^\wedge)^2\|_{r/2} \left\| \frac{\psi^{\wedge 2}}{\phi^\wedge} \right\|_{\frac{r}{r-2}} = \|f^\wedge * p^\wedge\|_r^2 \left\| \frac{\psi^{\wedge 2}}{\phi^\wedge} \right\|_{\frac{r}{r-2}},$$

where Hölder inequality was applied with $2 \leq r \leq \infty$. Then, Young's convolution and Hausdorff-Young inequalities (Beckner, 1975) yield

$$\|f^\wedge * p^\wedge\|_r \leq \|f^\wedge\|_1 \cdot \|p^\wedge\|_r \leq \|f^\wedge\|_1 \cdot \|p\|_{\frac{r}{r-1}} < \infty$$

Thus, by Lemma 18, $r_{\boldsymbol{j}}^* \in R(C)$ indicates $r_{\boldsymbol{j}}^* \in \mathcal{G}$. ■

Appendix C. Proof of Theorem 3

Proof Suppose that $\widehat{\alpha}_j^{(i)} = 0$ and $\widetilde{\beta}_j^{(i)} \left(= -\widehat{\beta}_j^{(i)} \right) \geq 0$ for all i and j. Computing the integral in (18) shows that

$$\begin{aligned}
\widehat{D}_{\widehat{\boldsymbol{g}}}[\boldsymbol{x}|\boldsymbol{y}] &= \sum_{j=1}^D \int_{y^{(j)}}^{x^{(j)}} \widehat{g}_j(x^{(1)}, \ldots, x^{(j-1)}, z^{(j)}, y^{(j+1)}, \ldots, y^{(D)}) \mathrm{d}z^{(j)} \\
&= \sum_{j=1}^D \sum_{i=1}^n \widetilde{\beta}_j^{(i)} \left[\phi\left(\frac{\|\boldsymbol{z}_x^j - \boldsymbol{x}_i\|^2}{2\sigma_j^2} \right) - \phi\left(\frac{\|\boldsymbol{z}_y^j - \boldsymbol{x}_i\|^2}{2\sigma_j^2} \right) \right],
\end{aligned} \tag{48}$$

where we used the relation $\partial_j' \phi\left(\|\boldsymbol{x}-\boldsymbol{x}'\|^2\right) = -\partial_j \phi\left(\|\boldsymbol{x}-\boldsymbol{x}'\|^2\right)$, and

$$\begin{aligned}
\boldsymbol{z}_y^j &= (x^{(1)}, \ldots, x^{(j-1)}, y^{(j)}, y^{(j+1)}, \ldots, y^{(D)})^\top \\
\boldsymbol{z}_x^j &= (x^{(1)}, \ldots, x^{(j-1)}, x^{(j)}, y^{(j+1)}, \ldots, y^{(D)})^\top.
\end{aligned} \tag{49}$$

Note that the j-th elements in $\boldsymbol{z}_y^j$ and $\boldsymbol{z}_x^j$ only differ. To ensure that the right-hand side in (48) is non-negative, we need to show that for all j,

$$\sum_{i=1}^n \widetilde{\beta}_j^{(i)} \left[\phi\left(\frac{\|\boldsymbol{z}_x^j - \boldsymbol{x}_i\|^2}{2\sigma_j^2} \right) - \phi\left(\frac{\|\boldsymbol{z}_y^j - \boldsymbol{x}_i\|^2}{2\sigma_j^2} \right) \right] \geq 0. \tag{50}$$

To obtain a lower bound of the left-hand side in (50), we use the following inequality, which comes from the convexity of ϕ:

$$\phi\left(\frac{\|\boldsymbol{z}_x^j-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right)-\phi\left(\frac{\|\boldsymbol{z}_y^j-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right) \geq \frac{1}{2\sigma_j^2}\varphi\left(\frac{\|\boldsymbol{z}_y^j-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right)\left[(y^{(j)}-x_i^{(j)})^2-(x^{(j)}-x_i^{(j)})^2\right]. \tag{51}$$

Since all $\widetilde{\beta}_j^{(i)}$ are assumed to be non-negative, (51) provides

$$\begin{aligned}
&\sum_{i=1}^n \widetilde{\beta}_j^{(i)}\left[\phi\left(\frac{\|\boldsymbol{z}_x^j-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right)-\phi\left(\frac{\|\boldsymbol{z}_y^j-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right)\right]\\
&\geq \frac{1}{2\sigma_j^2}\sum_{i=1}^n \widetilde{\beta}_j^{(i)}\varphi\left(\frac{\|\boldsymbol{z}_y^j-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right)\left[(y^{(j)}-x_i^{(j)})^2-(x^{(j)}-x_i^{(j)})^2\right]\\
&= \frac{1}{2\sigma_j^2}\sum_{i=1}^n \widetilde{\beta}_j^{(i)}\varphi\left(\frac{\|\boldsymbol{z}_y^j-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right)\left[(y^{(j)})^2-(x^{(j)})^2\right]-\underbrace{\sum_{i=1}^n \widetilde{\beta}_j^{(i)}x_i^{(j)}\varphi\left(\frac{\|\boldsymbol{z}_y^j-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right)}_{(\star)}\frac{(y^{(j)}-x^{(j)})}{\sigma_j^2}.
\end{aligned} \tag{52}$$

Finally, we set $\boldsymbol{y}=\boldsymbol{z}_k^\tau$ and $\boldsymbol{x}=\boldsymbol{z}_k^{\tau+1}$ in $\widehat{D}_{\widehat{\boldsymbol{g}}}[\boldsymbol{x}|\boldsymbol{y}]$, and therefore

$$\begin{aligned}
\boldsymbol{z}_y^j &= (z_k^{(\tau+1,1)},\ldots,z_k^{(\tau+1,j-1)},z_k^{(\tau,j)},z_k^{(\tau,j+1)},\ldots,z_k^{(\tau,D)})^\top=\tilde{\boldsymbol{z}}_k^\tau\\
\boldsymbol{z}_x^j &= (z_k^{(\tau+1,1)},\ldots,z_k^{(\tau+1,j-1)},z_k^{(\tau+1,j)},z_k^{(\tau+1,j+1)},\ldots,z_k^{(\tau,D)})^\top.
\end{aligned}$$

Applying the coordinate-wise update rule (15) to $(\star)$, the right-hand side in (52) becomes

$$\frac{1}{2\sigma_j^2}\sum_{i=1}^n \widetilde{\beta}_j^{(i)}\varphi\left(\frac{\|\tilde{\boldsymbol{z}}_k^\tau-\boldsymbol{x}_i\|^2}{2\sigma_j^2}\right)(z_k^{(\tau,j)}-z_k^{(\tau+1,j)})^2\geq 0.$$

This proves (50), and thus the proof was completed. ■

Appendix D. Proof of Theorem 7

Proof Under the path (17),

$$D_{\boldsymbol{g}}[\boldsymbol{x}|\boldsymbol{y}]-\widehat{D}_{\widehat{\boldsymbol{g}}}[\boldsymbol{x}|\boldsymbol{y}]=\sum_{j=1}^D\int_{j-1}^j {}_1<\boldsymbol{g}(\boldsymbol{\gamma}_j(t))-\widehat{\boldsymbol{g}}(\boldsymbol{\gamma}_j(t)),\dot{\boldsymbol{\gamma}}_j(t)>\mathrm{d}t$$

where the curve $\boldsymbol{\gamma}_j(t)$, $t\in[j-1,j]$ connects $\boldsymbol{z}_y^j$ and $\boldsymbol{z}_x^j$ by the line segment whose definition is given in (49). Then, we obtain

$$\left|\int_{j-1}^j {}_1<\boldsymbol{g}(\boldsymbol{\gamma}_j(t))-\widehat{\boldsymbol{g}}(\boldsymbol{\gamma}_j(t)),\dot{\boldsymbol{\gamma}}_j(t)>\mathrm{d}t\right|\leq\|\boldsymbol{g}-\widehat{\boldsymbol{g}}\|_\infty|y^{(j)}-x^{(j)}|.$$

Therefore,

$$|D_{\boldsymbol{g}}[\boldsymbol{x}|\boldsymbol{y}]-D_{\widehat{\boldsymbol{g}}}[\boldsymbol{x}|\boldsymbol{y}]|\leq\|\boldsymbol{g}-\widehat{\boldsymbol{g}}\|_\infty\|\boldsymbol{y}-\boldsymbol{x}\|_1.$$

Finally, with Lemma 12, the theorem was proved. ■

Appendix E. Proof of Theorem 9

We modify the proof of Theorem 1 in Chen et al. (2016b), and apply Lemma 12.

Proof Suppose that a mode point $\boldsymbol{\mu}_j\in\mathcal{M}$ is uniquely approximated by an estimated mode point $\widehat{\boldsymbol{\mu}}_j\in\widehat{\mathcal{M}}$. Then, the Taylor expansion gives

$$\begin{aligned}
\widehat{\boldsymbol{g}}(\boldsymbol{\mu}_j) &= \widehat{\boldsymbol{g}}(\widehat{\boldsymbol{\mu}}_j)+\nabla\widehat{\boldsymbol{g}}(\boldsymbol{\mu}_j)(\boldsymbol{\mu}_j-\widehat{\boldsymbol{\mu}}_j)+o(\|\boldsymbol{\mu}_j-\widehat{\boldsymbol{\mu}}_j\|)\\
&= \nabla\widehat{\boldsymbol{g}}(\boldsymbol{\mu}_j)(\boldsymbol{\mu}_j-\widehat{\boldsymbol{\mu}}_j)+o(\|\boldsymbol{\mu}_j-\widehat{\boldsymbol{\mu}}_j\|),
\end{aligned} \tag{53}$$

where $\widehat{\boldsymbol{g}}(\widehat{\boldsymbol{\mu}}_j)=\boldsymbol{0}$. On the other hand, from Lemma 12,

$$\widehat{g}_j(\boldsymbol{\mu}_j)=\widehat{g}_j(\boldsymbol{\mu}_j)-g_j(\boldsymbol{\mu}_j)=O_{\mathrm{P}}\left(n^{-\min\left\{\frac{1}{4},\frac{\gamma}{2(\gamma+1)}\right\}}\right), \tag{54}$$

where $g_j(\boldsymbol{\mu}_j)=0$. Since all eigenvalues of $\nabla\boldsymbol{g}(\boldsymbol{\mu}_j)$ are strictly negative by the definition in (20), the following relation and Lemma 12 ensures that $\nabla\widehat{\boldsymbol{g}}(\boldsymbol{\mu}_j)$ is invertible with a high probability: By the derivative reproducing property (Zhou, 2008),

$$|\partial_i g_j(\boldsymbol{x})-\partial_i\widehat{g}_j(\boldsymbol{x})|=|\langle g_j-\widehat{g}_j,\partial_i k(\boldsymbol{x},\cdot)\rangle_{\mathcal{H}}|\leq\|g_j-\widehat{g}_j\|_{\mathcal{H}}|\partial_i'\partial_i k(\boldsymbol{x}',\boldsymbol{x})|_{\boldsymbol{x}'=\boldsymbol{x}}|=O\left(\|g_j-\widehat{g}_j\|_{\mathcal{H}}\right),$$

where the Cauchy-Schwarz inequality was applied, ∂_i' denotes the partial derivative with respect to the i-th element in $\boldsymbol{x}'$, and $\partial_i'\partial_i k$ is assumed to be uniformly bounded. Thus, combining (53) with (54) yields

$$\|\boldsymbol{\mu}_j-\widehat{\boldsymbol{\mu}}_j\|=O_{\mathrm{P}}\left(n^{-\min\left\{\frac{1}{4},\frac{\gamma}{2(\gamma+1)}\right\}}\right).$$

The fact,

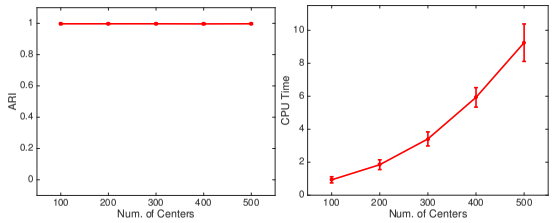
$$\mathrm{Haus}(\widehat{\mathcal{M}},\mathcal{M})=\max_j\|\boldsymbol{\mu}_j-\widehat{\boldsymbol{\mu}}_j\|,$$

proves the theorem. ■

Appendix F. Reducing the Kernel Centers

This appendix investigates clustering performance and computational costs of LSLDGC when the number of kernel centers is changed. We performed similar experiments in Section 5.1. In the experiments, datasets with the three Gaussian blobs (Fig.6(g)) were used.

Fig.11 shows that LSLDGC with a small number of kernel centers significantly reduces the computation costs without scarifying the clustering performance.



(a) ARI (b) CPU time

Figure 11: Clustering performance and CPU time against the number of kernel centers. Each point and error bar denote the average and standard deviation of (a) ARI and (b) CPU time over 50 runs, respectively. The dataset used in this figure is the three Gaussian blobs in Section 5.1 when $(D, n) = (5, 500)$

Appendix G. Proof of Lemma 12

Proof For ϵ', the Cauchy-Schwarz inequality gives

$$|\widehat{g}_j(\boldsymbol{x}) - g_j(\boldsymbol{x})| = |\langle \widehat{g}_j - g_j, k(\cdot, \boldsymbol{x})\rangle_{\mathcal{H}}| \leq \|\widehat{g}_j - g_j\|_{\mathcal{H}} |k(\boldsymbol{x}, \boldsymbol{x})|.$$

Since $k(\boldsymbol{x}, \boldsymbol{x})$ is assumed to be finite,

$$\epsilon' = \max_j \|\widehat{g}_j(\boldsymbol{x}) - g_j(\boldsymbol{x})\|_\infty \leq O\left(\|\widehat{g}_j - g_j\|_{\mathcal{H}}\right) = O_{\mathrm{P}}\left(n^{-\min\left\{\frac{1}{4}, \frac{\gamma}{2(\gamma+1)}\right\}}\right), \tag{55}$$

where we applied Theorem 1.

For ϵ'', similar computation yields

$$\begin{aligned}
&|[\widehat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{x})]_{ij} - [\boldsymbol{\Sigma}^{-1}(\boldsymbol{x})]_{ij}| \\
&= |-(\widehat{g}_i(\boldsymbol{x})\widehat{g}_j(\boldsymbol{x}) - g_i(\boldsymbol{x})g_j(\boldsymbol{x})) + [\widehat{\boldsymbol{H}}(\boldsymbol{x})]_{ij} - [\boldsymbol{H}(\boldsymbol{x})]_{ij}| \\
&\leq |\widehat{g}_i(\boldsymbol{x})\widehat{g}_j(\boldsymbol{x}) - g_i(\boldsymbol{x})g_j(\boldsymbol{x})| + |[\widehat{\boldsymbol{H}}(\boldsymbol{x})]_{ij} - [\boldsymbol{H}(\boldsymbol{x})]_{ij}| \\
&\leq |\widehat{g}_i(\boldsymbol{x})| \cdot |\widehat{g}_j(\boldsymbol{x}) - g_j(\boldsymbol{x})| + |g_j(\boldsymbol{x})| \cdot |\widehat{g}_i(\boldsymbol{x}) - g_i(\boldsymbol{x})| + |[\widehat{\boldsymbol{H}}(\boldsymbol{x})]_{ij} - [\boldsymbol{H}(\boldsymbol{x})]_{ij}| \\
&\leq \left\{|g_i(\boldsymbol{x})| \cdot \|\widehat{g}_j - g_j\|_{\mathcal{H}} + |\widehat{g}_j(\boldsymbol{x})| \cdot \|\widehat{g}_i - g_i\|_{\mathcal{H}} + \|[\widehat{\boldsymbol{H}}]_{ij} - [\boldsymbol{H}]_{ij}\|_{\mathcal{H}}\right\} |k(\boldsymbol{x}, \boldsymbol{x})|,
\end{aligned}$$

where we applied the following inequality on the fourth line:

$$\begin{aligned}
|\widehat{g}_i(\boldsymbol{x})\widehat{g}_j(\boldsymbol{x}) - g_i(\boldsymbol{x})g_j(\boldsymbol{x})| &= |\widehat{g}_i(\boldsymbol{x})\widehat{g}_j(\boldsymbol{x}) - \widehat{g}_i(\boldsymbol{x})g_j(\boldsymbol{x}) + \widehat{g}_i(\boldsymbol{x})g_j(\boldsymbol{x}) - g_i(\boldsymbol{x})g_j(\boldsymbol{x})| \\
&= |\widehat{g}_i(\boldsymbol{x})(\widehat{g}_j(\boldsymbol{x}) - g_j(\boldsymbol{x})) + g_j(\boldsymbol{x})(\widehat{g}_i(\boldsymbol{x}) - g_i(\boldsymbol{x}))| \\
&\leq |\widehat{g}_i(\boldsymbol{x})| \cdot |\widehat{g}_j(\boldsymbol{x}) - g_j(\boldsymbol{x})| + |g_j(\boldsymbol{x})| \cdot |\widehat{g}_i(\boldsymbol{x}) - g_i(\boldsymbol{x})|.
\end{aligned}$$

Thus, we obtain

$$\begin{aligned}
\epsilon'' &= \max_{ij} \max_{\boldsymbol{x}} |[\widehat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{x})]_{ij} - [\boldsymbol{\Sigma}^{-1}(\boldsymbol{x})]_{ij}| \\
&\leq \max_{ij} O\left(\max(\|\widehat{g}_j - g_j\|_{\mathcal{H}}, \|\widehat{g}_i - g_i\|_{\mathcal{H}}, \|[\widehat{\boldsymbol{H}}]_{ij} - [\boldsymbol{H}]_{ij}\|_{\mathcal{H}})\right) \\
&= O_{\mathrm{P}}\left(n^{-\min\left\{\frac{1}{4}, \frac{\gamma}{2(\gamma+1)}\right\}}\right),
\end{aligned} \tag{56}$$

where it follows from Theorem 1.

For ϵ''', we resort to the derivative reproducing property proved in Zhou (2008): For all $f \in \mathcal{H}$,

$$\partial_j f(\boldsymbol{x}) = {}_1 < f, \partial_j k(\cdot, \boldsymbol{x}) >_{\mathcal{H}}.$$

Using this relation, we obtain

$$\begin{aligned}
|[\widehat{\boldsymbol{\Sigma}}^{-1\prime}(\boldsymbol{x})]_{ij} - [\boldsymbol{\Sigma}^{-1\prime}(\boldsymbol{x})]_{ij}| &= |\partial_j[\mathrm{vec}(\widehat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{x}))]_i - \partial_j[\mathrm{vec}(\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}))]_i| \\
&= |\langle [\mathrm{vec}(\widehat{\boldsymbol{\Sigma}}^{-1})]_i - [\mathrm{vec}(\boldsymbol{\Sigma}^{-1})]_i, \partial_j k(\cdot, \boldsymbol{x})\rangle_{\mathcal{H}}| \\
&\leq \|\mathrm{vec}(\widehat{\boldsymbol{\Sigma}}^{-1})]_i - [\mathrm{vec}(\boldsymbol{\Sigma}^{-1})]_i\|_{\mathcal{H}} |\partial'_i \partial_i k(\boldsymbol{x}', \boldsymbol{x})|_{\boldsymbol{x}'=\boldsymbol{x}}|,
\end{aligned}$$

where ∂'_j denote the derivative with respect to the j-th coordinate in $\boldsymbol{x}'$. Since $|\partial'_j \partial_j k|$ is assumed to be finite, (56) provides

$$\begin{aligned}
\epsilon''' &= \max_{ij} \max_{\boldsymbol{x}} |[\widehat{\boldsymbol{\Sigma}}^{-1\prime}(\boldsymbol{x})]_{ij} - [\boldsymbol{\Sigma}^{-1\prime}(\boldsymbol{x})]_{ij}| \\
&\leq \max_i O\left(\|[\mathrm{vec}(\widehat{\boldsymbol{\Sigma}}^{-1})]_i - [\mathrm{vec}(\boldsymbol{\Sigma}^{-1})]_i\|_{\mathcal{H}}\right) \\
&= O_{\mathrm{P}}\left(n^{-\min\left\{\frac{1}{4}, \frac{\gamma}{2(\gamma+1)}\right\}}\right),
\end{aligned}$$

where the last equation comes from (56) because $[\mathrm{vec}(\boldsymbol{\Sigma}^{-1})(\boldsymbol{x})]_i$ denotes a single element in $\boldsymbol{\Sigma}^{-1}(\boldsymbol{x})$. ■

References

E. Arias-Castro, D. Mason, and B. Pelletier. On the estimation of the gradient lines of a density and the consistency of the mean-shift algorithm. *Journal of Machine Learning Research*, 17:1–28, 2016.

K. Bache and M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml/`.

W. Beckner. Inequalities in Fourier analysis. *Annals of Mathematics*, 102(1):159–182, 1975.

R. Beran. Adaptive estimates for autoregressive processes. *Annals of the Institute of Statistical Mathematics*, 28(1):77–89, 1976.

A.W. Bowman. An alternative method of cross-validation for the smoothing of density estimates. *Biometrika*, 71(2):353–360, 1984.

M.Á. Carreira-Perpiñán. Reconstruction of sequential data with probabilistic models and continuity constraints. In *Advances in neural information processing systems*, pages 414–420, 2000.

M.Á. Carreira-Perpiñán. *Continuous latent variable models for dimensionality reduction and sequential data reconstruction*. PhD thesis, University of Sheffield, 2001. (Section 7.3).

M.Á. Carreira-Perpiñán. Acceleration strategies for Gaussian mean-shift image segmentation. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1160–1167, 2006.

M.Á. Carreira-Perpiñán. Gaussian mean-shift is an EM algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):767–776, 2007.

M.Á. Carreira-Perpiñán. A review of mean-shift algorithms for clustering. *arXiv preprint arXiv:1503.00687*, 2015.

Y.-C. Chen, C. R. Genovese, S. Ho, and L. Wasserman. Optimal ridge detection using coverage risk. In *Advances in Neural Information Processing Systems*, pages 316–324, 2015a.

Y.-C. Chen, C. R. Genovese, and L. Wasserman. Asymptotic theory for density ridges. *The Annals of Statistics*, 43(5):1896–1928, 2015b.

Y.-C. Chen, C.R. Genovese, R.J. Tibshirani, and L. Wasserman. Nonparametric modal regression. *The Annals of Statistics*, 44(2):489–514, 2016a.

Y.-C. Chen, C.R. Genovese, and L. Wasserman. A comprehensive approach to mode clustering. *Electronic Journal of Statistics*, 10(1):210–241, 2016b.

Y.-C. Chen, S. Ho, P.E. Freeman, C.R. Genovese, and L. Wasserman. Cosmic web reconstruction through density ridges: Method and algorithm. *Monthly Notices of the Royal Astronomical Society*, 454(1):1140–1156, 2016c.

Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

R. T. Collins. Mean-shift blob tracking through scale space. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 234–240, 2003.

D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 142–149, 2000.

D. D. Cox. A penalty method for nonparametric estimation of the logarithmic derivative of a density function. *Annals of the Institute of Statistical Mathematics*, 37(1):271–288, 1985.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, 39(1):1–38, 1977.

M. J. Drinkwater, Q. A. Parker, D. Proust, E. Slezak, and H. Quintana. The large scale distribution of galaxies in the shapley supercluster. *Publications of the Astronomical Society of Australia*, 21 (1):89–96, 2004.

T. Duong, A. Cowling, I. Koch, and M. P. Wand. Feature significance for multivariate kernel density estimation. *Computational Statistics & Data Analysis*, 52(9):4225–4242, 2008.

D. Eberly. *Ridges in Image and Data Analysis*. Springer, 1996.

J. Einbeck and G. Tutz. Modelling beyond regression functions: an application of multimodal regression to speed–flow data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 55(4):461–475, 2006.

M. Fashing and C. Tomasi. Mean shift is a bound optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):471–474, 2005.

M. Forina, C. Armanino, S. Lanteri, and E. Tiscornia. Classification of olive oils from their fatty acid composition. In *Food research and data analysis*, pages 189–214. Applied Science Publishers, London, 1983.

K. Fukumizu, F.R. Bach, and A. Gretton. Statistical consistency of kernel canonical correlation analysis. *Journal of Machine Learning Research*, 8:361–383, 2007.

K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975.

T. Gasser, P. Hall, and B. Presnell. Nonparametric estimation of the mode of a distribution of random curves. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60 (4):681–691, 1998.

C. R. Genovese, M. Perone-Pacifico, I. Verdinelli, and L. Wasserman. Nonparametric ridge estimation. *The Annals of Statistics*, 42(4):1511–1545, 2014.

C. R. Genovese, M. Perone-Pacifico, I. Verdinelli, and L. Wasserman. Non-parametric inference for density modes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78 (1):99–126, 2016.

Y. A. Ghassabeh. On the convergence of the mean shift algorithm in the one-dimensional space. *Pattern Recognition Letters*, 34(12):1423–1427, 2013.

Y. A. Ghassabeh, T. Linder, and G. Takahara. On some convergence properties of the subspace constrained mean shift. *Pattern Recognition*, 46(11):3140–3147, 2013.

F. Godtliebsen, J. S. Marron, and P. Chaudhuri. Significance in scale space for bivariate density estimation. *Journal of Computational and Graphical Statistics*, 11(1):1–21, 2002.

L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

A. Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.

A. Kain and M. W. Macon. Spectral voice conversion for text-to-speech synthesis. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 285–288, 1998.

T. Kanamori, S. Hido, and M. Sugiyama. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10:1391–1445, 2009.

T. Kanamori, T. Suzuki, and M. Sugiyama. Statistical analysis of kernel-based least-squares density-ratio estimation. *Machine Learning*, 86(3):335–367, 2012.

S. Kpotufe. Lipschitz density-ratios, structured data, and data-driven tuning. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54, pages 1320–1328, 2017.

X. Li, Z. Hu, and F. Wu. A note on the convergence of the mean shift. *Pattern recognition*, 40(6): 1756–1762, 2007.

J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, Berkeley, CA, USA, 1967. University of California Press.

V. Melnykov and R. Maitra. Finite mixture models and model-based clustering. *Statistics Surveys*, 4:80–116, 2010.

C. A. Micchelli, Y. Xu, and H. Zhang. Universal kernels. *Journal of Machine Learning Research*, 7:2651–2667, 2006.

A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, volume 14, pages 849–856, 2001.

X. Nguyen, M. J. Wainwright, and M. I. Jordan. Estimating divergence functionals and the likelihood ratio by penalized convex risk minimization. In *Advances in neural information processing systems (NIPS)w*, pages 1089–1096, 2008.

U. Ozertem and D. Erdogmus. Locally defined principal curves and surfaces. *Journal of Machine Learning Research*, 12:1249–1286, 2011.

E. Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

S. Pulkkinen. Ridge-based method for finding curvilinear structures from noisy data. *Computational Statistics & Data Analysis*, 82:89–109, 2015.

T. W. Sager and R. A. Thisted. Maximum likelihood estimation of isotonic modal regression. *The Annals of Statistics*, 10(3):690–707, 1982.

H. Sasaki, A. Hyvärinen, and M. Sugiyama. Clustering via mode seeking by direct estimation of the gradient of a log-density. In *Machine Learning and Knowledge Discovery in Databases Part III- European Conference, ECML/PKDD 2014*, volume 8726, pages 19–34, 2014.

H. Sasaki, Y. K. Noh, and M. Sugiyama. Direct density-derivative estimation and its application in KL-divergence approximation. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 809–818, 2015.

H. Sasaki, Y. Ono, and M. Sugiyama. Modal regression via direct log-density gradient estimation. In *Proceedings of the 23th International Conference on Neural Information Processing (ICONIP)*, volume 9948, pages 108–116. Springer, 2016.

H. Sasaki, T. Kanamori, and M. Sugiyama. Estimating density ridges by direct estimation of density-derivative-ratios. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54, pages 204–212, 2017.

S. J. Sheather. Density estimation. *Statistical Science*, 19(4):588–597, 2004.

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. CRC press, 1986.

B. Sriperumbudur, K. Fukumizu, A. Gretton, and A. Hyvärinen. Density estimation in infinite dimensional exponential families. *arXiv preprint arXiv:1312.3516 (ver.3)*, 2013.

B. Sriperumbudur, K. Fukumizu, A. Gretton, A. Hyvärinen, and R. Kumar. Density estimation in infinite dimensional exponential families. *Journal of Machine Learning Research*, 18(57):1–59, 2017.

I. Steinwart and A. Christmann. *Support vector machines*. Springer, 2008.

G. Strang. *Calculus*. Wellesley-Cambridge Press, 1991.

J. Su, A. Srivastava, and F.W. Huffer. Detection, classification and estimation of individual shapes in 2D and 3D point clouds. *Computational Statistics & Data Analysis*, 58:227–241, 2013.

M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in neural information processing systems (NIPS)*, pages 1433–1440, 2008.

M. Sugiyama, T. Suzuki, and T. Kanamori. *Density Ratio Estimation in Machine Learning*. Cambridge University Press, 2012.

M. Sugiyama, G. Niu, M. Yamada, M. Kimura, and H. Hachiya. Information-maximization clustering based on squared-loss mutual information. *Neural Computation*, 26(1):84–131, 2014.

W. Tao, H. Jin, and Y. Zhang. Color image segmentation based on mean shift and normalized cuts. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 37(5):1382–1389, 2007.

A. B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009.

P. D. Turney. Robust classification with context-sensitive features. In *Proceedings of the 6th international conference on Industrial and engineering applications of artificial intelligence and expert systems*, pages 268–276. Gordon & Breach Science Publishers, 1993.

J. Wang, B. Thiesson, Y. Xu, and M. Cohen. Image and video segmentation by anisotropic kernel mean shift. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 238–249, 2004.

L. Wasserman. Topological data analysis. *Annual Review of Statistics and Its Application*, 5:501–532, 2018.

H. Wendland. *Scattered data approximation*. Cambridge university press, 2004.

W. Yao, B. G. Lindsay, and R. Li. Local modal regression. *Journal of nonparametric statistics*, 24 (3):647–663, 2012.

S. You, E. Bas, D. Erdogmus, and J. Kalpathy-Cramer. Principal curved based retinal vessel segmentation towards diagnosis of retinal diseases. In *Proceedings of IEEE International Conference on Healthcare Informatics, Imaging and Systems Biology (HISB)*, pages 331–337, 2011.

D.X. Zhou. Derivative reproducing properties for kernel methods in learning theory. *Journal of Computational and Applied Mathematics*, 220(1-2):456–463, 2008.

Journal of Machine Learning Research 18 (2018) 1-18 Submitted 5/17; Revised 2/18; Published 4/18

To Tune or Not to Tune the Number of Trees in Random Forest

Philipp Probst probst@ibe.med.uni-muenchen.de
Institut für medizinische Informationsverarbeitung, Biometrie und Epidemiologie
Marchioninistr. 15, 81377 München

Anne-Laure Boulesteix boulesteix@ibe.med.uni-muenchen.de
Institut für medizinische Informationsverarbeitung, Biometrie und Epidemiologie
Marchioninistr. 15, 81377 München

Editor: Isabelle Guyon

Abstract

The number of trees T in the random forest (RF) algorithm for supervised learning has to be set by the user. It is unclear whether T should simply be set to the largest computationally manageable value or whether a smaller T may be sufficient or in some cases even better. While the principle underlying bagging is that more trees are better, in practice the classification error rate sometimes reaches a minimum before increasing again for increasing number of trees. The goal of this paper is four-fold: (i) providing theoretical results showing that the expected error rate may be a non-monotonous function of the number of trees and explaining under which circumstances this happens; (ii) providing theoretical results showing that such non-monotonous patterns cannot be observed for other performance measures such as the Brier score and the logarithmic loss (for classification) and the mean squared error (for regression); (iii) illustrating the extent of the problem through an application to a large number (n = 306) of datasets from the public database OpenML; (iv) finally arguing in favor of setting T to a computationally feasible large number as long as classical error measures based on average loss are considered.

Keywords: Random forest, number of trees, bagging, out-of-bag, error rate

1. Introduction

The random forest (RF) algorithm for classification and regression, which is based on the aggregation of a large number T of decision trees, was first described in its entirety by Breiman (2001). T is one of several important parameters which have to be carefully chosen by the user. Some of these parameters are *tuning parameters* in the sense that both too high and too low parameter values yield sub-optimal performances; see Segal (2004) for an early study on the effect of such parameters. It is unclear, however, whether the number of trees T should simply be set to the largest computationally manageable value or whether a smaller T may be sufficient or in some cases even better, in which case T should ideally be tuned carefully. This question is relevant to any user of RF and has been the topic of much informal discussion in the scientific community, but has to our knowledge never been addressed systematically from a theoretical and empirical point of view.

Breiman (2001) provides proofs of convergence for the generalization error in the case of classification random forest for growing number of trees. This means that the error rate for a given test or training dataset converges to a certain value. Moreover, Breiman (2001) proves that there exists an upper bound for the generalization error. Similarly he proves the convergence of the mean squared generalization error for regression random forests and also provides an upper bound. However, these results do not answer the question of whether the number of trees is a tuning parameter or should be set as high as computationally feasible, although convergence properties may at first view be seen as an argument in favor of a high number of trees. Breiman (1996a) and Friedman (1997) note that bagging and aggregation methods can make good predictors better but poor predictors can be transformed into worse. Hastie et al. (2001) show in a simple example that for a single observation that is incorrectly classified (in the binary case), bagging can worsen the expected missclassification rate. In Section 3.1 we will further analyse this issue and examine the outcome of aggregating performances for several observations.

Since each tree is trained individually and without knowledge of previously trained trees, however, the risk of overfitting when adding more trees discussed by Friedman (2001) in the case of boosting is not relevant here.

The number of trees is sometimes considered as a tuning parameter in current literature (Raghu et al., 2015); see also Barman et al. (2014) for a study in which different random seeds are tested to obtain better forests—a strategy implicitly assuming that a random forest with few trees may be better than a random forest with many trees. The R package `RFmarkerDetector` (Palla and Armano, 2016) even provides a function, 'tuneNTREE', to tune the number of trees. Of note, the question of whether a smaller number of trees may be better has often been discussed in online forums (see Supplementary File 1 for a non-exhaustive list of links) and seems to remain a confusing issue to date, especially for beginners.

A related but different question is whether a smaller number of trees is *sufficient* (as opposed to "better") in the sense that more trees do not improve accuracy. This question is examined, for example, in the very early study by Latinne et al. (2001) or by Hernández-Lobato et al. (2013). Another important contribution to that question is the study by Oshiro et al. (2012), which compared the performance in terms of the Area Under the ROC Curve (AUC) of random forests with different numbers of trees on 29 datasets. Their main conclusion is that the performance of the forest does not always substantially improve as the number of trees grows and after having trained a certain number of trees (in their case 128) the AUC performance gain obtained by adding more trees is minimal. The study of Oshiro et al. (2012) provides important empirical support for the existence of a "plateau", but does not directly address the question of whether a smaller number of trees may be substantially better and does not investigate this issue from a theoretical perspective, thus making the conclusions dependent on the 29 examined datasets.

In this context, the goal of our paper is four-fold: (i) providing theoretical results showing that, in the case of binary classification, the expected error rate may be a non-monotonous function of the number of trees and explaining under which circumstances this happens; (ii) providing theoretical results showing that such non-monotonous patterns cannot be observed for other performance measures such as the Brier score and the logarithmic loss (for classification) and the mean squared error (for regression); (iii) illustrating the extent of the problem through an application to a large number (n = 306) of datasets from the public database OpenML; (iv) finally arguing in favor of setting it to a computationally feasible

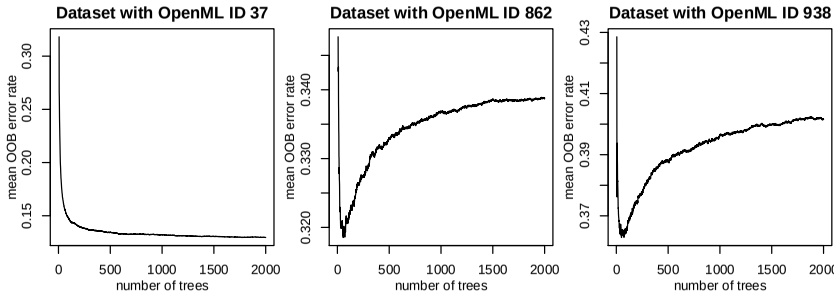


Figure 1: Mean OOB error rate curves for OpenML datasets with IDs 37, 862 and 938. The curves are averaged over 1000 independent runs of random forest.

large number as long as classical error measures based on average loss are considered. Furthermore, we introduce our new R package `OOBCurve`, which can be used to examine the convergence of various performance measures.

To set the scene, we first address this issue empirically by looking at the curve depicting the out-of-bag (OOB) error rate (see Section 2 for a definition of the OOB error) for different number of trees (also called OOB error rate curve) for various datasets from the OpenML database (Vanschoren et al., 2013). To obtain more stable results and better estimations for the expected error rate we repeat this procedure 1000 times for each dataset and average the results.

For most datasets we observe monotonously decreasing curves with growing number of trees as in the left panel of Figure 1, while others yield strange non-monotonous patterns, for example the curves of the datasets with the OpenML ID 862 and 938, which are also depicted in Figure 1. The initial error rate drops steeply before starting to increase after a certain number of trees before finally reaching a plateau.

At first view, such non-monotonous patterns are a clear argument in favor of tuning T. We claim, however, that it is important to understand why and in which circumstances such patterns happen in order to decide whether or not T should be tuned in general. In Section 3, we address this issue from a theoretical point of view, by formulating the expected error rate as a function of the probabilities ε_i of correct classification by a single tree for each observation i of the training dataset, for $i = 1, \ldots, n$ (with n denoting the size of the training dataset). This theoretical view provides a clear explanation of the non-monotonous error rate curve patterns in the case of classification. With a similar approach, we show that such non-monotonous patterns cannot be obtained with the Brier score or the logarithmic loss as performance measures, which are based on probability estimations and also not for the mean squared error in the case of regression. Only for the AUC we can see non-monotonous curves as well.

The rest of this paper is structured as follows. Section 2 gives a brief introduction into random forest and performance estimation. Theoretical results are presented in Section 3, while the results of a large empirical study based on 306 datasets from the public database OpenML are reported in Section 4. More precisely, we empirically validate our theoretical model for the error as a function of the number of trees as well as our statements regarding the properties of datasets yielding non-monotonous patterns. We finally argue in Section 5 that there is no inconvenience—except additional computational cost—in adding trees to a random forest and that T should thus not be seen as a tuning parameter as long as classical performance measures based on the average loss are considered.

2. Background: Random Forest and Measures of Performance

In this section we introduce the random forest method, the general notation and some well known performance measures.

2.1 Random Forest

The random forest (RF) is an ensemble learning technique consisting of the aggregation of a large number T of decision trees, resulting in a reduction of variance compared to the single decision trees. In this paper we consider the original version of RF first described by Breiman (2001), while acknowledging that other variants exist, for example RF based on conditional inference trees (Hothorn et al., 2006) which address the problem of variable selection bias investigated by Strobl et al. (2007). Our considerations are however generalizable to many of the available RF variants and other methods that use randomization techniques.

A prediction is obtained for a new observation by aggregating the predictions made by the T single trees. In the case of regression RF, the most straightforward and common procedure consists of averaging the prediction of the single trees, while majority voting is usually applied to aggregate classification trees. This means that the new observation is assigned to the class that was most often predicted by the T trees.

While RF can be used for various types of response variables including censored survival times or (as empirically investigated in Section 4) multicategorical variables, in this paper we mainly focus on the two most common cases, binary classification and regression.

2.2 General Notations

From now on, we consider a fixed training dataset D consisting of n observations, which is used to derive prediction rules by applying the RF algorithm with a number T of trees. Ideally, the performance of these prediction rules is estimated based on an independent test dataset, denoted as D_{test}, consisting of n_{test} test observations.

Considering the ith observation from the test dataset ($i = 1, \ldots, n_{test}$), we denote its true response as y_i, which can be either a numeric value (in the case of regression) or the binary label 0 vs. 1 (in the case of binary classification). The predicted value output by tree t (with $t = 1, \ldots, T$) is denoted as $\hat{y}_{it}$, while $\hat{y}_i$ stands for the predicted value output by the whole random forest. Note that, in the case of regression, $\hat{y}_i$ is usually obtained by

averaging as

$$\hat{y}_i = \frac{1}{T}\sum_{t=1}^{T} \hat{y}_{it}.$$

In the case of classification, $\hat{y}_i$ is usually obtained by majority voting. For binary classification, it is equivalent to computing the same average as for regression, which now takes the form

$$\hat{p}_i = \frac{1}{T}\sum_{t=1}^{T} I(\hat{y}_{it} = 1)$$

and is denoted as $\hat{p}_i$ (standing for probability), and finally deriving $\hat{y}_i$ as

$$\hat{y}_i = \begin{cases} 1 & \text{if } \hat{p}_i > 0.5, \\ 0 & \text{otherwise.} \end{cases}$$

2.3 Measures of Perfomance for Binary Classification and Regression

In regression as well as in classification, the performance of a RF for observation i is usually quantified through a so-called loss function measuring the discrepancy between the true response y_i and the predicted response $\hat{y}_i$ or, in the case of binary classification, between y_i and $\hat{p}_i$. For both regression and binary classification, the classical and most straightforward measure is defined for observation i as

$$e_i \;=\; (y_i - \hat{y}_i)^2 \;=\; L(y_i, \hat{y}_i),$$

with $L(.,.)$ standing for the loss function $L(x, y) = (x - y)^2$. In the case of regression this is simply the squared error. Another common loss function in the regression case is the absolute loss $L(x, y) = |x - y|$. For binary classification both measures simplify to $e_i = 0$ if observation i is classified correctly by the RF, $e_i = 1$ otherwise, which we will simply denote as *error* from now on. One can also consider the performance of single trees, that means the discrepancy between y_i and $\hat{y}_{it}$. We define e_{it} as

$$e_{it} = L(y_i, \hat{y}_{it}) = (y_i - \hat{y}_{it})^2$$

and the mean error—a quantity we need to derive our theoretical results on the dependence of performance measures on the number of tree T—as

$$\varepsilon_i = E(e_{it}),$$

where the expectation is taken over the possible trees conditionally on D. The term ε_i can be interpreted as the difficulty to predict y_i with single trees. In the case of binary classification, we have $(y_i - \hat{y}_{it})^2 = |y_i - \hat{y}_{it}|$ and ε_i can be simply estimated as $|y_i - \hat{p}_i|$ from a RF with a large number of trees.

In the case of binary classification, it is also common to quantify performance through the use of the Brier score, which has the form

$$b_i \;=\; (y_i - \hat{p}_i)^2 = L(y_i, \hat{p}_i)$$

or of the logarithmic loss

$$l_i \;=\; -(y_i \ln(\hat{p}_i) + (1 - y_i)\ln(1 - \hat{p}_i)).$$

Both of them are based on $\hat{p}_i$ rather than $\hat{y}_i$, and can thus be only defined for the whole RF and not for single trees.

The area under the ROC curve (AUC) cannot be expressed in terms of single observations, as it takes into account all observations at once by ranking the $\hat{p}_i$-values. It can be interpreted as the probability that the classifier ranks a randomly chosen observation with $y_i = 1$ higher than a randomly chosen observation with $y_i = 0$. The larger the AUC, the better the discrimination between the two classes. The (empirical) AUC is defined as

$$\text{AUC} = \frac{\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} S(\hat{p}_i^{\star}, \hat{p}_j^{\star\star})}{n_1 n_2},$$

where $\hat{p}_1^{\star}, ..., \hat{p}_{n_1}^{\star}$ are probability estimations for the n_1 observations with $y_i = 1$, $\hat{p}_1^{\star\star}, ..., \hat{p}_{n_2}^{\star\star}$ are probability estimations for the n_2 observations with $y_i = 0$ and $S(.,.)$ is defined as $S(p, q) = 0$ if $p < q$, $S(p, q) = 0.5$ if $p = q$ and $S(p, q) = 1$ if $p > q$. The AUC can also be interpreted as the Mann-Whitney U-Statistic divided by the product of n_1 and n_2.

2.4 Measures for Multiclass Classification

The measures defined in the previous section can be extended to the multiclass classification case. Let K denote the number of classes ($K > 2$). The response y_i takes values in $\{1, ..., K\}$. The error for observation i is then defined as

$$e_i = I(y_i \neq \hat{y}_i).$$

We denote the estimated probability of class k for observation i as

$$\hat{p}_{ik} = \frac{1}{T}\sum_{t=1}^{T} I(\hat{y}_{it} = k).$$

The logarithmic loss is then defined as

$$l_i = \sum_{k=1}^{K} -I(y_i = k)\log(\hat{p}_{ik})$$

and the generalized Brier score is defined as

$$b_i = \sum_{k=1}^{K} (\hat{p}_{ik} - I(y_i = k))^2,$$

which in the binary case is twice the value of the definition that was used in the previous section. Following Hand and Till (2001), the AUC can also be generalized to the multiclass case as

$$\text{AUC} = \frac{1}{K(K-1)}\sum_{j=1}^{K}\sum_{\substack{k=1 \\ k\neq j}}^{K} \text{AUC}(j, k),$$

where $\text{AUC}(j, k)$ is the AUC between class k and j, see also Ferri et al. (2009) for more details. It is equivalent to the definition given in Section 2.3 in the binary classification case.

2.5 Test Dataset Error vs. Out-of-Bag Error

In the cases where a test dataset D_{test} is available, performance can be assessed by averaging the chosen performance measure (as described in the previous paragraphs) over the n_{test} observations. For example the classical error rate (for binary classification) and the mean squared error (for regression) are computed as

$$\frac{1}{n_{test}} \sum_{i=1}^{n_{test}} L(y_i, \hat{y}_i),$$

with $L(x, y) = (x-y)^2$, while the mean absolute error (for regression) is obtained by defining $L(.,.)$ as $L(x, y) = |x - y|$. Note that, in the context of regression, Rousseeuw (1984) proposes to consider the median $med\,(L(y_1, \hat{y}_1), ..., L(y_{n_{test}}, \hat{y}_{n_{test}}))$, instead of averaging, which results in the median squared error for the loss function $L(x, y) = (x - y)^2$ and in the median absolute error for the loss function $L(x, y) = |x - y|$. These measures are more robust against outliers and contamination (Rousseeuw, 1984).

An alternative to the use of a test dataset is the out-of-bag error which is calculated by using the out-of-bag (OOB) estimations of the training observations. OOB predictions are calculated by predicting the class, the probability (in the classification case) or the real value (in the regression case) for each training observation i (for $i = 1, \ldots, n$) by using only the trees for which this observation was not included in the bootstrap sample (i.e., it was not used to construct the tree). Note that these predictions are obtained based on a subset of trees—including on average $T \times 0.368$ trees. These predictions are ultimately compared to the true values by calculating performance measures (see Sections 2.3, 2.4 and 2.5).

3. Theoretical Results

In this section we compute the expected performance—according to the error, the Brier score and the logarithmic loss outlined in Section 2.3—of a binary classification or regression RF consisting of T trees as estimated based on the n_{test} test observations, while considering the training dataset as fixed. For the AUC we prove that it can be a non-monotonous function in T. The case of other measures (mean absolute error, median of squared error and median of absolute error for regression) and multiclass classification is much more more complex to investigate from a theoretical point of view. It will be examined empirically in Section 4.

In this section we are concerned with *expected* performances, where expectation is taken over the sets of T trees. Our goal is to study the monotonicity of the expected errors with respect to T. The number T of trees is considered a parameter of the RF and now mentioned in parentheses everytime we refer to the whole forest.

3.1 Error Rate (Binary Classification)

We first show that for single observations the expected error rate curve can be increasing and then show exemplified how this can influence the shape of the average curve of several observations. The observation that bagging can worsen the expected error rate of a single observation was already done by Hastie et al. (2001), Breiman (1996a) and Friedman (1997). In this section we provide a general formula explaining this observation, and then extend our theoretical considerations to further performance measures in the following sections.

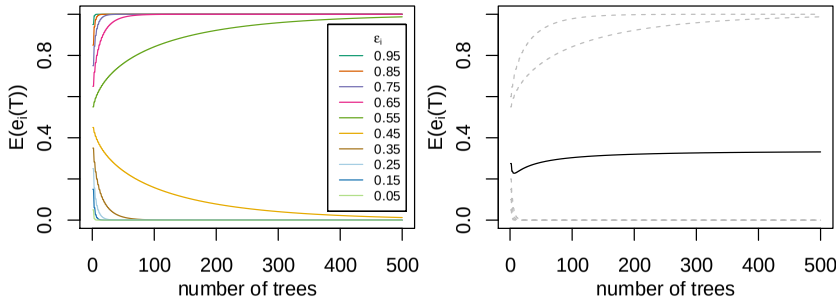


Figure 2: Left: Expected error rate curves for different ε_i values. Right: Plot of the average curve (black) of the curves with $\varepsilon_1 = 0.05$, $\varepsilon_2 = 0.1$, $\varepsilon_3 = 0.15$, $\varepsilon_4 = 0.2$, $\varepsilon_5 = 0.55$ and $\varepsilon_6 = 0.6$ (depicted in grey and dotted)

3.1.1 Theoretical Considerations

Let us first consider the classical error rate $e_i(T)$ for observation i with a RF including T trees and derive its expectation, conditionally on the training set D,

$$E(e_i(T)) \quad = \quad E\left(I\left(\frac{1}{T}\sum_{t=1}^{T} e_{it} > 0.5\right)\right) = P\left(\sum_{t=1}^{T} e_{it} > 0.5 \cdot T\right).$$

We note that e_{it} is a binary variable with $E(e_{it}) = \varepsilon_i$. Given a fixed training dataset D and observation i, the e_{it}, $t = 1, ..., T$ are mutually independent. It follows that the sum $X_i = \sum_t^T e_{it}$ follows the binomial distribution $B(T, \varepsilon_i)$. It is immediate that the contribution of observation i to the expected error rate, $P(X_i > 0.5 \cdot T)$, is an increasing function in T for $\varepsilon_i > 0.5$ and a decreasing function in T for $\varepsilon_i < 0.5$.

Note that so far we ignored the case where $\sum_{t=1}^{T} e_{it} = 0.5 \cdot T$, which may happen when T is even. In this case, the standard implementation in R (`randomForest`) assigns the observation randomly to one of the two classes. This implies that $0.5 \cdot P(\sum_{t=1}^{T} e_{it} = 0.5 \cdot T)$ has to be added to the above term, which does not affect our considerations on the ε_i's role.

3.1.2 Impact on Error Rate Curves

The error rate curve for observation i is defined as the curve described by the function $e_i : T \to \mathbb{R}$. The expectation $E(e_i(T))$ of the error rate curve for observation i with the mentioned adjustment in the case of an even number of trees can be seen in the left plot of Figure 2 for different values of ε_i. Very high and very low values of ε_i lead to rapid convergence, while for ε_i-values close to 0.5 more trees are needed to reach the plateau. The error rate curve obtained for a test dataset consists of the average of the error rate curves of the single observations. Of course, if trees are good classifiers we should have $\varepsilon_i < 0.5$ for most observations. In many cases, observations with $\varepsilon_i > 0.5$ will be compensated by

observations with $\varepsilon_i < 0.5$ in such a way that the expected error rate curve is monotonously decreasing. This is typically the case if there are many observations with $\varepsilon_i \approx 0$ and a few with $\varepsilon_i \approx 1$. However, if there are many observations with $\varepsilon_i \approx 0$ and a few observations with $\varepsilon_i \geq 0.5$ that are close to 0.5, the expected error rate curve initially falls down quickly because of the observation with $\varepsilon_i \approx 0$ and then grows again slowly as the number of trees increases because of the observations with $\varepsilon_i \geq 0.5$ close to 0.5. In the right plot of Figure 2 we can see (black solid line) the mean of the expected error rate curves for $\varepsilon_1 = 0.05$, $\varepsilon_2 = 0.1$, $\varepsilon_3 = 0.15$, $\varepsilon_4 = 0.2$, $\varepsilon_5 = 0.55$ and $\varepsilon_6 = 0.6$ (displayed as gray dashed lines) and can see exactly the non-monotonous pattern that we expected: due to the ε_i's 0.55 and 0.6 the average curve increases again after reaching a minimum. In Section 4 we will see that the two example datasets whose non-monotonous out-of-bag error rate curves are depicted in the introduction have a similar distribution of ε_i.

We see that the convergence rate of the error rate curve is only dependent on the distribution of the ε_i's of the observations. Hence, the convergence rate of the error rate curve is not directly dependent on the number of observations n or the number of features, but these characteristics could influence the empirical distribution of the ε_i's and hence possibly the convergence rate as outlined in Section 4.4.1.

3.2 Brier Score (Binary Classification) and Squared Error (Regression)

We now turn to the Brier score and compute the expected Brier score contribution of observation i for a RF including T trees, conditional on the training set D. We obtain

$$\begin{aligned} E(b_i(T)) &= E((y_i - \hat{p}_i(T))^2) = E\left(\left(y_i - \frac{1}{T}\sum_{t=1}^{T}\hat{y}_{it}\right)^2\right) \\ &= E\left(\left(\frac{1}{T}\sum_{t=1}^{T}(y_i - \hat{y}_{it})\right)^2\right) = E\left(\left(\frac{1}{T}\sum_{t=1}^{T}e_{it}\right)^2\right). \end{aligned}$$

From $E(Z^2) = E(Z)^2 + Var(Z)$ with $Z = \frac{1}{T}\sum_{t=1}^{T} e_{it}$ it follows:

$$E(b_i(T)) = E(e_{it})^2 + \frac{Var(e_{it})}{T},$$

which is obviously a strictly monotonous decreasing function of T. This also holds for the average over the observations of the test dataset. In the case of binary classification, we have $e_{it} \sim \mathcal{B}(1, \varepsilon_i)$, yielding $E(e_{it}) = \varepsilon_i$ and $Var(e_{it}) = \varepsilon_i(1-\varepsilon_i)$, thus allowing the formulation of $E(b_i(T))$ as $E(b_i(T)) = \varepsilon_i^2 + \frac{\varepsilon_i(1-\varepsilon_i)}{T}$. Note that the formula $E(b_i(T)) = E(e_{it})^2 + Var(e_{it})/T$ is also valid for the squared error in the regression case, except that in this case we would write $\hat{y}_i$ instead of $\hat{p}_i$ in the first line.

3.3 Logarithmic Loss (Binary Classification)

As outlined in Section 2.3, another usual performance measure based on the discrepancy between y_i and $\hat{p}_i$ is the logarithmic loss $l_i(T) = -(y_i \ln(\hat{p}_i(T)) + (1 - y_i)\ln(1 - \hat{p}_i(T)))$. Noticing that $\hat{p}_i(T) = 1 - \frac{1}{T}\sum_{t=1}^{T} e_{it}$ for $y_i = 1$ and $\hat{p}_i(T) = \frac{1}{T}\sum_{t=1}^{T} e_{it}$ for $y_i = 0$, it can

be in both cases $y_i = 0$ and $y_i = 1$ reformulated as

$$l_i(T) = -\ln\left(1 - \frac{1}{T}\sum_{t=1}^{T} e_{it}\right).$$

In the following we ensure that the term inside the logarithm is never zero by adding a very small value a to $1 - \frac{1}{T}\sum_{t=1}^{T} e_{it}$. The logarithmic loss $l_i(T)$ is then always defined and its expectation exists. This is similar to the solution adopted in the `mlr` package, where 10^{-15} is added in case that the inner term of the logarithm equals zero.

With $Z := 1 - \frac{1}{T}\sum_{t=1}^{T} e_{it} + a$, we can use the Taylor expansion,

$$\begin{aligned} E\left[f(Z)\right] =& E\left[f(\mu_Z + (Z - \mu_Z))\right] \\ \approx& E\left[f(\mu_Z) + f'(\mu_Z)\left(Z - \mu_Z\right) + \frac{1}{2}f''(\mu_Z)\left(Z - \mu_Z\right)^2\right] \\ =& f(\mu_Z) + \frac{f''(\mu_Z)}{2} \cdot Var(Z) = f(E(Z)) + \frac{f''(E(Z))}{2} \cdot Var(Z) \end{aligned}$$

where μ_Z stands for $E(Z)$ and $f(.)$ as $f(.) = -\ln(.)$. We have $Var(Z) = \frac{\varepsilon_i(1-\varepsilon_i)}{T}$, $E(Z) = 1 - \varepsilon_i + a$, $f(E(Z)) = -\ln(1 - \varepsilon_i + a)$ and $f''(E(Z)) = (1 - \varepsilon_i + a)^{-2}$, finally yielding

$$E(l_i(T)) \approx -\ln(1 - \varepsilon_i + a) + \frac{\varepsilon_i(1 - \varepsilon_i)}{2T(1 - \varepsilon_i + a)^2},$$

which is obviously a decreasing function of T. The Taylor approximation gets better and better for increasing T, since the variance of $l_i(T)$ decreases with increasing T and thus $l_i(T)$ tends to get closer to its expectancy.

3.4 Area Under the ROC Curve (AUC) (Classification)

For the AUC, considerations such as those we made for the error rate, the Brier score and the logarithmic loss are impossible, since the AUC is not the sum of individual contributions of the observations. It is however relatively easy to see that the expected AUC is not always an increasing function of the number T of trees. For example, think of the trivial example of a test dataset consisting of two observations with responses y_1 resp. y_2 and $E(\hat{p}_1(T)) = 0.4$ resp. $E(\hat{p}_2(T)) = 0.6$. If $y_1 = 0$ and $y_2 = 1$, the expected AUC curve increases monotonously with T, as the probability of a correct ordering according to the calculated scores $\hat{p}_1(T)$ and $\hat{p}_2(T)$ increases. However, if $y_1 = 1$ and $y_2 = 0$, we obtain a monotonously decreasing function, as the probability of a wrong ordering gets higher with increasing number of trees. It is easy to imagine that for different combinations of $E(\hat{p}_i(T))$, one can obtain increasing curves, decreasing curves or non-monotonous curves.

3.5 Adapting the Models to the OOB Error

The "OOB estimator" of the performance outlined in Section 2.5 is commonly considered as an acceptable proxy of the performance estimator obtained through the use of an independent test dataset or through resampling-techniques such as cross-validation (Breiman, 1996b) for a random forest including $T \times 0.368$ trees. Compared to these techniques, the

OOB estimator has the major advantage that it neither necessitates to fit additional random forests (which is advantageous in terms of computational resources) nor to reduce the size of the dataset through data splitting. For these reasons, we will consider OOB performance estimators in our empirical study.

However, if we consider the OOB error instead of the test error from an independent dataset, the formulas given in the previous subsections are not directly applicable. After having trained T trees, for making an OOB estimation for an observation we can only use the trees for which the observation was out-of-bag. If we take a simple bootstrap sample from the n training observation when bagging we have *on average* only $T \cdot (1 - \frac{1}{n})^n \approx T \cdot \exp(-1) \approx T \cdot 0.368$ trees for predicting the considered observation. This means that we would have to replace T by $T \cdot \exp(-1)$ in the above formulas and that the formulas are no longer exact because $T \cdot \exp(-1)$ is only an average. Nonetheless it is still a good approximation as confirmed in our benchmark experiments.

4. Empirical Results

This section shows a large-scale empirical study based on 193 classification tasks and 113 regression tasks from the public database OpenML (Vanschoren et al., 2013). The datasets are downloaded with the help of the `OpenML` R package (Casalicchio et al., 2017). The goals of this study are to (i) give an order of magnitude of the frequency of non-monotonous patterns of the error rate curve in real data settings; (ii) empirically confirm our statement that observations with ε_i greater than (but close to) 0.5 are responsible for non-monotonous patterns; (iii) analyse the results for other classification measures, the multiclass classification and several regression measures; (iv) analyse the convergence rate of the OOB curves.

4.1 Selection of Datasets

To select the datasets to be included in our study we define a set of candidate datasets—in our case the datasets available from the OpenML platform (Vanschoren et al., 2013)—and a set of inclusion criteria as recommended in Boulesteix et al. (2017). In particular, we do not select datasets with respect to the results they yield, thus warranting representativity.

Our inclusion criteria are as follows: (i) the dataset has predefined tasks in OpenML (see Vanschoren et al., 2013, for details on the OpenML nomenclature); (ii) it includes less than 1000 observations; (iii) it includes less than 1000 features. The two latter criteria aim at keeping the computation time feasible.

Cleaning procedures such as the deletion of duplicated datasets (whole datasets that appear twice in the OpenML database) are also applied to obtain a decent collection of datasets. No further modification of the tasks and datasets were done.

This procedure yields a total of 193 classification tasks and 113 regression tasks.

From the 193 classification tasks, 149 are binary classification tasks and 44 multiclass classification tasks.

The tasks contained easy, medium and difficult tasks - for binary classification tasks the mean (out-of-bag) AUC of a random forest with 2000 trees was 0.841, the minimum 0.502, the first quartile 0.732, the median 0.870, the third quartile 0.962 and the maximum 1. Similarly the regression tasks contained easy and difficult tasks with a mean R^2 of 0.559.

4.2 Study Design

For each dataset we run the RF algorithm with $T = 2000$ trees 1000 times successively with different seeds using the R package `randomForest` (Liaw and Wiener, 2002) with the default parameter settings. We choose 2000 trees because in a preliminary study on a subset of the datasets we could observe that convergence of the OOB curves was reached within these 2000 trees. Note that all reported results regarding the performance gain and convergence are made with the out-of-bag predictions. As for these predictions on average only $\exp(-1) \cdot T$ of the T trees are used, the convergence of independent test data is faster by the factor 2.7. For the classification tasks we calculate the OOB curves for the error rate, the balanced error rate, the (multiclass) Brier score, the logarithmic loss and the (multiclass) AUC using our new package `OOBCurve`, see details in the next section.

For the regression tasks we calculate the OOB curves using the mean squared error, the mean absolute error, the median squared error and the median of absolute error as performance measures. We parallelize the computations using the R package `batchtools` (version 0.9.0) (Lang et al., 2017). For each measure and each dataset, the final curve is obtained by simply averaging over the 1000 runs of RF. We plot each of them in three files separetely for binary classification, multiclass classification and regression. In the plots the x-axis starts at $T = 11$ since overall performance estimates are only defined if each observation was out-of-bag in at least one of the T trees, which is not always the case in practice for $T < 10$. We plot the curves only until $T = 500$, as no interesting patterns can be observed after this number of trees (data not shown). The graphics, the R-codes and the results of our experiment can be found on `https://github.com/PhilippPro/tuneNtree`.

4.3 The R Package `OOBCurve`

The calculation of out-of-bag estimates for different performance measures is implemented in our new R package `OOBCurve`. More precisely, it takes a random forest constructed with the R package `randomForest` (Liaw and Wiener, 2002) or `ranger` (Wright, 2016) as input and can calculate the OOB curve for any measure that is available from the `mlr` package (Bischl et al., 2016). The `OOBCurve` package is available on CRAN R package repository and also on Github (`https://github.com/PhilippPro/OOBCurve`). It is also possible to calculate OOB curves of other hyperparameters of RF such as `mtry` with this package.

4.4 Results for Binary Classification

The average gain in performance in the out-of-bag performance for 2000 trees instead of 11 trees is -0.0324 for the error rate, -0.0683 for the brier score, -2.383 for the logarithmic loss and 0.0553 for the AUC. In the following we will concentrate on the visual analysis of the graphs and are especially interested in the results of the error rate.

4.4.1 Overall Results for the OOB Error Rate Curves

We observe in the graphs of the OOB error rate curves that for most datasets the curve is quickly decreasing until it converges to a dataset-specific plateau value. In 16 cases which make approximately 10% of the datasets, however, the curve grows again after reaching its lowest value, leading to a value at 2000 trees that is by at least 0.005 bigger than the

lowest value of the OOB error rate curve for $T \in [10, 250]$. This happens mainly for smaller datasets, where a few observations can have a high impact on the error curve. Of these 16 cases 15 belong to the smaller half of the datasets—ordered by the number of observations multiplied with the number of features. The mean increase of these 16 datasets was 0.020 (median: 0.012). The difference in mean and median is mainly caused by one outlier where the increase was around 0.117.

4.4.2 Datasets with Non-Monotonous OOB Error Rate Curve

We now examine in more detail the datasets yielding non-monotonous patterns. In particular, the histograms of the estimates $\hat{\varepsilon}_i = |y_i - \hat{p}_i|$ of the observation-specific errors ε_i are of interest, since our theoretical results prove that the distribution of the ε_i determines the form of the expected error rate curve. To get these histograms we compute the estimates $\hat{\varepsilon}_i$ of the observation-specific errors ε_i (as defined in Section 2.3) from a RF with a big number $T = 100000$: the more trees, the more accurate the estimates of ε_i.

The histograms for the exemplary datasets considered in the introduction (see Figure 1) are displayed in Figure 3. A typical histogram for an OOB curve with monotonously decreasing error rate curve is displayed in the left panel. The heights of the bins of this histogram of the $\hat{\varepsilon}_i$ are monotonously decreasing from 0 to 1.

The histograms for the non-monotonous error rate curves from the introduction can be seen in the middle (OpenML ID 862) and right (OpenML ID 938) panels of Figure 3. In both cases we see that a non-negligible proportion of observations have ε_i larger than but close to 0.5. This is in agreement with our theoretical results. With growing number of trees the chance that these observations are incorrectly classified increases, while the chance for observations with $\varepsilon_i \approx 0$ is already very low—and thus almost constant. Intuitively we expect such shapes of histograms for datasets with few observations—where by chance the shape of the histogram of the $\hat{\varepsilon}_i$ could look like in our two examples. For bigger datasets we expect smoother shapes of the histogram, yielding strictly decreasing error rate curves.

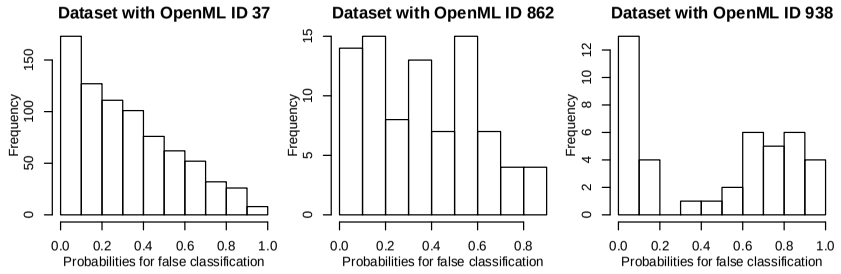


Figure 3: Histograms of the estimates of ε_i $(i = 1, \ldots, n)$ from random forests with 100000 trees for dataset with IDs 36, 862 and 938

	error rate		Brier score		logarithmic loss		AUC	
error rate	**1.00**	(1.00)	**0.28**	(0.44)	**0.27**	(0.45)	**-0.18**	(-0.43)
Brier score	**0.72**	(0.86)	**1.00**	(1.00)	**0.96**	(0.98)	**-0.63**	(-0.87)
logarithmic loss	**0.65**	(0.84)	**0.93**	(0.95)	**1.00**	(1.00)	**-0.63**	(-0.87)
AUC	**-0.64**	(-0.85)	**-0.84**	(-0.95)	**-0.81**	(-0.92)	**1.00**	(1.00)

Table 1: Linear (bottom-left) and rank (top-right) correlation results for binary classification datasets and for multiclass classification (in brackets)

4.4.3 Other Measures

For the Brier score and the logarithmic loss we observe, as expected, monotonically decreasing curves for all datasets. The expected AUC curve usually appears as a growing function in T. In a few datasets such as the third binary classification example (OpenML ID 905), however, it falls after reaching a maximum.

To assess the similarity between the different curves, we calculate the Bravais-Pearson linear correlation and Kendall's τ rank correlation between the values of the OOB curves of the different performance measures and average these correlation matrices over all datasets. Note that we do not perform any correlation tests, since the assumption of independent identically distributed observations required by these tests is not fulfilled: our correlation analyses are meant to be explorative. The results can be seen in Table 1. The Brier score and logarithmic loss have the highest correlation. They are also more correlated to the AUC than to the error rate, which has the lowest correlation to all other measures.

4.5 Results for Multiclass Classification

The average gain in out-of-bag performance for 2000 trees instead of 11 trees is -0.0753 for the error rate, -0.1282 for the brier score, -5.3486 for the logarithmic loss and 0.0723 for the AUC. These values are higher than the ones from binary classification. However, the visual observations we made for the binary classification also hold for the multiclass classification. For 5 of the 44 datasets the minimum error rate for $T \in [11; 250]$ is lower by more than 0.005 than the error rate for $T = 2000$. In contrast to the binary classification case, 3 of these 5 datasets belong to the bigger half of the datasets. The results for the correlation are quite similar, although the correlation (see Table 1) is in general slightly higher than in the binary case.

4.6 Results for Regression

The average performance gain regarding the out-of-bag performance of the R^2 for 2000 trees compared to 11 trees is 0.1249. In the OOB curves for regression we can observe the monotonously decreasing pattern expected from theory in the case of the most widely used mean squared error (mse). The mean absolute error (mae) is also strictly decreasing for all the datasets considered in our study.

For the median squared error (medse) and the median absolute error (medae), we get a performance gain by using 2000 trees instead of 10 in most but not all cases (around 80% of the datasets). In many cases (around 50%) the minimum value for $T \in [11; 250]$ is smaller

than the value for $T = 2000$ which means that growing more trees is rather disadvantagous in these cases in terms of medse and medae. This could be explained by the fact that each tree in a random forest tries to minimize the squared error in the splits and therefore adding more trees to the forest will improve the mean squared error but not necessarily measures that use the median. More specifically, one could imagine that the additional trees focus on the reduction of the error for outlying observations at the price of an increase of the median error. In a simulated dataset (linear model with 200 observations, 5 relevant features and 5 non-relevant features drawn from a multivariate normal distribution) we could observe this pattern (data not shown). Without outlier all expected curves are strictly decreasing. When adding an outlier (changing the outcome of one observation to a very big value) the expected curves of mse and mae are still strictly decreasing, while the expected curves of medse and medae show are increasing for higher T. The curves of the measures which take the mean of the losses of all observations have a high linear and rank correlation (> 0.88), as well as the curves of the measures which take the median of the losses (> 0.97). Correlation between these two groups of measures are lower, around 0.5 for the linear correlation coefficient and around 0.2 for the rank correlation coefficient.

4.7 Convergence

It is clearly visible from the out-of-bag curves (`https://github.com/PhilippPro/tuneNtree/tree/master/graphics`) that increasing the number of trees yields a substantial performance gain in most of the cases, but the biggest performance gain in the out-of-bag curves can be seen while growing the first 250 trees. Setting the number of trees from 10 to 250 in the binary classification case provides an average decrease of 0.0306 of the error rate and an increase of 0.0521 of the AUC. On the other hand, using 2000 trees instead of 250 does not yield a big performance gain, the average error rate improvement is only 0.0018 (AUC: 0.0032). The improvement in the multiclass case is bigger with an average improvement of the error rate of 0.0739 (AUC: 0.0665) from 10 trees to 250 and an average improvement of 0.0039 (AUC: 0.0057) for using 2000 trees instead of 250. For regression we have an improvement of 0.1210 of the R^2 within the first 250 trees and an improvement of 0.0039 for using 2000 trees instead of 250. These results are concordant with a comment by Breiman (1996a) (Section 6.2) who notes that fewer bootstrap replicates are necessary when the outcome is numerical and more are required for an increasing number of classes.

5. Conclusions and Extensions

In this section we draw conclusions of the given results and discuss possible extensions.

5.1 Assessment of the Convergence

For the assessment of the convergence in the classification case we generally recommend using measures other than the error rate, such as AUC, the Brier score or the logarithmic loss for which the OOB curves are much more similar as we have seen in our correlation analysis. Their convergence rate is not so dependent on observations with ε_i close to 0.5 (in the binary classification case), and they give an indication of the general stability of the probability estimations of all observations. This can be especially important if the threshold

for classification is not set a priori to 0.5. The new `OOBCurve` R package is a tool to examine the rate of convergence of the trained RF with any measure that is available in the `mlr` R package. It is important to remember that for the calculation of the OOB error curve at T only $\exp(-1) \cdot T$ trees are used. Thus, as far as future independent data is concerned, the convergence of the performances is by $\exp(1) \approx 2.7$ faster than observed from our OOB curves. Having this in mind, our observations (see Section 4.7) are in agreement with the results of Oshiro et al. (2012), who conclude that after growing 128 trees no big gain in the AUC performance could be achieved by growing more trees.

5.2 Why More Trees Are Better

Non-monotonous expected error rate curves observed in the case of binary classification might be seen as an argument in favour of tuning the number T of trees. Our results, however, suggest that tuning is not recommendable in the case of classification. Firstly, non-monotonous patterns are observed only with some performance measures such as the error rate and the AUC in case of classification. Measures such as the Brier score or the logarithmic loss, which are based on probabilities rather than on the predicted class and can thus be seen as more refined, do not yield non-monotonous patterns, as theoretically proved in Section 3 and empirically observed based on a very large number of datasets in Section 4. Secondly, non-monotonous patterns in the expected error rate curves are the result of a particular rare combination of ε_i's in the training data. Especially if the training dataset is small, the chance is high that the distribution of the ε_i will be different for independent test data, for example values of ε_i close to but larger than 0.5 may not be present. In this case, the expected error rate curve for this independent future dataset would not be non-monotonous, and a large T is better. Thirdly, even in the case of non-monotonic expected error rate curves, the minimal error rate value is usually only slightly smaller than the value at convergence (see Section 4.4.1). We argue that this very small gain - which, as outlined above, is relevant only for future observations with $\varepsilon_i > 0.5$ - probably does not compensate the advantage of using more trees in terms of other performance measures or in terms of the precision of the variable importance measures, which are very commonly used in practice.

In the case of regression, our theoretical results show that the expected out-of-bag mse curve is monotonously decreasing. For the mean absolute error the empirical results suggest the same. In terms of the less common measures *median* squared error and *median* absolute error (as opposed to *mean* losses), however, performance may get worse with increasing number of trees. More research is needed.

5.3 Extensions

Note that our theoretical results are not only valid for random forest but generalizable to any ensemble method that uses a randomization technique, since the fact that the base learners are trees and the specific randomization procedure (for example bagging) do not play any role in our proofs. Our theoretical results could possibly be extended to the multiclass case, as supported by our results obtained with 44 multiclass datasets.

Although we claim that increasing the number of trees cannot harm noticeably as far as measures based on average loss are considered, our empirical results show that for most of the examined datasets, the biggest performance gain is achieved when training the first

100 trees. However, the rate of convergence may be influenced by other hyperparameters of the RF. For example lower sample size while taking bootstrap samples for each tree, bigger constraints on the tree depth or more variables lead to less correlated trees and hence more trees are needed to reach convergence.

One could also think of an automatic break criterion which stops the training automatically according to the convergence of the OOB curves. For example, training could be stopped if the last T_{last} trees did not improve performance by more than Δ, where T_{last} and Δ are parameters that should be fixed by the user as a compromise between performance and computation time. Note that, if variable importances are computed, it may be recommended to also consider their convergence. This issue also requires more research.

Acknowledgments

We would like to thank Alexander Dürre for useful comments on the approximation of the logarithmic loss and Jenny Lee for language editing.

References

Ranjan Kumar Barman, Sudipto Saha, and Santasabuj Das. Prediction of interactions between viral and host proteins using supervised machine learning methods. *PLOS ONE*, 9(11):1–10, 2014.

Bernd Bischl, Michel Lang, Lars Kotthoff, Julia Schiffner, Jakob Richter, Erich Studerus, Giuseppe Casalicchio, and Zachary M. Jones. mlr: Machine learning in R. *Journal of Machine Learning Research*, 17(170):1–5, 2016. R package version 2.9.

Anne-Laure Boulesteix, Rory Wilson, and Alexander Hapfelmeier. Towards evidence-based computational statistics: lessons from clinical research on the role and design of real-data benchmark studies. *BMC Medical Research Methodology*, 17(1):138, 2017.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996a.

Leo Breiman. Out-of-bag estimation. *Technical report, Statistics Department, University of California 1996*, 1996b.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Giuseppe Casalicchio, Jakob Bossek, Michel Lang, Dominik Kirchhoff, Pascal Kerschke, Benjamin Hofner, Heidi Seibold, Joaquin Vanschoren, and Bernd Bischl. OpenML: An R package to connect to the machine learning platform OpenML. *Computational Statistics*, 32(3):1–15, 2017.

César Ferri, José Hernández-Orallo, and R Modroiu. An experimental comparison of performance measures for classification. *Pattern Recognition Letters*, 30(1):27–38, 2009.

Jerome H Friedman. On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1(1):55–77, 1997.

Jerome H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232, 2001.

David J Hand and Robert J Till. A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning*, 45(2):171–186, 2001.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

Daniel Hernández-Lobato, Gonzalo Martínez-Muñoz, and Alberto Suárez. How large should ensembles of classifiers be? *Pattern Recognition*, 46(5):1323–1336, 2013.

Torsten Hothorn, Kurt Hornik, and Achim Zeileis. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15 (3):651–674, 2006.

Michel Lang, Bernd Bischl, and Dirk Surmann. batchtools: Tools for R to work on batch systems. *The Journal of Open Source Software*, 2(10), 2017.

Patrice Latinne, Olivier Debeir, and Christine Decaestecker. Limiting the number of trees in random forests. In *International Workshop on Multiple Classifier Systems*, pages 178–187. Springer, 2001.

Andy Liaw and Matthew Wiener. Classification and regression by randomForest. *R News*, 2(3):18–22, 2002. R package version 4.6-12.

Thais Mayumi Oshiro, Pedro Santoro Perez, and José Augusto Baranauskas. How many trees in a random forest? In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 154–168. Springer, 2012.

Piergiorgio Palla and Giuliano Armano. *RFmarkerDetector: Multivariate Analysis of Metabolomics Data using Random Forests*, 2016. R package version 1.0.1.

Arvind Raghu, Praveen Devarsetty, Peiris David, Tarassenko Lionel, and Clifford Gari. Implications of cardiovascular disease risk assessment using the who/ish risk prediction charts in rural india. *PLOS ONE*, 10(8):1–13, 2015.

Peter J. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880, 1984.

Mark R Segal. Machine learning benchmarks and random forest regression. *Center for Bioinformatics & Molecular Biostatistics*, 2004.

Carolin Strobl, Anne-Laure Boulesteix, Achim Zeileis, and Torsten Hothorn. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, 8(1):25, 2007.

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. OpenML: Networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013.

Marvin N. Wright. *ranger: A Fast Implementation of Random Forests*, 2016. R package version 0.6.0.

Journal of Machine Learning Research 18 (2018) 1-26 Submitted 6/17; Revised 1/18; Published 4/18

Divide-and-Conquer for Debiased l_1-norm Support Vector Machine in Ultra-high Dimensions

Heng Lian HENGLIAN@CITYU.EDU.HK
Department of Mathematics
City University of Hong Kong
Kowloon Tong, Hong Kong

Zengyan Fan ZENGYANFAN@HOTMAIL.COM
Department of Statistics and Applied Probability
National University of Singapore
Singapore

Editor: Jie Peng

Abstract

1-norm support vector machine (SVM) generally has competitive performance compared to standard 2-norm support vector machine in classification problems, with the advantage of automatically selecting relevant features. We propose a divide-and-conquer approach in the large sample size and high-dimensional setting by splitting the data set across multiple machines, and then averaging the *debiased* estimators. Extension of existing theoretical studies to SVM is challenging in estimation of the inverse Hessian matrix that requires approximating the Dirac delta function via smoothing. We show that under appropriate conditions the aggregated estimator can obtain the same convergence rate as the central estimator utilizing all observations.

Keywords: classification, debiased estimator, distributed estimator, divide and conquer, sparsity

1. Introduction

The support vector machine (SVM) is a widely used tool for classification (Vapnik, 2013; Scholkopf and Smola, 2001; Cristianini and Shawe-Taylor, 2000). Although the original motivation of Cortes and Vapnik (1995) is in terms of finding a maximum-margin hyperplane, its equivalent formulation as a regularized functional optimization problem is perhaps more easily understood by statisticians and more amenable for statistical asymptotic analysis. In the standard formulation the penalized functional is a sum of the hinge loss plus an l_2-norm regularization term. Statistical properties of the SVM, especially its nonlinear version using general kernels, has been studied in a lot of works recently including but not limited to Bartlett et al. (2006); Blanchard et al. (2008); Lin (2000, 2004); Steinwart and Scovel (2007); Steinwart (2005); Zhang (2004). In this work, we focus on penalized linear SVM with large sample size and large dimension, with particular emphasis on dealing with distributed estimation in such contexts.

Data sets with thousands of features have become increasingly common recently in many real-world applications. For example, a microarray data set typically contains more than 10,000 genes. A drawback of standard SVM based on l_2-norm penalty is that it can be adversely affected if many redundant variables are included in building the decision rule. A modern approach to feature selection is based on the idea of shrinkage. This approach involves fitting a model involving all p predictors. However, the estimated coefficients are shrunk towards zero. In particular with appropriate choice of penalty some of the coefficients may be estimated to be exactly zero. Automatic variable selection using penalized estimation that can shrink some coefficients to be exactly zero was pioneered in Tibshirani (1996) using an l_1-norm penalty (or called lasso penalty). Other penalties proposed include those in Fan and Li (2001), Zou (2006) and Zhang (2010).

The idea of using l_1 norm to automatically select variables has been extended to classification problems. van de Geer (2008) analyzed lasso penalized estimator for generalized linear models which include logistic regression as a special case. Zhu et al. (2003) proposed the l_1-norm support vector machine and oracle properties of SCAD-penalized support vector machines were established in Park et al. (2012), based on the Bahadur representation of Koo et al. (2008). See also the earlier work of Bradley and Mangasarjan (1998); Song et al. (2002). When feature dimension is larger than the sample size, Peng et al. (2016); Zhang et al. (2016) obtained the convergence rate of the SCAD and lasso-penalized estimators for SVM, respectively. Our work follows the lead of these works on understanding the statistical properties of the estimated SVM coefficients, instead of on generalization error rates or empirical risk.

In this paper, we focus on distributed estimation of l_1 penalized linear SVM coefficients using multiple computing machines. The simplest and most popular approach in data parallelism is averaging: each machine uses a part of the data and obtains a local estimator using the standard estimation methods and sends it back to the master machine which combines the local estimators by simple averaging into an aggregated estimator. In the classical regime concerning fixed dimensional problems, this has been advocated in McDonald et al. (2009), and was also studied by Zinkevich et al. (2010); Zhang et al. (2013, 2015); Balcan et al. (2015); Zhao et al. (2016). In all these studies, the typical outcome of asymptotic analysis is that under suitable assumptions, in particular that the number of machines are not excessive compared to the sample size, the aggregated estimator enjoys the same or similar statistical properties as the centralized estimator obtained by a single machine using all observations (if the centralized estimator can be feasibly obtained). Such results convincingly illustrate that the divide-and-conquer strategy works in the big data world. In the high dimensional regime, for lasso penalized estimators, there is a well-known bias-variance trade-off. When the tuning parameter in the penalty is chosen optimally in each local machine, the size of bias and standard deviation are of the same order. Aggregation can decrease the variance thanks to the magic of central limit theorem or some related finite-sample bounds for averages of mean zero random variables, but it cannot decrease the bias in general. Thus debiasing becomes crucial to reduce the bias to a smaller order before aggregation. This is done for sparse linear regression in Lee et al. (2017), which shows the debiased estimator in van de Geer et al. (2014) works satisfactorily for parameter inferences. Lee et al. (2017) studied both the least squares estimator and the more general M-estimator with *smooth* loss functions using an l_1 (LASSO) penalty and applied it to distributed estimation. Our study of linear SVM coefficients differs significantly from Lee et al. (2017) due to the unsmooth nature of the hinge loss function. In particular, estimation of the

Hessian matrix which involves Dirac delta function requires a smoothing procedure, which is nontrivial to analyze.

The rest of the paper is organized as follow. After a brief introduction of some notations below, we consider debiased l_1-norm SVM in Section 2.1. Although the main focus is on distributed estimation, we need to first consider statistical properties of debiased estimator on a single machine, which requires a lengthy and detailed analysis. Once this is done, properties of the aggregated estimator are relatively easy to establish, as is done in Section 2.2. In terms of l_∞ norm, the aggregated estimator has the convergence rate $O_p(\sqrt{\log p/N})$ when the number of features is p and the total sample size is N under appropriate assumptions. However, its convergence rate in l_1 or l_2 norm is unacceptably larger, which motivated a further thresholding step in Section 2.3. Section 3 report some numerical results to demonstrate the finite sample performance of the proposed estimators. Finally, we conclude this paper with a discussion in Section 4.

Notations. For a vector $\mathbf{a} = (a_1, \ldots, a_n)^{\mathrm{T}}$, $\|\mathbf{a}\|_\infty = \max_j |a_j|$, $\|\mathbf{a}\|_1 = \sum_j |a_j|$, $\|\mathbf{a}\| = (\sum_j a_j^2)^{1/2}$ and $\|\mathbf{a}\|_0$ is the number of nonzero components of $\mathbf{a}$. For a matrix $\mathbf{A} = \{a_{ij}\}_{i,j=1}^n$, $\|\mathbf{A}\|_\infty = \max_{i,j} |a_{ij}|$, $\|\mathbf{A}\|_1 = \sum_{i,j} |a_{ij}|$ and $\|\mathbf{A}\|_{L_1} = \max_i \sum_j |a_{ij}|$. Throughout the paper, C denotes a generic constant that may assume different values even on the same line.

2. Divide-and-conquer for l_1-SVM

2.1 Debiased l_1-SVM

We begin with the basic setup of SVM for binary classification. We observe a simple random sample $(\mathbf{x}_i, y_i), i = 1, \ldots, N$, from an unknown distribution $P(\mathbf{x}, y)$. Here $y_i \in \{-1, 1\}$ is the class label and $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^{\mathrm{T}}$ is the p-dimensional features. For simplicity of presentation, we do not use any special treatment for the intercept, although the intercept term is typically not shrunk in l_1-SVM. The standard linear SVM estimates the parameters by solving

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} N^{-1} \sum_{i=1}^N L(y_i, \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|^2,$$

where L is the hinge loss function $L(y, t) = \max\{0, 1 - yt\}$ and λ is the regularization parameter which changes with N (typically converging to zero as N goes to infinity), but we suppress its dependence on N in our notation. Throughout the paper we make the mild assumption that

$$\log N = O(\log p).$$

This does not mean $p \geq N$, but exclude the case that p is fixed. This restriction is mainly to make the notation slightly simpler. Without this restriction, Theorem 1 below still hold with $\log p$ replaced by $\log(\max\{p, N\})$, and the probability $1 - p^{-C}$ replaced by $1 - (\max\{p, N\})^{-C}$.

Variable selection is of particular interest when p is large compared to N, due to its ability to avoid overfitting as well as to enhance interpretation. The l_1-SVM (Zhu et al., 2003) estimates the parameter by solving

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} N^{-1} \sum_{i=1}^N L(y_i, \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_1. \tag{1}$$

The l_1 penalty here encourages sparsity of the solution (Tibshirani, 1996, 1997).

Let $\boldsymbol{\beta}_0 = (\beta_{01}, \ldots, \beta_{0p})^{\mathrm{T}}$ be the true parameter, which is defined as the minimizer of the population hinge loss,

$$\boldsymbol{\beta}_0 = \arg\min_{\boldsymbol{\beta}} E[L(y, \mathbf{x}^{\mathrm{T}} \boldsymbol{\beta})]. \tag{2}$$

We assume $\boldsymbol{\beta}_0$ exists and is unique. Koo et al. (2008) provided some regularity conditions under which $\boldsymbol{\beta}_0$ is unique and $\boldsymbol{\beta}_0 \neq \mathbf{0}$. Towards variable selection in SVM, it is natural to assume $\boldsymbol{\beta}_0$ is sparse. Let $A = \{1 \leq j \leq p : \beta_{0j} \neq 0\}$ be the support set of $\boldsymbol{\beta}_0$ with $s = |A|$ the cardinality of A.

As calculated rigorously in Koo et al. (2008), the gradient vector and the Hessian matrix of the population hinge loss in Equation 2 is given by

$$S(\boldsymbol{\beta}) \quad = \quad -E[I\{y\mathbf{x}^{\mathrm{T}} \boldsymbol{\beta} \leq 1\} \mathbf{x} y]$$

and

$$\mathbf{H}(\boldsymbol{\beta}) = E[\delta(1 - y\mathbf{x}^{\mathrm{T}} \boldsymbol{\beta}) \mathbf{x}\mathbf{x}^{\mathrm{T}}],$$

respectively, where $\delta(.)$ is the Dirac delta function. Let f and g be the conditional density of $\mathbf{x}$ given $y = 1$ and $y = -1$, respectively.

(A1) The densities f and g are bounded and continuously differentiable with bounded partial derivatives, with compact support. x_j's are bounded random variables. Without loss of generality, we assume the distribution of $\mathbf{x}$ has a support contained in $[0, 1]^p$.

Under assumption (A1), $\mathbf{H}(\boldsymbol{\beta})$ is well-defined and continuous in $\boldsymbol{\beta}$.

Due to the penalty term, the penalized estimator is generally biased (i.e. shrunk towards zero). Conceptually, λ controls the trade-off between bias and standard deviation of the estimator. While averaging will reduce the standard deviation of the estimator, it generally cannot reduce the bias. Thus it is important to apply a debiasing mechanism before we aggregate estimators from different machines. For simplicity of presentation, for now we focus on the properties of the debiased estimator using all observations and later we will argue (almost trivially) these properties hold for the local estimates, uniformly over M machines. In this subsection, with $M = 1$, we have $N = n$ where n is the sample size on a single machine.

Let $\widehat{\boldsymbol{\beta}}$ be the penalized estimator obtained from Equation 1. It is known that $\widehat{\boldsymbol{\beta}}$ satisfies the Karush-Kuhn-Tucker (KKT) conditions:

$$\frac{1}{N} \sum_{i=1}^N \mathbf{x}_i L_t(y_i, \mathbf{x}_i^{\mathrm{T}} \widehat{\boldsymbol{\beta}}) + \lambda \kappa = 0 \tag{3}$$

where $L_t(y, t)$ is a sub-derivative of $L(y, t)$ with respect to t, and $\kappa = (\kappa_1, \ldots, \kappa_p)^{\mathrm{T}}$ with $\kappa_j = \mathrm{sign}(\widehat{\beta}_j)$ if $\widehat{\beta}_j \neq 0$ and $\kappa_j \in [-1, 1]$ if $\widehat{\beta}_j = 0$.

When the loss is twice differentiable, a simple Taylor's expansion can be used to expand $L_t(y_i, \mathbf{x}_i^{\mathrm{T}} \widehat{\boldsymbol{\beta}})$ at $\boldsymbol{\beta}_0$ as in van de Geer et al. (2014). For the nonsmooth loss function here, we

need to use empirical processes techniques. Let $G_n = \sqrt{n}(P_n - P)$ be the empirical process, where P is the population distribution of $(\mathbf{x}, y)$ and P_n is the empirical distribution of the observations. Informally, when $\widehat{\boldsymbol{\beta}}$ is close to $\boldsymbol{\beta}_0$, $G_n(\mathbf{x}\{L_t(y, \mathbf{x}^{\mathrm{T}}\widehat{\boldsymbol{\beta}}) - L_t(y, \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0)\})$ is small, and thus

$$\begin{aligned} & \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i L_t(y_i, \mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}) \\ \approx\ & \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i L_t(y_i, \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0) + E\mathbf{x}L_t(y, \mathbf{x}^{\mathrm{T}}\widehat{\boldsymbol{\beta}}) - E\mathbf{x}L_t(y, \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 \\ \approx\ & \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i L_t(y_i, \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0) + \mathbf{H}(\boldsymbol{\beta}_0)(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0). \end{aligned} \tag{4}$$

Then

$$\begin{aligned} 0 &= \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i L_t(y_i, \mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}) + \lambda\kappa \\ &\approx \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i L_t(y_i, \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0) + \mathbf{H}(\boldsymbol{\beta}_0)(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0) + \lambda\kappa \\ &= \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i L_t(y_i, \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0) + \mathbf{H}(\boldsymbol{\beta}_0)(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0) - \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i L_t(y_i, \mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}), \end{aligned}$$

where we used Equation 3 in both the first and the last inequality, and this leads to

$$\widehat{\boldsymbol{\beta}} \approx \boldsymbol{\beta}_0 + [\mathbf{H}(\boldsymbol{\beta}_0)]^{-1}\frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i L_t(y_i, \mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}) - [\mathbf{H}(\boldsymbol{\beta}_0)]^{-1}\frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i L_t(y_i, \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0),$$

if $\mathbf{H}(\boldsymbol{\beta}_0)$ is invertible. Since the last term in the right hand side above has mean zero, we are motivated to define the debiased estimator as

$$\widetilde{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}} - [\mathbf{H}(\boldsymbol{\beta}_0)]^{-1}\frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i L_t(y_i, \mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}), \tag{5}$$

and we set $L_t(y,t) = -yI\{yt \le 1\}$. However, $\mathbf{H}(\boldsymbol{\beta}_0)$ is unknown in two aspects. On one hand, the true parameter $\boldsymbol{\beta}_0$ is unknown and should be replaced by its estimator, say $\widehat{\boldsymbol{\beta}}$. On the other hand, $\mathbf{H}(\boldsymbol{\beta}) = E[\delta(1 - y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})\mathbf{x}\mathbf{x}^{\mathrm{T}}]$ is an expectation which should be approximated by samples. Although expectations are usually easily estimated by a simple moment estimator, this is not the case here, since there may not even be a single sample that satisfies exactly $y_i\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta} = 1$ and $\delta(.)$ as a generalized function should be treated carefully. Finally, after $\mathbf{H}(\boldsymbol{\beta})$ is approximated by samples, high-dimensionality means that the usual algebraic inverse of the estimator may not be well-defined and some approximate inverse must be used.

Thus it seems one major component of the estimation is the approximation of $[\mathbf{H}(\boldsymbol{\beta}_0)]^{-1}$ and we deal with this problem first. To motivate an estimator of $\mathbf{H}(\boldsymbol{\beta})$, we can start from

its antiderivative $S(\boldsymbol{\beta}) = E[-I\{y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta} \le 1\}\mathbf{x}y]$. For any given $\boldsymbol{\beta}$, $S(\boldsymbol{\beta})$ can be approximated by $-N^{-1}\sum_{i=1}^{N} I\{y_i\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta} \le 1\}\mathbf{x}_i\mathbf{y}_i$. If this were differentiable, we could use its derivative as an estimator of $\mathbf{H}(\boldsymbol{\beta})$. This observation motivates us to smooth the indicator function using some cumulative distribution function, say Q, and approximates $S(\boldsymbol{\beta})$ by $-N^{-1}\sum_{i=1}^{N} Q((1 - y_i\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta})/h)\mathbf{x}_i y_i$. When the bandwidth parameter h is sufficiently small, $Q(./h)$ will approximate $I\{. \ge 0\}$ well. Assuming Q is differentiable, then it is natural to approximate $\mathbf{H}(\boldsymbol{\beta})$ by

$$\widehat{\mathbf{H}}(\boldsymbol{\beta}) = N^{-1}\sum_{i=1}^{N}(1/h)q((1 - y_i\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta})/h)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}},$$

where $q(.)$ is the density of the distribution Q (derivative of Q), and thus $\mathbf{H}(\boldsymbol{\beta}_0)$ is estimated by $\widehat{\mathbf{H}}(\widehat{\boldsymbol{\beta}})$ where $\widehat{\boldsymbol{\beta}}$ is the l_1-SVM estimator.

Since the rank of $\widehat{\mathbf{H}}(\widehat{\boldsymbol{\beta}})$ is at most N, $\widehat{\mathbf{H}}(\widehat{\boldsymbol{\beta}})$ is singular when p is larger than N. Even when p is smaller than N and $\widehat{\mathbf{H}}(\widehat{\boldsymbol{\beta}})$ is non-singular, the standard inverse $[\widehat{\mathbf{H}}(\widehat{\boldsymbol{\beta}})]^{-1}$ is often not a good estimator of $\mathbf{H}(\boldsymbol{\beta}_0)$ when p is diverging with N. An approximate inverse of $\mathbf{H}(\boldsymbol{\beta}_0)$ can be found via an approach similar to that used in Cai et al. (2011) as

$$\begin{aligned} &\widehat{\boldsymbol{\Theta}} = \arg\min \|\boldsymbol{\Theta}\|_1 \\ &\text{subject to } \|\boldsymbol{\Theta}\widehat{\mathbf{H}}(\widehat{\boldsymbol{\beta}}) - \mathbf{I}\|_\infty \le Cb_N, \end{aligned} \tag{6}$$

for some tuning parameter $b_N \to 0$. Note that Cai et al. (2011) would use a slightly different constraint on $\|\widehat{\mathbf{H}}(\widehat{\boldsymbol{\beta}})\boldsymbol{\Theta} - \mathbf{I}\|_\infty$, while our constraint is more convenient in the current context since we *pre-multiply* the gradient of the loss by $[\mathbf{H}(\boldsymbol{\beta}_0)]^{-1}$ in Equation 5. We note that the constraint $\|\boldsymbol{\Theta}\widehat{\mathbf{H}}(\widehat{\boldsymbol{\beta}}) - \mathbf{I}\|_\infty \le Cb_N$ is obviously the same as $\|\boldsymbol{\Theta}_{j\cdot}\widehat{\mathbf{H}}(\widehat{\boldsymbol{\beta}}) - \mathbf{e}_j^{\mathrm{T}}\|_\infty \le Cb_N, \forall j$, where $\boldsymbol{\Theta}_{j\cdot}$ is the j-th row of $\boldsymbol{\Theta}$ (as a row vector) and $\mathbf{e}_j$ is the unit vector with j-th componenet 1. Thus the optimization problem can be solved row by row. This actually was noted in Cai et al. (2011) in their Lemma 1 (since their constraint is $\|\widehat{\mathbf{H}}(\widehat{\boldsymbol{\beta}})\boldsymbol{\Theta} - \mathbf{I}\|_\infty \le Cb_N$, their problem can be solved column by column).

Before proceeding, we impose some additional assumptions.

(A2) $\boldsymbol{\beta}_0 \ne \mathbf{0}$ and without loss of generality we assume $\beta_{01} = \max_{1\le j\le p}|\beta_{0j}|$.

(A3) $\|\boldsymbol{\Theta}_0\|_{L_1} \le C_N$, where $\boldsymbol{\Theta}_0 = [\mathbf{H}(\boldsymbol{\beta}_0)]^{-1}$.

(A4) $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\|_1 \le Cs\sqrt{\frac{\log p}{N}}$ with probability at least $1 - p^{-C}$, where $s = |\mathrm{supp}\{\boldsymbol{\beta}_0\}|$ is the number of nonzero entries in $\boldsymbol{\beta}_0$.

(A5) The density q is an even function, twice continuously differentiable, with $\int x^2 q(x)dx < \infty$, $\int q^2(x)dx < \infty$, $\int (q')^2(x)dx < \infty$, $\sup_x q'(x) < \infty$, $\sup_x q''(x) < \infty$, where q' and q'' are the first two derivatives of q.

(A6) $\|\widehat{\boldsymbol{\beta}}\|_0 \le K$ with probability at least $1 - p^{-C}$.

Assumption (A2) is mild. Koo et al. (2008) gives sufficient conditions that guarantee $\boldsymbol{\beta}_0 \ne \mathbf{0}$. To simplify the bounds below one can think of β_{01} as bounded away from zero so that it will disappear from the bounds. Note that β_{01} is the largest nonzero coefficient. In the

literature of sparse regression, it is often assumed that the *smallest* nonzero coefficient is large enough so that it can be distinguished from zero coefficients, which is a totally different assumption. Cai et al. (2011) assumed that their inverse Hessian matrix has a bounded L_1 norm (such an assumption is obviously related to sparsity of the matrix), which motivated our assumption (A3). We allow C_N in (A3) to be diverging for slightly greater generality. In particular, this mean we need to have a control on the l_1 norm of the rows of the inverse Hessian matrix. Due to that l_1 norm is a convex relaxation of the l_0 norm, we call such matrix as approximately sparse. Again, it is probably easier for the reader to regard C_N as a fixed constant. We further discuss (A3) in detail in Appendix B. Theorem 4 of Peng et al. (2016) showed $\|\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}_0\| = O_p(\sqrt{s\log p/N})$ which together with their Lemma 2 implies $\|\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}_0\|_1 = O_p(s\sqrt{\log p/N})$ as in (A4). It is also easy to choose a density that satisfies (A5), such as the standard normal density, which will be used in our numerical studies. In (A6), we assume the estimator is sufficiently sparse. This can be guaranteed in several different ways. First, we conjecture it could be proved that $\|\widehat{\boldsymbol{\beta}}\|_0 = O_p(s)$ for SVM coefficient using a similar strategy as for Theorem 3 of Belloni and Chernozhukov (2011), although the details looks quite lengthy. Second, one could add a thresholding step similar to what we will use in subsection 2.3 later to get a sparse estimator. Finally, we could add an constraint $\|\boldsymbol{\beta}\| \le K$ to the lasso problem. Such a constrained penalized problem was also proposed in Fan and Lv (2013); Zheng et al. (2014). In any case, one could expect that K is of the same order as s, the sparsity of $\boldsymbol{\beta}_0$.

We first state several propositions whose proof is left to Appendix A. The first proposition considers the accuracy bound of $\widehat{\boldsymbol{\Theta}}$ as an approximation of the inverse of $\mathbf{H}(\boldsymbol{\beta}_0)$. The second proposition shows that the first approximation in Equation 4 is sufficiently accurate based on the empirical processes results. Finally, the third proposition establishes a Lipschitz property of the Hessian matrix which implies that the second approximation in Equation 4 is sufficiently accurate.

Proposition 1 *Under assumptions (A1)-(A5), with probability at least* $1-p^{-C}$,

$$\|\widehat{\boldsymbol{\Theta}}\widehat{\mathbf{H}}(\widehat{\boldsymbol{\beta}}) - \mathbf{I}\|_\infty \le Cb_N,$$
$$\|\widehat{\boldsymbol{\Theta}}\mathbf{H}(\boldsymbol{\beta}_0) - \mathbf{I}\|_\infty \le Cb_N,$$
$$\textit{and}$$
$$\|\widehat{\boldsymbol{\Theta}}\|_{L_1} \le C_N,$$

when we set $b_N = C_N\left(\left(\frac{1}{\beta_{01}} + \sqrt{\frac{\log p}{Nh^3\beta_{01}}} + \frac{\log p}{Nh^2}\right)s\sqrt{\frac{\log p}{N}} + \frac{s^2\log p}{Nh^3} + \frac{h}{\beta_{01}^2} + \sqrt{\frac{\log p}{Nh\beta_{01}}}\right)$.

Proposition 2 *Under assumptions (A1)-(A6), with probability at least* $1-p^{-C}$,

$$\left\|\frac{1}{N}\sum_i y_i\mathbf{x}_i(I\{y_i\mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}} \le 1\} - I\{y_i\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0 \le 1\}) - Ey\mathbf{x}(I\{y\mathbf{x}^{\mathrm{T}}\widehat{\boldsymbol{\beta}} \le 1\} - I\{y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 \le 1\})\right\|_\infty$$
$$\le Ca_N,$$

where $a_N = \left(\left(\frac{s}{\beta_{01}}\sqrt{\frac{\log p}{N}}\right)^{1/2}\sqrt{\frac{K\log p}{N}} + \frac{K\log p}{N}\right)$.

Proposition 3 *(Local Lipschitz property of* $\mathbf{H}(\boldsymbol{\beta})$*) Under assumptions (A1) and (A2) and in addition* $2\|\boldsymbol{\beta}-\boldsymbol{\beta}_0\|_1 \le \beta_{01} := \max_j|\beta_{0j}|$*, we have*

$$\|\mathbf{H}(\boldsymbol{\beta}) - \mathbf{H}(\boldsymbol{\beta}_0)\|_\infty \le \frac{C}{\beta_{01}^3}\|\boldsymbol{\beta}_0\|_1\|\boldsymbol{\beta}-\boldsymbol{\beta}_0\|_1.$$

Now we derive a finite-sample bound for $\|\widetilde{\boldsymbol{\beta}}-\boldsymbol{\beta}_0\|_\infty$.

Theorem 1 *Under assumptions (A1)-(A6) and that* $s\sqrt{\log p/N} = o(\beta_{01})$*, we have*

$$\|\widetilde{\boldsymbol{\beta}}-\boldsymbol{\beta}_0\|_\infty \le C\left(C_N\left(a_N + \sqrt{\frac{\log p}{N}} + \frac{\|\boldsymbol{\beta}_0\|_1}{\beta_{01}^3}\frac{s^2\log p}{N}\right) + b_N s\sqrt{\frac{\log p}{N}}\right)$$

with probability at least $1-p^{-C}$.

Proof of Theorem 1. We have

$$\begin{aligned}
&\widetilde{\boldsymbol{\beta}}-\boldsymbol{\beta}_0\\
=\;& \widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}_0 + \widehat{\boldsymbol{\Theta}}\left\{\frac{1}{N}\sum_i y_i\mathbf{x}_i I\{y_i\mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}} \le 1\}\right\}\\
=\;& (\mathbf{I}-\widehat{\boldsymbol{\Theta}}\mathbf{H}(\boldsymbol{\beta}_0))(\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}_0) + \widehat{\boldsymbol{\Theta}}\mathbf{H}(\boldsymbol{\beta}_0)(\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}_0) + \widehat{\boldsymbol{\Theta}}\left\{\frac{1}{N}\sum_i y_i\mathbf{x}_i I\{y_i\mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}} \le 1\}\right\}\\
=\;& (\mathbf{I}-\widehat{\boldsymbol{\Theta}}\mathbf{H}(\boldsymbol{\beta}_0))(\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}_0)\\
&+\widehat{\boldsymbol{\Theta}}\Big\{\frac{1}{N}\sum_i y_i\mathbf{x}_i I\{y_i\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0 \le 1\} + E[y\mathbf{x}I\{y\mathbf{x}^{\mathrm{T}}\widehat{\boldsymbol{\beta}} \le 1\}] - E[y\mathbf{x}I\{y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 \le 1\}]\\
&\qquad +\mathbf{a}_N + \mathbf{H}(\boldsymbol{\beta}_0)(\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}_0)\Big\},
\end{aligned} \tag{7}$$

where $\mathbf{a}_N = \frac{1}{N}\sum_i y_i\mathbf{x}_i(I\{y_i\mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}} \le 1\} - I\{y_i\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0 \le 1\}) - Ey\mathbf{x}(I\{y\mathbf{x}^{\mathrm{T}}\widehat{\boldsymbol{\beta}} \le 1\} - I\{y_i\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0 \le 1\})$ with $\|\mathbf{a}_N\|_\infty \le a_N$ with probability $1-p^{-C}$.

Using Proposition 1, the first term above is bounded by $\|\mathbf{I}-\widehat{\boldsymbol{\Theta}}\mathbf{H}(\boldsymbol{\beta}_0)\|_\infty\|\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}_0\|_1 \le b_N s\sqrt{\log p/N}$ with probability at least $1-p^{-C}$. We also have

$$\begin{aligned}
&\left\|E[y\mathbf{x}I\{y\mathbf{x}^{\mathrm{T}}\widehat{\boldsymbol{\beta}} \le 1\}] - E[y\mathbf{x}I\{y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 \le 1\}] + \mathbf{H}(\boldsymbol{\beta}_0)(\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}_0)\right\|_\infty\\
=\;& \|(\mathbf{H}(\boldsymbol{\beta}^*) - \mathbf{H}(\boldsymbol{\beta}_0))(\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}_0)\|_\infty\\
\le\;& \|\mathbf{H}(\boldsymbol{\beta}^*) - \mathbf{H}(\boldsymbol{\beta}_0)\|_\infty\|\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}_0\|_1\\
\le\;& \frac{C}{\beta_{01}^3}\|\boldsymbol{\beta}_0\|_1\|\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}_0\|_1^2,
\end{aligned}$$

where $\boldsymbol{\beta}^*$ lies between $\boldsymbol{\beta}_0$ and $\widehat{\boldsymbol{\beta}}$. Furthermore, using Hoeffding's inequality and the union bound,

$$P\left(\left\|\frac{1}{N}\sum_i y_i\mathbf{x}_i I\{y_i\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0 \le 1\}\right\|_\infty > t\right) \le 2p\exp\{-CNt^2\}.$$

and thus

$$\left\| \frac{1}{N} \sum_i y_i x_i I\{y_i \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta}_0 \le 1\} \right\|_\infty \le C\sqrt{\frac{\log p}{N}},$$

with probability at least $1 - p^{-C}$.

Finally, combining the various bounds above and using that for any vector $\mathbf{a}$, $\|\widehat{\boldsymbol{\Theta}}\mathbf{a}\|_\infty \le \|\widehat{\boldsymbol{\Theta}}\|_{L_1}\|\mathbf{a}\|_\infty$, we can get that the second term of Equation 7 is bounded by a constant multiple of $C_n\left(a_N + \sqrt{\frac{\log p}{N}} + \frac{\|\boldsymbol{\beta}_0\|_1}{\beta_{01}^3}\frac{s^2\log p}{N}\right)$ with probability at least $1 - p^{-C}$. ■

In l_∞ norm, based on Theorem 1, we can see that under reasonable assumptions (see for example corollary 1) the debiased estimator has the convergence rate $\sqrt{\log p/N}$, which is the dominating term in the bound. However, in terms of l_1 or l_2 norm, since $\widetilde{\boldsymbol{\beta}}$ is non-sparse, we generally have $\|\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\|_1 = O_p(p\sqrt{\log p/N})$ and $\|\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\| = O_p(\sqrt{p\log p/N})$, which is much larger than the bounds for the centralized estimator $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\|_1 = O_p(s\sqrt{\log p/N})$ and $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\| = O_p(\sqrt{s\log p/N})$ (Peng et al., 2016), where s is the number of nonzero components in $\boldsymbol{\beta}_0$. Post-processing using thresholding can be used to address this problem, which we will consider after we discuss distributed estimation next.

2.2 Distributed estimation

We now consider distributed estimation, in which the whole data set is evenly distributed to M machines, with M possibly diverging with N. The size of the data at each local machine is $n = N/M$, assumed to be an integer for simplicity. On each machine m, $1 \le m \le M$, we use the local data to obtain $\widehat{\boldsymbol{\beta}}^{(m)}$, $\widehat{\boldsymbol{\Theta}}^{(m)}$, and the debiased estimator $\widetilde{\boldsymbol{\beta}}^{(m)}$. Finally, the aggregated estimator is defined by

$$\bar{\boldsymbol{\beta}} = \frac{1}{M}\sum_{m=1}^M \widetilde{\boldsymbol{\beta}}^{(m)}.$$

Theorem 2 *Under assumptions (A1)-(A6) (with N replaced by n, and in (A4) and (A6) $\widehat{\boldsymbol{\beta}}$ replaced by $\widehat{\boldsymbol{\beta}}^{(m)}$, $m = 1, \ldots, M$), and that $s\sqrt{\log p/n} = o(\beta_{01})$, we have*

$$\|\bar{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\|_\infty \le C\left(C_n\left(a_n + \sqrt{\frac{\log p}{N}} + \frac{\|\boldsymbol{\beta}_0\|_1}{\beta_{01}^3}\frac{s^2\log p}{n}\right) + b_n s\sqrt{\frac{\log p}{n}}\right),$$

with probability at least $1 - p^{-C}$, where $a_n = \left(\left(\frac{s}{\beta_{01}}\sqrt{\frac{\log p}{n}}\right)^{1/2}\sqrt{\frac{K\log p}{n}} + \frac{K\log p}{n}\right)$ and $b_n = C_n\left(\left(\frac{1}{\beta_{01}} + \sqrt{\frac{\log p}{nh^3\beta_{01}}} + \frac{\log p}{nh^2}\right)s\sqrt{\frac{\log p}{n}} + \frac{s^2\log p}{nh^3} + \frac{h}{\beta_{01}^2} + \sqrt{\frac{\log p}{nh\beta_{01}}}\right)$

Proof of Theorem 2. Let $D_m \subset \{1, \ldots, N\}$ with cardinality $|D_m| = n$ be the indices of the sub-data-set distributed to machine m. We have

$$\begin{aligned}
&\bar{\boldsymbol{\beta}} - \boldsymbol{\beta}_0 \\
= \;& \frac{1}{M}\sum_{m=1}^M (\mathbf{I} - \widehat{\boldsymbol{\Theta}}^{(m)}\mathbf{H}(\boldsymbol{\beta}_0))(\widehat{\boldsymbol{\beta}}^{(m)} - \boldsymbol{\beta}_0) \\
&+\frac{1}{M}\sum_{m=1}^M \widehat{\boldsymbol{\Theta}}^{(m)}\left\{\frac{1}{n}\sum_{i\in D_m} y_i\mathbf{x}_i I\{y_i\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta} \le 1\}\right\} \\
&-\frac{1}{M}\sum_{m=1}^M \widehat{\boldsymbol{\Theta}}^{(m)}\left\{E\left[y\mathbf{x}\left(I\{y\mathbf{x}^{\mathrm{T}}\widehat{\boldsymbol{\beta}}^{(m)} \le 1\} - I\{y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 \le 1\}\right)\right] + \mathbf{H}(\boldsymbol{\beta}_0)(\widehat{\boldsymbol{\beta}}^{(m)} - \boldsymbol{\beta}_0)\right\} \\
&-\frac{1}{M}\sum_{m=1}^M \widehat{\boldsymbol{\Theta}}^{(m)}\mathbf{a}_n^{(m)},
\end{aligned}$$

where $\mathbf{a}_n^{(m)} = \frac{1}{n}\sum_{i\in D_m} y_i\mathbf{x}_i(I\{y_i\mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}} \le 1\} - I\{y_i\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0 \le 1\}) - Ey\mathbf{x}(I\{y\mathbf{x}^{\mathrm{T}}\widehat{\boldsymbol{\beta}} \le 1\} - I\{y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 \le 1\})$ with $\|\mathbf{a}_n\|_\infty \le a_n$ with probability at least $1 - p^{-C}$.

For terms other than the second one above, the proof is exactly the same as for the proof of Theorem 1. Note that for each machine m, all derived inequalities hold with probability at least $1 - p^{-C}$ (note C can be chosen to be arbitrarily large) and thus they hold with probability at least $1 - Mp^{-C}$ simultaneously for all M machines. Since we assumed $\log M \le \log N = O(\log p)$, $1 - Mp^{-C}$ can again be written as $1 - p^{-C}$ (with a different C). For the second term above, the difference from the calculations in Theorem 1 is that here $\widehat{\boldsymbol{\Theta}}^{(m)}$ is different for different m. Let $\mathbf{e}_j \in \mathbb{R}^p$ be the unit vector with a single one for the j-th entry and let $a_{ij} = y_i\mathbf{e}_j^{\mathrm{T}}\widehat{\boldsymbol{\Theta}}^{(m)}\mathbf{x}_i$ if $i \in D_m$. Note $|a_{ij}| = |\mathbf{e}_j^{\mathrm{T}}\widehat{\boldsymbol{\Theta}}^{(m)}\mathbf{x}_i| \le \|\mathbf{e}_j\|_\infty\|\widehat{\boldsymbol{\Theta}}^{(m)}\|_{L_1}\|\mathbf{x}_i\|_\infty \le CC_n$ with probability at least $1 - p^{-C}$. By Hoeffding's inequality, we have

$$\begin{aligned}
& P\left(\left|\frac{1}{M}\sum_{m=1}^M \mathbf{e}_j^{\mathrm{T}}\widehat{\boldsymbol{\Theta}}^{(m)}\left\{\frac{1}{n}\sum_{i\in D_m} y_i\mathbf{x}_i I\{y_i\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0 \le 1\}\right\}\right| > t\right) \\
= \;& P\left(\left|\frac{1}{N}\sum_{m\le M, i\in D_m} a_{ij} I\{y_i\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0 \le 1\}\right| > t\right) \\
\le \;& 2\exp\left\{-CC_n^{-2}Nt^2\right\}. \qquad (8)
\end{aligned}$$

Thus

$$\left\|\frac{1}{M}\sum_{m=1}^M \widehat{\boldsymbol{\Theta}}^{(m)}\left\{\frac{1}{n}\sum_{i\in D_m} y_i\mathbf{x}_i I_{\{y_i\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0 \le 1\}}\right\}\right\|_\infty \le CC_n\sqrt{\frac{\log p}{N}},$$

with probability at least $1 - p^{-C}$. ■

Under reasonable assumptions, the dominating term in the bounds in Theorem 2 is $\sqrt{\log p/N}$. One version of such assumptions is presented below without proof, since it is based on simple algebra.

Corollary 1 *Assume the same conditions as in Theorem 2. In addition, we assume* $s, K, \|\boldsymbol{\beta}_0\|$ *are bounded,* β_{01} *is bounded away from zero,* $h \sim n^{-1/5}$, $\log p/n^{2/5} \to 0$ *and* $M^3 = O(N/\log p)$, *then*

$$\|\bar{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\|_\infty \leq C\sqrt{\frac{\log p}{N}},$$

with probability at least $1 - p^{-C}$.

Remark 1 *In our case,* M *can scale like* $(N/\log p)^{1/3}$ *while for the smooth loss consider in Lee et al. (2017),* M *can scale as* $(N/\log p)^{1/2}$. *This is mainly due to that for the unsmooth loss function, the empirical process as in Proposition 2 has a slower rate. In particular, in Proposition 2 the derived bound in terms of* N *scales as* $N^{-3/4}$. *For smooth loss, this term would have been* N^{-1}, *which eventually leads to the constraint that* M *should scale like* $N^{1/3}$ *instead of* $N^{1/2}$. *Although we are not claiming the bound of Proposition 2 is optimal, it is common to see that for unsmooth functions the empirical process converges slower than that for smooth functions (Belloni and Chernozhukov, 2011).*

2.3 Thresholding aggregated estimator

As mentioned in subsection 2.1, the l_2 norm of $\bar{\boldsymbol{\beta}} - \boldsymbol{\beta}_0$ is generally unfavorably large compared to the centralized estimator using all observations. This is also illustrated in our simulations. To improve performance, thresholding can be used as a post-processing step which produced a sparse aggregate estimator. Let c be a threshold level. We define $\bar{\boldsymbol{\beta}}^c = (\bar{\beta}_1^c, \ldots, \bar{\beta}_p^c)^{\mathrm{T}}$ where $\bar{\beta}_j^c = \bar{\beta}_j I\{|\bar{\beta}_j| > c\}$. Here for illustration we used hard thresholding and similar results hold for soft thresholding. Under appropriate choice of the threshold, the thresholded estimator has the same convergence rate as the centralized estimator in l_1 and l_2 norm, when choosing $c \asymp \sqrt{\log p/N}$.

Theorem 3 *On the event* $c > \|\bar{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\|_\infty$, *we have* $\|\bar{\boldsymbol{\beta}}^c - \boldsymbol{\beta}_0\|_\infty \leq 2c$, $\|\bar{\boldsymbol{\beta}}^c - \boldsymbol{\beta}_0\|_1 \leq 2sc$ *and* $\|\bar{\boldsymbol{\beta}}^c - \boldsymbol{\beta}_0\| \leq 2\sqrt{s}c$.

Proof of Theorem 3. Using $\|\bar{\boldsymbol{\beta}}^c - \boldsymbol{\beta}_0\|_\infty \leq \|\bar{\boldsymbol{\beta}}^c - \bar{\boldsymbol{\beta}}\|_\infty + \|\bar{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\|_\infty \leq 2c$ giving the first result. Since $c > \|\bar{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\|_\infty$, we have $\bar{\boldsymbol{\beta}}_j^c = 0$ if $\boldsymbol{\beta}_{0j} = 0$ and thus the support of $\bar{\boldsymbol{\beta}}^c$ is contained in that of $\boldsymbol{\beta}_0$. This implies $\|\bar{\boldsymbol{\beta}}^c - \boldsymbol{\beta}_0\|_1 \leq s\|\bar{\boldsymbol{\beta}}^c - \boldsymbol{\beta}_0\|_\infty \leq 2sc$ and $\|\bar{\boldsymbol{\beta}}^c - \boldsymbol{\beta}_0\| \leq \sqrt{s}\|\bar{\boldsymbol{\beta}}^c - \boldsymbol{\beta}_0\|_\infty \leq 2\sqrt{s}c$. ■

3. Simulations

We illustrate the performances of the distributed estimators of the linear SVM coefficients via simulations. We generate the data from the following model. First $y_i, i = 1, \ldots, N$ are generated from the binary distribution $P(y_i = 1) = P(y_i = -1) = 0.5$. Given $y_i = 1$, $\mathbf{x}_i$ is generated from a multivariate normal distribution with mean $\boldsymbol{\mu} = (0.2, -0.2, 0.3, 0.4, 0.5, 0, \ldots, 0)^{\mathrm{T}}$ and covariance matrix $\boldsymbol{\Sigma} = (\sigma_{jj'})$ with $\sigma_{jj} = 1$ for all j, $\sigma_{jj'} = 0.2$ if $j \leq 5, j' \leq 5, j \neq j'$, $\sigma_{jj'} = 0$ otherwise. Given $y_i = -1$, $\mathbf{x}_i$ is generated from a multivariate normal distribution with mean $-\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. By the calculations in Appendix B of Peng et al. (2016), the true parameter can be found to be $\boldsymbol{\beta}_0 = (0.217, -0.503, 0.397, 0.573, 0.750, 0, \ldots, 0)^{\mathrm{T}}$.

The tuning parameters λ in the penalty and the bound Cb_n used in finding the matrix inverse are selected by 5-fold cross-validation in each local machine. For the threshhold c, we choose c such that the number of nonzero components of $\bar{\boldsymbol{\beta}}$ is equal to the maximum number of nonzero components in the M local estimates. The bandwidth h is another tuning parameter. Kato (2012) has derived the optimal bandwidth for quantile regression, but it is hard to see whether similar results can be obtained in the current setting. Thus we have used Silverman's rule of thumb for kernel density estimation $h = 1.06\widehat{\sigma}n^{-1/5}$ where $\widehat{\sigma}$ is the sample standard deviation of $1 - y_i\mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}$. In our context this rule of thumb has no theoretical support, but seems to work well in practice. In our case, it seems we do not need to estimate the Hessian optimally since our purpose is not to perform inferences of $\boldsymbol{\beta}$. Finally, we use standard normal density as the smoothing kernel q.

We compute the centralized estimator (CE), the naive aggregated estimator without using bias correction (NAE), the aggregated estimator after debiasing (AE), and the final thresholded estimator (TE). The accuracy of the estimators are assessed by the l_∞ error ($\|\boldsymbol{\beta} - \boldsymbol{\beta}_0\|_\infty$), the l_2 error ($\|\boldsymbol{\beta} - \boldsymbol{\beta}_0\|$), as well as the prediction error based on independently generated $50,000$ observations.

First, we set $N = 20,000$, $M = 1, 5, 10, 15, 20, 25, 30$ ($M = 1$ is the centralized estimator) and $p = 5000$. Figure 1 shows errors of the estimators that change with M, based on 100 data sets generated in each scenario. The performances generally deteriorate with the increase of M. In terms of l_∞ error the effect of thresholding is very small if any, and both TE and AE (almost identical) are better than NAE. In terms of l_2 error, AE is much worse than NAE. This is due to that AE is non-sparse and the summation of small errors over p variables can be very large. On the other hand, although NAE may have large errors on the nonzero coefficients due to the large bias, the error is small on many zero coefficients which are estimated exactly as zero (NAE is sparse). Thresholding is effective in reducing the l_2 error as well as prediction error.

In the second set of simulations, we still use $p = 5000$ and consider different sample sizes $N = 10000, 20000, 30000, 40000, 50000$, and fix the number of samples in each local machine to be $n = 5000$ (and thus the number of machines M increases with N from 2 to 10). For this simulation, as suggested by a reviewer, we also compute the thresholded estimator (after debiasing and aggregation) using the true Hessian (the true Hessian can be computed as explained in Appendix B). From the reported results in Figure 2, it is seen that the proposed estimator TE has errors decreasing with total sample size, while the errors of the naive aggregated estimator are much larger. For AE which is non-sparse, its performance in terms of l_2 and prediction error is the worst among different estimators. It is also seen that the thresholded estimator using the true Hessian has similar performances as TE.

The simulations are carried out on the computational cluster Katana in the University of New South Wales. For the second set of simulations for example, the central estimators require from 4 to 29 hours to compute depending on the sample size, to finish all 100 repetitions, while the distributed estimator requires about 2 hours for all sample sizes.

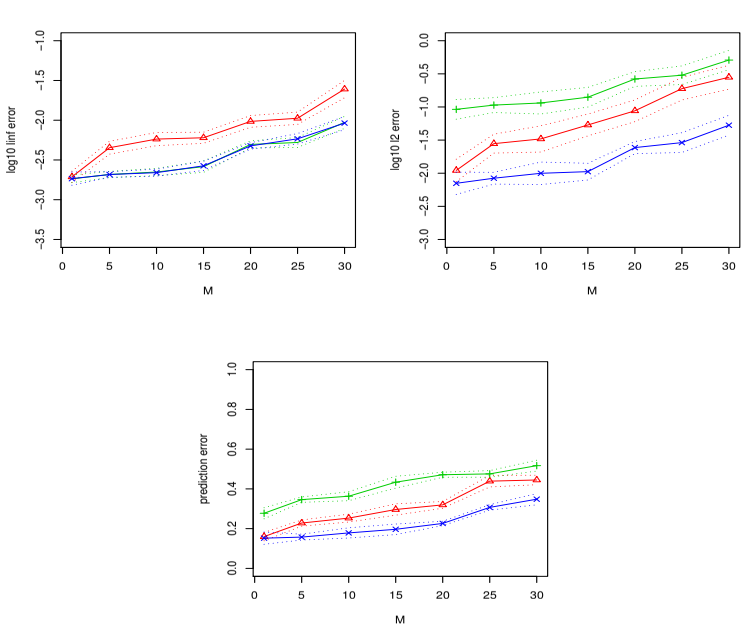


Figure 1: The l_∞, l_2 and prediction errors of estimates with $M \in \{1, 5, 10, 15, 20, 25, 30\}$ ($M = 1$ represents the centralized estimator). $\triangle$(red): naive aggregated estimator (NAE); +(green): the aggregated estimator after debiasing (AE); $\times$(blue): the thresholded estimator (TE). The l_∞ and l_2 errors are in the logarithmic scale with base 10. The dotted lines are computed based on the 100 repetitions showing 2 standard deviations of the estimated error.

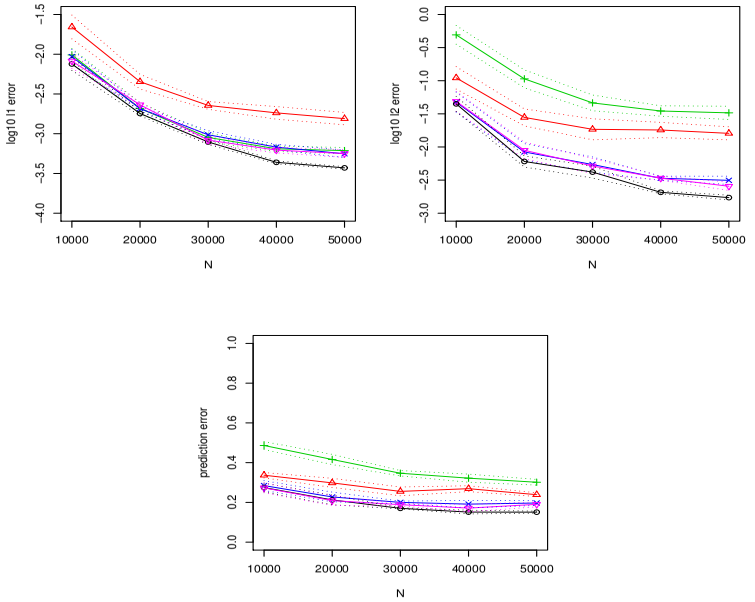


Figure 2: The l_∞ and l_2 errors of estimates with $p = 5000$ and $N \in \{10000, 20000, 30000, 40000, 50000\}$. $\circ$(black): centralized estimator (CE); $\triangle$(red): naive aggregated estimator (NAE); +(green): the aggregated estimator after debiasing (AE); $\times$(blue): the thresholded estimator (TE). $\triangledown$(purple): the thresholded estimator when the true Hessian is used in debiasing. The l_∞ and l_2 errors are in the logarithmic scale with base 10. The dotted lines are computed based on the 100 repetitions showing 2 standard deviations of the estimated error.

4. Conclusion

In this paper, we consider distributed estimation of l_1-penalized linear SVM. As long as the number of machines is not too large, the distributed estimator has the same convergence rate as the centralized estimator in l_∞, l_1 and l_2 norms, if the estimator is thresholded to retain sparsity.

We note that the optimization problem of Equation 6 can be solved row by row, and thus can also be done in a distributed way. Using local data, each local machine can obtain estimates for p/M rows of $\mathbf{\Theta}$ and then these estimates can be combined to obtain a *single* estimate of $\mathbf{\Theta}$ that satisfies $\|\widehat{\mathbf{\Theta}}\mathbf{H}(\boldsymbol{\beta}_0)-\mathbf{I}\|_\infty \le Cb_n$ with probability at least $1-p^{-C}$, where $b_n = C_n\left(\left(\frac{1}{\beta_{01}}+\sqrt{\frac{\log p}{nh^3\beta_{01}}}+\frac{\log p}{nh^2}\right)s\sqrt{\frac{\log p}{n}}+\frac{s^2\log p}{nh^3}+\frac{h}{\beta_{01}^2}+\sqrt{\frac{\log p}{nh\beta_{01}}}\right)$ and the same bound as in Theorem 2 for $\|\bar{\boldsymbol{\beta}}-\boldsymbol{\beta}_0\|_\infty$ hold with minor modifications of the proofs. For ease of implementation, we do not investigate this alternative in our numerical studies.

Once an aggregated estimator of $\boldsymbol{\beta}$ is obtained, one can use this estimator in the evaluation of the inverse of $\mathbf{H}(\widehat{\boldsymbol{\beta}})$ in each local machine. This iterative approach requires further communications among the central machine and local machines, and we do not observe improved performances of the iterative estimator empirically.

A few problems remain open in this study, among possibly many others. First, there is a gap in theory and simulation in that (A1) assumed that the covariates are bounded while the simulations are based on a multivariate normal distribution for covariates. We used the assumption of bounded covariates for at least two reasons. One reason is that we rely on the results of Peng et al. (2016), who assumed boundedness of predictors. The second reason is that we have used boundedness in our own derivation in various places, even though such assumption may be relaxed with more efforts and much messier proof. Another open question is whether the implied constraint on M mentioned at the end of Section 2.2 is tight. This appears to be a challenging open question that we are not currently able to answer.

Acknowlegements

The authors sincerely thank the Action Editor Professor Jie Peng, and two anonymous reviewers for their insightful comments and suggestions which greatly improved the paper. The research of Lian is partially supported by City University of Hong Kong Start Up Grant 7200521.

Appendix A. Proofs of Propositions.

Proof of Proposition 1. We first show that

$$\|\widehat{\mathbf{\Theta}}\widehat{\mathbf{H}}(\widehat{\boldsymbol{\beta}})-\mathbf{I}\|_\infty \le Cb_N \text{ and } \|\widehat{\mathbf{\Theta}}\|_{L_1} \le C_N. \tag{9}$$

First, we note these will be implied by that

$$\|\mathbf{\Theta}_0\widehat{\mathbf{H}}(\widehat{\boldsymbol{\beta}})-\mathbf{I}\|_\infty \le Cb_N. \tag{10}$$

In fact, Equation 10 means the constrained optimization problem is feasible and the feasibility of $\widehat{\mathbf{\Theta}}$ immediately implies the first equation of Equation 9. For the second equation

of Equation 9, we only need to note that by Equation 10 and the definition of the constrained optimization problem, which can be solved for each row of $\mathbf{\Theta}$ separately, we have $\|\widehat{\mathbf{\Theta}}\|_{L_1} \le \|\mathbf{\Theta}_0\|_{L_1}$.

To establish Equation 10, we bound $\|\mathbf{\Theta}_0\widehat{\mathbf{H}}(\widehat{\boldsymbol{\beta}})-\mathbf{I}\|_\infty = \|\mathbf{\Theta}_0(\widehat{\mathbf{H}}(\widehat{\boldsymbol{\beta}})-\mathbf{H}(\boldsymbol{\beta}_0))\|_\infty \le \|\mathbf{\Theta}_0\|_{L_1}\|\widehat{\mathbf{H}}(\widehat{\boldsymbol{\beta}})-\mathbf{H}(\boldsymbol{\beta}_0)\|_\infty$. Furthermore, we write

$$\begin{aligned} & \|\widehat{\mathbf{H}}(\widehat{\boldsymbol{\beta}})-\mathbf{H}(\boldsymbol{\beta}_0)\|_\infty \\ \le\ & \|E[\widehat{\mathbf{H}}(\boldsymbol{\beta}_0)]-\mathbf{H}(\boldsymbol{\beta}_0)\|_\infty + \|\widehat{\mathbf{H}}(\boldsymbol{\beta}_0)-E[\widehat{\mathbf{H}}(\boldsymbol{\beta}_0)]\|_\infty + \|\widehat{\mathbf{H}}(\widehat{\boldsymbol{\beta}})-\widehat{\mathbf{H}}(\boldsymbol{\beta}_0)\|_\infty \\ =:\ & I_1+I_2+I_3. \end{aligned}$$

First we consider I_1 and show that for any bounded function $s(\mathbf{x})$ whose partial derivatives are also bounded, we have

$$\left|E\left[\frac{1}{h}q\left(\frac{1-y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0}{h}\right)s(\mathbf{x})\right]-E\left[\delta(1-y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0)s(\mathbf{x})\right]\right| \le Ch/\beta_{01}^2,$$

and we will then obtain $I_1 \le Ch/\beta_{01}^2$ by setting $s(\mathbf{x}) = x_j x_{j'}, 1 \le j, j' \le p$.

In fact, since, for example, $E[\delta(1-y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0)s(\mathbf{x})] = P(y=1)E[\delta(1-y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0)s(\mathbf{x})|y=1]+P(y=-1)E[\delta(1-y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0)s(\mathbf{x})|y=-1]$, we only need to consider conditional expectation given $y=1$ (conditional expectation given $y=-1$ is similar). Write $\boldsymbol{\beta}_{0,-1} = (\beta_{02},\ldots,\beta_{0p})^{\mathrm{T}}$ and $\mathbf{x}_{-1} = (x_2,\ldots,x_p)^{\mathrm{T}}$. Then, by a change of variable $(x_1,\ldots,x_p) \to (z_1,x_2,\ldots,x_p)$ with $z_1 = \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0$, we have

$$\begin{aligned} & E[\frac{1}{h}q\left(\frac{1-\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0}{h}\right)s(\mathbf{x})|y=1] \\ = & \int \frac{1}{h}q\left(\frac{1-\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0}{h}\right)s(\mathbf{x})f(\mathbf{x})d\mathbf{x} \\ = & \int \frac{1}{h}q\left(\frac{1-z_1}{h}\right)s\left(\frac{z_1-\mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}}{\beta_{01}},\mathbf{x}_{-1}\right)f\left(\frac{z_1-\mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}}{\beta_{01}},\mathbf{x}_{-1}\right)\frac{1}{\beta_{01}}dz_1d\mathbf{x}_{-1} \\ \overset{u=(1-z_1)/h}{=} & \int q(u)s\left(\frac{1-uh-\mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}}{\beta_{01}},\mathbf{x}_{-1}\right)f\left(\frac{1-uh-\mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}}{\beta_{01}},\mathbf{x}_{-1}\right)\frac{1}{\beta_{01}}dud\mathbf{x}_{-1} \\ = & \int q(u)(sf)\left(\frac{1-\mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}}{\beta_{01}},\mathbf{x}_{-1}\right)\frac{1}{\beta_{01}}dud\mathbf{x}_{-1} \\ & -\int q(u)(sf)^{(1)}\left(\frac{1-*-\mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}}{\beta_{01}},\mathbf{x}_{-1}\right)\frac{1}{\beta_{01}^2}uhdud\mathbf{x}_{-1}, \end{aligned}$$

where $(sf)^{(1)}$ is the partial derivative of $(sf)(.)$ with respect to its first variable and $*$ represents a value between 0 and uh, and

$$\begin{aligned} & E[\delta(1-\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0)s(\mathbf{x})|y=1] \\ = & \int \delta(1-\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0)s(\mathbf{x})f(\mathbf{x})d\mathbf{x} \\ = & \int s\left(\frac{1-\mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}}{\beta_{01}},\mathbf{x}_{-1}\right)f\left(\frac{1-\mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}}{\beta_{01}},\mathbf{x}_{-1}\right)\frac{1}{\beta_{01}}dz_1d\mathbf{x}_{-1}. \end{aligned}$$

Using $\int q(u)du = 1$, $\int |u|q(u)du < \infty$, and that $(sf)^{(1)}$ is bounded, we get

$$\begin{aligned} & \left| E[\frac{1}{h}q\left(\frac{1-\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0}{h}\right)s(\mathbf{x})|y=1] - E[\delta(1-\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0)s(\mathbf{x})|y=1]\right| \\ = & \left| \int q(u)(sf)^{(1)}\left(\frac{1-*-\mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}}{\beta_{01}}, \mathbf{x}_{-1}\right)\frac{1}{\beta_{01}^2}uhdud\mathbf{x}_{-1}\right| \\ \leq & Ch/\beta_{01}^2, \end{aligned}$$

and thus

$$I_1 \leq Ch/\beta_{01}^2. \tag{11}$$

Next we deal with I_2. Again with a bounded function s, we have $\left|\frac{1}{h}q\left(\frac{1-\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0}{h}\right)s(\mathbf{x})\right| \leq C/h$ and

$$\begin{aligned} & E\left[\left(\frac{1}{h}q\left(\frac{1-\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0}{h}\right)s(\mathbf{x})\right)^2 |y=1]\right] \\ = & \int \frac{1}{h^2}q^2\left(\frac{1-z_1}{h}\right)s^2\left(\frac{z_1-\mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}}{\beta_{01}}, \mathbf{x}_{-1}\right)f\left(\frac{z_1-\mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}}{\beta_{01}}, \mathbf{x}_{-1}\right)\frac{1}{\beta_{01}}dz_1d\mathbf{x}_{-1} \\ = & \int \frac{1}{h}q^2(u)(s^2f)\left(\frac{1-uh-\mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}}{\beta_{01}}, \mathbf{x}_{-1}\right)\frac{1}{\beta_{01}}dud\mathbf{x}_{-1} \\ \leq & C/(h\beta_{01}), \end{aligned} \tag{12}$$

since $\int q^2(u)du < \infty$. Thus

$$E\left[\left(\frac{1}{h}q\left(\frac{1-\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0}{h}\right)s(\mathbf{x})\right)^r \Big| y=1\right] \leq (C/h)^{r-2}(1/(h\beta_{01})),\ r \geq 2.$$

By Bernstein's inequality (Lemma 2.2.11 in van der Vaart and Wellner (1996)),

$$\begin{aligned} & P\left(\left|\frac{1}{N}\sum_i I_{\{y_i=1\}}\frac{1}{h}q\left(\frac{1-y_i\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0}{h}\right)s(\mathbf{x}_i) - E\left[I_{\{y=1\}}\frac{1}{h}q\left(\frac{1-y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0}{h}\right)s(\mathbf{x})\right]\right| > t\right) \\ \leq & 2\exp\left\{-C\frac{Nht^2}{t+\beta_{01}^{-1}}\right\}, \end{aligned}$$

and the same inequality holds with $y=1$, $y_i=1$ replaced by $y=-1$, $y_i=-1$, and thus

$$\left|\frac{1}{N}\sum_i \frac{1}{h}q\left(\frac{1-y_i\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0}{h}\right)s(\mathbf{x}_i) - E\left[\frac{1}{h}q\left(\frac{1-y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0}{h}\right)s(\mathbf{x})\right]\right| \leq C\left(\sqrt{\frac{\log p}{Nh\beta_{01}}} + \frac{\log p}{Nh}\right) \tag{13}$$

with probability at least $1-p^{-C}$ (note C here can be arbitrarily large as long as we are willing to set the constant C in the display above to be large enough). By choosing $s(\mathbf{x}) = x_jx_{j'}$, using the the union bound,

$$I_2 \leq C\left(\sqrt{\frac{\log p}{Nh\beta_{01}}} + \frac{\log p}{Nh}\right), \tag{14}$$

with probability at least $1-p^{-C}$.

Finally, we bound I_3. By Taylor's expansion, we have

$$\begin{aligned} & \left|\frac{1}{N}\sum_i I_{\{y_i=1\}}x_{ij}x_{ij'}q((1-\mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}})/h)/h - \frac{1}{N}\sum_i I_{\{y_i=1\}}x_{ij}x_{ij'}q((1-\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0)/h)/h\right| \\ \leq & \left|\frac{C}{N}\sum_i I_{\{y_i=1\}}x_{ij}x_{ij'}q'((1-\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0)/h)/h^2\cdot\mathbf{x}_i^{\mathrm{T}}(\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}_0)\right| \\ & + \left|\frac{C}{2N}\sum_i I_{\{y_i=1\}}x_{ij}x_{ij'}q''(*)/h^3\cdot(\mathbf{x}_i^{\mathrm{T}}(\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}_0))^2\right| \\ =: & J_1 + J_2, \end{aligned}$$

where $*$ denotes a value between $(1-\mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}})/h$ and $(1-\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0)/h$. The second term above is easy to deal with. Since $q''(.)$ is bounded, we get

$$J_2 \leq \frac{Cs^2\log p}{Nh^3},$$

with probability $1-p^{-C}$. Although J_1 could be bounded similar to J_2, a more careful calculation will yield a tighter bound. For this we write

$$\begin{aligned} & \left|\frac{1}{N}\sum_i I_{\{y_i=1\}}x_{ij}x_{ij'}q'(1-\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0/h)/h^2\right| \\ \leq & \left|\frac{1}{N}\sum_i I_{\{y_i=1\}}x_{ij}x_{ij'}q'((1-\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0)/h)/h^2 - E[I_{\{y=1\}}x_jx_{j'}q'(1-\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0/h)/h^2]\right| \\ & + \left|E[I_{\{y=1\}}x_jx_{j'}q'(1-\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0/h)/h^2]\right|. \end{aligned}$$

Similar to Equation 12, we have $|q'((1-\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0)/h)/h^2| \leq C/h^2$ and $E[(q'((1-\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0)/h)/h^2)^2|y=1] \leq C/(h^3\beta_{01})$, and by the same arguments as those that lead to Equation 13, we get

$$\begin{aligned} & \max_{j,j'}\left|\frac{1}{N}\sum_i I_{\{y_i=1\}}x_{ij}x_{ij'}q'((1-\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0)/h)/h^2 - E[I_{\{y=1\}}x_jx_{j'}q'(1-\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0/h)/h^2]\right| \\ \leq & C\left(\sqrt{\frac{\log p}{Nh^3\beta_{01}}} + \frac{\log p}{Nh^2}\right), \end{aligned}$$

with probability at least $1-p^{-C}$. Furthermore, for any bounded $s(\mathbf{x})$ whose partial derivatives are also bounded, we have

$$\begin{aligned}
& \left|E[s(\mathbf{x})q'((1-\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0)/h)/h^2|y=1]\right| \\
= & \left|\int q'((1-z_1)/h)/h^2\cdot(sf)\left(\frac{z_1-\mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}}{\beta_{01}},\mathbf{x}_{-1}\right)\frac{1}{\beta_{01}}dz_1d\mathbf{x}_{-1}\right| \\
= & \left|\int q'(u)\cdot(sf)\left(\frac{1-uh-\mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}}{\beta_{01}},\mathbf{x}_{-1}\right)\frac{1}{h\beta_{01}}dud\mathbf{x}_{-1}\right| \\
= & \left|\int (sf)\left(\frac{1-\mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}}{\beta_{01}},\mathbf{x}_{-1}\right)(\int q'(u)du)\frac{1}{h\beta_{01}}d\mathbf{x}_{-1}\right|+\left|\int q'(u)u(sf)^{(1)}(*)\frac{1}{\beta_{01}^2}dud\mathbf{x}_{-1}\right| \\
\leq & C/\beta_{01}^2,
\end{aligned}$$

using that $\int q'(u)du=0$. Thus

$$J_1\leq C\left(\frac{1}{\beta_{01}^2}+\sqrt{\frac{\log p}{Nh^3\beta_{01}}}+\frac{\log p}{Nh^2}\right)\cdot s\sqrt{\frac{\log p}{N}}$$

with probability at least $1-p^{-C}$ and then

$$I_3\leq C\left(\left(\frac{1}{\beta_{01}^2}+\sqrt{\frac{\log p}{Nh^3\beta_{01}}}+\frac{\log p}{Nh^2}\right)\cdot s\sqrt{\frac{\log p}{N}}+\frac{s^2\log p}{Nh^3}\right), \tag{15}$$

with probability at least $1-p^{-C}$. Combining bounds in Equation 11, Equation 14 and Equation 15, we get

$$\|\widehat{\mathbf{H}}(\widehat{\boldsymbol{\beta}})-\mathbf{H}(\boldsymbol{\beta}_0)\|_\infty\leq C\left(\left(\frac{1}{\beta_{01}^2}+\sqrt{\frac{\log p}{Nh^3\beta_{01}}}+\frac{\log p}{Nh^2}\right)\cdot s\sqrt{\frac{\log p}{N}}+\frac{s^2\log p}{Nh^3}+\frac{h}{\beta_{01}^2}+\sqrt{\frac{\log p}{Nh\beta_{01}}}\right),$$

and thus

$$\|\boldsymbol{\Theta}_0\widehat{\mathbf{H}}(\widehat{\boldsymbol{\beta}})-\mathbf{I}\|_\infty\leq Cb_N,$$

with probability at least $1-p^{-C}$.

Finally, we also have

$$\begin{aligned}
& \|\widehat{\boldsymbol{\Theta}}\mathbf{H}(\boldsymbol{\beta}_0)-\mathbf{I}\|_\infty \\
\leq & \|\widehat{\boldsymbol{\Theta}}\widehat{\mathbf{H}}(\widehat{\boldsymbol{\beta}})-\mathbf{I}\|_\infty+\|\widehat{\boldsymbol{\Theta}}(\widehat{\mathbf{H}}(\widehat{\boldsymbol{\beta}})-\mathbf{H}(\boldsymbol{\beta}_0))\|_\infty \\
\leq & Cb_N+\|\widehat{\boldsymbol{\Theta}}\|_{L_1}\|\widehat{\mathbf{H}}(\widehat{\boldsymbol{\beta}})-\mathbf{H}(\boldsymbol{\beta}_0)\|_\infty \\
\leq & Cb_N,
\end{aligned}$$

using the bound for $\|\widehat{\mathbf{H}}(\widehat{\boldsymbol{\beta}})-\mathbf{H}(\boldsymbol{\beta}_0)\|_\infty$ above. ■

Proof of Proposition 2. We define $\Omega=\{\boldsymbol{\beta}\in\mathbb{R}^p:\|\boldsymbol{\beta}\|_0\leq K,\|\boldsymbol{\beta}-\boldsymbol{\beta}_0\|_1\leq Cs\sqrt{\log p/N}\}$ and we have $\widehat{\boldsymbol{\beta}}\in\Omega$ with probability at least $1-p^{-C}$. Define the class of functions

$$\mathcal{G}_j=\{yx_j(I\{y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}\leq 1\}-I\{y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0\leq 1\}):\boldsymbol{\beta}\in\Omega\},$$

with squared integrable envelope function $F(\mathbf{x},y)=|x_j|$.

We decompose Ω as $\Omega=\cup_{T\subset\{1,\ldots,p\},|T|\leq K}\Omega(T)$ with $\Omega(T)=\{\boldsymbol{\beta}:\text{ support of }\boldsymbol{\beta}\subset T\}\cap\Omega$. We also define $\mathcal{G}_j(T)=\{yx_j(I\{y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}\leq 1\}-I\{y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0\leq 1\}):\boldsymbol{\beta}\in\Omega(T)\}$.

By Lemma 2.6.15, Lemma 2.6.18 (vi) and (viii) (actually by the proof of Lemma 2.6.18 (viii)) in van der Vaart and Wellner (1996), for each fixed $T\subset\{1,\ldots,p\}$ with $|T|\leq K$, $\mathcal{G}_j(T)$ is a VC-subgraph with index bounded by $K+2$ and by Theorem 2.6.7 of van der Vaart and Wellner (1996), we have

$$N(\epsilon,\mathcal{G}_j(T),L_2(P_n))\leq\left(\frac{C\|F\|_{L_2(P_n)}}{\epsilon}\right)^{CK}\leq\left(\frac{C}{\epsilon}\right)^{CK}.$$

Since there are at most $\binom{p}{K}\leq(ep/K)^K$ different such T, we have

$$N(\epsilon,\mathcal{G}_j,L_2(P_n))\leq\left(\frac{C}{\epsilon}\right)^{CK}\left(\frac{ep}{K}\right)^K\leq\left(\frac{Cp}{\epsilon}\right)^{CK},$$

and thus

$$N(\epsilon,\cup_{j=1}^p\mathcal{G}_j,L_2(P_n))\leq p\left(\frac{Cp}{\epsilon}\right)^{CK}.$$

Let $\sigma^2=\sup_{f\in\cup_j\mathcal{G}_j}Pf^2$. Then by Theorem 3.12 of Koltchinskii (2011), we have

$$E\|R_n\|_{\cup_j\mathcal{G}_j}\leq C\left(\sigma\sqrt{\frac{K\log p}{N}}+\frac{K\log p}{N}\right),$$

where $\|R_n\|_{\cup_j\mathcal{G}_j}=\sup_{f\in\cup_j\mathcal{G}_j}N^{-1}\sum_{i=1}^N\varepsilon_if(\mathbf{x}_i,y_i)$ with ε_i being i.i.d. Rademacher random variables. Using the symmetrization inequality which states that $E\|P_n-P\|_{\cup_j\mathcal{G}_j}\leq 2E\|R_n\|_{\cup_j\mathcal{G}_j}$, where $\|P_n-P\|_{\cup_j\mathcal{G}_j}=\sup_{f\in\cup_j\mathcal{G}_j}N^{-1}\sum_i f(\mathbf{x}_i,y_i)-Ef(\mathbf{x},y)$, Talagrand's inequality (page 24 of Koltchinskii (2011)) gives

$$P\left(\|P_n-P\|_{\cup_j\mathcal{G}_j}\geq C\left(\sigma\sqrt{\frac{K\log p}{N}}+\frac{K\log p}{N}+\sqrt{\frac{\sigma^2t}{N}}+\frac{t}{N}\right)\right)\leq e^{-t},$$

that is, with probability at least $1-p^{-C}$,

$$\begin{aligned}
& \left\|\frac{1}{N}\sum_i y_i\mathbf{x}_i(I\{y_i\mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}\leq 1\}-I\{y_i\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0\leq 1\})-Ey\mathbf{x}(I\{y\mathbf{x}^{\mathrm{T}}\widehat{\boldsymbol{\beta}}\leq 1\}-I\{y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0\leq 1\})\right\|_\infty \\
\leq & C\left(\sigma\sqrt{\frac{K\log p}{N}}+\frac{K\log p}{N}\right).
\end{aligned}$$

Finally, we need to decide the size of σ^2. For $\boldsymbol{\beta}\in\Omega$, we have that

$$\begin{aligned}
& E[(I\{\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}\leq 1\}-I\{\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0\leq 1\})^2|y=1] \\
\leq & P(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}\leq 1,\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0\geq 1|y=1)+P(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}\geq 1,\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0\leq 1|y=1) \\
\leq & P(1\leq\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0\leq 1+Cs\sqrt{\log p/N}|y=1)+P(1-Cs\sqrt{\log p/N}\leq\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0\leq 1|y=1) \\
\leq & Cs\sqrt{\log p/N}/\beta_{01},
\end{aligned}$$

where we used in the second inequality the fact that $\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta} \le 1 \le \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0$ implies $\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 \le 1 + |\mathbf{x}^{\mathrm{T}}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)| \le 1 + C\|\boldsymbol{\beta} - \boldsymbol{\beta}_0\|_1$ and similarly that $\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 \ge 1 - C\|\boldsymbol{\beta} - \boldsymbol{\beta}_0\|_1$, and in the last inequality we used that the density of $\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0$ conditional on $y = 1$ is bounded by $1/\beta_{01}$. This last observation follows easily from that, by change of variable $z_1 = \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0$, the joint density of $(z_1, \mathbf{x}_{-1})$ conditional on $y = 1$ is given by

$$f((z_1 - \mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1})/\beta_{01}, \mathbf{x}_{-1})/\beta_{01}.$$

Thus we have $\sigma^2 \le Cs\sqrt{\log p/N}/\beta_{01}$ which proved the proposition. ■

Proof of Proposition 3. By integrating over x_1 first, we have for $s(\mathbf{x}) = x_j x_{j'}$,

$$\begin{aligned}
&\int \delta(1 - \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}) s(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} - \int \delta(1 - \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0) s(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} \\
= &\int \frac{1}{\beta_1}(sf)\left(\frac{1 - \mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}}{\beta_1}, \mathbf{x}_{-1}\right) d\mathbf{x}_{-1} - \int \frac{1}{\beta_{01}}(sf)\left(\frac{1 - \mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{0,-1}}{\beta_{01}}, \mathbf{x}_{-1}\right) d\mathbf{x}_{-1} \\
= &\frac{\beta_{01} - \beta_1}{\beta_1\beta_{01}} \int (sf)\left(\frac{1 - \mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}}{\beta_1}, \mathbf{x}_{-1}\right) d\mathbf{x}_{-1} + \frac{1}{\beta_{01}} \int (sf)^{(1)}(*, \mathbf{x}_{-1}) \mathbf{x}_{-1}^{\mathrm{T}} \left(\frac{\boldsymbol{\beta}_{0,-1}}{\beta_{01}} - \frac{\boldsymbol{\beta}_{-1}}{\beta_1}\right) d\mathbf{x}_{-1},
\end{aligned}$$

where $*$ represents a value between $\frac{1 - \mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{0,-1}}{\beta_{01}}$ and $\frac{1 - \mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}}{\beta_1}$. Using $|\beta_1 - \beta_{01}| \le \|\boldsymbol{\beta} - \boldsymbol{\beta}_0\|_1 \le \beta_{01}/2$, $\beta_1 \ge \beta_{01} - |\beta_1 - \beta_{01}| \ge (1/2)\beta_{01}$, and $\|\frac{\boldsymbol{\beta}_{0,-1}}{\beta_{01}} - \frac{\boldsymbol{\beta}_{-1}}{\beta_1}\|_1 \le \frac{\|\boldsymbol{\beta}_0\|_1 \cdot \|\boldsymbol{\beta}_1 - \boldsymbol{\beta}_{01}\|_1}{|\beta_{01}\beta_1|} \le \frac{C}{\beta_{01}^2}\|\boldsymbol{\beta}_0\|_1\|\boldsymbol{\beta}_1 - \boldsymbol{\beta}_{01}\|_1$, the lemma is proved. ■

Appendix B. Discussions of Assumption (A3).

First we show that $\mathbf{H}(\boldsymbol{\beta}_0)$ can also be expressed as

$$c_1 E[\mathbf{x}\mathbf{x}^{\mathrm{T}} | y = 1, \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 = 1] + c_2 E[\mathbf{x}\mathbf{x}^{\mathrm{T}} | y = -1, \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 = -1], \tag{16}$$

for two positive constants c_1, c_2. Indeed, let $h(z, x_2, \ldots, x_p)$ be the joint density of $(z = \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0, x_2, \ldots, x_p)^{\mathrm{T}}$. We have $h(z, x_2, \ldots, x_p) = f(\frac{z - \mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{0,-1}}{\beta_{01}}, x_2, \ldots, x_p)\frac{1}{\beta_{01}}$. Then, for any function $s(\mathbf{x})$, we have

$$\begin{aligned}
&E[\delta(1 - \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0) s(\mathbf{x}) | y = 1] \\
= &\int \delta(1 - \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0) s(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} \\
= &\int s(\frac{1 - \mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{0,-1}}{\beta_{01}}, x_2, \ldots, x_p) f(\frac{z - \mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{0,-1}}{\beta_{01}}, x_2, \ldots, x_p)\frac{1}{\beta_{01}} d\mathbf{x}_{-1},
\end{aligned}$$

and

$$\begin{aligned}
&E[s(\mathbf{x}) | y = 1, \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 = 1] \\
= &E[s(\frac{z - \mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{0,-1}}{\beta_{01}}, x_2, \ldots, x_p) | y = 1, z = 1] \\
= &\int s(\frac{1 - \mathbf{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{0,-1}}{\beta_{01}}, x_2, \ldots, x_p) \frac{h(1, x_2, \ldots, x_p)}{h_z(1)} d\mathbf{x}_{-1},
\end{aligned}$$

where h_z is the marginal density of $z = \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0$ conditional on $y = 1$. Thus we see that

$$E[\delta(1 - \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0) s(\mathbf{x}) | y = 1] = h_z(1) E[s(\mathbf{x}) | y = 1, \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 = 1],$$

which implies Equation 16

Let's further assume that y is independent of $\mathbf{x}$ given $\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0$. This is a natural sufficient dimension reduction type of assumption. This is the case for example if the class label is generated as in our simulations, or if the data follows the popular logistic regression model. Also assume $\mathbf{x}$ is multivariate normal. Then $E[\mathbf{x}\mathbf{x}^{\mathrm{T}} | y = 1, \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 = 1] = E[\mathbf{x}\mathbf{x}^{\mathrm{T}} | \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 = 1]$ by the conditional independence. Since $\mathbf{x}$ has a symmetric distribution, $E[\mathbf{x}\mathbf{x}^{\mathrm{T}} | \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 = 1] = E[\mathbf{x}\mathbf{x}^{\mathrm{T}} | \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 = -1]$ and thus the Hessian is equal to a constant multiple of $E[\mathbf{x}\mathbf{x}^{\mathrm{T}} | \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 = 1]$. Now we examine the inverse of $E[\mathbf{x}\mathbf{x}^{\mathrm{T}} | \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 = 1]$.

Assume $E[\mathbf{x}] = \mathbf{0}$ and $Cov(\mathbf{x}) = E[\mathbf{x}\mathbf{x}^{\mathrm{T}}] = \mathbf{S}$. The literature on high-dimensional precision matrix estimation typically assume that $\mathbf{S}^{-1}$ is sparse or approximately sparse to make the estimation feasible. We first see how the inverse of $E[\mathbf{x}\mathbf{x}^{\mathrm{T}} | \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 = 1]$ is related to $\mathbf{S}$ under normality. By the normality of $\mathbf{x}$, $(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0, x_1, \ldots, x_p)$ is again (degenerate) normal with covariance matrix

$$\begin{pmatrix} \boldsymbol{\beta}_0^{\mathrm{T}}\mathbf{S}\boldsymbol{\beta}_0 & \boldsymbol{\beta}_0^{\mathrm{T}}\mathbf{S} \\ \mathbf{S}\boldsymbol{\beta}_0 & \mathbf{S} \end{pmatrix}.$$

Then by the property of multivariate normal distribution,

$$\begin{aligned}
E[\mathbf{x} | \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 = 1] &= \mathbf{S}\boldsymbol{\beta}_0/(\boldsymbol{\beta}_0^{\mathrm{T}}\mathbf{S}\boldsymbol{\beta}_0), \\
Cov(\mathbf{x} | \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 = 1) &= \mathbf{S} - \frac{\mathbf{S}\boldsymbol{\beta}_0\boldsymbol{\beta}_0^{\mathrm{T}}\mathbf{S}}{\boldsymbol{\beta}_0^{\mathrm{T}}\mathbf{S}\boldsymbol{\beta}_0},
\end{aligned}$$

and thus

$$E[\mathbf{x}\mathbf{x}^{\mathrm{T}} | \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 = 1] = E[\mathbf{x} | \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 = 1] E^{\mathrm{T}}[\mathbf{x} | \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 = 1] + Cov(\mathbf{x} | \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 = 1) = \mathbf{S} + a\mathbf{S}\boldsymbol{\beta}_0\boldsymbol{\beta}_0^{\mathrm{T}}\mathbf{S},$$

where a is a scalar. By the Sherman-Morrison formula,

$$\begin{aligned}
&(E[\mathbf{x}\mathbf{x}^{\mathrm{T}} | \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 = 1])^{-1} \\
= &\mathbf{S}^{-1} - \frac{a}{1 + a\boldsymbol{\beta}_0^{\mathrm{T}}\mathbf{S}\boldsymbol{\beta}_0}\boldsymbol{\beta}_0\boldsymbol{\beta}_0^{\mathrm{T}}.
\end{aligned}$$

Thus the Hessian matrix is sparse if both $\mathbf{S}^{-1}$ and $\boldsymbol{\beta}_0$ are sparse.

Now we consider several popular and concrete cases.

Case 1. Consider the autoregressive correlation matrix where the (i, j) entry of $\mathbf{S}$ is $s_{ij} = \rho^{|i-j|}, |\rho| < 1$. In this case, it is known that

$$\mathbf{S}^{-1} = \frac{1}{1 - \rho^2} \begin{pmatrix} 1 & -\rho & & & & \\ -\rho & 1 + \rho^2 & -\rho & & & \\ & -\rho & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & 1 + \rho^2 & -\rho \\ & & & & -\rho & 1 \end{pmatrix}.$$

In particular we can see $\|\mathbf{S}^{-1}\|_{L_1} = 1/(1-|\rho|)$ independent of the size of the matrix.

Case 2. Assume $\mathbf{S}$ is a banded matrix with a fixed bandwidth, then by Theorem 2.2 of Demko (1977), the (i,j) entry of $\mathbf{S}^{-1}$ is bounded by $C\gamma^{|i-j|}$ for some constants $C > 0$, $0 < \gamma < 1$. Thus $\mathbf{S}^{-1}$ is approximately sparse in the sense that $\|\mathbf{S}^{-1}\|_{L_1}$ is bounded.

Case 3. Consider the exchangeable correlation matrix where all the non-diagonal entries of $\mathbf{S}$ are equal to ρ. Here we are not able to give theoretical properties of $\mathbf{S}^{-1}$ but will numerically compute the L_1 norm of the inverse of $(E[\mathbf{x}\mathbf{x}^{\mathrm{T}}|\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 = 1])^{-1}$.

For all three cases, we set $\boldsymbol{\beta}_0 = (1,1,1,1,1,0,\ldots,0)^{\mathrm{T}}$ and $\rho = 0.3$ and report the numerical value of the L_1 norm of the inverse of $(E[\mathbf{x}\mathbf{x}^{\mathrm{T}}|\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 = 1])^{-1}$ in Table 1. We see that in case 1 and 2 the L_1 norm does not change with p. For case 1, this can be theoretically shown easily. It seems to be an extremely cumbersome exercise to show this for case 2, however, and thus we do not try to establish this theoretically. For case 3, we see that numerically the norm increases very slowly with p.

Table 1: L_1 norm of the inverse of $E[\mathbf{x}\mathbf{x}^{\mathrm{T}}|\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 = 1]$.

	$p = 50$	$p = 100$	$p = 200$	$p = 500$	$p = 1000$	$p = 5000$
case 1	5.578	5.578	5.578	5.578	5.578	5.578
case 2	5.889	5.889	5.889	5.889	5.889	5.889
case 3	7.066	7.230	7.316	7.368	7.385	7.399

References

Maria-Florina Balcan, Yingyu Liang, Le Song, David Woodruff, and Bo Xie. Communication efficient distributed kernel principal component analysis. *arXiv:1503.06858*, mar 2015.

Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

A Belloni and V Chernozhukov. l1-penalized quantile regression in high-dimensional sparse models. *The Annals of Statistics*, 39(1):82–130, 2011.

Gilles Blanchard, Olivier Bousquet, and Pascal Massart. Statistical performance of support vector machines. *The Annals of Statistics*, 36:489–531, 2008.

P. S. Bradley and O. L. Mangasarjan. Feature selection via concave minimization and support vector machines. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML '98)*, number 98, pages 82–90, 1998.

Tony Cai, Weidong Liu, and Xi Luo. A constrained l1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494): 594–607, 2011.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3): 273–297, 1995.

Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, Cambridge, 2000.

S. Demko. Inverses of band matrices and local convergence of spline projection. *SIAM Journal on Numerical Analysis*, 14:616–619, 1977.

J Q Fan and R Z Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

Yingying Fan and Jinchi Lv. Asymptotic equivalence of regularization methods in thresholded parameter space. *Journal of the American Statistical Association*, 108(503):1044–1061, 2013.

Kengo Kato. Asymptotic normality of Powell's kernel estimator. *Annals of the Institute of Statistical Mathematics*, 64(2):255–273, 2012.

V Koltchinskii. *Oracle inequalities in empirical risk minimization and sparse recovery problems*. Springer, New York, 2011.

J Y Koo, Y Lee, Y Kim, and C Park. A Bahadur representation of the linear support vector machine. *Journal of Machine Learning Research*, 9:1343–1368, 2008.

Jason D Lee, Qiang Liu, Yuekai Sun, and Jonathan E Taylor. Communication-efficient sparse regression. *Journal of Machine Learning Research*, 18:1–30, 2017.

Y Lin. Some asymptotic properties of the support vector machine. *TR1029, University of Wisconsin, Madison*, 2000.

Yi Lin. A note on margin-based loss functions in classification. *Statistics & Probability Letters*, 68(1):73–82, 2004.

Ryan McDonald, Gideon Mann, and Nathan Silberman. Efficient large-scale distributed training of conditional maximum entropy models. *Proceedings of Advances in Neural Information Processing Systems*, pages 1231–1239, 2009.

Changyi Park, Kwang Rae Kim, Rangmi Myung, and Ja Yong Koo. Oracle properties of SCAD-penalized support vector machine. *Journal of Statistical Planning and Inference*, 142(8):2257–2270, 2012.

Bo Peng, Lan Wang, and Yichao Wu. An error bound for l1-norm support vector machine coefficients in ultra-high dimension. *Journal of Machine Learning Research*, 17(236):1–26, 2016.

Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT Press, Cambridge, MA, 2001.

Minghu Song, Curt M. Breneman, Jinbo Bi, N. Sukumar, Kristin P. Bennett, Steven Cramer, and Nihal Tugcu. Prediction of protein retention times in anion-exchange chromatography systems using support vector regression. *Journal of Chemical Information and Computer Sciences*, 42(6):1347–1357, 2002.

Ingo Steinwart. Consistency of support vector machines and other regularized kernel classifiers. *IEEE Transactions on Information Theory*, 51(1):128–142, 2005.

Ingo Steinwart and Clint Scovel. Fast rates for support vector machines using Gaussian kernels. *The Annals of Statistics*, 35:575–607, 2007.

R Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society Series B-Methodological*, 58(1):267–288, 1996.

R Tibshirani. The lasso method for variable selection in the Cox model. *Statistics in Medicine*, 16(4):385–395, 1997.

Sara van de Geer, Peter Buhlmann, Ya'acov Ritov, and Ruben Dezeure. On asymptotically optimal confidence regions and tests for high-dimensional models. *Annals of Statistics*, 42(3):1166–1202, 2014.

Sara A. van de Geer. High-dimensional generalized linear models and the Lasso. *Annals of Statistics*, 36(2):614–645, 2008.

A W van der Vaart and J A Wellner. *Weak convergence and empirical processes*. Springer Verlag, 1996.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer, New York, 2013.

C H Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894–942, 2010.

Tong Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, 32(1):56–134, 2004.

Xiang Zhang, Yichao Wu, Lan Wang, and Runze Li. Variable selection for support vector machines in moderately high dimensions. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 78(1):53–76, 2016.

Yuchen Zhang, John C. Duchi, and Martin J. Wainwright. Communication-efficient algorithms for statistical optimization. *Journal of Machine Learning Research*, 14:3321–3363, 2013.

Yuchen Zhang, John C. Duchi, and Martin J. Wainwright. Divide and conquer kernel ridge regression: a distributed algorithm with minimax optimal rates. *Journal of Machine Learning Research*, 16:3299–3340, 2015.

Tianqi Zhao, Guang Cheng, and Han Liu. A partially linear framework for massive heterogeneous data. *Annals of Statistics*, 44(4):1400–1437, 2016.

Zemin Zheng, Yingying Fan, and Jinchi Lv. High dimensional thresholded regression and shrinkage effect. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(3):627–649, 2014.

Ji Zhu, Saharon Rosset, Trevor Hastie, and Rob Tibshirani. 1-norm support vector machines. In *Neural Information Processing Systems*, 2003.

Martin a Zinkevich, Alex Smola, and Markus Weimer. Parallelized stochastic gradient descent. In *Proceedings of Advances in Neural Information Processing Systems*, pages 2595–2603, 2010.

H Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

Journal of Machine Learning Research 18 (2018) 1-24 Submitted 6/17; Revised 1/18; Published 4/18

Beyond the Hazard Rate: More Perturbation Algorithms for Adversarial Multi-armed Bandits

Zifan Li zifanli@umich.edu
Department of Statistics
University of Michigan
Ann Arbor, MI 48109, USA

Ambuj Tewari tewaria@umich.edu
Department of Statistics
University of Michigan
Ann Arbor, MI 48109, USA

Editor: Peter Auer

Abstract

Recent work on follow the perturbed leader (FTPL) algorithms for the adversarial multi-armed bandit problem has highlighted the role of the hazard rate of the distribution generating the perturbations. Assuming that the hazard rate is bounded, it is possible to provide regret analyses for a variety of FTPL algorithms for the multi-armed bandit problem. This paper pushes the inquiry into regret bounds for FTPL algorithms beyond the bounded hazard rate condition. There are good reasons to do so: natural distributions such as the uniform and Gaussian violate the condition. We give regret bounds for both bounded support and unbounded support distributions without assuming the hazard rate condition. We also disprove a conjecture that the Gaussian distribution cannot lead to a low-regret algorithm. In fact, it turns out that it leads to near optimal regret, up to logarithmic factors. A key ingredient in our approach is the introduction of a new notion called the generalized hazard rate.

Keywords: online learning, regret, multi-armed bandits, follow the perturbed leader, gradient based algorithms

1. Introduction

Starting from the seminal work of Hannan (1957) and later developments due to Kalai and Vempala (2005), perturbation based algorithms (called "Follow the Perturbed Leader (FTPL)") have occupied a central place in online learning. Another major family of online learning algorithms, called "Follow the Regularized Leader (FTRL)", is based on the idea of regularization. In special cases, such as the exponential weights algorithm for the experts problem, it has been folk knowledge that regularization and perturbation ideas are connected. That is, the exponential weights algorithm can be understood as either using negative entropy regularization or Gumbel distributed perturbations (for example, see the discussion in Abernethy et al. (2014)).

Recent work have begun to further uncover the connections between perturbation and regularization. For example, in online linear optimization, one can understand regularization and perturbation as simply two different ways to smooth a non-smooth potential function. The former corresponds to infimal convolution smoothing and the latter corresponds to stochastic (or integral convolution) smoothing (Abernethy et al., 2014). Having a generic framework for understanding perturbations allows one to study a wide variety of online linear optimization games and a number of interesting perturbations.

FTRL and FTPL algorithms have also been used beyond "full information" settings. "Full information" refers to the fact that the learner observes the entire move of the adversary. The multi-armed bandit problem is one of the most fundamental examples of "partial information" settings. Regret analysis of the multi-armed bandit problem goes back to the work of Robbins (1952) who formulated the stochastic version of the problem. The non-stochastic, or adversarial, version was formulated by Auer et al. (2002), who provided the EXP3 algorithm achieving $O(\sqrt{NT \log N})$ regret in T rounds with N arms. They also showed a lower bound of $\Omega(\sqrt{NT})$, which was later matched by the Poly-INF algorithm (Audibert and Bubeck, 2009; Audibert et al., 2011). The Poly-INF algorithm can be interpreted as an FTRL algorithm with negative Tsallis entropy regularization (Audibert et al., 2011; Abernethy et al., 2015). For a recent survey of both stochastic and non-stochastic bandit problems, see Bubeck and Cesa-Bianchi (2012).

For the non-stochastic multi-armed bandit problem, Kujala and Elomaa (2005) and Poland (2005) both showed that using the exponential (actually double exponential/Laplace) distribution in an FTPL algorithm coupled with standard unbiased estimation technique yields near-optimal $O(\sqrt{NT \log N})$ regret. Unbiased estimation needs access to arm probabilities that are not explicitly available when using an FTPL algorithm. Neu and Bartók (2013) introduced the geometric resampling scheme to approximate these probabilities while still guaranteeing low regret. Recently, Abernethy et al. (2015) analyzed FTPL for adversarial multi-armed bandits and provided regret bounds under the condition that the hazard rate of the perturbation distribution is bounded. This condition allowed them to consider a variety of perturbation distributions beyond the exponential, such as Gamma, Gumbel, Frechet, Pareto, and Weibull.

Unfortunately, the bounded hazard rate condition is violated by two of the most widely known distributions: namely the uniform[1] and the Gaussian distributions. Therefore, the results of Abernethy et al. (2015) say nothing about the regret incurred in an adversarial multi-armed bandit problem when we use these distributions (without forced exploration) to generate perturbations. Contrast this to the full information experts setting where using these distributions as perturbations yields optimal $\sqrt{T}$ regret and even yields the optimal $\sqrt{\log N}$ dependence on the dimension in the Gaussian case (Abernethy et al., 2014).

The Gaussian distribution has lighter tails than the exponential. The hazard rate of a Gaussian increases linearly on the real line (and is hence unbounded) whereas the exponential has a constant hazard rate. Does having too light a tail make a perturbation inherently bad? The uniform is even worse from a light tail point of view: it has bounded support! In fact, Kujala and Elomaa (2005) had trouble dealing with the uniform distribution and remarked, "we failed to analyze the expert setting when the perturbation distribution was uniform." Does having a bounded support make a perturbation even worse? Or is it that the hazard rate condition is just a sufficient condition without being anywhere close to nec-

1. The uniform distribution is also historically significant as it was used in the original FTPL algorithm of Hannan (1957).

essary for a good regret bound to exist. The analysis of Abernethy et al. (2015) suggests that perhaps a bounded hazard rate is critical. They even made the following conjecture.

Conjecture 1 *If a distribution $\mathcal{D}$ has a monotonically increasing hazard rate $h_{\mathcal{D}}(x)$ that does not converge as $x \to +\infty$ (e.g., Gaussian), then there is a sequence of gains that causes the corresponding FTPL algorithm to incur at least a linear regret.*

The main contribution of this paper is to provide answers to the questions raised above. First, we show that boundedness of the hazard rate is certainly not a requirement for achieving sublinear (in T) regret. Bounded support distributions, like the uniform, violate the boundedness condition on the hazard rate in the most extreme way. Their hazard rate blows up not just asymptotically at infinity, as in the Gaussian case, but as one approaches the right edge of the support. Yet, we can show (Corollary 5) that using the uniform distribution results in a regret bound of $O((NT)^{2/3})$. This bound is clearly not optimal. But optimality is not the point here. What is surprising, especially if one regards Conjecture 1 as plausible, is that a non-trivial sublinear bound holds at all. In fact, we show (Corollary 6) that using *any* continuous distribution with bounded support and bounded density results in a sublinear regret bound.

Second, moving beyond bounded support distributions to ones with unbounded support, we settle Conjecture 1 in the negative. In Theorem 12 we show that, instead of suffering linear regret as predicted by Conjecture 1, a perturbation algorithm using the Gaussian distribution enjoys a near optimal regret bound of $O(\sqrt{NT \log N} \log T)$. A key ingredient in our approach is a new quantity that we call the *generalized hazard rate* of a distribution. We show that bounded generalized hazard rate is enough to guarantee sublinear regret in T (Theorem 8).

Finally, we investigate the relationship between tail behavior of random perturbations and the regret they induce. We show that heavy tails, along with some fairly mild assumptions, guarantee a bounded hazard rate (Theorem 15) and hence previous results can yield regret bounds for these perturbations. However, light tails can fail to have a bounded hazard rate. Nevertheless, we show that under reasonable conditions, light tailed distributions do have a bounded *generalized* hazard rate (Theorem 16). This result allows us to show that reasonably behaved light-tailed distributions lead to near optimal regret (Corollary 17). In particular, the exponential power (or generalized normal) family of distributions yields near optimal regret (Theorem 19)

2. Follow the Perturbed Leader Algorithm for Bandits

Recall the setting of the adversarial multi-armed bandit problem (Auer et al., 2002). An adversary (or Nature) chooses gain vectors $g_t \in [-1, 0]^N$ for $1 \le t \le T$ ahead of the game. Such an adversary is called *oblivious*. At round $t = 1, \dots, T$ in a repeated game, the learner must choose a distribution $p_t \in \Delta_N$ over the set of N available arms (or actions). The learner plays action i_t sampled according to p_t and accumulates the gains $g_{t,i_t} \in [-1, 0]$. The learner observes only g_{t,i_t} and receives no information about the values $g_{t,j}$ for $j \neq i_t$.

The learner's goal is to minimize the *regret*. Regret is defined to be the difference in the realized gains and the gains of the best fixed action in hindsight:

$$\text{Regret}_T := \max_{i \in [N]} \sum_{t=1}^{T} (g_{t,i} - g_{t,i_t}).$$

To be precise, we consider the *expected* regret, where the expectation is taken with respect to the learner's randomization. Note that, under an oblivious adversary, the only random variables in the above expression are the actions i_t of the learner. For convenience, define the cumulative gain vectors $G_t, t = 1, 2, \dots, T$ by

$$G_t := \sum_{s=1}^{t} g_s.$$

2.1 The Gradient-Based Algorithmic Template

We will consider the algorithmic template described in Framework 1, which is the Gradient Based Prediction Algorithm (GBPA) (see, for example, Abernethy et al. (2015)). Let Δ^N be the $(N-1)$-dimensional probability simplex in $\mathbb{R}^N$. Denote the standard basis vector along the ith dimension by $\mathbf{e}_i$. At any round t, the action choice i_t is made by sampling from the distribution p_t which is obtained by applying the gradient of a convex function $\tilde{\Phi}$ to the estimate $\hat{G}_{t-1}$ of the cumulative gain vector so far. The choice of $\tilde{\Phi}$ is flexible but it must be a differentiable convex function such that its gradient is always in Δ^N.

Note that we do not require the range of $\nabla\tilde{\Phi}$ be contained in the *interior* of the probability simplex. If we required the gradient to lie in the interior, we would not be able to deal with bounded support distributions such as the uniform distribution. Even though some entries of the probability vector p_t might be 0, the estimation step is always well defined since $p_{t,i_t} > 0$. But allowing $p_{t,i}$ to be zero means that $\hat{g}_t$ is not exactly an unbiased estimator of g_t. Instead, it is an unbiased estimator on the support of p_t. That is, $\mathbb{E}[\hat{g}_{t,i} | i_{1:t-1}] = g_{t,i}$ for any i such that $p_{t,i} > 0$. Here, $i_{1:t-1}$ is shorthand for $i_1, \dots, i_{t-1}$. Therefore, irrespective of whether $p_{t,i} = 0$ or not, we always have

$$\mathbb{E}[p_{t,i} \hat{g}_{t,i} | i_{1:t-1}] = p_{t,i} g_{t,i}. \tag{1}$$

When $p_{t,i} = 0$, we have $\hat{g}_{t,i} = 0$ but $g_{t,i} \le 0$, which means that $\hat{g}_t$ overestimates g_t outside the support of p_t. Hence, we also have

$$\mathbb{E}[\hat{g}_t | i_{1:t-1}] \succeq g_t, \tag{2}$$

where $\succeq$ means element-wise greater than.

We now present a basic result bounding the expected regret of GBPA in the multi-armed bandit setting. It is basically just a simple modification of the arguments in Abernethy et al. (2015) to deal with the possibility that $p_{t,i} = 0$. We state and prove this result here for completeness without making any claim of novelty.

Framework 1: Gradient-Based Prediction Alg. (GBPA) Template for Multi-Armed Bandits.

GBPA($\tilde{\Phi}$): $\tilde{\Phi}$ is a differentiable convex function such that $\nabla\tilde{\Phi} \in \Delta^N$
Nature: Adversary chooses gain vectors $g_t \in [-1,0]^N$ for $t = 1, \ldots, T$
Learner initializes $\hat{G}_0 = 0$
for $t = 1$ to T **do**
  Sampling: Learner chooses i_t according to the distribution $p_t = \nabla\tilde{\Phi}(\hat{G}_{t-1})$
  Cost: Learner incurs (and observes) gain $g_{t,i_t} \in [-1,0]$
  Estimation: Learner creates estimate of gain vector $\hat{g}_t := \frac{g_{t,i_t}}{p_{t,i_t}}\mathbf{e}_{i_t}$
  Update: Cumulative gain estimate so far $\hat{G}_t = \hat{G}_{t-1} + \hat{g}_t$
end for

Lemma 1 (Decomposition of the Expected Regret) *Define the non-smooth potential* $\Phi(G) = \max_i G_i$. *The expected regret of GBPA*($\tilde{\Phi}$) *can be written as*

$$\mathbb{E}\text{Regret}_T = \Phi(G_T) - \mathbb{E}\left[\sum_{t=1}^{T}\langle p_t, g_t\rangle\right]. \tag{3}$$

Furthermore, the expected regret of GBPA($\tilde{\Phi}$) *can be bounded by the sum of an overestimation, an underestimation, and a divergence penalty:*

$$\mathbb{E}\text{Regret}_T \le \underbrace{\tilde{\Phi}(0)}_{\text{overestimation penalty}} + \mathbb{E}\left[\underbrace{\Phi(\hat{G}_T) - \tilde{\Phi}(\hat{G}_T)}_{\text{underestimation penalty}}\right] + \mathbb{E}\left[\sum_{t=1}^{T}\underbrace{\mathbb{E}[D_{\tilde{\Phi}}(\hat{G}_t, \hat{G}_{t-1})|i_{1:t-1}]}_{\text{divergence penalty}}\right],$$

where the expectations are over the sampling of i_t *and* $D_{\tilde{\Phi}}$ *is the Bregman divergence induced by* $\tilde{\Phi}$.

Proof First, note that the regret, by definition, is

$$\text{Regret}_T = \Phi(G_T) - \sum_{t=1}^{T}\langle \mathbf{e}_{i_t}, g_t\rangle.$$

Under an oblivious adversary, only the summation on the right hand side is random. Moreover $\mathbb{E}[\langle \mathbf{e}_{i_t}, g_t\rangle | i_{1:t-1}] = \langle p_t, g_t\rangle$. This proves the claim in (3).

From (1), we know that $\mathbb{E}[\langle p_t, \hat{g}_t\rangle | i_{1:t-1}] = \langle p_t, g_t\rangle$ even if some entries in p_t might be zero. Therefore, we have

$$\mathbb{E}\text{Regret}_T = \Phi(G_T) - \mathbb{E}\left[\sum_{t=1}^{T}\langle p_t, \hat{g}_t\rangle\right]. \tag{4}$$

From (2), we know that $G_T \le \mathbb{E}[\hat{G}_T]$. This implies

$$\Phi(G_T) \le \Phi(\mathbb{E}[\hat{G}_T]) \le \mathbb{E}[\Phi(\hat{G}_T)], \tag{5}$$

where the first inequality is because $G \succeq G' \Rightarrow \Phi(G) \ge \Phi(G')$, and the second inequality is due to the convexity of Φ. Plugging (5) into (4) yields

$$\mathbb{E}\text{Regret}_T \le \mathbb{E}\left[\Phi(\hat{G}_T) - \sum_{t=1}^{T}\langle p_t, \hat{g}_t\rangle\right]. \tag{6}$$

Now, recalling the definition of Bregman divergence

$$D_{\tilde{\Phi}}(\hat{G}_t, \hat{G}_{t-1}) = \tilde{\Phi}(\hat{G}_t) - \tilde{\Phi}(\hat{G}_{t-1}) - \left\langle \nabla\tilde{\Phi}(\hat{G}_{t-1}), \hat{G}_t - \hat{G}_{t-1}\right\rangle,$$

we can write,

$$\begin{aligned}
-\sum_{t=1}^{T}\langle p_t, \hat{g}_t\rangle &= -\sum_{t=1}^{T}\left\langle \nabla\tilde{\Phi}(\hat{G}_{t-1}), \hat{g}_t\right\rangle \\
&= -\sum_{t=1}^{T}\left\langle \nabla\tilde{\Phi}(\hat{G}_{t-1}), \hat{G}_t - \hat{G}_{t-1}\right\rangle \\
&= \sum_{t=1}^{T}\left(D_{\tilde{\Phi}}(\hat{G}_t, \hat{G}_{t-1}) + \tilde{\Phi}(\hat{G}_{t-1}) - \tilde{\Phi}(\hat{G}_t)\right) \\
&= \tilde{\Phi}(\hat{G}_0) - \tilde{\Phi}(\hat{G}_T) + \sum_{t=1}^{T} D_{\tilde{\Phi}}(\hat{G}_t, \hat{G}_{t-1}).
\end{aligned} \tag{7}$$

The proof ends by plugging (7) into (6) and noting that $\tilde{\Phi}(\hat{G}_0) = \tilde{\Phi}(0)$ is not random. ■

2.2 Stochastic Smoothing of Potential Function

Let $\mathcal{D}$ be a continuous distribution with finite expectation, probability density function f, and cumulative distribution function F. Consider GBPA with potential function of the form:

$$\tilde{\Phi}(G; \mathcal{D}) = \mathbb{E}_{Z_1,\ldots,Z_N \overset{\text{i.i.d.}}{\sim} \mathcal{D}}\Phi(G + Z), \tag{8}$$

which is a *stochastic smoothing* of the non-smooth function $\Phi(G) = \max_i G_i$. Note that $Z = (Z_1, \ldots, Z_N) \in \mathbb{R}^N$. We will often hide the dependence on the distribution $\mathcal{D}$ if the distribution is obvious from the context or when the dependence on $\mathcal{D}$ is not of importance in the argument. Since Φ is convex, $\tilde{\Phi}$ is also convex. For stochastic smoothing, we have the following result to control the underestimation and overestimation penalty.

Lemma 2 *For any* G, *we have*

$$\Phi(G) + \mathbb{E}[Z_1] \le \tilde{\Phi}(G) \le \Phi(G) + EMAX(N),$$

where $EMAX(N)$ *is any function such that*

$$\mathbb{E}_{Z_1,\ldots,Z_N}[\max_i Z_i] \le EMAX(N).$$

In particular, this implies that the overestimation penalty $\tilde{\Phi}(0)$ is upper bounded by $\Phi(0) + EMAX(N) = EMAX(N)$ and the underestimation penalty $\Phi(\hat{G}_T) - \tilde{\Phi}(\hat{G}_T)$ is upper bounded by $-\mathbb{E}[Z_1]$.

Proof We have,

$$\begin{aligned}\Phi(G) + \mathbb{E}[Z_1] &= \max_i G_i + \mathbb{E}[Z_i] = \max_i (G_i + \mathbb{E}[Z_i]) \\ &\le \mathbb{E}[\max_i (G_i + Z_i)] = \tilde{\Phi}(G) \\ &\le \mathbb{E}[\max_i G_i + \max_i Z_i] = \max_i G_i + \mathbb{E}[\max_i Z_i] = \Phi(G) + \mathbb{E}[\max_i Z_i].\end{aligned}$$

Noting that $\mathbb{E}[\max_i Z_i] \le EMAX(N)$ finishes the proof. ■

Observe that $\Phi(G+Z)$ as a function of G is differentiable with probability 1 (under the randomness of the Z_i's) due to the fact that Z_i's are random variables with a density. By Proposition 2.3 of Bertsekas (1973), we can swap the order of differentiation and expectation:

$$\nabla\tilde{\Phi}(G;\mathcal{D}) = \mathbb{E}_{Z_1,\ldots,Z_N \overset{\text{i.i.d}}{\sim} \mathcal{D}} e_{i^*}, \text{ where } i^* = \arg\max_{i=1,\ldots,N} \{G_i + Z_i\}.$$

Note that, for any G, the random index i^* is unique with probability 1. Hence, ties between arms can be resolved arbitrarily. It is clear from above that $\nabla\tilde{\Phi}$, being an expectation of vectors in the probability simplex, is in the probability simplex. Thus, it is a valid potential to be used in Framework 1. Now we derive an identity to write the gradient of the smoothed potential function in terms of the expectation of the cumulative distribution function,

$$\begin{aligned}\nabla_i\tilde{\Phi}(G) = \frac{\partial\tilde{\Phi}}{\partial G_i} &= \mathbb{E}_{Z_1,\ldots,Z_N} \mathbf{1}\{G_i + Z_i > G_j + Z_j, \forall j \ne i\} \\ &= \mathbb{E}_{\tilde{G}_{-i}}[\mathbb{P}_{Z_i}[Z_i > \tilde{G}_{-i} - G_i]] = \mathbb{E}_{\tilde{G}_{-i}}[1 - F(\tilde{G}_{-i} - G_i)]\end{aligned}$$

where $\tilde{G}_{-i} = \max_{j\ne i} G_j + Z_j$. If $\mathcal{D}$ has unbounded support then this partial derivative is non-zero for all i given any G. However, it can be zero if $\mathcal{D}$ has bounded support. Similarly, we have the following useful identity that writes the diagonal of the Hessian of the smoothed potential function in terms of the expectation of the probability density function.

$$\begin{aligned}\nabla^2_{ii}\tilde{\Phi}(G) = \frac{\partial}{\partial G_i}\nabla_i\tilde{\Phi}(G) &= \frac{\partial}{\partial G_i}\mathbb{E}_{\tilde{G}_{-i}}[1 - F(\tilde{G}_{-i} - G_i)] \\ &= \mathbb{E}_{\tilde{G}_{-i}}\left[\frac{\partial}{\partial G_i}(1 - F(\tilde{G}_{-i} - G_i))\right] = \mathbb{E}_{\tilde{G}_{-i}} f(\tilde{G}_{-i} - G_i).\end{aligned} \tag{9}$$

2.3 Connection to Follow the Perturbed Leader

The sampling step of Framework 1 with a stochastically smoothed Φ as the potential $\tilde{\Phi}$ (Equation 8) can be done efficiently. Instead of evaluating the expectation (Equation 2.2), we just take a random sample. Doing so gives us an equivalent of Follow the Perturbed Leader Algorithm (FTPL) (Kalai and Vempala, 2005) applied to the bandit setting. On the other hand, the estimation step is hard because generally there is no closed-form expression for $\nabla\tilde{\Phi}$.

To address this issue, Neu and Bartók (2013) proposed Geometric Resampling (GR), an iterative resampling process to estimate $\nabla\tilde{\Phi}$ (with bias). They showed that the extra regret after stopping at M iterations of GR introduces an estimation bias that is at most $\frac{NT}{eM}$ as an additive term. That is, all GBPA regret bounds that we prove will hold for the corresponding FTPL algorithm that does M iterations of GR at every time step, with an extra additive $\frac{NT}{eM}$ term. This extra term does not affect the regret rate as long as $M = \sqrt{NT}$, because the lower bound for any adversarial multi-armed bandit algorithm is of the order $\sqrt{NT}$.

2.4 The Role of the Hazard Rate and Its Limitation

In previous work, Abernethy et al. (2015) proved that for a continuous random variable Z with finite and nonnegative expectation and support on the whole real line $\mathbb{R}$, if the hazard rate of the random variable is bounded, i.e.,

$$\sup_z \frac{f(z)}{1 - F(z)} < \infty,$$

then the expected regret of GBPA can be upper bounded as

$$\mathbb{E}\text{Regret}_T = O\Big(\sqrt{NT \times EMAX(N)}\Big).$$

Common families of distributions whose regret can be controlled in this way include Gumbel, Frechet, Weibull, Pareto, and Gamma (see Abernethy et al. (2015) for details). However, there are many other families of distributions where the hazard rate condition fails. For example, if the random variable has a bounded support, then the hazard rate would certainly explode at the end of the support. This is, in some sense, an extreme case of violation because the random variable does not even have a tail. There are also some random variables that do have support on $\mathbb{R}$ but have unbounded hazard rate, e.g., Gaussian, where the hazard rate monotonically increases to infinity. How can we perform analyses of the expected regret of GBPA using those random variables as perturbations? To address these issues, we need to go beyond the hazard rate.

3. Perturbations with Bounded Support

In this section, we prove that GBPA with any continuous distribution that has bounded support and bounded density enjoys sublinear expected regret. From Lemma 1 we see that the expected regret can be upper bounded by the sum of three terms. The overestimation penalty can be bounded very easily via Lemma 2 for a distribution with bounded support. The underestimation penalty is non-positive as long as the distribution has non-negative expectation. The only term that needs to be controlled with some effort is the divergence penalty.

We first present a general lemma that allows us to write the divergence penalty for a stochastically smoothed potential $\tilde{\Phi}$ as a sum involving certain double integrals.

Lemma 3 *When using a stochastically smoothed potential as in (8), the divergence penalty can be written as*

$$\mathbb{E}\left[D_{\tilde{\Phi}}(\hat{G}_t, \hat{G}_{t-1})|i_{1:t-1}\right] = \sum_{i \in supp(p_t)} p_{t,i} \int_0^{\left|\frac{g_{t,i}}{p_{t,i}}\right|} \mathbb{E}_{\hat{G}_{-i}}\left[\int_0^s f(\hat{G}_{-i} - \hat{G}_{t-1,i} + r)dr\right] ds \quad (10)$$

where $p_t = \nabla\tilde{\Phi}(\hat{G}_{t-1})$, $\hat{G}_{-i} = \max_{j \neq i} \hat{G}_{t-1,j} + Z_j$ *and* $supp(p_t) = \{i : p_{t,i} > 0\}$.

Proof To reduce clutter, we drop the time subscripts: we use $\hat{G}$ to denote the cumulative estimate $\hat{G}_{t-1}$, $\hat{g}$ to denote the marginal estimate $\hat{g}_t = \hat{G}_t - \hat{G}_{t-1}$, p to denote p_t, and g to denote the true gain g_t. Note that by definition of Framework 1, $\hat{g}$ is a sparse vector with one non-zero and non-positive coordinate $\hat{g}_{i_t} = g_{i_t}/p_{i_t} = -|g_{i_t}/p_{i_t}|$. Moreover, conditioned on $i_{1:t-1}$, i_t takes value i with probability p_i. For any $i \in \text{supp}(p)$, let

$$h_i(r) = D_{\tilde{\Phi}}(\hat{G} - r\mathbf{e}_i, \hat{G}),$$

so that $h_i'(r) = -\nabla_i\tilde{\Phi}\left(\hat{G} - r\mathbf{e}_i\right) + \nabla_i\tilde{\Phi}\left(\hat{G}\right)$ and $h_i''(r) = \nabla_{ii}^2\tilde{\Phi}\left(\hat{G} - r\mathbf{e}_i\right)$. Now we write:

$$\begin{aligned}
\mathbb{E}[D_{\tilde{\Phi}}(\hat{G} + \hat{g}, \hat{G})|i_{1:t-1}] &= \sum_{i \in \text{supp}(p)} p_i D_{\tilde{\Phi}}(\hat{G} + g_i/p_i\mathbf{e}_i, \hat{G}) = \sum_{i \in \text{supp}(p)} p_i D_{\tilde{\Phi}}(\hat{G} - |g_i/p_i|\,\mathbf{e}_i, \hat{G}) \\
&= \sum_{i \in \text{supp}(p)} p_i h_i(|g_i/p_i|) = \sum_{i \in \text{supp}(p)} p_i \int_0^{|g_i/p_i|} \int_0^s h_i''(r)dr\,ds \\
&= \sum_{i \in \text{supp}(p)} p_i \int_0^{|g_i/p_i|} \int_0^s \nabla_{ii}^2\tilde{\Phi}\left(\hat{G} - r\mathbf{e}_i\right) dr\,ds \\
&= \sum_{i \in \text{supp}(p)} p_i \int_0^{|g_i/p_i|} \int_0^s \mathbb{E}_{\hat{G}_{-i}} f(\hat{G}_{-i} - \hat{G}_i + r)dr\,ds \\
&= \sum_{i \in \text{supp}(p_t)} p_{t,i} \int_0^{|g_i/p_i|} \mathbb{E}_{\hat{G}_{-i}}\left[\int_0^s f(\hat{G}_{-i} - \hat{G}_i + r)dr\right] ds.
\end{aligned}$$

The second equality on the first line implicitly used the assumption that $g_i \leq 0$, i.e., the "gains" are non-positive. The second equality on the second line used that $h_i(0) = 0$, and the equality on the fourth line used Equation (9). ■

Note that each summand in the divergence penalty expression above involves an integral of the density function of the distribution $\mathcal{D}$ over an interval. The main idea to control the divergence penalty for a bounded support distribution is to truncate the interval at the end of the support. For points that are close to the end of the support, we bound the integral by the product of the bound on the density and the interval length. For points that are far from the end of the support, we bound the integral through the hazard rate as was done by Abernethy et al. (2015).

For a general continuous random variable Z with bounded density, bounded support, we first shift it (which obviously does not change the distribution of the random action choice

i_t and hence the expected regret) and scale it so that the support is a subset of $[0, 1]$ with $\sup\{z : F(z) = 0\} = 0$ and $\inf\{z : F(z) = 1\} = 1$ where F denotes the CDF of Z. A benefit of this normalization is that the expectation of the random variable becomes non-negative so the underestimation penalty is guaranteed to be non-positive. After scaling, we assume that the bound on the density is L. We consider the perturbation ηZ where $\eta > 0$ is a tuning parameter. Write $F_\eta(x)$ and $f_\eta(x)$ to denote the CDF and PDF of the scaled random variable ηZ respectively. If F is strictly increasing, we know that F^{-1} exists. If not, define $F^{-1}(y) = \inf\{z : F(z) = y\}$. Elementary calculation gives the following useful facts:

$$F_\eta(z) = F(\frac{z}{\eta}), f_\eta(z) = \frac{f(\frac{z}{\eta})}{\eta}, F_\eta^{-1}(y) = \eta F^{-1}(y).$$

Theorem 4 (Divergence Penalty Control, Bounded Support) *The divergence penalty in the GBPA regret bound using the scaled perturbation ηZ, where Z is drawn from a bounded support distribution satisfying the conditions above, can be upper bounded, for any $\epsilon > 0$, by*

$$NL\Big(\frac{1}{2\eta\epsilon} + 1 - F^{-1}(1-\epsilon)\Big).$$

Proof From Lemma 3, we have, with $\hat{G}_{-i} = \max_{j \neq i} \hat{G}_{t-1,j} + \eta Z_j$,

$$\begin{aligned}
&\mathbb{E}\left[D_{\tilde{\Phi}}(\hat{G}_t, \hat{G}_{t-1})|i_{1:t-1}\right] \\
&= \sum_{i \in \text{supp}(p_t)} p_{t,i} \int_0^{\left|\frac{g_{t,i}}{p_{t,i}}\right|} \mathbb{E}_{\hat{G}_{-i}}\left[\int_0^s f_\eta(\hat{G}_{-i} - \hat{G}_{t-1,i} + r)dr\right] ds \\
&= \sum_{i \in \text{supp}(p_t)} p_{t,i} \int_0^{\left|\frac{g_{t,i}}{p_{t,i}}\right|} \mathbb{E}_{\hat{G}_{-i}}\left[\int_{\hat{G}_{-i} - \hat{G}_{t-1,i}}^{\hat{G}_{-i} - \hat{G}_{t-1,i} + s} f_\eta(z)dz\right] ds \\
&\leq \sum_{i \in \text{supp}(p_t)} p_{t,i} \int_0^{\left|\frac{g_{t,i}}{p_{t,i}}\right|} \Big(\mathbb{E}_{\hat{G}_{-i}}\Big[\underbrace{\int_{[\hat{G}_{-i} - \hat{G}_{t-1,i}, \hat{G}_{-i} - \hat{G}_{t-1,i} + s] \setminus [F_\eta^{-1}(1-\epsilon), \eta]} f_\eta(z)dz}_{(I)}\Big] \\
&\quad + \underbrace{\int_{[F_\eta^{-1}(1-\epsilon), \eta]} f_\eta(z)dz}_{(II)}\Big) ds. \qquad (11)
\end{aligned}$$

We bound the two integrals above differently. For the first integral, we add the restriction $f_\eta(z) > 0$ by intersecting the integral interval with the support of the function $f_\eta(z)$, denoted

as $I_{f_\eta(z)}$ so that $1 - F_\eta(z)$ is not 0 on the interval to be integrated. Thus, we get,

$$\begin{aligned}(I) &= \int_{([\hat{G}_{-i}-\hat{G}_{t-1,i},\hat{G}_{-i}-\hat{G}_{t-1,i}+s]\setminus[F_\eta^{-1}(1-\epsilon),\eta])\cap I_{f_\eta(z)}} f_\eta(z)dz \\ &= \int_{([\hat{G}_{-i}-\hat{G}_{t-1,i},\hat{G}_{-i}-\hat{G}_{t-1,i}+s]\setminus[F_\eta^{-1}(1-\epsilon),\eta])\cap I_{f_\eta(z)}} (1-F_\eta(z))\cdot\frac{f_\eta(z)}{1-F_\eta(z)}dz \\ &\le \int_{([\hat{G}_{-i}-\hat{G}_{t-1,i},\hat{G}_{-i}-\hat{G}_{t-1,i}+s]\setminus[F_\eta^{-1}(1-\epsilon),\eta])\cap I_{f_\eta(z)}} (1-F_\eta(z))\cdot\frac{L}{\eta\epsilon}dz \\ &\le (1-F_\eta(\hat{G}_{-i}-\hat{G}_{t-1,i}))\frac{sL}{\eta\epsilon}. \end{aligned} \tag{12}$$

The first inequality holds because $f_\eta(z) \le L/\eta$ and $(1-F_\eta(z)) \ge \epsilon$ on the set of z's over which we are integrating. The second inequality holds because on the set under consideration $1 - F_\eta(z) \le 1 - F_\eta(\hat{G}_{-i} - \hat{G}_{t-1,i})$ and the measure of the set is at most s.

For the second integral, we use the bound $f_\eta(z) \le L/\eta$ again to get,

$$(II) = \int_{[F_\eta^{-1}(1-\epsilon),\eta]} f_\eta(z)dz \le \frac{L}{\eta}\cdot(\eta - F_\eta^{-1}(1-\epsilon)). \tag{13}$$

Plugging (12) and (13) into (11), we can bound the divergence penalty by,

$$\begin{aligned}&\le \sum_{i\in\text{supp}(p_t)} p_{t,i}\int_0^{\left|\frac{g_{t,i}}{p_{t,i}}\right|}\Big(\mathbb{E}_{\hat{G}_{-i}}[1-F_\eta(\hat{G}_{-i}-\hat{G}_{t-1,i})]\frac{sL}{\eta\epsilon}+\frac{L(\eta-F_\eta^{-1}(1-\epsilon))}{\eta}\Big)ds \\ &= \sum_{i\in\text{supp}(p_t)} p_{t,i}\int_0^{\left|\frac{g_{t,i}}{p_{t,i}}\right|}\Big(p_{t,i}\frac{sL}{\eta\epsilon}+L(1-F^{-1}(1-\epsilon))\Big)ds \\ &= \sum_{i\in\text{supp}(p_t)} p_{t,i}\Big(p_{t,i}\frac{L}{\eta\epsilon}\frac{g_{t,i}^2}{2p_{t,i}^2}+L(1-F^{-1}(1-\epsilon))\frac{|g_{t,i}|}{p_{t,i}}\Big) \\ &\le \sum_{i\in\text{supp}(p_t)} \Big(\frac{L}{2\eta\epsilon}+L(1-F^{-1}(1-\epsilon))\Big) \\ &\le NL\Big(\frac{1}{2\eta\epsilon}+1-F^{-1}(1-\epsilon)\Big).\end{aligned}$$

The second to last inequality holds because $|g_{t,i}| \le 1$ and the last inequality holds because the sum over i is at most over all N arms. ■

The regret bound for the uniform distribution is now an easy corollary.

Corollary 5 (Regret Bound for Uniform) *For GBPA run with a stochastically smoothed potential using an appropriately scaled $[0,1]$ uniform perturbation where $\eta = (NT)^{2/3}$, the expected regret can be upper bounded by $3(NT)^{2/3}$.*

Proof For $[0,1]$ uniform distribution, we have $L = 1$, $F^{-1}(1-\epsilon) = 1-\epsilon$ so the divergence penalty is upper bounded by

$$NT(\frac{1}{2\eta\epsilon}+\epsilon).$$

If we let $\epsilon = \frac{1}{\sqrt{2\eta}}$, we can see that the divergence penalty is upper bounded by $NT\sqrt{\frac{2}{\eta}}$. Together with the overestimation penalty which is trivially bounded by η and a non-positive underestimation penalty, we see that the final regret bound is

$$NT\sqrt{\frac{2}{\eta}}+\eta.$$

Setting $\eta = (NT)^{2/3}$ gives the desired result. ■

For a general perturbation with bounded support and bounded density, the rate at which $1 - F^{-1}(1-\epsilon)$ goes to 0 as $\epsilon \to 0$ can vary but we can always guarantee sublinear expected regret.

Corollary 6 (Asymptotic Regret Bound for Bounded Support) *For stochastically smoothed GBPA using general continuous random variable ηZ where Z has bounded density and bounded support contained in $[0,1]$ and $\eta = (NT)^{2/3}$, the expected regret grows sublinearly, i.e.,*

$$\lim_{T\to\infty}\frac{\mathbb{E}\text{Regret}_T}{T} = 0.$$

Proof For a general distribution, let $\epsilon = \frac{1}{\sqrt{\eta}}$. Since the overestimation penalty is trivially bounded by η and the underestimation penalty is non-positive, the expected regret can be upper bounded by

$$LNT\Big(\frac{1}{2\sqrt{\eta}}+1-F^{-1}(1-\frac{1}{\sqrt{\eta}})\Big)+\eta.$$

Setting $\eta = (NT)^{2/3}$ we see that the expected regret can be upper bounded by

$$(\frac{L}{2}+1)(NT)^{2/3}+LNT(1-F^{-1}(1-\frac{1}{\sqrt{\eta}}).$$

Since

$$\lim_{T\to\infty} 1-F^{-1}(1-\frac{1}{\sqrt{\eta}}) = \lim_{\eta\to\infty} 1-F^{-1}(1-\frac{1}{\sqrt{\eta}}) = 1-F^{-1}(1) = 0,$$

we conclude that

$$\lim_{T\to\infty}\frac{\mathbb{E}\text{Regret}_T}{T} = 0.$$

■

4. Perturbations with Unbounded Support

Unlike perturbations with bounded support, perturbations with unbounded support (on the right) do have non-zero right tail probabilities, ensuring that $p_{t,i} > 0$ always. However, the tail behavior may be such that the hazard rate is unbounded. Still, under mild assumptions, perturbations with unbounded support (on the right) can also be shown to have near optimal expected regret in T, using the notion of *generalized hazard rate* that we now introduce.

4.1 Generalized Hazard Rate

We already know how to control the underestimation and overestimation penalties via Lemma 2. So our main focus will be to control the divergence penalty. Towards this end, we define the generalized hazard rate for a continuous random variable Z with support unbounded on the right, parameterized by $\alpha \in [0, 1)$, as

$$h_\alpha(z) := \frac{f(z)|z|^\alpha}{(1 - F(z))^{1-\alpha}},$$

where $f(z)$ and $F(z)$ denotes the PDF and CDF of Z respectively. Note that by setting $\alpha = 0$ we recover the standard hazard rate.

One of the main results of this paper is the following. Note that it includes the result (Lemma 4.3) of Abernethy et al. (2015) as a special case.

Theorem 7 (Divergence Penalty Control via Generalized Hazard Rate) *Let $\alpha \in [0, 1)$. Suppose we have $\forall z \in \mathbb{R}, h_\alpha(z) \le C$. Then,*

$$\mathbb{E}[D_{\tilde{\Phi}}(\hat{G}_t, \hat{G}_{t-1})|i_{1:t-1}] \le \frac{2C}{1-\alpha} \times N.$$

Proof Because of the unbounded support of Z, $\mathrm{supp}(p_t) = \{1, \dots, N\}$. Lemma 3 gives us:

$$\begin{aligned}
\mathbb{E}[D_{\tilde{\Phi}}(\hat{G}_t, \hat{G}_{t-1})|i_{1:t-1}] =& \sum_{i=1}^N p_{t,i} \int_0^{|g_{t,i}/p_{t,i}|} \mathbb{E}_{\tilde{G}_{-i}} \int_0^s f(\tilde{G}_{-i} - \hat{G}_{t-1,i} + r) dr ds \\
=& \sum_{i=1}^N p_{t,i} \int_0^{|g_{t,i}/p_{t,i}|} \mathbb{E}_{\tilde{G}_{-i}} \int_{\tilde{G}_{-i} - \hat{G}_{t-1,i}}^{\tilde{G}_{-i} - \hat{G}_{t-1,i} + s} f(z) dz ds \\
\le& C \sum_{i=1}^N p_{t,i} \int_0^{|g_{t,i}/p_{t,i}|} \mathbb{E}_{\tilde{G}_{-i}} \int_{\tilde{G}_{-i} - \hat{G}_{t-1,i}}^{\tilde{G}_{-i} - \hat{G}_{t-1,i} + s} (1 - F(z))^{1-\alpha} |z|^{-\alpha} dz\, ds \\
\le& C \sum_{i=1}^N p_{t,i} \int_0^{|g_{t,i}/p_{t,i}|} \mathbb{E}_{\tilde{G}_{-i}} (1 - F(\tilde{G}_{-i} - \hat{G}_{t-1,i}))^{1-\alpha} \\
& \times \int_{\tilde{G}_{-i} - \hat{G}_{t-1,i}}^{\tilde{G}_{-i} - \hat{G}_{t-1,i} + s} |z|^{-\alpha} dz\, ds.
\end{aligned}$$

Since the function $|z|^{-\alpha}$ is symmetric in z, monotonically decreasing as $|z| \to \infty$, we have

$$\int_{\tilde{G}_{-i} - \hat{G}_{t-1,i}}^{\tilde{G}_{-i} - \hat{G}_{t-1,i} + s} |z|^{-\alpha} dz \le \int_{-s/2}^{s/2} |z|^{-\alpha} dz = \frac{2^\alpha}{1-\alpha} s^{1-\alpha}.$$

Also, note that $z^{1-\alpha}$ is a concave function in z. Hence, by Jensen's inequality,

$$\mathbb{E}_{\tilde{G}_{-i}}[(1 - F(\tilde{G}_{-i} - \hat{G}_{t-1,i}))^{1-\alpha}] \le (\mathbb{E}_{\tilde{G}_{-i}}[1 - F(\tilde{G}_{-i} - \hat{G}_{t-1,i}])^{1-\alpha} = p_{t,i}^{1-\alpha}.$$

Therefore,

$$\begin{aligned}
\mathbb{E}[D_{\tilde{\Phi}}(\hat{G}_t, \hat{G}_{t-1})|i_{1:t-1}] &\le \frac{2^\alpha C}{1-\alpha} \sum_{i=1}^N p_{t,i} \int_0^{|g_{t,i}/p_{t,i}|} p_{t,i}^{1-\alpha} s^{1-\alpha} ds \\
&= \frac{2^\alpha C}{1-\alpha} \sum_{i=1}^N p_{t,i}^{2-\alpha} \int_0^{|g_{t,i}/p_{t,i}|} s^{1-\alpha} ds \\
&= \frac{2^\alpha C}{(1-\alpha)(2-\alpha)} \sum_{i=1}^N p_{t,i}^{2-\alpha} |g_{t,i}/p_{t,i}|^{2-\alpha} \\
&= \frac{2^\alpha C}{(1-\alpha)(2-\alpha)} \sum_{i=1}^N |g_{t,i}|^{2-\alpha} \\
&\le \frac{2^\alpha C}{(1-\alpha)(2-\alpha)} N \le \frac{2C}{1-\alpha} N.
\end{aligned}$$

■

A regret bound now easily follows.

Theorem 8 (Regret Bound via Generalized Hazard Rate) *Suppose we use a stochastically smoothed GBPA with perturbation ηZ, with Z's generalized hazard rate being bounded: $h_\alpha(x) \le C, \forall x \in \mathbb{R}$ for some $\alpha \in [0, 1)$, and*

$$\mathbb{E}_{Z_1,\dots,Z_N}[\max_i Z_i] - \mathbb{E}[Z_1] \le Q(N),$$

where $Q(N)$ is some function of N. Then, if we set $\eta = (\frac{2CNT}{(1-\alpha)Q(N)})^{1/(2-\alpha)}$, the expected regret of GBPA is no greater than

$$2 \times (\frac{2C}{1-\alpha})^{1/(2-\alpha)} \times (NT)^{1/(2-\alpha)} \times Q(N)^{(1-\alpha)/(2-\alpha)}.$$

In particular, this implies that the algorithm has sublinear expected regret.

Proof The divergence penalty can be controlled through Theorem 7 once we have bounded generalized hazard rate. It remains to control the overestimation and underestimation penalty. By Lemma 2, they are at most $\mathbb{E}_{Z_1,\dots,Z_n}[\max_i Z_i]$ and $-\mathbb{E}[Z_1]$ respectively. Suppose we scale the perturbation Z by $\eta > 0$, i.e., we add ηZ_i to each coordinate. It is easy to see that $\mathbb{E}[\max_{i=1,\dots,n} \eta Z_i] = \eta \mathbb{E}[\max_{i=1,\dots,n} Z_i]$ and $\mathbb{E}[\eta Z_1] = \eta \mathbb{E}[Z_1]$. For the divergence penalty, observe that $F_\eta(t) = F(t/\eta)$ and thus $f_\eta(t) = \frac{1}{\eta} f(t/\eta)$. Hence, the bound on the generalized hazard rate for perturbation ηZ is $\eta^{\alpha-1} C$. Plugging new bounds for the scaled perturbations into Lemma 1 gives us

$$\mathbb{E}\mathrm{Regret}_T \le \eta^{\alpha-1} \frac{2C}{1-\alpha} \times NT + \eta Q(N).$$

Setting $\eta = (\frac{2CNT}{(1-\alpha)Q(N)})^{1/(2-\alpha)}$ finishes the proof. ■

4.2 Gaussian Perturbation

In this section we prove that GBPA with the standard Gaussian perturbation incurs a near optimal expected regret in both N and T. Let $F(z)$ and $f(z)$ denote the CDF and PDF of standard Gaussian distribution.

Lemma 9 (Baricz (2008)) *For standard Gaussian random variable, we have*

$$z < \frac{f(z)}{1-F(z)} < \frac{z}{2} + \frac{\sqrt{z^2+4}}{2}.$$

This lemma together with example 2.6 in Thomas (1971) show that the hazard rate of a standard Gaussian random variable increases monotonically to infinity. However, we can still bound the generalized hazard rate for strictly positive α.

Lemma 10 (Generalized Hazard Bound for Gaussian) *For any $\alpha \in (0,1)$, we have*

$$\frac{f(z)|z|^\alpha}{(1-F(z))^{1-\alpha}} \le \frac{2}{\alpha}.$$

The proof of this lemma is deferred to the appendix.

The bounded generalized hazard rate shown in the above lemma can be used to control the divergence penalty. Combined with other knowledge of the standard Gaussian random variable we are able to give a bound on the expected regret.

Corollary 11 *The expected regret of GBPA with an appropriately scaled standard Gaussian random variable as perturbation where $\eta = \left(\frac{4NT}{\alpha(1-\alpha)\sqrt{2\log N}}\right)^{1/(2-\alpha)}$ has an expected regret at most*

$$2(C_1C_2NT)^{1/(2-\alpha)}(\sqrt{2\log N})^{(1-\alpha)/(2-\alpha)}$$

where $C_1 = \frac{2}{\alpha}$, $C_2 = \frac{2}{1-\alpha}$, for any $\alpha \in (0,1)$.

Proof It is known that for standard Gaussian random variable, we have $\mathbb{E}[Z_1] = 0$ and

$$\mathbb{E}_{Z_1,\ldots,Z_n}[\max_i Z_i] \le \sqrt{2\log N}.$$

Plug in to Theorem 8 gives the result. ■

It remains to optimally tune α in the above bound.

Theorem 12 (Regret Bound for Gaussian) *The expected regret of GBPA with an appropriately scaled standard Gaussian random variable as perturbation where $\eta = \left(\frac{4NT}{\alpha(1-\alpha)\sqrt{2\log N}}\right)^{1/(2-\alpha)}$ and $\alpha = \frac{1}{\log T}$ has an expected regret at most*

$$96\sqrt{NT} \times N^{1/\log T}\sqrt{\log N}\log T$$

for $T > 4$. If we assume that $T > N$, the expected regret can be upper bounded by

$$278\sqrt{NT} \times \sqrt{\log N}\log T.$$

The proof of this theorem is also deferred to the appendix.

4.3 Sufficient Condition for Near Optimal Regret

In Section 4.1 we showed that if the generalized hazard rate of a distribution is bounded, the expected regret of the GBPA can be controlled. In this section, we are going to prove that under reasonable assumptions on the distribution of the perturbation, the FTPL enjoys near optimal expected regret. Note that most proofs in this section are deferred to the appendix.

Assumptions (a)-(c). Before we proceed, let us formally state our assumptions on the distributions we will consider. The distribution needs to (a) be continuous and have bounded density (b) have finite expectation (c) have support unbounded in the $+\infty$ direction.

Note that if the expectation of the random perturbation is negative, we shift it so that the expectation is zero. Hence the underestimation penalty is non-positive. In addition to the assumptions we have made above, we make another assumption on the eventual monotonicity of the hazard rate.

Assumption (d) $h_0(z) = \dfrac{f(z)}{1-F(z)}$ is eventually monotone.

"Eventually monotone" means that $\exists z_0 \ge 0$ such that if $z > z_0$, $\frac{f(z)}{1-F(z)}$ is non-decreasing or non-increasing. This assumption might appear hard to check, but numerous theorems are available to establish the monotonicity of hazard rate, which is much stronger than what we are assuming here. For example, see Theorem 2.4 in Thomas (1971), Theorem 2 and Theorem 4 in Chechile (2003), Chechile (2009). In fact, most natural distributions do satisfy this assumption (Bagnoli and Bergstrom, 2005).

Before we proceed, we mention a standard classification of random variables into two classes based on their tail property.

Definition 13 (see, for example, Foss et al. (2009)) *A function $f(z) \ge 0$ is said to be heavy-tailed if and only if*

$$\limsup_{z\to\infty} f(z)e^{\lambda z} = \infty \quad \text{for all } \lambda > 0.$$

A distribution with CDF $F(z)$ and $\overline{F}(z) = 1 - F(z)$ is said to be heavy-tailed if and only if $\overline{F}(z)$ is heavy-tailed. If the distribution is not heavy-tailed, we say that it is light-tailed.

It turns out that under assumptions (a)-(d), if the distribution is also heavy-tailed, then the hazard rate itself is bounded. If the distribution is light-tailed, we need an additional assumption on the eventual monotonicity of a function similar to the generalized hazard rate to ensure the boundedness of the generalized hazard rate. But before we state and prove the main results, we introduce some functions and prove an intermediate lemma that will be useful to prove the main results.

Define $R(z) = -\log\overline{F}(z)$ so that we have $\overline{F}(z) = e^{-R(x)}$ and $R'(z) = \frac{f(z)}{\overline{F}(z)} = h_0(z)$.

Lemma 14 *Under assumptions (a)-(d), we have*

$$\overline{F}(z)e^{\lambda z} \text{ is eventually monotone } \forall \lambda > 0.$$

Proof Let $g(z) = \overline{F}(z)e^{\lambda z}$, then $g'(z) = e^{\lambda z}\overline{F}(z)(\lambda - \frac{f(z)}{\overline{F}(z)})$. Since $\frac{f(z)}{\overline{F}(z)}$ is eventually monotone by assumption (d), $g'(z)$ is eventually positive, negative or zero. The lemma immediately follows. ■

We are finally ready to present the main results in this section.

Theorem 15 (Heavy Tail Implies Bounded Hazard) *Under assumptions (a) - (d), if the distribution is also heavy-tailed, then the hazard rate is bounded, i.e.,*

$$\sup_z \frac{f(z)}{\overline{F}(z)} < \infty.$$

Unlike heavy-tailed distributions, the hazard rate of light-tailed distributions might be unbounded. However, it turns out that if we make an additional assumption on the eventual monotonicity of a function similar to the generalized hazard rate, we can still guarantee the boundedness of the generalized hazard rate.

Assumption (e) $\exists \delta \in (0, 1]$ such that $\frac{f(z)}{(1-F(z))^{1-\delta}}$ is eventually monotone.

Theorem 16 (Light Tail Implies Bounded Generalized Hazard) *Under assumptions (a) - (e), if the distribution is also light-tailed, then for any $\alpha \in (\delta, 1)$, the generalized hazard rate $h_\alpha(z)$ is bounded, i.e.,*

$$\sup_z \frac{f(z)|z|^\alpha}{(\overline{F}(z))^{1-\alpha}} < \infty.$$

Combining the above result with control of the divergence penalty gives us the following corollary.

Corollary 17 *Under assumptions (a)-(e), if the distribution is also light-tailed, the expected regret of GBPA with appropriately scaled perturbations drawn from that distribution is, for all $\alpha \in (\delta, 1)$ and $\xi > 0$,*

$$O\Big((TN)^{1/(2-\alpha)}N^\xi\Big).$$

In particular, if assumption (e) holds for all $\delta \in (0, 1)$, then the expected regret of GBPA is $O\Big((TN)^{1/2+\epsilon}\Big)$ for all $\epsilon > 0$, i.e., it is near optimal in both N and T.

Next we consider a family of light-tailed distributions that do not have a bounded hazard rate.

Definition 18 *The exponential power (or generalized normal) family of distributions, denoted as $\mathcal{D}_\beta$ where $\beta > 1$, is defined via the cdf*

$$f_\beta(z) = C_\beta e^{-z^\beta}, \quad z \geq 0.$$

The next theorem shows that GBPA with perturbations from this family of distributions enjoys near optimal expected regret in both N and T.

Theorem 19 (Regret Bound for Exponential Power Family) *$\forall \beta > 1$, the expected regret of GBPA with appropriately scaled perturbations drawn from $\mathcal{D}_\beta$ is, for all $\epsilon > 0$, $O\Big((TN)^{1/2+\epsilon}\Big)$.*

5. Conclusion and Future Work

Previous work on providing regret guarantees for FTPL algorithms in the adversarial multi-armed bandit setting required a bounded hazard rate condition. We have shown how to go beyond the hazard rate condition but a number of questions remain open. For example, what if we use FTPL with perturbations from discrete distributions such as Bernoulli distribution? In the full information setting Devroye et al. (2013) and Van Erven et al. (2014) have considered random walk perturbation and dropout perturbation, both leading to minimax optimal regret. But to the best of our knowledge those distributions have not been analyzed in the adversarial multi-armed bandit problem.

An unsatisfactory aspect of even the tightest bounds for FTPL algorithms from existing work, including ours, is that they never reach the minimax optimal $O(\sqrt{NT})$ bound. They come very close to it: up to logarithmic factors. It is known that FTRL algorithms, using the negative Tsallis entropy as the regularizer, can achieve the optimal bound (Audibert and Bubeck, 2009; Audibert et al., 2011; Abernethy et al., 2015). Is there a perturbation that can achieve the optimal bound?

We only considered multi-armed bandits in this work. There has been some interest in using FTPL algorithms for combinatorial bandit problems (see, for example, Neu and Bartók (2013)). In future work, it will be interesting to extend our analysis to combinatorial bandit problems.

Acknowledgments

We thank Jacob Abernethy and Chansoo Lee for helpful discussions. We acknowledge the support of NSF CAREER grant IIS-1452099 and a Sloan Research Fellowship.

Appendix A. Proofs

A.1 Proof of Lemma 10

Proof Since the numerator of the left hand side is an even function of z, and the denominator is a decreasing function, and the inequality is trivially true when $z = 0$, it suffices to prove for $z > 0$, which we assume for the rest of the proof. From Lemma 9 we can derive that

$$\frac{f(z)}{1-F(z)} < z + 1.$$

Therefore,

$$\begin{aligned}\frac{f(z)|z|^\alpha}{(1-F(z))^{1-\alpha}} &\leq \frac{f(z)z^\alpha}{(\frac{f(z)}{z+1})^{1-\alpha}} = (f(z)z)^\alpha(z+1)^{1-\alpha} \\ &\leq f(z)^\alpha(z+1) \leq zf(z)^\alpha + 1 = \sqrt{\frac{1}{2\pi}}ze^{-\alpha z^2/2} + 1.\end{aligned}$$

Let $g(z) = ze^{-\alpha z^2/2}$, $g'(z) = (1-\alpha z^2)e^{-\alpha z^2/2}$. Therefore $g(z)$ is maximized at $z^* = \sqrt{\frac{1}{\alpha}}$. Therefore,

$$\frac{f(z)|z|^\alpha}{(1-F(z))^{1-\alpha}} \le \sqrt{\frac{1}{2\pi}} ze^{-\alpha z^2/2} + 1 \le \sqrt{\frac{1}{2\pi}} z^* + 1 \le z^* + 1 = \sqrt{\frac{1}{\alpha}} + 1 \le \frac{2}{\alpha}.$$

■

A.2 Proof of Theorem 12

Proof From Corollary 11 we see that the expected regret can be upper bounded by

$$2(C_1C_2NT)^{1/(2-\alpha)}(\sqrt{2\log N})^{(1-\alpha)/(2-\alpha)}$$

where $C_1 = \frac{2}{\alpha}$ and $C_1 = \frac{2}{1-\alpha}$. Note that

$$\begin{aligned}
&2(C_1C_2NT)^{1/(2-\alpha)}(\sqrt{2\log N})^{(1-\alpha)/(2-\alpha)} \\
\le& 4(C_1C_2)^{1/(2-\alpha)} N^{1/(2-\alpha)} \sqrt{\log N}^{(1-\alpha)/(2-\alpha)} T^{1/(2-\alpha)} \\
=& 4N^{1/(2-\alpha)} \sqrt{\log N}^{(1-\alpha)/(2-\alpha)} T^{1/2} \times (C_1C_2)^{1/(2-\alpha)} T^{\alpha/(4-2\alpha)} \\
\le& 4N^{1/2} N^{\alpha/(4-2\alpha)} \sqrt{\log N} T^{1/2} \times (\frac{4}{\alpha(1-\alpha)})^{1/(2-\alpha)} T^{\alpha/(4-2\alpha)} \\
\le& 4N^{1/2} N^{\alpha} \sqrt{\log N} T^{1/2} \times \frac{4T^\alpha}{\alpha(1-\alpha)} \\
\le& 16\sqrt{NT} N^\alpha \sqrt{\log N} \times \frac{T^\alpha}{\alpha(1-\alpha)}.
\end{aligned}$$

■

If we let $\alpha = \frac{1}{\log T}$, then $T^\alpha = T^{1/\log T} = e < 3$. Then, we have, for $T > 4$,

$$\frac{T^\alpha}{\alpha(1-\alpha)} \le \frac{3\log T}{1-\frac{1}{\log T}} = \frac{3\log^2 T}{\log T - 1} \le 6\log T.$$

Putting things together finishes the proof.

A.3 Proof of Theorem 15

Proof If the distribution is heavy-tailed, we have

$$\lim_{z\to\infty} \sup \overline{F}(z) e^{\lambda z} = \infty \quad \text{for all } \lambda > 0.$$

By Lemma 14, we can erase the supremum operator and just write

$$\lim_{z\to\infty} \overline{F}(z) e^{\lambda z} = \infty \quad \text{for all } \lambda > 0.$$

Hence,

$$\lim_{z\to\infty} \overline{F}(z) e^{\lambda z} = \lim_{x\to\infty} e^{-R(z)+\lambda z} = \infty \text{ for all } \lambda > 0 \Rightarrow \limsup_{z\to\infty} \frac{R(z)}{z} = 0.$$

Note that $R'(z) = \frac{f(z)}{\overline{F}(z)}$, which is eventually monotone by assumption. Therefore, we can conclude that

$$\limsup_{z\to\infty} R'(z) < \infty \Rightarrow \sup_z \frac{f(z)}{\overline{F}(z)} < \infty.$$

■

A.4 Proof of Theorem 16

Proof If the distribution is light-tailed, we have

$$\lim_{z\to\infty} \overline{F}(z) e^{\lambda^* z} < \infty \quad \text{for some } \lambda^* > 0.$$

This immediately implies that

$$\lim_{z\to\infty} \overline{F}(z)^a z^b = 0 \quad \forall a, b > 0. \tag{14}$$

Consider $\lim_{z\to\infty} \frac{f(z)}{\overline{F}(z)} = \lim_{z\to\infty} R'(z)$. If $\lim_{z\to\infty} R'(z) < \infty$ we can immediately conclude that $\sup_z \frac{f(z)}{1-F(z)} < \infty$. If $\lim_{z\to\infty} R'(z) = \infty$ instead, note that

$$\lim_{z\to\infty} \int_{-z}^{z} R'(t) e^{-\delta R(t)} dt = -\frac{1}{\delta} e^{-\delta R(z)} \Big|_{z=-\infty}^{z=+\infty} = \frac{1}{\delta} < \infty.$$

Moreover, since $\lim_{z\to\infty} R'(z) = \infty$, $R'(z)e^{-\delta R(z)}$ is strictly positive for all $z > z_0$ for some z_0. Furthermore, $R'(z)e^{-\delta R(z)} = \frac{f(z)}{(\overline{F}(z))^{1-\delta}}$ is eventually monotone by assumption (e),

Therefore, we can conclude that

$$\lim_{z\to\infty} R'(z) e^{-\delta R(z)} = \frac{f(z)}{(\overline{F}(z))^{1-\delta}} = 0.$$

$\forall \alpha \in (\delta, 1)$, from Equation (14) we have $\lim_{z\to+\infty} z^\alpha \overline{F}(z)^{\alpha-\delta} = 0$, so

$$\lim_{z\to+\infty} \frac{f(z) z^\alpha}{(\overline{F}(z))^{1-\alpha}} = \lim_{z\to+\infty} \frac{f(z)}{\overline{F}(z)^{1-\delta}} \times z^\alpha \overline{F}(z)^{\alpha-\delta} = 0.$$

and hence

$$\sup_z \frac{f(z) z^\alpha}{(1-F(z))^{1-\alpha}} < \infty \quad \forall \alpha \in (\delta, 1).$$

■

A.5 Proof of Corollary 17

Proof For a light-tailed distribution $\mathcal{D}$, we have

$$\lim_{z\to\infty} \overline{F}_{\mathcal{D}}(z)e^{\lambda^* z} < \infty \quad \text{for some } \lambda^* > 0.$$

This implies that

$$\overline{F}_{\mathcal{D}}(z) \le Ce^{-\lambda^* z} \text{ for some } C > 0, z > z_0.$$

Let random variable Z follows distribution $\mathcal{D}$. Since Z might take negative values, we define a new distribution $\mathcal{D}'$ that only takes non-negative value by

$$f_{D'}(z) = \begin{cases} \frac{1}{p_{\mathcal{D}+}} f_D(z) & \text{if } z \ge 0 \\ 0 & \text{otherwise} \end{cases}$$

where $p_{\mathcal{D}+} = \mathbb{P}(Z \ge 0) > 0$ by right unbounded support assumption. Clearly, with this definition of $\mathcal{D}'$ we see that $\mathbb{E}_{Z_1,\dots,Z_N\sim\mathcal{D}}[\max_i Z_i] \le \mathbb{E}_{Z_1,\dots,Z_N\sim\mathcal{D}'}[\max_i Z_i]$ and for $z > z_0$, we have $\overline{F}_{\mathcal{D}'}(z) = \frac{\overline{F}_{\mathcal{D}}(z)}{p_{\mathcal{D}+}} \le C'e^{-\lambda^* z}$ where $C' = \frac{C}{p_{\mathcal{D}+}}$. Note that

$$\begin{aligned}
\mathbb{E}_{Z_1,\dots,Z_N\sim\mathcal{D}}[\max_i Z_i] &\le \mathbb{E}_{Z_1,\dots,Z_N\sim\mathcal{D}'}[\max_i Z_i] \\
&= \int_0^\infty \mathbb{P}(\max_i Z_i > x)dx \\
&\le u + \int_u^\infty \mathbb{P}(\max_i Z_i > z)dz \\
&\le u + N\int_u^\infty \mathbb{P}(Z_i > z)dz \\
&\le u + N\int_u^\infty C'e^{-\lambda^* z}dz \quad \text{assuming } u > z_0 \\
&= u + \frac{C'N}{\lambda^*}e^{-\lambda^* u}.
\end{aligned}$$

If we let $u = \frac{\log(N)}{\lambda^*}$, obviously $u > z_0$ if N is sufficiently large. Thus, we see that

$$\mathbb{E}_{Z_1,\dots,Z_N\sim\mathcal{D}}[\max_i Z_i] \le \frac{\log(N)}{\lambda^*} + C' = O(N^\xi) \quad \forall \xi > 0. \tag{15}$$

From Theorem 16 we see that $\forall \alpha \in (\delta, 1)$,

$$\frac{f(z)z^\alpha}{(1-F(z))^{1-\alpha}} \le C_\alpha \quad \forall z \in \mathbb{R}. \tag{16}$$

Plug 15 and 16 into Theorem 8 gives the desired result. ■

A.6 Proof of Corollary 19

Proof By Corollary 17 we only need to check that assumptions (a)-(d) hold for distribution $\mathcal{D}_\beta$, exponential power family is light-tailed, and assumption (e) also holds for any $\delta \in (0, 1)$. By observing the density function f_β we can trivially see that assumptions (a)-(c) hold and that the exponential power family is light-tailed. Therefore, define

$$g_{\delta,\beta}(z) = \frac{f_\beta(z)}{(\overline{F}_\beta(z))^{1-\delta}} = \frac{f_\beta(z)}{(1-F_\beta(z))^{1-\delta}},$$

it suffices to show that $\forall \delta \in [0, 1), g_{\delta,\beta}(z)$ is eventually monotone. Note that

$$\begin{aligned}
g'_{\delta,\beta}(z) &= \frac{f'_\beta(z)(1-F_\beta(z))^{1-\delta} + (1-\delta)(1-F_\beta(z))^{-\delta}f_\beta^2(z)}{(1-F_\beta(z))^{2-2\delta}} \\
&= \frac{C_\beta^2 e^{-z^\beta}}{(1-F_\beta(z))^{2-\delta}} \times \Big((1-\delta)e^{-z^\beta} - \beta z^{\beta-1}\int_z^\infty e^{-t^\beta}dt\Big).
\end{aligned}$$

It further suffices to show that

$$m_{\delta,\beta}(z) = (1-\delta)e^{-z^\beta} - \beta z^{\beta-1}\int_z^\infty e^{-t^\beta}dt$$

is eventually non-negative or non-positive $\forall \beta > 1, \delta \in [0, 1)$. Note that since $\beta > 1$,

$$\beta z^{\beta-1}\int_z^\infty e^{-t^\beta}dt = \int_z^\infty \beta z^{\beta-1}e^{-t^\beta}dt < \int_z^\infty \beta t^{\beta-1}e^{-t^\beta}dt = e^{-z^\beta}. \tag{17}$$

Therefore, $m_{0,\beta}(z) > 0$ for all $z \ge 0$, i.e., the hazard rate is always increasing and assumption (d) is satisfied. Now, we are left to show that $m_{\delta,\beta}(z)$ is eventually non-negative or non-positive for any $\delta \in (0, 1)$. Note that

$$\begin{aligned}
\beta z^{\beta-1}\int_z^\infty e^{-t^\beta}dt &= \beta(\frac{z}{z+1})^{\beta-1}(z+1)^{\beta-1}\int_z^\infty e^{-t^\beta}dt \\
&\ge \beta(\frac{z}{z+1})^{\beta-1}(z+1)^{\beta-1}\int_z^{z+1} e^{-t^\beta}dt \\
&\ge (\frac{z}{z+1})^{\beta-1}\int_z^{z+1}\beta t^{\beta-1}e^{-t^\beta}dt \\
&= (\frac{z}{z+1})^{\beta-1}\Big(e^{-z^\beta} - e^{-(z+1)^\beta}\Big).
\end{aligned}$$

Therefore,

$$\begin{aligned}
\liminf_{z\to\infty} \frac{\beta z^{\beta-1}\int_z^\infty e^{-t^\beta}dt}{e^{-z^\beta}} &\ge \liminf_{z\to\infty} \frac{(\frac{z}{z+1})^{\beta-1}\Big(e^{-z^\beta} - e^{-(z+1)^\beta}\Big)}{e^{-z^\beta}} \\
&= \lim_{z\to\infty}(\frac{z}{z+1})^{\beta-1} - \lim_{z\to\infty}(\frac{z}{z+1})^{\beta-1}e^{z^\beta-(z+1)^\beta} \\
&= 1.
\end{aligned}$$

From Equation (17) we know that

$$\limsup_{z\to\infty} \frac{\beta z^{\beta-1}\int_z^\infty e^{-t^\beta}dt}{e^{-z^\beta}} \leq 1.$$

Hence, we conclude that

$$\lim_{z\to\infty} \frac{\beta z^{\beta-1}\int_z^\infty e^{-t^\beta}dt}{e^{-z^\beta}} = 1,$$

which implies that $m_{\delta,\beta}(z)$ is eventually non-positive for any $\delta \in (0,1)$, i.e., assumption (e) holds for any $\delta \in (0,1)$.

■

References

Jacob Abernethy, Chansoo Lee, Abhinav Sinha, and Ambuj Tewari. Online linear optimization via smoothing. In *Conference on Learning Theory*, 2014.

Jacob Abernethy, Chansoo Lee, and Ambuj Tewari. Fighting bandits with a new kind of smoothness. In *Advances in Neural Information Processing Systems*, 2015.

Jean-Yves Audibert and Sébastien Bubeck. Minimax policies for adversarial and stochastic bandits. In *Conference on Learning Theory*, 2009.

Jean-Yves Audibert, Sébastien Bubeck, and Gábor Lugosi. Minimax policies for combinatorial prediction games. In *Conference on Learning Theory*, 2011.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The non-stochastic multi-armed bandit problem. *SIAM J. Comput.*, 32:48–77, 2002.

Mark Bagnoli and Ted Bergstrom. Log-concave probability and its applications. *Economic Theory*, 26(2):445–469, 2005.

Árpád Baricz. Mills' ratio: Monotonicity patterns and functional inequalities. *J. Math. Anal. Appl.*, 340(2):1362–1370, 2008.

Dimitri P. Bertsekas. Stochastic optimization problems with nondifferentiable cost functionals. *Journal of Optimization Theory and Applications*, 12(2):218–231, 1973. ISSN 0022-3239.

Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1): 1–122, 2012.

Richard A. Chechile. Mathematical tools for hazard function analysis. *J. Math. Psychol.*, 47:478–494, 2003.

Richard A. Chechile. Corrigendum to: mathematical tools for hazard function analysis [J. Math. Psychol. 47 (2003) 478-494]. *J. Math. Psychol.*, 53:298–299, 2009.

Luc Devroye, Gábor Lugosi, and Gergely Neu. Prediction by random-walk perturbation. In *Conference on Learning Theory*, 2013.

Sergey Foss, Dmitry Korshunov, and Stan Zachary. *An Introduction to Heavy-tailed and Subexponential Distributions*. Springer, 2009.

J. Hannan. Approximation to Bayes risk in repeated play. *In M. Dresher, A. W. Tucker, and P. Wolfe, editors, Contributions to the Theory of Games, volume III*, pages 97–139, 1957.

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

Jussi Kujala and Tapio Elomaa. On following the perturbed leader in the bandit setting. In *Algorithmic Learning Theory*, pages 371–385. Springer, 2005.

Gergely Neu and Gábor Bartók. An efficient algorithm for learning with semi-bandit feedback. In *Algorithmic Learning Theory*, pages 234–248. Springer, 2013.

Jan Poland. FPL analysis for adaptive bandits. In Oleg B. Lupanov, Oktay M. Kasim-Zade, Alexander V. Chaskin, and Kathleen Steinhöfel, editors, *Stochastic Algorithms: Foundations and Applications: Third International Symposium, SAGA 2005, Moscow, Russia, October 20-22, 2005. Proceedings*, pages 58–69. Springer Berlin Heidelberg, 2005.

Herbert Robbins. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.*, 58(5):527–535, 1952.

Ewart A. C. Thomas. Sufficient conditions for monotone hazard rate an application to latency-probability curves. *J. Math. Psychol.*, 8:303–332, 1971.

Tim Van Erven, Wojciech Kotlowski, and Manfred K Warmuth. Follow the leader with dropout perturbations. In *Conference on Learning Theory*, 2014.

Journal of Machine Learning Research 18 (2018) 1-24 Submitted 3/17; Revised 9/17; Published 4/18

On Faster Convergence of Cyclic Block Coordinate Descent-type Methods for Strongly Convex Minimization*

Xingguo Li LIXX1661@UMN.EDU
Department of Electrical and Computer Engineering
University of Minnesota Twin Cities
Minneapolis, MN 55455, USA
Tuo Zhao TOURZHAO@GATECH.EDU
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30318, USA
Raman Arora ARORA@CS.JHU.EDU
Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218, USA
Han Liu HANLIU@NORTHWESTERN.EDU
Department of Electrical Engineering and Computer Science
Northwestern University
Evanston, IL 60208, USA
Mingyi Hong MHONG@UMN.EDU
Department of Electrical and Computer Engineering
University of Minnesota Twin Cities
Minneapolis, MN 55455, USA

Editor: Qiang Liu

Abstract

The cyclic block coordinate descent-type (CBCD-type) methods, which perform iterative updates for a few coordinates (a block) simultaneously throughout the procedure, have shown remarkable computational performance for solving strongly convex minimization problems. Typical applications include many popular statistical machine learning methods such as elastic-net regression, ridge penalized logistic regression, and sparse additive regression. Existing optimization literature has shown that for strongly convex minimization, the CBCD-type methods attain iteration complexity of $\mathcal{O}(p\log(1/\epsilon))$, where ϵ is a pre-specified accuracy of the objective value, and p is the number of blocks. However, such iteration complexity explicitly depends on p, and therefore is at least p times worse than the complexity $\mathcal{O}(\log(1/\epsilon))$ of gradient descent (GD) methods. To bridge this theoretical gap, we propose an improved convergence analysis for the CBCD-type methods. In particular, we first show that for a family of quadratic minimization problems, the iteration complexity $\mathcal{O}(\log^2(p)\cdot\log(1/\epsilon))$ of the CBCD-type methods matches that of the GD methods in term of dependency on p, up to a $\log^2 p$ factor. Thus our complexity bounds are sharper than the existing bounds by at least a factor of $p/\log^2(p)$. We also provide a lower bound to confirm that our improved complexity bounds are tight (up to a $\log^2(p)$ factor), under the assumption that the largest and smallest eigenvalues of the Hessian matrix do not scale with p. Finally, we generalize our analysis to other strongly convex minimization problems beyond quadratic ones.

Keywords: cyclic block coordinate descent, gradient descent, strongly convex minimization, quadratic minimization, improved iteration complexity

*. Some preliminary results in this paper were presented at the 19th International Conference on Artificial Intelligence and Statistics (Li et al., 2016).

1. Introduction

We consider a class of composite convex minimization problems:

$$x^* = \operatorname*{argmin}_{x\in\mathbb{R}^d} \mathcal{F}(x),\ \mathcal{F}(x) = \mathcal{L}(x) + \mathcal{R}(x), \tag{1}$$

where $\mathcal{L}(\cdot)$ is a twice differentiable loss function and $\mathcal{R}(\cdot)$ is a possibly nonsmooth and strongly convex penalty function. Many popular statistical machine learning problems are of the form (1), such as elastic-net regression (Zou and Hastie, 2005), ridge penalized logistic regression (Hastie et al., 2009), support vector machine (Vapnik and Vapnik, 1998) and many others (Hastie et al., 2009). For notational simplicity, we assume that there exists a partition of d coordinates such that

$$x = [x_1^\top, \ldots, x_p^\top]^\top \in \mathbb{R}^d,$$

where $x_j \in \mathbb{R}^{d_j}$, $d = \sum_{j=1}^p d_j$, and $d_j \ll p$. The penalty function $\mathcal{R}(x)$ in these applications is block coordinate decomposable, i.e., $\mathcal{R}(x) = \sum_{j=1}^p \mathcal{R}_j(x_j)$. Then we can rewrite the objective in (1) as

$$\mathcal{F}(x) = \mathcal{L}(x_1, \ldots, x_p) + \sum_{j=1}^p \mathcal{R}_j(x_j).$$

Many algorithms such as gradient decent-type (GD-type) methods (Nesterov, 2004, 2007; Beck and Teboulle, 2009b,a; Becker et al., 2011), cyclic block coordinate descent-type (CBCD-type) methods (Luo and Tseng, 1992; Tseng, 1993, 2001; Friedman et al., 2007; Liu et al., 2009; Tseng and Yun, 2009; Saha and Tewari, 2013; Nutini et al., 2015; Zhao and Liu, 2015; Zhao et al., 2014b,a, 2012; Li et al., 2015b), and alternating direction method of multipliers (ADMM) (Gabay and Mercier, 1976; Boyd et al., 2011; He and Yuan, 2015; Hong and Luo, 2012; Zhao and Liu, 2012; Liu et al., 2014, 2015; Li et al., 2015a)) have been proposed to solve (1). Among these algorithms, the CBCD-type methods have been immensely successful (Friedman et al., 2007, 2010; Mazumder et al., 2011; Tibshirani et al., 2012; Razaviyayn et al., 2013; Zhao et al., 2014a). One popular instance of the CBCD-type methods is the cyclic block coordinate minimization (CBCM) method, which minimizes (1) with respect to a single block of variables while holding the rest fixed. Particularly, at the $(t+1)$-th iteration, given $x^{(t)}$, we choose to solve a collection of optimization problems: For $j = 1, \ldots, p$,

$$x_j^{(t+1)} = \operatorname*{argmin}_{x_j}\ \mathcal{L}\left(x_{1:(j-1)}^{(t+1)}, x_j, x_{(j+1):p}^{(t)}\right) + \mathcal{R}_j(x_j), \tag{2}$$

where $x_{1:(j-1)}^{(t+1)}$ and $x_{(j+1):p}^{(t)}$ are defined as

$$x_{1:(j-1)}^{(t+1)} = [x_1^{(t+1)\top}, \ldots, x_{j-1}^{(t+1)\top}]^\top \text{ and } x_{(j+1):p}^{(t)} = [x_{j+1}^{(t)\top}, \ldots, x_p^{(t)\top}]^\top.$$

For some applications (e.g. elastic-net penalized linear regression), we can obtain a simple closed form solution to (2), but for many other applications (e.g. ridge-penalized logistic regression), (2) does not admit a closed form solution and requires more sophisticated optimization procedures.

A popular alternative to CBCD-type methods is to solve a quadratic approximation of (2) using the cyclic block coordinate gradient descent (CBCGD) method. For notational simplicity, we denote the partial gradient $\nabla_{x_j}\mathcal{L}(x)$ by $\nabla_j\mathcal{L}(x)$. Then the CBCGD method solves a collection of optimization problems: For $j = 1, \ldots, p$,

$$x_j^{(t+1)} = \underset{x_j}{\operatorname{argmin}} \ (x_j - x_j^{(t)})^\top \nabla_j \mathcal{L}\left(x_{1:(j-1)}^{(t+1)}, x_{j:p}^{(t)}\right) + \frac{\eta_j}{2}\|x_j - x_j^{(t)}\|^2 + \mathcal{R}_j(x_j), \tag{3}$$

where $\eta_j > 0$ is a step-size parameter for the j-th block.

There have been many results on iteration complexity of block coordinate descent-type (BCD-type) methods, but most of them focus on the randomized BCD-type methods, where blocks are randomly chosen with replacement in each iteration (Shalev-Shwartz and Tewari, 2011; Richtárik and Takáč, 2012; Lu and Xiao, 2015), which demonstrate better iteration complexities than cyclic BCD-type methods in the worst case scenarios (Lee and Wright, 2016; Sun and Ye, 2016). In contrast, existing literature on cyclic BCD-type methods is rather limited. Beck and Tetruashvili (2013) focus on minimizing smooth objective functions, and has shown that given a pre-specified accuracy ϵ for the objective value, the CBCGD method attains linear iteration complexity of $\mathcal{O}(\log(1/\epsilon))$ for minimizing smooth and strongly convex problems, and sublinear iteration complexity of $\mathcal{O}(1/\epsilon)$ for smooth and nonstrongly convex problems. Hong et al. (2017); Yun (2014); Sun and Hong (2015) focus on minimizing nonsmooth composite objective functions such as (1), and has shown that the CBCM and CBCGD methods attain sublinear iteration complexity of $\mathcal{O}(1/\epsilon)$, when the objective function is nonstronlgy convex.

Here, we are interested in establishing an improved iteration complexity of the CBCM and CBCGD methods, when the nonsmooth composite objective function is strongly convex. Beck and Tetruashvili (2013) has shown that for smooth minimization, the CBCGD method attains linear iteration complexity of

$$\mathcal{O}\left(\frac{L_{\max} \cdot pL^2 \log(1/\epsilon)}{L_{\min}^2 \cdot \mu}\right), \tag{4}$$

where L is the Lipschitz constant of the gradient of the objective function, μ is the strongly convex constant of the objective function, $L_{\max} = \max_i L_i$, $L_{\min} = \min_i L_i$, and L_i is the Lipschitz constant of i-th block of the gradient of the objective function. However, such an iteration complexity depends on p (the number of blocks), and therefore is at least p times worse than the complexity $\mathcal{O}\left(\mu^{-1}L\log(1/\epsilon)\right)$ of the gradient descent (GD) methods.

To bridge this theoretical gap, we propose an improved convergence analysis for the CBCD-type methods. Specifically, we show that for a family of quadratic minimization problems, the iteration complexity of the CBCD-type methods matches that of the GD

methods in term of dependency on p up to a $\log^2(p)$ factor. More precisely, when $\mathcal{L}(x)$ is quadratic, the iteration complexity of the CBCGD method is

$$\mathcal{O}\left(\frac{\log^2(p)L^2\log(1/\epsilon)}{L_{\min}^{\mu} \cdot \mu}\right), \tag{5}$$

where $L_{\min}^{\mu} = \min_i\{L_i + \mu_i\}$ and μ_i is the strongly convex constant with respect to the i-th block of variables. Note that $L_{\min}^{\mu} \geq L_{\min}$. As can be seen easily, (5) is better than (4) by a factor of at least $\frac{L_{\max}\cdot p}{L_{\min}\cdot\log^2(p)}$. Note that We also provide a lower bound analysis that confirms that our improved iteration complexity is tight up to a $\log^2(p)$ factor if the largest and smallest eigenvalues of the Hessian matrix do not scale with p. Similar results hold for the CBCM method. We remark here that when the problem is quadratic (e.g., ridge-penalized linear regression with squared loss), (1) can be written as solving a linear system. Then the CBCM method is equivalent to the Gauss-Siedel method, which also has a linear convergence rate (Golub and Van Loan, 2012). Nevertheless, our major effort is to improve the dependence of the constant factor on the problem parameters (e.g., the block size p and Lipchitz constant L) in the iteration complexity, which is more difficult to analyze in the blockwise minimization case for the Gauss-Siedel method.[1] In addition, the Gauss-Siedel method is not applicable beyond the quadratic case in general.

Finally, we generalize our analysis to other strongly convex minimization problems beyond quadratic minimization. Specifically, for smooth minimization, the iteration complexity of the CBCGD method is

$$\mathcal{O}\left(\frac{L_{\max}^{\beta} \cdot pL^{\beta} \log(1/\epsilon)}{L_{\min}^{\beta} \cdot \mu}\right), \tag{6}$$

where $L^{\beta} = L + \beta$, $L_{\max}^{\beta} = \max_i\{L_i + \beta_i\}$, $L_{\min}^{\beta} = \min_i\{L_i + \beta_i\}$, β is the Lipchitz constant of gradient $\nabla\mathcal{R}(\cdot)$, and β_i is the Lipchitz constant of i-th block of gradient $\nabla_i\mathcal{R}(\cdot)$ for smooth $\mathcal{R}(\cdot)$. Note that L^{β}, $L_{\max}^{\beta}$, and $L_{\min}^{\beta}$ are the Lipchitz constants of the gradient (and the corresponding blocks) of the objective function, which are identical to those considered in Beck and Tetruashvili (2013) when the objective is of the composite form as in (1). This indicates that (6) is better than (4) by a factor of $L^{\beta}/L_{\min}^{\beta}$, which is at least of order $\sqrt{p}$ and can be much more significant for ill-conditioned problems. Similar results hold for nonsmooth regularized minimization and their counter parts for the CBCM method; for more details refer to Table 1. It is worth mentioning that all the above results on the CBCD-type methods can be used to establish the iteration complexity for the popular permuted BCM (PBCM) and permuted BCGD (PBCGD) methods, in which the blocks are randomly sampled without replacement in each round. Improvement in terms of the dependence on constants are provided in Sun and Hong (2015) for the nonstronlgy convex problems.

The rest of the paper is organized as follows. In Section 2, we introduce notations and preliminary assumptions. Then we provide the main results of improved convergence

1. It requires to find an upper bound of the contraction constant $\|H_L^{-1}(H - H_L)\|$ in the Gauss-Siedel method, where H is the coefficient matrix of the linear system and H_L is the lower triangular matrix of H. Note that it requires $\|H_L^{-1}(H - H_L)\| < 1$ for the convergence of the Gauss-Siedel method.

Table 1: Compared with Beck and Tetruashvili (2013), our contributions are manyfold: (1) Developing the iteration complexity bounds of the CBCM and CBCGD methods for different specifications on $\mathcal{L}(\cdot)$ and $\mathcal{R}(\cdot)$; (2) Developing the iteration complexity bound of CBCGD for quadratic $\mathcal{L}(\cdot)$ + nonsmooth $\mathcal{R}(\cdot)$; (3) Improving the iteration complexity bound of CBCGD for smooth $\mathcal{R}(\cdot)$.

	Method	$\mathcal{L}(\cdot)$	$\mathcal{R}(\cdot)$	Our analysis	Beck & Tetruashvili (2013)
[a]	CBCGD	Quadratic	Smooth	$\mathcal{O}\left(\frac{\log^2(p)L^2\log(1/\epsilon)}{L^\mu_{\min}\cdot\mu}\right)$	$\mathcal{O}\left(\frac{L_{\max}\cdot pL^2\log(1/\epsilon)}{L^2_{\min}\cdot\mu}\right)$
[b]	CBCGD	Quadratic	Nonsmooth	$\mathcal{O}\left(\frac{\log^2(p)L^2\log(1/\epsilon)}{L^\mu_{\min}\cdot\mu}\right)$	N/A
[c]	CBCGD	General Convex	Smooth	$\mathcal{O}\left(\frac{L^\beta_{\max}\cdot pL^\beta\log(1/\epsilon)}{L^\beta_{\min}\cdot\mu}\right)$	$\mathcal{O}\left(\frac{L^\beta_{\max}\cdot p(L^\beta)^2\log(1/\epsilon)}{(L^\beta_{\min})^2\cdot\mu}\right)$
[d]	CBCGD	General Convex	Nonsmooth	$\mathcal{O}\left(\frac{L_{\max}\cdot pL\log(1/\epsilon)}{L^\mu_{\min}\cdot\mu}\right)$	N/A
[e]	CBCM	Quadratic	Smooth	$\mathcal{O}\left(\frac{\log^2(p)L^2\log(1/\epsilon)}{\mu_{\min}\cdot\mu}\right)$	N/A
[f]	CBCM	Quadratic	Nonsmooth	$\mathcal{O}\left(\frac{\log^2(p)L^2\log(1/\epsilon)}{\mu_{\min}\cdot\mu}\right)$	N/A
[g]	CBCM	General Convex	Smooth	$\mathcal{O}\left(\frac{L_{\max}\cdot pL\log(1/\epsilon)}{\mu_{\min}\cdot\mu}\right)$	N/A
[h]	CBCM	General Convex	Nonsmooth	$\mathcal{O}\left(\frac{L_{\max}\cdot pL\log(1/\epsilon)}{\mu_{\min}\cdot\mu}\right)$	N/A

Remark: See Theorem 3 for [a] and [b]; See Theorem 4 for [e] and [f]; See Theorem 7 for [c]; See Theorem 8 for [d], [g], and [h]. When $\mathcal{R}(\cdot)$ is nonsmooth, the optimization problem is actually solved by the cyclic block coordinate proximal gradient (CBCPGD) method. For notational convenience in this paper, however, we simply call it the CBCGD method.

analysis for CBCD-type approachs in Section 3. Numerical evaluations are provided in Section 4, followed by further discussions in Section 5.

2. Notations and Assumptions

We start with introducing notations used in this paper. Given a vector $\boldsymbol{v} = (v_1, \ldots, v_d)^\top \in \mathbb{R}^d$, we define vector norms: $\|\boldsymbol{v}\|_1 = \sum_j |v_j|$, $\|\boldsymbol{v}\|^2 = \sum_j v_j^2$, and $\|\boldsymbol{v}\|_\infty = \max_j |v_j|$. Let $\{\mathcal{A}_1, \ldots, \mathcal{A}_p\}$ be a partition of all d coordinates with $|\mathcal{A}_j| = d_j$ and $\sum_{j=1}^p d_j = d$. We use $\boldsymbol{v}_j$ to denote the subvector of $\boldsymbol{v}$ with all indices in $\mathcal{A}_j$. Given a matrix $A \in \mathbb{R}^{d\times d}$, we use $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ to denote the largest and smallest eigenvalues of A. We denote $\|A\|$ as the spectral norm of A (i.e., the largest singular value). We denote $\otimes$ and $\odot$ as the Kronecker product and Hadamard (entrywise) product for two matrices respectively.

Before we proceed with our analysis, we introduce some assumptions on $\mathcal{L}(\cdot)$ and $\mathcal{R}(\cdot)$.

Assumption 1 *$\mathcal{L}(\cdot)$ is convex, and its gradient mapping $\nabla\mathcal{L}(\cdot)$ is Lipschitz continuous and also blockwise Lipschitz continuous, i.e., there exist positive constants L and L_j's such that for any $x, x' \in \mathbb{R}^d$ and $j = 1, \ldots, p$, we have*

$$\|\nabla\mathcal{L}(x') - \nabla\mathcal{L}(x)\| \le L\|x - x'\| \text{ and } \|\nabla_j\mathcal{L}\left(x_{1:(j-1)}, x'_j, x_{(j+1):p}\right) - \nabla_j\mathcal{L}(x)\| \le L_j\|x_j - x'_j\|.$$

We define $L_{\max} = \max_j L_j$ and $L_{\min} = \min_j L_j$.

Assumption 2 *$\mathcal{R}(\cdot)$ is strongly convex and also blockwise strongly convex, i.e., there exist positive constants μ and μ_j's such that for any $x, x' \in \mathbb{R}^d$ and $j = 1, \ldots, p$, we have*

$$\mathcal{R}(x) \ge \mathcal{R}(x') + (x - x')^\top\xi' + \frac{\mu}{2}\|x - x'\|^2 \text{ and}$$
$$\mathcal{R}_j(x_j) \ge \mathcal{R}_j(x'_j) + (x_j - x'_j)^\top\xi'_j + \frac{\mu_j}{2}\|x_j - x'_j\|^2,$$

for all $\xi' \in \partial\mathcal{R}(x')$. We define $\mu_{\min} = \min_j \mu_j$.

For notational simplicity, we define auxiliary variables

$$L^\mu_{\min} = \min_j L_j + \mu_j \text{ and } y^{(t,j)} = [x^{(t)\top}_{1:(j-1)}, x^{(t-1)\top}_{j:p}]^\top,\ j = 1, \ldots, p. \tag{7}$$

We remark that $y^{(t,j)}$ serves as an intermediate variable in our analysis, which has the first $j-1$ blocks of variables updated and the remaining blocks unchanged at t-th iteration. Our analysis considers $L_{\min}$, $L_{\max}$, $L^\mu_{\min}$, $\mu_{\min}$, μ, and $d_{\max} = \max_j d_j$ as constants, which do not scale with the block size p as in existing literature (Beck and Tetruashvili, 2013).

3. Improved Convergence Analysis

Our analysis consists of the following three steps:

(1) Characterize the successive descent after each CBCD iteration;

(2) Characterize the gap towards the optimal objective value after each CBCD iteration;

(3) Combine (1) and (2) to establish the iteration complexity bound.

We present our analysis under different specifications on $\mathcal{L}(\cdot)$ and $\mathcal{R}(\cdot)$.

3.1 Quadratic Minimization

We first consider a scenario, where $\mathcal{L}(\cdot)$ is a quadratic function. Particularly, we solve

$$x^* = \operatorname*{argmin}_{x\in\mathbb{R}^d} \mathcal{L}(x) + \mathcal{R}(x) = \operatorname*{argmin}_{\substack{x_j\in\mathbb{R}^{d_j}\\ j=1,\ldots,p}} \frac{1}{2}\left\|\sum_{j=1}^p A_{*j}x_j - b\right\|^2 + \sum_{j=1}^p \mathcal{R}_j(x_j), \tag{8}$$

where $A_{*j} \in \mathbb{R}^{n\times d_j}$ for $j = 1, ..., p$. Typical applications of (8) in statistical machine learning include ridge regression, elastic-net penalized regression, and sparse additive regression.

We first characterize the successive descent of the CBCGD method.

Lemma 1 *Recall that $\mathcal{F}$ is the objective defined in (1). Suppose that Assumptions 1 and 2 hold. We choose $\eta_j = L_j$ for the CBCGD method. Then for all $t \ge 1$, we have*

$$\mathcal{F}(x^{(t)}) - \mathcal{F}(x^{(t+1)}) \ge \frac{L^\mu_{\min}}{2}\|x^{(t)} - x^{(t+1)}\|^2.$$

Proof At t-th iteration, there exists a $\xi_j^{(t+1)} \in \partial \mathcal{R}_j(x_j^{(t+1)})$ satisfying the optimality condition for each sub-problem:

$$\nabla_j \mathcal{L}(y^{(t+1,j)}) + \eta_j(x_j^{(t+1)} - x_j^{(t)}) + \xi_j^{(t+1)} = 0. \tag{9}$$

Then by definition of CBCGD given in (3), we have

$$\begin{aligned}\mathcal{F}(y^{(t+1,j+1)}) \leq \mathcal{L}(y^{(t+1,j)}) &+ (y^{(t+1,j+1)} - y^{(t+1,j)})^\top \nabla \mathcal{L}(y^{(t+1,j)}) \\ &+ \frac{L_j}{2}\|y^{(t+1,j)} - y^{(t+1,j+1)}\|^2 + \mathcal{R}(y^{(t+1,j+1)}).\end{aligned}$$

This further implies

$$\begin{aligned}&\mathcal{F}(y^{(t+1,j)}) - \mathcal{F}(y^{(t+1,j+1)}) = \mathcal{L}(y^{(t+1,j)}) + \mathcal{R}(y^{(t+1,j)}) \\ &\geq (y^{(t+1,j)} - y^{(t+1,j+1)})^\top \nabla \mathcal{L}(y^{(t+1,j)}) - \frac{L_j}{2}\|y^{(t+1,j)} - y^{(t+1,j+1)}\|^2 \\ &\qquad + \mathcal{R}(y^{(t+1,j)}) - \mathcal{R}(y^{(t+1,j+1)}) \\ &= (x_j^{(t)} - x_j^{(t+1)})^\top \nabla_j \mathcal{L}(y^{(t+1,j+1)}) - \frac{L_j}{2}\|x_j^{(t+1)} - x_j^{(t)}\|^2 + \mathcal{R}_j(x_j^{(t)}) - \mathcal{R}_j(x_j^{(t+1)}).\end{aligned} \tag{10}$$

By Assumptions 2, we have

$$\mathcal{R}_j(x_j^{(t)}) - \mathcal{R}_j(x_j^{(t+1)}) \geq (x_j^{(t)} - x_j^{(t+1)})^\top \xi_j^{(t+1)} + \frac{\mu_j}{2}\|x_j^{(t)} - x_j^{(t+1)}\|^2. \tag{11}$$

Combining (9), (10), and (11), we have

$$\mathcal{F}(y^{(t+1,j)}) - \mathcal{F}(y^{(t+1,j+1)}) \geq \frac{L_j + \mu_j}{2}\|x_j^{(t)} - x_j^{(t+1)}\|^2. \tag{12}$$

We complete the proof via summation of (12) over $j = 1, \ldots, p$ and the definition of $L_{\min}^{\mu}$. ■

Next, we characterize the gap towards the optimal objective value.

Lemma 2 *Suppose that Assumptions 1 and 2 hold with $d \geq 2$. Then for all $t \geq 1$, we have*

$$\mathcal{F}(x^{(t+1)}) - \mathcal{F}(x^*) \leq \frac{8L^2 \log^2(3pd_{\max})}{\mu}\|x^{(t+1)} - x^{(t)}\|^2.$$

Proof Since $\mathcal{L}(x)$ is quadratic, its second order Taylor expansion is tight, i.e.

$$\mathcal{L}(x^*) = \mathcal{L}(x^{(t+1)}) + \langle \nabla \mathcal{L}(x^{(t+1)}), x^* - x^{(t+1)} \rangle + \frac{1}{2}\|A(x^{(t+1)} - x^*)\|^2, \tag{13}$$

where $A = [A_{*1}, \ldots, A_{*p}] \in \mathbb{R}^{n \times d}$.

Consider matrices $\widetilde{P}$ and $\widetilde{A}$, defined as

$$\widetilde{P} = \begin{bmatrix} L_1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & L_2 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\ 0 & 0 & & \ldots & 0 & L_p \end{bmatrix} \in \mathbb{R}^{p \times p} \quad \text{and} \quad \widetilde{A} = \begin{bmatrix} A_{*1} & 0 & 0 & \ldots & 0 & 0 \\ 0 & A_{*2} & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\ 0 & 0 & & \ldots & 0 & A_{*p} \end{bmatrix} \in \mathbb{R}^{np \times d}.$$

For simplicity, we assume that $d_1 = \ldots = d_p = m = d/p$. For any $s \in \mathbb{Z}^+$, we define the lower triangular matrix $D_s \in \mathbb{R}^{s \times s}$ as

$$D_s = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 & 0 \\ 1 & 1 & 0 & \ldots & 0 & 0 \\ 1 & 1 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\ 1 & 1 & 1 & \ldots & 1 & 1 \end{bmatrix}$$

By the definition of L_j, we have that for all $j = 1, \ldots, p$,

$$L_j \geq \lambda_{\max}(A_j^\top A_j).$$

which gives us the following inequality

$$\widetilde{P} \otimes I_m \succeq \widetilde{A}^\top \widetilde{A}. \tag{14}$$

To characterize the gap towards the optimal objective, we have

$$\begin{aligned}&\mathcal{F}(x^{(t+1)}) - \mathcal{F}(x^*) + \frac{\mu}{2}\|x^{(t+1)} - x^*\|^2 \\ &\overset{\text{(i)}}{\leq} \langle \nabla \mathcal{L}(x^{(t+1)}), x^{(t+1)} - x^* \rangle + \mathcal{R}(x^{(t+1)}) - \mathcal{R}(x^*) + \frac{\mu}{2}\|x^{(t+1)} - x^*\|^2 \\ &\overset{\text{(ii)}}{\leq} \langle \nabla \mathcal{L}(x^{(t+1)}), x^{(t+1)} - x^* \rangle + \langle \xi^{(t+1)}, x^{(t+1)} - x^* \rangle \\ &\overset{\text{(iii)}}{\leq} \sum_{j=1}^{p} \langle \nabla_j \mathcal{L}(x^{(t+1)}) - \nabla_j \mathcal{L}\left(x^{(t)}\right), x_j^{(t+1)} - x_j^* \rangle - \sum_{j=1}^{p} L_j \langle x_j^{(t+1)} - x_j^{(t)}, x_j^{(t+1)} - x_j^* \rangle \\ &= \sum_{j=1}^{p} \Big\langle \sum_{k \geq j} A_k(x_k^{(t+1)} - x_k^{(t)}), A_j(x_j^{(t+1)} - x_j^*) \Big\rangle - (x^{(t+1)} - x^{(t)})^\top (\widetilde{P} \otimes I_m)(x^{(t+1)} - x^*) \\ &\leq (x^{(t+1)} - x^{(t)})^\top \widetilde{A}^\top (D_p \otimes I_n) \widetilde{A}(x^{(t+1)} - x^*) - (x^{(t+1)} - x^{(t)})^\top (\widetilde{P} \otimes I_m)(x^{(t+1)} - x^*) \\ &= (x^{(t+1)} - x^{(t)})^\top \left(\widetilde{A}^\top (D_p \otimes I_n) \widetilde{A} - \widetilde{P} \otimes I_m \right) (x^{(t+1)} - x^*) \\ &\overset{\text{(iv)}}{=} (x^{(t+1)} - x^{(t)})^\top \left(\left(A^\top A - \widetilde{A}^\top \widetilde{A} \right) \odot D_d + \widetilde{A}^\top \widetilde{A} - \widetilde{P} \otimes I_m \right) (x^{(t+1)} - x^*),\end{aligned}$$

where (i) is from (13) as $\|A(x^{(t+1)} - x^*)\|^2 \geq 0$, (ii) is from Assumption 2, (iii) is from the optimality condition to the subproblem associated with x_j,

$$\langle \nabla_j \mathcal{L}(y^{(t+1,j)}) + L_j(x_j^{(t+1)} - x_j^{(t)}) + \xi_j^{(t+1)}, x_j - x_j^{(t+1)} \rangle \geq 0 \text{ for any } x_j \in \mathbb{R}^m,$$

and (iv) comes from the fact that

$$\widetilde{A}^\top (D_p \otimes I_m) \widetilde{A} = \left(A^\top A - \widetilde{A}^\top \widetilde{A} \right) \odot D_d + \widetilde{A}^\top \widetilde{A}, \tag{15}$$

where $\odot$ denotes the Hadamard product. We rewrite (15) in this way because a tight upper bound of the spectral norm of R.H.S. is easier to be obtained than the spectral norm of L.H.S. after subtracting $\widetilde{P} \otimes I_m$.

For notational convenience, we define

$$B = \left(A^\top A - \widetilde{A}^\top \widetilde{A}\right) \odot D_d + \widetilde{A}^\top \widetilde{A} - \widetilde{P} \otimes I_m,$$

then we have

$$\mathcal{F}(x^{(t+1)}) - \mathcal{F}(x^*) \leq (x^{(t+1)} - x^{(t)})^\top B(x^{(t+1)} - x^*) - \frac{\mu}{2}\|x^{(t+1)} - x^*\|^2. \tag{16}$$

Minimizing the R.H.S. of the above inequality over x^*, we obtain

$$-\mu(x^* - x^{(t+1)}) - B^\top(x^{(t+1)} - x^{(t)}) = 0.$$

which implies

$$x^* = -\frac{B^\top(x^{(t+1)} - x^{(t)})}{\mu} + x^{(t+1)}. \tag{17}$$

In addition, we have

$$\begin{aligned}
\|B\| &\overset{\text{(i)}}{\leq} \|\left(A^\top A - \widetilde{A}^\top \widetilde{A}\right) \odot D_d\| + \|\widetilde{A}^\top \widetilde{A} - \widetilde{P} \otimes I_m\| \\
&\overset{\text{(ii)}}{\leq} \|A^\top A - \widetilde{A}^\top \widetilde{A}\| \left(1 + \frac{1}{\pi} + \frac{\log(d)}{\pi}\right) + \|\widetilde{A}^\top \widetilde{A} - \widetilde{P} \otimes I_m\| \\
&\overset{\text{(iii)}}{\leq} \left(\|A^\top A\| + \|\widetilde{A}^\top \widetilde{A}\|\right) \left(1 + \frac{1}{\pi} + \frac{\log(d)}{\pi}\right) + \|\widetilde{A}^\top \widetilde{A} - \widetilde{P} \otimes I_m\| \\
&\overset{\text{(iv)}}{\leq} 4\|A^\top A\| \left(1 + \frac{1}{\pi} + \frac{\log(d)}{\pi}\right) \overset{\text{(v)}}{\leq} 4L\log(3pd_{\max}),
\end{aligned} \tag{18}$$

where (i) and (iii) are from the triangle inequality, (iv) is from (14) and the fact that $\|\widetilde{A}^\top \widetilde{A}\| \leq \|A^\top A\|$, and (v) is from $\|A^\top A\| \leq L$, $1 + \frac{1}{\pi} + \frac{\log(d)}{\pi} \leq \log(3d)$ for all $d \geq 2$, and $d \leq pd_{\max}$. Inequality (ii) follows from the result on the spectral norm of the triangular truncation operator in Angelos et al. (1992) (Theorem 1). More specifically, let us define

$$L_d = \max\left\{\frac{\|A \odot D_d\|}{\|A\|} : A \in \mathbb{R}^{d\times d}, A \neq 0\right\},$$

which is the largest ratio between the spectral norm of the triangular truncation of A (the Hadamard product of A and D_d) and the spectral norm of A. Then for any $d \geq 2$, we have from Angelos et al. (1992) that

$$\left|\frac{L_d}{\log d} - \frac{1}{\pi}\right| \leq \frac{\left(1 + \frac{1}{\pi}\right)}{\log d}.$$

Plugging (17) into (16), we obtain

$$\begin{aligned}
\mathcal{F}(x^{(t+1)}) - \mathcal{F}(x^*) &\leq \frac{1}{2\mu}\|B(x^{(t+1)} - x^{(t)})\|^2 \overset{\text{(i)}}{\leq} \frac{\|B\|^2}{2\mu}\|x^{(t+1)} - x^{(t)}\|^2 \\
&\overset{\text{(ii)}}{\leq} \frac{8L^2\log^2(3pd_{\max})}{\mu}\|x^{(t+1)} - x^{(t)}\|^2,
\end{aligned}$$

where (i) comes from the Cauchy-Schwarz inequality, (ii) is from (18). ■

Using Lemmas 1 and 2, we establish the iteration complexity bound of the CBCGD method for minimizing (8) in the next theorem.

Theorem 3 *Suppose that Assumptions 1 and 2 hold with $d \geq 2$. We choose $\eta_j = L_j$ for the CBCGD method. Given a pre-specified accuracy $\epsilon > 0$ of the objective value, we need at most*

$$\left\lceil \frac{\mu L^\mu_{\min} + 16L^2\log^2(3pd_{\max})}{\mu L^\mu_{\min}} \log\left(\frac{\mathcal{F}(x^{(0)}) - \mathcal{F}(x^*)}{\epsilon}\right)\right\rceil$$

iterations for the CBCGD method to ensure that $\mathcal{F}(x^{(t)}) - \mathcal{F}(x^*) \leq \epsilon$, where $L^\mu_{\min}$ is defined in (7).

Proof Combining Lemmas 1 and 2, we obtain

$$\begin{aligned}
\mathcal{F}(x^{(t)}) - \mathcal{F}(x^*) &= [\mathcal{F}(x^{(t)}) - \mathcal{F}(x^{(t+1)})] + [\mathcal{F}(x^{(t+1)}) - \mathcal{F}(x^*)] \\
&\geq \frac{L^\mu_{\min}}{2}\|x^{(t)} - x^{(t+1)}\|^2 + [\mathcal{F}(x^{(t+1)}) - \mathcal{F}(x^*)] \\
&\geq \left(1 + \frac{L^\mu_{\min}\mu}{16L^2\log^2(3pd_{\max})}\right)[\mathcal{F}(x^{(t+1)}) - \mathcal{F}(x^*)].
\end{aligned}$$

Recursively applying the above inequality for $t \geq 1$, we obtain

$$\frac{\mathcal{F}(x^{(t)}) - \mathcal{F}(x^*)}{\mathcal{F}(x^{(0)}) - \mathcal{F}(x^*)} \leq \left(1 - \frac{\mu L^\mu_{\min}}{\mu L^\mu_{\min} + 16L^2\log^2(3pd_{\max})}\right)^t.$$

To ensure $\mathcal{F}(x^{(t)}) - \mathcal{F}(x^*) \leq \epsilon$, we only need a large enough t to ensure that

$$\left(1 - \frac{\mu L^\mu_{\min}}{\mu L^\mu_{\min} + 16L^2\log^2(3pd_{\max})}\right)^t [\mathcal{F}(x^{(0)}) - \mathcal{F}(x^*)] \leq \epsilon. \tag{19}$$

We complete the proof by combining (19) and the basic inequality $\kappa \geq \log^{-1}\left(\frac{\kappa}{\kappa-1}\right)$. ■

As can be seen in Theorem 3, the iteration complexity depends on p only in the order of $\log^2(p)$, which is generally mild in practice. The iteration complexity of the CBCM method can be established in a similar manner.

Theorem 4 *Suppose that Assumptions 1 and 2 hold with $d \geq 2$. Given a pre-specified accuracy ϵ, we need at most*

$$\left\lceil \frac{\mu\mu_{\min} + 64L^2\log^2(3pd_{\max})}{\mu\mu_{\min}} \log\left(\frac{\mathcal{F}(x^{(0)}) - \mathcal{F}(x^*)}{\epsilon}\right)\right\rceil$$

iterations for the CBCM method such that $\mathcal{F}(x^{(t)}) - \mathcal{F}(x^*) \leq \epsilon$

Proof The overall proof also consists of three major steps: (i) successive descent, (ii) gap towards the optimal objective value, and (iii) iteration complexity.

Successive Descent: At t-th iteration, there exists a $\xi_j^{(t+1)} \in \partial\mathcal{R}_j(x_j^{(t+1)})$ satisfying the optimality condition:

$$\nabla_j\mathcal{L}(y^{(t+1,j+1)}) + \xi_j^{(t+1)} = 0. \tag{20}$$

Then we have

$$\begin{aligned}
&\mathcal{F}(y^{(t+1,j)}) - \mathcal{F}(y^{(t+1,j+1)}) \overset{\text{(i)}}{\geq} (x_j^{(t)} - x_j^{(t+1)})^\top \nabla_j\mathcal{L}(y^{(t+1,j+1)}) + \mathcal{R}_j(x_j^{(t)}) - \mathcal{R}_j(x_j^{(t+1)}) \\
&\overset{\text{(ii)}}{\geq} \left(\nabla_j\mathcal{L}(y^{(t+1,j+1)}) + \xi_j^{(t+1)}\right)^\top (x_j^{(t)} - x_j^{(t+1)}) + \frac{\mu_j}{2}\|x_j^{(t)} - x_j^{(t+1)}\|^2 \\
&\overset{\text{(iii)}}{=} \frac{\mu_j}{2}\|x_j^{(t)} - x_j^{(t+1)}\|^2,
\end{aligned} \tag{21}$$

where (i) is from the convexity of $\mathcal{L}(\cdot)$, (ii) is from Assumptions 2, and (iii) is from (20). By summation of (12) over $j = 1, \ldots, p$, we have

$$\mathcal{F}(x^{(t)}) - \mathcal{F}(x^{(t+1)}) \geq \frac{\mu_{\min}}{2}\|x^{(t)} - x^{(t+1)}\|^2.$$

Gap towards the Optimal Objective Value: The proof follows the same arguments with the proof of Lemma 2, with a few differences.

First,with the optimality condition to the subproblem associated with x_j,

$$\langle \nabla_j\mathcal{L}(x^{(t)}) + \xi_j^{(t+1)}, x_j - x_j^{(t+1)}\rangle \geq 0 \text{ for any } x_j \in \mathbb{R}^m,$$

we have

$$\mathcal{F}(x^{(t+1)}) - \mathcal{F}(x^*) \leq (x^{(t+1)} - x^{(t)})^\top B(x^{(t+1)} - x^*) - \frac{\mu}{2}\|x^{(t+1)} - x^*\|^2,$$

where $B = \left(A^\top A - \widetilde{A}^\top\widetilde{A}\right) \odot D_d + \widetilde{A}^\top\widetilde{A}$.

Then, using the same technique to bound the eigenvalues for matrices with Hadamard product, we have

$$\mathcal{F}(x^{(t+1)}) - \mathcal{F}(x^*) \leq \frac{L^2\log^2(3d) + L_{\max}^2}{\mu}\|x^{(t+1)} - x^{(t)}\|^2 \leq \frac{2L^2\log^2(3d)}{\mu}\|x^{(t+1)} - x^{(t)}\|^2.$$

Iteration Complexity: The analysis follows from the counter part of Theorem 3. ■

Theorem 4 establishes that the iteration complexity of the CBCM method matches that of the CBCGD method. To the best of our knowledge, Theorems 3 and 4 are the sharpest iteration complexity analysis of the CBCD-type methods for minimizing (8). We further provide an example to establish the tightness of the above result in Appendix A.

3.2 General Smooth Minimization

We next consider general strongly convex smooth minimization, which includes Beck and Tetruashvili (2013) as a special case with $\mathcal{R}(x) = 0$. Here we require $\mathcal{R}(x)$ to be smooth and strongly convex.

Assumption 3 *$\mathcal{R}(\cdot)$ is smooth and also blockwise smooth, i.e., there exist positive constants β and β_j's such that for $x, x' \in \mathbb{R}^d$ and $j = 1, \ldots, p$, we have*

$$\mathcal{R}(x) \leq \mathcal{R}(x') + (x - x')^\top\nabla\mathcal{R}(x') + \frac{\beta}{2}\|x - x'\|^2 \text{ and}$$

$$\mathcal{R}_j(x_j) \leq \mathcal{R}_j(x_j') + (x_j - x_j')^\top\nabla_j\mathcal{R}(x') + \frac{\beta_j}{2}\|x_j - x_j'\|^2.$$

Moreover, we define $\beta_{\max} = \max_j \beta_j$.

Moreover, we assume that the Hessian matrix H of the objective function $\mathcal{F}$ exists, which is denoted as $H_{ij}(x) = \frac{\partial\mathcal{F}(x)}{\partial x_i \partial x_j}$.

Since the objective function is globally smooth, the CBCGD method can directly take the update form: For $j = 1, \ldots, p$,

$$x_j^{(t+1)} = x_j^{(t)} - \eta_j\left(\nabla_j\mathcal{L}(y^{(t+1,j+1)}) + \nabla\mathcal{R}_j(x_j^{(t)})\right),$$

where $\eta_j > 0$ is a step-size parameter for the j-th block.

Typical applications of the general strongly convex smooth minimization in statistical machine learning includes ridge penalized logistic regression, and ridge penalized multinomial regression. It is worth mentioning that our analysis for the general case is applicable to smooth quadratic minimization, but is very different from the analysis in previous sections for quadratic minimization.

We first characterize the successive descent after each coordinate gradient descent (CGD) iteration.

Lemma 5 *Suppose that Assumptions 1 and 3 hold. We choose $\eta_j = L_j + \beta_j$ for the CBCGD method. Then for all $t \geq 1$, there exists $\boldsymbol{z}^{(t,j)}$ in the line segment of $(x^{(t)}, y^{(t,j)})$ for each $j \in \{1, \ldots, p\}$ such that*

$$\mathcal{F}(x^{(t)}) - \mathcal{F}(x^{(t+1)}) \geq \frac{\|\nabla\mathcal{F}(x^{(t)})\|^2}{2\left(L_{\max}^\beta + \frac{\|H\|^2}{L_{\min}^\beta}\right)},$$

where H is defined as

$$H \triangleq \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 & 0 \\ H_{21} & 0 & 0 & \ldots & 0 & 0 \\ H_{31} & H_{32} & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ldots & \ddots & \vdots \\ H_{p1} & H_{p2} & H_{p3} & \ldots & H_{p,p-1} & 0 \end{bmatrix}, \tag{22}$$

with $H_{ji} \triangleq H_{ji}(\boldsymbol{z}^{(t,j)}) = \frac{\partial\mathcal{F}(\boldsymbol{z}^{(t,j)})}{\partial z_j \partial z_i}$, and $L_{\min}^\beta$ and $L_{\max}^\beta$ are defined as

$$L_{\min}^\beta = \min\{L_j^\beta = L_j + \beta_j, j = 1, \ldots, p\} \quad \text{and} \quad L_{\max}^\beta = \max\{L_j^\beta = L_j + \beta_j, j = 1, \ldots, p\}.$$

Proof We first provide a lower bound of the successive descent using the gradient of $\mathcal{F}(\cdot)$ based on the Lipschitz continuity of $\nabla\mathcal{F}(\cdot)$. We have that $y^{(t,j)}$ and $y^{(t,k+1)}$ only differ at the k-th coordinate, and $\nabla_j\mathcal{F}(y^{t,k})$ has Lipschitz gradient with Lipschitz constant F_j, which implied

$$\begin{aligned}\mathcal{F}(y^{(t,j+1)}) &\leq \mathcal{F}(y^{(t,j)}) + (y^{(t,j+1)} - y^{(t,j)})^\top \nabla_j \mathcal{F}(y^{(t,j)}) + \frac{F_j}{2}\|y^{(t,j+1)} - y^{(t,j)}\|^2 \\ &\overset{\text{(i)}}{=} \mathcal{F}(y^{(t,j)}) - \frac{2L_j^\beta - F_j}{2(L_j^\beta)^2}\|\nabla_j \mathcal{F}(y^{(t,j)})\|^2 \overset{\text{(ii)}}{\leq} \mathcal{F}(y^{(t,j)}) - \frac{1}{2L_j^\beta}\|\nabla_j \mathcal{F}(y^{(t,j)})\|^2,\end{aligned}$$

where (i) is from that $x_j^{(t+1)} = x_j^{(t)} - \frac{\nabla_j \mathcal{F}(y^{(t,j)})}{L_j^\beta}$, and (ii) is from the fact that $L_j^\beta \geq F_j$. Then the decrease of the objective is

$$\mathcal{F}(x^{(t)}) - \mathcal{F}(x^{(t+1)}) = \sum_{k=1}^p \mathcal{F}(y^{(t,j)}) - \mathcal{F}(y^{(t,j+1)}) \geq \sum_{k=1}^p \frac{1}{2L_j^\beta}\|\nabla_j \mathcal{F}(y^{(t,j)})\|^2. \tag{23}$$

For simplicity, we assume that $d_1 = ... = d_p = m = d/p$. By the Mean Value Theorem, there exists $\boldsymbol{z}^{(t,j)}$ such that

$$\begin{aligned}&\nabla_j \mathcal{F}(x^{(t)}) = \nabla_j \mathcal{F}(x^{(t)}) - \nabla_j \mathcal{F}(y^{(t,j)}) + \nabla_j \mathcal{F}(y^{(t,j)}) \\ &\overset{\text{(i)}}{=} \nabla(\nabla_j \mathcal{F}(\boldsymbol{z}^{(t,j)}))^\top (x^{(t)} - y^{(t,j)}) + \nabla_j \mathcal{F}(y^{(t,j)}) \\ &= \left[\tfrac{\partial \mathcal{F}(\boldsymbol{z}^{(t,j)})}{\partial z_j \partial z_1}, \ldots, \tfrac{\partial \mathcal{F}(\boldsymbol{z}^{(t,j)})}{\partial z_j \partial z_{j-1}}, 0, \ldots, 0\right] \left[\tfrac{\left(x_1^{(t)} - x_1^{(t+1)}\right)^\top}{L_1^\beta}, \ldots, \tfrac{\left(x_{j-1}^{(t)} - x_{j-1}^{(t+1)}\right)^\top}{L_{j-1}^\beta}, 0, \ldots, 0\right]^\top + \nabla_j \mathcal{F}(y^{(t,j)}) \\ &= \left[\tfrac{H_{j1}}{\sqrt{L_1^\beta}}, \ldots, \tfrac{H_{j,j-1}}{\sqrt{L_{j-1}^\beta}}, 0, \ldots, 0\right] \left[\tfrac{\left(x_1^{(t)} - x_1^{(t+1)}\right)^\top}{\sqrt{L_1^\beta}}, \ldots, \tfrac{\left(x_{j-1}^{(t)} - x_{j-1}^{(t+1)}\right)^\top}{\sqrt{L_{j-1}^\beta}}, 0, \ldots, 0\right]^\top + \nabla_j \mathcal{F}(y^{(t,j)}) \\ &= \left[\tfrac{H_{j1}}{\sqrt{L_1^\beta}}, \ldots, \tfrac{H_{j,j-1}}{\sqrt{L_{j-1}^\beta}}, \sqrt{L_j^\beta} \cdot I_m, 0, \ldots, 0\right] \left[\tfrac{\nabla_1 \mathcal{F}(y^{(t,1)})^\top}{\sqrt{L_1^\beta}}, \ldots, \tfrac{\nabla_p \mathcal{F}(y^{(t,p)})^\top}{\sqrt{L_p^\beta}}\right]^\top = h_j^\top f\end{aligned}$$

where (i) is from the mean-value theorem, $h_j = \left[\frac{H_{j1}}{\sqrt{L_1^\beta}}, \ldots, \frac{H_{j,j-1}}{\sqrt{L_{j-1}^\beta}}, \sqrt{L_j^\beta} \cdot I_m, 0, \ldots, 0\right]^\top$ and $f = \left[\frac{\nabla_1 \mathcal{F}(y^{(t,1)})^\top}{\sqrt{L_1^\beta}}, \ldots, \frac{\nabla_p \mathcal{F}(y^{(t,p)})^\top}{\sqrt{L_p^\beta}}\right]^\top$. Let $\widetilde{H}$ be

$$\widetilde{H} = \begin{bmatrix} h_1^\top \\ \vdots \\ h_p^\top \end{bmatrix} = \begin{bmatrix} \sqrt{L_1^\beta} \cdot I_m & 0 & 0 & \ldots & 0 & 0 \\ \frac{H_{21}(\boldsymbol{z}^{(t,2)})}{\sqrt{L_1^\beta}} & \sqrt{L_2^\beta} \cdot I_m & 0 & \ldots & 0 & 0 \\ \frac{H_{31}(\boldsymbol{z}^{(t,3)})}{\sqrt{L_1^\beta}} & \frac{H_{32}(\boldsymbol{z}^{(t,3)})}{\sqrt{L_2^\beta}} & \sqrt{L_3^\beta} \cdot I_m & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ldots & \ddots & \vdots \\ \frac{H_{p1}(\boldsymbol{z}^{(t,p)})}{\sqrt{L_1^\beta}} & \frac{H_{p2}(\boldsymbol{z}^{(t,p)})}{\sqrt{L_2^\beta}} & \frac{H_{p3}(\boldsymbol{z}^{(t,p)})}{\sqrt{L_3^\beta}} & \ldots & \frac{H_{p,p-1}(\boldsymbol{z}^{(t,p)})}{\sqrt{L_{p-1}^\beta}} & \sqrt{L_p^\beta} \cdot I_m \end{bmatrix}$$

Then we have

$$\begin{aligned}\|\nabla\mathcal{F}(x^{(t)})\|^2 &= \sum_{j=1}^p \|\nabla_j \mathcal{F}(x^{(t)})\|^2 = \sum_{j=1}^p \|h_j^\top f\|^2 = \|\widetilde{H}f\|^2 \\ &\leq \|\widetilde{H}\|^2\|f\|^2 = \|\widetilde{H}\|^2 \sum_{k=1}^p \frac{1}{2L_j^\beta}\|\nabla_j \mathcal{F}(y^{(t,j)})\|^2.\end{aligned} \tag{24}$$

Let $\widetilde{P}$ be defined as in the proof of Lemma 2. Then we have

$$\|\widetilde{H}\|^2 = \|\widetilde{P}^{1/2} + H\widetilde{P}^{-1/2}\|^2 \leq 2\left(\|\widetilde{P}^{1/2}\|^2 + \|H\widetilde{P}^{-1/2}\|^2\right) \leq 2\left(L_{\max}^\beta + \frac{\|H\|^2}{L_{\min}^\beta}\right), \tag{25}$$

Combining (23), (24) and (25), we have

$$\mathcal{F}(x^{(t)}) - \mathcal{F}(x^{(t+1)}) \geq \sum_{k=1}^p \frac{1}{2L_j^\beta}\|\nabla_j \mathcal{F}(y^{(t,j)})\|^2 \geq \frac{\|\nabla\mathcal{F}(x^{(t)})\|^2}{\|\widetilde{H}\|^2} \geq \frac{\|\nabla\mathcal{F}(x^{(t)})\|^2}{2\left(L_{\max}^\beta + \frac{\|H\|^2}{L_{\min}^\beta}\right)}.$$

■

We now characterize the gap towards the optimal objective after each CGD iteration.

Lemma 6 *Suppose that Assumptions 1 and 2 hold. Then, for all $t \geq 1$, we have*

$$\mathcal{F}(x^{(t)}) - \mathcal{F}(x^*) \leq \frac{\|\nabla\mathcal{F}(x^{(t)})\|^2}{2\mu}.$$

Proof From the convexity of $\mathcal{L}(\cdot)$ and strong convexity of $\mathcal{R}(\cdot)$, we have

$$\mathcal{F}(x^{(t)}) - \mathcal{F}(x^*) \leq (x^{(t)} - x^*)^\top \nabla\mathcal{F}(x^{(t)}) - \frac{\mu}{2}\|x^{(t)} - x^*\|^2.$$

Minimizing the right hand side over x^*, we have $x^* = x^{(t)} - \frac{\nabla\mathcal{F}(x^{(t)})}{\mu}$ and the desired result. ■

Combining the two lemmas above, we establish the iteration complexity bound of CGD.

Theorem 7 *Suppose that Assumption 1, 2 and 3 hold. We choose $\eta_j = L_j + \beta_j$. Then, given a pre-specified accuracy ϵ, we need at most*

$$\left\lceil \frac{L_{\max}^\beta + \frac{pL^\beta L_{\max}^\beta}{L_{\min}^\beta}}{\mu} \log\left(\frac{\mathcal{F}(x^{(0)}) - \mathcal{F}(x^*)}{\epsilon}\right) \right\rceil$$

iterations such that $\mathcal{F}(x^{(t)}) - \mathcal{F}(x^*) \leq \epsilon$, where $L^\beta = L + \beta$.

Proof We first bound $\|H\|$. Since $\mathcal{F}$ is convex, we have that $\nabla^2\mathcal{F}$ is positive semi-definite (PSD). This implies that there exists a matrix A such that $\nabla^2\mathcal{F} = AA^\top$, where A can be written as

$$A = \left[A_{1*}^\top, \ldots, A_{p*}^\top\right]^\top$$

and A_{i*} is the i-th row submatrix of A with $\|A_{i*}\| \leq \sqrt{L_i^\beta}$ and $\|A\| \leq \sqrt{L^\beta}$. Then we have

$$\begin{aligned}\|H\|^2 = \|H^\top H\| &\overset{\text{(i)}}{\leq} \sum_{i=1}^p \|(H^\top H)_{ii}\| = \sum_{i=1}^p \|H_{*i}\|^2 \leq \sum_{i=1}^p \|\nabla^2_{*i}\mathcal{F}\|^2 = \sum_{i=1}^p \|A_{i*}A\|^2 \\ &\leq \sum_{j=1}^p \|A_{*i}\|^2\|A\|^2 \leq pL^\beta L_i^\beta \leq pL^\beta L_{\max}^\beta, \end{aligned} \tag{26}$$

where (i) is from the norm compression inequality for block partitioned PSD matrix Horn and Johnson (2012) (Section 3.5). Thus we only need to combine Lemmas 5 and 6, and complete the proof by following similar lines to the proof of Theorem 3. ■

As can be seen from Theorem 7, the established iteration complexity bound is sharper than that in Beck and Tetruashvili (2013) by a factor of $L^\beta/L_{\min}^\beta$, which is at least of order $\sqrt{p}$ in generic settings and can be $\gg \sqrt{p}$ for ill-conditioned problems.

3.3 General Nonsmooth Minimization

We provide an iteration complexity bound of the CBCM and CBCGD methods for a general $\mathcal{L}(\cdot)$ and a nonsmooth $\mathcal{R}(\cdot)$.

Theorem 8 *Suppose that Assumptions 1 and 2 hold. We choose $\eta_j = L_j$ for the CBCGD method. Then given a pre-specified accuracy ϵ of the objective value, we need at most*

$$\left\lceil \frac{\mu L_{\min}^\mu + 4pL\cdot L_{\max}}{\mu L_{\min}^\mu} \log\left(\frac{\mathcal{F}(x^{(0)}) - \mathcal{F}(x^*)}{\epsilon}\right)\right\rceil$$

iterations for the CBCGD method and at most

$$\left\lceil \frac{\mu\mu_{\min} + pL\cdot L_{\max}}{\mu\mu_{\min}} \log\left(\frac{\mathcal{F}(x^{(0)}) - \mathcal{F}(x^*)}{\epsilon}\right)\right\rceil$$

iterations for the CBCM method to guarantee $\mathcal{F}(x^{(t)}) - \mathcal{F}(x^*) \leq \epsilon$.

Proof The three major steps are as follows.

Successive Descent: For CBCGD, using the same analysis of Lemma 1, we have that for all $t \geq 1$,

$$\mathcal{F}(x^{(t)}) - \mathcal{F}(x^{(t+1)}) \geq \frac{L_{\min}^\mu}{2}\|x^{(t)} - x^{(t+1)}\|^2.$$

For CBCM, using the same analysis of Theorem 4, we have that for all $t \geq 1$,

$$\mathcal{F}(x^{(t)}) - \mathcal{F}(x^{(t+1)}) \geq \frac{\mu_{\min}}{2}\|x^{(t)} - x^{(t+1)}\|^2.$$

Gap towards the Optimal Objective Value: By the strong convexity of $\mathcal{R}(\cdot)$, we have

$$\mathcal{F}(x) - \mathcal{F}(x^{(t+1)}) \geq \frac{\mu}{2}\|x - x^{(t+1)}\|^2 + (x - x^{(t+1)})^\top(\nabla\mathcal{L}(x^{(t+1)}) + \xi^{(t+1)}), \tag{27}$$

where $\xi_j^{(t+1)} \in \partial\mathcal{R}_j(x_j^{(t+1)})$. We then minimize both sides of (27) with respect to x and obtain

$$\mathcal{F}(x^{(t+1)}) - \mathcal{F}(x^*) \leq \frac{\|\nabla\mathcal{L}(x^{(t+1)}) + \xi^{(t+1)}\|^2}{2\mu}, \tag{28}$$

For CBCGD, we have

$$\begin{aligned}&\|\nabla\mathcal{L}(x^{(t+1)}) + \xi^{(t+1)}\|^2 \overset{\text{(i)}}{\leq} \sum_{j=1}^p \|\nabla_j\mathcal{L}(x^{(t+1)}) - \nabla_j\mathcal{L}(y^{(t+1,j+1)}) - L_j(x_j^{(t+1)} - x_j^{(t)})\|^2 \\ &\leq \sum_{j=1}^p 2\|\nabla_j\mathcal{L}(x^{(t+1)}) - \nabla_j\mathcal{L}(y^{(t+1,j+1)})\|^2 + 2L_j^2\|x_j^{(t+1)} - y_j^{(t+1,j)}\|^2 \\ &\overset{\text{(ii)}}{\leq} \sum_{j=1}^p 2\|\nabla(\nabla_j\mathcal{L}(z))\|\cdot\|(x^{(t+1)} - y^{(t+1,j+1)})\|^2 + 2L_j^2\|x_j^{(t+1)} - y_j^{(t+1,j)}\|^2 \\ &\overset{\text{(iii)}}{\leq} 4pL\cdot L_{\max}\|x^{(t+1)} - x^{(t)}\|^2, \end{aligned} \tag{29}$$

where (i) comes from the optimality condition

$$\nabla_j\mathcal{L}(y^{(t+1,j+1)}) + L_j(x_j^{(t+1)} - x_j^{(t)}) + \xi_j^{(t+1)} = 0,$$

(ii) is from the mean-value theorem and Cauchy-Schwarz inequality, and (iii) is from the same argument as in the proof of Theorem 7. Combining (28) and (29), we have

$$\mathcal{F}(x^{(t+1)}) - \mathcal{F}(x^*) \leq \frac{2pL\cdot L_{\max}\|x^{(t+1)} - x^{(t)}\|^2}{\mu}.$$

For CBCM, we have

$$\begin{aligned}\|\nabla\mathcal{L}(x^{(t+1)}) + \xi^{(t+1)}\|^2 &\overset{\text{(i)}}{\leq} \sum_{j=1}^p \|\nabla_j\mathcal{L}(x^{(t+1)}) - \nabla_j\mathcal{L}(y^{(t+1,j+1)})\|^2 \\ &\overset{\text{(ii)}}{\leq} pL\cdot L_{\max}\|x^{(t+1)} - x^{(t)}\|^2, \end{aligned} \tag{30}$$

where (i) comes from the optimality condition

$$\nabla_j\mathcal{L}(y^{(t+1,j+1)}) + \xi_j^{(t+1)} = 0$$

and (ii) is from the same argument as (29). Combining (28) and (30), we have

$$\mathcal{F}(x^{(t+1)}) - \mathcal{F}(x^*) \leq \frac{pL \cdot L_{\max}\|x^{(t+1)} - x^{(t)}\|^2}{2\mu}.$$

Iteration Complexity: The analysis follows from the counter part of Theorem 3. ■

Theorem 8 is a general result for the minimization of smooth loss function plus a non-smooth penalty function. In contrast, Beck and Tetruashvili (2013) only cover general smooth minimization. We also remark that the results of Theorem 7 and Theorem 8 are no better than their quadratic counterparts in Theorem 3 and Theorem 4 as $L \leq pL_{\max}$ in general.

3.4 Extensions to Nonstrongly Convex Minimization

For nonstrongly convex minimization, we only need to add a strongly convex perturbation to the objective function

$$\widehat{x} = \operatorname{argmin} \mathcal{F}(x) + \frac{\sigma}{2}\|x\|^2, \tag{31}$$

where $\sigma > 0$ is a perturbation parameter. Then, the results above can be used to analyze the CBCD-type methods for minimizing (31). Eventually, by setting σ as a reasonable small value, we can establish $\mathcal{O}(1/\epsilon)$-type iteration complexity bounds up to a $\log(1/\epsilon)$ factor. See Shalev-Shwartz and Zhang (2014) for more details.

4. Numerical Results

We consider two typical statistical machine learning problems as examples to illustrate our analysis.

(I) Elastic-net Penalized Linear Regression: Let $A \in \mathbb{R}^{n\times d}$ be the design matrix, and $b \in \mathbb{R}^n$ be the response vector. We solve the following optimization problem

$$\min_{x\in\mathbb{R}^d} \quad \frac{1}{2n}\|b - Ax\|^2 + \lambda_1\|x\|^2 + \lambda_2\|x\|_1,$$

where λ is the regularization parameter. We set n = 10,000 and d = 20,000. We simply treat each coordinate as a block (i.e., $d_{\max} = 1$). Each row of A is independently sampled from a 20,000-dimensional Gaussian distribution with mean 0 and covariance matrix Σ. We randomly select 2,000 entries of x, each of which is independently sampled from a uniform distribution over support $(-2, +2)$. The response vector b is generated by the linear model $b = Ax + \epsilon$, where ϵ is sampled from an n-variate Gaussian distribution $N(0, I_n)$. We set $\lambda_1 = \sqrt{\log 1/n}$ and $\lambda_2 = \sqrt{\log d/n} \approx 0.0315$. We normalize A to have $\|A_{*j}\| = \sqrt{n}$ for $j = 1, .., d$, where A_{*j} denotes the j-th column of A. For the BCGD method, we choose $\eta_j = 1$. For the gradient descent method, we either choose $\eta = \lambda_{\max}\left(\frac{1}{n}A^\top A\right)$, or adaptively select η by backtracking line search.

(II) Ridge Penalized Logistic Regression: We solve the following optimization problem

$$\min_{x\in\mathbb{R}^d} \quad \frac{1}{n}\sum_{i=1}^{n}\left[\log(1+\exp(x^\top A_{i*})) - b_i x^\top A_{i*}\right] + \lambda\|x\|^2.$$

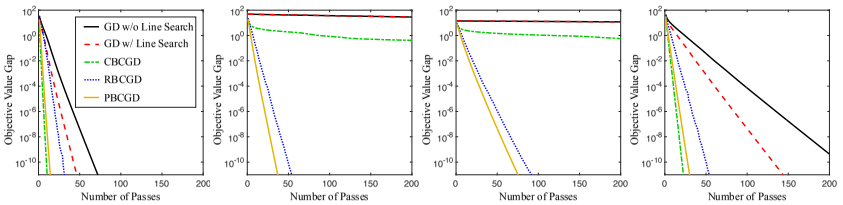
We generate the design matrix A and regression coefficient vector x using the same scheme as sparse linear regression. Again we treat each coordinate as a block (i.e., $d_{\max} = 1$). The response $b = [b_1, ..., b_n]^\top$ is generated by the logistic model $b_i = \text{Bernoulli}([1 + \exp(-x^\top A_{*i})]^{-1})$. We set $\lambda = \sqrt{1/n}$. For the BCGD method, we choose $\eta_j = \frac{1}{4}$. For gradient descent methods, we choose either the step size $\eta = \frac{1}{4}\lambda_{\max}\left(\frac{1}{n}A^\top A\right)$ or adaptively select η by backtracking line search.

We evaluate the computational performance using the number of passes over p blocks of coordinates (normalized iteration complexity). For the CBCGD method, we count one iteration as one pass (all p blocks). For the randomized BCGD (RBCGD) method, we count p iterations as one pass (since it only updates one block in each iteration). Besides the CBCGD and RBCGD methods, we also consider a variant of the CBCGD method named the permuted BCGD (PBCGD) method, which randomly permutes all indices for the p blocks in each iteration. Since the RBCGD and PBCGD methods are inherently stochastic, we report the objective values averaged over 20 different runs. Moreover, for the RBCGD method, the block of coordinates is selected uniformly at random in each iteration. We consider four different settings for both elastic-net penalized linear regression and ridge penalized logistic regression based on different choices of the covariance matrix Σ for generating the design matrix. We always choose $\Sigma_{jj} = 1$ for $j = 1, \ldots, d$, and for any $k \neq j$, we set (I) $\Sigma_{jk} = 0$; (II) $\Sigma_{jk} = 0.5$; (III) $\Sigma_{jk} = 0.75$; (IV) $\Sigma_{jk} = 0.5^{|j-k|}$. Note that the condition number of the Hessian matrix depends on Σ. Setting (I) and (IV) tend to yield well-conditioned Hessian matrices whereas Settings (II) and (III) tend to yield a badly conditioned Hessian matrix.

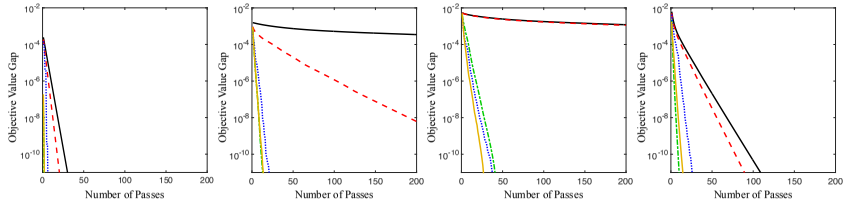
Figure 1 plots the gap between the objective value and the optimal as a function of number of passes for different methods. Our empirical findings can be summarized as follows: (1) All BCD-type methods attain better performance than the GD methods; (2) When the Hessian matrix is ill conditioned, i.e., in Setting (II) and (III), the CBCGD performs worse than the RBCGD and PBCGD methods, which suggests that there is a gap between cyclic and randomized BCGD. (3) When the Hessian matrix is well conditioned (e.g., in Settings (I) and (IV)), all three BCD-type methods attain good performance, and the CBCGD method slightly outperforms the PBCGD method; (4) The CBCGD method outperforms the RBCGD method in Setting (IV).

5. Discussions

Existing literature has established an iteration complexity of $\mathcal{O}(L\cdot\log(1/\epsilon)/\mu)$ for the gradient descent methods when solving strongly convex composite problems. However, our analysis shows that the CBCD-type methods only attains an iteration complexity of $\mathcal{O}(pLL_{\max}\cdot\log(1/\epsilon)/(L_{\min}\mu))$. Even though our analysis further shows that the iteration complexity of the CBCD-type methods can be further improved to $\mathcal{O}(\log^2(p)L^2\cdot\log(1/\epsilon)/(L_{\min}\mu))$ for a quadratic $\mathcal{L}(\cdot)$, there still exists a gap of factor $L\log^2 p/L_{\min}$. As our numerical experiments



(a) Enet (I) (b) Enet (II) (c) Enet (III) (d) Enet (IV)



(e) Logit (I) (f) Logit (II) (g) Logit (III) (h) Logit (IV)

Figure 1: Comparison among different methods under different settings. "RBCGD" and "PBCGD" denote the randomized BCD-type and permuted BCD-type methods respectively. The vertical axis corresponds to the gap towards the optimal objective value, $\log[\mathcal{F}(x) - \mathcal{F}(x^*)]$; the horizontal axis corresponds to the number of passes over p blocks of coordinates. Though all methods attain linear iteration complexity, their empirical behaviors are different from each others. Note that in plot (b) the curves for the CBCGD method and the RBCGD methods overlap.

show, however, the CBCD-type methods can actually attain a better computational performance than the gradient methods regardless of whether $\mathcal{L}(\cdot)$ is quadratic or not, thereby suggesting that perhaps there is still room for improvement in the iteration complexity analysis of the CBCD-type methods.

It is also worth mentioning that though some literature claims that the CBCD-type methods works as well as the randomized BCD-type methods in practice, there do exist some counter examples, e.g. our experiment in Setting (I), where the CBCD-type methods fail significantly. This suggests that the CBCD-type methods do have some possible disadvantages in practice. To the best of our knowledge, we are not aware of any similar experimental results reported in existing literature.

Furthermore, our numerical results show that the permuted BCD-type methods, which can be viewed as a hybrid of the cyclic and the randomized BCD-type (RBCD-type) methods, has a stable performance irrespective of the problem being well conditioned or not. But to the best of our knowledge, no iteration complexity result has been established for the permuted BCD-type (PBCD-type) methods. We leave these problems for future investigation.

Acknowledgments

This research is supported by NSF DMS1454377-CAREER; NSF IIS 1546482-BIGDATA; NIH R01MH102339; NSF IIS1408910; NSF IIS1332109; NIH R01GM083084; NSF CMMI1727757.

Appendix A. The Tightness of the Iteration Complexity for Quadratic Problems

We next provide an example to establish the tightness of the above result. We consider the following optimization problem

$$\min_x \mathcal{H}(x) := \|Bx\|^2, \tag{32}$$

where $B \in \mathbb{R}^{p\times p}$ is a tridiagonal Toplitz matrix defined as follows:

$$B = \begin{bmatrix} 3 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & 3 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 3 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & 3 & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & 3 \end{bmatrix}.$$

Note that the minimizer to (32) is $x^* = [0, 0, \dots, 0]^\top$, and the eigenvalues of B are given by $3 + 2\cos(j\pi/(j+1))$ for $j = 1, \dots, p$. Since the Hessian matrix of (32) is $2B^\top B$, we have

$$L = \lambda_{\max}(2B^\top B) \le 50, \mu = \lambda_{\min}(2B^\top B) \ge 2, \mu_{\min} = 10.$$

Clearly, for this problem the largest and smallest eigenvalues of the Hessian matrix, as well as L/γ do not scale with p. We consider each coordinate $x_j \in \mathbb{R}$ as a block. Then the problem can be rewritten as $\min \|\sum_{j=1}^p B_{*j}x_j\|$, where B_{*j} denotes the j-th column of B. Given an initial solution $x^{(0)}$, we can show that $x^{(1)}$ is generated by

$$\begin{cases} x_1^{(1)} &= -\frac{1}{4}\left(4x_2^{(0)} + x_3^{(0)}\right), \\ x_2^{(1)} &= -\frac{1}{5}\left(4x_1^{(1)} + 4x_3^{(0)} + x_4^{(0)}\right) \\ x_3^{(1)} &= -\frac{1}{5}\left(x_1^{(1)} + 4x_2^{(1)} + x_4^{(0)} + x_5^{(0)}\right), \\ x_j^{(1)} &= -\frac{1}{5}\left(x_{j-2}^{(1)} + 4x_{j-1}^{(1)} + x_{j+1}^{(0)} + x_{j+2}^{(0)}\right), \\ x_{p-1}^{(1)} &= -\frac{1}{5}\left(x_{p-3}^{(1)} + 4x_{p-2}^{(0)} + 4x_p^{(0)}\right), \\ x_p^{(1)} &= -\frac{1}{4}\left(x_{p-2}^{(1)} + 4x_{p-1}^{(1)}\right). \end{cases} \tag{33}$$

Now we choose the initial solution

$$x^{(0)} = \left[1, \frac{9}{32}, \frac{7}{8}, 1, \dots, 1, 1\right]^\top.$$

Then by (33), we obtain

$$x^{(1)} = \left[-\frac{1}{2}, -\frac{1}{2}, \ldots, -\frac{1}{2}, -\frac{3}{10}, -\frac{17}{40}\right]^\top,$$

which yields

$$\mathcal{H}(x^{(1)}) - \mathcal{H}(x^*) \geq \frac{25}{4}(p-3),$$
$$\|x^{(0)} - x^*\|^2 \leq p - 2 + \left(\frac{9}{32}\right)^2 + \left(\frac{7}{8}\right)^2 \leq p - 1.$$

Therefore, we have

$$\frac{\mathcal{H}(x^{(1)}) - \mathcal{H}(x^*)}{\|x^{(0)} - x^*\|^2} \geq \frac{25(p-3)}{4p} \geq \frac{22}{4}.$$

This implies that when the largest and smallest eigenvalues of the Hessian matrix do not scale with p (the number of blocks), the iteration complexity is independent of p, and cannot be further improved. Though the independence of p is only shown for the first iteration, we have similar claims in the subsequent iterations. We omit the detailed derivation due the heavy algebraic calculation.

References

James R Angelos, Carl C Cowen, and Sivaram K Narayan. Triangular truncation and finding the norm of a Hadamard multiplier. *Linear Algebra and its Applications*, 170: 117–135, 1992.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009a.

Amir Beck and Marc Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *Image Processing, IEEE Transactions on*, 18(11):2419–2434, 2009b.

Amir Beck and Luba Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM Journal on Optimization*, 23(4):2037–2060, 2013.

Stephen Becker, Jérôme Bobin, and Emmanuel J Candès. Nesta: A fast and accurate first-order method for sparse recovery. *SIAM Journal on Imaging Sciences*, 4(1):1–39, 2011.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

Jerome Friedman, Trevor Hastie, Holger Höfling, and Robert Tibshirani. Pathwise coordinate optimization. *Annals of Applied Statistics*, 1(2):302–332, 2007.

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–13, 2010.

Daniel Gabay and Bertrand Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.

Gene H Golub and Charles F Van Loan. *Matrix Computations*, volume 3. JHU Press, 2012.

Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. *Elements of Statistical Learning*. Springer, 2009.

Bingsheng He and Xiaoming Yuan. On non-ergodic convergence rate of douglas–rachford alternating direction method of multipliers. *Numerische Mathematik*, 130(3):567–577, 2015.

Mingyi Hong and Zhi-Quan Luo. On the linear convergence of the alternating direction method of multipliers. *Mathematical Programming*, pages 1–35, 2012.

Mingyi Hong, Xiangfeng Wang, Meisam Razaviyayn, and Zhi-Quan Luo. Iteration complexity analysis of block coordinate descent methods. *Mathematical Programming*, 163 (1-2):85–114, 2017.

Roger A Horn and Charles R Johnson. *Matrix Analysis*. Cambridge university press, 2012.

Ching-Pei Lee and Stephen J Wright. Random permutations fix a worst case for cyclic coordinate descent. *arXiv preprint arXiv:1607.08320*, 2016.

Xingguo Li, Tuo Zhao, Xiaoming Yuan, and Han Liu. The flare package for high dimensional linear regression and precision matrix estimation in R. *Journal of Machine Learning Research*, 16(1):553–557, 2015a.

Xingguo Li, Tuo Zhao, Tong Zhang, and Han Liu. The picasso package for nonconvex regularized m-estimation in high dimensions in R. *Technical Report*, 2015b.

Xingguo Li, Tuo Zhao, Raman Arora, Han Liu, and Mingyi Hong. An improved convergence analysis of cyclic block coordinate descent-type methods for strongly convex minimization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 491–499, 2016.

Han Liu, Mark Palatucci, and Jian Zhang. Blockwise coordinate descent procedures for the multi-task lasso, with applications to neural semantic basis discovery. In *International Conference on Machine Learning (ICML)*, pages 649–656, 2009.

Han Liu, Lie Wang, and Tuo Zhao. Sparse covariance matrix estimation with eigenvalue constraints. *Journal of Computational and Graphical Statistics*, 23(2):439–459, 2014.

Han Liu, Lie Wang, and Tuo Zhao. Calibrated multivariate regression with application to neural semantic basis discovery. *Journal of Machine Learning Research*, 16:1579–1606, 2015.

Zhaosong Lu and Lin Xiao. On the complexity analysis of randomized block-coordinate descent methods. *Mathematical Programming*, 152(1-2):615–642, 2015.

Zhi-Quan Luo and Paul Tseng. On the linear convergence of descent methods for convex essentially smooth minimization. *SIAM Journal on Control and Optimization*, 30(2): 408–425, 1992.

Rahul Mazumder, Jerome H Friedman, and Trevor Hastie. Sparsenet: Coordinate descent with nonconvex penalties. *Journal of the American Statistical Association*, 106:1125–1138, 2011.

Yu Nesterov. Gradient methods for minimizing composite objective function. Technical report, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2007.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer, 2004.

Julie Nutini, Mark Schmidt, Issam Laradji, Michael Friedlander, and Hoyt Koepke. Coordinate descent converges faster with the gauss-southwell rule than random selection. In *International Conference on Machine Learning (ICML)*, pages 1632–1641, 2015.

Meisam Razaviyayn, Mingyi Hong, and Zhi-Quan Luo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153, 2013.

Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, pages 1–38, 2012.

Ankan Saha and Ambuj Tewari. On the nonasymptotic convergence of cyclic coordinate descent methods. *SIAM Journal on Optimization*, 23(1):576–601, 2013.

Shai Shalev-Shwartz and Ambuj Tewari. Stochastic methods for ℓ_1-regularized loss minimization. *Journal of Machine Learning Research*, 12:1865–1892, 2011.

Shai Shalev-Shwartz and Tong Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Mathematical Programming*, pages 1–41, 2014.

Ruoyu Sun and Mingyi Hong. Improved iteration complexity bounds of cyclic block coordinate descent for convex problems. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1306–1314, 2015.

Ruoyu Sun and Yinyu Ye. Worst-case complexity of cyclic coordinate descent: $O(n^2)$ gap with randomized version. *arXiv preprint arXiv:1604.07130*, 2016.

Robert Tibshirani, Jacob Bien, Jerome Friedman, Trevor Hastie, Noah Simon, Jonathan Taylor, and Ryan J Tibshirani. Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(2): 245–266, 2012.

Paul Tseng. Dual coordinate ascent methods for non-strictly convex minimization. *Mathematical Programming*, 59(1-3):231–247, 1993.

Paul Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109(3):475–494, 2001.

Paul Tseng and Sangwoon Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117(1-2):387–423, 2009.

Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical Learning Theory*, volume 1. Wiley New York, 1998.

Sangwoon Yun. On the iteration complexity of cyclic coordinate gradient descent methods. *SIAM Journal on Optimization*, 24(3):1567–1580, 2014.

Tuo Zhao and Han Liu. Sparse additive machine. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1435–1443, 2012.

Tuo Zhao and Han Liu. Accelerated path-following iterative shrinkage thresholding algorithm with application to semiparametric graph estimation. *Journal of Computational and Graphical Statistics*, 2015.

Tuo Zhao, Han Liu, Kathryn Roeder, John Lafferty, and Larry Wasserman. The huge package for high-dimensional undirected graph estimation in R. *Journal of Machine Learning Research*, 13:1059–1062, 2012.

Tuo Zhao, Han Liu, and Tong Zhang. A general theory of pathwise coordinate optimization. *arXiv preprint arXiv:1412.7477*, 2014a.

Tuo Zhao, Mo Yu, Yiming Wang, Raman Arora, and Han Liu. Accelerated mini-batch randomized block coordinate descent method. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3329–3337, 2014b.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

Journal of Machine Learning Research 18 (2018) 1-52 Submitted 11/16; Revised 12/17; Published 4/18

Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization

Lisha Li LISHAL@CS.CMU.EDU
Carnegie Mellon University, Pittsburgh, PA 15213
Kevin Jamieson JAMIESON@CS.WASHINGTON.EDU
University of Washington, Seattle, WA 98195
Giulia DeSalvo GIULIAD@GOOGLE.COM
Google Research, New York, NY 10011
Afshin Rostamizadeh ROSTAMI@GOOGLE.COM
Google Research, New York, NY 10011
Ameet Talwalkar TALWALKAR@CMU.EDU
Carnegie Mellon University, Pittsburgh, PA 15213
Determined AI

Editor: Nando de Freitas

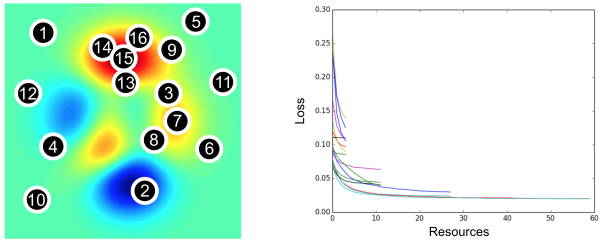
Abstract

Performance of machine learning algorithms depends critically on identifying a good set of hyperparameters. While recent approaches use Bayesian optimization to adaptively select configurations, we focus on speeding up random search through adaptive resource allocation and early-stopping. We formulate hyperparameter optimization as a pure-exploration non-stochastic infinite-armed bandit problem where a predefined resource like iterations, data samples, or features is allocated to randomly sampled configurations. We introduce a novel algorithm, HYPERBAND, for this framework and analyze its theoretical properties, providing several desirable guarantees. Furthermore, we compare HYPERBAND with popular Bayesian optimization methods on a suite of hyperparameter optimization problems. We observe that HYPERBAND can provide over an order-of-magnitude speedup over our competitor set on a variety of deep-learning and kernel-based learning problems.

Keywords: hyperparameter optimization, model selection, infinite-armed bandits, online optimization, deep learning

1. Introduction

In recent years, machine learning models have exploded in complexity and expressibility at the price of staggering computational costs. Moreover, the growing number of tuning parameters associated with these models are difficult to set by standard optimization techniques. These "hyperparameters" are inputs to a machine learning algorithm that govern how the algorithm's performance generalizes to new, unseen data; examples of hyperparameters include those that impact model architecture, amount of regularization, and learning rates. The quality of a predictive model critically depends on its hyperparameter configuration, but it is poorly understood how these hyperparameters interact with each other to affect the resulting model.



(a) Configuration Selection (b) Configuration Evaluation

Figure 1: (a) The heatmap shows the validation error over a two-dimensional search space with red corresponding to areas with lower validation error. Configuration selection methods adaptively choose new configurations to train, proceeding in a sequential manner as indicated by the numbers. (b) The plot shows the validation error as a function of the resources allocated to each configuration (i.e. each line in the plot). Configuration evaluation methods allocate more resources to promising configurations.

Consequently, practitioners often default to brute-force methods like random search and grid search (Bergstra and Bengio, 2012).

In an effort to develop more efficient search methods, the problem of hyperparameter optimization has recently been dominated by *Bayesian optimization* methods (Snoek et al., 2012; Hutter et al., 2011; Bergstra et al., 2011) that focus on optimizing hyperparameter *configuration selection*. These methods aim to identify good configurations more quickly than standard baselines like random search by selecting configurations in an adaptive manner; see Figure 1(a). Existing empirical evidence suggests that these methods outperform random search (Thornton et al., 2013; Eggensperger et al., 2013; Snoek et al., 2015b). However, these methods tackle the fundamentally challenging problem of simultaneously fitting and optimizing a high-dimensional, non-convex function with unknown smoothness, and possibly noisy evaluations.

An orthogonal approach to hyperparameter optimization focuses on speeding up *configuration evaluation*; see Figure 1(b). These approaches are adaptive in computation, allocating more resources to promising hyperparameter configurations while quickly eliminating poor ones. Resources can take various forms, including size of training set, number of features, or number of iterations for iterative algorithms. By adaptively allocating resources, these approaches aim to examine orders-of-magnitude more hyperparameter configurations than approaches that uniformly train all configurations to completion, thereby quickly identifying good hyperparameters. While there are methods that combine Bayesian optimization with adaptive resource allocation (Swersky et al., 2013, 2014; Domhan et al., 2015; Klein et al.,

2017a), we focus on speeding up random search as it offers a simple and theoretically principled launching point (Bergstra and Bengio, 2012).[1]

We develop a novel configuration evaluation approach by formulating hyperparameter optimization as a pure-exploration adaptive resource allocation problem addressing how to allocate resources among randomly sampled hyperparameter configurations.[2] Our procedure, Hyperband, relies on a principled early-stopping strategy to allocate resources, allowing it to evaluate orders-of-magnitude more configurations than black-box procedures like Bayesian optimization methods. Hyperband is a general-purpose technique that makes minimal assumptions unlike prior configuration evaluation approaches (Domhan et al., 2015; Swersky et al., 2014; György and Kocsis, 2011; Agarwal et al., 2011; Sparks et al., 2015; Jamieson and Talwalkar, 2015).

Our theoretical analysis demonstrates the ability of Hyperband to adapt to unknown convergence rates and to the behavior of validation losses as a function of the hyperparameters. In addition, Hyperband is 5× to 30× faster than popular Bayesian optimization algorithms on a variety of deep-learning and kernel-based learning problems. A theoretical contribution of this work is the introduction of the pure-exploration, infinite-armed bandit problem in the non-stochastic setting, for which Hyperband is one solution. When Hyperband is applied to the special-case stochastic setting, we show that the algorithm comes within log factors of known lower bounds in both the infinite (Carpentier and Valko, 2015) and finite K-armed bandit settings (Kaufmann et al., 2015).

The paper is organized as follows. Section 2 summarizes related work in two areas: (1) hyperparameter optimization, and (2) pure-exploration bandit problems. Section 3 describes Hyperband and provides intuition for the algorithm through a detailed example. In Section 4, we present a wide range of empirical results comparing Hyperband with state-of-the-art competitors. Section 5 frames the hyperparameter optimization problem as an infinite-armed bandit problem and summarizes the theoretical results for Hyperband. Finally, Section 6 discusses possible extensions of Hyperband.

2. Related Work

In Section 1, we briefly discussed related work in the hyperparameter optimization literature. Here, we provide a more thorough coverage of the prior work, and also summarize significant related work on bandit problems.

2.1 Hyperparameter Optimization

Bayesian optimization techniques model the conditional probability $p(y|\lambda)$ of a configuration's performance on an evaluation metric y (i.e., test accuracy), given a set of hyperparameters λ.

1. Random search will asymptotically converge to the optimal configuration, regardless of the smoothness or structure of the function being optimized, by a simple covering argument. While the rate of convergence for random search depends on the smoothness and is exponential in the number of dimensions in the search space, the same is true for Bayesian optimization methods without additional structural assumptions (Kandasamy et al., 2015).
2. A preliminary version of this work appeared in Li et al. (2017). We extend the previous paper with a thorough theoretical analysis of Hyperband; an infinite horizon version of the algorithm with application to stochastic infinite-armed bandits; additional intuition and discussion of Hyperband to facilitate its use in practice; and additional results on a collection of 117 multistage model selection tasks.

Sequential Model-based Algorithm Configuration (SMAC), Tree-structure Parzen Estimator (TPE), and Spearmint are three well-established methods (Feurer et al., 2014). SMAC uses random forests to model $p(y|\lambda)$ as a Gaussian distribution (Hutter et al., 2011). TPE is a non-standard Bayesian optimization algorithm based on tree-structured Parzen density estimators (Bergstra et al., 2011). Lastly, Spearmint uses Gaussian processes (GP) to model $p(y|\lambda)$ and performs slice sampling over the GP's hyperparameters (Snoek et al., 2012).

Previous work compared the relative performance of these Bayesian searchers (Thornton et al., 2013; Eggensperger et al., 2013; Bergstra et al., 2011; Snoek et al., 2012; Feurer et al., 2014, 2015). An extensive survey of these three methods by Eggensperger et al. (2013) introduced a benchmark library for hyperparameter optimization called HPOlib, which we use for our experiments. Bergstra et al. (2011) and Thornton et al. (2013) showed Bayesian optimization methods empirically outperform random search on a few benchmark tasks. However, for high-dimensional problems, standard Bayesian optimization methods perform similarly to random search (Wang et al., 2013). Recent methods specifically designed for high-dimensional problems assume a lower effective dimension for the problem (Wang et al., 2013) or an additive decomposition for the target function (Kandasamy et al., 2015). However, as can be expected, the performance of these methods is sensitive to required inputs; i.e. the effective dimension (Wang et al., 2013) or the number of additive components (Kandasamy et al., 2015).

Gaussian processes have also been studied in the bandit setting using confidence bound acquisition functions (GP-UCB), with associated sublinear regret bounds (Srinivas et al., 2010; Grünewälder et al., 2010). Wang et al. (2016) improved upon GP-UCB by removing the need to tune a parameter that controls exploration and exploitation. Contal et al. (2014) derived a tighter regret bound than that for GP-UCB by using a mutual information acquisition function. However, van der Vaart and van Zanten (2011) showed that the learning rate of GPs are sensitive to the definition of the prior through an example with a poor prior where the learning rate degraded from polynomial to logarithmic in the number of observations n. Additionally, without structural assumptions on the covariance matrix of the GP, fitting the posterior is $O(n^3)$ (Wilson et al., 2015). Hence, Snoek et al. (2015a) and Springenberg et al. (2016) proposed using Bayesian neural networks, which scale linearly with n, to model the posterior.

Adaptive configuration evaluation is not a new idea. Maron and Moore (1997) and Mnih and Audibert (2008) considered a setting where the training time is relatively inexpensive (e.g., k-nearest-neighbor classification) and evaluation on a large validation set is accelerated by evaluating on an increasing subset of the validation set, stopping early configurations that are performing poorly. Since subsets of the validation set provide unbiased estimates of its expected performance, this is an instance of the *stochastic* best-arm identification problem for multi-armed bandits (see the work by Jamieson and Nowak, 2014, for a brief survey).

In contrast, we address a setting where the evaluation time is relatively inexpensive and the goal is to early-stop long-running training procedures by evaluating partially trained models on the full validation set. Previous approaches in this setting either require strong assumptions or use heuristics to perform adaptive resource allocation. György and Kocsis (2011) and Agarwal et al. (2011) made parametric assumptions on the convergence behavior of training algorithms, providing theoretical performance guarantees under these assumptions. Unfortunately, these assumptions are often hard to verify, and empirical performance can

drastically suffer when they are violated. Krueger et al. (2015) proposed a heuristic based on sequential analysis to determine stopping times for training configurations on increasing subsets of the data. However, the theoretical correctness and empirical performance of this method are highly dependent on a user-defined "safety zone."

Several hybrid methods combining adaptive configuration selection and evaluation have also been introduced (Swersky et al., 2013, 2014; Domhan et al., 2015; Kandasamy et al., 2016; Klein et al., 2017a; Golovin et al., 2017). The algorithm proposed by Swersky et al. (2013) uses a GP to learn correlation between related tasks and requires the subtasks as input, but efficient subtasks with high informativeness for the target task are unknown without prior knowledge. Similar to the work by Swersky et al. (2013), Klein et al. (2017a) modeled the conditional validation error as a Gaussian process using a kernel that captures the covariance with downsampling rate to allow for adaptive evaluation. Swersky et al. (2014), Domhan et al. (2015), and Klein et al. (2017a) made parametric assumptions on the convergence of learning curves to perform early-stopping. In contrast, Golovin et al. (2017) devised an early-stopping rule based on predicted performance from a nonparametric GP model with a kernel designed to measure the similarity between performance curves. Finally, Kandasamy et al. (2016) extended GP-UCB to allow for adaptive configuration evaluation by defining subtasks that monotonically improve with more resources.

In another line of work, Sparks et al. (2015) proposed a halving style bandit algorithm that did not require explicit convergence behavior, and Jamieson and Talwalkar (2015) analyzed a similar algorithm originally proposed by Karnin et al. (2013) for a different setting, providing theoretical guarantees and encouraging empirical results. Unfortunately, these halving style algorithms suffer from the "n versus B/n" problem, which we will discuss in Section 3.1. Hyperband addresses this issue and provides a robust, theoretically principled early-stopping algorithm for hyperparameter optimization.

We note that Hyperband can be combined with any hyperparameter sampling approach and does not depend on random sampling; the theoretical results only assume the validation losses of sampled hyperparameter configurations are drawn from some stationary distribution. In fact, subsequent to our submission, Klein et al. (2017b) combined adaptive configuration selection with Hyperband by using a Bayesian neural network to model learning curves and only selecting configurations with high predicted performance to input into Hyperband.

2.2 Bandit Problems

Pure exploration bandit problems aim to minimize the simple regret, defined as the distance from the optimal solution, as quickly as possible in any given setting. The pure-exploration multi-armed bandit problem has a long history in the stochastic setting (Even-Dar et al., 2006; Bubeck et al., 2009), and was recently extended to the non-stochastic setting by Jamieson and Talwalkar (2015). Relatedly, the stochastic pure-exploration infinite-armed bandit problem was studied by Carpentier and Valko (2015), where a pull of each arm i yields an i.i.d. sample in $[0, 1]$ with expectation ν_i, where ν_i is a loss drawn from a distribution with cumulative distribution function, F. Of course, the value of ν_i is unknown to the player, so the only way to infer its value is to pull arm i many times. Carpentier and Valko (2015) proposed an anytime algorithm, and derived a tight (up to polylog factors) upper bound on its error assuming what we will refer to as the β-parameterization of F described in

Section 5.3.2. However, their algorithm was derived specifically for the β-parameterization of F, and furthermore, they must estimate β before running the algorithm, limiting the algorithm's practical applicability. Also, the algorithm assumes stochastic losses from the arms and thus the convergence behavior is known; consequently, it does not apply in our hyperparameter optimization setting.[3] Two related lines of work that both make use of an underlying metric space are Gaussian process optimization (Srinivas et al., 2010) and X-armed bandits (Bubeck et al., 2011), or bandits defined over a metric space. However, these works either assume stochastic rewards or need to know something about the underlying function (e.g. an appropriate kernel or level of smoothness).

In contrast, Hyperband is devised for the non-stochastic setting and automatically adapts to unknown F without making any parametric assumptions. Hence, we believe our work to be a generally applicable pure exploration algorithm for infinite-armed bandits. To the best of our knowledge, this is also the first work to test out such an algorithm on a real application.

3. Hyperband Algorithm

In this section, we present the Hyperband algorithm. We provide intuition for the algorithm, highlight the main ideas via a simple example that uses iterations as the adaptively allocated resource, and present a few guidelines on how to deploy Hyperband in practice.

3.1 Successive Halving

Hyperband extends the SuccessiveHalving algorithm proposed for hyperparameter optimization by Jamieson and Talwalkar (2015) and calls it as a subroutine. The idea behind the original SuccessiveHalving algorithm follows directly from its name: uniformly allocate a budget to a set of hyperparameter configurations, evaluate the performance of all configurations, throw out the worst half, and repeat until one configuration remains. The algorithm allocates exponentially more resources to more promising configurations. Unfortunately, SuccessiveHalving requires the number of configurations n as an input to the algorithm. Given some finite budget B (e.g., an hour of training time to choose a hyperparameter configuration), B/n resources are allocated on average across the configurations. However, for a fixed B, it is not clear a priori whether we should (a) consider many configurations (large n) with a small average training time; or (b) consider a small number of configurations (small n) with longer average training times.

We use a simple example to better understand this tradeoff. Figure 2 shows the validation loss as a function of total resources allocated for two configurations with terminal validation losses ν_1 and ν_2. The shaded areas bound the maximum deviation of the intermediate losses from the terminal validation loss and will be referred to as "envelope" functions.[4] It is possible to distinguish between the two configurations when the envelopes no longer overlap. Simple arithmetic shows that this happens when the width of the envelopes is less than $\nu_2 - \nu_1$, i.e., when the intermediate losses are guaranteed to be less than $\frac{\nu_2 - \nu_1}{2}$ away from the

3. See the work by Jamieson and Talwalkar (2015) for detailed discussion motivating the non-stochastic setting for hyperparameter optimization.
4. These envelope functions are guaranteed to exist; see discussion in Section 5.2 where we formally define these envelope (or γ) functions.

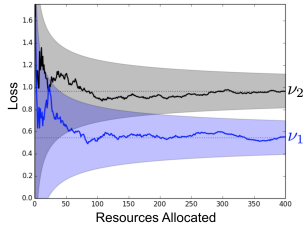


Figure 2: The validation loss as a function of total resources allocated for two configurations is shown. ν_1 and ν_2 represent the terminal validation losses at convergence. The shaded areas bound the maximum distance of the intermediate losses from the terminal validation loss and monotonically decrease with the resource.

terminal losses. There are two takeaways from this observation: more resources are needed to differentiate between the two configurations when either (1) the envelope functions are wider or (2) the terminal losses are closer together.

However, in practice, the optimal allocation strategy is unknown because we do not have knowledge of the envelope functions nor the distribution of terminal losses. Hence, if more resources are required before configurations can differentiate themselves in terms of quality (e.g., if an iterative training method converges very slowly for a given data set or if randomly selected hyperparameter configurations perform similarly well), then it would be reasonable to work with a small number of configurations. In contrast, if the quality of a configuration is typically revealed after a small number of resources (e.g., if iterative training methods converge very quickly for a given data set or if randomly selected hyperparameter configurations are of low-quality with high probability), then n is the bottleneck and we should choose n to be large.

Certainly, if meta-data or previous experience suggests that a certain tradeoff is likely to work well in practice, one should exploit that information and allocate the majority of resources to that tradeoff. However, without this supplementary information, practitioners are forced to make this tradeoff, severely hindering the applicability of existing configuration evaluation methods.

3.2 Hyperband

Hyperband, shown in Algorithm 1, addresses this "n versus B/n" problem by considering several possible values of n for a fixed B, in essence performing a grid search over feasible value of n. Associated with each value of n is a minimum resource r that is allocated to all configurations before some are discarded; a larger value of n corresponds to a smaller r and hence more aggressive early-stopping. There are two components to Hyperband; (1) the inner loop invokes SuccessiveHalving for fixed values of n and r (lines 3–9) and (2) the outer loop iterates over different values of n and r (lines 1–2). We will refer to each such run of SuccessiveHalving within Hyperband as a "bracket." Each bracket is designed to use approximately B total resources and corresponds to a different tradeoff between n

Algorithm 1: Hyperband algorithm for hyperparameter optimization.

input : R, η (default $\eta = 3$)
initialization : $s_{\max} = \lfloor \log_\eta(R) \rfloor$, $B = (s_{\max}+1)R$
1 **for** $s \in \{s_{\max}, s_{\max}-1, \ldots, 0\}$ **do**
2 $\quad n = \lceil \frac{B}{R}\frac{\eta^s}{(s+1)} \rceil$, $\quad r = R\eta^{-s}$
$\quad$ `// begin SuccessiveHalving with` (n, r) `inner loop`
3 $\quad T =$ `get_hyperparameter_configuration`(n)
4 $\quad$ **for** $i \in \{0, \ldots, s\}$ **do**
5 $\quad\quad n_i = \lfloor n\eta^{-i} \rfloor$
6 $\quad\quad r_i = r\eta^i$
7 $\quad\quad L = \{$`run_then_return_val_loss`$(t, r_i) : t \in T\}$
8 $\quad\quad T =$ `top_k`$(T, L, \lfloor n_i/\eta \rfloor)$
9 $\quad$ **end**
10 **end**
11 **return** *Configuration with the smallest intermediate loss seen so far.*

and B/n. Hence, a single execution of Hyperband takes a finite budget of $(s_{max}+1)B$; we recommend repeating it indefinitely.

Hyperband requires two inputs (1) R, the maximum amount of resource that can be allocated to a single configuration, and (2) η, an input that controls the proportion of configurations discarded in each round of SuccessiveHalving. The two inputs dictate how many different brackets are considered; specifically, $s_{\max}+1$ different values for n are considered with $s_{\max} = \lfloor \log_\eta(R) \rfloor$. Hyperband begins with the most aggressive bracket $s = s_{\max}$, which sets n to maximize exploration, subject to the constraint that at least one configuration is allocated R resources. Each subsequent bracket reduces n by a factor of approximately η until the final bracket, $s = 0$, in which every configuration is allocated R resources (this bracket simply performs classical random search). Hence, Hyperband performs a geometric search in the average budget per configuration and removes the need to select n for a fixed budget at the cost of approximately $s_{\max}+1$ times more work than running SuccessiveHalving for a single value of n. By doing so, Hyperband is able to exploit situations in which adaptive allocation works well, while protecting itself in situations where more conservative allocations are required.

Hyperband requires the following methods to be defined for any given learning problem:

- `get_hyperparameter_configuration`(n) – a function that returns a set of n i.i.d. samples from some distribution defined over the hyperparameter configuration space. In this work, we assume uniformly sampling of hyperparameters from a predefined space (i.e., hypercube with min and max bounds for each hyperparameter), which immediately yields consistency guarantees. However, the more aligned the distribution is towards high quality hyperparameters (i.e., a useful prior), the better Hyperband will perform (see Section 6 for further discussion).

	$s=4$		$s=3$		$s=2$		$s=1$		$s=0$	
i	n_i	r_i	n_i	r_i	n_i	r_i	n_i	r_i	n_i	r_i
0	81	1	27	3	9	9	6	27	5	81
1	27	3	9	9	3	27	2	81		
2	9	9	3	27	1	81				
3	3	27	1	81						
4	1	81								

Table 1: The values of n_i and r_i for the brackets of Hyperband corresponding to various values of s, when $R = 81$ and $\eta = 3$.

- `run_then_return_val_loss(t, r)` – a function that takes a hyperparameter configuration t and resource allocation r as input and returns the validation loss after training the configuration for the allocated resources.

- `top_k(configs, losses, k)` – a function that takes a set of configurations as well as their associated losses and returns the top k performing configurations.

3.3 Example Application with Iterations as a Resource: LeNet

We next present a concrete example to provide further intuition about Hyperband. We work with the MNIST data set and optimize hyperparameters for the LeNet convolutional neural network trained using mini-batch stochastic gradient descent (SGD).[5] Our search space includes learning rate, batch size, and number of kernels for the two layers of the network as hyperparameters (details are shown in Table 2 in Appendix A).

We define the resource allocated to each configuration to be number of iterations of SGD, with one unit of resource corresponding to one epoch, i.e., a full pass over the data set. We set R to 81 and use the default value of $\eta = 3$, resulting in $s_{\max} = 4$ and thus 5 brackets of SuccessiveHalving with different tradeoffs between n and B/n. The resources allocated within each bracket are displayed in Table 1.

Figure 3 shows an empirical comparison of the average test error across 70 trials of the individual brackets of Hyperband run separately as well as standard Hyperband. In practice, we do not know a priori which bracket $s \in \{0, \ldots, 4\}$ will be most effective in identifying good hyperparameters, and in this case neither the most ($s = 4$) nor least aggressive ($s = 0$) setting is optimal. However, we note that Hyperband does nearly as well as the optimal bracket ($s = 3$) and outperforms the baseline uniform allocation (i.e., random search), which is equivalent to bracket $s = 0$.

3.4 Different Types of Resources

While the previous example focused on iterations as the resource, Hyperband naturally generalizes to various types of resources:

5. Code and description of algorithm used is available at `http://deeplearning.net/tutorial/lenet.html`.

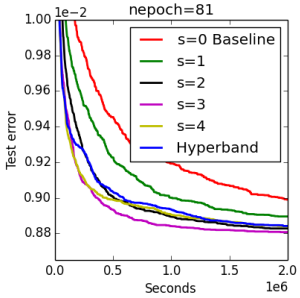


Figure 3: Performance of individual brackets s and Hyperband.

- **Time** – Early-stopping in terms of time can be preferred when various hyperparameter configurations differ in training time and the practitioner's chief goal is to find a good hyperparameter setting in a fixed wall-clock time. For instance, training time could be used as a resource to quickly terminate straggler jobs in distributed computation environments.

- **Data Set Subsampling** – Here we consider the setting of a black-box batch training algorithm that takes a data set as input and outputs a model. In this setting, we treat the resource as the size of a random subset of the data set with R corresponding to the full data set size. Subsampling data set sizes using Hyperband, especially for problems with super-linear training times like kernel methods, can provide substantial speedups.

- **Feature Subsampling** – Random features or Nyström-like methods are popular methods for approximating kernels for machine learning applications (Rahimi and Recht, 2007). In image processing, especially deep-learning applications, filters are usually sampled randomly, with the number of filters having an impact on the performance. Downsampling the number of features is a common tool used when hand-tuning hyperparameters; Hyperband can formalize this heuristic.

3.5 Setting R

The resource R and η (which we address next) are the only required inputs to Hyperband. As mentioned in Section 3.2, R represents the maximum amount of resources that can be allocated to any given configuration. In most cases, there is a natural upper bound on the maximum budget per configuration that is often dictated by the resource type (e.g., training set size for data set downsampling; limitations based on memory constraint for feature downsampling; rule of thumb regarding number of epochs when iteratively training neural networks). If there is a range of possible values for R, a smaller R will give a result faster (since the budget B for each bracket is a multiple of R), but a larger R will give a better guarantee of successfully differentiating between the configurations.

Moreover, for settings in which either R is unknown or not desired, we provide an infinite horizon version of Hyperband in Section 5. This version of the algorithm doubles

the budget over time, $B \in \{2, 4, 8, 16, \ldots\}$, and for each B, tries all possible values of $n \in \left\{2^k : k \in \{1, \ldots, \log_2(B)\}\right\}$. For each combination of B and n, the algorithm runs an instance of the (infinite horizon) SuccessiveHalving algorithm, which implicitly sets $R = \frac{B}{2\log_2(n)}$, thereby growing R as B increases. The main difference between the infinite horizon algorithm and Algorithm 1 is that the number of unique brackets grows over time instead of staying constant with each outer loop. We will analyze this version of Hyperband in more detail in Section 5 and use it as the launching point for the theoretical analysis of standard (finite horizon) Hyperband.

Note that R is also the number of configurations evaluated in the bracket that performs the most exploration, i.e $s = s_{\max}$. In practice one may want $n \leq n_{\max}$ to limit overhead associated with training many configurations on a small budget, i.e., costs associated with initialization, loading a model, and validation. In this case, set $s_{\max} = \lfloor \log_\eta(n_{\max}) \rfloor$. Alternatively, one can redefine one unit of resource so that R is artificially smaller (i.e., if the desired maximum iteration is 100k, defining one unit of resource to be 100 iterations will give $R = 1,000$, whereas defining one unit to be 1k iterations will give $R = 100$). Thus, one unit of resource can be interpreted as the minimum desired resource and R as the ratio between maximum resource and minimum resource.

3.6 Setting η

The value of η is a knob that can be tuned based on *practical* user constraints. Larger values of η correspond to more aggressive elimination schedules and thus fewer rounds of elimination; specifically, each round retains $1/\eta$ configurations for a total of $\lfloor \log_\eta(n) \rfloor + 1$ rounds of elimination with n configurations. If one wishes to receive a result faster at the cost of a sub-optimal asymptotic constant, one can increase η to reduce the budget per bracket $B = (\lfloor \log_\eta(R) \rfloor + 1)R$. We stress that results are not very sensitive to the choice of η. If our theoretical bounds are optimized (see Section 5), they suggest choosing $\eta = e \approx 2.718$, but in practice we suggest taking η to be equal to 3 or 4.

Tuning η will also change the number of brackets and consequently the number of different tradeoffs that Hyperband tries. Usually, the possible range of brackets is fairly constrained, since the number of brackets is logarithmic in R; namely, there are $(\lfloor \log_\eta(R) \rfloor + 1) = s_{\max} + 1$ brackets. For our experiments in Section 4, we chose η to provide 5 brackets for the specified R; for most problems, 5 is a reasonable number of n versus B/n tradeoffs to explore. However, for large R, using $\eta = 3$ or 4 can give more brackets than desired. The number of brackets can be controlled in a few ways. First, as mentioned in the previous section, if R is too large and overhead is an issue, then one may want to control the overhead by limiting the maximum number of configurations to $n_{\max}$, thereby also limiting $s_{\max}$. If overhead is not a concern and aggressive exploration is desired, one can (1) increase η to reduce the number of brackets while maintaining R as the maximum number of configurations in the most exploratory bracket, or (2) still use $\eta = 3$ or 4 but only try brackets that do a baseline level of exploration, i.e., set $n_{\min}$ and only try brackets from $s_{\max}$ to $s = \lfloor \log_\eta(n_{\min}) \rfloor$. For computationally intensive problems that have long training times and high-dimensional search spaces, we recommend the latter. Intuitively, if the number of configurations that can be trained to completion (i.e., trained using R resources) in a reasonable amount of time is on the order of the dimension of the search space and not exponential in the dimension, then

it will be impossible to find a good configuration without using an aggressive exploratory tradeoff between n and B/n.

3.7 Overview of Theoretical Results

The theoretical properties of Hyperband are best demonstrated through an example. Suppose there are n configurations, each with a given terminal validation error ν_i for $i = 1, \ldots, n$. Without loss of generality, index the configurations by performance so that ν_1 corresponds to the best performing configuration, ν_2 to the second best, and so on. Now consider the task of identifying the best configuration. The optimal strategy would allocate to each configuration i the minimum resource required to distinguish it from ν_1, i.e., enough so that the envelope functions (see Figure 2) bound the intermediate loss to be less than $\frac{\nu_i - \nu_1}{2}$ away from the terminal value. In contrast, the naive uniform allocation strategy, which allocates B/n to each configuration, has to allocate to every configuration the maximum resource required to distinguish any arm ν_i from ν_1. Remarkably, the budget required by SuccessiveHalving is only a small factor of the optimal because it capitalizes on configurations that are easy to distinguish from ν_1.

The relative size of the budget required for uniform allocation and SuccessiveHalving depends on the envelope functions bounding deviation from terminal losses as well as the distribution from which ν_i's are drawn. The budget required for SuccessiveHalving is smaller when the optimal n versus B/n tradeoff discussed in Section 3.1 requires fewer resources per configuration. Hence, if the envelope functions tighten quickly as a function of resource allocated, or the average distances between terminal losses is large, then SuccessiveHalving can be substantially faster than uniform allocation. These intuitions are formalized in Section 5 and associated theorems/corollaries are provided that take into account the envelope functions and the distribution from which ν_i's are drawn.

In practice, we do not have knowledge of either the envelope functions or the distribution of ν_i's, both of which are integral in characterizing SuccessiveHalving's required budget. With Hyperband we address this shortcoming by hedging our aggressiveness. We show in Section 5.3.3 that Hyperband, despite having no knowledge of the envelope functions nor the distribution of ν_i's, requires a budget that is only log factors larger than that of SuccessiveHalving.

4. Hyperparameter Optimization Experiments

In this section, we evaluate the empirical behavior of Hyperband with three different resource types: iterations, data set subsamples, and feature samples. For all experiments, we compare Hyperband with three well known Bayesian optimization algorithms—SMAC, TPE, and Spearmint—using their default settings. We exclude Spearmint from the comparison set when there are conditional hyperparameters in the search space because it does not natively support them (Eggensperger et al., 2013). We also show results for SuccessiveHalving corresponding to repeating the most exploratory bracket of Hyperband to provide a baseline for aggressive early-stopping.[6] Additionally, as standard baselines against which to measure

6. This is not done for the experiments in Section 4.2.1, since the most aggressive bracket varies from dataset to dataset with the number of training points.

all speedups, we consider random search and "random 2×," a variant of random search with twice the budget of other methods. Of the hybrid methods described in Section 2, we compare to a variant of SMAC using the early termination criterion proposed by Domhan et al. (2015) in the deep learning experiments described in Section 4.1. We think a comparison of Hyperband to more sophisticated hybrid methods introduced recently by Klein et al. (2017a) and Kandasamy et al. (2017) is a fruitful direction for future work.

In the experiments below, we followed these loose guidelines when determining how to configuration Hyperband:

1. The maximum resource R should be reasonable given the problem, but ideally large enough so that early-stopping is beneficial.

2. η should depend on R and be selected to yield ≈ 5 brackets with a minimum of 3 brackets. This is to guarantee that Hyperband will use a baseline degree of early-stopping and prevent too coarse of a grid of n vs B tradeoffs.

4.1 Early-Stopping Iterative Algorithms for Deep Learning

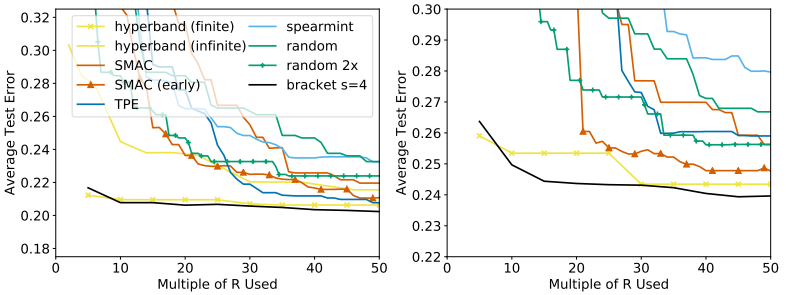
For this benchmark, we tuned a convolutional neural network[7] with the same architecture as that used in Snoek et al. (2012) and Domhan et al. (2015). The search spaces used in the two previous works differ, and we used a search space similar to that of Snoek et al. (2012) with 6 hyperparameters for stochastic gradient decent and 2 hyperparameters for the response normalization layers (see Appendix A for details). In line with the two previous works, we used a batch size of 100 for all experiments.

Data sets: We considered three image classification data sets: CIFAR-10 (Krizhevsky, 2009), rotated MNIST with background images (MRBI) (Larochelle et al., 2007), and Street View House Numbers (SVHN) (Netzer et al., 2011). CIFAR-10 and SVHN contain 32×32 RGB images while MRBI contains 28×28 grayscale images. Each data set was split into a training, validation, and test set: (1) CIFAR-10 has 40k, 10k, and 10k instances; (2) MRBI has 10k, 2k, and 50k instances; and (3) SVHN has close to 600k, 6k, and 26k instances for training, validation, and test respectively. For all data sets, the only preprocessing performed on the raw images was demeaning.

Hyperband Configuration: For these experiments, one unit of resource corresponds to 100 mini-batch iterations (10k examples with a batch size of 100). For CIFAR-10 and MRBI, R was set to 300 (or 30k total iterations). For SVHN, R was set to 600 (or 60k total iterations) to accommodate the larger training set. Given R for these experiments, we set $\eta = 4$ to yield five SuccessiveHalving brackets for Hyperband.

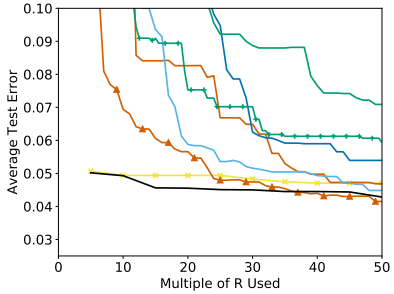
Results: Each searcher was given a total budget of $50R$ per trial to return the best possible hyperparameter configuration. For Hyperband, the budget is sufficient to run the outer loop twice (for a total of 10 SuccessiveHalving brackets). For SMAC, TPE, and random search, the budget corresponds to training 50 different configurations to completion. Ten independent trials were performed for each searcher. The experiments took the equivalent of over 1 year of GPU hours on NVIDIA GRID K520 cards available on Amazon EC2 `g2.8xlarge` instances. We set a total budget constraint in terms of

7. The model specification is available at `http://code.google.com/p/cuda-convnet/`.



(a) CIFAR-10

(b) MRBI



(c) SVHN

Figure 4: Average test error across 10 trials. Label "SMAC (early)" corresponds to SMAC with the early-stopping criterion proposed in Domhan et al. (2015) and label "bracket $s = 4$" corresponds to repeating the most exploratory bracket of Hyperband.

iterations instead of compute time to make comparisons hardware independent.[8] Comparing progress by iterations instead of time ignores overhead costs, e.g. the cost of configuration selection for Bayesian methods and model initialization and validation costs for Hyperband. While overhead is hardware dependent, the overhead for Hyperband is below 5% on EC2 `g2.8xlarge` machines, so comparing progress by time passed would not change results significantly.

For CIFAR-10, the results in Figure 4(a) show that Hyperband is over an order-of-magnitude faster than its competitiors. For MRBI, Hyperband is over an order-of-

8. Most trials were run on Amazon EC2 g2.8xlarge instances but a few trials were run on different machines due to the large computational demand of these experiments.

magnitude faster than standard configuration selection approaches and 5× faster than SMAC (early). For SVHN, while Hyperband finds a good configuration faster, Bayesian optimization methods are competitive and SMAC (early) outperforms Hyperband. The performance of SMAC (early) demonstrates there is merit to combining early-stopping and adaptive configuration selection.

Across the three data sets, Hyperband and SMAC (early) are the only two methods that consistently outperform random 2×. On these data sets, Hyperband is over 20× faster than random search while SMAC (early) is $\leq 7\times$ faster than random search within the evaluation window. In fact, the first result returned by Hyperband after using a budget of $5R$ is often competitive with results returned by other searchers after using $50R$. Additionally, Hyperband is less variable than other searchers across trials, which is highly desirable in practice (see Appendix A for plots with error bars).

As discussed in Section 3.6, for computationally expensive problems in high-dimensional search spaces, it may make sense to just repeat the most exploratory brackets. Similarly, if meta-data is available about a problem or it is known that the quality of a configuration is evident after allocating a small amount of resource, then one should just repeat the most exploratory bracket. Indeed, for these experiments, bracket $s = 4$ vastly outperforms all other methods on CIFAR-10 and MRBI and is nearly tied with SMAC (early) for first on SVHN.

While we set R for these experiments to facilitate comparison to Bayesian methods and random search, it is also reasonable to use infinite horizon Hyperband to grow the maximum resource until a desired level of performance is reached. We evaluate infinite horizon Hyperband on CIFAR-10 using $\eta = 4$ and a starting budget of $B = 2R$. Figure 4(a) shows that infinite horizon Hyperband is competitive with other methods but does not perform as well as finite horizon Hyperband within the $50R$ budget limit. The infinite horizon algorithm underperforms initially because it has to tune the maximum resource R as well and starts with a less aggressive early-stopping rate. This demonstrates that in scenarios where a max resource is known, it is better to use the finite horizon algorithm. Hence, we focus on the finite horizon version of Hyperband for the remainder of our empirical studies.

Finally, CIFAR-10 is a very popular data set and state-of-the-art models achieve much lower error rates than what is shown in Figure 4. The difference in performance is mainly attributable to higher model complexities and data manipulation (i.e. using reflection or random cropping to artificially increase the data set size). If we limit the comparison to published results that use the same architecture and exclude data manipulation, the best human expert result for the data set is 18% error and the best hyperparameter optimized results are 15.0% for Snoek et al. (2012)[9] and 17.2% for Domhan et al. (2015). These results exceed ours on CIFAR-10 because they train on 25% more data, by including the validation set, and also train for more epochs. When we train the best model found by Hyperband on the combined training and validation data for 300 epochs, the model achieved a test error of 17.0%.

9. We were unable to reproduce this result even after receiving the optimal hyperparameters from the authors through a personal communication.

4.2 Data Set Subsampling

We studied two different hyperparameter search optimization problems for which Hyperband uses data set subsamples as the resource. The first adopts an extensive framework presented in Feurer et al. (2015) that attempts to automate preprocessing and model selection. Due to certain limitations of the framework that fundamentally limited the impact of data set downsampling, we conducted a second experiment using a kernel classification task.

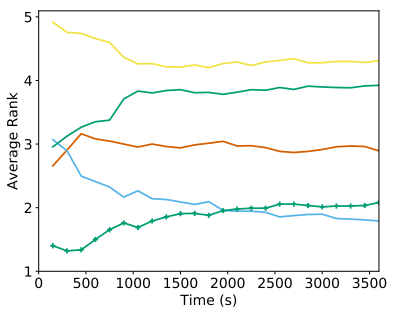
4.2.1 117 Data Sets

We used the framework introduced by Feurer et al. (2015), which explored a structured hyperparameter search space comprised of 15 classifiers, 14 feature preprocessing methods, and 4 data preprocessing methods for a total of 110 hyperparameters. We excluded the meta-learning component introduced in Feurer et al. (2015) used to warmstart Bayesian methods with promising configurations, in order to perform a fair comparison with random search and Hyperband. Similar to Feurer et al. (2015), we imposed a 3GB memory limit, a 6-minute timeout for each hyperparameter configuration and a one-hour time window to evaluate each searcher on each data set. Twenty trials of each searcher were performed per data set and all trials in aggregate took over a year of CPU time on `n1-standard-1` instances from Google Cloud Compute. Additional details about our experimental framework are available in Appendix A.

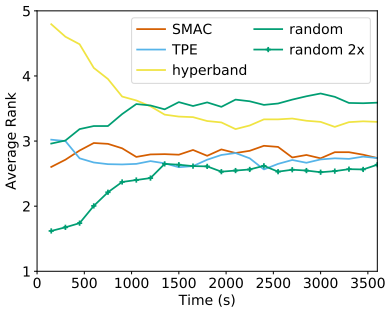
Data sets: Feurer et al. (2015) used 140 binary and multiclass classification data sets from OpenML, but 23 of them are incompatible with the latest version of the OpenML plugin (Feurer, 2015), so we worked with the remaining 117 data sets. Due to the limitations of the experimental setup (discussed in Appendix A), we also separately considered 21 of these data sets, which demonstrated at least modest (though still sublinear) training speedups due to subsampling. Specifically, each of these 21 data sets showed on average at least a 3× speedup due to 8× downsampling on 100 randomly selected hyperparameter configurations.

Hyperband Configuration: Due to the wide range of dataset sizes, with some datasets having fewer than 10k training points, we ran Hyperband with $\eta = 3$ to allow for at least 3 brackets without being overly aggressive in downsampling on small datasets. R was set to the full training set size for each data set and the maximum number of configurations for any bracket of SuccessiveHalving was limited to $n_{\max} = \max\{9, R/1000\}$. This ensured that the most exploratory bracket of Hyperband will downsample at least twice. As mentioned in Section 3.6, when $n_{\max}$ is specified, the only difference when running the algorithm is $s_{\max} = \lfloor \log_\eta(n_{\max}) \rfloor$ instead of $\lfloor \log_\eta(R) \rfloor$.

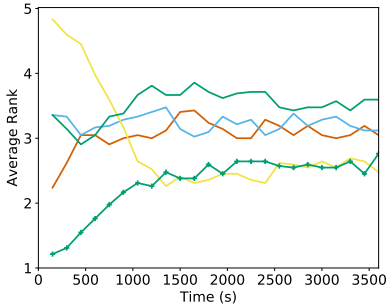
Results: The results on all 117 data sets in Figure 5(a,b) show that Hyperband outperforms random search in test error rank despite performing worse in validation error rank. Bayesian methods outperform Hyperband and random search in test error performance but also exhibit signs of overfitting to the validation set, as they outperform Hyperband by a larger margin on the validation error rank. Notably, random 2× outperforms all other methods. However, for the subset of 21 data sets, Figure 5(c) shows that Hyperband outperforms all other searchers on test error rank, including random 2× by a very small margin. While these results are more promising, the effectiveness of Hyperband was restricted in this experimental framework; for smaller data sets, the startup overhead was



(a) Validation Error on 117 Data Sets



(b) Test Error on 117 Data Sets



(c) Test Error on 21 Data Sets

Figure 5: Average rank across all data sets for each searcher. For each data set, the searchers are ranked according to the average validation/test error across 20 trials.

high relative to total training time, while for larger data sets, only a handful of configurations could be trained within the hour window.

We note that while average rank plots like those in Figure 5 are an effective way to aggregate information across many searchers and data sets, they provide no indication about the *magnitude* of the differences between the performance of the methods. Figure 6, which charts the difference between the test error for each searcher and that of random search across all 117 datasets, highlights the small difference in the magnitude of the test errors across searchers.

These results are not surprising; as mentioned in Section 2.1, vanilla Bayesian optimization methods perform similarly to random search in high-dimensional search spaces. Feurer et al. (2015) showed that using meta-learning to warmstart Bayesian optimization methods improved performance in this high-dimensional setting. Using meta-learning to identify a

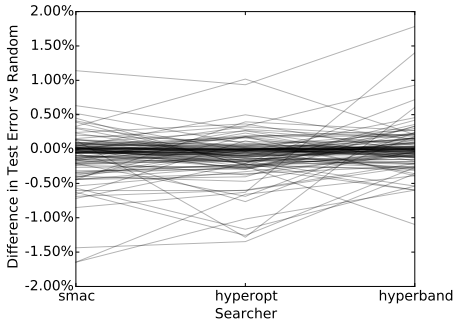


Figure 6: Each line plots, for a single data set, the difference in test error versus random search for each search, where lower is better. Nearly all the lines fall within the -0.5% and 0.5% band and, with the exception of a few outliers, the lines are mostly flat.

promising distribution from which to sample configurations as input into Hyperband is a direction for future work.

4.2.2 Kernel Regularized Least Squares Classification

For this benchmark, we tuned the hyperparameters of a kernel-based classifier on CIFAR-10. We used the multi-class regularized least squares classification model, which is known to have comparable performance to SVMs (Rifkin and Klautau, 2004; Agarwal et al., 2014) but can be trained significantly faster.[10] The hyperparameters considered in the search space include preprocessing method, regularization, kernel type, kernel length scale, and other kernel specific hyperparameters (see Appendix A for more details). For Hyperband, we set $R = 400$, with each unit of resource representing 100 datapoints, and $\eta = 4$ to yield a total of 5 brackets. Each hyperparameter optimization algorithm was run for ten trials on Amazon EC2 `m4.2xlarge` instances; for a given trial, Hyperband was allowed to run for two outer loops, bracket $s = 4$ was repeated 10 times, and all other searchers were run for 12 hours.

Figure 7 shows that Hyperband returned a good configuration after completing the first SuccessiveHalving bracket in approximately 20 minutes; other searchers failed to reach this error rate on average even after the entire 12 hours. Notably, Hyperband was able to evaluate over 250 configurations in this first bracket of SuccessiveHalving, while competitors were able to evaluate only three configurations in the same amount of time. Consequently, Hyperband is over 30× faster than Bayesian optimization methods and 70× faster than random search. Bracket $s = 4$ sightly outperforms Hyperband but the terminal

10. The default SVM method in Scikit-learn is single core and takes hours to train on CIFAR-10, whereas a block coordinate descent least squares solver takes less than 10 minutes on an 8 core machine.

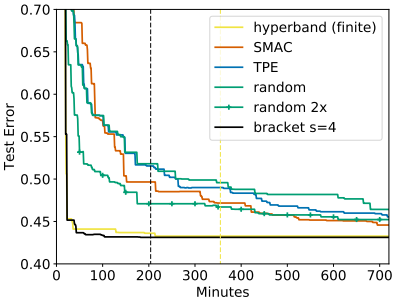


Figure 7: Average test error of the best kernel regularized least square classification model found by each searcher on CIFAR-10. The color coded dashed lines indicate when the last trial of a given searcher finished.

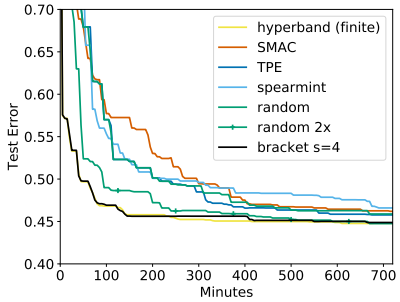


Figure 8: Average test error of the best random features model found by each searcher on CIFAR-10. The test error for Hyperband and bracket $s = 4$ are calculated in every evaluation instead of at the end of a bracket.

performance for the two algorithms are the same. Random 2× is competitive with SMAC and TPE.

4.3 Feature Subsampling to Speed Up Approximate Kernel Classification

Next, we examine the performance of Hyperband when using features as a resource on a random feature kernel approximations task. Features were randomly generated using the method described in Rahimi and Recht (2007) to approximate the RBF kernel, and these random features were then used as inputs to a ridge regression classifier. The hyperparameter search space included the preprocessing method, kernel length scale, and L_2 penalty. While it may seem natural to use infinite horizon Hyperband, since the fidelity of the approximation improves with more random features, in practice, the amount of available machine memory imposes a natural upper bound on the number of features. Thus, we used finite horizion Hyperband with a maximum resource of 100k random features, which comfortably fit into a machine with 60GB of memory. Additionally, we set one unit of resource to be 100 features, so $R = 1000$. Again, we set $\eta = 4$ to yield 5 brackets of SuccessiveHalving. We ran 10 trials of each searcher, with each trial lasting 12 hours on a `n1-standard-16` machine from Google Cloud Compute. The results in Figure 8 show that Hyperband is around 6× faster than Bayesian methods and random search. Hyperband performs similarly to bracket $s = 4$. Random 2× outperforms Bayesian optimization algorithms.

4.4 Experimental Discussion

While our experimental results show Hyperband is a promising algorithm for hyperparameter optimization, a few questions naturally arise:

1. What impacts the speedups provided by Hyperband?
2. Why does SuccessiveHalving seem to outperform Hyperband?
3. What about hyperparameters that should depend on the resource?

We next address each of these questions in turn.

4.4.1 Factors Impacting the Performance of Hyperband

For a given R, the most exploratory SuccessiveHalving round performed by Hyperband evaluates R configurations using a budget of $(\lfloor \log_\eta(R) \rfloor + 1)R$, which gives an upper bound on the potential speedup over random search. If training time scales linearly with the resource, the maximum speedup offered by Hyperband compared to random search is $\frac{R}{(\lfloor \log_\eta(R) \rfloor + 1)}$. For the values of η and R used in our experiments, the maximum speedup over random search is approximately 50× given linear training time. However, we observe a range of speedups from 6× to 70× faster than random search. The differences in realized speedup can be explained by three factors:

1. *How training time scales with the given resource.* In cases where training time is superlinear as a function of the resource, Hyperband can offer higher speedups. For instance, if training scales like a polynomial of degree $p > 1$, the maximum speedup for Hyperband over random search is approximately $\frac{\eta^p - 1}{\eta^p - 1} R$. In the kernel least square classifier experiment discussed in Section 4.2.2, the training time scaled quadratically as a function of the resource, which explains why the realized speedup of 70× is higher than the maximum expected speedup given linear scaling.

2. *Overhead costs associated with training.* Total evaluation time also depends on fixed overhead costs associated with evaluating each hyperparameter configuration, e.g., initializing a model, resuming previously trained models, and calculating validation error. For example, in the downsampling experiments on 117 data sets presented in Section 4.2.1, Hyperband did not provide significant speedup because many data sets could be trained in a matter of a few seconds and the initialization cost was high relative to training time.

3. *The difficulty of finding a good configuration.* Hyperparameter optimization problems can vary in difficulty. For instance, an 'easy' problem is one where a randomly sampled configuration is likely to result in a high-quality model, and thus we only need to evaluate a small number of configurations to find a good setting. In contrast, a 'hard' problem is one where an arbitrary configuration is likely to be bad, in which case many configurations must be considered. Hyperband leverages downsampling to boost the number of configurations that are evaluated, and thus is better suited for 'hard' problems where more evaluations are actually necessary to find a good setting. Generally, the difficulty of a problem scales with the dimensionality of the search space. For low-dimensional problems, the number of configurations evaluated by random search and Bayesian methods is exponential in the number of dimensions so good coverage can be achieved. For instance, the low-dimensional ($d = 3$) search space in our feature subsampling experiment in Section 4.3 helps explain why Hyperband is

only 6× faster than random search. In contrast, for the neural network experiments in Section 4.1, we hypothesize that faster speedups are observed for Hyperband because the dimension of the search space is higher.

4.4.2 Comparison to SuccessiveHalving

With the exception of the LeNet experiment (Section 3.3) and the 117 Datasets experiment (Section 4.2.1), the most aggressive bracket of SuccessiveHalving outperformed Hyperband in all of our experiments. In hindsight, we should have just run bracket $s = 4$, since aggressive early-stopping provides massive speedups on many of these benchmarking tasks. However, as previously mentioned, it was unknown a priori that bracket $s = 4$ would perform the best and that is why we have to cycle through all possible brackets with Hyperband. Another question is what happens when one increases s even further, i.e. instead of 4 rounds of elimination, why not 5 or even more with the same maximum resource R? In our case, $s = 4$ was the most aggressive bracket we could run given the minimum resource per configuration limits imposed for the previous experiments. However, for larger data sets, it is possible to extend the range of possible values for s, in which case, Hyperband may either provide even faster speedups if more aggressive early-stopping helps or be slower by a small factor if the most aggressive brackets are essentially throwaways.

We believe prior knowledge about a task can be particularly useful for limiting the range of brackets explored by Hyperband. In our experience, aggressive early-stopping is generally safe for neural network tasks and even more aggressive early-stopping may be reasonable for larger data sets and longer training horizons. However, when pushing the degree of early-stopping by increasing s, one has to consider the additional overhead cost associated with examining more models. Hence, one way to leverage meta-learning would be to use learning curve convergence rate, difficulty of different search spaces, and overhead costs of related tasks to determine the brackets considered by Hyperband.

4.4.3 Resource Dependent Hyperparameters

In certain cases, the setting for a given hyperparameter should depend on the allocated resource. For example, the maximum tree depth regularization hyperparameter for random forests should be higher with more data and more features. However, the optimal tradeoff between maximum tree depth and the resource is unknown and can be data set specific. In these situations, the rate of convergence to the true loss is usually slow because the performance on a smaller resource is not indicative of that on a larger resource. Hence, these problems are particularly difficult for Hyperband, since the benefit of early-stopping can be muted. Again, while Hyperband will only be a small factor slower than that of SuccessiveHalving with the optimal early-stopping rate, we recommend removing the dependence of the hyperparameter on the resource if possible. For the random forest example, an alternative regularization hyperparameter is minimum samples per leaf, which is less dependent on the training set size. Additionally, the dependence can oftentimes be removed with simple normalization. For example, the regularization term for our kernel least squares experiments were normalized by the training set size to maintain a constant tradeoff between the mean-squared error and the regularization term.

5. Theory

In this section, we introduce the pure-exploration non-stochastic infinite-armed bandit (NIAB) problem, a very general setting which encompasses our hyperparameter optimization problem of interest. As we will show, Hyperband is in fact applicable to problems far beyond just hyperparameter optimization. We begin by formalizing the hyperparameter optimization problem and then reducing it to the pure-exploration NIAB problem. We subsequently present a detailed analysis of Hyperband in both the infinite and finite horizon settings.

5.1 Hyperparameter Optimization Problem Statement

Let $\mathcal{X}$ denote the space of valid hyperparameter configurations, which could include continuous, discrete, or categorical variables that can be constrained with respect to each other in arbitrary ways (i.e. $\mathcal{X}$ need not be limited to a subset of $[0,1]^d$). For $k = 1, 2, \ldots$ let $\ell_k \colon \mathcal{X} \to [0,1]$ be a sequence of loss functions defined over $\mathcal{X}$. For any hyperparameter configuration $x \in \mathcal{X}$, $\ell_k(x)$ represents the validation error of the model trained using x with k units of resources (e.g. iterations). In addition, for some $R \in \mathbb{N} \cup \{\infty\}$, define $\ell_* = \lim_{k \to R} \ell_k$ and $\nu_* = \inf_{x \in \mathcal{X}} \ell_*(x)$. Note that $\ell_k(\cdot)$ for all $k \in \mathbb{N}$, $\ell_*(\cdot)$, and ν_* are all unknown to the algorithm a priori. In particular, it is uncertain how quickly $\ell_k(x)$ varies as a function of x for any fixed k, and how quickly $\ell_k(x) \to \ell_*(x)$ as a function of k for any fixed $x \in \mathcal{X}$.

We assume hyperparameter configurations are sampled randomly from a known probability distribution $p(x) \colon \mathcal{X} \to [0, \infty)$, with support on $\mathcal{X}$. In our experiments, $p(x)$ is simply the uniform distribution, but the algorithm can be used with any sampling method. If $X \in \mathcal{X}$ is a random sample from this probability distribution, then $\ell_*(X)$ is a random variable whose distribution is unknown since $\ell_*(\cdot)$ is unknown. Additionally, since it is unknown how $\ell_k(x)$ varies as a function of x or k, one cannot necessarily infer anything about $\ell_k(x)$ given knowledge of $\ell_j(y)$ for any $j \in \mathbb{N}$, $y \in \mathcal{X}$. As a consequence, we reduce the hyperparmeter optimization problem down to a much simpler problem that ignores all underlying structure of the hyperparameters: we only interact with some $x \in \mathcal{X}$ through its loss sequence $\ell_k(x)$ for $k = 1, 2, \ldots$. With this reduction, the particular value of $x \in \mathcal{X}$ does nothing more than index or uniquely identify the loss sequence.

Without knowledge of how fast $\ell_k(\cdot) \to \ell_*(\cdot)$ or how $\ell_*(X)$ is distributed, the goal of Hyperband is to identify a hyperparameter configuration $x \in \mathcal{X}$ that minimizes $\ell_*(x) - \nu_*$ by drawing as many random configurations as desired while using as few total resources as possible.

5.2 The Pure-Exploration Non-stochastic Infinite-Armed Bandit Problem

We now formally define the bandit problem of interest, and relate it to the problem of hyperparameter optimization. Each "arm" in the NIAB game is associated with a sequence that is drawn randomly from a distribution over sequences. If we "pull" the ith drawn arm exactly k times, we observe a loss $\ell_{i,k}$. At each time, the player can either draw a new arm (sequence) or pull a previously drawn arm an additional time. There is no limit on the number of arms that can be drawn. We assume the arms are identifiable only by their index

i (i.e. we have no side-knowledge or feature representation of an arm), and we also make the following two additional assumptions:

Assumption 1 *For each $i \in \mathbb{N}$ the limit $\lim_{k\to\infty} \ell_{i,k}$ exists and is equal to ν_i.*[11]

Assumption 2 *Each ν_i is a bounded i.i.d. random variable with cumulative distribution function F.*

The objective of the NIAB problem is to identify an arm $\hat{\imath}$ with small $\nu_{\hat{\imath}}$ using as few total pulls as possible. We are interested in characterizing $\nu_{\hat{\imath}}$ as a function of the total number of pulls from all the arms. Clearly, the hyperparameter optimization problem described above is an instance of the NIAB problem. In this case, arm i correspondes to a configuration $x_i \in \mathcal{X}$, with $\ell_{i,k} = \ell_k(x_i)$; Assumption 1 is equivalent to requiring that $\nu_i = \ell_*(x_i)$ exists; and Assumption 2 follows from the fact that the arms are drawn i.i.d. from $\mathcal{X}$ according to distribution function $p(x)$. F is simply the cumulative distribution function of $\ell_*(X)$, where X is a random variable drawn from the distribution $p(x)$ over $\mathcal{X}$. Note that since the arm draws are independent, the ν_i's are also independent. Again, this is not to say that the validation losses do not depend on the settings of the hyperparameters; the validation loss could well be correlated with certain hyperparameters, but this is not used in the algorithm and no assumptions are made regarding the correlation structure.

In order to analyze the behavior of Hyperband in the NIAB setting, we must define a few additional objects. Let $\nu_* = \inf\{m : \mathbb{P}(\nu \le m) > 0\} > -\infty$, since the domain of the distribution F is bounded. Hence, the cumulative distribution function F satisfies

$$\mathbb{P}(\nu_i - \nu_* \le \epsilon) = F(\nu_* + \epsilon) \tag{1}$$

and let $F^{-1}(y) = \inf_x\{x : F(x) \le y\}$. Define $\gamma\colon \mathbb{N} \to \mathbb{R}$ as the pointwise smallest, monotonically decreasing function satisfying

$$\sup_i |\ell_{i,j} - \ell_{i,*}| \le \gamma(j)\ , \quad \forall j \in \mathbb{N}. \tag{2}$$

The function γ is guaranteed to exist by Assumption 1 and bounds the deviation from the limit value as the sequence of iterates j increases. For hyperparameter optimization, this follows from the fact that ℓ_k uniformly converges to ℓ_* for all $x \in \mathcal{X}$. In addition, γ can be interpretted as the deviation of the validation error of a configuration trained on a subset of resources versus the maximum number of allocatable resources. Finally, define R as the first index such that $\gamma(R) = 0$ if it exists, otherwise set $R = \infty$. For $y \ge 0$ let $\gamma^{-1}(y) = \min\{j \in \mathbb{N}\colon \gamma(j) \le y\}$, using the convention that $\gamma^{-1}(0) := R$ which we recall can be infinite.

As previously discussed, there are many real-world scenarios in which R is finite and known. For instance, if increasing subsets of the full data set is used as a resource, then the maximum number of resources cannot exceed the full data set size, and thus $\gamma(k) = 0$ for all $k \ge R$ where R is the (known) full size of the data set. In other cases such as iterative training problems, one might not want to or know how to bound R. We separate these two settings into the *finite horizon* setting where R is finite and known, and the *infinite horizon*

11. We can always define $\ell_{i,k}$ so that convergence is guaranteed, i.e. taking the infimum of a sequence.

SuccessiveHalving (Infinite horizon)
Input: Budget B, n arms where $\ell_{i,k}$ denotes the kth loss from the ith arm
Initialize: $S_0 = [n]$.
For $k = 0, 1, \ldots, \lceil \log_2(n) \rceil - 1$
  Pull each arm in S_k for $r_k = \lfloor \frac{B}{|S_k|\lceil \log_2(n) \rceil} \rfloor$ times.
  Keep the best $\lfloor |S_k|/2 \rfloor$ arms in terms of the r_kth observed loss as S_{k+1}.
Output : $\hat{\imath}$, $\ell_{\hat{\imath}, \lfloor \frac{B/2}{\lceil \log_2(n) \rceil} \rfloor}$ where $\hat{\imath} = S_{\lceil \log_2(n) \rceil}$

Hyperband (Infinite horizon)
Input: None
For $k = 1, 2, \ldots$
  For $s \in \mathbb{N}$ s.t. $k - s \ge \log_2(s)$
    $B_{k,s} = 2^k$, $n_{k,s} = 2^s$
    $\hat{\imath}_{k,s}, \ell_{\hat{\imath}_{k,s}, \lfloor \frac{2^{k-1}}{s} \rfloor} \leftarrow$ SuccessiveHalving $(B_{k,s}, n_{k,s})$

Figure 9: (Top) The SuccessiveHalving algorithm proposed and analyzed in Jamieson and Talwalkar (2015) for the non-stochastic setting. Note this algorithm was originally proposed for the stochastic setting in Karnin et al. (2013). (Bottom) The Hyperband algorithm for the infinite horizon setting. Hyperband calls SuccessiveHalving as a subroutine.

setting where no bound on R is known and it is assumed to be infinite. While our empirical results suggest that the finite horizon may be more practically relevant for the problem of hyperparameter optimization, the infinite horizon case has natural connections to the literature, and we begin by analyzing this setting.

5.3 Infinite Horizon Setting ($R = \infty$)

Consider the Hyperband algorithm of Figure 9. The algorithm uses SuccessiveHalving (Figure 9) as a subroutine that takes a finite set of arms as input and outputs an estimate of the best performing arm in the set. We first analyze SuccessiveHalving (SH) for a given set of limits ν_i and then consider the performance of SH when ν_i are drawn randomly according to F. We then analyze the Hyperband algorithm. We note that the algorithm of Figure 9 was originally proposed by Karnin et al. (2013) for the stochastic setting. However, Jamieson and Talwalkar (2015) analyzed it in the non-stochastic setting and also found it to work well in practice. Extending the result of Jamieson and Talwalkar (2015) we have the following theorem:

Theorem 1 *Fix n arms. Let $\nu_i = \lim_{\tau\to\infty} \ell_{i,\tau}$ and assume $\nu_1 \leq \cdots \leq \nu_n$. For any $\epsilon > 0$ let*

$$z_{SH} = 2\lceil\log_2(n)\rceil \max_{i=2,\ldots,n} i\,(1+\gamma^{-1}\left(\max\left\{\tfrac{\epsilon}{4}, \tfrac{\nu_i-\nu_1}{2}\right\}\right))$$
$$\leq 2\lceil\log_2(n)\rceil\left(n+\sum_{i=1,\ldots,n}\gamma^{-1}\left(\max\left\{\tfrac{\epsilon}{4}, \tfrac{\nu_i-\nu_1}{2}\right\}\right)\right)$$

If the SuccessiveHalving *algorithm of Figure 9 is run with any budget $B > z_{SH}$ then an arm $\hat{\imath}$ is returned that satisfies $\nu_{\hat{\imath}} - \nu_1 \leq \epsilon/2$. Moreover, $|\ell_{\hat{\imath},\lfloor\frac{B/2}{\lceil\log_2(n)\rceil}\rfloor} - \nu_1| \leq \epsilon$.*

The next technical lemma will be used to characterize the problem dependent term $\sum_{i=1,\ldots,n}\gamma^{-1}\left(\max\left\{\frac{\epsilon}{4}, \frac{\nu_i-\nu_1}{2}\right\}\right)$ when the sequences are drawn from a probability distribution.

Lemma 2 *Fix $\delta \in (0,1)$. Let $p_n = \frac{\log(2/\delta)}{n}$. For any $\epsilon \geq 4(F^{-1}(p_n) - \nu_*)$ define*

$$\mathbf{H}(F,\gamma,n,\delta,\epsilon) := 2n\int_{\nu_*+\epsilon/4}^{\infty}\gamma^{-1}(\tfrac{t-\nu_*}{4})dF(t) + \left(\tfrac{4}{3}\log(2/\delta) + 2nF(\nu_*+\epsilon/4)\right)\gamma^{-1}\left(\tfrac{\epsilon}{16}\right)$$

and $\mathbf{H}(F,\gamma,n,\delta) := \mathbf{H}(F,\gamma,n,\delta,4(F^{-1}(p_n)-\nu_*))$ so that

$$\mathbf{H}(F,\gamma,n,\delta) = 2n\int_{p_n}^{1}\gamma^{-1}(\tfrac{F^{-1}(t)-\nu_*}{4})dt + \tfrac{10}{3}\log(2/\delta)\gamma^{-1}\left(\tfrac{F^{-1}(p_n)-\nu_*}{4}\right).$$

For n arms with limits $\nu_1 \leq \cdots \leq \nu_n$ drawn from F, then

$$\nu_1 \leq F^{-1}(p_n) \quad \text{and} \quad \sum_{i=1}^{n}\gamma^{-1}\left(\max\left\{\tfrac{\epsilon}{4}, \tfrac{\nu_i-\nu_1}{2}\right\}\right) \leq \mathbf{H}(F,\gamma,n,\delta,\epsilon)$$

for any $\epsilon \geq 4(F^{-1}(p_n)-\nu_*)$ with probability at least $1-\delta$.

Setting $\epsilon = 4(F^{-1}(p_n)-\nu_*)$ in Theorem 1 and using the result of Lemma 2 that $\nu_* \leq \nu_1 \leq \nu_* + (F^{-1}(p_n)-\nu_*)$, we immediately obtain the following corollary.

Corollary 3 *Fix $\delta \in (0,1)$ and $\epsilon \geq 4(F^{-1}(\frac{\log(2/\delta)}{n}) - \nu_*)$. Let $B = 4\lceil\log_2(n)\rceil\mathbf{H}(F,\gamma,n,\delta,\epsilon)$ where $\mathbf{H}(F,\gamma,n,\delta,\epsilon)$ is defined in Lemma 2. If the* SuccessiveHalving *algorithm of Figure 9 is run with the specified B and n arm configurations drawn randomly according to F, then an arm $\hat{\imath} \in [n]$ is returned such that with probability at least $1-\delta$ we have $\nu_{\hat{\imath}} - \nu_* \leq \left(F^{-1}(\frac{\log(2/\delta)}{n}) - \nu_*\right) + \epsilon/2$. In particular, if $B = 4\lceil\log_2(n)\rceil\mathbf{H}(F,\gamma,n,\delta)$ and $\epsilon = 4(F^{-1}(\frac{\log(2/\delta)}{n}) - \nu_*)$ then $\nu_{\hat{\imath}} - \nu_* \leq 3\left(F^{-1}(\frac{\log(2/\delta)}{n}) - \nu_*\right)$ with probability at least $1-\delta$.*

Note that for any fixed $n \in \mathbb{N}$ we have for any $\Delta > 0$

$$\mathbb{P}(\min_{i=1,\ldots,n}\nu_i - \nu_* \geq \Delta) = (1-F(\nu_*+\Delta))^n \approx e^{-nF(\nu_*+\Delta)}$$

which implies $\mathbb{E}[\min_{i=1,\ldots,n}\nu_i - \nu_*] \approx F^{-1}(\frac{1}{n}) - \nu_*$. That is, n needs to be sufficiently large so that it is probable that a good limit is sampled. On the other hand, for any fixed n, Corollary 3 suggests that the total resource budget B needs to be large enough in order to overcome the rates of convergence of the sequences described by γ. Next, we relate SH to a naive approach that uniformly allocates resources to a fixed set of n arms.

5.3.1 Non-Adaptive Uniform Allocation

The non-adaptive uniform allocation strategy takes as inputs a budget B and n arms, allocates B/n to each of the arms, and picks the arm with the lowest loss. The following results allow us to compare with SuccessiveHalving.

Proposition 4 *Suppose we draw n random configurations from F, train each with $j = \min\{B/n, R\}$ iterations, and let $\hat{\imath} = \arg\min_{i=1,\ldots,n}\ell_j(X_i)$. Without loss of generality assume $\nu_1 \leq \ldots \leq \nu_n$. If*

$$B \geq n\gamma^{-1}\left(\tfrac{1}{2}(F^{-1}(\tfrac{\log(1/\delta)}{n}) - \nu_*)\right) \tag{3}$$

then with probability at least $1-\delta$ we have $\nu_{\hat{\imath}} - \nu_* \leq 2\left(F^{-1}\left(\frac{\log(1/\delta)}{n}\right) - \nu_*\right)$. In contrast, there exists a sequence of functions ℓ_j that satisfy F and γ such that if

$$B \leq n\gamma^{-1}\left(2(F^{-1}(\tfrac{\log(c/\delta)}{n+\log(c/\delta)}) - \nu_*)\right)$$

then with probability at least δ, we have $\nu_{\hat{\imath}} - \nu_* \geq 2(F^{-1}(\frac{\log(c/\delta)}{n+\log(c/\delta)}) - \nu_*)$, where c is a constant that depends on the regularity of F.

For any fixed n and sufficiently large B, Corollary 3 shows that SuccessiveHalving outputs an $\hat{\imath} \in [n]$ that satisfies $\nu_{\hat{\imath}} - \nu_* \lesssim F^{-1}(\frac{\log(2/\delta)}{n}) - \nu_*$ with probability at least $1-\delta$. This guarantee is similar to the result in Proposition 4. However, SuccessiveHalving achieves its guarantee as long as[12]

$$B \simeq \log_2(n)\left[\log(1/\delta)\gamma^{-1}\left(F^{-1}(\tfrac{\log(1/\delta)}{n}) - \nu_*\right) + n\int_{\frac{\log(1/\delta)}{n}}^{1}\gamma^{-1}(F^{-1}(t)-\nu_*)dt\right], \tag{4}$$

and this sample complexity may be substantially smaller than the budget required by uniform allocation shown in Eq. (3) of Proposition 4. Essentially, the first term in Eq. (4) represents the budget allocated to the constant number of arms with limits $\nu_i \approx F^{-1}(\frac{\log(1/\delta)}{n})$ while the second term describes the number of times the sub-optimal arms are sampled before discarded. The next section uses a particular parameterization for F and γ to help better illustrate the difference between the sample complexity of uniform allocation (Equation 3) versus that of SuccessiveHalving (Equation 4).

5.3.2 A Parameterization of F and γ for Interpretability

To gain some intuition and relate the results back to the existing literature, we make explicit parametric assumptions on F and γ. We stress that all of our results hold for general F and γ as previously stated, and this parameterization is simply a tool to provide intuition. First assume that there exists a constant $\alpha > 0$ such that

$$\gamma(j) \simeq \left(\frac{1}{j}\right)^{1/\alpha}. \tag{5}$$

12. We say $f \simeq g$ if there exist constants c, c' such that $cg(x) \leq f(x) \leq c'g(x)$.

Note that a large value of α implies that the convergence of $\ell_{i,k} \to \nu_i$ is very slow.

We will consider two possible parameterizations of F. First, assume there exists positive constants β such that

$$F(x) \simeq \begin{cases} (x-\nu_*)^\beta & \text{if } x \geq \nu_* \\ 0 & \text{if } x < \nu_* \end{cases}. \tag{6}$$

Here, a large value of β implies that it is very rare to draw a limit close to the optimal value ν_*. The same model was studied in Carpentier and Valko (2015). Fix some $\Delta > 0$. As discussed in the preceding section, if $n = \frac{\log(1/\delta)}{F(\nu_*+\Delta)} \simeq \Delta^{-\beta}\log(1/\delta)$ arms are drawn from F then with probability at least $1-\delta$ we have $\min_{i=1,\dots,n} \nu_i \leq \nu_* + \Delta$. Predictably, both uniform allocation and SuccessiveHalving output a $\nu_{\hat{\imath}}$ that satisfies $\nu_{\hat{\imath}} - \nu_* \lesssim \left(\frac{\log(1/\delta)}{n}\right)^{1/\beta}$ with probability at least $1-\delta$ provided their measurement budgets are large enough. Thus, if $n \simeq \Delta^{-\beta}\log(1/\delta)$ and the measurement budgets of the uniform allocation (Equation 3) and SuccessiveHalving (Equation 4) satisfy

$$\begin{aligned} \text{Uniform allocation} \quad & B \simeq \Delta^{-(\alpha+\beta)}\log(1/\delta) \\ \text{SuccessiveHalving} \quad & B \simeq \log_2(\Delta^{-\beta}\log(1/\delta))\left[\Delta^{-\alpha}\log(1/\delta) + \frac{\Delta^{-\beta}-\Delta^{-\alpha}}{1-\alpha/\beta}\log(1/\delta)\right] \\ & \simeq \log(\Delta^{-1}\log(1/\delta))\log(\Delta^{-1})\ \Delta^{-\max\{\beta,\alpha\}}\log(1/\delta) \end{aligned}$$

then both also satisfy $\nu_{\hat{\imath}} - \nu_* \lesssim \Delta$ with probability at least $1-\delta$.[13] SuccessiveHalving's budget scales like $\Delta^{-\max\{\alpha,\beta\}}$, which can be significantly smaller than the uniform allocation's budget of $\Delta^{-(\alpha+\beta)}$. However, because α and β are unknown in practice, neither method knows how to choose the optimal n or B to achieve this Δ accuracy. In Section 5.3.3, we show how Hyperband addresses this issue.

The second parameterization of F is the following discrete distribution:

$$F(x) = \frac{1}{K}\sum_{j=1}^{K} \mathbf{1}\{x \leq \mu_j\} \qquad \text{with} \qquad \Delta_j := \mu_j - \mu_1 \tag{7}$$

for some set of unique scalars $\mu_1 < \mu_2 < \cdots < \mu_K$. Note that by letting $K \to \infty$ this discrete CDF can approximate any piecewise-continuous CDF to arbitrary accuracy. In particular, this model can have multiple means take the same value so that α mass is on μ_1 and $1-\alpha$ mass is on $\mu_2 > \mu_1$, capturing the stochastic infinite-armed bandit model of Jamieson et al. (2016). In this setting, both uniform allocation and SuccessiveHalving output a $\nu_{\hat{\imath}}$ that is within the top $\frac{\log(1/\delta)}{n}$ fraction of the K arms with probability at least $1-\delta$ if their budgets are sufficiently large. Thus, let $q > 0$ be such that $n \simeq q^{-1}\log(1/\delta)$. Then, if the measurement budgets of the uniform allocation (Equation 3) and SuccessiveHalving (Equation 4) satisfy

$$\begin{aligned} \text{Uniform allocation} \quad & B \simeq \log(1/\delta)\begin{cases} K \max_{j=2,\dots,K} \Delta_j^{-\alpha} & \text{if } q = 1/K \\ q^{-1}\Delta_{\lceil qK \rceil}^{-\alpha} & \text{if } q > 1/K \end{cases} \\ \text{SuccessiveHalving} \quad & B \simeq \log(q^{-1}\log(1/\delta))\log(1/\delta)\begin{cases} \Delta_2^{-\alpha} + \sum_{j=2}^{K} \Delta_j^{-\alpha} & \text{if } q = 1/K \\ \Delta_{\lceil qK \rceil}^{-\alpha} + \frac{1}{qK}\sum_{j=\lceil qK \rceil}^{K} \Delta_j^{-\alpha} & \text{if } q > 1/K, \end{cases} \end{aligned}$$

an arm that is in the best q-fraction of arms is returned, i.e. $\hat{\imath}/K \approx q$ and $\nu_{\hat{\imath}} - \nu_* \lesssim \Delta_{\lceil \max\{2,qK\} \rceil}$, with probability at least $1-\delta$. This shows that the average resource per arm for uniform allocation is that required to distinguish the top q-fraction from the best, while that for SuccessiveHalving is a small multiple of the average resource required to distinguish an arm from the best; the difference between the max and the average can be very large in practice. We remark that the value of ϵ in Corollary 3 is carefully chosen to make the SuccessiveHalving budget and guarantee work out. Also note that one would never take $q < 1/K$ because $q = 1/K$ is sufficient to return the best arm.

5.3.3 Hyperband Guarantees

The Hyperband algorithm of Figure 9 addresses the tradeoff between the number of arms n versus the average number of times each one is pulled B/n by performing a two-dimensional version of the so-called "doubling trick." For each fixed B, we non-adaptively search a predetermined grid of values of n spaced geometrically apart so that the incurred loss of identifying the "best" setting takes a budget no more than $\log(B)$ times the budget necessary if the best setting of n were known ahead of time. Then, we successively double B so that the cumulative number of measurements needed to arrive at the necessary B is no more than $2B$. The idea is that even though we do not know the optimal setting for B, n to achieve some desired error rate, the hope is that by trying different values in a particular order, we will not waste too much effort.

Fix $\delta \in (0,1)$. For all (k,s) pairs defined in the Hyperband algorithm of Figure 9, let $\delta_{k,s} = \frac{\delta}{2k^3}$. For all (k,s) define

$$\mathcal{E}_{k,s} := \{B_{k,s} > 4\lceil \log_2(n_{k,s}) \rceil \mathbf{H}(F, \gamma, n_{k,s}, \delta_{k,s})\} = \{2^k > 4s\mathbf{H}(F, \gamma, 2^s, 2k^3/\delta)\}$$

Then by Corollary 3 we have

$$\mathbb{P}\left(\bigcup_{k=1}^{\infty}\bigcup_{s=1}^{k}\{\nu_{i_{k,s}} - \nu_* > 3\big(F^{-1}\big(\tfrac{\log(4k^3/\delta)}{2^s}\big) - \nu_*\big)\} \cap \mathcal{E}_{k,s}\right) \leq \sum_{k=1}^{\infty}\sum_{s=1}^{k}\delta_{k,s} = \sum_{k=1}^{\infty}\frac{\delta}{2k^2} \leq \delta.$$

For sufficiently large k we will have $\bigcup_{s=1}^{k}\mathcal{E}_{k,s} \neq \emptyset$, so assume $B = 2^k$ is sufficiently large. Let $\hat{\imath}_B$ be the empirically best-performing arm output from SuccessiveHalving of round $k_B = \lfloor \log_2(B) \rfloor$ of Hyperband of Figure 9 and let $s_B \leq k_B$ be the largest value such that

13. These quantities are intermediate results in the proofs of the theorems of Section 5.3.3.

$\mathcal{E}_{k_B,s_B}$ holds. Then

$$\nu_{i_B} - \nu_* \leq 3\big(F^{-1}(\frac{\log(4\lfloor \log_2(B)\rfloor^3/\delta)}{2^{s_B}}) - \nu_*\big) + \gamma(\lfloor \frac{2^{\lfloor \log_2(B)\rfloor - 1}}{\lfloor \log_2(B)\rfloor}\rfloor).$$

Also note that on stage k at most $\sum_{i=1}^{k} iB_{i,1} \leq k\sum_{i=1}^{k} B_{i,1} \leq 2kB_{k,s} = 2\log_2(B_{k,s})B_{k,s}$ total samples have been taken. While this guarantee holds for general F, γ, the value of s_B, and consequently the resulting bound, is difficult to interpret. The following corollary considers the β, α parameterizations of F and γ, respectively, of Section 5.3.2 for better interpretation.

Theorem 5 *Assume that Assumptions 1 and 2 of Section 5.2 hold and that the sampled loss sequences obey the parametric assumptions of Equations 5 and 6. Fix $\delta \in (0,1)$. For any $T \in \mathbb{N}$, let $\hat{\imath}_T$ be the empirically best-performing arm output from* SuccessiveHalving *from the last round k of* Hyperband *of Figure 9 after exhausting a total budget of T from all rounds, then*

$$\nu_{\hat{\imath}_T} - \nu_* \leq c\left(\frac{\overline{\log}(T)^3\overline{\log}(\log(T)/\delta)}{T}\right)^{1/\max\{\alpha,\beta\}}$$

for some constant $c = \exp(O(\max\{\alpha,\beta\}))$ where $\overline{\log}(x) = \log(x)\log\log(x)$.

By a straightforward modification of the proof, one can show that if uniform allocation is used in place of SuccessiveHalving in Hyperband, the uniform allocation version achieves $\nu_{\hat{\imath}_T} - \nu_* \leq c\left(\frac{\log(T)\overline{\log}(\log(T)/\delta)}{T}\right)^{1/(\alpha+\beta)}$. We apply the above theorem to the stochastic infinite-armed bandit setting in the following corollary.

Corollary 6 *[Stochastic Infinite-armed Bandits] For any step k, s in the infinite horizon* Hyperband *algorithm with $n_{k,s}$ arms drawn, consider the setting where the jth pull of the ith arm results in a stochastic loss $Y_{i,j} \in [0,1]$ such that $\mathbb{E}[Y_{i,j}] = \nu_i$ and $\mathbb{P}(\nu_i - \nu_* \leq \epsilon) = c_1^{-1}\epsilon^{\beta}$. If $\ell_j(i) = \frac{1}{j}\sum_{s=1}^{j} Y_{i,s}$ then with probability at least $1 - \delta/2$ we have $\forall k \geq 1, 0 \leq s \leq k, 1 \leq i \leq n_{k,s}, 1 \leq j \leq B_k$,*

$$|\nu_i - \ell_{i,j}| \leq \sqrt{\frac{\log(B_k n_{k,s}/\delta_{k,s})}{2j}} \leq \sqrt{\log(\tfrac{16B_k}{\delta})}\left(\tfrac{2}{j}\right)^{1/2}.$$

Consequently, if after B total pulls we define $\widehat{\nu}_B$ as the mean of the empirically best arm output from the last fully completed round k, then with probability at least $1 - \delta$

$$\widehat{\nu}_B - \nu_* \leq \mathrm{polylog}(B/\delta)\max\{B^{-1/2}, B^{-1/\beta}\}.$$

The result of this corollary matches the anytime result of Section 4.3 of Carpentier and Valko (2015) whose algorithm was built specifically for the case of stochastic arms and the β parameterization of F defined in Eq. (6). Notably, this result also matches the *lower bounds* shown in that work up to poly-logarithmic factors, revealing that Hyperband is nearly tight for this important special case. However, we note that this earlier work has a more careful analysis for the fixed budget setting.

Theorem 7 *Assume that Assumptions 1 and 2 of Section 5.2 hold and that the sampled loss sequences obey the parametric assumptions of Equations 5 and 7. For any $T \in \mathbb{N}$, let $\hat{\imath}_T$ be the empirically best-performing arm output from* SuccessiveHalving *from the last round k of* Hyperband *of Figure 9 after exhausting a total budget of T from all rounds. Fix $\delta \in (0,1)$ and $q \in (1/K, 1)$ and let $z_q = \log(q^{-1})(\Delta_{\lceil \max\{2,qK\}\rceil}^{-\alpha} + \frac{1}{qK}\sum_{i=\lceil \max\{2,qK\}\rceil}^{K} \Delta_i^{-\alpha})$. Once $T = \widetilde{\Omega}\left(z_q \log(z_q)\log(1/\delta)\right)$ total pulls have been made by* Hyperband *we have $\widehat{\nu}_T - \nu_* \leq \Delta_{\lceil \max\{2,qK\}\rceil}$ with probability at least $1 - \delta$ where $\widetilde{\Omega}(\cdot)$ hides $\log\log(\cdot)$ factors.*

Appealing to the stochastic setting of Corollary 6 so that $\alpha = 2$, we conclude that the sample complexity sufficient to identify an arm within the best q proportion with probabiltiy $1 - \delta$, up to log factors, scales like $\log(1/\delta)\log(q^{-1})(\Delta_{\lceil qK\rceil}^{-\alpha} + \frac{1}{qK}\sum_{i=\lceil qK\rceil}^{K}\Delta_i^{-\alpha})$. One may interpret this result as an extension of the distribution-dependent pure-exploration results of Bubeck et al. (2009); but in our case, our bounds hold when the number of pulls is potentially much smaller than the number of arms K. When $q = 1/K$ this implies that the best arm is identified with about $\log(1/\delta)\log(K)\{\Delta_2^{-2} + \sum_{i=2}^{K}\Delta_i^{-2}\}$ which matches known upper bounds Karnin et al. (2013); Jamieson et al. (2014) and lower bounds Kaufmann et al. (2015) up to log factors. Thus, for the stochastic K-armed bandit problem Hyperband recovers many of the known sample complexity results up to log factors.

5.4 Finite Horizon Setting ($R < \infty$)

In this section we analyze the algorithm described in Section 3, i.e. finite horizon Hyperband. We present similar theoretical guarantees as in Section 5.3 for infinite horizon Hyperband, and fortunately much of the analysis will be recycled. We state the finite horizon version of the SuccessiveHalving and Hyperband algorithms in Figure 10.

The finite horizon setting differs in two major ways. First, in each bracket at least one arm will be pulled R times, but no arm will be pulled more than R times. Second, the number of brackets, each representing SuccessiveHalving with a different tradeoff between n and B, is fixed at $\log_\eta(R) + 1$. Hence, since we are sampling sequences randomly i.i.d., increasing B over time would just multiply the number of arms in each bracket by a constant, affecting performance only by a small constant.

Theorem 8 *Fix n arms. Let $\nu_i = \ell_{i,R}$ and assume $\nu_1 \leq \cdots \leq \nu_n$. For any $\epsilon > 0$ let*

$$z_{SH} = \eta(\log_\eta(R) + 1)\Big[n + \sum_{i=1}^{n} \min\left\{R, \gamma^{-1}\left(\max\left\{\tfrac{\epsilon}{4}, \tfrac{\nu_i - \nu_1}{2}\right\}\right)\right\}\Big]$$

If the Successive Halving algorithm of Figure 10 is run with any budget $B \geq z_{SH}$ then an arm $\hat{\imath}$ is returned that satisfies $\nu_{\hat{\imath}} - \nu_1 \leq \epsilon/2$.

Recall that $\gamma(R) = 0$ in this setting and by definition $\sup_{y \geq 0}\gamma^{-1}(y) \leq R$. Note that Lemma 2 still applies in this setting and just like above we obtain the following corollary.

Corollary 9 *Fix $\delta \in (0,1)$ and $\epsilon \geq 4(F^{-1}(\frac{\log(2/\delta)}{n}) - \nu_*)$. Let $\mathbf{H}(F,\gamma,n,\delta,\epsilon)$ be as defined in Lemma 2 and $B = \eta\log_\eta(R)(n + \max\{R, \mathbf{H}(F,\gamma,n,\delta,\epsilon)\})$. If the* SuccessiveHalving *algorithm of Figure 10 is run with the specified B and n arm configurations drawn randomly*

SuccessiveHalving (Finite horizon)

input: Budget B, and n arms where $\ell_{i,k}$ denotes the kth loss from the ith arm, maximum size R, $\eta \geq 2$ ($\eta = 3$ by default).

Initialize: $S_0 = [n]$, $s = \min\{t \in \mathbb{N} : nR(t+1)\eta^{-t} \leq B, t \leq \log_\eta(\min\{R, n\})\}$.

For $k = 0, 1, \ldots, s$

Set $n_k = \lfloor n\eta^{-k} \rfloor$, $r_k = \lfloor R\eta^{k-s} \rfloor$

Pull each arm in S_k for r_k times.

Keep the best $\lfloor n\eta^{-(k+1)} \rfloor$ arms in terms of the r_kth observed loss as S_{k+1}.

Output : $\hat{\imath}, \ell_{\hat{\imath},R}$ where $\hat{\imath} = \arg\min_{i \in S_{s+1}} \ell_{i,R}$

Hyperband (Finite horizon)

Input: Budget B, maximum size R, $\eta \geq 2$ ($\eta = 3$ by default)

Initialize: $s_{\max} = \lfloor \log(R)/\log(\eta) \rfloor$

For $k = 1, 2, \ldots$

For $s = s_{\max}, s_{\max} - 1, \ldots, 0$

$B_{k,s} = 2^k$, $n_{k,s} = \lceil \frac{2^k \eta^s}{R(s+1)} \rceil$

$\hat{\imath}_s, \ell_{\hat{\imath}_s,R} \leftarrow$ SuccessiveHalving $(B_{k,s}, n_{k,s}, R, \eta)$

Figure 10: The finite horizon SuccessiveHalving and Hyperband algorithms are inspired by their infinite horizon counterparts of Figure 9 to handle practical constraints. Hyperband calls SuccessiveHalving as a subroutine.

according to F then an arm $\hat{\imath} \in [n]$ is returned such that with probability at least $1 - \delta$ we have $\nu_{\hat{\imath}} - \nu_* \leq \left(F^{-1}(\frac{\log(2/\delta)}{n}) - \nu_*\right) + \epsilon/2$. In particular, if $B = 4\lceil \log_2(n) \rceil \mathbf{H}(F, \gamma, n, \delta)$ and $\epsilon = 4(F^{-1}(\frac{\log(2/\delta)}{n}) - \nu_*)$ then $\nu_{\hat{\imath}} - \nu_* \leq 3\left(F^{-1}(\frac{\log(2/\delta)}{n}) - \nu_*\right)$ with probability at least $1 - \delta$.

As in Section 5.3.2 we can apply the α, β parameterization for interpretability, with the added constraint that $\sup_{y \geq 0} \gamma^{-1}(y) \leq R$ so that $\gamma(j) \simeq \mathbf{1}_{j<R} \left(\frac{1}{j}\right)^{1/\alpha}$. Note that the approximate sample complexity of SuccessiveHalving given in Eq. (4) is still valid for the finite horizon algorithm.

Fixing some $\Delta > 0$, $\delta \in (0, 1)$, and applying the parameterization of Eq. (6) we recognize that if $n \simeq \Delta^{-\beta} \log(1/\delta)$ and the sufficient sampling budgets (treating η as an absolute constant) of the uniform allocation (Equation 3) and SuccessiveHalving (Eq. (4)) satisfy

$$\text{Uniform allocation} \qquad B \simeq R\Delta^{-\beta} \log(1/\delta)$$

$$\text{SuccessiveHalving} \qquad B \simeq \log(\Delta^{-1} \log(1/\delta)) \log(1/\delta) \left[R + \Delta^{-\beta} \frac{1 - (\alpha/\beta) R^{1-\beta/\alpha}}{1 - \alpha/\beta} \right]$$

then both also satisfy $\nu_{\hat{\imath}} - \nu_* \lesssim \Delta$ with probability at least $1 - \delta$. Recalling that a larger α means slower convergence and that a larger β means a greater difficulty of sampling a good limit, note that when $\alpha/\beta < 1$ the budget of SuccessiveHalving behaves like $R + \Delta^{-\beta} \log(1/\delta)$ but as $\alpha/\beta \to \infty$ the budget asymptotes to $R\Delta^{-\beta} \log(1/\delta)$.

We can also apply the discrete-CDF parameterization of Eq. (7). For any $q \in (0, 1)$, if $n \simeq q^{-1} \log(1/\delta)$ and the measurement budgets of the uniform allocation (Equation 3) and SuccessiveHalving (Equation 4) satisfy

$$\text{Uniform allocation:} \qquad B \simeq \log(1/\delta) \begin{cases} K \min\left\{R, \max_{j=2,\ldots,K} \Delta_j^{-\alpha}\right\} & \text{if } q = 1/K \\ q^{-1} \min\{R, \Delta_{\lceil qK \rceil}^{-\alpha}\} & \text{if } q > 1/K \end{cases}$$

SuccessiveHalving:

$$B \simeq \log(q^{-1} \log(1/\delta)) \log(1/\delta) \begin{cases} \min\{R, \Delta_2^{-\alpha}\} + \sum_{j=2}^{K} \min\{R, \Delta_j^{-\alpha}\} & \text{if } q = 1/K \\ \min\{R, \Delta_{\lceil qK \rceil}^{-\alpha}\} + \frac{1}{qK} \sum_{j=\lceil qK \rceil}^{K} \min\{R, \Delta_j^{-\alpha}\} & \text{if } q > 1/K \end{cases}$$

then an arm that is in the best q-fraction of arms is returned, i.e. $\hat{\imath}/K \approx q$ and $\nu_{\hat{\imath}} - \nu_* \lesssim \Delta_{\lceil \max\{2, qK\} \rceil}$, with probability at least $1 - \delta$. Once again we observe a stark difference between uniform allocation and SuccessiveHalving, particularly when $\Delta_j^{-\alpha} \ll R$ for many values of $j \in \{1, \ldots, n\}$.

Armed with Corollary 9, all of the discussion of Section 5.3.3 preceding Theorem 5 holds for the finite case ($R < \infty$) as well. Predictably analogous theorems also hold for the finite horizon setting, but their specific forms (with the polylog factors) provide no additional insights beyond the sample complexities sufficient for SuccessiveHalving to succeed, given immediately above.

It is important to note that in the finite horizon setting, for all sufficiently large B (e.g. $B > 3R$) and all distributions F, the budget B of SuccessiveHalving should scale *linearly* with $n \simeq \Delta^{-\beta} \log(1/\delta)$ as $\Delta \to 0$. Contrast this with the infinite horizon setting in which the ratio of B to n can become unbounded based on the values of α, β as $\Delta \to 0$. One consequence of this observation is that in the finite horizon setting it suffices to set B large enough to identify an Δ-good arm with just constant probability, say 1/10, and then repeat SuccessiveHalving m times to boost this constant probability to probability $1 - (\frac{9}{10})^m$. While in this theoretical treatment of Hyperband we grow B over time, in practice we recommend fixing B as a multiple of R as we have done in Section 3. The fixed budget version of finite horizon Hyperband is more suitable for practical application due to the constant time, instead of exponential time, between configurations trained to completion in each outer loop.

6. Conclusion

We conclude by discussing three potential extensions related to parallelizing Hyperband for distributed computing, adjusting for training methods with different convergence rates, and combining Hyperband with non-random sampling methods.

Distributed implementations. Hyperband has the potential to be parallelized since arms are independent and sampled randomly. The most straightforward parallelization scheme is to distribute individual brackets of SuccessiveHalving to different machines. This can be done asynchronously and as machines free up, new brackets can be launched

with a different set of arms. One can also parallelize a single bracket so that each round of SuccessiveHalving runs faster. One drawback of this method is that if R can be computed on one machine, the number of tasks decreases exponentially as arms are whittled down so a more sophisticated job priority queue must be managed. Devising parallel generalizations of Hyperband that efficiently leverage massive distributed clusters while minimizing overhead costs is an interesting avenue for future work.

Adjusting for different convergence rates. A second open challenge involves generalizing the ideas behind Hyperband to settings where configurations have drastically differing convergence rates. Configurations can have different convergence rates if they have hyperparameters that impact convergence (e.g., learning rate decay for SGD or neural networks with differing numbers of layers or hidden units), and/or if they correspond to different model families (e.g., deep networks versus decision trees). The core issue arises when configurations with drastically slower convergence rates ultimately result in better models. To address these issues, we should be able to adjust the resources allocated to each configuration so that a fair comparison can be made at the time of elimination.

Incorporating non-random sampling. Finally, Hyperband can benefit from different sampling schemes aside from simple random search. Quasi-random methods like Sobol or latin hypercube which were studied in Bergstra and Bengio (2012) may improve the performance of Hyperband by giving better coverage of the search space. Alternatively, meta-learning can be used to define intelligent priors informed by previous experimentation (Feurer et al., 2015). Finally, as mentioned in Section 2, exploring ways to combine Hyperband with adaptive configuration selection strategies is a very promising future direction.

Acknowledgments

KJ is supported by ONR awards N00014-15-1-2620 and N00014-13-1-0129. AT is supported in part by a Google Faculty Award and an AWS in Education Research Grant award.

Appendix A. Additional Experimental Results

Additional details for experiments presented in Section 3 and 4 are provided below.

A.1 LeNet Experiment

The search space for the LeNet example discussed in Section 3.3 is shown in Table 2.

Hyperparameter	Scale	Min	Max
Learning Rate	log	1e-3	1e-1
Batch size	log	1e1	1e3
Layer-2 Num Kernels (k2)	linear	10	60
Layer-1 Num Kernels (k1)	linear	5	k2

Table 2: Hyperparameter space for the LeNet application of Section 3.3. Note that the number of kernels in Layer-1 is upper bounded by the number of kernels in Layer-2.

A.2 Experiments Using Alex Krizhevsky's CNN Architecture

For the experiments discussed in Section 4.1, the exact architecture used is the 18% model provided on cuda-convnet for CIFAR-10.[14]

Search Space: The search space used for the experiments is shown in Table 3. The learning rate reductions hyperparameter indicates how many times the learning rate was reduced by a factor of 10 over the maximum iteration window. For example, on CIFAR-10, which has a maximum iteration of 30,000, a learning rate reduction of 2 corresponds to reducing the learning every 10,000 iterations, for a total of 2 reductions over the 30,000 iteration window. All hyperparameters, with the exception of the learning rate decay reduction, overlap with those in Snoek et al. (2012). Two hyperparameters in Snoek et al. (2012) were excluded from our experiments: (1) the width of the response normalization layer was excluded due to limitations of the Caffe framework and (2) the number of epochs was excluded because it is incompatible with dynamic resource allocation.

Data Splits: For CIFAR-10, the training (40,000 instances) and validation (10,000 instances) sets were sampled from data batches 1-5 with balanced classes. The original test set (10,000 instances) was used for testing. For MRBI, the training (10,000 instances) and validation (2,000 instances) sets were sampled from the original training set with balanced classes. The original test set (50,000 instances) was used for testing. Lastly, for SVHN, the train, validation, and test splits were created using the same procedure as that in Sermanet et al. (2012).

Comparison with Early-Stopping: Domhan et al. (2015) proposed an early-stopping method for neural networks and combined it with SMAC to speed up hyperparameter optimization. Their method stops training a configuration if the probability of the configuration beating the current best is below a specified threshold. This probability is estimated by extrapolating learning curves fit to the intermediate validation error losses of a configuration.

14. The model specification is available at `http://code.google.com/p/cuda-convnet/`.

Hyperparameter	Scale	Min	Max
Learning Parameters			
Initial Learning Rate	log	$5 * 10^{-5}$	5
Conv1 L_2 Penalty	log	$5 * 10^{-5}$	5
Conv2 L_2 Penalty	log	$5 * 10^{-5}$	5
Conv3 L_2 Penalty	log	$5 * 10^{-5}$	5
FC4 L_2 Penalty	log	$5 * 10^{-3}$	500
Learning Rate Reductions	integer	0	3
Local Response Normalization			
Scale	log	$5 * 10^{-6}$	5
Power	linear	0.01	3

Table 3: Hyperparameters and associated ranges for the three-layer convolutional network.

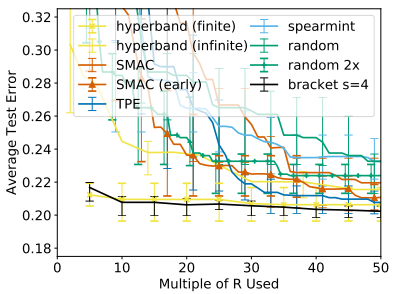
If a configuration is terminated early, the predicted terminal value from the estimated learning curves is used as the validation error passed to the hyperparameter optimization algorithm. Hence, if the learning curve fit is poor, it could impact the performance of the configuration selection algorithm. While this approach is heuristic in nature, it could work well in practice so we compare Hyperband to SMAC with early termination (labeled SMAC (early) in Figure 11). We used the conservative termination criterion with default parameters and recorded the validation loss every 400 iterations and evaluated the termination criterion 3 times within the training period (every 8k iterations for CIFAR-10 and MRBI and every 16k iterations for SVHN).[15] Comparing the performance by the number of total iterations as mulitple of R is conservative because it does not account for the time spent fitting the learning curve in order to check the termination criterion.

A.3 117 Data Sets Experiment

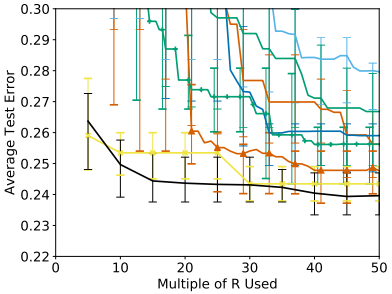
For the experiments discussed in Section 4.2.1, we followed Feurer et al. (2015) and imposed a 3GB memory limit, a 6-minute timeout for each hyperparameter configuration and a one-hour time window to evaluate each searcher on each data set. Moreover, we evaluated the performance of each searcher by aggregating results across all data sets and reporting the average rank of each method. Specifically, the hour training window is divided up into 30 second intervals and, at each time point, the model with the best validation error at that time is used in the calculation of the average error across all trials for each (data set-searcher) pair. Then, the performance of each searcher is ranked by data set and averaged across all data sets. All experiments were performed on Google Cloud Compute n1-standard-1 instances in us-central1-f region with 1 CPU and 3.75GB of memory.

Data Splits: Feurer et al. (2015) split each data set into 2/3 training and 1/3 test set, whereas we introduce a validation set to avoid overfitting to the test data. We also used 2/3 of the data for training, but split the rest of the data into two equally sized validation and test sets. We reported results on both the validation and test data. Moreover, we performed

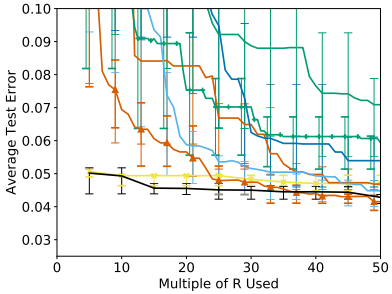
15. We used the code provided at https://github.com/automl/pylearningcurvepredictor.



(a) CIFAR-10



(b) MRBI



(c) SVHN

Figure 11: Average test error across 10 trials is shown in all plots. Error bars indicate the top and bottom quartiles of the test error corresponding to the model with the best validation error

20 trials of each (data set-searcher) pair, and as in Feurer et al. (2015) we kept the same data splits across trials, while using a different random seed for each searcher in each trial.

Shortcomings of the Experimental Setup: The benchmark contains a large variety of training set sizes and feature dimensions[16] resulting in random search being able to test 600 configurations on some data sets but just dozens on others. Hyperband was designed under the implicit assumption that computation scaled at least linearly with the data set size. For very small data sets that are trained in seconds, the initialization overheads dominate the computation and subsampling provides no computational benefit. In addition, many of the classifiers and preprocessing methods under consideration return memory errors as they require storage quadratic in the number of features (e.g., covariance matrix) or the number of observations (e.g., kernel methods). These errors usually happen immediately (thus wasting little time); however, they often occur on the full data set and not on subsampled data sets. A searcher like Hyperband that uses a subsampled data set could spend significant time training on a subsample only to error out when attempting to train it on the full data set.

A.4 Kernel Classification Experiments

Table 4 shows the hyperparameters and associated ranges considered in the kernel least squares classification experiment discussed in Section 4.2.2.

Hyperparameter	Type	Values
preprocessor	Categorical	min/max, standardize, normalize
kernel	Categorical	rbf, polynomial, sigmoid
C	Continuous	log $[10^{-3}, 10^{5}]$
gamma	Continuous	log $[10^{-5}, 10]$
degree	if kernel=poly	integer [2, 5]
coef0	if kernel=poly, sigmoid	uniform [-1.0, 1.0]

Table 4: Hyperparameter space for kernel regularized least squares classification problem discussed in Section 4.2.2.

The cost term C is divided by the number of samples so that the tradeoff between the squared error and the L_2 penalty would remain constant as the resource increased (squared error is summed across observations and not averaged). The regularization term λ is equal to the inverse of the scaled cost term C. Additionally, the average test error with the top and bottom quartiles across 10 trials are show in Figure 12.

Table 5 shows the hyperparameters and associated ranges considered in the random features kernel approximation classification experiment discussed in Section 4.3. The regularization term λ is divided by the number of features so that the tradeoff between the squared error and the L_2 penalty would remain constant as the resource increased. Additionally, the average test error with the top and bottom quartiles across 10 trials are show in Figure 13.

16. Training set size ranges from 670 to 73,962 observations, and number of features ranges from 1 to 10,935.

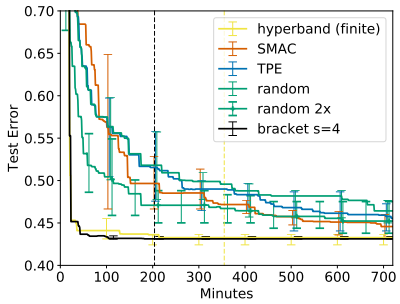


Figure 12: Average test error of the best kernel regularized least square classification model found by each searcher on CIFAR-10. The color coded dashed lines indicate when the last trial of a given searcher finished. Error bars correspond to the top and bottom quartiles of the test error across 10 trials.

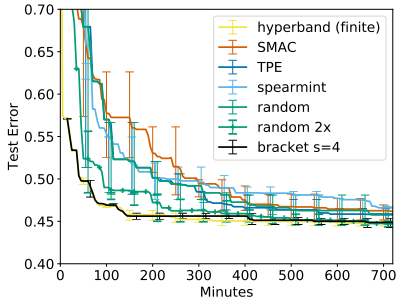


Figure 13: Average test error of the best random features model found by each searcher on CIFAR-10. The test error for Hyperband and bracket $s = 4$ are calculated in every evaluation instead of at the end of a bracket. Error bars correspond to the top and bottom quartiles of the test error across 10 trials.

Hyperparameter	Type	Values
preprocessor	Categorical	none, min/max, standardize, normalize
λ	Continuous	$\log\,[10^{-3}, 10^{5}]$
gamma	Continuous	$\log\,[10^{-5}, 10]$

Table 5: Hyperparameter space for random feature kernel approximation classification problem discussed in Section 4.3.

Appendix B. Proofs

In this section, we provide proofs for the theorems presented in Section 5.

B.1 Proof of Theorem 1

Proof First, we verify that the algorithm never takes a total number of samples that exceeds the budget B:

$$\sum_{k=0}^{\lceil \log_2(n)\rceil - 1} |S_k| \left\lfloor \frac{B}{|S_k|\lceil \log(n)\rceil} \right\rfloor \leq \sum_{k=0}^{\lceil \log_2(n)\rceil - 1} \frac{B}{\lceil \log(n)\rceil} \leq B\,.$$

For notational ease, let $\ell_{i,j} := \ell_j(X_i)$. Again, for each $i \in [n] := \{1, \ldots, n\}$ we assume the limit $\lim_{k\to\infty} \ell_{i,k}$ exists and is equal to ν_i. As a reminder, $\gamma : \mathbb{N} \to \mathbb{R}$ is defined as the pointwise smallest, monotonically decreasing function satisfying

$$\max_i |\ell_{i,j} - \nu_i| \leq \gamma(j)\ , \quad \forall j \in \mathbb{N}. \tag{8}$$

Note γ is guaranteed to exist by the existence of ν_i and bounds the deviation from the limit value as the sequence of iterates j increases.

Without loss of generality, order the terminal losses so that $\nu_1 \leq \nu_2 \leq \cdots \leq \nu_n$. Assume that $B \geq z_{SH}$. Then we have for each round k

$$\begin{aligned}
r_k &\geq \frac{B}{|S_k|\lceil \log_2(n)\rceil} - 1 \\
&\geq \frac{2}{|S_k|} \max_{i=2,\ldots,n} i\left(1 + \gamma^{-1}\big(\max\big\{\frac{\epsilon}{4}, \frac{\nu_i - \nu_1}{2}\big\}\big)\right) - 1 \\
&\geq \frac{2}{|S_k|}\left(\lfloor |S_k|/2\rfloor + 1\right)\left(1 + \gamma^{-1}\big(\max\big\{\frac{\epsilon}{4}, \frac{\nu_{\lfloor |S_k|/2\rfloor+1} - \nu_1}{2}\big\}\big)\right) - 1 \\
&\geq \left(1 + \gamma^{-1}\big(\max\big\{\frac{\epsilon}{4}, \frac{\nu_{\lfloor |S_k|/2\rfloor+1} - \nu_1}{2}\big\}\big)\right) - 1 \\
&= \gamma^{-1}\big(\max\big\{\frac{\epsilon}{4}, \frac{\nu_{\lfloor |S_k|/2\rfloor+1} - \nu_1}{2}\big\}\big),
\end{aligned}$$

where the fourth line follows from $\lfloor |S_k|/2\rfloor \geq |S_k|/2 - 1$.

First we show that $\ell_{i,t} - \ell_{1,t} > 0$ for all $t \geq \tau_i := \gamma^{-1}\big(\frac{\nu_i - \nu_1}{2}\big)$. Given the definition of γ, we have for all $i \in [n]$ that $|\ell_{i,t} - \nu_i| \leq \gamma(t) \leq \frac{\nu_i - \nu_1}{2}$ where the last inequality holds for $t \geq \tau_i$. Thus, for $t \geq \tau_i$ we have

$$\begin{aligned}
\ell_{i,t} - \ell_{1,t} &= \ell_{i,t} - \nu_i + \nu_i - \nu_1 + \nu_1 - \ell_{1,t} \\
&= \ell_{i,t} - \nu_i - (\ell_{1,t} - \nu_1) + \nu_i - \nu_1 \\
&\geq -2\gamma(t) + \nu_i - \nu_1 \\
&\geq -2\frac{\nu_i - \nu_1}{2} + \nu_i - \nu_1 \\
&= 0.
\end{aligned}$$

Under this scenario, we will eliminate arm i before arm 1 since on each round the arms are sorted by their empirical losses and the top half are discarded. Note that by the assumption the ν_i limits are non-decreasing in i so that the τ_i values are non-increasing in i.

Fix a round k and assume $1 \in S_k$ (note, $1 \in S_0$). The above calculation shows that

$$t \geq \tau_i \implies \ell_{i,t} \geq \ell_{1,t}. \tag{9}$$

Consequently,

$$\begin{aligned}
\{1 \in S_k,\ \ 1 \notin S_{k+1}\} &\iff \left\{\sum_{i\in S_k} \mathbf{1}\{\ell_{i,r_k} < \ell_{1,r_k}\} \geq \lfloor |S_k|/2\rfloor\right\} \\
&\implies \left\{\sum_{i\in S_k} \mathbf{1}\{r_k < \tau_i\} \geq \lfloor |S_k|/2\rfloor\right\} \\
&\implies \left\{\sum_{i=2}^{\lfloor |S_k|/2\rfloor+1} \mathbf{1}\{r_k < \tau_i\} \geq \lfloor |S_k|/2\rfloor\right\} \\
&\iff \left\{r_k < \tau_{\lfloor |S_k|/2\rfloor+1}\right\}.
\end{aligned}$$

where the first line follows by the definition of the algorithm, the second by Equation 9, and the third by τ_i being non-increasing (for all $i < j$ we have $\tau_i \geq \tau_j$ and consequently, $\mathbf{1}\{r_k < \tau_i\} \geq \mathbf{1}\{r_k < \tau_j\}$ so the *first* indicators of the sum not including 1 would be on before any other i's in $S_k \subset [n]$ sprinkled throughout $[n]$). This implies

$$\{1 \in S_k,\ \ r_k \geq \tau_{\lfloor |S_k|/2\rfloor+1}\} \implies \{1 \in S_{k+1}\}. \tag{10}$$

Recalling that $r_k \geq \gamma^{-1}\big(\max\big\{\frac{\epsilon}{4}, \frac{\nu_{\lfloor |S_k|/2\rfloor+1} - \nu_1}{2}\big\}\big)$ and $\tau_{\lfloor |S_k|/2\rfloor+1} = \gamma^{-1}\big(\frac{\nu_{\lfloor |S_k|/2\rfloor+1} - \nu_1}{2}\big)$, we examine the following three exhaustive cases:

- **Case 1:** $\frac{\nu_{\lfloor |S_k|/2\rfloor+1} - \nu_1}{2} \geq \frac{\epsilon}{4}$ and $1 \in S_k$

  In this case, $r_k \geq \gamma^{-1}\big(\frac{\nu_{\lfloor |S_k|/2\rfloor+1} - \nu_1}{2}\big) = \tau_{\lfloor |S_k|/2\rfloor+1}$. By Equation 10 we have that $1 \in S_{k+1}$ since $1 \in S_k$.

- **Case 2:** $\frac{\nu_{\lfloor |S_k|/2 \rfloor + 1} - \nu_1}{2} < \frac{\epsilon}{4}$ and $1 \in S_k$

  In this case $r_k \geq \gamma^{-1}\left(\frac{\epsilon}{4}\right)$ but $\gamma^{-1}\left(\frac{\epsilon}{4}\right) < \tau_{\lfloor |S_k|/2 \rfloor + 1}$. Equation 10 suggests that it may be possible for $1 \in S_k$ but $1 \notin S_{k+1}$. On the good event that $1 \in S_{k+1}$, the algorithm continues and on the next round either case 1 or case 2 could be true. So assume $1 \notin S_{k+1}$. Here we show that $\{1 \in S_k, \quad 1 \notin S_{k+1}\} \implies \max_{i \in S_{k+1}} \nu_i \leq \nu_1 + \epsilon/2$. Because $1 \in S_0$, this guarantees that SuccessiveHalving either exits with arm $\widehat{i} = 1$ or some arm $\widehat{i}$ satisfying $\nu_{\widehat{i}} \leq \nu_1 + \epsilon/2$.

  Let $p = \min\{i \in [n] : \frac{\nu_i - \nu_1}{2} \geq \frac{\epsilon}{4}\}$. Note that $p > \lfloor |S_k|/2 \rfloor + 1$ by the criterion of the case and

  $$r_k \geq \gamma^{-1}\left(\frac{\epsilon}{4}\right) \geq \gamma^{-1}\left(\frac{\nu_i - \nu_1}{2}\right) = \tau_i, \quad \forall i \geq p.$$

  Thus, by Equation 9 ($t \geq \tau_i \implies \ell_{i,t} \geq \ell_{1,t}$) we have that arms $i \geq p$ would always have $\ell_{i,r_k} \geq \ell_{1,r_k}$ and be eliminated before or at the same time as arm 1, presuming $1 \in S_k$. In conclusion, if arm 1 is eliminated so that $1 \in S_k$ but $1 \notin S_{k+1}$ then $\max_{i \in S_{k+1}} \nu_i \leq \max_{i<p} \nu_i < \nu_1 + \epsilon/2$ by the definition of p.

- **Case 3:** $1 \notin S_k$

  Since $1 \in S_0$, there exists some $r < k$ such that $1 \in S_r$ and $1 \notin S_{r+1}$. For this r, only case 2 is possible since case 1 would proliferate $1 \in S_{r+1}$. However, under case 2, if $1 \notin S_{r+1}$ then $\max_{i \in S_{r+1}} \nu_i \leq \nu_1 + \epsilon/2$.

Because $1 \in S_0$, we either have that 1 remains in S_k (possibly alternating between cases 1 and 2) for all k until the algorithm exits with the best arm 1, or there exists some k such that case 3 is true and the algorithm exits with an arm $\widehat{i}$ such that $\nu_{\widehat{i}} \leq \nu_1 + \epsilon/2$. The proof is complete by noting that

$$|\ell_{\widehat{i}, \lfloor \frac{B/2}{\lceil \log_2(n) \rceil} \rfloor} - \nu_1| \leq |\ell_{\widehat{i}, \lfloor \frac{B/2}{\lceil \log_2(n) \rceil} \rfloor} - \nu_{\widehat{i}}| + |\nu_{\widehat{i}} - \nu_1| \leq \epsilon/4 + \epsilon/2 \leq \epsilon$$

by the triangle inequality and because $B \geq 2\lceil \log_2(n) \rceil \gamma^{-1}(\epsilon/4)$ by assumption.

The second, looser, but perhaps more interpretable form of z_{SH} follows from the fact that $\gamma^{-1}(x)$ is non-increasing in x so that

$$\max_{i=2,\ldots,n} i\,\gamma^{-1}\left(\max\left\{\tfrac{\epsilon}{4}, \tfrac{\nu_i - \nu_1}{2}\right\}\right) \leq \sum_{i=1,\ldots,n} \gamma^{-1}\left(\max\left\{\tfrac{\epsilon}{4}, \tfrac{\nu_i - \nu_1}{2}\right\}\right).$$

■

B.2 Proof of Lemma 2

Proof Let $p_n = \frac{\log(2/\delta)}{n}$, $M = \gamma^{-1}\left(\frac{\epsilon}{16}\right)$, and $\mu = \mathbb{E}[\min\{M, \gamma^{-1}\left(\frac{\nu_i - \nu_*}{4}\right)\}]$. Define the events

$$\xi_1 = \{\nu_1 \leq F^{-1}(p_n)\}$$
$$\xi_2 = \left\{\sum_{i=1}^{n} \min\{M, \gamma^{-1}\left(\tfrac{\nu_i - \nu_*}{4}\right)\} \leq n\mu + \sqrt{2n\mu M \log(2/\delta)} + \tfrac{2}{3} M \log(2/\delta)\right\}$$

Note that $\mathbb{P}(\xi_1^c) = \mathbb{P}(\min_{i=1,\ldots,n} \nu_i > F^{-1}(p_n)) = (1-p_n)^n \leq \exp(-np_n) \leq \frac{\delta}{2}$. Moreover, $\mathbb{P}(\xi_2^c) \leq \frac{\delta}{2}$ by Bernstein's inequality since

$$\mathbb{E}\left[\min\{M, \gamma^{-1}\left(\tfrac{\nu_i - \nu_*}{4}\right)\}^2\right] \leq \mathbb{E}\left[M \min\{M, \gamma^{-1}\left(\tfrac{\nu_i - \nu_*}{4}\right)\}\right] = M\mu.$$

Thus, $\mathbb{P}(\xi_1 \cap \xi_2) \geq 1 - \delta$ so in what follows assume these events hold.

First we show that if $\nu_* \leq \nu_1 \leq F^{-1}(p_n)$, which we will refer to as equation $(*)$, then $\max\left\{\frac{\epsilon}{4}, \frac{\nu_i - \nu_1}{2}\right\} \geq \max\left\{\frac{\epsilon}{4}, \frac{\nu_i - \nu_*}{4}\right\}$.

Case 1: $\frac{\epsilon}{4} \leq \frac{\nu_i - \nu_1}{2}$ and $\epsilon \geq 4(F^{-1}(p_n) - \nu_*)$.

$$\frac{\nu_i - \nu_1}{2} \overset{(*)}{\geq} \frac{\nu_i - \nu_* + \nu_* - F^{-1}(p_n)}{2} = \frac{\nu_i - \nu_*}{4} + \frac{\nu_i - \nu_*}{4} - \frac{F^{-1}(p_n) - \nu_*}{2} \overset{(*)}{\geq} \frac{\nu_i - \nu_*}{4} + \frac{\nu_i - \nu_1}{4} - \frac{F^{-1}(p_n) - \nu_*}{2} \overset{\text{Case 1}}{\geq} \frac{\nu_i - \nu_*}{4}.$$

Case 2: $\frac{\epsilon}{4} > \frac{\nu_i - \nu_1}{2}$ and $\epsilon \geq 4(F^{-1}(p_n) - \nu_*)$.

$$\frac{\nu_i - \nu_*}{4} = \frac{\nu_i - \nu_1}{4} + \frac{\nu_1 - \nu_*}{4} \overset{\text{Case 2}}{<} \frac{\epsilon}{8} + \frac{\nu_1 - \nu_*}{4} \overset{(*)}{\leq} \frac{\epsilon}{8} + \frac{F^{-1}(p_n) - \nu_*}{4} \overset{\text{Case 2}}{<} \frac{\epsilon}{4}$$

which shows the desired result.

Consequently, for any $\epsilon \geq 4(F^{-1}(p_n) - \nu_*)$ we have

$$\begin{aligned}
\sum_{i=1}^{n} \gamma^{-1}\left(\max\left\{\tfrac{\epsilon}{4}, \tfrac{\nu_i - \nu_1}{2}\right\}\right) &\leq \sum_{i=1}^{n} \gamma^{-1}\left(\max\left\{\tfrac{\epsilon}{4}, \tfrac{\nu_i - \nu_*}{4}\right\}\right) \\
&\leq \sum_{i=1}^{n} \gamma^{-1}\left(\max\left\{\tfrac{\epsilon}{16}, \tfrac{\nu_i - \nu_*}{4}\right\}\right) \\
&= \sum_{i=1}^{n} \min\{M, \gamma^{-1}\left(\tfrac{\nu_i - \nu_*}{4}\right)\} \\
&\leq n\mu + \sqrt{2n\mu M \log(1/\delta)} + \tfrac{2}{3} M \log(1/\delta) \\
&\leq \left(\sqrt{n\mu} + \sqrt{\tfrac{2}{3} M \log(2/\delta)}\right)^2 \leq 2n\mu + \tfrac{4}{3} M \log(2/\delta).
\end{aligned}$$

A direct computation yields

$$\begin{aligned}
\mu &= \mathbb{E}[\min\{M, \gamma^{-1}\left(\tfrac{\nu_i - \nu_*}{4}\right)\}] \\
&= \mathbb{E}[\gamma^{-1}\left(\max\left\{\tfrac{\epsilon}{16}, \tfrac{\nu_i - \nu_*}{4}\right\}\right)] \\
&= \gamma^{-1}\left(\tfrac{\epsilon}{16}\right) F(\nu_* + \epsilon/4) + \int_{\nu_* + \epsilon/4}^{\infty} \gamma^{-1}(\tfrac{t - \nu_*}{4}) dF(t)
\end{aligned}$$

so that

$$\begin{aligned}
&\sum_{i=1}^{n} \gamma^{-1}\left(\max\left\{\tfrac{\epsilon}{4}, \tfrac{\nu_i - \nu_1}{2}\right\}\right) \leq 2n\mu + \tfrac{4}{3} M \log(2/\delta) \\
&\quad = 2n \int_{\nu_* + \epsilon/4}^{\infty} \gamma^{-1}(\tfrac{t - \nu_*}{4}) dF(t) + \left(\tfrac{4}{3} \log(2/\delta) + 2nF(\nu_* + \epsilon/4)\right) \gamma^{-1}\left(\tfrac{\epsilon}{16}\right)
\end{aligned}$$

which completes the proof. ■

B.3 Proof of Proposition 4

We break the proposition up into upper and lower bounds and prove them seperately.

B.4 Uniform Allocation

Proposition 10 *Suppose we draw n random configurations from F, train each with a budget of j,*[17] *and let $\hat{\imath} = \arg\min_{i=1,\dots,n} \ell_j(X_i)$. Let $\nu_i = \ell_*(X_i)$ and without loss of generality assume $\nu_1 \leq \ldots \leq \nu_n$. If*

$$B \geq n\gamma^{-1}\left(\tfrac{1}{2}(F^{-1}(\tfrac{\log(1/\delta)}{n}) - \nu_*)\right) \tag{11}$$

then with probability at least $1-\delta$ we have $\nu_{\hat{\imath}} - \nu_* \leq 2\left(F^{-1}\left(\frac{\log(1/\delta)}{n}\right) - \nu_*\right)$.

Proof Note that if we draw n random configurations from F and $i_* = \arg\min_{i=1,\dots,n} \ell_*(X_i)$ then

$$\begin{aligned}\mathbb{P}\left(\ell_*(X_{i_*}) - \nu_* \leq \epsilon\right) &= \mathbb{P}\Big(\bigcup_{i=1}^{n}\{\ell_*(X_i) - \nu_* \leq \epsilon\}\Big) \\ &= 1 - (1 - F(\nu_* + \epsilon))^n \geq 1 - e^{-nF(\nu_*+\epsilon)}\,,\end{aligned}$$

which is equivalent to saying that with probability at least $1-\delta$, $\ell_*(X_{i_*}) - \nu_* \leq F^{-1}(\log(1/\delta)/n) - \nu_*$. Furthermore, if each configuration is trained for j iterations then with probability at least $1-\delta$

$$\begin{aligned}&\ell_*(X_i) - \nu_* \leq \ell_j(X_i) - \nu_* + \gamma(j) \leq \ell_j(X_{i_*}) - \nu_* + \gamma(j) \\ &\leq \ell_*(X_{i_*}) - \nu_* + 2\gamma(j) \leq F^{-1}\left(\tfrac{\log(1/\delta)}{n}\right) - \nu_* + 2\gamma(j).\end{aligned}$$

If our measurement budget B is constrained so that $B = nj$ then solving for j in terms of B and n yields the result. ■

The following proposition demonstrates that the upper bound on the error of the uniform allocation strategy in Proposition 4 is in fact tight. That is, for any distribution F and function γ there exists a loss sequence that requires the budget described in Eq. (3) in order to avoid a loss of more than ϵ with high probability.

Proposition 11 *Fix any $\delta \in (0,1)$ and $n \in \mathbb{N}$. For any $c \in (0,1]$, let $\mathcal{F}_c$ denote the space of continuous cumulative distribution functions F satisfying*[18] $\inf_{x\in[\nu_*,1-\nu_*]} \inf_{\Delta\in[0,1-x]} \frac{F(x+\Delta)-F(x+\Delta/2)}{F(x+\Delta)-F(x)} \geq c$. *And let Γ denote the space of monotonically decreasing functions over $\mathbb{N}$. For any $F \in \mathcal{F}_c$ and $\gamma \in \Gamma$ there exists a probability distribution μ over $\mathcal{X}$ and a sequence of functions $\ell_j : \mathcal{X} \to \mathbb{R}$ $\forall j \in \mathbb{N}$ with $\ell_* := \lim_{j\to\infty} \ell_j$, $\nu_* = \inf_{x\in\mathcal{X}} \ell_*(x)$ such that $\sup_{x\in\mathcal{X}} |\ell_j(x) - \ell_*(x)| \leq \gamma(j)$ and $\mathbb{P}_\mu(\ell_*(X) - \nu_* \leq \epsilon) = F(\epsilon)$. Moreover, if n configurations $X_1, \dots, X_n$ are drawn from μ and $\hat{\imath} = \arg\min_{i\in1,\dots,n} \ell_{B/n}(X_i)$ then with probability at least δ*

$$\ell_*(X_{\hat{\imath}}) - \nu_* \geq 2(F^{-1}(\tfrac{\log(c/\delta)}{n+\log(c/\delta)}) - \nu_*)$$

whenever $B \leq n\gamma^{-1}\left(2(F^{-1}(\frac{\log(c/\delta)}{n+\log(c/\delta)}) - \nu_*)\right)$.

Proof Let $\mathcal{X} = [0,1]$, $\ell_*(x) = F^{-1}(x)$, and μ be the uniform distribution over $[0,1]$. Define $\widehat{\nu} = F^{-1}(\frac{\log(c/\delta)}{n+\log(c/\delta)})$ and set

$$\ell_j(x) = \begin{cases}\widehat{\nu} + \frac{1}{2}\gamma(j) + (\widehat{\nu} + \frac{1}{2}\gamma(j) - \ell_*(x)) & \text{if } |\widehat{\nu} + \frac{1}{2}\gamma(j) - \ell_*(x)| \leq \frac{1}{2}\gamma(j) \\ \ell_*(x) & \text{otherwise.}\end{cases}$$

Essentially, if $\ell_*(x)$ is within $\frac{1}{2}\gamma(j)$ of $\widehat{\nu} + \frac{1}{2}\gamma(j)$ then we set $\ell_j(x)$ equal to $\ell_*(x)$ reflected across the value $2\widehat{\nu} + \gamma(j)$. Clearly, $|\ell_j(x) - \ell_*(x)| \leq \gamma(j)$ for all $x \in \mathcal{X}$.

Since each $\ell_*(X_i)$ is distributed according to F, we have

$$\mathbb{P}\Big(\bigcap_{i=1}^{n}\{\ell_*(X_i) - \nu_* \geq \epsilon\}\Big) = (1 - F(\nu_* + \epsilon))^n \geq e^{-nF(\nu_*+\epsilon)/(1-F(\nu_*+\epsilon))}\,.$$

Setting the right-hand-side greater than or equal to δ/c and solving for ϵ, we find $\nu_* + \epsilon \geq F^{-1}(\frac{\log(c/\delta)}{n+\log(c/\delta)}) = \widehat{\nu}$.

Define $I_0 = [\nu_*, \widehat{\nu})$, $I_1 = [\widehat{\nu}, \widehat{\nu} + \frac{1}{2}\gamma(B/n))$ and $I_2 = [\widehat{\nu} + \frac{1}{2}\gamma(B/n), \widehat{\nu} + \gamma(B/n)]$. Furthermore, for $j \in \{0,1,2\}$ define $N_j = \sum_{i=1}^{n} \mathbf{1}_{\ell_*(X_i)\in I_j}$. Given $N_0 = 0$ (which occurs with probability at least δ/c), if $N_1 = 0$ then $\ell_*(X_i) - \nu_* \geq F^{-1}(\frac{\log(c/\delta)}{n+\log(c/\delta)}) + \frac{1}{2}\gamma(B/n)$ and the claim is true.

Below we will show that if $N_2 > 0$ whenever $N_1 > 0$ then the claim is also true. We now show that this happens with at least probability c whenever $N_1 + N_2 = m$ for any $m > 0$. Observe that

$$\begin{aligned}\mathbb{P}(N_2 > 0 | N_1 + N_2 = m) &= 1 - \mathbb{P}(N_2 = 0 | N_1 + N_2 = m) \\ &= 1 - (1 - \mathbb{P}(\nu_i \in I_2 | \nu_i \in I_1 \cup I_2))^m \geq 1 - (1-c)^m \geq c\end{aligned}$$

since

$$\mathbb{P}(\nu_i \in I_2 | \nu_i \in I_1 \cup I_2) = \frac{\mathbb{P}(\nu_i \in I_2)}{\mathbb{P}(\nu_i \in I_1 \cup I_2)} = \frac{\mathbb{P}(\nu_i \in [\widehat{\nu} + \frac{1}{2}\gamma, \widehat{\nu} + \gamma])}{\mathbb{P}(\nu_i \in [\widehat{\nu}, \widehat{\nu} + \gamma])} = \frac{F(\widehat{\nu} + \gamma) - F(\widehat{\nu} + \frac{1}{2}\gamma)}{F(\widehat{\nu} + \gamma) - F(\widehat{\nu})} \geq c\,.$$

Thus, the probability of the event that $N_0 = 0$ and $N_2 > 0$ whenever $N_1 > 0$ occurs with probability at least $\delta/c \cdot c = \delta$, so assume this is the case in what follows.

Since $N_0 = 0$, for all $j \in \mathbb{N}$, each X_i must fall into one of three cases:

1. $\ell_*(X_i) > \widehat{\nu} + \gamma(j) \iff \ell_j(X_i) > \widehat{\nu} + \gamma(j)$
2. $\widehat{\nu} \leq \ell_*(X_i) < \widehat{\nu} + \frac{1}{2}\gamma(j) \iff \widehat{\nu} + \frac{1}{2}\gamma(j) < \ell_j(X_i) \leq \widehat{\nu} + \gamma(j)$

17. Here j can be bounded (finite horizon) or unbounded (infinite horizon).

18. Note that this condition is met whenever F is convex. Moreover, if $F(\nu_* + \epsilon) = c_1^{-1}\epsilon^\beta$ then it is easy to verify that $c = 1 - 2^{-\beta} \geq \frac{1}{2}\min\{1, \beta\}$.

3. $\widehat{\nu} + \frac{1}{2}\gamma(j) \le \ell_*(X_i) \le \widehat{\nu} + \gamma(j) \iff \widehat{\nu} \le \ell_j(X_i) \le \widehat{\nu} + \frac{1}{2}\gamma(j)$

The first case holds since within that regime we have $\ell_j(x) = \ell_*(x)$, while the last two cases hold since they consider the regime where $\ell_j(x) = 2\widehat{\nu} + \gamma(j) - \ell_*(x)$. Thus, for any i such that $\ell_*(X_i) \in I_2$ it must be the case that $\ell_j(X_i) \in I_1$ and vice versa. Because $N_2 \ge N_1 > 0$, we conclude that if $\hat{\imath} = \arg\min_i \ell_{B/n}(X_i)$ then $\ell_{B/n}(X_{\hat{\imath}}) \in I_1$ and $\ell_*(X_{\hat{\imath}}) \in I_2$. That is, $\nu_{\hat{\imath}} - \nu_* \ge \widehat{\nu} - \nu_* + \frac{1}{2}\gamma(j) = F^{-1}(\frac{\log(c/\delta)}{n+\log(c/\delta)}) - \nu_* + \frac{1}{2}\gamma(j)$. So if we wish $\nu_{\hat{\imath}} - \nu_* \le 2(F^{-1}(\frac{\log(c/\delta)}{n+\log(c/\delta)}) - \nu_*)$ with probability at least δ then we require $B/n = j \ge \gamma^{-1}\left(2(F^{-1}(\frac{\log(c/\delta)}{n+\log(c/\delta)}) - \nu_*)\right)$. ■

B.5 Proof of Theorem 5

Proof Step 1: Simplify $\mathbf{H}(F,\gamma,n,\delta)$. We begin by simplifying $\mathbf{H}(F,\gamma,n,\delta)$ in terms of just n,δ,α,β. In what follows, we use a constant c that may differ from one inequality to the next but remains an absolute constant that depends on α,β only. Let $p_n = \frac{\log(2/\delta)}{n}$ so that

$$\gamma^{-1}\left(\tfrac{F^{-1}(p_n)-\nu_*}{4}\right) \le c\left(F^{-1}(p_n) - \nu_*\right)^{-\alpha} \le c\, p_n^{-\alpha/\beta}$$

and

$$\int_{p_n}^1 \gamma^{-1}(\tfrac{F^{-1}(t)-\nu_*}{4})dt \le c\int_{p_n}^1 t^{-\alpha/\beta}dt \le \begin{cases} c\log(1/p_n) & \text{if } \alpha=\beta \\ c\frac{1-p_n^{1-\alpha/\beta}}{1-\alpha/\beta} & \text{if } \alpha\neq\beta. \end{cases}$$

We conclude that

$$\begin{aligned}\mathbf{H}(F,\gamma,n,\delta) &= 2n\int_{p_n}^1 \gamma^{-1}(\tfrac{F^{-1}(t)-\nu_*}{4})dt + \tfrac{10}{3}\log(2/\delta)\gamma^{-1}\left(\tfrac{F^{-1}(p_n)-\nu_*}{4}\right) \\ &\le c p_n^{-\alpha/\beta}\log(1/\delta) + cn\begin{cases}\log(1/p_n) & \text{if } \alpha=\beta \\ \frac{1-p_n^{1-\alpha/\beta}}{1-\alpha/\beta} & \text{if } \alpha\neq\beta.\end{cases}\end{aligned}$$

Step 2: Solve for $(B_{k,l}, n_{k,l})$ in terms of Δ. Fix $\Delta > 0$. Our strategy is to describe $n_{k,l}$ in terms of Δ. In particular, parameterize $n_{k,l}$ such that $p_{n_{k,l}} = c\frac{\log(4k^3/\delta)}{n_{k,l}} = \Delta^\beta$ so that $n_{k,l} = c\Delta^{-\beta}\log(4k^3/\delta)$ so

$$\begin{aligned}\mathbf{H}(F,\gamma,n_{k,l},\delta_{k,l}) &\le c p_{n_{k,l}}^{-\alpha/\beta}\log(1/\delta_{k,l}) + cn_{k,l}\begin{cases}\log(1/p_{n_{k,l}}) & \text{if } \alpha=\beta \\ \frac{1-p_{n_{k,l}}^{1-\alpha/\beta}}{1-\alpha/\beta} & \text{if } \alpha\neq\beta.\end{cases} \\ &\le c\log(k/\delta)\left[\Delta^{-\alpha} + \begin{cases}\Delta^{-\beta}\log(\Delta^{-1}) & \text{if } \alpha=\beta \\ \frac{\Delta^{-\beta}-\Delta^{-\alpha}}{1-\alpha/\beta} & \text{if } \alpha\neq\beta\end{cases}\right] \\ &\le c\log(k/\delta)\min\{\tfrac{1}{|1-\alpha/\beta|}, \log(\Delta^{-1})\}\Delta^{-\max\{\beta,\alpha\}}\end{aligned}$$

where the last line follows from

$$\begin{aligned}\Delta^{\max\{\beta,\alpha\}}\frac{\Delta^{-\beta}-\Delta^{-\alpha}}{1-\alpha/\beta} &= \beta\frac{\Delta^{\max\{0,\alpha-\beta\}}-\Delta^{\max\{0,\beta-\alpha\}}}{\beta-\alpha} \\ &= \beta\begin{cases}\frac{1-\Delta^{\beta-\alpha}}{\beta-\alpha} & \text{if } \beta>\alpha \\ \frac{1-\Delta^{\alpha-\beta}}{\alpha-\beta} & \text{if } \beta<\alpha\end{cases} \le c\min\{\tfrac{1}{|1-\alpha/\beta|}, \log(\Delta^{-1})\}.\end{aligned}$$

Using the upperbound $\lceil\log(n_{k,l})\rceil \le c\log(\log(k/\delta)\Delta^{-1}) \le c\log(\log(k/\delta))\log(\Delta^{-1})$ and letting $z_\Delta = \log(\Delta^{-1})^2\Delta^{-\max\{\beta,\alpha\}}$, we conclude that

$$\begin{aligned}B_{k,l} &< \min\{2^k : 2^k > 4\lceil\log(n_{k,l})\rceil\mathbf{H}(F,\gamma,n_{k,l},\delta_{k,l})\} \\ &< \min\{2^k : 2^k > c\log(k/\delta)\log(\log(k/\delta))z_\Delta\} \\ &\le c z_\Delta \log(\log(z_\Delta)/\delta)\log(\log(\log(z_\Delta)/\delta)) \\ &= c z_\Delta \overline{\log}(\log(z_\Delta)/\delta).\end{aligned}$$

Step 3: Count the total number of measurements. Moreover, the total number of measurements before $\hat{\imath}_{k,l}$ is output is upperbounded by

$$T = \sum_{i=1}^k\sum_{j=l}^i B_{i,j} \le k\sum_{i=1}^k B_{i,1} \le 2kB_{k,1} = 2B_{k,1}\log_2(B_{k,1})$$

where we have employed the so-called "doubling trick": $\sum_{i=1}^k B_{i,1} = \sum_{i=1}^k 2^i \le 2^{k+1} = 2B_{k,\hat{\imath}}$. Simplifying,

$$T \le c z_\Delta\overline{\log}(\log(z_\Delta)/\delta)\overline{\log}(z_\Delta\log(\log(z_\Delta)/\delta)) \le c\Delta^{-\max\{\beta,\alpha\}}\overline{\log}(\Delta^{-1})^3\overline{\log}(\log(\Delta^{-1})/\delta)$$

Solving for Δ in terms of T obtains

$$\Delta = c\left(\frac{\overline{\log}(T)^3\overline{\log}(\log(T)/\delta)}{T}\right)^{1/\max\{\alpha,\beta\}}.$$

Because the output arm is just the empirical best, there is some error associated with using the empirical estimate. The arm returned returned on round (k,l) is pulled $\lfloor\frac{2^{k-1}}{l}\rfloor \gtrsim B_{k,l}/\log(B_{k,l})$ times so the possible error is bounded by $\gamma(B_{k,l}/\log(B_{k,l})) \le c\left(\frac{\log(B_{k,l})}{B_{k,l}}\right)^{1/\alpha} \le c\left(\frac{\log(B)^2\log(\log(B))}{B}\right)^{1/\alpha}$ which is dominated by the value of Δ solved for above. ■

B.6 Proof of Theorem 7

Proof Step 1: Simplify $\mathbf{H}(F,\gamma,n,\delta,\epsilon)$. We begin by simplifying $\mathbf{H}(F,\gamma,n,\delta,\epsilon)$ in terms of just n,δ,α,β. As before, we use a constant c that may differ from one inequality to the next but remains an absolute constant. Let $p_n = \frac{\log(2/\delta)}{n}$ First we solve for ϵ by noting that we identify the best arm if $\nu_{\hat{\imath}} - \nu_* < \Delta_2$. Thus, if $\nu_{\hat{\imath}} - \nu_* \le \left(F^{-1}(p_n) - \nu_*\right) + \epsilon/2$ then we set

$$\epsilon = \max\left\{2\left(\Delta_2 - \left(F^{-1}(p_n) - \nu_*\right)\right), 4\left(F^{-1}(p_n) - \nu_*\right)\right\}$$

so that

$$\nu_i - \nu_* \leq \max\left\{3\big(F^{-1}(p_n) - \nu_*\big), \Delta_2\right\} = \Delta_{\lceil \max\{2, cKp_n\}\rceil}.$$

We treat the case when $3\big(F^{-1}(p_n) - \nu_*\big) \leq \Delta_2$ and the alternative separately.

First assume $3\big(F^{-1}(p_n) - \nu_*\big) > \Delta_2$ so that $\epsilon = 4\big(F^{-1}(p_n) - \nu_*\big)$ and $\mathbf{H}(F, \gamma, n, \delta, \epsilon) = \mathbf{H}(F, \gamma, n, \delta)$. We also have

$$\gamma^{-1}\left(\tfrac{F^{-1}(p_n)-\nu_*}{4}\right) \leq c\left(F^{-1}\left(p_n\right) - \nu_*\right)^{-\alpha} \leq c\,\Delta^{-\alpha}_{\lceil p_n K\rceil}$$

and

$$\int_{p_n}^1 \gamma^{-1}(\tfrac{F^{-1}(t)-\nu_*}{4})dt = \int_{F^{-1}(p_n)}^1 \gamma^{-1}(\tfrac{x-\nu_*}{4})dF(x) \leq \frac{c}{K}\sum_{i=\lceil p_n K\rceil}^K \Delta_i^{-\alpha}$$

so that

$$\begin{aligned}\mathbf{H}(F, \gamma, n, \delta) &= 2n\int_{p_n}^1 \gamma^{-1}(\tfrac{F^{-1}(t)-\nu_*}{4})dt + \tfrac{10}{3}\log(2/\delta)\gamma^{-1}\left(\tfrac{F^{-1}(p_n)-\nu_*}{4}\right)\\ &\leq c\Delta^{-\alpha}_{\lceil p_n K\rceil}\log(1/\delta) + \frac{cn}{K}\sum_{i=\lceil p_n K\rceil}^K \Delta_i^{-\alpha}.\end{aligned}$$

Now consider the case when $3\big(F^{-1}(p_n) - \nu_*\big) \leq \Delta_2$. In this case $F(\nu_* + \epsilon/4) = 1/K$, $\gamma^{-1}\left(\frac{\epsilon}{16}\right) \leq c\Delta_2^{-\alpha}$, and $\int_{\nu_*+\epsilon/4}^\infty \gamma^{-1}(\frac{t-\nu_*}{4})dF(t) \leq c\sum_{i=2}^K \Delta_i^{-\alpha}$ so that

$$\begin{aligned}\mathbf{H}(F, \gamma, n, \delta, \epsilon) &= 2n\int_{\nu_*+\epsilon/4}^\infty \gamma^{-1}(\tfrac{t-\nu_*}{4})dF(t) + \left(\tfrac{4}{3}\log(2/\delta) + 2nF(\nu_* + \epsilon/4)\right)\gamma^{-1}\left(\tfrac{\epsilon}{16}\right)\\ &\leq c(\log(1/\delta) + n/K)\Delta_2^{-\alpha} + \frac{cn}{K}\sum_{i=2}^K \Delta_i^{-\alpha}.\end{aligned}$$

Step 2: Solve for $(B_{k,l}, n_{k,l})$ in terms of Δ. Note there is no improvement possible once $p_{n_{k,l}} \leq 1/K$ since $3\big(F^{-1}(1/K) - \nu_*\big) \leq \Delta_2$. That is, when $p_{n_{k,l}} \leq 1/K$ the algorithm has found the best arm but will continue to take samples indefinetely. Thus, we only consider the case when $q = 1/K$ and $q > 1/K$. Fix $\Delta > 0$. Our strategy is to describe $n_{k,l}$ in terms of q. In particular, parameterize $n_{k,l}$ such that $p_{n_{k,l}} = c\frac{\log(4k^3/\delta)}{n_{k,l}} = q$ so that $n_{k,l} = cq^{-1}\log(4k^3/\delta)$ so

$$\begin{aligned}\mathbf{H}(F, \gamma, n_{k,l}, \delta_{k,l}, \epsilon_{k,l}) &\leq c\begin{cases}(\log(1/\delta_{k,l}) + \frac{n_{k,l}}{K})\Delta_2^{-\alpha} + \frac{n_{k,l}}{K}\sum_{i=2}^K \Delta_i^{-\alpha} & \text{if } 5\big(F^{-1}(p_{n_{k,l}}) - \nu_*\big) \leq \Delta_2\\ \Delta^{-\alpha}_{\lceil p_{n_{k,l}} K\rceil}\log(1/\delta_{k,l}) + \frac{n_{k,l}}{K}\sum_{i=\lceil p_{n_{k,l}} K\rceil}^K \Delta_i^{-\alpha} & \text{if otherwise}\end{cases}\\ &\leq c\log(k/\delta)\begin{cases}\Delta_2^{-\alpha} + \sum_{i=2}^K \Delta_i^{-\alpha} & \text{if } q = 1/K\\ \Delta^{-\alpha}_{\lceil qK\rceil} + \frac{1}{qK}\sum_{i=\lceil qK\rceil}^K \Delta_i^{-\alpha} & \text{if } q > 1/K.\end{cases}\\ &\leq c\log(k/\delta)\Delta^{-\alpha}_{\lceil \max\{2, qK\}\rceil} + \frac{1}{qK}\sum_{i=\lceil \max\{2, qK\}\rceil}^K \Delta_i^{-\alpha}\end{aligned}$$

Using the upperbound $\lceil\log(n_{k,l})\rceil \leq c\log(\log(k/\delta)q^{-1}) \leq c\log(\log(k/\delta))\log(q^{-1})$ and letting $z_q = \log(q^{-1})(\Delta^{-\alpha}_{\lceil \max\{2,qK\}\rceil} + \frac{1}{qK}\sum_{i=\lceil \max\{2,qK\}\rceil}^K \Delta_i^{-\alpha})$, we apply the exact sequence of steps as in the proof of Theorem 5 to obtain

$$T \leq cz_q\overline{\log}(\log(z_q)/\delta)\overline{\log}(z_q\log(\log(z_q)/\delta))$$

Because the output arm is just the empirical best, there is some error associated with using the empirical estimate. The arm returned on round (k, l) is pulled $\lceil\frac{2^{k-1}}{l}\rceil \geq cB_{k,l}/\log(B_{k,l})$ times so the possible error is bounded by $\gamma(B_{k,l}/\log(B_{k,l})) \leq c\left(\frac{\log(B_{k,l})}{B_{k,l}}\right)^{1/\alpha} \leq c\left(\frac{\log(T)^2\log(\log(T))}{T}\right)^{1/\alpha}$. This is dominated by $\Delta_{\lceil \max\{2,qK\}\rceil}$ for the value of T prescribed by the above calculation, completing the proof. ■

B.7 Proof of Theorem 8

Proof Let s denote the index of the last stage, to be determined later. If $\widetilde{r}_k = R\eta^{k-s}$ and $\widetilde{n}_k = n\eta^{-k}$ so that $r_k = \lfloor\widetilde{r}_k\rfloor$ and $n_k = \lfloor\widetilde{n}_k\rfloor$ then

$$\sum_{k=0}^s n_k r_k \leq \sum_{k=0}^s \widetilde{n}_k\widetilde{r}_k = nR(s+1)\eta^{-s} \leq B$$

since, by definition, $s = \min\{t \in \mathbb{N} : nR(t+1)\eta^{-t} \leq B, t \leq \log_\eta(\min\{R, n\})\}$. It is straightforward to verify that $B \geq z_{SH}$ ensures that $r_0 \geq 1$ and $n_s \geq 1$.

The proof for Theorem 1 holds here with a few modifications. First, we derive a lower bound on the resource per arm r_k per round if $B \geq z_{SH}$ with generalized elimination rate η:

$$\begin{aligned}r_k &\geq \frac{B}{|S_k|(\lfloor\log_\eta(n)\rfloor + 1)} - 1\\ &\geq \frac{\eta}{|S_k|}\max_{i=2,\ldots,n} i\left(1 + \min\left\{R, \gamma^{-1}\big(\max\left\{\frac{\epsilon}{4}, \frac{\nu_i - \nu_1}{2}\right\}\big)\right\}\right) - 1\\ &\geq \frac{\eta}{|S_k|}\left(\lfloor|S_k|/\eta\rfloor + 1\right)\left(1 + \min\left\{R, \gamma^{-1}\big(\max\left\{\frac{\epsilon}{4}, \frac{\nu_{\lfloor|S_k|/2\rfloor+1} - \nu_1}{2}\right\}\big)\right\}\right) - 1\\ &\geq \left(1 + \min\left\{R, \gamma^{-1}\big(\max\left\{\frac{\epsilon}{4}, \frac{\nu_{\lfloor|S_k|/2\rfloor+1} - \nu_1}{2}\right\}\big)\right\}\right) - 1\\ &= \min\left\{R, \gamma^{-1}\big(\max\left\{\frac{\epsilon}{4}, \frac{\nu_{\lfloor|S_k|/2\rfloor+1} - \nu_1}{2}\right\}\big)\right\}.\end{aligned}$$

Also, note that $\gamma(R) = 0$, hence if the minimum is ever active, $\ell_{i,R} = \nu_i$ and we know the true loss. The rest of the proof is same as that for Theorem 1 for η in place of 2.

In addition, we note that

$$\max_{i=n_s+1,\ldots,n} i\,\gamma^{-1}\left(\max\left\{\tfrac{\epsilon}{4}, \tfrac{\nu_i-\nu_1}{2}\right\}\right) \leq n_s\gamma^{-1}\left(\max\left\{\tfrac{\epsilon}{4}, \tfrac{\nu_{n_s+1}-\nu_1}{2}\right\}\right) + \sum_{i>n_s}\gamma^{-1}\left(\max\left\{\tfrac{\epsilon}{4}, \tfrac{\nu_i-\nu_1}{2}\right\}\right).$$

■

References

A. Agarwal, J. Duchi, P. L. Bartlett, and C. Levrard. Oracle inequalities for computationally budgeted model selection. In *Conference On Learning Theory (COLT)*, 2011.

A. Agarwal, S. Kakade, N. Karampatziakis, L. Song, and G. Valiant. Least squares revisited: Scalable approaches for multi-class prediction. In *International Conference on Machine Learning (ICML)*, pages 541–549, 2014.

J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

J. Bergstra, R. Bardenet, Y. Bengio, and B. Kegl. Algorithms for hyper-parameter optimization. In *Neural Information Processing Systems (NIPS)*, 2011.

S. Bubeck, R. Munos, and G. Stoltz. Pure exploration in multi-armed bandits problems. In *International Conference on Algorithmic Learning Theory (ALT)*, 2009.

S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvari. X-armed bandits. *Journal of Machine Learning Research*, 12:1655–1695, 2011.

A. Carpentier and M. Valko. Simple regret for infinitely many armed bandits. In *International Conference on Machine Learning (ICML)*, 2015.

E. Contal, V. Perchet, and N. Vayatis. Gaussian process optimization with mutual information. In *International Conference on Machine Learning (ICML)*, 2014.

T. Domhan, J. T. Springenberg, and F. Hutter. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

K. Eggensperger et al. Towards an empirical foundation for assessing Bayesian optimization of hyperparameters. In *Neural Information Processing Systems (NIPS) Bayesian Optimization Workshop*, 2013.

E. Even-Dar, S. Mannor, and Y. Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research*, 7:1079–1105, 2006.

M. Feurer. Personal communication, 2015.

M. Feurer, J. Springenberg, and F. Hutter. Using meta-learning to initialize Bayesian optimization of hyperparameters. In *ECAI Workshop on Meta-Learning and Algorithm Selection*, 2014.

M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter. Efficient and robust automated machine learning. In *Neural Information Processing Systems (NIPS)*, 2015.

D. Golovin, B. Sonik, S. Moitra, G. Kochanski, J. Karro, and D.Sculley. Google Vizier: A service for black-box optimization. In *Knowledge Discovery and Data Mining (KDD)*, 2017.

S. Grünewälder, J. Audibert, M. Opper, and J. Shawe–Taylor. Regret bounds for Gaussian process bandit problems. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

A. György and L. Kocsis. Efficient multi-start strategies for local search algorithms. *Journal of Artificial Intelligence Research*, 41:407–444, 2011.

F. Hutter, H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *International Conference on Learning and Intelligent Optimization (LION)*, 2011.

K. Jamieson and R. Nowak. Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In *Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE, 2014.

K. Jamieson and A. Talwalkar. Non-stochastic best arm identification and hyperparameter optimization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

K. Jamieson, M. Malloy, R. Nowak, and S. Bubeck. lil'UCB: An optimal exploration algorithm for multi-armed bandits. In *Conference On Learning Theory (COLT)*, pages 423–439, 2014.

K. G. Jamieson, D. Haas, and B. Recht. The power of adaptivity in identifying statistical alternatives. In *Neural Information Processing Systems (NIPS)*, pages 775–783, 2016.

K. Kandasamy, J. Schneider, and B. Póczos. High dimensional Bayesian optimization and bandits via additive models. In *International Conference on Machine Learning (ICML)*, 2015.

K. Kandasamy, G. Dasarathy, J. B. Oliva, J. G. Schneider, and B. Póczos. Gaussian process bandit optimization with multi-fidelity evaluations. In *Neural Information Processing Systems (NIPS)*, 2016.

K. Kandasamy, G. Dasarathy, J. Schneider, and B. Póczos. Multi-fidelity Bayesian optimization with continuous approximation. In *International Conference on Machine Learning (ICML)*, 2017.

Z. Karnin, T. Koren, and O. Somekh. Almost optimal exploration in multi-armed bandits. In *International Conference on Machine Learning (ICML)*, 2013.

E. Kaufmann, O. Cappé, and A. Garivier. On the complexity of best arm identification in multi-armed bandit models. *Journal of Machine Learning Research*, 16:1–42, 2015.

A. Klein, S. Falkner, S. Bartels, P. Hennig, and F. Hutter. Fast Bayesian optimization of machine learning hyperparameters on large datasets. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017a.

A. Klein, S. Falkner, J. T. Springenberg, and F. Hutter. Learning curve prediction with Bayesian neural networks. In *International Conference On Learning Representation (ICLR)*, 2017b.

A. Krizhevsky. Learning multiple layers of features from tiny images. In *Technical report, Department of Computer Science, Univsersity of Toronto*, 2009.

T. Krueger, D. Panknin, and M. Braun. Fast cross-validation via sequential testing. *Journal of Machine Learning Research*, 16:1103–1155, 2015.

H. Larochelle et al. An empirical evaluation of deep architectures on problems with many factors of variation. In *International Conference on Machine Learning (ICML)*, 2007.

L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar. Hyperband: Bandit-based configuration evaluation for hyperparameter optimization. In *International Conference On Learning Representation (ICLR)*, 2017.

O. Maron and A. Moore. The racing algorithm: Model selection for lazy learners. *Artificial Intelligence Review*, 11:193–225, 1997.

V. Mnih and J.-Y. Audibert. Empirical Bernstein stopping. In *International Conference on Machine Learning (ICML)*, 2008.

Y. Netzer et al. Reading digits in natural images with unsupervised feature learning. In *Neural Information Processing Systems (NIPS) Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Neural Information Processing Systems (NIPS)*, 2007.

R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.

P. Sermanet, S. Chintala, and Y. LeCun. Convolutional neural networks applied to house numbers digit classification. In *International Conference on Pattern Recognition (ICPR)*, 2012.

J. Snoek, H. Larochelle, and R. Adams. Practical Bayesian optimization of machine learning algorithms. In *Neural Information Processing Systems (NIPS)*, 2012.

J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M. Patwary, Prabhat, and R. Adamst. Scalable Bayesian optimization using deep neural networks. In *International Conference on Machine Learning (ICML)*, 2015a.

J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M. Patwary, M. Prabhat, and R. Adams. Bayesian optimization using deep neural networks. In *International Conference on Machine Learning (ICML)*, 2015b.

E. Sparks, A. Talwalkar, D. Haas, M. J. Franklin, M. I. Jordan, and T. Kraska. Automating model search for large scale machine learning,. In *ACM Symposium on Cloud Computing (SOCC)*, 2015.

J. Springenberg, A. Klein, S. Falkner, and F. Hutter. Bayesian optimization with robust Bayesian neural networks. In *Neural Information Processing Systems (NIPS)*, 2016.

N. Srinivas, A. Krause, M. Seeger, and S. M. Kakade. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning (ICML)*, 2010.

K. Swersky, J. Snoek, and R. Adams. Multi-task Bayesian optimization. In *Neural Information Processing Systems (NIPS)*, 2013.

K. Swersky, J. Snoek, and R. P. Adams. Freeze-thaw Bayesian optimization. *arXiv preprint arXiv:1406.3896*, 2014.

C. Thornton et al. Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In *Knowledge Discovery and Data Mining (KDD)*, 2013.

A. van der Vaart and H. van Zanten. Information rates of nonparametric Gaussian process methods. *Journal of Machine Learning Research*, 12:2095–2119, 2011.

Z. Wang, M. Zoghi, F. Hutter, D. Matheson, and N. de Freitas. Bayesian optimization in high dimensions via random embeddings. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2013.

Z. Wang, B. Zhou, and S. Jegelka. Optimization as estimation with Gaussian processes in bandit settings. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.

A. G. Wilson, C. Dann, and H. Nickisch. Thoughts on massively scalable Gaussian processes. *arXiv:1511.01870*, 2015.

Journal of Machine Learning Research 18 (2018) 1-52 Submitted 6/17; Revised 2/18; Published 4/18

Submatrix localization via message passing

Bruce Hajek B-HAJEK@ILLINOIS.EDU
Department of ECE and Coordinated Science Lab
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA

Yihong Wu YIHONG.WU@YALE.EDU
Department of Statistics and Data Science
Yale University
New Haven, CT 06520, USA

Jiaming Xu XU972@PURDUE.EDU
Krannert School of Management
Purdue University
West Lafayette, IN 47907, USA

Editor: Qiang Liu

Abstract

The principal submatrix localization problem deals with recovering a $K \times K$ principal submatrix of elevated mean μ in a large $n \times n$ symmetric matrix subject to additive standard Gaussian noise, or more generally, mean zero, variance one, subgaussian noise. This problem serves as a prototypical example for community detection, in which the community corresponds to the support of the submatrix. The main result of this paper is that in the regime $\Omega(\sqrt{n}) \le K \le o(n)$, the support of the submatrix can be weakly recovered (with $o(K)$ misclassification errors on average) by an optimized message passing algorithm if $\lambda = \mu^2 K^2/n$, the signal-to-noise ratio, exceeds $1/e$. This extends a result by Deshpande and Montanari previously obtained for $K = \Theta(\sqrt{n})$ and $\mu = \Theta(1)$. In addition, the algorithm can be combined with a voting procedure to achieve the information-theoretic limit of exact recovery with sharp constants for all $K \ge \frac{n}{\log n}(\frac{1}{8e} + o(1))$. The total running time of the algorithm is $O(n^2 \log n)$.

Another version of the submatrix localization problem, known as noisy biclustering, aims to recover a $K_1 \times K_2$ submatrix of elevated mean μ in a large $n_1 \times n_2$ Gaussian matrix. The optimized message passing algorithm and its analysis are adapted to the bicluster problem assuming $\Omega(\sqrt{n_i}) \le K_i \le o(n_i)$ and $K_1 \asymp K_2$. A sharp information-theoretic condition for the weak recovery of both clusters is also identified.

Keywords: Submatrix localization, biclustering, message passing, spectral algorithms computational complexity, high-dimensional statistics

1. Introduction

The problem of *submatrix detection* and *localization*, also known as *noisy biclustering* (Hartigan, 1972; Shabalin et al., 2009; Kolar et al., 2011; Butucea and Ingster, 2013; Butucea et al., 2015; Ma and Wu, 2015; Chen and Xu, 2014; Cai et al., 2017), deals with finding a submatrix with an elevated mean in a large noisy matrix, which arises in many applications such as social network analysis and gene expression data analysis. A widely studied statistical model is the following:

$$W = \mu \mathbf{1}_{C_1^*} \mathbf{1}_{C_2^*}^\top + Z, \tag{1}$$

where $\mu > 0$, $\mathbf{1}_{C_1^*}$ and $\mathbf{1}_{C_2^*}$ are indicator vectors of the row and column support sets $C_1^* \subset [n_1]$ and $C_2^* \subset [n_2]$ of cardinality K_1 and K_2, respectively, and Z is an $n_1 \times n_2$ matrix consisting of independent standard normal entries. The objective is to accurately locate the submatrix by estimating the row and column support based on the large matrix W.

For simplicity we start by considering the symmetric version of this problem, namely, locating a principal submatrix, and later extend our theoretic and algorithmic findings to the asymmetric case. To this end, consider

$$W = \mu \mathbf{1}_{C^*} \mathbf{1}_{C^*}^\top + Z, \tag{2}$$

where $C^* \subset [n]$ has cardinality K and Z is an $n \times n$ symmetric matrix with $\{Z_{ij}\}_{1 \le i \le j \le n}$ being mutually independent standard normal. Given the data matrix W, the problem of interest is to recover C^*. This problem has been investigated in (Deshpande and Montanari, 2015; Montanari et al., 2015; Hajek et al., 2017) as a prototypical example of the *hidden community problem*,[1] because the distribution of the entries exhibits a community structure, namely, $W_{i,j} \sim \mathcal{N}(\mu, 1)$ if both i and j belong to C^* and $W_{i,j} \sim \mathcal{N}(0, 1)$ if otherwise.

Assuming that C^* is drawn from all subsets of $[n]$ of cardinality K uniformly at random, we focus on the following two types of recovery guarantees.[2] Let $\xi = \mathbf{1}_{C^*} \in \{0,1\}^n$ denote the indicator of the community. Let $\widehat{\xi} = \widehat{\xi}(A) \in \{0,1\}^n$ be an estimator.

- We say that $\widehat{\xi}$ *exactly recovers* ξ if, as $n \to \infty$, $\mathbb{P}[\xi \ne \widehat{\xi}] \to 0$.
- We say that $\widehat{\xi}$ *weakly recovers* ξ if, as $n \to \infty$, $d(\xi, \widehat{\xi})/K \to 0$ in probability, where d denotes the Hamming distance.

The weak recovery guarantee is phrased in terms of convergence in probability, which turns out to be equivalent to convergence in mean. Indeed, the existence of an estimator satisfying $d(\xi, \widehat{\xi})/K \to 0$ is equivalent to the existence of an estimator such that $\mathbb{E}[d(\xi, \widehat{\xi})] = o(K)$ (see (Hajek et al., 2017, Appendix A) for a proof). Clearly, any estimator achieving exact recovery also achieves weak recovery; for bounded K, these two criteria are equivalent.

Intuitively, for a fixed matrix size n, as either the submatrix size K or the signal strength μ decreases, it becomes more difficult to locate the submatrix. A key role is played by the parameter

$$\lambda = \frac{\mu^2 K^2}{n},$$

which is the signal-to-noise ratio for classifying an index i according to the statistic $\sum_j W_{i,j}$, which is distributed according to $\mathcal{N}(\mu K, n)$ if $i \in C^*$ and $\mathcal{N}(0, n)$ if $i \notin C^*$. As shown in

1. A slight variation of the model in (Deshpande and Montanari, 2015; Hajek et al., 2017) is that the data matrix therein is assumed to have zero diagonal. As shown in (Hajek et al., 2017), the absence of the diagonal has no impact on the statistical limit of the problem as long as $K \to \infty$, which is the case considered in the present paper.
2. Exact and weak recovery are called strong consistency and weak consistency in (Amini et al., 2013; Mossel et al., 2015), respectively.

Appendix A, it turns out that if the submatrix size K grows linearly with n, the information-theoretic limits[3] of both weak and exact recovery are easily attainable via thresholding. To see this, note that in the case of $K \asymp n$ simply thresholding the row sums can provide weak recovery in $O(n^2)$ time provided that $\lambda \to \infty$, which coincides with the information-theoretic conditions of weak recovery as proved in (Hajek et al., 2017). Moreover, in this case, one can show that this thresholding algorithm followed by a linear-time voting procedure achieves exact recovery whenever information-theoretically possible. Thus, this paper concentrates on weak and exact recovery in the sublinear regime of

$$\Omega(\sqrt{n}) \le K \le o(n). \tag{3}$$

We show that an optimized message passing algorithm provides weak recovery in nearly linear – $O(n^2 \log n)$ – time if $\lambda > 1/e$. This extends the sufficient conditions obtained in (Deshpande and Montanari, 2015) for the regime $K = \Theta(\sqrt{n})$ and $\mu = \Theta(1)$.[4] Our algorithm is the same as the message passing algorithm proposed in (Deshpande and Montanari, 2015), except that we find the polynomial that maximizes the signal-to-noise ratio via Hermite polynomials instead of using the truncated Taylor series as in (Deshpande and Montanari, 2015). The proofs follow closely those in (Deshpande and Montanari, 2015), with the most essential differences described at the end of Section 2.

We observe that $\lambda > 1/e$ is much more stringent than $\lambda > \frac{4K}{n} \log \frac{n}{K}$, the information-theoretic weak recovery threshold established in (Hajek et al., 2017). It is an open problem whether any polynomial-time algorithm can provide weak recovery for $\lambda \le 1/e$. In addition, we show that if $\lambda > 1/e$, the message passing algorithm followed by a linear-time voting procedure can provide exact recovery whenever information-theoretically possible. This procedure achieves the optimal exact recovery threshold determined in (Hajek et al., 2017) with sharp constants if $K \ge (\frac{1}{8e} + o(1)) \frac{n}{\log n}$. See Section 3.1 for a detailed comparison with information-theoretic limits.

The message passing algorithm is simpler to formulate and analyze for the principal submatrix recovery problem; nevertheless, we show in Section 5 how to adapt the message passing algorithm and its analysis to the biclustering problem. Sharp conditions for exact recovery for the biclustering problem was obtained in (Butucea et al., 2015). We show that calculations in (Butucea et al., 2015) with minor adjustments provide information-theoretic conditions for weak recovery as well. The connection between weak and exact recovery via the voting procedure described in (Hajek et al., 2017) carries over to the biclustering problem.

The analysis of the message passing algorithm is based on the moment method adopted in (Deshpande and Montanari, 2015). When the noise matrix Z is Gaussian, an alternative technique to analyze message passing algorithms is introduced in (Bayati and Montanari, 2011) and generalized by (Javanmard and Montanari, 2013). A distinct advantage of the moment method in our context is that the Gaussian assumption can be relaxed to a sub-gaussian assumption. Accordingly, we introduce the following assumption.

3. In this paper, by information-theoretic limits, we mean the sufficient and necessary conditions for attaining weak or exact recovery by any estimator, regardless of its computational cost.
4. The main results (Theorems 1 and 3) of (Deshpande and Montanari, 2015) assume $\mu = \Theta(1)$ but not $K = \Omega(\sqrt{n})$). This is because, as pointed out at the end of the proof of (Deshpande and Montanari, 2015, Theorem 3), if $K = \omega(\sqrt{N})$, then the spectral method and its proof in (Alon et al., 1998) already work. However, the state evolution analysis of message passing algorithm still assumes $K = \Theta(\sqrt{n})$ as stated in (Deshpande and Montanari, 2015, Lemma 2.2).

Assumption 1 *Given $C^* \in [n]$ and $\mu > 0$, the following holds. W is an $n \times n$ symmetric matrix with $\{W_{ij}\}_{1 \le i \le j \le n}$ being mutually independent random variables. Let $Z_{ij} = W_{ij} - \mu \mathbf{I}_{\{i,j \in C^*\}}$. Then $\mathbb{E}[Z_{ij}] = 0$ for all i, j, and $\mathsf{var}(Z_{ij}) = 1$ for $(i,j) \notin C^* \times C^*$. Finally, there is a constant $\gamma > 0$ that does not depend on n such that $\mathbb{E}\left[e^{sZ_{ij}}\right] \le e^{\gamma s^2/2}$ for $s \in \mathbb{R}$, i.e. the W's and Z's are subgaussian with proxy variance γ.*

The variance of a subgaussian random variable is less than or equal to its proxy variance, so Assumption 1 implies $\gamma \ge 1$, and $\mathsf{var}(Z_{ij}) \le \gamma$ for all $i, j \in [n]$ and all $n \ge 1$. Of course, Assumption 1 holds in the Gaussian case such that the Z_{ij} are all $\mathcal{N}(0,1)$ random variables.

Notation For any positive integer n, let $[n] = \{1, \dots, n\}$. For any set $T \subset [n]$, let $|T|$ denote its cardinality and T^c denote its complement. For two sets S, T, let $S \Delta T = (S \setminus T) \cup (T \setminus S)$ denote the set difference. For an $m \times n$ matrix M, let $\|M\|$ and $\|M\|_{\mathrm{F}}$ denote its spectral and Frobenius norm, respectively. Let $\sigma_i(M)$ denote its singular values ordered decreasingly. For any $S \subset [m], T \subset [n]$, let $M_{ST} \in \mathbb{R}^{|S| \times |T|}$ denote $(M_{ij})_{i \in S, j \in T}$ and for $m = n$ abbreviate $M_S = M_{SS}$. For a vector x, let $\|x\|$ denote its Euclidean norm. We use standard big O notations, e.g., for any sequences $\{a_n\}$ and $\{b_n\}$, $a_n = \Theta(b_n)$ or $a_n \asymp b_n$ if there is an absolute constant $c > 0$ such that $1/c \le a_n/b_n \le c$. All logarithms are natural and we use the convention $0 \log 0 = 0$. Let Φ and Q denote the cumulative distribution function (CDF) and complementary CDF of the standard normal distribution, respectively. For $\epsilon \in [0,1]$, define the binary entropy function $h(\epsilon) \triangleq \epsilon \log \frac{1}{\epsilon} + (1-\epsilon) \log \frac{1}{1-\epsilon}$. We say a sequence of events $\mathcal{E}_n$ holds with high probability, if $\mathbb{P}\{\mathcal{E}_n\} \to 1$ as $n \to \infty$. Denote the Kolmogorov-Smirnov (KS) distance between distributions μ and ν by $d_{\mathrm{KS}}(\mu, \nu) \triangleq \sup_{x \in \mathbb{R}} |\mu((-\infty, x]) - \nu((-\infty, x])|$.

2. Algorithms and main results

To avoid a plethora of factors $\frac{1}{\sqrt{n}}$ in the notation, we describe the message-passing algorithm using the scaled version

$$A = \frac{1}{\sqrt{n}} W. \tag{4}$$

Under Assumption 1, the entries A_{ij} are subgaussian with proxy variance $\frac{\gamma}{n}$, mean 0 or $\frac{\mu}{\sqrt{n}}$, and variance $\frac{1}{n}$ for $(i,j) \notin C^* \times C^*$. This section presents algorithms and theoretical guarantees for the symmetric model (2). Extensions to the asymmetric case for the biclustering problem (1) are given in Section 5.2.

Let $f(\cdot, t) \colon \mathbb{R} \to \mathbb{R}$ be a scalar function for each iteration t. Let $\theta_{i \to j}^{t+1}$ denote the message transmitted from index i to index j at iteration $t+1$, which is given by

$$\theta_{i \to j}^{t+1} = \sum_{\ell \in [n] \setminus \{i,j\}} A_{\ell i} f(\theta_{\ell \to i}^t, t), \quad \forall j \ne i \in [n]. \tag{5}$$

with the initial conditions $\theta^0_{i\to j} \equiv 0$. Moreover, let θ^{t+1}_i denote index i's belief at iteration $t+1$, which is given by

$$\theta^{t+1}_i = \sum_{\ell \in [n]\setminus\{i\}} A_{\ell i} f(\theta^t_{\ell\to i}, t). \tag{6}$$

The form of (5) is inspired by belief propagation algorithms, which have the natural non-backtracking property: the message sent from i to j at time $t+1$ does not depend on the message sent from j to i at time t, thereby reducing the effect of echoes of messages sent by j.

To present an informal derivation of the *state evolution equations*, which track the asymptotic distributions of the messages, let us postulate the following assumptions: Suppose that for each fixed t, as $n \to \infty$: (a) the empirical distribution of $(\theta^t_i : i \in C^*)$ converges to $\mathcal{N}(\mu_t, \tau_t^2)$ and the empirical distribution of $(\theta^t_i : i \in [n]\backslash C^*)$ converges to $\mathcal{N}(0, \tau_t^2)$; (b) $\{\theta^t_{i\to j}\}$ are independent of A; (c) $\theta^t_{i\to j} \approx \theta^t_i$. Then it follows from (6) and $K = o(n)$ that for any $i \in C^*$,

$$\begin{aligned}
\mathbb{E}\left[\theta^{t+1}_i \mid \{\theta^t_{\ell\to i} : \ell \neq i\}\right] &\overset{(b)}{=} \sum_{\ell\in[n]\setminus\{i\}} \mathbb{E}\left[A_{\ell i}\right] f(\theta^t_{\ell\to i}, t) \\
&= \frac{\mu}{\sqrt{n}} \sum_{\ell \in C^*\setminus\{i\}} f(\theta^t_{\ell\to i}, t) \\
&\xrightarrow[n\to\infty]{(a),(c)} \sqrt{\lambda}\, \mathbb{E}\left[f(\mu_t + \tau_t Z, t)\right],
\end{aligned}$$

and for any $i \in [n]$,

$$\begin{aligned}
\mathsf{var}\left(\theta^{t+1}_i \mid \{\theta^t_{\ell\to i} : \ell \neq i\}\right) &\overset{(b)}{=} \sum_{\ell\in[n]\setminus\{i\}} \mathsf{var}\left(A_{\ell i}\right) f(\theta^t_{\ell\to i}, t)^2 \\
&= \frac{1}{n} \sum_{\ell\in[n]\setminus(C^*\cup\{i\})} f(\theta^t_{\ell\to i}, t)^2 + o(1) \\
&\xrightarrow[n\to\infty]{(a),(c)} \mathbb{E}\left[f(\tau_t Z, t)^2\right],
\end{aligned}$$

where Z represents a generic standard normal random variable. Since the conditional means and variances have deterministic limits, those are also the limits of the unconditional means and variances. Therefore, we get the following recursive equations for $t \geq 0$:

$$\mu_{t+1} = \sqrt{\lambda}\mathbb{E}\left[f(\mu_t + \tau_t Z, t)\right], \tag{7}$$

$$\tau_{t+1} = \mathbb{E}\left[f(\tau_t Z, t)^2\right], \tag{8}$$

where the initial conditions are $\mu_0 = \tau_0 = 0$. Following (Deshpande and Montanari, 2015), we call (7) and (8) the *state evolution equations*. The heuristic derivation of state evolution equations given above is certainly not rigorous mainly due to the dependency between $\theta^t_{i\to j}$'s and A. In Section 6, we present a rigorous justification of state evolution equations via the moment method following (Deshpande and Montanari, 2015). A crucial fact that we exploit

is the non-backtracking property of the message passing rule (5), which has the effect of reducing the dependency between $\theta^t_{i\to j}$'s and A.

Suppose, for the time being, that message distributions are Gaussian with parameters accurately tracked by the state evolution equations. Then it is reasonable to estimate C^* by selecting those indices i such that θ^{t+1}_i exceeds a given threshold. More specifically, classifying an index i based on θ^{t+1}_i boils down to testing two Gaussian hypotheses with signal-to-noise ratio $\frac{\mu_{t+1}}{\tau_{t+1}}$. This gives guidance for selecting the functions $f(\cdot, t)$ based on μ_t and τ_t to maximize $\frac{\mu_{t+1}}{\tau_{t+1}}$. For $t = 0$ any choice of f is equivalent, so long as $f(0,0) > 0$. Without loss of generality, for $t \geq 1$, we can assume that the variances are normalized, namely, $\tau_t = 1$ (e.g., we take $f(0,0) = 1$ to make $\tau_1 = 1$) and choose $f(\cdot, t)$ to be the maximizer of

$$\max_g \{\mathbb{E}[g(\mu_t + Z)] \colon \mathbb{E}[g(Z)^2] = 1\} \tag{9}$$

where $Z \sim \mathcal{N}(0,1)$. By change of measure, $\mathbb{E}[g(\mu_t + Z)] = \mathbb{E}[g(Z)\rho(Z)]$, where

$$\rho(x) = \frac{\mathrm{d}\mathcal{N}(\mu_t, 1)}{\mathrm{d}\mathcal{N}(0,1)}(x) = e^{x\mu_t - \mu_t^2/2}. \tag{10}$$

Clearly, the best g aligns with ρ and we obtain

$$f(x,t) = \frac{\rho(x)}{\sqrt{\mathbb{E}[\rho^2(Z)]}} = e^{x\mu_t - \mu_t^2}. \tag{11}$$

With this optimized f, we have $\tau_t \equiv 1$ and the state evolution (7) reduces to

$$\mu_{t+1} = \sqrt{\lambda}\mathbb{E}\left[f(\mu_t + Z, t)\right] = \sqrt{\lambda} e^{\frac{\mu_t^2}{2}},$$

or, equivalently,

$$\mu^2_{t+1} = \lambda e^{\mu_t^2}. \tag{12}$$

Therefore if $\lambda > 1/e$, then (12) has no fixed point and hence $\mu_t \to \infty$ as $t \to \infty$.

Directly carrying out the above heuristic program, however, seems challenging. To rigorously justify the state evolution equations in Section 6, we rely on the the method of moments, requiring f to be a polynomial, which prompts us to look for the best polynomial of a given degree that maximizes the signal-to-noise ratio. Denoting the corresponding state evolution by $(\widehat{\mu}_t, \widehat{\tau}_t)$, we aim to solve the following finite-degree version of (9):

$$\max\{\mathbb{E}[g(\widehat{\mu}_t + Z)] \colon \mathbb{E}[g(Z)^2] = 1, \deg(g) \leq d\}. \tag{13}$$

As shown in Lemma 7, this problem can be easily solved via Hermite polynomials, which form an orthogonal basis with respect to the Gaussian measure, and the optimal choice, denoted by $f_d(\cdot, t)$, is the best degree-d L_2-approximation of the the likelihood ratio (10), which can be obtained by normalizing the first $d+1$ terms in the orthogonal expansion of (10). Compared to (Deshpande and Montanari, 2015, Lemma 2.3) which shows the existence of a good choice of polynomial that approximates the ideal state evolution (12) based on Taylor expansions, our approach is to find the best message-passing rule of a given

degree which results in the following state evolution that is optimal among all polynomial f of degree d:

$$\widehat{\mu}_{t+1}^2 = \lambda \sum_{k=0}^{d} \frac{\widehat{\mu}_t^{2k}}{k!}. \tag{14}$$

For any $\lambda > 1/e$, there is an explicit choice of the degree d depending only on λ denoted by $d^*(\lambda)$,[5] so that $\widehat{\mu}_t \to \infty$ as $t \to \infty$ and the state evolution (14) for fixed t correctly predicts the asymptotic behavior of the messages when $n \to \infty$. Therefore, as discussed above, $\widetilde{C}$ produced by thresholding messages θ_i^t, is likely to contain a large portion of C^*, but since $K = o(n)$, it may (and most likely will) also contain a large number of indices not in C^*. Following (Deshpande and Montanari, 2015, Lemma 2.4), we show that the power iteration[6] (a standard spectral method) in Algorithm 1 can remove a large portion of the outlier vertices in $\widetilde{C}$.

Combining message passing plus spectral cleanup yields Algorithm 1 for estimating C^* based on the messages θ_i^t, with theoretical guarantees given in Theorem 1.

Algorithm 1 Message passing

1: Input: $n, K \in \mathbb{N}$, $\mu > 0$, $A \in \mathbb{R}^{n\times n}$, $d^*, t^* \in \mathbb{N}$, and $s^* > 0$.
2: Initialize: $\theta_{i\to j}^0 = 0$ for all $i, j \in [n]$ with $i \neq j$ and $\theta_i^0 = 0$. For $t \geq 0$, define the sequence of degree-d^* polynomials $f_{d^*}(\cdot, t)$ as per Lemma 7 and $\widehat{\mu}_t$ in (14).
3: Run $t^* - 1$ iterations of message passing as in (5) with $f = f_{d^*}$ and compute $\theta_i^{t^*}$ for all $i \in [n]$ as per (6).
4: Find the set $\widetilde{C} = \{i \in [n] : \theta_i^{t^*} \geq \widehat{\mu}_{t^*}/2\}$.
5: (Cleanup via power method) Recall that $A_{\widetilde{C}}$ denotes the restriction of A to the rows and columns with index in $\widetilde{C}$. Sample u^0 uniformly from the unit sphere in $\mathbb{R}^{|\widetilde{C}|}$ and compute $u^{t+1} = A_{\widetilde{C}} u^t / \|A_{\widetilde{C}} u^t\|$ for $0 \leq t \leq \lceil s^* \log n \rceil - 1$. Let $\widehat{u} = u^{\lceil s^* \log n \rceil}$. Return $\widehat{C}$, the set of K indices i in $\widetilde{C}$ with the largest values of $|\widehat{u}_i|$.

Theorem 1 *Fix $\lambda > 1/e$. Let K and μ depend on n in such a way that $\mu^2 K^2/n \to \lambda$ and $\Omega(\sqrt{n}) \leq K \leq o(n)$ as $n \to \infty$. Suppose either C^* is deterministic with $|C^*| \equiv K$, or C^* is random such that $|C^*|/K \to 1$ in probability as $n \to \infty$. Suppose Assumption 1 holds for some $\gamma > 0$. Let $d = d^*(\lambda)$ as in (28). For every $\eta \in (0, 1)$, there exist explicit positive constants t^*, s^*, c depending on λ, η and γ such that Algorithm 1 returns $\widehat{C}$ satisfying $|\widehat{C} \Delta C^*| \leq \eta K$, with probability converging to one as $n \to \infty$, and the total time complexity is bounded by $c(\eta, \lambda, \gamma) n^2 \log n$, where $c(\eta, \lambda, \gamma) \to \infty$ as either $\eta \to 0$ or $\lambda \to 1/e$.*

Remark 2 *Algorithm 1 requires the knowledge of the parameter λ to define the sequence of polynomials $f_{d^*}(\cdot, t)$ and $\widehat{\mu}_t$, and the knowledge of the parameter K in the spectral cleanup step. To avoid the need to know K, we can simply replace the last step of the spectral cleanup (involving choosing the K coordinates of the largest magnitude of $\widehat{u}$) by applying k-means with $k = 2$ on the set $\{|\widehat{u}_i| : i \in \widetilde{C}\}$. See Appendix C for details. With this modification, Theorem 1 continues to hold as long as λ (or a lower bound thereof) is known in order to set the degree d^* and the iteration number t^*.*

5. See (28) and Remark 9 for the expression.
6. As far as statistical utility is concerned, we could replace $\widehat{u}$ produced by the power iteration by the leading singular vector of $A_{\widetilde{C}}$, but that would incur a higher time complexity because singular value decomposition in general takes $O(n^3)$ time to compute.

Remark 3 *As pointed out in (Deshpande and Montanari, 2015, Remeark 2.5), the effective signal-to-noise ratio λ can be potentially improved by a suitable entrywise pre-processing of the observed matrix W. In particular, in (4) we let $A_{ij} = g(W_{ij})$ for some transformation function $g : \mathbb{R} \to \mathbb{R}$. The optimal transformation g in the Gaussian case for which $\gamma = 1$ is given by the maximizer of*

$$\max_g \left\{ \mathbb{E}\left[g(\mu + Z)\right] - \mathbb{E}\left[g(Z)\right] : \mathbb{E}\left[g(Z)^2\right] = \frac{1}{n} \right\}.$$

In view of (9) and (10), the optimal transformation is the scaled likelihood ratio:

$$A_{ij} = \frac{1}{\sqrt{n(e^{\mu^2} - 1)}} \left(\frac{\mathrm{d}\mathcal{N}(\mu, 1)}{\mathrm{d}\mathcal{N}(0, 1)}(W_{ij}) - 1 \right) = \frac{1}{\sqrt{n(e^{\mu^2} - 1)}} \left(e^{W_{ij}\mu - \mu^2/2} - 1 \right)$$

and the signal-to-noise ratio λ is increased to

$$\widetilde{\lambda} = \frac{K^2}{n} \left(e^{\mu^2} - 1 \right).$$

If the resulting A_{ij} is subgaussian with scale $O(1/n)$, then Theorem 1 still applies. However, even if the results extend, in the regime of $K \gg \sqrt{n}$ which we are mostly interested in, we have $\mu \to 0$ and $\widetilde{\lambda} = \lambda(1 + o(1))$, and thus pre-processing cannot boost the signal-to-noise ratio asymptotically.

After the message passing algorithm and spectral cleanup are applied in Algorithm 1, a final linear-time voting procedure is deployed to obtain weak or exact recovery, leading to Algorithm 2 next. As in (Deshpande and Montanari, 2015), we consider a threshold estimator for each vertex i based on a sum over $\widehat{C}$ given by $r_i = \sum_{j \in \widehat{C}} A_{ij}$. Intuitively, r_i can be viewed as the aggregated "votes" received by the index i in $\widehat{C}$, and the algorithm picks the set of K indices with the most significant "votes". To show that this voting procedure succeeds in weak recovery, a key step is to prove that r_i is close to $\sum_{j \in C^*} A_{ij}$. If $\mu = \Theta(1)$ as in (Deshpande and Montanari, 2015), given that $|\widehat{C} \triangle C^*| = o(K)$, the error incurred by summing over $\widehat{C}$ instead of over C^* could be bounded by truncating A_{ij} to a large magnitude. However, for $\mu \to 0$ that approach fails. Our approach is to introduce the clean-up procedure in Algorithm 2 based on the *successive withholding* method described in (Hajek et al., 2017) (see also (Condon and Karp, 2001; Mossel et al., 2014) for variants of this method). In particular, we randomly partition the set of vertices into $1/\delta$ subsets. One at a time, one subset, say S, is withheld to produce a reduced set of vertices S^c, on which we apply Algorithm 1. The estimate obtained from S^c is then used by the voting procedure to classify the vertices in S. The analysis of the two stages is decoupled because conditioned on C^*, the outcome of Algorithm 1 depends only on A_{S^c}, which is independent of A_{SS^c} used in the voting.

Algorithm 2 Message passing plus voting

1: Input: $n, K \in \mathbb{N}$, $\mu > 0$, $A \in \mathbb{R}^{n\times n}$, $\delta \in (0,1)$ with $1/\delta, n\delta \in \mathbb{N}$, $d^*, t^* \in \mathbb{N}$, and $s^* > 0$.
2: Partition $[n]$ into $1/\delta$ subsets S_k of size $n\delta$ randomly.
3: (Approximate recovery) For each $k = 1, \ldots, 1/\delta$, run Algorithm 1 (message passing for approximate recovery) with input $\big(n(1-\delta), \lceil K(1-\delta)\rceil, \mu, A_{S_k^c}, d^*, t^*, s^*\big)$ which outputs $\widehat{C}_k$.
4: (Clean up) For each $k = 1, \ldots, 1/\delta$ compute $r_i = \sum_{j\in\widehat{C}_k} A_{ij}$ for all $i \in S_k$ and return C', the set of K indices in $[n]$ with the largest values of r_i.

The following theorem provides a sufficient condition for the message passing plus voting cleanup procedure (Algorithm 2) to achieve weak recovery, and, if an additional sufficient condition is also satisfied, exact recovery.

Theorem 4 *Suppose K and μ depend on n in such a way that $\frac{\mu^2 K^2}{n} \to \lambda$ for some fixed $\lambda > 1/e$, and $\Omega(\sqrt{n}) \le K \le o(n)$ as $n \to \infty$. Suppose Assumption 1 holds for some $\gamma > 0$ and $|C^*| = K$. Let $\delta > 0$ be such that $\lambda e(1-\delta) > 1$. Define $d^* = d^*(\lambda(1-\delta))$ as per (28). Then there exist positive constants t^*, s^*, c determined explicitly by δ, λ and γ, such that*

1. *(Weak recovery) Algorithm 2 returns C' with $|C'\Delta C^*|/K \to 0$ in probability as $n \to \infty$.*
2. *(Exact recovery) Furthermore, assume that*

$$\liminf_{n\to\infty} \frac{\sqrt{K}\mu}{\sqrt{2\log K} + \sqrt{2\log n}} > \sqrt{\gamma}. \tag{15}$$

Let $\delta > 0$ be chosen such that for all sufficiently large n,

$$\min\left\{\lambda e(1-\delta), \frac{K\mu(1-2\delta)}{\sqrt{2K\gamma\log K} + \sqrt{2K\gamma\log(n-K)} + \delta\sqrt{K}}\right\} > 1.$$

Then Algorithm 2 returns C' with $\mathbb{P}\{C' \neq C^*\} \to 0$ as $n \to \infty$.

The total time complexity is bounded by $c(\delta,\lambda,\gamma)n^2\log n$, where $c(\delta,\lambda,\gamma) \to \infty$ as $\delta \to 0$ or $\lambda \to 1/e$.

Remark 5 *Theorem 4 ensures Algorithm 2 achieves exact recovery if both (15) and $\lambda > 1/e$ hold; it is of interest to compare these two conditions. Note that*

$$\frac{\sqrt{K}\mu}{\sqrt{2\log K} + \sqrt{2\log n}} = \sqrt{\lambda e} \times \sqrt{\frac{n}{8eK\log n}} \frac{2}{(1 + \sqrt{\log K/\log n})}.$$

Hence, if $\liminf_{n\to\infty} K\log n/n \ge 1/(8e\gamma)$, then $\liminf_{n\to\infty} \log K/\log n = 1$; thus (15) implies $\lambda > 1/e$ and hence (15) alone is sufficient for Algorithm 2 to succeed. If instead $\limsup_{n\to\infty} K\log n/n \le 1/(8e\gamma)$, then $\lambda > 1/e$ implies (15) and thus $\lambda > 1/e$ alone is sufficient for Algorithm 2 to succeed. The asymptotic regime considered in (Deshpande and Montanari, 2015) entails $K = \Theta(\sqrt{n})$, in which case the condition $\lambda > 1/e$ is sufficient for exact recovery, as shown in (Deshpande and Montanari, 2015). The idea of upgrading weak recovery to exact recovery via a local voting procedure has also appeared in (Abbe et al., 2016; Mossel et al., 2015; Abbe and Sandon, 2015; Yun and Proutiere, 2015) under the context of stochastic block models with community sizes scaling linearly in n. As shown in (Hajek et al., 2017, Corollary 4) in the Gaussian case for which $\gamma = 1$, the condition (15), with strict inequality replaced by greater than or equal, is necessary for exact recovery.

We finish this section by discussing the connections and distinctions to the previous work (Deshpande and Montanari, 2015). Versions of Theorems 1 and 4 are given in (Deshpande and Montanari, 2015) for the case $K = \Theta(\sqrt{n})$ and $\mu = \Theta(1)$. We extend the range of K to $\Omega(\sqrt{n}) \le K \le o(n)$, showing that the message passing plus a cleanup procedure achieves the optimal exact recovery threshold in the Gaussian case with sharp constants if $K \ge (\frac{1}{8e} + o(1))\frac{n}{\log n}$. The algorithms and proofs are nearly the same; we comment here on the main difficulties we encountered when allowing $K/\sqrt{n} \to \infty$ and $\mu \to 0$.

First, a key ingredient in the proof of Theorem 1 is Lemma 6. Its proof is based on the moment method and a larger K requires modification of bounds from (Deshpande and Montanari, 2015) used in calculating the moments of messages, i.e., $\mathbb{E}\left[(\theta^t_{i\to j})^m\right]$ for fixed $m \in \mathbb{N}$, by a more careful counting argument. We refer the interested readers to Remark 31 right after the proof of Lemma 6 for more details.

Secondly, after the message passing algorithm and spectral cleanup are applied in Algorithm 1, a final cleanup procedure is applied to obtain weak recovery or exact recovery (when possible). As in (Deshpande and Montanari, 2015), we consider a threshold estimator for each vertex i based on a sum over $\widehat{C}$. If $K = \Theta(\sqrt{n})$ as assumed in (Deshpande and Montanari, 2015), then λ being a constant implies that the mean μ is bounded away from zero. In this case if $|\widehat{C}\triangle C^*| = o(K)$, the error incurred by summing over $\widehat{C}$ instead of over C^* could be bounded by truncating A_{ij} to a large magnitude $\bar{\rho}$ and bounding the difference of sums by $\bar{\rho}\big|C^*\triangle\widehat{C}\big| = o(K) \ll \mu K$. However, for $K \gg \sqrt{n}$ with vanishing μ this approach fails. Instead, we rely on the cleanup procedure in Algorithm 2 which entails running Algorithm 1 for $1/\delta$ times on subsampled vertices. A related difference we encounter is that if K is large enough then the condition $\lambda > 1/e$ alone is not sufficient for exact recovery, but adding the information-theoretic condition (15) suffices.

Lastly, the method of moments requires $f(\cdot, t)$ to be a polynomial so that the exponential function (11), which results in the ideal state evolution (12), cannot be directly applied. It is shown in (Deshpande and Montanari, 2015, Lemma 2.3) that for any $\lambda > 1/e$ and any threshold M there exists $d^* = d^*(\lambda, M)$ so that taking f to be the truncated Taylor series of (11) up to degree d^* results in the state evolution $\widehat{\mu}_t$ which exceeds M after some finite time $t^*(\lambda, M)$; however, no explicit formula of d^*, which is needed to instantiate Algorithm 1, is provided. Although in principle this does not pose any algorithmic problem as d^* can be found by an exhaustive search in $O(1)$ time independent of n, it is more satisfactory to find the best polynomial message passing rule explicitly which maximizes the signal-to-noise ratio for a given degree (Lemma 7) and provides an explicit formula of d^* as a function of λ only (Remark 9).

3. Statistical optimality and computational considerations

3.1 Comparison with information-theoretic limits in the Gaussian case

As noted in the introduction, in the regime $K = \Theta(n)$, a thresholding algorithm based on row sums provides weak and, if a voting procedure is also used, exact recovery whenever it is informationally possible in the Gaussian case. In this subsection, we compare the performance of the message passing algorithms to the information-theoretic limits on the recovery problem in the regime (3). Throughout this section we restrict attention to the Gaussian case, such that $Z_{ij} \sim \mathcal{N}(0,1)$ and $\gamma = 1$. Also, for converse results, we assume the true community C^* is a subset of $[n]$ of cardinality K selected uniformly at random. Notice that the comparison here takes into account the sharp constant factors. Information-theoretic limits for the biclustering problem are discussed in Section 5.1.

Weak recovery The information-theoretic threshold for weak recovery has been determined in (Hajek et al., 2017, Theorem 2), which, in the regime of (3), boils down to the following: If

$$\liminf_{n\to\infty} \frac{K\mu^2}{4\log\frac{n}{K}} > 1, \tag{16}$$

then weak recovery is possible; conversely, if weak recovery is possible, then

$$\liminf_{n\to\infty} \frac{K\mu^2}{4\log\frac{n}{K}} \geq 1. \tag{17}$$

This implies that the minimal signal-to-noise ratio for weak recovery is

$$\lambda \geq (4+\epsilon)\frac{K}{n}\log\frac{n}{K}$$

for any $\epsilon > 0$, which vanishes in the sublinear regime of $K = o(n)$. In contrast, in the regime of (3), message passing (Algorithm 1) demands a non-vanishing signal-to-noise ratio, namely, $\lambda > 1/e$, to achieve weak recovery. No polynomial-time algorithm is known to succeed if $\lambda \leq 1/e$, suggesting that computational complexity might incur a severe penalty on the statistical optimality when $K = o(n)$.

Exact recovery When the submatrix size satisfies (3), if (15) with $\gamma = 1$ holds, then exact recovery is possible; conversely, if exact recovery is possible, then

$$\liminf_{n\to\infty} \frac{\sqrt{K}\mu}{\sqrt{2\log K} + \sqrt{2\log n}} \geq 1. \tag{18}$$

See (Hajek et al., 2017, Corollary 4). This implies that the minimal signal-to-noise ratio for exact recovery is

$$\lambda \geq (2+\epsilon)\frac{K}{n}\left(\sqrt{\log n} + \sqrt{\log K}\right)^2 \tag{19}$$

for any $\epsilon > 0$. Consequently, we find that the critical submatrix size for which message passing (plus cleanup) can achieve optimal exact recovery is $\frac{n}{\log n}$. Specifically,

- $K = \omega\left(\frac{n}{\log n}\right)$. In this regime, the right hand side of (19) goes to ∞ and hence the minimal signal-to-noise ratio for exact recovery is much higher than that of weak recovery via message passing, namely, $1/e$. Thus, exact recovery can be attainable in polynomial-time by message-passing plus voting clean-up (Algorithm 2).

- $K = \Theta(\frac{n}{\log n})$. In this regime, if we let $K = \frac{\rho n}{\log n}$, the right hand side of (19) is at least $1/e$ if $\rho \geq \frac{1}{8e}$ and strictly less than $1/e$ otherwise. In view of Theorem 4, we conclude that message passing plus voting cleanup (Algorithm 2) achieves the sharp threshold of exact recovery if

$$K \geq \left(\frac{1}{8e} + o(1)\right)\frac{n}{\log n}. \tag{20}$$

- $K = o\left(\frac{n}{\log n}\right)$. In this regime, the right hand side of (19) is $o(1)$. No polynomial-time algorithm (including semidefinite programming relaxation (Hajek et al., 2016)) is known to achieve weak, let alone exact, recovery, when $\lambda = o(1)$.

A counterpart of this conclusion for the biclustering problem is obtained in Remark 20 in terms of the submatrix sizes.

3.2 Comparison with the spectral limit

It is reasonable to conjecture that $\lambda > 1$ is the spectral limit for recoverability by spectral estimation methods. This conjecture is rather vague, because it is difficult to define what constitutes spectral methods. Nevertheless, some evidence for this conjecture is provided by (Deshpande and Montanari, 2015, Proposition 1.1), which, in turn, is based on results on the spectrum of a random matrix perturbed by adding a rank-one deterministic matrix (Knowles and Yin, 2013, Theorem 2.7).

The message passing framework used in this paper itself provides some evidence for the conjecture. Indeed, if $f(x,0) \equiv 1$ and $f(x,t) = x$ for all $t \geq 1$, the iterates θ^t are close to what is obtained by iterated multiplication by the matrix A, beginning with the all one vector, which is the power method for computation of the eigenvector corresponding to the principal eigenvalue of A.[7] To be more precise, with this linear f the message passing equation (5) can be expressed in terms of powers of the *non-backtracking matrix* $\mathbf{B} \in \mathbb{R}^{\binom{n}{2}\times\binom{n}{2}}$ associated with the matrix A, defined by $B_{ef} = A_{e_1,e_2}\mathbf{1}_{\{e_2=f_1\}}\mathbf{1}_{\{e_1\neq f_2\}}$, where $e = (e_1, e_2)$ and $f = (f_1, f_2)$ are directed pairs of indices. Let $\Theta^t \in \mathbb{R}^{n(n-1)}$ denote the messages on directed edges with $\Theta^t_e = \theta^t_{e_1\to e_2}$. Then, (5) simply becomes $\Theta^t = \mathbf{B}^t\mathbf{1}$. To evaluate the performance of this method, we turn to the state evolution equations (7) and (8), which yield $\mu_t = \lambda^{t/2}$ and $\tau_t = 1$ for all $t \geq 1$. Therefore, by a simple variation of Algorithm 1 and Theorem 1, if $\lambda > 1$, the linear message passing algorithm can provide weak recovery.

For the submatrix *detection* problem, namely, testing $\mu = 0$ (pure noise) versus $\mu > 0$, as opposed to support recovery, if λ is fixed with $\lambda > 1$, a simple thresholding test based

7. If we included i, j in the summation in (5) and (6), then we would have $\theta^t = A^t\mathbf{1}$ exactly. Since the entries of A are $O_P(1/\sqrt{n})$, we expect this only incurs a small difference to the sum for finite number of iterations.

on the largest eigenvalue of the matrix A provides detection error probability converging to zero (Féral and Péché, 2007), while if $\lambda < 1$ no test based solely on the eigenvalues of A can achieve vanishing probability of error (Montanari et al., 2015). It remains, however, to establish a solid connection between the detection and estimation problem for submatrix localization for spectral methods.

3.3 Computational barriers in the Gaussian case

A recent line of work (Kolar et al., 2011; Ma and Wu, 2015; Chen and Xu, 2014; Cai et al., 2017) has uncovered a fascinating interplay between statistical optimality and computational efficiency for the *recovery* problem and the related *detection* and *estimation* problem.[8] Assuming the hardness of the planted clique problem, rigorous computational lower bounds have been obtained in (Ma and Wu, 2015; Cai et al., 2017) through reduction arguments in the Gaussian case. In particular, it is shown in (Ma and Wu, 2015) that when $K = n^\alpha$ for $0 < \alpha < 2/3$, merely achieving the information-theoretic limits of detection within any constant factor (let alone sharp constants) is as hard as detecting the planted clique; the same hardness also carries over to exact recovery in the same regime. Furthermore, it is shown that the hardness of estimating this type of matrix, which is both low-rank and sparse, highly depends on the loss function (Ma and Wu, 2015, Section 5.2). For example, for $K = \Theta(\sqrt{n})$, entry-wise thresholding attains an $O(\log n)$ factor of the minimax mean-square error; however, if the error is gauged in squared operator norm instead of Frobenius norm, attaining an $O(\sqrt{n}/\log n)$ factor of the minimax risk is as hard as solving planted clique. Similar reductions have been shown in (Cai et al., 2017) for exact recovering of the submatrix of size $K = n^\alpha$ and the planted clique recovery problem for any $0 < \alpha < 1$.

The results in (Ma and Wu, 2015; Cai et al., 2017) revealed that the difficulty of submatrix localization crucially depends on the size and planted clique hardness kicks in if $K = n^{1-\Theta(1)}$. In search of the exact phase transition point where statistical and computational limits depart, we further zoom into the regime of $K = n^{1-o(1)}$. We showed in (Hajek et al., 2016) no computational gap exists in the regime $K = \omega(n/\log n)$, since a semidefinite programming relaxation of the maximum likelihood estimator can achieve the information limit for exact recovery with sharp constants. The current paper further pushes the boundary to $K \geq \frac{n}{\log n}(\frac{1}{8e} + o(1))$, in which case the sharp information limits can be attained in nearly linear-time via message passing plus clean-up. However, as soon as $K \leq \frac{n}{\log n}(\frac{1}{8e} - \epsilon)$ for any $\epsilon > 0$, a gap emerges between the statistical limits and the sufficient condition of message passing plus clean-up, given by $\lambda > 1/e$.

4. Proofs of algorithm correctness

We first justify the state evolution equations via the following key lemma, which establishes the asymptotic normality of the empirical distribution of messages with mean and variance given by (7) and (8). A version of this lemma is proved in (Deshpande and Montanari, 2015) by assuming $\mu = \Theta(1)$ and $K = \Theta(\sqrt{n})$. The proof is given in Section 6 using the method of moments, closely following (Deshpande and Montanari, 2015).

8. The papers (Kolar et al., 2011; Ma and Wu, 2015; Chen and Xu, 2014; Cai et al., 2017) considered the biclustering version of the submatrix localization problem (1).

Lemma 6 *Let $f(\cdot, t)$ be a finite-degree polynomial for each $t \geq 0$. Let K and μ depend on n such that $\frac{K^2\mu^2}{n} \equiv \lambda$ for some $\lambda > 0$ and $\Omega(\sqrt{n}) \leq K \leq o(n)$. Suppose Assumption 1 holds for some $\gamma > 0$, and suppose either C^* is deterministic with $|C^*| \equiv K$, or C^* is random such that $|C^*|/K \to 1$ in probability as $n \to \infty$. Let $A = W/\sqrt{n}$ and set $\theta^0_{i\to j} = 0$. Consider the message passing algorithm defined by (5) and (6). Then for each fixed t, as $n \to \infty$,*

$$d_{\mathrm{KS}}\left(\frac{1}{|C^*|}\sum_{i\in C^*}\delta_{\theta_i^t},\, \mathcal{N}(\mu_t, \tau_t^2)\right) \xrightarrow{p} 0,$$
$$d_{\mathrm{KS}}\left(\frac{1}{n-|C^*|}\sum_{i\notin C^*}\delta_{\theta_i^t},\, \mathcal{N}(0, \tau_t^2)\right) \xrightarrow{p} 0,$$

where μ_t and τ_t are defined in (7) and (8), respectively; $\frac{1}{|C^*|}\sum_{i\in C^*}\delta_{\theta_i^t}$ and $\frac{1}{n-|C^*|}\sum_{i\notin C^*}\delta_{\theta_i^t}$ are the empirical distributions of θ_i^t for $i \in C^*$ and $i \notin C^*$, respectively.

Next we prove Theorems 1-4. Lemma 6 implies that if $i \in C^*$, then $\theta_i^t \sim \mathcal{N}(\mu_t, \tau_t^2)$; if $i \notin C^*$, then $\theta_i^t \sim \mathcal{N}(0, \tau_t^2)$. Ideally, one would pick the optimal $f(x,t) = e^{\mu_t(x-\mu_t)}$ which result in the optimal state evolution $\mu_{t+1} = \sqrt{\lambda}e^{\mu_t^2/2}$ and $\tau_t = 1$ for all $t \geq 1$. Furthermore, if $\lambda > 1/e$, then $\mu_t \to \infty$ as $t \to \infty$, and thus we can hope to estimate C^* by selecting the indices i such that θ_i^t exceeds a certain threshold. The caveat is that Lemma 6 needs f to be a polynomial of finite degree. Next we proceed to find the best degree-d polynomial for iteration t, denoted by $f_d(\cdot, t)$, which maximizes the signal to noise ratio.

Recall that the Hermite polynomials $\{H_k : k \geq 0\}$ are the orthogonal polynomials with respect to the standard normal distribution (cf. (Szegö, 1975, Section 5.5)), given by

$$H_k(x) = (-1)^k\frac{\varphi^{(k)}(x)}{\varphi(x)} = \sum_{i=0}^{\lfloor k/2\rfloor}(-1)^i(2i-1)!!\binom{k}{2i}x^{k-2i}, \tag{21}$$

where φ denotes the standard normal density and $\varphi^{(k)}(x)$ is its k-th derivative; in particular, $H_0(x) = 1, H_1(x) = x, H_2(x) = x^2 - 1$. Furthermore, $\deg(H_k) = k$ and $\{H_0, \ldots, H_d\}$ span all polynomials of degree at most d. For $Z \sim \mathcal{N}(0,1)$, $\mathbb{E}[H_m(Z)H_n(Z)] = m!\delta_{m=n}$ and $\mathbb{E}[H_k(\mu + Z)] = \mu^k$ for all $\mu \in \mathbb{R}$; hence the relative density $\frac{\mathrm{d}\mathcal{N}(\mu,1)}{\mathrm{d}\mathcal{N}(0,1)}(x) = e^{\mu x - \mu^2/2}$ admits the following expansion:

$$e^{\mu x - \mu^2/2} = \sum_{k=0}^{\infty}H_k(x)\frac{\mu^k}{k!}. \tag{22}$$

Truncating and normalizing the series at the first $d+1$ terms immediately yields the solution to (13) as the best degree-d L_2-approximation to the relative density, described as follows:

Lemma 7 *Fix $d \in \mathbb{N}$ and define $\widehat{\mu}_t$ according to the iteration (14) with $\widehat{\mu}_0 = 0$, namely,*

$$\widehat{\mu}_{t+1}^2 = \lambda G_d(\widehat{\mu}_t^2), \qquad G_d(\mu) = \sum_{k=0}^{d}\frac{\mu^k}{k!}. \tag{23}$$

Define

$$f_d(x,t) = \sum_{k=0}^{d} a_k H_k(x), \tag{24}$$

where $a_k \triangleq \frac{\widehat{\mu}_t^k}{k!}(\sum_{k=0}^{d} \frac{\widehat{\mu}_t^{2k}}{k!})^{-1/2}$. *Then* $f_d(\cdot,t)$ *is the unique maximizer of (13) and the state evolution (7) and (8) with* $f = f_d$ *coincides with* $\tau_t = 1$ *and* $\mu_t = \widehat{\mu}_t$. *Furthermore, for any* $d \geq 2$ *the equation*

$$G_d(a) = aG_{d-1}(a) \tag{25}$$

has a unique positive solution, denoted by a_d^*. *Let*

$$\lambda_d^* = \frac{1}{G_{d-1}(a_d^*)} \tag{26}$$

and define $\lambda_1^* = 1$. *Then*

1. *for any* $d \in \mathbb{N}$ *and any* $\lambda > \lambda_d^*$, $\widehat{\mu}_t \to \infty$ *as* $t \to \infty$ *and hence for any* $M > 0$,

$$t^*(\lambda, M) = \inf\{t : \widehat{\mu}_t > M\} \tag{27}$$

   is finite;

2. $\lambda_d^* \downarrow 1/e$ *monotonically as* $d \to \infty$ *according to* $\lambda_d^* = 1/e + \frac{1/e^2 + o(1)}{(d+1)!}$.

Remark 8 *The best affine update gives* $\lambda_1^* = 1$; *for the best quadratic update,* $a_2^* = \sqrt{2}$ *and hence* $\lambda_2^* = \frac{1}{1+\sqrt{2}} \approx 0.414$. *More values of the threshold are given below, which converges to* $1/e \approx 0.368$ *rapidly.*

d	1	2	3	4	5
λ_d^*	1	0.414	0.376	0.369	0.368

Remark 9 *Let*

$$d^*(\lambda) = \inf\{d \in \mathbb{N} : \lambda_d^* < \lambda\}, \tag{28}$$

which is finite for any $\lambda > 1/e$. *Then for any* $d \geq d^*$, $\widehat{\mu}_t \to \infty$ *as* $t \to \infty$. *As* λ *approaches the critical value* $1/e$, *the degree* $d^*(\lambda)$ *blows up according to* $d^*(\lambda) = \Theta(\log \frac{1}{\lambda e - 1} / \log\log \frac{1}{\lambda e - 1})$, *as a consequence of the last part of Lemma 7.*

Remark 10 (Best affine message passing) *For* $d = 1$, *the best state evolution is given by*

$$\widehat{\mu}_{t+1}^2 = \lambda(1 + \widehat{\mu}_t^2)$$

and the corresponding optimal update rule is

$$f_1(x,t) = \frac{1 + \widehat{\mu}_t x}{\sqrt{1 + \widehat{\mu}_t^2}}.$$

This is strictly better than $f(x,t) = x$ *described in Section 3.2 which gives* $\widehat{\mu}_{t+1}^2 = \lambda \widehat{\mu}_t^2$; *nevertheless, in order to have* $\widehat{\mu}_t \to \infty$ *we still need to assume the spectral limit* $\lambda \geq 1$.

Proof [Proof of Lemma 7] Note that any degree-d polynomial g can be written in terms of the linear combination:

$$f_d(x,t) = \sum_{k=0}^{d} c_k H_k(x),$$

where the coefficients $\{c_k\}$ satisfy $\mathbb{E}[g^2(Z)] = \sum_{k=0}^{d} k! c_k^2 = 1$. By a change of measure, $\mathbb{E}[g(\widehat{\mu}_t + Z)] = \mathbb{E}[g(Z) e^{\widehat{\mu}_t Z - \widehat{\mu}_t^2/2}] = \sum_{k=0}^{d} c_k \widehat{\mu}_t^k$, in view of the orthogonal expansion (22). Thus, to solve the maximization problem (13), it is equivalent to solving

$$\max_{c_k} \left\{ \sum_{k=0}^{d} c_k \widehat{\mu}_t^k : \sum_{k=0}^{d} k! c_k^2 = 1 \right\}.$$

By Cauchy-Schwarz inequality, the optimal coefficients and the optimal polynomial $f_d(\cdot,t)$ are given by (24), resulting in the following state evolution

$$\widehat{\mu}_{t+1} = \sqrt{\lambda} \max\{\mathbb{E}[g(\widehat{\mu}_t + Z)] \colon \mathbb{E}[g(Z)^2] = 1, \deg(g) \leq d\} = \left(\lambda \sum_{k=0}^{d} \frac{\widehat{\mu}_t^{2k}}{k!} \right)^{1/2},$$

which is equivalent to (23).

Next we analyze the behavior of the iteration (23). The case of $d = 1$ follows from the obvious fact that $\widehat{\mu}_{t+1}^2 = \lambda(\widehat{\mu}_t^2 + 1)$ diverges if and only if $\lambda \geq 1$. For $d \geq 2$, note that G_d is a strictly convex function with $G_d(0) = 1$ and $G_d' = G_{d-1}$. Also, $(G_d(a) - aG_{d-1}(a))' = -aG_d''(a) < 0$. Thus, $G_d(a) - aG_{d-1}(a)$ is strictly decreasing on $a > 0$ with value $\frac{1}{d!}$ at $a = 1$ and limit $-\infty$ as $a \to \infty$, so (25) has a unique positive solution a_d^* and it satisfies $a_d^* > 1$. Furthermore, $(G_d(a) - aG_{d-1}(a))'\big|_{a=1} = -\sum_{k=0}^{d-2} \frac{1}{k!}$, so by Taylor's theorem,

$$G_d(a) - aG_{d-1}(a) = \frac{1}{d!} - (a-1) \sum_{k=0}^{d-2} \frac{1}{k!} + O((a-1)^2),$$

yielding

$$a_d^* = 1 + \frac{1}{d! \sum_{k=0}^{d-2} \frac{1}{k!}} + O(1/(d!)^2).$$

Consider next the values of λ such that $\widehat{\mu}_t$ diverges. For very large λ, $G_d(a)$ dominates a/λ pointwise and $\widehat{\mu}_t$ diverges. The critical value of λ is when $G_d(a)$ and a/λ meet tangentially, namely,

$$\lambda G_{d-1}(a) = 1, \quad \lambda G_d(a) = a,$$

whose solution is given by $a = a_d^*$ and $\lambda = \lambda_d^*$, where

$$\begin{aligned}
\lambda_d^* &\triangleq \frac{1}{G_{d-1}(a_d^*)} = \frac{1}{G_{d-1}(1) + G_{d-1}'(1)(a_d^* - 1) + O((a_d^* - 1)^2)} \\
&= \frac{1}{\sum_{k=0}^{d} \frac{1}{k!} + O(1/(d!)^2)} = 1/e + \frac{\sum_{k=d+1}^{\infty} 1/k! + O(1/(d!)^2)}{e \sum_{k=0}^{d} 1/k!} \\
&= 1/e + \frac{1/e^2 + o(1)}{(d+1)!}.
\end{aligned}$$

Thus, λ_d^* is the minimum value such that for all $\lambda > \lambda_d^*$, $\lambda G_d(a) > a$ for all $a > 0$, so that starting from any $\widehat{\mu}_t \geq 0$ we have $\widehat{\mu}_t \to \infty$ monotonically. The fact λ_d^* is decreasing in d follows from the fact G_d is pointwise increasing in d. ■

Lemmas 6 and 7 immediately imply the following partial recovery results.

Lemma 11 *Assume that $\lambda > 1/e$ and $\Omega(\sqrt{n}) \leq K \leq o(n)$. Fix any $\epsilon \in (0,1)$. Let $M = \sqrt{8\log(1/\epsilon)}$ and run the message passing algorithm for t iterations with $f = f_{d^*}$, $d^* = d^*(\lambda)$ as in (28), and $t = t^*(\lambda, M)$ as in (27). Let $\widetilde{C} = \{i : \theta_i^{t^*} \geq \widehat{\mu}_{t^*}/2\}$. Then with probability converging to one as $n \to \infty$,*

$$\frac{1}{K}|\widetilde{C} \cap C^*| \geq 1 - \epsilon \tag{29}$$

$$K(1-\epsilon) \leq |\widetilde{C}| \leq n\epsilon. \tag{30}$$

Proof Notice that

$$|\widetilde{C} \cap C^*| = \sum_{i \in C^*} \mathbf{1}_{\{\theta_i^{t^*} \geq \widehat{\mu}_{t^*}/2\}}.$$

By the choice of $f = f_d$ in (24), we have $\tau_t = 1$ for all $t \geq 1$. It follows from Lemma 6 that

$$\lim_{n\to\infty} \frac{1}{K}|\widetilde{C} \cap C^*| = \mathbb{P}\left\{\widehat{\mu}_{t^*} + Z \geq \widehat{\mu}_{t^*}/2\right\}, \tag{31}$$

where the convergence is in probability. Notice that we have used $d = d^*(\lambda)$ and $t = t^*(\lambda, M)$ defined by (28) and (27) in Lemma 7. Thus $\widehat{\mu}_{t^*} \geq M = \sqrt{8\log(1/\epsilon)}$ and

$$\mathbb{P}\left\{\widehat{\mu}_{t^*} + Z \leq \widehat{\mu}_{t^*}/2\right\} = Q(\widehat{\mu}_{t^*}/2) \leq e^{-\widehat{\mu}_{t^*}^2/8} \leq \epsilon,$$

which, in view of (31), implies (29) with probability converging to one as $n \to \infty$. Similarly, Lemma 6 implies that in probability

$$\lim_{n\to\infty} \frac{1}{n}|\widetilde{C} \backslash C^*| = \mathbb{P}\left\{Z \geq \widehat{\mu}_{t^*}/2\right\} = Q(\widehat{\mu}_{t^*}/2) \leq \epsilon. \tag{32}$$

Since $K = o(n)$, we have $\mathbb{P}\{K(1-\epsilon) \leq |\widetilde{C}| \leq n\epsilon\} \to 1$. ■

Although $\widetilde{C}$ contains a large portion of C^*, since $|\widetilde{C}|$ is linear in n with high probability, i.e., $|\widetilde{C}|/n \to Q(\widehat{\mu}_{t^*}/2)$ by Lemma 6, it is bound to contain a large number of outlier indices. The next lemma, closely following (Deshpande and Montanari, 2015, Lemma 2.4), shows that given the conclusion of Lemma 11, the power iteration in Algorithm 1 can remove most of the outlier indices in $\widetilde{C}$. Its proof is presented in Appendix B.

Lemma 12 *Suppose $\lambda = \frac{\mu^2 K^2}{n} \geq 1/e$, $K \to \infty$, $|C^*|/K \to 1$ in probability, and $\widetilde{C}$ is a set (possibly depending on A) such that (29) – (30) hold for some $0 < \epsilon < \epsilon_0$, where $0 < \epsilon_0 \leq 1/2$ is determined by $1 - \epsilon_0 = 8e\sqrt{4\gamma h(\epsilon_0) + 10\gamma\epsilon_0}$. Let*

$$s^* = \frac{2}{\log(\sqrt{\lambda}(1-\epsilon)/(8\sqrt{4e\gamma h(\epsilon) + 10e\gamma\epsilon}))}, \tag{33}$$

where $h(\epsilon) \triangleq \epsilon\log\frac{1}{\epsilon} + (1-\epsilon)\log\frac{1}{1-\epsilon}$ is the binary entropy function. Then $\widehat{C}$ with $|\widehat{C}| \equiv K$ produced by Algorithm 1 returns $|\widehat{C}\Delta C^*| \leq \eta(\epsilon,\lambda,\gamma)K$, with probability converging to one as $n \to \infty$, where

$$\eta(\epsilon,\lambda,\gamma) = 2\epsilon + e\gamma\frac{4608h(\epsilon) + 11520\epsilon}{\lambda(1-\epsilon)^2}. \tag{34}$$

Proof [Proof of Theorem 1] Given $\eta \in (0,1)$, choose an arbitrary $\epsilon \in (0,\epsilon_0)$ such that $\eta(\epsilon,\lambda,\gamma)$ defined in (34) is at most η. With t^* specified in Lemma 11 and s^* specified in Lemma 12, the probabilistic performance guarantee in Theorem 1 readily follows by combining Lemmas 11 and 12. The time complexity of Algorithm 1 follows from the fact that for both the BP algorithm and the power method each iteration has complexity $O(n^2)$ and Algorithm 1 entails running BP and the power method for t^* and $\lceil s^*\log n\rceil$ iterations respectively; both t^* and s^* are constants depending only on η, λ, and γ. ■

Proof [Proof of Theorem 4] (Weak recovery) Fix $k \in [1/\delta]$ and let $C_k^* = C^* \cap S_k^c$. Define the $n(1-\delta)\times n(1-\delta)$ matrix $A_k \triangleq A_{S_k^c}$, which corresponds to the submatrix localization problem for a planted community C_k^* whose size has a hypergeometric distribution, resulting from sampling without replacement, with parameters $(n, K, (1-\delta)n)$ and mean $(1-\delta)K$. By a result of (Hoeffding, 1963), the distribution of $|C_k^*|$ is convex order dominated by the distribution that would result from sampling with replacement, namely, the Binom $\left(n(1-\delta), \frac{K}{n}\right)$ distribution. In particular, Chernoff bounds for Binom$(n(1-\delta), \frac{K}{n})$ also hold for $|C_k^*|$, so $|C_k^*|/((1-\delta)K) \to 1$ in probability as $n \to \infty$. Note that $\frac{((1-\delta)K)^2\mu^2}{n(1-\delta)} \to \lambda(1-\delta)$ and $\lambda(1-\delta)e > 1$ by the choice of δ. Let $d^*(\lambda(1-\delta))$ be given in (28), i.e.,

$$d^*(\lambda(1-\delta)) = \inf\{d \in \mathbb{N} : \lambda_d^* < \lambda(1-\delta)\}.$$

Choose an arbitrary $\epsilon \in (0,\epsilon_0)$ to satisfy $\eta(\epsilon,\lambda(1-\delta),\gamma) \leq \delta$, i.e.,

$$2\epsilon + e\gamma\frac{4608h(\epsilon) + 11520\epsilon}{\lambda(1-\delta)(1-\epsilon)^2} \leq \delta.$$

Define $\widehat{\mu}_t$ recursively according to (14) with λ replaced by $\lambda(1-\delta)$ and $\widehat{\mu}_0 = 0$, i.e.,

$$\widehat{\mu}_{t+1}^2 = \lambda(1-\delta)\sum_{k=0}^{d}\frac{\widehat{\mu}_t^{2k}}{k!}.$$

Define $t^*(\delta,\lambda,\gamma)$ according to (27) with $M = 8\log(1/\epsilon)$, and $s^*(\delta,\lambda,\gamma)$ according to (33) with λ replaced by $\lambda(1-\delta)$. Then Theorem 1 with n and K replaced by $n(1-\delta)$ and $\lceil K(1-\delta)\rceil$ implies that as $n \to \infty$,

$$\mathbb{P}\left\{|\widehat{C}_k\triangle C_k^*| \leq \delta K \text{ for } 1 \leq k \leq 1/\delta\right\} \to 1.$$

Given $(C_k^*, \widehat{C}_k)$, each of the random variables $r_i\sqrt{n}$ for $i \in S_k$ is conditionally subgaussian with proxy variance at most $K\gamma$. Furthermore, on the event, $\mathcal{E}_k = \{|\widehat{C}_k\triangle C_k^*| \leq \delta K\}$,

$$|\widehat{C}_k \cap C_k^*| \geq |\widehat{C}_k| - |\widehat{C}_k\triangle C_k^*| = \lceil K(1-\delta)\rceil - |\widehat{C}_k\triangle C_k^*| \geq K(1-2\delta).$$

Therefore, on the event $\mathcal{E}_k$, for $i \in S_k \cap C^*$, $r_i\sqrt{n}$ has mean greater than or equal to $K(1-2\delta)\mu$, and for $i \in S_k \backslash C^*$, r_i has mean zero.

Define the following set by thresholding

$$C_o' = \{i \in [n] : r_i \geq (1-2\delta)\sqrt{\lambda}/2\}$$

The number of indices in S_k incorrectly classified by $C_o' \cap S_k$ satisfies (use $|S_k| = \delta n$):

$$\mathbb{E}\left[\left|(C_o' \cap S_k)\Delta(C^* \cap S_k)\right|\right] \leq \delta n e^{-\Omega(n/K)},$$

where the last inequality follows because r_i is subgaussian with proxy variance at most $\gamma K/n$. Summing over $k \in [1/\delta]$ yields $\mathbb{E}\left[|C_o'\Delta C^*|\right] \leq ne^{-\Omega(n/K)}$. By Markov's inequality,

$$\mathbb{P}\left\{|C_o'\Delta C^*| \geq K^2/n\right\} \leq \frac{n^2}{K^2} e^{-\Omega(n/K)} \overset{K=o(n)}{=} o(1).$$

Instead of C_o', Algorithm 2 outputs C' which selects the K indices in $[n]$ with the largest values of r_i. Applying the same argument as that at the end of the proof of Lemma 12, we get $|C^*\triangle C'| \leq 2|C^*\triangle C_o'| + ||C^*| - K|$, and hence $|C^*\triangle C'|/K \to 0$ in probability.

(Exact recovery) By the union bound and Chernoff's bound for subgaussian random variables, the maximum of m subgaussian random variables X_i with zero mean and proxy variance at most γ satisfies that

$$\begin{aligned}\mathbb{P}\left\{\max_{1\leq i\leq m} X_i \geq \sqrt{\gamma}\left(\sqrt{2\log m} + t\right)\right\} &\leq m\exp\left(-\left(\sqrt{2\log m}+t\right)^2/2\right) \\ &= \exp\left(-t\sqrt{2\log m} - t^2/2\right).\end{aligned}$$

It follows that $\max_{1\leq i\leq m} X_i$ is at most $\sqrt{2\gamma\log m} + o_P(1)$ as $m \to \infty$. Also, for $k \in [1/\delta]$, $|S_k \cap C^*| \leq |C^*| = K$ and $|S_k\backslash C^*| \leq |[n]\backslash C^*| = n - K$. Therefore,

$$\min_{i\in S_k\cap C^*} r_i\sqrt{n} \geq K(1-2\delta)\mu - \sqrt{2K\gamma\log K} + o_P(\sqrt{K}) \tag{35}$$

$$\max_{j\in S_k\backslash C^*} r_i\sqrt{n} \leq \sqrt{2K\gamma\log(n-K)} + o_P(\sqrt{K}). \tag{36}$$

Since k ranges over a finite number of values, namely, $[1/\delta]$, (35) and (36) continue to hold with left-hand sides replaced by $\min_{i\in C^*} r_i\sqrt{n}$ and $\max_{j\notin C^*} r_i\sqrt{n}$, respectively. Therefore, by the choice of δ, $\min_{i\in C^*} r_i\sqrt{n} > \max_{j\in[n]\backslash C^*} r_i\sqrt{n}$ with probability converging to one as $n \to \infty$ and so $C' = C^*$ with probability converging to one as well.

(Time complexity) The running time of Algorithm 2 is dominated by invoking Algorithm 1 for a constant number, $1/\delta$, of times, and the number of iterations within Algorithm 1 is $(t^* + s^*\log n)n^2$, with both t^* and $s^* \to \infty$ as either $\delta \to 0$ or $\lambda \to 1/e$. In particular, the threshold comparisons require $O(n^2)$ computations. Thus, the total complexity of Algorithm 2 is as stated in the theorem. ■

5. The Gaussian biclustering problem

We return to the biclustering problem where the goal is to locate a submatrix whose row and column support need not coincide. Consider the model (1) parameterized by $(n_1, n_2, K_1, K_2, \mu)$ indexed by a common n with $n \to \infty$. In Section 5.1 we present the information limits for weak and exact recovery for the Gaussian bicluster model. The sharp conditions given for exact recovery are from (Butucea et al., 2015), and calculations from (Butucea et al., 2015) with minor adjustment provide conditions for weak recovery as well. Section 5.2 shows how the optimized message passing algorithm and its analysis can be extended from the symmetric case to the asymmetric case for biclustering and compares its performance to the fundamental limits. As originally observed in (Hajek et al., 2017) for recovering the principal submatrix, the connection between weak and exact recovery via the voting procedure extends to the biclustering problem as well. Note that for the sake of simplicity, we focus on Gaussian biclustering where the noise matrix Z is Gaussian; the results can be readily extended to the case where Z has subguassian entries as we did in the symmetric case.

5.1 Information-theoretic limits for Gaussian biclustering

Information-theoretic conditions ensuring exact recovery of both C_1^* and C_2^* by the maximum likelihood estimator (MLE), i.e.,

$$(\widehat{C}_1^{\mathsf{MLE}}, \widehat{C}_2^{\mathsf{MLE}}) = \operatorname*{arg\,max}_{\substack{|C_1|=K_1\\|C_2|=K_2}} \sum_{\substack{i\in C_1\\ j\in C_2}} W_{ij}$$

are obtained in (Butucea et al., 2015). While (Butucea et al., 2015) does not focus on conditions for weak recovery, the calculations therein combined with the voting procedure for exact recovery described in (Hajek et al., 2017) in fact resolve the information limits for both weak and exact recovery in the bicluster Gaussian model. Throughout this section we assume that $K_i = o(n_i)$ for $i = 1, 2$. For the converse results we assume C_i^* is a subset of $[n_i]$ of cardinality K_i selected uniformly at random for $i = 1, 2$, with C_1^* independent of C_2^*. Let

$$\lambda_i = \frac{K_i^2\mu^2}{n_i}, \quad \text{for } i = 1, 2.$$

Theorem 13 (Weak recovery thresholds for Gaussian biclustering) *If*

$$\liminf_{n\to\infty} \frac{\mu\sqrt{K_1K_2}}{\sqrt{2(K_1\log(n_1/K_1) + K_2\log(n_2/K_2))}} > 1, \tag{37}$$

then both C_1^* and C_2^* can be weakly recovered by the MLE. Conversely, if both C_1^* and C_2^* can be weakly recovered by some estimator, then

$$\liminf_{n\to\infty} \frac{\mu\sqrt{K_1K_2}}{\sqrt{2(K_1\log(n_1/K_1) + K_2\log(n_2/K_2)}} \geq 1. \tag{38}$$

If C_2^* (C_1^*) can be weakly recovered, one can further obtain exact recovery of C_1^* (C_2^*) via a voting cleanup procedure similar to Algorithm 2; it uses the method of successive withholding. We give the voting procedure in Algorithm 3 for exact recovery of C_1^* based on weak recovery of C_2^*; exact recovery of C_2^* based on weak recovery of C_1^* is analogous.

Algorithm 3 Weak recovery of C_2^* plus cleanup for exact recovery of C_1^*

1: Input: $n_1, n_2, K_1, K_2 \in \mathbb{N}$, $\mu > 0$, $A \in \mathbb{R}^{n_1 \times n_2}$, $\delta \in (0,1)$ with $1/\delta, n_1\delta \in \mathbb{N}$.
2: (Partition) Partition $[n_1]$ into $1/\delta$ subsets S_k of size $n_1\delta$ randomly.
3: (Approximate recovery) For each $k = 1, \ldots, 1/\delta$, let A_k denote the restriction of A to the rows with index in S_k^c, run an estimator capable of weak recovery of C_2^* with input $(n_1(1-\delta), n_2, \lceil K_1(1-\delta)\rceil, K_2, \mu, A_k)$ which outputs $\widehat{C}_{2k}$.
4: (Clean up) For each $k = 1, \ldots, 1/\delta$ compute $r_i = \sum_{j \in \widehat{C}_{2k}} A_{ij}$ for all $i \in S_k$ and return C_1', the set of K indices in $[n_1]$ with the largest values of r_i.

Theorem 14 (Exact recovery thresholds for Gaussian biclustering)
If for some small $\delta > 0$, C_2^* can be weakly recovered even if a fraction δ of the rows of the matrix are hidden, and if

$$\liminf_{n\to\infty} \frac{\sqrt{K_2}\mu}{\sqrt{2\log K_1} + \sqrt{2\log n_1}} > 1, \tag{39}$$

then C_1^* can be exactly recovered by the voting procedure. Conversely, if C_1^* can be exactly recovered by some estimator, then

$$\liminf_{n\to\infty} \frac{\sqrt{K_2}\mu}{\sqrt{2\log K_1} + \sqrt{2\log n_1}} \geq 1. \tag{40}$$

Similarly, if for some small $\delta > 0$, C_1^* can be weakly recovered even if a fraction δ of the columns of the matrix are hidden, and if

$$\liminf_{n\to\infty} \frac{\sqrt{K_1}\mu}{\sqrt{2\log K_2} + \sqrt{2\log n_2}} > 1, \tag{41}$$

then C_2^* can be exactly recovered by the voting procedure. Conversely, if C_2^* can be exactly recovered by some estimator, then

$$\liminf_{n\to\infty} \frac{\sqrt{K_1}\mu}{\sqrt{2\log K_2} + \sqrt{2\log n_2}} \geq 1. \tag{42}$$

The proofs of Theorems 13 and 14 are given in Appendix D. The sufficient conditions involving δ in Theorem 14 require a certain robustness of the estimator for weak recovery. If the rows indexed by a set S, with $S \subset [n_1]$ and $|S| = \delta n_1$, are hidden, then the observed matrix has dimensions $n_1(1-\delta)\times n_2$ and the planted submatrix has $K_1 - |S \cap C_1^*| \approx K_1(1-\delta)$ rows and K_2 columns. It is shown in (Hajek et al., 2017, Section IV.B) that the MLE has this robustness property for weak recovery of a principal submatrix, and a similar extension can be established for weak recovery for biclustering. The estimator used is the MLE based on the assumption that the submatrix to be found has shape $K_1(1-\delta) \times K_2$. With that extension in hand, the following corollary is a consequence of the two theorems, and it recovers the main result of (Butucea et al., 2015).

Corollary 15 *If (37), (39), and (41) hold, then C_1^* and C_2^* can both be exactly recovered by the MLE. Conversely, if exact recovery is possible, then (38), (40), and (42) hold.*

We conclude this subsection with a few remarks on Theorems 13 and 14:

1. If $n_1 = n_2$ and $K_1 = K_2$, the sufficient conditions and the necessary conditions for weak and for exact recovery, respectively, are identical to those in (Hajek et al., 2017) for the recovery of a $K \times K$ principal submatrix with elevated mean, in a symmetric $n \times n$ Gaussian matrix. Basically, in the bicluster problem the data matrix provides roughly twice the information (because the matrix is not symmetric) and there is twice the information to be learned, namely C_1^* and C_2^* instead of only C^*, and the factors of two cancel to yield the same conditions. It therefore follows from (Hajek et al., 2017, Remark 7), that if $n_1 = n_2$ and $K_1 = K_2 \leq n_1^{1/9}$, then (37) implies (39) and (41); in this regime, (37) alone is the sharp condition for both weak and exact recovery.

2. If $\lambda_i = \frac{K_i^2\mu^2}{n_i}$ are two fixed positive constants and if $K_1 \asymp K_2$, then (37) holds for all sufficiently large n, so weak recovery is information theoretically possible. In contrast, our proof that the optimized message passing algorithm provides weak recovery in this regime requires $(\lambda_1, \lambda_2) \in \mathcal{G}$, where $\mathcal{G}$ is defined in (52) in the next subsection.

5.2 Message passing algorithm for the Gaussian biclustering model

Suppose $n_i \to \infty$ and $\Omega(\sqrt{n_i}) \leq K_i \leq o(n_i)$ for $i \in \{0,1\}$, as $n \to \infty$. The belief propagation algorithm and our analysis of it for recovery of a single set of indices can be naturally adapted to the biclustering model.

Let $f(\cdot, t) : \mathbb{R} \to \mathbb{R}$ be a scalar function for each iteration t. To be definite, we shall describe the algorithm such that at each iteration, the messages are passed either from the row indices to the column indices, or vice-versa, but not both. The messages are defined as follows for $t \geq 0$:

$$(t \text{ even}) \qquad \theta_{i\to j}^{t+1} = \frac{1}{\sqrt{n_2}} \sum_{\ell \in [n_2]\backslash\{j\}} W_{i\ell} f(\theta_{\ell\to i}^t, t), \quad \forall i \in [n_1], j \in [n_2] \tag{43}$$

$$(t \text{ odd}) \qquad \theta_{j\to i}^{t+1} = \frac{1}{\sqrt{n_1}} \sum_{\ell \in [n_1]\backslash\{i\}} W_{\ell j} f(\theta_{\ell\to j}^t, t), \quad \forall j \in [n_2], i \in [n_1], \tag{44}$$

with the initial condition $\theta_{\ell\to i}^0 = 0$ for $(\ell, i) \in [n_2] \times [n_1]$. Moreover, let the aggregated beliefs be given by

$$(t \text{ even}) \qquad \theta_i^{t+1} = \frac{1}{\sqrt{n_2}} \sum_{\ell \in [n_2]} W_{i\ell} f(\theta_{\ell\to i}^t, t), \quad \forall i \in [n_1] \tag{45}$$

$$(t \text{ odd}) \qquad \theta_j^{t+1} = \frac{1}{\sqrt{n_1}} \sum_{\ell \in [n_1]} W_{\ell j} f(\theta_{\ell\to j}^t, t), \quad \forall j \in [n_2]. \tag{46}$$

Recall $\lambda_i = \frac{K_i^2\mu^2}{n_i}$ for $i = 1, 2$. Suppose as $n \to \infty$, for t even (odd), θ_i^t is approximately $\mathcal{N}(\mu_t, \tau_t)$ for $i \in C_1^*$ ($i \in C_2^*$) and $\mathcal{N}(0, \tau_t)$ for $i \in [n_1]\backslash C_1^*$ ($i \in [n_2]\backslash C_2^*$). Then similar to

the symmetric case, the update equations of message passing and the fact that $\theta^t_{i\to j} \approx \theta^t_i$ for all i, j suggest the following state evolution equations for $t \geq 0$:

$$\mu_{t+1} = \begin{cases} \sqrt{\lambda_2}\mathbb{E}\left[f(\mu_t + \tau_t Z, t)\right] & t \text{ even} \\ \sqrt{\lambda_1}\mathbb{E}\left[f(\mu_t + \tau_t Z, t)\right] & t \text{ odd} \end{cases} \tag{47}$$

$$\tau_{t+1} = \mathbb{E}\left[f(\tau_t Z, t)^2\right]. \tag{48}$$

The optimal choice of f for maximizing the signal-to-noise ratio $\frac{\mu_{t+1}}{\tau_{t+1}}$ is again $f(x,t) = e^{x\mu_t - \mu_t^2}$. With this optimized f, we have $\tau_{t+1} = 1$ and the state evolution equations reduce to

$$\mu^2_{t+1} = \begin{cases} \lambda_2 e^{\mu_t^2} & t \text{ even} \\ \lambda_1 e^{\mu_t^2} & t \text{ odd} \end{cases} \tag{49}$$

with $\mu_0 = 0$.

To justify the state evolution equations, we rely on the method of moments, requiring f to be polynomial. Thus, we choose $f = f_d(\cdot, t)$ as per Lemma 7, which maximizes the signal-to-noise ratio among all polynomials with degree up to d. With $f = f_d$, we have $\tau_{t+1} = 1$ and the state evolution equations reduce to

$$\widehat{\mu}^2_{t+1} = \begin{cases} \lambda_2 G_d(\widehat{\mu}_t^2) & t \text{ even} \\ \lambda_1 G_d(\widehat{\mu}_t^2) & t \text{ odd} \end{cases} \tag{50}$$

where $G_d(\mu) = \sum_{k=0}^d \frac{\mu^k}{k!}$.

Combining message passing with spectral cleanup, we obtain the following algorithm for estimating C_1^* and C_2^*.

Algorithm 4 Message passing for biclustering

1: Input: $n_1, n_2, K_1, K_2 \in \mathbb{N}$, $\mu > 0$, $W \in \mathbb{R}^{n_1 \times n_2}$, $d^* \in \mathbb{N}$, $t^* \in 2\mathbb{N}$, and $s^* > 0$.
2: Initialize: $\theta^0_{\ell\to i} = 0$ for $(\ell, i) \in [n_2] \times [n_1]$. For $t \geq 0$, define the sequence of degree-d^* polynomials $f_{d^*}(\cdot, t)$ as per Lemma 7 and $\widehat{\mu}_t$ according to (50).
3: Run t^* iterations of message passing as in (43) and (44) with $f = f_{d^*}$ and compute $\theta_i^{t^*}$ for all $i \in [n_1]$ as per (45) and $\theta_j^{t^*+1}$ for all $j \in [n_2]$ as per (46).
4: Find the sets $\widetilde{C}_1 = \{i \in [n_1] : \theta_i^{t^*} \geq \widehat{\mu}_{t^*}/2\}$ and $\widetilde{C}_2 = \{j \in [n_2] : \theta_j^{t^*+1} \geq \widehat{\mu}_{t^*+1}/2\}$.
5: (Cleanup via power method) Denote the restricted matrix $W_{\widetilde{C}_1\widetilde{C}_2}$ by $\widetilde{W}$. Sample u^0 uniformly from the unit sphere in $\mathbb{R}^{|\widetilde{C}_1|}$ and compute $u^{t+2} = \widetilde{W}\widetilde{W}^\top u^t / \|\widetilde{W}\widetilde{W}^\top u^t\|$, for t even and $0 \leq t \leq 2\lceil s^* \log(n_1 n_2)\rceil - 2$. Let $\widehat{u} = u^{2\lceil s^* \log(n_1 n_2)\rceil}$. Return $\widehat{C}_1$, the set of K_1 indices i in $\widetilde{C}_1$ with the largest values of $|\widehat{u}_i|$. Compute the power iteration with $\widetilde{W}^\top\widetilde{W}$ for odd values of t and return $\widehat{C}_2$ similarly.

We now turn to the performance of Algorithm 4. Let

$$\mathcal{G} = \{(\lambda_1, \lambda_2) : \mu_t \to \infty\}, \tag{51}$$

$$\mathcal{G}_d = \{(\lambda_1, \lambda_2) : \widehat{\mu}_t \to \infty\}. \tag{52}$$

As $d \to \infty$, $G_d(\mu) \to e^\mu$ uniformly over bounded intervals. It suggests that if $(\lambda_1, \lambda_2) \in \mathcal{G}$, then there exists a $d^*(\lambda_1, \lambda_2)$ such that $(\lambda_1, \lambda_2) \in \mathcal{G}_{d^*}$ and hence $\widehat{\mu}_t \to \infty$ as $t \to \infty$. The following lemma confirms this intuition.

Lemma 16 *For $d \geq 1$, $\mathcal{G}_d \subset \mathcal{G}_{d+1}$ with $\mathcal{G}_1 = \{(\lambda_1, \lambda_2) : \lambda_1\lambda_2 \geq 1\}$, and $\cup_{d=1}^\infty \mathcal{G}_d = \mathcal{G}$.*

Proof By definition, $G_1(x) = 1 + x$ and thus for t even, $\widehat{\mu}^2_{t+2} = \lambda_1(1 + \lambda_2(1 + \widehat{\mu}_t^2))$. As a consequence, $\widehat{\mu}_t \to \infty$ if and only if $\lambda_1\lambda_2 \geq 1$, proving the claim for $\mathcal{G}_1$. Let $\phi_d(x) \triangleq \lambda_1 G_d(\lambda_2 G_d(x))$ so that $\widehat{\mu}^2_{t+2} = \phi_d(\widehat{\mu}_t^2)$ for t even . The fact $\mathcal{G}_d \subset \mathcal{G}_{d+1} \subset \mathcal{G}$ follows from the fact $\phi_d(x)$ is increasing in d and $\phi_d(x) < \phi(x)$, where ϕ is defined in Remark 17. To prove $\cup_{d=1}^\infty \mathcal{G}_d = \mathcal{G}$, fix $(\lambda_1, \lambda_2) \in \mathcal{G}$. It suffices to show that $(\lambda_1, \lambda_2) \in \mathcal{G}_d$ for d sufficiently large. Since $\phi_2(x)/x^2 \to \infty$ as $x \to \infty$, there exists an absolute constant $x_0 > 1$ such that $\phi_d(x) \geq x^2$ whenever $x \geq x_0$ and $d \geq 2$. Let t_0 be an even number such that $\mu^2_{t_0} > x_0$. Since $\phi_d(x)$ converges to $\phi(x)$ uniformly on bounded intervals, it follows that the first $t_0/2$ iterates using ϕ_d converge to the corresponding iterates using ϕ. So, for d large enough, $\widehat{\mu}^2_{t_0} > x_0$, and hence, for such d, $\widehat{\mu}^2_t \to \infty$ as $t \to \infty$, so $(\lambda_1, \lambda_2) \in \mathcal{G}_d$. ■

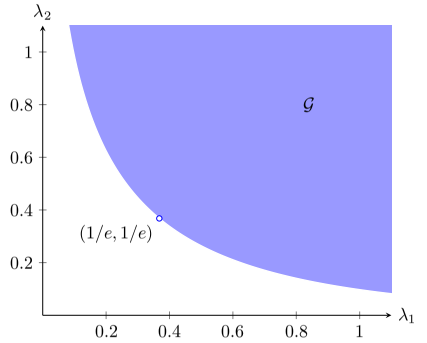


Figure 1: Required signal-to-noise ratios by Algorithm 4 for biclustering.

Remark 17 *Clearly $\mathcal{G}$ is an open subset of $\mathbb{R}^2_+$ and $\mathcal{G}$ is an upper closed set. Let $\partial\mathcal{G}$ denote its boundary and let $\phi(x) \triangleq \lambda_1 e^{\lambda_2 e^x}$, so that $\mu^2_{t+2} = \phi(\mu_t^2)$ for t even. Note that $(\lambda_1, \lambda_2) \in \partial\mathcal{G}$ if and only if the function is such that for some $x > 0$, $\phi(x) = x$ and $\phi'(x) = 1$. Since $\phi'(x) = \phi(x)y$, where $y = \lambda_2 e^x$, it follows that $xy = 1$ where $x = \lambda_1 e^y$. Therefore, it is convenient to express the boundary of $\mathcal{G}$ in the parametric form*

$$\partial\mathcal{G} = \{(xe^{-1/x}, x^{-1}e^{-x}) : x > 0\}.$$

It follows that $(1/e, 1/e) \in \partial G$ and $\{(\lambda_1, \lambda_2) \in \mathbb{R}^2_+ : \lambda_1\lambda_2 \geq e^{-2}\}\backslash\{(1/e, 1/e)\} \subset \mathcal{G}$ (see Fig. 1 for an illustration). Boundaries of $\mathcal{G}_d$ can be determined similar to (25) (see Fig. 2 for plots).

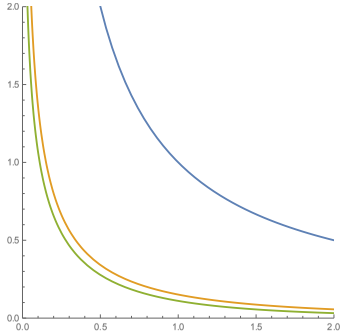


Figure 2: Boundaries of the regions $\mathcal{G}_d$ for $d = 1, 2, 3$; as d increases, $\mathcal{G}_d$ converges to $\mathcal{G}$ in Fig. 1.

The correctness proof for the spectral clean-up procedure in Algorithm 4 is given by Lemma 18 below with s^* defined by (56); it is similar to Lemma 12 used in Theorem 1 but applies to rectangular matrices and uses singular value decomposition.

Lemma 18 *Suppose*

$$\frac{\mu\sqrt{K_1 K_2}}{\sqrt{n_1}+\sqrt{n_2}} \geq \frac{1}{c_0} \tag{53}$$

for some $c_0 > 0$. For $i = 1, 2$, suppose that $\frac{|C_i^*|}{K_i} \to 1$ in probability and $\widetilde{C}_i$ is a set (possibly depending on W) such that

$$\frac{1}{K_i}|\widetilde{C}_i \cap C_i^*| \geq 1 - \epsilon \tag{54}$$

$$K_i(1-\epsilon) \leq |\widetilde{C}_i| \leq n_i\epsilon \tag{55}$$

hold for some $0 < \epsilon < \epsilon_0$, where ϵ_0 depends only on c_0. Let

$$s^* = \left(\log \frac{1-\epsilon-3c_0\sqrt{h(\epsilon)+\epsilon}}{3c_0\sqrt{h(\epsilon)+\epsilon}}\right)^{-1} \tag{56}$$

where $h(\epsilon) \triangleq \epsilon \log \frac{1}{\epsilon} + (1-\epsilon)\log\frac{1}{1-\epsilon}$ is the binary entropy function. Then $\widehat{C}_i$ returned by Algorithm 4 satisfies $|\widehat{C}_i \Delta C_i^*| \leq \eta(\epsilon) K_i$ for $i = 1, 2$, with probability converging to one as $n \to \infty$, where

$$\eta(\epsilon) = 2\epsilon + 650c_0^2\,\frac{h(\epsilon)+\epsilon}{(1-\epsilon)^2}. \tag{57}$$

With Lemma 18, we are ready to show that the bicluster message passing algorithm (Algorithm 4) approximately recovers C_1^* and C_2^*, provided that $(\lambda_1, \lambda_2) \in \mathcal{G}$.

Theorem 19 *Fix $\lambda_1, \lambda_2 > 0$. Suppose $\frac{K_i^2\mu^2}{n_i} \to \lambda_i$, $K_1 \asymp K_2$, and $\Omega(\sqrt{n_i}) \leq K_i \leq o(n_i)$ as $n \to \infty$, for $i = 1, 2$. Consider the model (1) with $|C_i^*|/K_i \to 1$ in probability as $n \to \infty$. Suppose $(\lambda_1, \lambda_2) \in \mathcal{G}$ and define $d^*(\lambda_1, \lambda_2)$ as in (59). For every $\eta \in (0, 1)$, there exist explicit positive constants t^*, s^* depending on $(\lambda_1, \lambda_2, \eta)$ such that Algorithm 4 returns $|\widehat{C}_i \cap C_i^*| \geq (1-\eta)K_i$ for $i = 1, 2$ with probability converging to 1 as $n \to \infty$, and the total running time is bounded by $c(\eta, \lambda_1, \lambda_2) n_1 n_2 \log(n_1 n_2)$, where $c(\eta, \lambda_1, \lambda_2) \to \infty$ as either $\eta \to 0$ or (λ_1, λ_2) approaches $\partial\mathcal{G}$.*

Remark 20 (Exact biclustering via message passing) *If the assumptions of Theorem 19 hold and the voting condition* (39) *(respectively,* (41)*) holds, then C_1^* (respectively, C_2^*) can be exactly recovered by message passing plus a voting procedure as described in Algorithm 3. Similar to the analysis in the symmetric case, whenever information-theoretic sufficient conditions for exact recovery* (39)–(41) *imply the sufficient condition of message passing for weak recovery, i.e., $(\lambda_1, \lambda_2) \in \mathcal{G}$ defined in (52), there is no computational gap for exact recovery.*

To be more precise, consider $K_i = \frac{\rho_i n}{\log n}$ for $i = 1, 2$. Then (39) *and* (41) *are equivalent to $\lambda_i > 8\rho_i$. Thus, whenever K_1 and K_2 are large enough so that $(8\rho_1, 8\rho_2)$ lies in the closure $\boldsymbol{cl}(\mathcal{G})$, or more generally,*

$$\left(\liminf_{n\to\infty} \frac{K_1 \log n_1}{n_1},\ \liminf_{n\to\infty} \frac{K_2 \log n_2}{n_2}\right) \in \frac{1}{8}\boldsymbol{cl}(\mathcal{G}), \tag{58}$$

then Algorithm 4 plus voting achieves the information-theoretic exact recovery threshold with optimal constants. This result can be viewed as a two-dimensional counterpart of (20) obtained for the symmetric case.

Proof [Proof of Theorem 19] The proof follows step-by-step that of Theorem 1; we shall point out the minor differences. Given λ_1 and λ_2, define

$$d^*(\lambda_1, \lambda_2) = \inf\{d \in \mathbb{N} : (\lambda_1, \lambda_2) \in \mathcal{G}_d\}. \tag{59}$$

By the assumptions of Theorem 19, there exists $c_0 > 0$ so that (53) holds. Given any $\eta \in (0, 1)$, choose an arbitrary $\epsilon \in (0, \epsilon_0)$ such that $\eta(\epsilon)$ defined in (57) is at most η. Notice that ϵ_0 is determined by c_0. Let $M = 8\log(1/\epsilon)$ and choose

$$t^*(\lambda_1, \lambda_2, M) = \inf\left\{t : \min\{\widehat{\mu}_t, \widehat{\mu}_{t+1}\} > M\right\}. \tag{60}$$

In view of Lemma 16 and the assumption that $(\lambda_1, \lambda_2) \in \mathcal{G}$, d^* is finite. Since $(\lambda_1, \lambda_2) \in \mathcal{G}_{d^*}$, it follows that $\widehat{\mu}_t \to \infty$ and thus $t^*(\lambda_1, \lambda_2, M)$ is finite.

The assumptions of Theorem 19 imply that $n_1 \asymp n_2$. Lemmas 27 - 29 therefore go through as before, with n in the upper bounds taken to be $\min\{n_1, n_2\}$, so that $\frac{1}{\sqrt{n_i}} \leq \frac{1}{\sqrt{n}}$. This modification then implies that Lemma 6, justifying the state evolution equations, goes through as before. See Section 6.1 for more details.

Finally, the proof is complete by invoking Lemma 18. ∎

6. Justification of state evolution equations

In this section we prove Lemma 6. Let $f(x,t) = \sum_{i=0}^{d} q_i^t x^i$ with $|q_i^t| \le C$ for a constant C. Let $\{A^t, t \ge 1\}$ be i.i.d. matrices distributed as A conditional on C^* and let $A^0 = A$. We now define a sequence of vectors $\{\xi^t, t \ge 1\}$ with $\xi^t \in \mathbb{R}^n$ given by

$$\xi_{i\to j}^{t+1} = \sum_{\ell \in [n]\setminus \{i,j\}} A_{\ell i}^t f(\xi_{\ell\to i}^t, t), \quad \forall j \ne i \in [n] \tag{61}$$

$$\xi_i^{t+1} = \sum_{\ell \in [n]\setminus\{i\}} A_{\ell i}^t f(\xi_{\ell\to i}^t, t)$$

$$\xi_{i\to j}^0 = 0. \tag{62}$$

In the definition of ξ^t, fresh samples, A^t, of A are used at each iteration, and thus the moments of ξ^t in the asymptotic limit are easier to compute than those of θ^t. Use of the fresh samples A^t does not make the messages $(\xi_{i\to\ell}^t : i \in [n]\setminus\ell)$ independent for fixed $\ell \in [n]$ and fixed $t \ge 2$, because at $t = 1$ the messages sent by any one vertex to all other vertices are statistically dependent, so at $t = 2$ the messages sent by all vertices are statistically dependent. However, we can take advantage of the fact that the contribution of each individual message is small in the limit as $n \to \infty$. Hence, we first prove that ξ^t and θ^t have the same moments of all orders as $n \to \infty$, and then prove the lemma using the method of moments.

The first step is to represent $(\theta_{i\to j}^t, \theta_i^t)$ and $(\xi_{i\to j}^t, \xi_i^t)$ as sums over a family of finite rooted labeled trees as shown by (Deshpande and Montanari, 2015, Lemma 3.3). We next introduce this family in detail. We shall consider rooted trees T of the following form. All edges are directed towards the root. The set of vertices and the set of (directed) edges in a tree T are denoted by $V(T)$ and $E(T)$, respectively. Each vertex has at most d children. The set of leaf vertices of T, denoted by $L(T)$, is the set of vertices with no children. Every vertex in the tree has a *label* which includes the *type* of the vertex, where the types are selected from $[n]$. The label of the root vertex consists of the type of the root vertex, and for every non-root vertex the label has two arguments, where the first argument in the label is the type of the vertex (in $[n]$), and the second one is the *mark* (in $\{0,\dots,d\}$). For a vertex v in T, let $\ell(v)$ denote its type, $r(v)$ its mark (if v is not the root), and $|v|$ its distance from the root in T. For clarity, we restate the definition of family of rooted labeled trees introduced in (Deshpande and Montanari, 2015, Definition 3.2).

Definition 21 *Let $\mathcal{T}^t$ denote the family of labeled trees T with exactly t generations satisfying the conditions:*

1. *The root of T has degree 1.*
2. *Any path $(v_1, v_2, \dots, v_k)$ in the tree is non-backtracking, i.e., the types $\ell(v_i), \ell(v_{i+1}), \ell(v_{i+2})$ are distinct for all i, k.*
3. *For a vertex u that is not the root or a leaf, the mark $r(u)$ is set to the number of children of v.*
4. *Note that $t = \max_{v\in L(T)} |v|$. All leaves u with $|u| \le t-1$ have mark 0.*

Let $\mathcal{T}_{i\to j}^t \subset \mathcal{T}^t$ be the subfamily satisfying the following additional conditions:

1. *The type of the root is i.*
2. *The root has a single child with type distinct from i and j.*

Similarly, let $\mathcal{T}_i^t \subset \mathcal{T}^t$ be the subfamily satisfying the following:

1. *The type of the root is i.*
2. *The root has a single child with type distinct from i.*

We point out that under the above definition, a vertex of a tree in $\mathcal{T}^t$ can have siblings of the same type and mark. Also two trees in $\mathcal{T}^t$ are considered to be the same if and only if the labels of all vertices are the same, with the understanding that the order of the children of any given vertex matters. In addition, the mark of a leaf u with $|u| = t$ is not specified and can possibly take any value in $\{0,\dots,d\}$. The following lemma is proved by induction on t and the proof can be found in (Deshpande and Montanari, 2015, Lemma 3.3).

Lemma 22

$$\theta_{i\to j}^t = \sum_{T\in\mathcal{T}_{i\to j}^t} A(T)\Gamma(T,\mathbf{q},t)\theta(T),$$

$$\theta_i^t = \sum_{T\in\mathcal{T}_i^t} A(T)\Gamma(T,\mathbf{q},t)\theta(T),$$

where[9]

$$A(T) \triangleq \prod_{u\to v\in E(T)} A_{\ell(u),\ell(v)},$$

$$\Gamma(T,\mathbf{q},t) \triangleq \prod_{u\to v\in E(T)} q_{r(u)}^{t-|u|},$$

$$\theta(T) \triangleq \prod_{u\to v\in E(T): u\in L(T)} (\theta_{\ell(u)\to\ell(v)}^0)^{r(u)}.$$

Similarly,

$$\xi_{i\to j}^t = \sum_{T\in\mathcal{T}_{i\to j}^t} \bar{A}(T)\Gamma(T,\mathbf{q},t)\theta(T),$$

$$\xi_i^t = \sum_{T\in\mathcal{T}_i^t} \bar{A}(T)\Gamma(T,\mathbf{q},t)\theta(T),$$

where

$$\bar{A}(T) \triangleq \prod_{u\to v\in E(T)} A_{\ell(u),\ell(v)}^{t-|u|}.$$

9. Often the initial messages for message passing are taken, with some abuse of notation, to have the form $\theta_{i\to j}^0 = \theta_i^0$ for all j, and then only the n variables θ_i^0 need to be specified. In that case, the expression for $\theta(T)$ simplifies to $\theta(T) \triangleq \prod_{u\in L(T)} (\theta_{\ell(u)}^0)^{r(u)}$.

Since the initial messages are zero, $f(\theta^0_{i\to j},0)=q^0_0$. *Thus, for notational convenience in what follows, we can assume without loss of generality that $f(x,0)\equiv q^0_0$, i.e., $f(x,0)$ is a degree zero polynomial.* With this assumption, it follows that for a labeled tree $T\in\mathcal{T}^t$, $\Gamma(T,\mathbf{q},t)=0$ unless the mark of every leaf of T is zero. If the mark of every leaf is zero, then $\theta(T)=1$, because in this case $\theta(T)$ is a product of terms of the form 0^0, which are all one, by convention. Therefore, $\Gamma(T,\mathbf{q},t)\theta(T)=\Gamma(T,\mathbf{q},t)$ for all $T\in\mathcal{T}_t$. Consequently, the factor $\theta(T)$ can be dropped from the representations of $\theta^t_{i\to j}$, θ^t_i, $\xi^t_{i\to j}$, and ξ^t_i given in Lemma 22. Applying Lemma 22, we can prove that all finite moments of θ^t_i and ξ^t_i are asymptotically the same. Before that, we need two key auxiliary lemmas.

Let $\phi(T)_{rs}$ denote the number of occurrences of edges $(u\to v)$ in the tree T with types $\ell(u),\ell(v)=\{r,s\}$.

Definition 23 *For $m\geq 1$ and given an m-tuple of trees $T_1,\ldots,T_m$, let G denote the undirected graph obtained by identifying the vertices of the same type in the trees and removing the edge directions. Let $E(G)$ denote the edge set of G. Then an edge (r,s) is in $E(G)$ if and only if $\sum_{\ell=1}^m\phi(T_\ell)_{rs}\geq 1$, i.e., the number of times covered is at least one. Let G_1 denote the restriction of G to the vertices in C^* and G_2 the restriction of G to the vertices in $[n]\backslash C^*$. Let $E(G_1)$ and $E(G_2)$ denote the edge set of G_1 and G_2, respectively. Let E_J denote the set of edges in G with one endpoint in G_1 and the other endpoint in G_2.*

Lemma 24 *Suppose an m-tuple of trees $T_1,\ldots,T_m\in\mathcal{T}^t$ has α edges in total, and there are k different edges (r,s) in $E(G_1)$ which are covered exactly once, i.e., $\sum_{\ell=1}^m\phi(T_\ell)_{rs}=1$. Then*

$$\left|\mathbb{E}\left[\prod_{\ell=1}^m A(T_\ell)\right]\right|\leq c\mu^k n^{-\alpha/2}$$

for a consant c independent of n. The same conclusion also holds when replacing $A(T_\ell)$ by $\bar{A}(T_\ell)$.

Proof By the definition of $A(T)$,

$$\begin{aligned}
\left|\mathbb{E}\left[\prod_{\ell=1}^m A(T_\ell)\right]\right| &= \left|\mathbb{E}\left[\prod_{j<j'}(A_{jj'})^{\sum_{\ell=1}^m\phi(T_\ell)_{jj'}}\right]\right| \leq \prod_{j<j'}\left|\mathbb{E}\left[(A_{jj'})^{\sum_{\ell=1}^m\phi(T_\ell)_{jj'}}\right]\right| \\
&= \left(\frac{\mu}{\sqrt{n}}\right)^k \prod_{j<j':\sum_{\ell=1}^m\phi(T_\ell)_{jj'}\geq 2}\left|\mathbb{E}\left[(A_{jj'})^{\sum_{\ell=1}^m\phi(T_\ell)_{jj'}}\right]\right| \\
&\leq \left(\frac{\mu}{\sqrt{n}}\right)^k \prod_{j<j':\sum_{\ell=1}^m\phi(T_\ell)_{jj'}\geq 2}\mathbb{E}\left[|A_{jj'}|^{\sum_{\ell=1}^m\phi(T_\ell)_{jj'}}\right] \\
&\leq c\left(\frac{\mu}{\sqrt{n}}\right)^k \prod_{j<j':\sum_{\ell=1}^m\phi(T_\ell)_{jj'}\geq 2}\left(\frac{1}{\sqrt{n}}\right)^{\sum_{\ell=1}^m\phi(T_\ell)_{jj'}} \\
&= c\mu^k n^{-\alpha/2},
\end{aligned}$$

where the last inequality follows because Z_{ij} are zero-mean subgaussian random variables with proxy variance γ and consequently for $1\leq p\leq\alpha$, $\mathbb{E}\left[\left|\frac{Z_{ij}}{\sqrt{n}}\right|^p\right]\leq cn^{-p/2}$ and $\mathbb{E}\left[\left|\frac{Z_{ij}+\mu}{\sqrt{n}}\right|^p\right]\leq cn^{-p/2}$ where c is a finite constant depending on γ,α, and $\mu_{\max}$, where $\mu_{\max}$ is an upper bound on μ for all n, which is finite[10] by the assumptions that $K=\Omega(\sqrt{n})$ and $\lambda=\Theta(1)$. ■

We next define an equivalence relation on the family of m-tuples of trees in $\mathcal{T}^t$, which is useful for enumerating such m-tuples. Fix a set $D\subset[n]$ of distinguished types; the notion of equivalence depends on D. In this paper when we focus on the messages formed by one vertex $\{i\}$ we take $D=\{i\}$ and when we focus on the covariance of messages formed by two vertices, i and j, we take $D=\{i,j\}$.

Definition 25 *For $D\subset[n]$, two m-tuples of trees in $\mathcal{T}^t$ are* equivalent *(relative to D) if there is a permutation of the set of types $[n]$ such that i maps to i for each $i\in D$, C^* maps to C^*, so also $[n]\backslash C^*$ maps to $[n]\backslash C^*$, such that the following is true: the second m-tuple of trees is obtained by applying the permutation to the types of the vertices of the first m-tuple of trees.*

To clarify, if two m-tuples of trees are equivalent, in particular, the marks of the two m-tuples must be the same, and the set of vertices with type $i\in D$ is the same. Recall that for trees to be considered equal, the order of children matters. The same is true when considering trees to be equivalent relative to D. Thus, for example, if (T_1,T_2) is equivalent to (T_1',T_2') and if (T_1,T_2) has the following property: the first child of the first child of the root in T_1 has the same type as the third child of the second child of the root in T_2, then (T_1',T_2') must have the same property. Furthermore, if the common type for those two vertices in (T_1,T_2) is some $i\in D$, then those two vertices in (T_1',T_2') must also have type i. If the common type for those two vertices in (T_1,T_2) is some $k\in C^*\backslash D$, then those two vertices in (T_1',T_2') must also have some common type $k'\in C^*\backslash D$.

Lemma 26 *For a given set of distinguished types D, let $\mathcal{S}$ denote the set of equivalence classes on the family of m-tuples of trees in $\mathcal{T}^t$. Then $|\mathcal{S}|\leq c$ for a constant c dependent on only $m,t,d,|D|$ (not on n). Moreover, fix any equivalence class $S\in\mathcal{S}$ and a representative m-tuple of trees $(T_1,\ldots,T_m)$ which has t_1 and t_2 distinct types in $C^*\backslash D$ and $[n]\backslash(C^*\cup D)$, respectively. Then $|S|\leq K^{t_1}n^{t_2}$.*

Proof The total number of vertices of an m-tuple $(T_1,\ldots,T_m)$ is bounded by a function of m,t,d alone and thus independently of n, therefore so are the number of ways to partition these vertices into subsets of vertices with the same type. For each such subset, we need to designate whether the type of the vertices in this subset is one of the distinguished types $i\in D$, or is in $C^*\backslash D$, or is in $[n]\backslash(C^*\cup D)$. The total number of distinct such designations is bounded by a function of $m,t,d,|D|$ independently of n. Finally, we need to assign marks to the vertices of the trees, and the number of distinct assignments is bounded by a function of m,t,d alone. Hence, $|\mathcal{S}|$ is bounded by a function of m,t,d alone.

10. This is where the assumption $K=\Omega(\sqrt{n})$ is used because $\frac{K^2\mu^2}{n}$ is assumed to be a constant λ.

Fix a given equivalence class $S \in \mathcal{S}$ and a representative m-tuple of trees $(T_1, \ldots, T_m)$ in S. Consider all the types from $[n]\backslash D$ that appear at least once for some vertex of some tree in the m-tuple. Then t_1 is the number of such types in $C^*\backslash D$ and t_2 is the number of such types in $[n]\backslash(C^* \cup D)$. The cardinality of S is at most the product of the number of partial permutations of length t_1 of elements chosen from $C^*\backslash D$, times the number of partial permutations of length t_2 of elements chosen from $[n]\backslash(C^* \cup D)$. The conclusion follows. ■

Combining Lemmas 22, 24, and 26, we have the following lemma, showing that all finite moments of θ_i^t and ξ_i^t are asymptotically close.

Lemma 27 *For any $t \geq 1$, there exists a constant c independent of n and dependent on m, t, d, C such that for any $i \in [n]$:*

$$\left|\mathbb{E}\left[(\theta_i^t)^m\right] - \mathbb{E}\left[(\xi_i^t)^m\right]\right| \leq cn^{-1/2}.$$

Proof As explained right after Lemma 22, the assumption that $f(x, 0) \equiv q_0^0$ implies that the factor $\theta(T)$ can be dropped in the representations given in Lemma 22. Therefore, it follows from Lemma 22 that for $t \geq 1$,

$$\mathbb{E}\left[(\theta_i^t)^m\right] = \sum_{T_1,\ldots,T_m \in \mathcal{T}_i^t} \prod_{\ell=1}^m \Gamma(T_\ell, \mathbf{q}, t)\mathbb{E}\left[\prod_{\ell=1}^m A(T_\ell)\right],$$
$$\mathbb{E}\left[(\xi_i^t)^m\right] = \sum_{T_1,\ldots,T_m \in \mathcal{T}_i^t} \prod_{\ell=1}^m \Gamma(T_\ell, \mathbf{q}, t)\mathbb{E}\left[\prod_{\ell=1}^m \bar{A}(T_\ell)\right].$$

Because the coefficients in the polynomial are bounded by C and there are m trees with each tree containing at most $1+d+\cdots+d^{t-1} \leq (d+1)^t$ edges, $|\prod_{\ell=1}^m \Gamma(T_\ell, \mathbf{q}, t)| \leq C^{m(d+1)^t}$. Therefore, it suffices to show

$$\sum_{T_1,\ldots,T_m \in \mathcal{T}_i^t} \left|\mathbb{E}\left[\prod_{\ell=1}^m A(T_\ell)\right] - \mathbb{E}\left[\prod_{\ell=1}^m \bar{A}(T_\ell)\right]\right| \leq cn^{-1/2}.$$

In the following, let c denote a constant only depending on m, t, d and its value may change line by line. Recall (G, G_1, G_2) obtained from a given m-tuple of trees $T_1, \ldots, T_m$ as defined in Definition 23. We partition set $\{(T_1, \ldots, T_m) : T_\ell \in \mathcal{T}_i^t\}$ as a union of four disjoint sets $Q \cup R_1 \cup R_2 \cup R_3$, where

1. Q consists of m-tuples of trees $(T_1, \ldots, T_m)$ such that there exists an edge (r, s) in $E(G_2) \cup E_J$ which is covered exactly once.
2. R_1 consists of m-tuples of trees $(T_1, \ldots, T_m)$ such that all edges in $E(G_2) \cup E_J$ are covered at least twice and at least one of them is covered at least 3 times.
3. R_2 consists of m-tuples of trees $(T_1, \ldots, T_m)$ such that each edge in $E(G_2) \cup E_J$ is covered exactly twice and the graph G contains a cycle.

4. R_3 consists of m-tuples of trees $(T_1, \ldots, T_m)$ such that each edge in $E(G_2) \cup E_J$ is covered exactly twice and the graph G is a tree.

Fix any $(T_1, \ldots, T_m) \in Q$ and let (r, s) be an edge in $E(G_2) \cup E(J)$ which is covered exactly once. Since $\mathbb{E}[A_{rs}] = 0$ and A_{rs} appears in the product $\prod_{\ell=1}^m A(T_\ell)$ once, it follows that $\mathbb{E}[\prod_{\ell=1}^m A(T_\ell)] = 0$. Similarly, $\mathbb{E}\left[\prod_{\ell=1}^m \bar{A}(T_\ell)\right] = 0$. Therefore, it is sufficient to show that for $j = 1, 2, 3$,

$$\sum_{(T_1,\ldots,T_m) \in R_j} \left|\mathbb{E}\left[\prod_{\ell=1}^m A(T_\ell)\right] - \mathbb{E}\left[\prod_{\ell=1}^m \bar{A}(T_\ell)\right]\right| \leq cn^{-1/2}.$$

First consider R_1. Further, divide R_1 according to the total number of edges in $T_1, \ldots, T_m$ and the number of edges in $E(G_1)$ which are covered exactly once. In particular, for $\alpha = 1, \ldots, m(d+1)^t$ and $k = 0, 1, \ldots, \alpha$, let $R_{1,\alpha,k}$ denote the subset of R_1 consisting of m-tuples of trees $T_1, \ldots, T_m$ such that there are α edges in $T_1, \ldots, T_m$ and there are k edges in $E(G_1)$ which are covered exactly once. It suffices to show that

$$\sum_{(T_1,\ldots,T_m) \in R_{1,\alpha,k}} \left|\mathbb{E}\left[\prod_{\ell=1}^m A(T_\ell)\right] - \mathbb{E}\left[\prod_{\ell=1}^m \bar{A}(T_\ell)\right]\right| \leq cn^{-1/2}. \tag{63}$$

Fix α, k and an m-tuple of trees $(T_1, \ldots, T_m) \in R_{1,\alpha,k}$. It follows from Lemma 24 that

$$\left|\mathbb{E}\left[\prod_{\ell=1}^m A(T_\ell)\right]\right| \leq c\mu^k n^{-\alpha/2}. \tag{64}$$

Throughout the proof of this lemma, let $D = \{i\}$ be the set of distinguished types for defining equivalence classes as specified in Definition 25. We consider breaking $R_{1,\alpha,k}$ down into a large number of smaller sets, where each set is an equivalence class. While large, it follows from Lemma 26 that the number of these smaller sets depends on m, t, d, but not on n. Hence, it suffices to upper bound $|S|$ for any given equivalence class $S \subset R_{1,\alpha,k}$. It follows from Lemma 26 that $|S| \leq K^{n_1} n^{n_2}$, where n_1 is the number of vertices in G_1 with types in $C^*\backslash\{i\}$ and n_2 is the number of vertices in G_2 with types in $[n]\backslash(C^* \cup \{i\})$.

We further upper bound n_1 and n_2. The graph G is connected (because all the trees have a root of type i), so $n_1 + n_2$ (the number of vertices of G minus one) is less than or equal to the number of edges in G. The number of edges in G is at most $k + \frac{\alpha-k-1}{2}$ because there are k edges in G covered once, and the rest are covered at least twice, with one edge covered at least three times. So $n_1 + n_2 \leq k + \frac{\alpha-k-1}{2}$. Also, since k of the edges in G have both endpoints in C^*, and the vertices of G_2 have types in $[n] - C^*$, there are at most $\frac{\alpha-k-1}{2}$ edges in G with at least one endpoint in G_2. Moreover, since G is connected, each connected component in G_2 is connected by at least one edge to a vertex in G_1. Therefore, $n_2 \leq |V(G_2)| \leq \frac{\alpha-k-1}{2}$. The bound $K^{n_1} n^{n_2}$ is maximized subject to $n_1 + n_2 \leq k + \frac{\alpha-k-1}{2}$ and $n_2 \leq \frac{\alpha-k-1}{2}$ by letting equality hold in both constraints, yielding $|S| \leq K^k n^{\frac{\alpha-k-1}{2}}$. Combining with (64) shows that

$$\sum_{(T_1,\ldots,T_m) \in S} \left|\mathbb{E}\left[\prod_{\ell=1}^m A(T_\ell)\right]\right| \leq c\mu^k n^{-\alpha/2} K^k n^{\frac{\alpha-k-1}{2}} = c\left(\frac{\mu K}{\sqrt{n}}\right)^k n^{-1/2} \leq cn^{-1/2}, \tag{65}$$

where we've used the fact that $\frac{\mu K}{\sqrt{n}}$ is bounded independently of n. In a similar way, it can be shown that

$$\sum_{(T_1,\ldots,T_m)\in S}\left|\mathbb{E}\left[\prod_{\ell=1}^{m}\bar{A}(T_\ell)\right]\right|\le cn^{-1/2}$$

and thus

$$\sum_{(T_1,\ldots,T_m)\in S}\left|\mathbb{E}\left[\prod_{\ell=1}^{m}A(T_\ell)\right]-\mathbb{E}\left[\prod_{\ell=1}^{m}\bar{A}(T_\ell)\right]\right|\le cn^{-1/2}. \tag{66}$$

Since the number of equivalence classes S does not depend on n, (63) follows.

Next consider R_2. The previous argument carries over with a minor adjustment on the step of upper bounding n_1 and n_2. In particular, define $R_{2,\alpha,k}$ accordingly as $R_{1,\alpha,k}$ and then consider an equivalence class $S\subset R_{2,\alpha,k}$ corresponding to some representative m-tuple in $R_{2,\alpha,k}$. The number of edges in G is at most $k+\frac{\alpha-k}{2}$ because there are k edges in G covered once, and the rest are covered at least twice. Since G has n_1+n_2+1 vertices, is connected, and has a cycle, n_1+n_2 is less than or equal to the number of edges of G minus one, so $n_1+n_2\le k+\frac{\alpha-k-2}{2}$. Also, since k of the edges in G have both endpoints with types in C^*, and V_2 has types in $[n]-C^*$, there are at most $\frac{\alpha-k}{2}$ edges in G with at least one endpoint in V_2. Moreover, since G is connected, each connected component in G_2 is connected by at least one edge to a vertex in G_1. Therefore, $n_2\le|V(G_2)|\le\frac{\alpha-k}{2}$. The bound $K^{n_1}n^{n_2}$ is maximized subject to these constraints by letting equality hold in both constraints, yielding $|S|\le K^{k-1}n^{\frac{\alpha-k}{2}}$. So $|S|\mu^k n^{-\alpha/2}\le\left(\frac{\mu K}{\sqrt{n}}\right)^k/K\le c/K\le cn^{-1/2}$, and the reminder of the proof for bounding the contribution of R_2 is the same as for R_1 above.

Finally, consider R_3. It suffices to establish the following claim. The claim is that for any m-tuple such that G has no cycles, if two directed edges $(a\to b)$ and $(c\to d)$ map to the same edge in G, then they are at the same level in their respective trees (their trees might be the same). Indeed, if the claim is true, then for any m-tuple $(T_1,\ldots,T_m)$ in R_3 and any pair $\{r,s\}\subset[n]$, A^t_{rs} appears in $\prod_{\ell=1}^m\bar{A}(T_\ell)$ for at most one value of t, so that $\mathbb{E}\left[\prod_{\ell=1}^m A(T_\ell)\right]=\mathbb{E}\left[\prod_{\ell=1}^m\bar{A}(T_\ell)\right]$.

We now prove the claim. Let $\{r,s\}$ denote the edge in G covered by both $(a\to b)$ and $(c\to d)$, i.e. $\{\ell(a),\ell(b)\}=\{\ell(c),\ell(d)\}=\{r,s\}$. First consider the case that $\ell(b)=\ell(d)$. Let $u_1,\ldots,u_k$ denote the directed path in the tree containing b that goes from b to the root of that tree, so $b=u_1$ and u_k is the root of the tree. Since there are no cycles in G, and hence no cycles in the set of edges $\{\{\ell(u_1),\ell(u_2)\},\ldots,\{\ell(u_{k-1}),\ell(u_k)\}\}$, (viewed as a simple set, i.e. with duplications removed) it follows from the non-backtracking property that $\ell(u_1),\ldots,\ell(u_k)$ are distinct vertices in G. That is, $(\ell(u_1),\ldots,\ell(u_k))$ is a simple path in G. Similarly, let $v_1,\ldots,v_{k'}$ denote the path in the tree containing d that goes from d to the root of that tree, so $d=v_1$ and $v_{k'}$ is the root of that tree. As for the first path, $(\ell(v_1),\ldots,\ell(v_{k'}))$ is also a simple path in G. Since the roots of all m trees have the same type, $\ell(u_k)$ and $\ell(v_{k'})$ are the same vertex in G. Therefore, $(\ell(u_1),\ldots,\ell(u_k),\ell(v_{k'-1}),\ldots,\ell(v_1))$ is a closed walk in G that is the concatenation of two simple paths. Since G has no cycles those two paths must be reverses of each other. That is, $k=k'$ and $\ell(u_j)=\ell(v_j)$ for all j, and hence $(a\to b)$ and $(c\to d)$ are at the same level in their trees.

Consider the remaining case, namely, that $\ell(b)=\ell(c)$. Let $u_1,\ldots,u_k$ be defined as before, and let $v_1,\ldots,v_{k'}$ denote the path in the tree containing c that goes from c to the root of that tree, so $c=v_1$, $d=v_2$, and $v_{k'}$ is the root of that tree. Arguing as before yields that $k=k'$ and $\ell(u_j)=\ell(v_j)$ for $1\le j\le k$. Note that $k'\ge 2$ and so $k\ge 2$ and $\ell(u_2)=\ell(v_2)=\ell(d)=\ell(a)$. Thus, the types along the directed path $a\to u_1\to u_2$ within one of the trees violates the non-backtracking property, so the case $\ell(b)=\ell(c)$ cannot occur. The claim is proved. This completes the proof of Lemma 27. ■

The second step is to compute the moments of ξ^t in the asymptotic limit $n\to\infty$. We need the following lemma to ensure that all moments of ξ^t are bounded by a constant independent of n.

Lemma 28 *For any $t\ge 1$, there exists a constant c independent of n and dependent on m,t,d,C,γ such that for any $i,j\in[n]$*

$$|\mathbb{E}\left[(\xi^t_{i\to j})^m\right]|\le c,\quad |\mathbb{E}\left[(\xi^t_i)^m\right]|\le c.$$

Proof We prove the claim for ξ^t_i; the claim for $\xi^t_{i\to j}$ follows by the similar argument. Since $\xi^0_{i\to j}=\theta^0_{i\to j}=0$ for all $i\in[n]$, it follows from Lemma 22 that

$$\mathbb{E}\left[(\xi^t_i)^m\right]=\sum_{T_1,\ldots,T_m\in\mathcal{T}_i^t}\prod_{\ell=1}^m\Gamma(T_\ell,\mathbf{q},t)\mathbb{E}\left[\prod_{\ell=1}^m\bar{A}(T_\ell)\right].$$

Recalling (G,G_1,G_2) defined as Definition 23 and following the same argument as used for proving Lemma 27, we can partition set $\{(T_1,\ldots,T_m):T_\ell\in\mathcal{T}_i^t\}$ as a union of four disjoint sets $Q\cup R_1\cup R_2\cup R_3$, and show that

$$\sum_{T_1,\ldots,T_m\in Q}\prod_{\ell=1}^m\Gamma(T_\ell,\mathbf{q},t)\mathbb{E}\left[\prod_{\ell=1}^m\bar{A}(T_\ell)\right]=0,$$

and

$$\sum_{T_1,\ldots,T_m\in R_1\cup R_2}\left|\prod_{\ell=1}^m\Gamma(T_\ell,\mathbf{q},t)\right|\left|\mathbb{E}\left[\prod_{\ell=1}^m\bar{A}(T_\ell)\right]\right|\le cn^{-1/2}.$$

Hence, we only need to check R_3. Again divide R_3 according to the total number of edges in $T_1,\ldots,T_m$ and the number of edges in $E(G_1)$ which are covered exactly once. In particular, $R_3=\cup_{1\le\alpha\le m(d+1)^t,0\le k\le\alpha}R_{3,\alpha,k}$, where $R_{3,\alpha,k}$ is defined in the similar way as $R_{1,\alpha,k}$. Furthermore, similar to $R_{1,\alpha,k}$, consider dividing $R_{3,\alpha,k}$ into a number of equivalence classes (defined relative to $D=\{i\}$), the number of which depends only on m,t,d, as shown in Lemma 26. To prove the lemma, it suffices to show that for any such equivalence class S,

$$\sum_{(T_1,\ldots,T_m)\in S}\left|\mathbb{E}\left[\prod_{\ell=1}^m\bar{A}(T_\ell)\right]\right|\le c.$$

Invoking Lemma 24, we have that

$$\left| \mathbb{E}\left[\prod_{\ell=1}^{m} \bar{A}(T_\ell)\right] \right| \le c\mu^k n^{-\alpha/2},$$

so

$$\sum_{(T_1,\ldots,T_m)\in S} \left| \mathbb{E}\left[\prod_{\ell=1}^{m} \bar{A}(T_\ell)\right] \right| \le c\mu^k n^{-\alpha/2}|S|.$$

Fix a representative m-tuple $(T_1,\ldots,T_m)$ for S. It follows from Lemma 26 that $|S| \le K^{n_1}n^{n_2}$, where n_1 is the number of vertices in G_1 with types in $C^*\backslash\{i\}$ and n_2 is the number of vertices in G_2 with types in $[n]\backslash(C^*\cup\{i\})$.

We can further bound n_1 and n_2 in the similar way as we did for $|R_{1,\alpha,k}|$, with the only adjustment being we cannot use the assumption that there exists at least one edge which is covered at least three times. There are n_1+n_2+1 vertices in the connected graph G and, since the m-tuple is in $R_{3,\alpha,k}$, there are at most $k+\frac{\alpha-k}{2}$ edges in G, so $n_1+n_2 \le k+\frac{\alpha-k}{2}$. Also, at most $\frac{\alpha-k}{2}$ edges of G have at least one endpoint in V_2 so $n_2 \le \frac{\alpha-k}{2}$. Therefore, $|S| \le K^{n_1}n^{n_2} \le K^k n^{\frac{\alpha-k}{2}}$. It follows that

$$\sum_{(T_1,\ldots,T_m)\in S} \left| \mathbb{E}\left[\prod_{\ell=1}^{m} \bar{A}(T_\ell)\right] \right| \le c\mu^k n^{-\alpha/2} K^k n^{\frac{\alpha-k}{2}} = c\left(\frac{K\mu}{\sqrt{n}}\right)^k \le c,$$

and the proof is complete. ■

We also need the following lemma to show the convergence of $\frac{1}{|C^*|}\sum_{i\in C^*}(\xi_i^t)^m$ in probability using the Chebyshev inequality.

Lemma 29 *For any $t \ge 1$, $m \ge 1$ and $i \in [n]$,*

$$\lim_{n\to\infty} \mathsf{var}\left(\frac{1}{K}\sum_{i\in C^*}(\xi_i^t)^m\right) = 0$$

$$\lim_{n\to\infty} \mathsf{var}\left(\frac{1}{K}\sum_{\ell\in C^*\backslash\{i\}}(\xi_{\ell\to i}^t)^m\right) = 0$$

$$\lim_{n\to\infty} \mathsf{var}\left(\frac{1}{n}\sum_{i\in[n]\backslash C^*}(\xi_i^t)^m\right) = 0$$

$$\lim_{n\to\infty} \mathsf{var}\left(\frac{1}{n}\sum_{\ell\in[n]\backslash(C^*\cup\{i\})}(\xi_{\ell\to i}^t)^m\right) = 0,$$

where the same also holds when replacing ξ^t by θ^t.

Proof We prove the first claim; the other claim follows by a similar argument. Notice that

$$\mathsf{var}\left(\frac{1}{K}\sum_{i\in C^*}(\xi_i^t)^m\right) = \frac{1}{K^2}\sum_{i,j\in C^*}\left(\mathbb{E}\left[(\xi_i^t)^m(\xi_j^t)^m\right] - \mathbb{E}\left[(\xi_i^t)^m\right]\mathbb{E}\left[(\xi_j^t)^m\right]\right).$$

There are K diagonal terms with $i=j$ in the last displayed equation and each diagonal term is bounded by a constant independent of n in view of Lemma 28. Hence, to prove the claim, it suffices to consider the cross terms. Since there are $\binom{K}{2}$ cross terms, we only need to show that for each cross term with $i\neq j$, $\mathbb{E}\left[(\xi_i^t)^m(\xi_j^t)^m\right] - \mathbb{E}\left[(\xi_i^t)^m\right]\mathbb{E}\left[(\xi_j^t)^m\right]$ converges to 0 as $n\to\infty$. Using the tree representation as shown by Lemma 22 yields

$$\begin{aligned}&\left|\mathbb{E}\left[(\xi_i^t)^m(\xi_j^t)^m\right] - \mathbb{E}\left[(\xi_i^t)^m\right]\mathbb{E}\left[(\xi_j^t)^m\right]\right| \\ &\le c\sum_{T_1,\ldots,T_m\in\mathcal{T}_i^t,T_1',\ldots,T_m'\in\mathcal{T}_j^t}\left|\mathbb{E}\left[\prod_{\ell=1}^{m}\bar{A}(T_\ell)\bar{A}(T_\ell')\right] - \mathbb{E}\left[\prod_{\ell=1}^{m}\bar{A}(T_\ell)\right]\mathbb{E}\left[\prod_{\ell=1}^{m}\bar{A}(T_\ell')\right]\right|,\end{aligned}$$

where c is a constant independent of n and dependent of m,t,d,γ.

Let (G,G_1,G_2) denote the undirected simple graphs obtained from $2m$-tuple of trees $(T_1,\ldots,T_m,T_1',\ldots,T_m')$ as defined in Definition 23. Notice that roots of $T_1,\ldots,T_m$ have type i and roots of $T_1',\ldots,T_m'$ have type j, so either G is disconnected with one component containing i and the other component containing j, or G is connected. In the former case, there is no edge $(r,s)\in E(G)$ which is covered by $T_1,\ldots,T_m$ and $T_1',\ldots,T_m'$ simultaneously and thus $\mathbb{E}\left[\prod_{\ell=1}^m\bar{A}(T_\ell)\bar{A}(T_\ell')\right] = \mathbb{E}\left[\prod_{\ell=1}^m\bar{A}(T_\ell)\right]\mathbb{E}\left[\prod_{\ell=1}^m\bar{A}(T_\ell')\right]$. In the latter case, i.e., G is connected. We partition set $\{(T_1,\ldots,T_m,T_1',\ldots,T_m') : T_\ell\in\mathcal{T}_i^t, T_\ell'\in\mathcal{T}_j^t\}$ as a union of two disjoint sets $Q\cup R$, where

1. Q consists of $2m$-tuples of trees such that G is connected and there exists an edge (r,s) in $E(G_2)\cup E_J$ which is covered exactly once.
2. R consists of $2m$-tuples of trees such that G is connected and all edges in $E(G_2)\cup E_J$ are covered at least twice.

If $(T_1,\ldots,T_m,T_1',\ldots,T_m')\in Q$, then

$$\mathbb{E}\left[\prod_{\ell=1}^{m}\bar{A}(T_\ell)\bar{A}(T_\ell')\right] = 0 \quad \text{and} \quad \mathbb{E}\left[\prod_{\ell=1}^{m}\bar{A}(T_\ell)\right]\mathbb{E}\left[\prod_{\ell=1}^{m}\bar{A}(T_\ell')\right] = 0.$$

We are left to check R. Following the argument used in Lemma 27, further divide R according to the total number of edges in trees and the number of edges in $E(G_1)$ which is covered exactly once. In particular, define $R_{\alpha,k}$ in the similar manner as $R_{1,\alpha,k}$. Invoking Lemma 24, it can be shown that for any $2m$-tuple in $R_{\alpha,k}$

$$\left|\mathbb{E}\left[\prod_{\ell=1}^{m}\bar{A}(T_\ell)\bar{A}(T_\ell')\right]\right| \le c\mu^k n^{-\alpha/2}$$

$$\left|\mathbb{E}\left[\prod_{\ell=1}^{m}\bar{A}(T_\ell)\right]\mathbb{E}\left[\prod_{\ell=1}^{m}\bar{A}(T_\ell')\right]\right| \le c\mu^k n^{-\alpha/2},$$

Furthermore, let $D = \{i, j\}$ to be the set of distinguished vertices in defining equivalence classes as specificed in Definition 25, so that $R_{\alpha,k}$ is divided into a number of equivalence classes, the number of which depends only on m, t, d, by Lemma 26. For any such equivalence class $S \subset R_{\alpha,k}$, it follows from the last displayed equation that

$$\sum_{T_1,\ldots,T_m,T_1',\ldots,T_m' \in S} \left| \mathbb{E}\left[\prod_{\ell=1}^m \bar{A}(T_\ell)\bar{A}(T_\ell')\right] \right| + \left| \mathbb{E}\left[\prod_{\ell=1}^m \bar{A}(T_\ell)\right] \mathbb{E}\left[\prod_{\ell=1}^m \bar{A}(T_\ell')\right] \right| \leq c\mu^k n^{-\alpha/2}|S|.$$

There are two distinguished vertices, i and j, in the graph G, corresponding to the type of the root vertices of the first m trees and the second m trees, respectively. It follows from Lemma 26 that $|S| \leq K^{n_1} n^{n_2}$, where n_1 is the number of vertices in G_1 with types in $C^* \backslash \{i, j\}$ and n_2 is the number of vertices in G_2 with types in $[n] \backslash (C^* \cup \{i, j\})$. There are $n_1 + n_2 + 2$ vertices in the connected graph G and at most $k + \frac{\alpha-k}{2}$ edges, so $n_1 + n_2 \leq k - 1 + \frac{\alpha-k}{2}$. At most $\frac{\alpha-k}{2}$ edges have at least one endpoint in V_2 and G is connected, so $n_2 \leq \frac{\alpha-k}{2}$. Thus, $|S| \leq K^{n_1} n^{n_2} \leq K^{k-1} n^{\frac{\alpha-k}{2}}$. Hence,

$$\sum_{(T_1,\ldots,T_m,T_1',\ldots,T_m') \in S} \left| \mathbb{E}\left[\prod_{\ell=1}^m \bar{A}(T_\ell)\bar{A}(T_\ell')\right] \right| + \left| \mathbb{E}\left[\prod_{\ell=1}^m \bar{A}(T_\ell)\right] \mathbb{E}\left[\prod_{\ell=1}^m \bar{A}(T_\ell')\right] \right| \leq c\mu^k n^{-\alpha/2} K^{k-1} n^{\frac{\alpha-k}{2}} = c\left(\frac{K\mu}{\sqrt{n}}\right)^k / K \leq c/K.$$

In conclusion, $\mathsf{var}\left(\frac{1}{K}\sum_{i \in C^*} (\xi_i^t)^m\right) \leq c/K$ and the first claim follows. ■

With Lemma 28 and Lemma 29 in hand, we are ready to compute the moments of ξ^t in the asymptotic limit $n \to \infty$.

Lemma 30 *For any $t \geq 0$, $m \geq 1$:*

$$\lim_{n\to\infty} \mathbb{E}\left[(\xi_{i\to j}^t)^m\right] = \mathbb{E}\left[(\mu_t + \tau_t Z_t)^m\right], \quad \forall i \in C^*, j \in [n], j \neq i \tag{67}$$

$$\lim_{n\to\infty} \mathbb{E}\left[(\xi_{i\to j}^t)^m\right] = \mathbb{E}\left[(\tau_t Z_t)^m\right], \quad \forall i \notin C^*, j \in [n], j \neq i. \tag{68}$$

$$\lim_{n\to\infty} \mathbb{E}\left[(\xi_i^t)^m\right] = \mathbb{E}\left[(\mu_t + \tau_t Z_t)^m\right], \quad \forall i \in C^*$$

$$\lim_{n\to\infty} \mathbb{E}\left[(\xi_i^t)^m\right] = \mathbb{E}\left[(\tau_t Z_t)^m\right], \quad \forall i \notin C^*.$$

Proof Below we shall use the following version of the Berry-Esseen central limit theorem. There is an absolute constant C_0 such that if $X_1, X_2, \ldots, X_n$ are independent mean zero random variables and $S_n = X_1 + \cdots + X_n$ then

$$\sup_{x\in\mathbb{R}} \left| \mathbb{P}\left\{ \frac{S_n}{\sqrt{\mathsf{var}(S_n)}} \leq x \right\} - \Phi(x) \right| \overset{(a)}{\leq} \frac{C_0 \sum_{i\in[n]} \mathbb{E}\left[|X_i|^3\right]}{(\mathsf{var}(S_n))^{3/2}} \overset{(b)}{\leq} \frac{C_0 n^{1/4} \left(\sum_{i\in[n]} \mathbb{E}\left[X_i^4\right]\right)^{3/4}}{(\mathsf{var}(S_n))^{3/2}},$$

where Φ is the standard normal CDF; (a) is the original result of Esseen (Esseen, 1942); (b) follows by Jensen's inequality.

We prove the first two claims, (67) and (68). The other two follow similarly. The proof is by induction, so suppose (67) and (68) hold for some t and all $n \geq 1$. We aim to show they also hold for $t+1$. The above identities hold for the base case $t = 0$, because $\xi_{i\to j}^0 = 0$ for all $i \neq j$ and $\mu_0 = \tau_0 = 0$. By the induction hypothesis, Lemma 29, and Chebyshev's inequality,

$$\lim_{n\to\infty} \frac{1}{K} \sum_{\ell \in C^* \backslash \{i\}} (\xi_{\ell\to i}^t)^m \overset{p}{=} \mathbb{E}\left[(\mu_t + \tau_t Z_t)^m\right], \quad \forall i \in [n], \tag{69}$$

$$\lim_{n\to\infty} \frac{1}{n} \sum_{\ell \in [n] \backslash (C^* \cup \{i\})} (\xi_{\ell\to i}^t)^m \overset{p}{=} \mathbb{E}\left[(\tau_t Z_t)^m\right], \quad \forall i \in [n], \tag{70}$$

where $Z_t \sim \mathcal{N}(0, 1)$.

Fix an $i \in C^*$. Then

$$\begin{aligned}
\lim_{n\to\infty} \mathbb{E}\left[\xi_{i\to j}^{t+1} | \mathcal{F}_t\right] &= \lim_{n\to\infty} \mathbb{E}\left[\sum_{\ell \in C^* \backslash \{i,j\}} A_{\ell i}^t f(\xi_{\ell\to i}^t) + \sum_{\ell \in [n] \backslash (C^* \cup \{j\})} A_{\ell i}^t f(\xi_{\ell\to i}^t) | \mathcal{F}_t\right] \\
&= \sqrt{\lambda} \lim_{n\to\infty} \frac{1}{K} \sum_{\ell \in C^* \backslash \{i,j\}} f(\xi_{\ell\to i}^t) \\
&\overset{p}{=} \sqrt{\lambda} \lim_{n\to\infty} \frac{1}{K} \sum_{\ell \in C^* \backslash \{i\}} f(\xi_{\ell\to i}^t) \\
&\overset{p}{=} \sqrt{\lambda} \mathbb{E}\left[f(\mu_t + \tau_t Z_t)\right] = \mu_{t+1},
\end{aligned} \tag{71}$$

where the first equality follows from the definition of ξ^{t+1} given by (61); the second equality holds because $\mathbb{E}\left[A_{\ell i}^t\right] = \mu/\sqrt{n}$ if $\ell \in C^*$ and $\mathbb{E}\left[A_{\ell i}^t\right] = 0$ otherwise; the third equality holds in view of Lemma 28, the fourth equality holds due to (69) and the fact f is a finite-degree polynomial; the last equality holds due to the definition of μ_{t+1}.

Similarly,

$$\begin{aligned}
\lim_{n\to\infty} \mathsf{var}\left(\xi_{i\to j}^{t+1} | \mathcal{F}_t\right) &= \lim_{n\to\infty} \sum_{\ell \in [n] \backslash \{i,j\}} \mathsf{var}\left(A_{\ell i}^t f(\xi_{\ell\to i}^t) | \mathcal{F}_t\right) \\
&= \lim_{n\to\infty} \frac{1}{n} \sum_{\ell \in [n] \backslash \{i,j\}} f(\xi_{\ell\to i}^t)^2 \qquad (72) \\
&= \lim_{n\to\infty} \frac{1}{n} \left\{ \sum_{\ell \in [n] \backslash C^* \cup \{j\}} f(\xi_{\ell\to i}^t)^2 + \sum_{\ell \in C^* \backslash \{i,j\}} f(\xi_{\ell\to i}^t)^2 \right\} \qquad (73) \\
&\overset{p}{=} \lim_{n\to\infty} \frac{1}{n} \sum_{\ell \in [n] \backslash C^*} f(\xi_{\ell\to i}^t)^2 \qquad (74) \\
&\overset{p}{=} \mathbb{E}\left[f(\tau_t Z_t)^2\right] = \tau_{t+1}^2, \qquad (75)
\end{aligned}$$

where the first equality follows from the conditional independence of $A_{\ell i}^t f(\xi_{\ell\to i}^t)$ for $\ell \in [n]$; the second equality holds because $\mathsf{var}(A_{\ell i}) = 1/n$ for all ℓ; the third equality is the result of breaking a sum into two parts, the fourth equality holds in view of Lemma 28 and the

assumption that $K = o(n)$; the fifth equality holds in view of (70) and the fact f is a finite-degree polynomial; the last equality holds due to the definition of τ_{t+1}.

Conditional on $\mathcal{F}_t$, $\xi_{i\to j}^{t+1} - \mathbb{E}\left[\xi_{i\to j}^{t+1}\right]$ is a sum of independent random variables. Therefore, by the form of the Berry-Esseen central limit theorem noted above,

$$\begin{aligned}
&\sup_{x\in\mathbb{R}}\left|\mathbb{P}\left\{\frac{\xi_{i\to j}^{t+1} - \mathbb{E}\left[\xi_{i\to j}^{t+1}|\mathcal{F}_t\right]}{\mathsf{var}(\xi_{i\to j}^{t+1}|\mathcal{F}_t)} \le x\middle|\mathcal{F}_t\right\} - \Phi(x)\right| \\
&\le \frac{C_0 n^{1/4}}{\left(\mathsf{var}(\xi_{i\to j}^{t+1}|\mathcal{F}_t)\right)^{3/2}}\left(\sum_{\ell\in[n]\setminus\{j\}} f(\xi_{\ell\to i}^t)^4\mathbb{E}\left[(A_{\ell i}^t - \mathbb{E}\left[A_{\ell i}^t\right])^4\right]\right)^{3/4} \\
&\le \frac{C_0 c^{3/4} n^{-1/2}}{\left(\mathsf{var}(\xi_{i\to j}^{t+1}|\mathcal{F}_t)\right)^{3/2}}\left(\frac{1}{n}\sum_{\ell\in[n]\setminus\{j\}} f(\xi_{\ell\to i}^t)^4\right)^{3/4}, \qquad (76)
\end{aligned}$$

where we used the fact $\mathbb{E}\left[(A_{\ell i}^t - \mathbb{E}\left[A_{\ell i}^t\right])^4\right] \le cn^{-2}$, for a constant c depending only on the constant γ appearing in the subgaussian assumption. Taking the limit $n\to\infty$ and noticing that $\mathsf{var}(\xi_{i\to j}^{t+1}|\mathcal{F}_t) \xrightarrow{p} \tau_{t+1}^2$ and $\frac{1}{n}\sum_{\ell\in[n]\setminus\{j\}} f(\xi_{\ell\to i}^t)^4 \xrightarrow{p} \mathbb{E}\left[f(\tau_t Z_t)^4\right]$ (using the same steps as in (72)-(75)), we find the righthand side of (76) converges to zero in probability. Thus, in view of (71), (75), and (76), for any $x\in\mathbb{R}$,

$$\lim_{n\to\infty}\mathbb{P}\left\{\xi_{i\to j}^{t+1} \le x|\mathcal{F}_t\right\} \stackrel{p}{=} \mathbb{P}\left\{\mu_{t+1} + \tau_{t+1}Z_{t+1} \le x\right\}.$$

It follows by the dominated convergence theorem that

$$\lim_{n\to\infty}\mathbb{P}\left\{\xi_{i\to j}^{t+1} \le x\right\} = \mathbb{P}\left\{\mu_{t+1} + \tau_{t+1}Z_{t+1} \le x\right\}.$$

and, since convergence in distribution is preserved under continuous transformations:

$$\lim_{n\to\infty}\mathbb{P}\left\{\left(\xi_{i\to j}^{t+1}\right)^m \le x\right\} = \mathbb{P}\left\{(\mu_{t+1} + \tau_{t+1}Z_{t+1})^m \le x\right\}.$$

Since $\mathbb{E}\left[\left|\xi_{i\to j}^{t+1}\right|^{m+1}\right] \le c$ for some c independent of n, the family of random variables $(\xi_{i\to j}^{t+1})^m$ is uniformly integrable, so that convergence in distribution implies convergence of means to the mean of the limiting distribution. Therefore (67) holds with t replaced by $t+1$.

To complete the proof by induction it remains to show (68) holds with t replaced by $t+1$, so fix $i\notin C^*$. Following the previous argument, one can easily check that

$$\begin{aligned}
\mathbb{E}\left[\xi_{i\to j}^{t+1}|\mathcal{F}_t\right] &= 0 \\
\lim_{n\to\infty}\mathsf{var}\left(\xi_{i\to j}^{t+1}|\mathcal{F}_t\right) &\stackrel{p}{=} \tau_{t+1}^2,
\end{aligned}$$

and that, by the central limit theorem, uniformly bounded $m+1^{th}$ moments, and uniform integrability, (68) holds with t replaced by $t+1$. ■

Proof [Proof of Lemma 6] We show the first claim; the second one follows analogously. Fix $t\ge 1$. The convergence property to be proved depends only on the sequence of random empirical distributions of $(\theta_i^t : t\in C^*)$ indexed by n. We may therefore assume without loss of generality that all the random variables $(\theta_i^t : t\in C^*)$ for different n are defined on a single underlying probability space; the joint distribution for different values of n can be arbitrary. To show the convergence in probability, it suffices to show that for any subsequence $\{n_k\}$ there exists a sub-subsequence $\{n_{k_\ell}\}$ such that for $j\neq i$,

$$\lim_{\ell\to\infty} d_{\mathrm{KS}}\left(\frac{1}{K_{k_\ell}}\sum_{i\in C^*}\delta_{\theta_i^t}, \mathcal{N}(\mu_t,\tau_t^2)\right) = 0, \text{ a.s.} \qquad (77)$$

Fix a subsequence n_k. In view of Lemmas 27 and 30, for any fixed integer m,

$$\lim_{k\to\infty}\mathbb{E}\left[(\theta_i^t)^m\right] = \mathbb{E}\left[(\mu_t + \tau_t Z_t)^m\right].$$

Combining Lemma 29 with Chebyshev's inequality,

$$\lim_{k\to\infty}\frac{1}{K_k}\sum_{i\in C^*}\left(\theta_i^t\right)^m \stackrel{p}{=} \mathbb{E}\left[(\mu_t + \tau_t Z_t)^m\right], \qquad (78)$$

which further implies, by a well-known property of convergence in probability, that there exists a sub-subsequence such that (78) holds almost surely. Using a standard diagonal argument, one can construct a sub-subsequence $\{n_{k_\ell}\}$ such that for all $m\ge 1$,

$$\lim_{\ell\to\infty}\frac{1}{K_{k_\ell}}\sum_{i\in C^*}\left(\theta_i^t\right)^m = \mathbb{E}\left[(\mu_t + \tau_t Z_t)^m\right] \text{ a.s.}$$

Since a Gaussian distribution is determined by its moments, by the method of moments (see, for example, (Chung, 2001, Theorem 4.5.5)), applied for each outcome ω in the underlying probability space (excluding some subset of probability zero), it follows that the sequence of empirical distribution of θ_i^t for $i\in C^*$ weakly converges to $\mathcal{N}(\mu_t,\tau_t^2)$, which, since Gaussian density is bounded, is equivalent to convergence in the KS distance,[11] proving the desired (77). ■

Remark 31 *We discuss the difference between the proof of Lemma 6 and that of (Deshpande and Montanari, 2015, Lemma 2.2). First, a larger K requires modification of bounds from (Deshpande and Montanari, 2015) to calculate the moments of messages in Lemmas 27 - 29. In particular, $(\theta_{i\to j}^t)^m$ can be expanded as a sum of monomials in terms of $A_{k\ell}$ for $k,\ell\in[n]$. A larger K implies that in the expansion, there are more monomials containing $A_{k\ell}$ for $k,\ell\in C^*$. That effect is offset by μ being smaller. Our approach is to balances these two effects by accounting separately the contributions of those $A_{k\ell}$'s which appear only once in a monomial. Such $A_{k\ell}$'s correspond to singly covered edges with both endpoints in C^*. See Lemma 24, as well as $R_{1,\alpha,k}$ in Lemma 27, $R_{3,\alpha,k}$ in Lemma 28, and $R_{\alpha,k}$ in Lemma 29. Finally, Lemma 6 is phrased in terms of KS distance while (Deshpande and Montanari, 2015, Lemma 2.2) is in terms of convergence of expected values of bounded Lipschitz functions.*

11. This follows from the fact that when one of the distributions has bounded density the Lévy distance, which metrizes weak convergence, is equivalent to the KS distance (see, e.g. (Petrov, 1995, 1.8.32)).

6.1 Justification of state evolution equations for biclustering

In this subsection, we briefly describe how to generalize the method of moments from symmetric case to asymmetric biclustering case. Recall $f(x,t)=\sum_{r=0}^{d} q_r^t x^r$ with $|q_r^t|\le C$ for some constant C. Let $\{W^t, t\ge 1\}$ be i.i.d. matrices distributed as W conditional on (C_1^*, C_2^*) and let $W^0=W$. Similar to (61), we define a sequence of vectors $\{\xi^t, t\ge 1\}$ given by

$$(t \text{ even}) \qquad \xi_{i\to j}^{t+1} = \frac{1}{\sqrt{n_2}} \sum_{\ell\in[n_2]\backslash\{j\}} W_{i\ell}^t f(\xi_{\ell\to i}^t, t), \quad \forall i\in[n_1], j\in[n_2] \tag{79}$$

$$(t \text{ odd}) \qquad \xi_{j\to i}^{t+1} = \frac{1}{\sqrt{n_1}} \sum_{\ell\in[n_1]\backslash\{i\}} W_{\ell j}^t f(\xi_{\ell\to j}^t, t), \quad \forall j\in[n_2], i\in[n_1], \tag{80}$$

with the initial condition $\xi_{\ell\to i}^0=0$ for $(\ell,i)\in[n_2]\times[n_1]$. Moreover, let

$$(t \text{ even}) \qquad \xi_i^{t+1} = \frac{1}{\sqrt{n_2}} \sum_{\ell\in[n_2]} W_{i\ell}^t f(\xi_{\ell\to i}^t, t), \quad \forall i\in[n_1] \tag{81}$$

$$(t \text{ odd}) \qquad \xi_j^{t+1} = \frac{1}{\sqrt{n_1}} \sum_{\ell\in[n_1]} W_{\ell j}^t f(\xi_{\ell\to j}^t, t), \quad \forall j\in[n_2]. \tag{82}$$

The first step is to represent $(\theta_{i\to j}^t, \theta_i^t)$ and $(\xi_{i\to j}^t, \xi_i^t)$ as sums over a family of finite rooted labeled trees. Abusing notation slightly, we treat elements from $[n_1]$ and $[n_2]$ as distinct elements. We define $\mathcal{T}^t$ as Definition 21 with an additional constraint: a vertex u must have type from $[n_1]$ if $t-|u|$ is odd and from $[n_2]$ if $t-|u|$ is even. In particular, all leaves u with $|u|=t$ must have type from $[n_2]$, and the root has type from $[n_1]$ if t is odd and from $[n_2]$ if t is even.

The following lemma similar to Lemma 22 can be proved by induction on t similar to (Deshpande and Montanari, 2015, Lemma 3.3).

Lemma 32

$$\theta_{i\to j}^t = \sum_{T\in\mathcal{T}_{i\to j}^t} A(T)\Gamma(T,\mathbf{q},t),$$
$$\theta_i^t = \sum_{T\in\mathcal{T}_i^t} A(T)\Gamma(T,\mathbf{q},t),$$

where

$$A(T) \triangleq \prod_{\substack{u\to v\in E(T)\\ \ell(u)\in[n_1]}} \frac{1}{\sqrt{n_1}} W_{\ell(u),\ell(v)} \prod_{\substack{u\to v\in E(T)\\ \ell(u)\in[n_2]}} \frac{1}{\sqrt{n_2}} W_{\ell(v),\ell(u)}$$

and $\Gamma(T,\mathbf{q},t)$ *is defined as before:*

$$\Gamma(T,\mathbf{q},t) = \prod_{u\to v\in E(T)} q_{r(u)}^{t-|u|}.$$

Similarly,

$$\xi_{i\to j}^t = \sum_{T\in\mathcal{T}_{i\to j}^t} \bar{A}(T)\Gamma(T,\mathbf{q},t),$$
$$\xi_i^t = \sum_{T\in\mathcal{T}_i^t} \bar{A}(T)\Gamma(T,\mathbf{q},t),$$

where

$$\bar{A}(T) \triangleq \prod_{\substack{u\to v\in E(T)\\ \ell(u)\in[n_1]}} \frac{1}{\sqrt{n_1}} W_{\ell(u),\ell(v)}^{t-|u|} \prod_{\substack{u\to v\in E(T)\\ \ell(u)\in[n_2]}} \frac{1}{\sqrt{n_2}} W_{\ell(v),\ell(u)}^{t-|u|}.$$

Similar to Definition 23, let G denote the undirected *bipartite* graph obtained by identifying the vertices of the same type in the tuple of trees $T_1,\dots,T_m$ and removing the edge directions. Note that the vertices of type from $[n_1]$ ($[n_2]$) in trees constitute the left (right) part of G. Let $E(G)$ denote the edge set of G. Let G_1 denote the restriction of G to the vertices in (C_1^*, C_2^*) and G_2 the restriction of G to the vertices in $([n_1]\backslash C_1^*, [n_2]\backslash C_2^*)$. Let $E(G_1)$ and $E(G_2)$ denote the edge set of G_1 and G_2, respectively. Let E_J denote the set of edges in G with one endpoint in G_1 and the other endpoint in G_2. Then Lemma 24 goes through as before, with n in the upper bound taken to be $\min\{n_1,n_2\}$. Note that in the definition of $A(T)$, either W_{ij} is divided by $\sqrt{n_1}$ or W_{ji} is divided by $\sqrt{n_2}$ depending on whether $i\in[n_1]$ or $i\in[n_2]$. However, it always holds that $\frac{1}{\sqrt{n_1}}\le\frac{1}{\sqrt{n}}$ and $\frac{1}{\sqrt{n_2}}\le\frac{1}{\sqrt{n}}$.

For a given set of distinguished types D, we can define the equivalence classes on the family of m-tuples of trees in $\mathcal{T}^t$ similar to Definition 25 except that we only allow either permutations of types in $[n_1]$ or permutations of types in $[n_2]$. Then Lemma 26 goes throught as before, with the upper bound in Lemma 26 changes to

$$|S| \le (\max\{K_1,K_2\})^{t_1} (\max\{n_1,n_2\})^{t_2},$$

where we assume a representative m-tuple of trees $(T_1,\dots,T_m)$ has t_1 and t_2 distinct types in $C_1^*\cup C_2^*\backslash D$ and $[n_1]\cup[n_2]\backslash(C_1^*\cup C_2^*\cup D)$, respectively. By assumptions that $\lambda_1,\lambda_2=\Theta(1)$ and $K_1\asymp K_2$, it follows that $n_1\asymp n_2\asymp n$. Furthermore, let $K=\min\{K_1,K_2\}$. Then $K_1\asymp K_2\asymp K$. Hence, $|S|\le cK^{t_1}n^{t_2}$ for an absolute constant $c>0$.

With Lemma 32, modified Lemma 24, and modified Lemma 26, Lemmas 27 - 29 go through as before. In particular, we still partition set $\{(T_1,\dots,T_m): T_\ell\in\mathcal{T}_i^t\}$ as a union of four disjoint sets $Q\cup R_1\cup R_2\cup R_3$, and introduce $R_{1,\alpha,k}$, $R_{2,\alpha,k}$, $R_{3,\alpha,k}$ and $R_{\alpha,k}$ as before.

Acknowledgments

We would like to acknowledge support for this project from the National Science Foundation (NSF grants IIS-9988642, CCF 14-09106, CCF-1527105, CCF-1755960, OIS 18-23145, and a CAREER award CCF-1651588) and the Multidisciplinary Research Program of the Department of Defense (MURI N00014-00-1-0637).

A. Row-wise thresholding

We describe a simple thresholding procedure for recovering C^*. For simplicity and information theoretic comparisons, we consider the Gaussian case. Let $R_i = \sum_j W_{i,j}$ for $i \in [n]$. Then $R_i \sim \mathcal{N}(K\mu, n)$ if $i \in C^*$ and $R_i \sim \mathcal{N}(0, n)$ if $i \notin C^*$. Let $\widehat{C} = \left\{ i \in [n] : R_i \geq \frac{K\mu}{2} \right\}$. Then $\mathbb{E}\left[\left| \widehat{C} \triangle C^* \right| \right] = nQ\left(\frac{K\mu}{2\sqrt{n}} \right)$. Recall that $\lambda = \frac{K^2\mu^2}{n}$. Hence, if

$$\lambda = \omega\left(\log \frac{n}{K} \right), \tag{83}$$

then we have $\mathbb{E}[|\widehat{C} \triangle C^*|] = o(K)$ and hence achieved weak recovery. In the regime $K \asymp n \asymp (n-K)$, $\lambda = \omega(\log \frac{n}{K})$ is equivalent to $\lambda \to \infty$, which is also equivalent to $K\mu^2 \to \infty$ and coincides with the necessary and sufficient condition for the information-theoretic possibility of weak recovery in this regime (Hajek et al., 2017, Corollary 2). (If instead $n - K = o(n)$, weak recovery is trivially provided by $\widehat{C} = [n]$.) Thus, row-wise thresholding provides weak recovery in the regime $K \asymp n \asymp (n-K)$ whenever information theoretically possible. Under the information-theoretic condition (15), an algorithm attaining exact recovery can be built using row-wise thresholding for weak recovery followed by voting, as in Algorithm 2 (see (Hajek et al., 2017, Theorem 3) and its proof). In the regime $\frac{n}{K} \log \frac{n}{K} = o(\log n)$, or equivalently $K = \omega(n \log\log n / \log n)$, condition (15) implies that $\lambda = \omega(\log \frac{n}{K})$, and hence in this regime exact recovery can be attained in linear time $O(n^2)$ whenever information theoretically possible.

B. Proof of Lemma 12

Proof We remind the reader that in this paper we let $A = W/\sqrt{n}$ so that $\mathsf{var}(A_{ij}) = 1/n$ for $i, j \in [n]$ and $\mathbb{E}[A_{ij}] = \mu/\sqrt{n}$ for $i, j \in C^*$.

Fix a $\widetilde{C}$ that satisfies (29) – (30), i.e., $|\widetilde{C} \cap C^*| \geq K(1-\epsilon)$ and $K(1-\epsilon) \leq |\widetilde{C}| \leq n\epsilon$. Let $m = |\widetilde{C}|$ and abbreviate the restricted matrix $A_{\widetilde{C}} \in \mathbb{R}^{|\widetilde{C}| \times |\widetilde{C}|}$ by $\widetilde{A}$. Let $\mathbf{1}_{\widetilde{C} \cap C^*} \in \mathbb{R}^{|\widetilde{C}|}$ denote the indicator vector of $\widetilde{C} \cap C^*$. Then the mean of $\widetilde{A}$ is the rank-one matrix $\mathbb{E}[\widetilde{A}] = \frac{\mu}{\sqrt{n}} \mathbf{1}_{\widetilde{C} \cap C^*} \mathbf{1}_{\widetilde{C} \cap C^*}^\top$, whose largest eigenvalue is $\frac{\mu |\widetilde{C} \cap C^*|}{\sqrt{n}}$ with the corresponding eigenvector $v \triangleq \frac{1}{\sqrt{|\widetilde{C} \cap C^*|}} \mathbf{1}_{\widetilde{C} \cap C^*}$. Let $Z = \widetilde{A} - \mathbb{E}[\widetilde{A}]$, and let u denote the principal eigenvector of $\widetilde{A}$ such that $\langle u, v \rangle \geq 0$. Note that $\|u - v\| = \sqrt{2(1 - \langle u, v \rangle)} \leq \sqrt{2(1 - \langle u, v \rangle^2)} = \sqrt{2} \sin\theta$, where θ is the angle between u and v. Combining this observation with the sin theorem of (Davis and Kahan, 1970) yields

$$\begin{aligned} \|u - v\| \leq \sqrt{2} \sin\theta \leq \sqrt{2} \min\left\{ 1, \frac{\|Z\|}{\mu |\widetilde{C} \cap C| / \sqrt{n} - \|Z\|} \right\} \\ \leq \frac{2\sqrt{2}\|Z\|}{\mu |\widetilde{C} \cap C| / \sqrt{n}} \leq \frac{2\sqrt{2}\|Z\|}{\sqrt{\lambda}(1-\epsilon)}, \end{aligned} \tag{84}$$

where the last inequality follows from the assumption (29). Observe that Z is a symmetric matrix such that $\{Z_{ij}\}_{i \leq j}$ are independent subgaussian random variables with zero mean

and proxy variance γ/n. To bound $\|Z\|$, we use the following standard concentration inequality, see e.g.. (Deshpande and Montanari, 2015, Lemma A.3): For any $t > 0$,

$$\mathbb{P}\left\{ \|Z\| \geq t \right\} \leq 2 \exp\left(-m \left(\frac{t^2 n}{16 e \gamma m} - \log \frac{5 t^2 n}{16 \gamma m} \right) \right).$$

Note that if

$$t \geq \sqrt{64 e \gamma h(\epsilon) + 160 e \gamma m / n},$$

then

$$m \left(\frac{t^2 n}{16 e \gamma m} - \log \frac{5 t^2 n}{16 \gamma m} \right) \geq \frac{t^2 n}{32 e \gamma} \geq 2 n h(\epsilon).$$

By assumption we have $K(1-\epsilon) \leq m \leq \epsilon n$. Therefore, by setting $\beta = \sqrt{64 e \gamma h(\epsilon) + 160 e \gamma \epsilon}$, we have for any fixed $\widetilde{C}$,

$$\mathbb{P}\left\{ \|Z\| \geq \beta \right\} \leq 2 e^{-2nh(\epsilon)}. \tag{85}$$

The number of possible choices of $\widetilde{C}$ that fulfills (30) so that $|\widetilde{C}| \leq \epsilon n$ is at most $\sum_{k \leq n\epsilon} \binom{n}{k}$ which is further upper bounded by $e^{nh(\epsilon)}$ (see, e.g., (Ash, 1965, Lemma 4.7.2)). In view of (85), the union bound yields $\|Z\| \leq \beta$ with high probability as $n \to \infty$.

Throughout the reminder of this proof we assume A and $\widetilde{C}$ are fixed with $\|Z\| \leq \beta$. Combining with (84), we have,

$$\|u - v\| \leq \frac{2\sqrt{2}\beta}{\sqrt{\lambda}(1-\epsilon)}. \tag{86}$$

Next, we argue that either $\widehat{u}$ or $-\widehat{u}$ is close to u, and hence, close to v by the triangle inequality. By the choice of the initial vector u^0, we can write $u^0 = z/\|z\|$ for a standard normal vector $z \in \mathbb{R}^m$. By the tail bounds for Chi-squared distributions, it follows that $\|z\| \leq 2\sqrt{m}$ with high probability. For any fixed u, the random variable $\langle u, z \rangle \sim \mathcal{N}(0, 1)$ and thus with high probability, $|\langle u, z \rangle|^2 \geq 1/\log n$, and hence

$$|\langle u, u^0 \rangle| = |\langle u, z \rangle| / \|z\| \geq (2\sqrt{n \log n})^{-1}. \tag{87}$$

Replacing u^0 by $-u^0$ would result in replacing u^t by $-u^t$ for each t, and since $\widehat{C}$ returned by Algorithm 1 only depends on $|u_i^t|$, replacing u by $-u$ would have no effect on the ouput $\widehat{C}$ of the algorithm. Thus, we can assume without loss of generality that $\langle u, u^0 \rangle \geq 0$, and (87) becomes

$$\langle u, u^0 \rangle \geq (2\sqrt{n \log n})^{-1}. \tag{88}$$

By Weyl's inequality, the maximal singular value of $\widetilde{A}$ satisfies $\sigma_1(\widetilde{A}) \geq \frac{\mu K(1-\epsilon)}{\sqrt{n}} - \beta$ and the other singular values are at most β. Let $r = \frac{\sigma_2}{\sigma_1}(\widetilde{A})$. By the assumption that $\epsilon < \epsilon_0$ and $\lambda \geq 1/e$, we have $\sqrt{\lambda}(1-\epsilon) > 2\beta$. As a consequence, $r \leq \frac{2\beta}{\sqrt{\lambda}(1-\epsilon)} < 1$. Since $u^t = \widetilde{A}^t u^0 / \|\widetilde{A}^t u^0\|$, it follows that

$$u^t = \frac{\langle u, u^0 \rangle u + y}{\|\langle u, u^0 \rangle u + y\|}$$

for some $y \in \mathbb{R}^m$, depending on t, such that $\|y\| \le r^t$. Hence,

$$\begin{aligned}\langle u^t, u\rangle &= \frac{\langle u, u^0\rangle + \langle y, u\rangle}{\|\langle u, u^0\rangle u + y\|} \\ &\ge \frac{\langle u, u^0\rangle - r^t}{\langle u, u^0\rangle + r^t} \\ &= 1 - \frac{2r^t}{\langle u, u^0\rangle + r^t},\end{aligned}$$

or, equivalently,

$$\|u^t - u\|^2 = 2(1 - \langle u^t, u\rangle) \le \frac{4r^t}{\langle u, u^0\rangle + r^t}. \tag{89}$$

Recall that $\widehat{u} = u^{\lceil s^* \log n\rceil}$. Thus, choosing $s^* = \frac{2}{\log(\sqrt{\lambda}(1-\epsilon)/(2\beta))}$ as in (33), we obtain $r^{\lceil s^* \log n\rceil} \le n^{-2}$ and consequently in view of (88) and (89),

$$\|\widehat{u} - u\|^2 \le \frac{4n^{-2}}{(2\sqrt{n \log n})^{-1} + n^{-2}} \le \frac{1}{n},$$

for sufficiently large n.

Therefore, by the triangle inequality,

$$\|\widehat{u} - v\| \le \|\widehat{u} - u\| + \|u - v\| \le n^{-1/2} + \frac{2\sqrt{2}\beta}{\sqrt{\lambda}(1-\epsilon)} \overset{(a)}{\le} \frac{3\beta}{\sqrt{\lambda}(1-\epsilon)} \triangleq \beta_o, \tag{90}$$

where (a) holds for sufficiently large n. Let $\widehat{C}_o$ be defined by using a threshold test to estimate C^* based on $\widehat{u}$:

$$\widehat{C}_o = \{i \in \widetilde{C} : |\widehat{u}_i| \ge \tau\},$$

where $\tau = 1 \Big/ \left(2\sqrt{|\widetilde{C} \cap C^*|}\right)$. Note that $v_i = 2\tau \mathbf{1}_{\{i \in \widetilde{C} \cap C^*\}}$. For any $i \in \widehat{C}_o \backslash (\widetilde{C} \cap C^*)$, we have $|\widehat{u}_i| \ge \tau$ and $v_i = 0$; for any $i \in (\widetilde{C} \cap C^*)\backslash \widehat{C}_o$, we have $|\widehat{u}_i| < \tau$ and $v_i = 2\tau$. Therefore $|\widehat{u}_i - v_i| \ge ||\widehat{u}_i| - |v_i|| \ge \tau$ for all $i \in \widehat{C}_o \triangle (\widetilde{C} \cap C^*)$ and thus

$$\|\widehat{u} - v\|^2 \ge |\widehat{C}_o \triangle (\widetilde{C} \cap C^*)|\tau^2.$$

In view of (90), the number of indices in $\widetilde{C}$ incorrectly classified by $\widehat{C}_o$ satisfies

$$|\widehat{C}_o \triangle (\widetilde{C} \cap C^*)| \le 4\beta_o^2 |\widetilde{C} \cap C^*| \le 4\beta_o^2 |C^*|.$$

Since $|C^* \backslash \widetilde{C}| \le \epsilon K$, we conclude that $|C^* \triangle \widehat{C}_o| \le \epsilon K + 4\beta_o^2 |C^*|$. Thus, if the algorithm were to output $\widehat{C}_o$ (instead of $\widehat{C}$) the lemma would be proved.

Rather than using a threshold test in the cleanup step, Algorithm 1 selects the K indices in $\widetilde{C}$ with the largest values of $|\widehat{u}_i|$. Consequently, with probability one, either $\widehat{C}_o \subset \widehat{C}$ or $\widehat{C} \subset \widehat{C}_o$. Therefore, it follows that

$$|C^* \triangle \widehat{C}| \le 2|C^* \triangle \widehat{C}_o| + \big||C^*| - K\big|.$$

By assumption, $|C^*|/K$ converges to one in probability, so that, in probability,

$$\limsup_{n\to\infty} \frac{|C^* \triangle \widehat{C}|}{K} \le 2\epsilon + 8\beta_o^2 \le \eta(\epsilon, \lambda), \tag{91}$$

where η is defined in (34), completing the proof. ■

C. An adaptive variant of Algorithm 1

Recall that the last step of the spectral clean-up of Algorithm 1 involves choosing the K coordinates of the largest magnitude of $\widehat{u}$. In order to be adaptive to the cluster size K, in this section we show that this step can be simply replaced by applying k-means clustering with $k = 2$ to $\{|\widehat{u}_i|\}_{i \in \widetilde{C}}$ so that Theorem 1 continues to hold. Let w denote an optimal solution, i.e., a minimizer of $\|x - |\widehat{u}|_{\widetilde{C}}\|$ over all x in $\mathbb{R}^{|\widetilde{C}|}$ whose coordinates take at most two distinct values. Since $|\widehat{u}_i|$ is a scalar, w can be easily found by sorting $\{|\widehat{u}_i|\}_{i \in \widetilde{C}}$ in descending order, and checking all vectors of the form $(a, \ldots, a, b, \ldots, b)$, where a and b with $a \ge b \ge 0$ are given by the average of the respective set of $|\widehat{u}_i|$'s. Thus w can be found in time $O(n \log n)$.

Define $\widehat{C} = \{i \in \widetilde{C} : w_i = a\}$. To show this $\widehat{C}$ fulfills the same performance guarantee as in Theorem 1, it suffices to modify the proof of Lemma 12 to show that, for any ϵ sufficiently small, if $\widetilde{C} \subset [n]$ satisfies (29) – (30), then $\mathbb{P}\left\{\frac{|C^* \triangle \widehat{C}|}{K} \le \eta\right\} \to 1$ as $n \to \infty$, where η is a function of ϵ and λ such that $\eta \to 0$ as $\epsilon \to 0$ for λ fixed. Without loss of generality we may also assume that

$$|\widetilde{C} \backslash C^*| = \Omega(K). \tag{92}$$

This extra condition is fulfilled by the output of the message-passing algorithm with high probability, because, in view of (32), $|\widetilde{C} \backslash C^*| = \Theta(n)$ with probability tending to one.

Recall that we have shown in (90) that $\min\{\|\widehat{u} - v\|, \|\widehat{u} + v\|\} \le \beta_o$. By the definition of w, since $v \ge 0$ is binary-valued componentwise, we have

$$\||\widehat{u}| - w\| \le \||\widehat{u}| - v\| \le \min\{\|\widehat{u} - v\|, \|\widehat{u} + v\|\} \le \beta_o,$$

and thus

$$\|w - v\| \le \|w - |\widehat{u}|\| + \||\widehat{u}| - v\| \le 2\beta_o.$$

Define

$$S = \{i \in \widetilde{C} : \big|w_i - v_i\big| \ge \tau\}, \qquad \tau \triangleq \frac{1}{2\sqrt{|\widetilde{C} \cap C^*|}}.$$

Then

$$|S|\tau^2 \le \|w - v\|^2 \le 4\beta_0^2,$$

and consequently, $|S| \le 16\beta_0^2 |\widetilde{C} \cap C^*|$. Since β_0 can be made to be sufficiently small by choosing ϵ to be small, we have $|S| < |\widetilde{C} \cap C^*|$. Furthermore, by the assumption that $|C^*|/K \to 1$ in probability and (92), we have $|S| < |\widetilde{C} \backslash C^*|$. Define $T_1 = (\widetilde{C} \cap C^*)\backslash S$ and $T_0 = (\widetilde{C} \backslash C^*)\backslash S$, both of which are non-empty. For each $i \in T_1$ and $j \in T_0$, we have

$$w_i - w_j \ge v_i - v_j - |w_i - v_i| - |w_j - v_j| > 2\tau - \tau - \tau = 0,$$

that is, $w_i = a > b = w_j$. Hence, $\widehat{C}\Delta(\widetilde{C} \cap C^*) \subset S$ and thus

$$|\widehat{C}\Delta(\widetilde{C} \cap C^*)| \leq |S| \leq 16\beta_0^2 |\widetilde{C} \cap C^*| \leq 16\beta_0^2 |C^*|.$$

Since $|C^* \backslash \widetilde{C}| \leq \epsilon K$, we have that $|\widehat{C}\Delta C^*| \leq \epsilon K + 16\beta_0^2 |C^*|$. Therefore,

$$\limsup_{n\to\infty} \frac{|C^*\Delta\widehat{C}|}{K} \leq \epsilon + 16\beta_0^2.$$

Since $\beta_0 \to 0$ as $\epsilon \to 0$, Theorem 1 holds for the adaptive variant of Algorithm 1.

D. Proofs of Theorems 13 and 14

In the proofs below we use the following notation. We write $\mathsf{p}_e(\pi_1, s^2)$ to denote the minimal average error probability for testing $\mathcal{N}(\mu_1, \sigma^2)$ versus $\mathcal{N}(\mu_0, \sigma^2)$ with priors π_1 and $1 - \pi_1$, where $\mu_1 \geq \mu_0$ and $s^2 = \frac{(\mu_0 - \mu_1)^2}{\sigma^2}$. That is,

$$\mathsf{p}_e(\pi_1, s^2) \triangleq \min_x \{\pi_1 Q(s - x) + (1 - \pi_1) Q(x)\}.$$

Proof [Proof of Theorem 13] The proof of sufficiency for weak recovery is closely based on the proof of sufficiency for exact recovery by the MLE given in (Butucea et al., 2015); the reader is referred to (Butucea et al., 2015) for the notation used in this paragraph. The proof in (Butucea et al., 2015) is divided into two sections. In our terminology, (Butucea et al., 2015, Section 3.1) establishes the weak recovery of C_1^* and C_2^* by the MLE under the assumptions (37), (39), and (41). However, the assumption (39) (and similarly, (41)) is used in only one place in the proof, namely for bounding the terms $T_{1,km}$ defined therein. We explain here why (37) alone is sufficient for the proof of weak recovery. Condition (37), in the notation[12] of (Butucea et al., 2015), implies that there exists some sufficiently small $\alpha > 0$ such that

$$\frac{a^2 m}{2\log(N/n)} \geq 1 + \alpha.$$

So (Butucea et al., 2015, (3.4)) can be replaced as: there exist some sufficiently small $\delta_1 > 0$ and $\alpha_1 > 0$ such that

$$\frac{(1-\delta_1)^2}{2} a^2 m \geq (1 + \alpha_1)\log(N/n) \geq (1 + \alpha_1)\log\left(\frac{\delta(N-n)}{n-k}\right),$$

where we use the assumption $0 \leq k < (1 - \delta)n$, or $n - k > \delta n$. Thus, for large enough n,

$$\begin{aligned} T_{1,km} &\leq \exp\left(-\frac{\delta n \alpha_1}{2}\left(\log\left(\frac{N-n}{n-k}\right)\right)\right) \\ &\leq \exp\left(-\frac{\delta n \alpha_1}{2}\log\left(\frac{N-n}{n}\right)\right) = o(1/n), \end{aligned}$$

12. The notation of (Butucea et al., 2015) is mapped to ours as $N \to n_1$, $M \to n_2$, $n \to K_1$, $m \to K_2$, and $a \to \mu$.

from which the desired conclusion, $\sum_{k:(n-k)>\delta n} T_{1,km} = o(1)$, follows. This completes the proof of sufficiency of (37) for weak recovery of both C_1^* and C_2^*, and marks the end of our use of notation from (Butucea et al., 2015).

The rate distortion argument used in the proof of (Hajek et al., 2017, Theorem 1) shows that (38) must hold if C_1^* and C_2^* are both weakly recoverable. ■

Proof [Proof of Theorem 14] We give the proof for exact recovery of C_1^*; the proof for exact recovery of C_2^* is analogous. For the sufficiency part, Recall that in Algorithm 3, the set $[n_1]$ is partitioned into sets, $S_1, \ldots, S_{1/\delta}$ of size $n_1\delta$. There are $1/\delta$ rounds of the algorithm, and indices in S_k are classified in the k^{th} round. For the k^{th} round, by assumption, given $\epsilon > 0$, there exists an estimator $\widehat{C}_{2k}$ based on observation of W with the rows indexed by S_k hidden such that $|\widehat{C}_{2k}\Delta C_2^*| \leq \epsilon K_2$ with high probability. Then the voting procedure estimates whether $i \in C_1^*$ for each $i \in S_k$ by comparing $\sum_{j\in\widehat{C}_{2k}} W_{i,j}$ to a threshold. This sum has approximately the $\mathcal{N}(K_2\mu, K_2)$ distribution if $i \in C_1^*$ and $\mathcal{N}(0, K_2)$ distribution otherwise ; the discrepancy can be made sufficiently small by choosing ϵ to be small (See (Hajek et al., 2017, Theorem 3) for a proof). Thus, the mean number of classification errors is well approximated by $n_1\mathsf{p}_e(K_1/n_1, K_2\mu^2)$, which converges to zero under (39), completing the sufficiency proof for exact recovery of C_1^*. The necessity part is proved in (Butucea et al., 2015, Section 4.2). ■

E. Proof of Lemma 18

Proof (Similar to proof of Lemma 12.) We prove the lemma for $\widehat{C}_1$; the proof for $\widehat{C}_2$ is identical. For the first part of the proof we assume that for $i = 1, 2$, $\widetilde{C}_i$ is fixed, and later use a union bound over all possible choices of $\widetilde{C}_i$. Recall that $W_{\widetilde{C}_1\widetilde{C}_2}$, which we abbreviate henceforth as $\widetilde{W}$, is the matrix W restricted to entries in $\widetilde{C}_1 \times \widetilde{C}_2$. Let $Z = \widetilde{W} - \mathbb{E}[\widetilde{W}]$ and

$$\mathbb{E}[\widetilde{W}] = \mu\sqrt{|\widetilde{C}_1 \cap C_1^*||\widetilde{C}_2 \cap C_2^*|}\, v_1 v_2^\top \tag{93}$$

is a rank-one matrix, where v_i is the unit vector in $\mathbb{R}^{|\widetilde{C}_i|}$ obtained by normalizing the indicator vector of $\widetilde{C}_i \cap C_i^*$. Thus, thanks to (54), the leading singular value of $\mathbb{E}[\widetilde{W}]$ is at least $\mu\sqrt{K_1K_2}(1-\epsilon)$ with left singular vector v_1 and right singular vector v_2.

It is well-known (see, e.g., (Vershynin, 2010, Corollary 5.35)) that if M is an $m_1 \times m_2$ matrix with i.i.d. *standard normal* entries, then $\mathbb{P}\left\{\|M\| \geq \sqrt{m_1} + \sqrt{m_2} + t\right\} \leq 2e^{-t^2/2}$. Applying this result for $m_i = |\widetilde{C}_i|$, which satisfies $m_i \leq \epsilon n_i$ by (55), and $t = 2\sqrt{h(\epsilon)(n_1+n_2)}$, we have for fixed $(\widetilde{C}_1, \widetilde{C}_2)$,

$$\mathbb{P}\left\{\|Z\| \geq (\sqrt{n_1} + \sqrt{n_2})\beta\right\} \leq 2e^{-2(n_1+n_2)h(\epsilon)},$$

where $\beta \triangleq 3\sqrt{\epsilon + h(\epsilon)}$). Similar to the proof of Lemma 12, the number of $(\widetilde{C}_1, \widetilde{C}_2)$ that satisfies (55) is at most $e^{(n_1+n_2)h(\epsilon)}$. By union bound, if we drop the assumption that $\widetilde{C}_i$ is fixed for $i = 1, 2$, we still have that with high probability, $\|Z\| \leq (\sqrt{n_1} + \sqrt{n_2})\beta$.

Denote by u the leading left singular vector of $W_{\widetilde{C}_1\widetilde{C}_2}$ such that $\langle u, v_1\rangle \geq 0$. Then, letting θ denote the angle between u and v_1,

$$\|u - v_1\| \leq \sqrt{2}\sin(\theta) \overset{(a)}{\leq} \sqrt{2}\min\left\{\frac{\|Z\|}{\left(\sigma_1\left(\widetilde{W}\right) - \sigma_2\left(\mathbb{E}\left[\widetilde{W}\right]\right)\right)_+}, 1\right\} \overset{(b)}{\leq} \frac{2\sqrt{2}\|Z\|}{\sigma_1(\mathbb{E}[\widetilde{W}])},$$

where (a) follows from Wedin's sin-θ theorem for SVD (Wedin, 1972), and (b) follows from $\sigma_2\left(\mathbb{E}\left[\widetilde{W}\right]\right) = 0$ and Weyl's inequality $\sigma_1(\widetilde{W}) \geq \sigma_1(\mathbb{E}[\widetilde{W}]) - \|Z\|$. In view of (93), conditioning on the high-probability event that $\|Z\| \leq (\sqrt{n_1} + \sqrt{n_2})\beta$, we have

$$\|u - v_1\| \leq \frac{2\sqrt{2}\beta(\sqrt{n_1} + \sqrt{n_2})}{\mu(1-\epsilon)\sqrt{K_1K_2}} \leq \frac{2\sqrt{2}c_0\beta}{1-\epsilon}, \tag{94}$$

where the last inequality follows from the standing assumption (53).

Next, we argue that $\widehat{u}$ or $-\widehat{u}$ is close to u, and hence, close to v_1 by the triangle inequality. By (88), the initial value $u^0 \in \mathbb{R}^{|\widetilde{C}_1|}$ satisfies $|\langle u, u^0\rangle| \geq (2\sqrt{n_1 \log n_1})^{-1}$ with high probability, and without loss of generality we can assume as in the proof of Lemma 12 that $\langle u, u^0\rangle \geq (2\sqrt{n_1 \log n_1})^{-1}$. By Weyl's inequality, the largest singular value of $\widetilde{W}$ is at least $\mu\sqrt{K_1K_2}(1-\epsilon) - (\sqrt{n_1}+\sqrt{n_2})\beta$, and the other singular values are at most $(\sqrt{n_1}+\sqrt{n_2})\beta$. In view of (53), $\frac{1-\epsilon}{c_0\beta} - 1 > 1$ for all $\epsilon < \epsilon_0$, where $\epsilon_0 > 0$ depends only on c_0. Let λ_1 and λ_2 denote the first and second eigenvalue of $\widetilde{W}\widetilde{W}^\top$ in absolute value, respectively. Let $r = \lambda_2/\lambda_1$. Then $r \leq (\frac{c_0\beta}{1-\epsilon-c_0\beta})^2$. Since for even t, $u^t = (\widetilde{W}\widetilde{W}^\top)^{t/2}u^0/\|(\widetilde{W}\widetilde{W}^\top)^{t/2}u^0\|$, the same analysis of power iteration that leads to (89) yields

$$\|u^t - u\|^2 = 2(1 - \langle u^t, u\rangle) \leq \frac{4r^{t/2}}{\langle u, u^0\rangle + r^{t/2}}.$$

Since $\widehat{u} = u^{2\lceil s^* \log n\rceil}$ and $s^* = (\log\frac{1-\epsilon-c_0\beta}{c_0\beta})^{-1}$, we have $r^{\lceil s^* \log n_1\rceil} \leq n_1^{-2}$ and thus $|\langle\widehat{u}, u\rangle^2| \geq 1 - n_1^{-1}$ and consequently, $\|uu^\top - \widehat{u}(\widehat{u})^\top\|_{\mathrm{F}}^2 = 2 - 2\langle u, \widehat{u}\rangle^2 \leq n_1^{-1}$. Similar to (90), applying (94) and the triangle inequality, we obtain

$$\|\widehat{u} - v\| \leq \|\widehat{u} - u\| + \|u - v\| \leq n^{-1/2} + \frac{2\sqrt{2}c_o\beta}{\sqrt{\lambda}(1-\epsilon)} < \frac{3c_o\beta}{\sqrt{\lambda}(1-\epsilon)} \triangleq \beta_o, \tag{95}$$

By the same argument that proves (91), we have $\limsup_{n\to\infty} |C_1^* \triangle \widehat{C}_1|/K_1 \leq 2\epsilon + 8\beta_0^2 \leq \eta(\epsilon)$ with η defined in (57), completing the proof. ■

References

E. Abbe and C. Sandon. Community detection in general stochastic block models: fundamental limits and efficient recovery algorithms. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 670–688, 2015. arXiv 1503.00609.

E. Abbe, A. S. Bandeira, and G. Hall. Exact recovery in the stochastic block model. *IEEE Transactions on Information Theory*, 62(1):471–487, 2016.

N. Alon, M. Krivelevich, and B. Sudakov. Finding a large hidden clique in a random graph. *Random Structures and Algorithms*, 13(3-4):457–466, 1998.

A. Amini, A. Chen, P. J. Bickel, and E. Levina. Pseudo-likelihood methods for community detection in large sparse networks. *The Annals of Statistics*, 41(4):2097–2122, 2013.

R. B. Ash. *Information Theory*. Dover Publications Inc., New York, NY, 1965.

M. Bayati and A. Montanari. The dynamics of message passing on dense graphs, with applications to compressed sensing. *IEEE Transactions on Information Theory*, 57(2): 764–785, 2011.

C. Butucea and Y. I. Ingster. Detection of a sparse submatrix of a high-dimensional noisy matrix. *Bernoulli*, 19(5B):2652–2688, 11 2013. doi: 10.3150/12-BEJ470.

C. Butucea, Y. Ingster, and I. Suslina. Sharp variable selection of a sparse submatrix in a high-dimensional noisy matrix. *ESAIM: Probability and Statistics*, 19:115–134, June 2015.

T. T. Cai, T. Liang, A. Rakhlin, et al. Computational and statistical boundaries for submatrix localization in a large noisy matrix. *The Annals of Statistics*, 45(4):1403–1430, 2017.

Y. Chen and J. Xu. Statistical-computational tradeoffs in planted problems and submatrix localization with a growing number of clusters and submatrices. In *Proceedings of ICML 2014 (Also arXiv:1402.1267)*, Feb 2014.

K. Chung. *A course in probability theory*. Academic press, 2nd edition, 2001.

A. Condon and R. M. Karp. Algorithms for graph partitioning on the planted partition model. *Random Struct. Algorithms*, 18(2):116–140, Mar. 2001.

C. Davis and W. Kahan. The rotation of eigenvectors by a perturbation. III. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

Y. Deshpande and A. Montanari. Finding hidden cliques of size $\sqrt{N/e}$ in nearly linear time. *Foundations of Computational Mathematics*, 15(4):1069–1128, August 2015.

C.-G. Esseen. "on the liapunoff limit of error in the theory of probability". *Arkiv för matematik, astronomi och fysik*, A28:1–19, 1942.

D. Féral and S. Péché. The largest eigenvalue of rank one deformation of large Wigner matrices. *Communications in mathematical physics*, 272(1):185–228, 2007.

B. Hajek, Y. Wu, and J. Xu. Semidefinite programs for exact recovery of a hidden community. In *Proceedings of Conference on Learning Theory (COLT)*, pages 1051–1095, New York, NY, Jun 2016. arXiv:1602.06410.

B. Hajek, Y. Wu, and J. Xu. Information limits for recovering a hidden community. *IEEE Trans. on Information Theory*, 63(8):4729 – 4745, 2017.

J. A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67(337):123–129, 1972.

W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

A. Javanmard and A. Montanari. State evolution for general approximate message passing algorithms, with applications to spatial coupling. *Information and Inference: A Journal of the IMA*, 2(2):115–144, 2013.

A. Knowles and J. Yin. The isotropic semicircle law and deformation of Wigner matrices. *Communications on Pure and Applied Mathematics*, 66(11):1663–1749, 2013.

M. Kolar, S. Balakrishnan, A. Rinaldo, and A. Singh. Minimax localization of structural information in large noisy matrices. In *Advances in Neural Information Processing Systems*, 2011.

Z. Ma and Y. Wu. Computational barriers in minimax submatrix detection. *The Annals of Statistics*, 43(3):1089–1116, 2015.

A. Montanari, D. Reichman, and O. Zeitouni. On the limitation of spectral methods: From the Gaussian hidden clique problem to rank one perturbations of Gaussian tensors. In *Advances in Neural Information Processing Systems*, pages 217–225, 2015. arXiv 1411.6149.

E. Mossel, J. Neeman, and S. Sly. Belief propagation, robust reconstruction, and optimal recovery of block models (extended abstract). In *JMLR Workshop and Conference Proceedings (COLT proceedings)*, volume 35, pages 1–35, 2014.

E. Mossel, J. Neeman, and A. Sly. Consistency thresholds for the planted bisection model. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, STOC '15, pages 69–75, New York, NY, USA, 2015. ACM.

V. V. Petrov. *Limit theorems of probability theory: Sequences of independent random variables*. Oxford Science Publications, Clarendon Press, Oxford, United Kingdom, 1995.

A. A. Shabalin, V. J. Weigman, C. M. Perou, and A. B. Nobel. Finding large average submatrices in high dimensional data. *The Annals of Applied Statistics*, 3(3):985–1012, 2009.

G. Szegö. *Orthogonal polynomials*. American Mathematical Society, Providence, RI, 4th edition, 1975.

R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. *Arxiv preprint arxiv:1011.3027*, 2010.

P. Wedin. Perturbation bounds in connection with singular value decomposition. *BIT Numerical Mathematics*, 12(1):99–111, 1972.

S. Yun and A. Proutiere. Optimal cluster recovery in the labeled stochastic block model. arXiv 1510.05956, Oct. 2015.

Journal of Machine Learning Research 18 (2018) 1-30 Submitted 9/16; Revised 4/17; Published 4/18

Quantized Neural Networks: Training Neural Networks with Low Precision Weights and Activations

Itay Hubara* ITAYHUBARA@GMAIL.COM
Department of Electrical Engineering
Technion - Israel Institute of Technology
Haifa, Israel

Matthieu Courbariaux* MATTHIEU.COURBARIAUX@GMAIL.COM
Department of Computer Science and Department of Statistics
Université de Montréal
Montréal, Canada

Daniel Soudry DANIEL.SOUDRY@GMAIL.COM
Department of Statistics
Columbia University
New York, USA

Ran El-Yaniv RANI@CS.TECHNION.AC.IL
Department of Computer Science
Technion - Israel Institute of Technology
Haifa, Israel

Yoshua Bengio YOSHUA.UMONTREAL@GMAIL.COM
Department of Computer Science and Department of Statistics
Université de Montréal
Montréal, Canada

**Indicates first authors.*

Editor: Nando de Freitas

Abstract

We introduce a method to train Quantized Neural Networks (QNNs) — neural networks with extremely low precision (e.g., 1-bit) weights and activations, at run-time. At train-time the quantized weights and activations are used for computing the parameter gradients. During the forward pass, QNNs drastically reduce memory size and accesses, and replace most arithmetic operations with bit-wise operations. As a result, power consumption is expected to be drastically reduced. We trained QNNs over the MNIST, CIFAR-10, SVHN and ImageNet datasets. The resulting QNNs achieve prediction accuracy comparable to their 32-bit counterparts. For example, our quantized version of AlexNet with 1-bit weights and 2-bit activations achieves 51% top-1 accuracy. Moreover, we quantize the parameter gradients to 6-bits as well which enables gradients computation using only bit-wise operation. Quantized recurrent neural networks were tested over the Penn Treebank dataset, and achieved comparable accuracy as their 32-bit counterparts using only 4-bits. Last but not least, we programmed a binary matrix multiplication GPU kernel with which it is possible to run our MNIST QNN 7 times faster than with an unoptimized GPU kernel, without suffering any loss in classification accuracy. The QNN code is available online.

Keywords: deep learning, neural networks compression, energy efficient neural networks, computer vision, language models

1. Introduction

Deep Neural Networks (DNNs) have substantially pushed Artificial Intelligence (AI) limits in a wide range of tasks, including but not limited to object recognition from images (Krizhevsky et al., 2012; Szegedy et al., 2015), speech recognition (Hinton et al., 2012; Sainath et al., 2013), statistical machine translation (Devlin et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2015), Atari and Go games (Mnih et al., 2015; Silver et al., 2016), and even computer generation of abstract art (Mordvintsev et al., 2015).

Training or even just using neural network (NN) algorithms on conventional general-purpose digital hardware (Von Neumann architecture) has been found highly inefficient due to the massive amount of multiply-accumulate operations (MACs) required to compute the weighted sums of the neurons' inputs. Today, DNNs are almost exclusively trained on one or many very fast and power-hungry Graphic Processing Units (GPUs) (Coates et al., 2013). As a result, it is often a challenge to run DNNs on target low-power devices, and substantial research efforts are invested in speeding up DNNs at run-time on both general-purpose (Vanhoucke et al., 2011; Gong et al., 2014; Romero et al., 2014; Han et al., 2015) and specialized computer hardware (Farabet et al., 2011a,b; Pham et al., 2012; Chen et al., 2014a,b; Esser et al., 2015).

The most common approach is to compress a trained (full precision) network. HashedNets (Chen et al., 2015) reduce model sizes by using a hash function to randomly group connection weights and force them to share a single parameter value. Gong et al. (2014) compressed deep convnets using vector quantization, which resulteds in only a 1% accuracy loss. However, both methods focused only on the fully connected layers. A recent work by Han et al. (2016) successfully pruned several state-of-the-art large scale networks and showed that the number of parameters could be reduced by an order of magnitude.

Recent works have shown that more computationally efficient DNNs can be constructed by quantizing some of the parameters during the training phase. In most cases, DNNs are trained by minimizing some error function using Back-Propagation (BP) or related gradient descent methods. However, such an approach cannot be directly applied if the weights are restricted to binary values. Soudry et al. (2014) used a variational Bayesian approach with Mean-Field and Central Limit approximation to calculate the posterior distribution of the weights (the probability of each weight to be +1 or -1). During the inference stage (test phase), their method samples from this distribution one binary network and used it to predict the targets of the test set (More than one binary network can also be used). Courbariaux et al. (2015) similarly used two sets of weights, real-valued and binary. They, however, updated the real valued version of the weights by using gradients computed by applying forward and backward propagation with the set of binary weights (which was obtained by quantizing the real-value weights to +1 and -1).

This study proposes a more advanced technique, referred to as Quantized Neural Network (QNN), for quantizing the neurons and weights during inference and training. In such networks, all MAC operations can be replaced with $XNOR$ and *population count* (i.e., counting the number of ones in the binary number) operations. This is especially useful in

QNNs with the extremely low precision — for example, when only 1-bit is used per weight and activation, leading to a Binarized Neural Network (BNN). The proposed method is particularly beneficial for implementing large convolutional networks whose neuron-to-weight ratio is very large.

This paper makes the following contributions:

- We introduce a method to train Quantized-Neural-Networks (QNNs), neural networks with low precision weights and activations, at run-time, and when computing the parameter gradients at train-time. In the extreme case QNNs use only 1-bit per weight and activation (*i.e.*, Binarized NN; see Section 2).
- We conduct two sets of experiments, each implemented on a different framework, namely Torch7 and Theano, which show that it is possible to train BNNs on MNIST, CIFAR-10 and SVHN and achieve near state-of-the-art results (see Section 4). Moreover, we report results on the challenging ImageNet dataset using binary weights/activations as well as quantized version of it (more than 1-bit).
- We present preliminary results on quantized gradients and show that it is possible to use only 6-bits with only small accuracy degradation.
- We present results for the Penn Treebank dataset using language models (vanilla RNNs and LSTMs) and show that with 4-bit weights and activations Recurrent QNNs achieve similar accuracies as their 32-bit floating point counterparts.
- We show that during the forward pass (both at run-time and train-time), QNNs drastically reduce memory consumption (size and number of accesses), and replace most arithmetic operations with bit-wise operations. A substantial increase in power efficiency is expected as a result (see Section 5). Moreover, a binarized CNN can lead to binary convolution kernel repetitions; we argue that dedicated hardware could reduce the time complexity by 60% .
- Last but not least, we programmed a binary matrix multiplication GPU kernel with which it is possible to run our MNIST BNN 7 times faster than with an unoptimized GPU kernel, without suffering any loss in classification accuracy (see Section 6).
- The code for training and applying our BNNs is available on-line (both the Theano [1] and the Torch framework [2]).

2. Binarized Neural Networks

In this section, we detail our binarization function, show how we use it to compute the parameter gradients, and how we backpropagate through it.

[1] https://github.com/MatthieuCourbariaux/BinaryNet
[2] https://github.com/itayhubara/BinaryNet

2.1 Deterministic vs Stochastic Binarization

When training a BNN, we constrain both the weights and the activations to either +1 or −1. Those two values are very advantageous from a hardware perspective, as we explain in Section 6. In order to transform the real-valued variables into those two values, we use two different binarization functions, as proposed by Courbariaux et al. (2015). The first binarization function is deterministic:

$$x^b = \text{sign}(x) = \begin{cases} +1 & \text{if } x \geq 0, \\ f-1 & \text{otherwise,} \end{cases} \tag{1}$$

where x^b is the binarized variable (weight or activation) and x the real-valued variable. It is very straightforward to implement and works quite well in practice. The second binarization function is stochastic:

$$x^b = \text{sign}(x-z) = \begin{cases} +1 & \text{with probability } p = \sigma(x), \\ -1 & \text{with probability } 1-p, \end{cases} \tag{2}$$

where $z \sim U[-1,1]$, a uniform random variable, and σ is the *"hard sigmoid"* function:

$$\sigma(x) = \text{clip}(\frac{x+1}{2}, 0, 1) = \max(0, \min(1, \frac{x+1}{2})). \tag{3}$$

This stochastic binarization is more appealing theoretically (see Section 4) than the sign function, but somewhat harder to implement as it requires the hardware to generate random bits when quantizing (Torii et al., 2016). As a result, we mostly use the deterministic binarization function (i.e., the sign function), with the exception of *activations at train-time* in some of our experiments.

2.2 Gradient Computation and Accumulation

Although our BNN training method utilizes binary weights and activations to compute the parameter gradients, the real-valued gradients of the weights are accumulated in real-valued variables, as per Algorithm 1. Real-valued weights are likely required for Stochasic Gradient Descent (SGD) to work at all. SGD explores the space of parameters in small and noisy steps, and that noise is *averaged out* by the stochastic gradient contributions accumulated in each weight. Therefore, it is important to maintain sufficient resolution for these accumulators, which at first glance suggests that high precision is absolutely required.

Moreover, adding noise to weights and activations when *computing* the parameter gradients provide a form of regularization that can help to generalize better, as previously shown with variational weight noise (Graves, 2011), Dropout (Srivastava et al., 2014) and DropConnect (Wan et al., 2013). Our method of training BNNs can be seen as a variant of Dropout, in which instead of randomly setting half of the activations to zero when computing the parameter gradients, we binarize both the activations and the weights.

2.3 Propagating Gradients Through Discretization

The derivative of the sign function is zero almost everywhere, making it apparently incompatible with back-propagation, since the exact gradients of the cost with respect to the

quantities before the discretization (pre-activations or weights) are zero. Note that this limitation remains even if stochastic quantization is used. Bengio (2013) studied the question of estimating or propagating gradients through stochastic discrete neurons. He found that the fastest training was obtained when using the "straight-through estimator," previously introduced in Hinton's lectures (Hinton, 2012). We follow a similar approach but use the version of the straight-through estimator that takes into account the saturation effect, and does use deterministic rather than stochastic sampling of the bit. Consider the sign function quantization over the activations values from a previous layer, r,

$$q = \text{sign}(r),$$

and assume that an estimator g_q of the gradient $\frac{\partial C}{\partial q}$ has been obtained (with the straight-through estimator when needed, where C is the cost function). Then, our straight-through estimator of $\frac{\partial C}{\partial r}$ is simply

$$g_r = g_q 1_{|r|\leq 1}. \tag{4}$$

Note that this preserves the gradient information and cancels the gradient when r is too large. Not cancelling the gradient when r is too large significantly worsens performance.

To better understand why the straight-through estimator works well, we consider the stochastic binarization scheme similar to that in Eq. (2),

$$q = \text{sign}\left(r - z\right),$$

where we recall that $z \sim U\left[-1, 1\right]$. During training, we update each weight $W_{ij,l}$ (connecting neuron j in layer $l-1$ to neuron i in layer l), using the gradient from the previous layer (here $\frac{\partial C}{\partial r_{i,l}}$) and the layer input:

$$\frac{\partial}{\partial W_{ij,l}}\mathbb{E}\left[C\right] = \mathbb{E}\left[\frac{\partial C}{\partial r_{i,l}} q_{j,l-1}\right] = \mathbb{E}_{/z_{i,l}}\left[\mathbb{E}_{z_{i,l}}\left[\frac{\partial C}{\partial r_{i,l}}\right] q_{j,l-1}\right],$$

where $\mathbb{E}$, $\mathbb{E}_{z_{i,l}}$, and $\mathbb{E}_{/z_{i,l}}$ are, respectively, expectations over everything (all the noise sources, (z), and data samples); only $z_{i,l}$ (the noise in neuron i at layer l); or everything except $z_{i,l}$.

Next, we calculate the expectation over $z_{i,l}$ (dropping all indices, for brevity):

$$\begin{aligned}
&\mathbb{E}_z\left[\frac{\partial}{\partial r} C\right] = \frac{\partial}{\partial r}\mathbb{E}_z\left[C\right] \\
&= \frac{\partial}{\partial r}\left[C\left(q=1\right)p\left(r > z|r\right) + C\left(q=-1\right)\left(1 - p\left(r > z|r\right)\right)\right] \\
&= \frac{\partial p\left(r > z|r\right)}{\partial r}\left[C\left(q=1\right) - C\left(q=-1\right)\right] \\
&= 2\left.\frac{\partial C}{\partial q}\right|_{q=0} \mathbf{1}_{|r|\leq 1} + O\left(\left.\frac{\partial^3 C}{\partial q^3}\right|_{q=0}\right),
\end{aligned}$$

where in the last line we used a Taylor expansion. The $O(\cdot)$ contribution to the last equation is usually negligible, since the output of a single neuron ($q = \pm 1$) typically has only a small effect on the cost, and therefore

$$\left.\frac{\partial C}{\partial q}\right|_{q=0} \gg \left.\frac{\partial^2 C}{\partial q^2}\right|_{q=0} \gg \left.\frac{\partial^3 C}{\partial q^3}\right|_{q=0}.$$

Thus, using a Taylor expansion we can approximate

$$\left.\frac{\partial C}{\partial q}\right|_{q=\pm 1} = \left.\frac{\partial C}{\partial q}\right|_{q=0} + O\left(\left.\frac{\partial^2 C}{\partial q^2}\right|_{q=0}\right)$$

Therefore, since $g_r = \partial C/\partial r$ and $g_q = \partial C/\partial q$, we obtain the straight-through estimator defined in Eq. (4), up to a scale constant 2. The use of this straight-through estimator is illustrated in Algorithm 1.

A similar binarization process was applied for weights in which we combine two ingredients:

- Project each real-valued weight to [-1,1], i.e., clip the weights during training, as per Algorithm 1. The real-valued weights would otherwise grow very large without any impact on the binary weights.
- When using a real-valued weight w^r, quantize it using $w^b = \text{Sign}(w^r)$.

Projecting the weights to [-1,1] is consistent with the gradient cancelling when $|w^r| > 1$, according to Eq. (4).

2.4 Shift-based Batch Normalization

Batch Normalization (BN) (Ioffe and Szegedy, 2015) accelerates the training and reduces the overall impact of the weight scale (Courbariaux et al., 2015). The normalization procedure may also help to regularize the model. However, at train-time, BN requires many multiplications (calculating the standard deviation and dividing by it, namely, dividing by the running variance, which is the weighted mean of the training set activation variance). Although the number of scaling calculations is the same as the number of neurons, in the case of ConvNets this number is quite large. For example, in the CIFAR-10 dataset (using our architecture), the first convolution layer, consisting of only $128 \times 3 \times 3 \times 3$ filter masks, converts an image of size $3 \times 32 \times 32$ to size $128 \times 28 \times 28$, which is more than an order of magnitude larger than the number of weights (29 to be exact). To achieve the results that BN would obtain, we use a shift-based batch normalization (SBN) technique, presented in Algorithm 2. SBN approximates BN almost without multiplications. Define AP2(z) as the approximate power-of-2 of z, LogAP2(z) as log(AP2(z)) (i.e., the index of the most significant bit), and $\ll\gg$ as both left and right binary shift. SBN replaces almost all multiplication with power-of-2-approximation and shift operations:

$$x \times y \rightarrow x \ll\gg \text{LogAP2}(y). \tag{5}$$

The only operation which is not a binary shift or an add is the inverse square root (see normalization operation Algorithm 2). From the early work of Lomont (2003) we know that the inverse-square operation could be applied with approximately the same complexity as multiplication. There are also faster methods, which involve lookup table tricks that typically obtain lower accuracy (this may not be an issue, since our procedure already adds a lot of noise). In our experiments we did not use those methods since it requires a dedicated hardware. However, the number of values on which we apply the inverse-square operation

Algorithm 1: Training a BNN. C is the cost function for minibatch, λ, the learning rate decay factor, and L, the number of layers. ($\circ$) stands for element-wise multiplication. The function Binarize($\cdot$) specifies how to (stochastically or deterministically) binarize the activations and weights, and Clip(), how to clip the weights. BatchNorm() specifies how to batch-normalize the activations, using either batch normalization (Ioffe and Szegedy, 2015) or its shift-based variant we describe in Algorithm 2. BackBatchNorm() specifies how to backpropagate through the normalization. Update() specifies how to update the parameters when their gradients are known, using either ADAM (Kingma and Ba, 2015) or the shift-based AdaMax we describe in Algorithm 3.

Require: a minibatch of inputs and targets (a_0, a^*), previous weights W, previous BatchNorm parameters θ, weight initialization coefficients from (Glorot and Bengio, 2010) γ, and previous learning rate η.

Ensure: updated weights W^{next}, updated BatchNorm parameters θ^{next} and updated learning rate η^{next}.

{1. Computing the parameter gradients:}

{1.1. Forward propagation:}

for $k = 1$ to L **do**

$W_k^b \leftarrow \text{Binarize}(W_k)$

$s_k \leftarrow a_{k-1}^b W_k^b$ {$a_0^b = a_0$, i.e., the input is not quantize, see Section 2.6}

$a_k \leftarrow \text{BatchNorm}(s_k, \theta_k)$

if $k < L$ **then**

$a_k^b \leftarrow \text{Binarize}(a_k)$

end if

end for

{1.2. Backward propagation:}

{Note that the gradients are not binary.}

Compute $g_{a_L} = \frac{\partial C}{\partial a_L}$ knowing a_L and a^*

for $k = L$ to 1 **do**

if $k < L$ **then**

$g_{a_k} \leftarrow g_{a_k^b} \circ 1_{|a_k| \leq 1}$

end if

$(g_{s_k}, g_{\theta_k}) \leftarrow \text{BackBatchNorm}(g_{a_k}, s_k, \theta_k)$

$g_{a_{k-1}^b} \leftarrow g_{s_k} W_k^b$

$g_{W_k^b} \leftarrow g_{s_k}^\top a_{k-1}^b$

end for

{2. Accumulating the parameter gradients:}

for $k = 1$ to L **do**

$\theta_k^{\text{next}} \leftarrow \text{Update}(\theta_k, \eta, g_{\theta_k})$

$W_k^{\text{next}} \leftarrow \text{Clip}(\text{Update}(W_k, \gamma_k \eta, g_{W_k^b}), -1, 1)$

$\eta^{\text{next}} \leftarrow \lambda \eta$

end for

Algorithm 2: Shift-based Batch Normalizing Transform, applied to activation x over a mini-batch. $\text{LogAP2}(x) = \text{sign}(x) \times 2^{\text{round}(\log 2|x|)}$ is the approximate power-of-2[(a)], and $\ll\gg$ stands for left or right shifting of x according to the index of the MSB of y.

Require: Values of x over a mini-batch: $B = \{x_{1...m}\}$; Parameters to be learned: γ, β

Ensure: $\{y_i = \text{BN}(x_i, \gamma, \beta)\}$

$\mu_B \leftarrow \frac{1}{m}\sum_{i=1}^m x_i$ {mini-batch mean}

$C(x_i) \leftarrow (x_i - \mu_B)$ {centered input}

$\sigma_B^2 \leftarrow \frac{1}{m}\sum_{i=1}^m (C(x_i) \ll\gg \text{LogAP2}(C(x_i)))$ {approximate variance}

$\hat{x_i} \leftarrow C(x_i) \ll\gg \text{LogAP2}((\sqrt{\sigma_B^2 + \epsilon})^{-1})$ {normalize}

$y_i \leftarrow \text{LogAP2}(\gamma) \ll\gg \hat{x_i}$ {scale and shift}

(a) Hardware implementation of LogAP2 is as simple as extracting the index of the most significant bit from the number's binary representation

Algorithm 3: Shift-based AdaMax learning rule (Kingma and Ba, 2015). g_t^2 indicates the element-wise square $g_t \circ g_t$. Good default settings are $\alpha = 2^{-10}$, $1 - \beta_1 = 2^{-3}$, $1 - \beta_2 = 2^{-10}$. All operations on vectors are element-wise. With β_1^t and β_2^t we denote β_1 and β_2 to the power t.

Require: Previous parameters θ_{t-1}, their gradient g_t, and learning rate α.

Ensure: Updated parameters θ_t

{Biased 1st and 2nd raw moment estimates:}

$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$

$v_t \leftarrow \max(\beta_2 \cdot v_{t-1}, |g_t|)$

{Updated parameters:}

$\theta_t \leftarrow \theta_{t-1} - (\alpha \ll\gg (1 - \beta_1)) \cdot \hat{m} \ll\gg \text{LogAP2}(v_t^{-1}))$

is rather small, since it is done after calculating the variance, i.e., after averaging (for a more precise calculation, see the BN analysis in Lin et al. (2015)). Furthermore, the size of the standard deviation vectors is relatively small. For example, these values make up only 0.3% of the network size (i.e., the number of learnable parameters) in the Cifar-10 network we used in our experiments.

In our experiment we observed no loss in accuracy when using the shift-based BN algorithm instead of the vanilla BN algorithm.

2.5 Shift-Based AdaMax

The ADAM learning method (Kingma and Ba, 2015) also reduces the impact of the weight scale. Since ADAM requires many multiplications, we suggest using instead the shift-based AdaMax we outlined in Algorithm 3. In the experiment we conducted we observed no loss in accuracy when using the shift-based AdaMax algorithm instead of the vanilla ADAM algorithm.

Algorithm 4: Running a BNN with L layers using only the XnorDotProduct kernel (see Section 5).

Require: 8-bit input vector a_0, binary weights W^b, and BatchNorm parameters θ.
Ensure: the MLP output a_L.
{1. First layer:}
$a_1 \leftarrow 0$
for $n = 1$ to 8 **do**
  $a_1 \leftarrow a_1 + 2^{n-1} \times \text{XnorDotProduct}(\text{a}_0^{\text{n}}, \text{W}_1^{\text{b}})$
end for
$a_1^b \leftarrow \text{Sign}(\text{BatchNorm}(a_1, \theta_1))$
{2. Remaining hidden layers:}
for $k = 2$ to $L - 1$ **do**
  $a_k \leftarrow \text{XnorDotProduct}(a_{k-1}^b, W_k^b)$
  $a_k^b \leftarrow \text{Sign}(\text{BatchNorm}(a_k, \theta_k))$
end for
{3. Output layer:}
$a_L \leftarrow \text{XnorDotProduct}(a_{L-1}^b, W_L^b)$
$a_L \leftarrow \text{BatchNorm}(a_L, \theta_L)$

2.6 First Layer

In a BNN, only the binarized values of the weights and activations are used in all calculations. As the output of one layer is the input of the next, the inputs of all the layers are binary, with the exception of the first layer. However, we do not believe this to be a major issue. First, in computer vision, the input representation typically has far fewer channels (e.g, red, green and blue) than internal representations (e.g., 512). Consequently, the first layer of a ConvNet is often the smallest convolution layer, both in terms of parameters and computations (Szegedy et al., 2015). Second, it is relatively easy to handle continuous-valued inputs as fixed point numbers, with m bits of precision. For example, in the common case of 8-bit fixed point inputs:

$$s = x \cdot w^b, \qquad s = \sum_{n=1}^{8} 2^{n-1}(x^n \cdot w^b), \tag{6}$$

where x is a vector of 8-bit inputs, x_1^8 is the most significant bit of the first input, w^b is a vector of 1-bit weights, and s is the resulting weighted sum. This method is used in Algorithm 4.

3. Quantized Neural network - More Than 1-bit

Observing Eq. (6), we can see that using 2-bit activations simply doubles the number of times we need to run our XnorDotProduct Kernel (i.e., directly proportional to the activation bitwidth). This idea was recently proposed by Zhou et al. (2016) (DoReFa net) and Miyashita et al. (2016) (published on arXive shortly after our preliminary technical report was published there). However, in contrast to to Zhou et al., we did not find it useful to initialize the network with weights obtained by training the network with full precision weights. Moreover, the Zhou et al. network did not quantize the weights of the first convolutional layer and the last fully-connected layer, whereas we binarized both. We followed the quantization schemes suggested by Miyashita et al. (2016), namely, linear quantization:

$$\textbf{LinearQuant}(x, bitwidth) = \text{Clip}\left(\text{round}\left(\frac{x}{2^{\text{bitwidth}}-1}\right) \times 2^{\text{bitwidth}-1}, \text{minV}, \text{maxV}\right) \tag{7}$$

and logarithmic quantization:

$$\textbf{LogQuant}(x, bitwidth)\,(\mathbf{x}) = \text{Clip}\left(\text{LogAP2}(x), -(2^{\text{bitwidth}-1}-1), 2^{\text{bitwidth}-1}\right), \tag{8}$$

where minV and maxV are the minimum and maximum scale range respectively and $\text{LogAP2}(x)$ is the log of the approximate-power-of-2 of x as described in Section 2.4. In our experiments (detailed in Section 4) we applied the above quantization schemes on the weights, activations and gradients and tested them on the more challenging ImageNet dataset.

4. Benchmark Results

We performed two sets of experiments, each based on a different framework, namely Torch7 and Theano. Other than the framework, the two sets of experiments are very similar:

- In both sets of experiments, we obtain near state-of-the-art results with BNNs on MNIST, CIFAR-10 and the SVHN benchmark datasets.
- In our Torch7 experiments, the activations are *stochastically* binarized at train-time, whereas in our Theano experiments they are *deterministically* binarized.
- In our Torch7 experiments, we use the *shift-based BN and AdaMax* variants, which are detailed in Algorithms 2 and 3, whereas in our Theano experiments, we use *vanilla BN and ADAM*.

Results are reported in Table 4.1. Implementation details are reported in Appendix A.

4.1 Results on MNIST,SVHN, and CIFAR-10

In this subsection we will detail the setting used in our MNIST,SVHN and Cifar-10 experiments.

4.1.1 MNIST

MNIST is an image classification benchmark dataset (LeCun et al., 1998). It consists of a training set of 60K and a test set of 10K 28×28 gray-scale images representing digits ranging from 0 to 9. The Multi-Layer-Perceptron (MLP) we train on MNIST consists of 3 hidden layers. In our Theano implementation we used hidden layers of size 4096 whereas

Data set	MNIST	SVHN	CIFAR-10
Binarized activations+weights, during training and test			
BNN (Torch7)	1.40%	2.53%	10.15%
BNN (Theano)	0.96%	2.80%	11.40%
Committee Machines' Array Baldassi et al. (2015)	1.35%	-	-
Binarized weights, during training and test			
BinaryConnect Courbariaux et al. (2015)	1.29± 0.08%	2.30%	9.90%
Binarized activations+weights, during test			
EBP Cheng et al. (2015)	2.2± 0.1%	-	-
Bitwise DNNs Kim and Smaragdis (2016)	1.33%	-	-
Ternary weights, binary activations, during test			
Hwang and Sung (2014)	1.45%	-	-
No binarization (standard results)			
No reg	1.3± 0.2%	2.44%	10.94%
Maxout Networks Goodfellow et al. (2013b)	0.94%	2.47%	11.68%
Gated pooling Lee et al. (2016)	-	1.69%	7.62%

Table 1: Classification test error rates of DNNs trained on MNIST (fully connected architecture), CIFAR-10 and SVHN (convnet). No unsupervised pre-training or data augmentation was used.

in our Torch implementation we used much smaller size 2048. This difference explains the accuracy gap between the two implementations. Both Implementations use the squared

hinge loss. In our theano implementation we tried to understand how low can the validation error decrease if we simply inflate the network. As can be seen from the results adding more units improves accuracy. The use of dropout in this case appears to be important.

4.1.2 CIFAR-10

CIFAR-10 is an image classification benchmark dataset. It consists of a training set of size 50K and a test set of size 10K, where instances are 32×32 color images representing airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships and trucks. Both implementations share the same structure as reported in Appendix A. Since the Torch implementation uses stochastic binarization, it achieved slightly better results. As expected the Torch version has better validation performance than the Theano implementation but worse training performance. We believe this is due to the shift-based batch normalization (SBN) and the stochastic units. Both are additional "noise" sources, which make the model harder to train, but can improve generalization.

4.1.3 SVHN

Street View House Numbers (SVHN) is also an image classification benchmark dataset. It consists of a training set of size 604K examples and a test set of size 26K, where instances

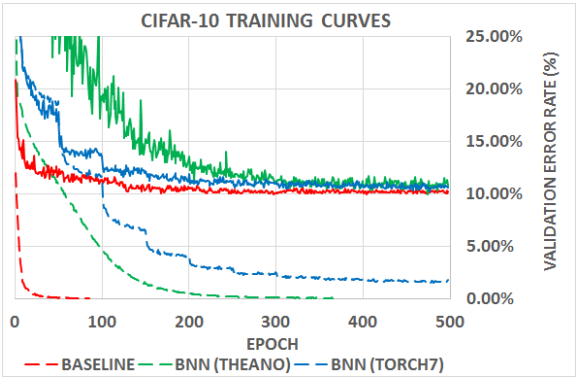


Figure 1: Training curves for different methods on the CIFAR-10 dataset. The dotted lines represent the training costs (square hinge losses) and the continuous lines the corresponding validation error rates. Although BNNs are slower to train, they are nearly as accurate as 32-bit float DNNs.

are 32×32 color images representing digits ranging from 0 to 9. Here again we obtained a small improvement in the performance by using stochastic binarization scheme.

In all our experiments with shift-base AdaMax and BN we did not observe any performance hit. The results tend to stay very close with and without the Shift-base approximation.

4.2 Results on ImageNet

To test the strength of our method, we applied it to the challenging ImageNet classification task, which is probably the most important classification benchmark dataset. It consists of a training set of size 1.2M samples and test set of size 50K. Each instance is labeled with one of 1000 categories including objects, animals, scenes, and even some abstract shapes. On ImageNet, it is customary to report two error rates: top-1 and top-5, where the top-x error rate is the fraction of test images for which the correct label is not among the x labels considered most probable by the model. Considerable research has been concerned with compressing ImageNet architectures while preserving high accuracy. Previous approaches include pruning near zero weights (Gong et al., 2014; Han et al., 2016) using matrix factorization techniques (Zhang et al., 2015), quantizing the weights (Gupta et al., 2015), using shared weights (Chen et al., 2015) and applying Huffman codes (Han et al., 2016) among others.

To the best of our knowledge, before the first revision of this paper was published on arXive, no one had reported on successfully quantizing the network's activations. On the contrary, a recent work (Han et al., 2016) showed that accuracy significantly deteriorates when trying to quantize convolutional layers' weights below 4-bit (FC layers are more ro-

bust to quantization and can operate quite well with only 2 bits). In the present work we attempted to tackle the difficult task of binarizing both weights and activations. Employing the well-known AlexNet and GoogleNet architectures, we applied our techniques and achieved 41.8% top-1 and 67.1% top-5 accuracy using AlexNet and 47.1% top-1 and 69.1% top-5 accuracy using GoogleNet. While these performance results leave room for improvement (relative to full precision nets), they are by far better than all previous attempts to compress ImageNet architectures using less than 4-bit precision for the weights. Moreover, this advantage is achieved while also binarizing neuron activations.

4.3 Relaxing "Hard-Tanh" Boundaries

We discovered that after training the network it is useful to widen the "hard tanh" boundaries and retrain the network. As explained in Section 2.3, the straight-through estimator (which can be written as "hard tanh") cancels gradients coming from neurons with absolute values higher than 1. Hence, towards the last training iterations most of the gradient values are zero and the weight values cease to update. Similarly we can decrease the standard deviation of the output of the batch-normalization layer (i.e., dividing the input by scale $\times \sigma_B$ where scale > 1). By relaxing the "hard tanh" boundaries we allow more gradients to flow in the back-propagation phase and improve top-1 accuracies by 1.5% on AlexNet topology using vanilla BNN.

4.4 2-bit Activations

While training BNNs on the ImageNet dataset we noticed that we could not force the training set error rate to converge to zero. In fact the training error rate stayed fairly close to the validation error rate. This observation led us to investigate a more relaxed activation quantization (more than 1-bit). As can be seen in Table 4.4, the results are quite impressive and illustrate an approximate 5.6% drop in performance (top-1 accuracy) relative to floating point representation, using only 1-bit weights and 2-bit activation. Following Miyashita et al. (2016), we also tried quantizing the gradients and discovered that only logarithmic quantization works. With 6-bit gradients we achieved 5.2% degradation. Those results are presently state-of-the-art, surpassing those obtained by the DoReFa net (Zhou et al., 2016). As opposed to DoReFa, we utilized a deterministic quantization process rather than a stochastic one. We did so because stochastic binarization is harder to optimize as it requires additional memory and computational resources. Moreover, it is important to note that while quantizing the gradients, DoReFa assigns for each instance in a mini-batch its own scaling factor, which increases the number of MAC operations.

While AlexNet can be compressed rather easily, compressing GoogleNet is much harder due to its small number of parameters. When using vanilla BNNs, we observed a large degradation in the top-1 results. However, by using QNNs with 4-bit weights and activation, we were able to achieve 66.5% top-1 accuracy (only a 5.1% drop in performance compared to the 32-bit floating point architecture), which is the current state-of-the-art-compression result over GoogleNet. Moreover, by using QNNs with 6-bit weights, activations and gradients we achieved 66.4% top-1 accuracy. Full implementation details of our experiments are reported in Appendix A.6.

Model	Top-1	Top-5
Binarized activations+weights, during training and test		
BNN	41.8%	67.1%
Xnor-Nets[3] (Rastegari et al., 2016)	44.2%	69.2%
Binary weights and Quantize activations during training and test		
QNN 2-bit activation	51.03%	73.67%
DoReFaNet 2-bit activation[3] (Zhou et al., 2016)	50.7%	72.57%
Quantize weights, during test		
Deep Compression 4/2-bit (conv/FC layer) (Han et al., 2016)	55.34%	77.67%
(Gysel et al., 2016) - 2-bit	0.01%	-%
No Quantization (standard results)		
AlexNet - our implementation	56.6%	80.2%

Table 2: Classification test error rates of the AlexNet model trained on the ImageNet 1000 classification task. No unsupervised pre-training or data augmentation was used.

Model	Top-1	Top-5
Binarized activations+weights, during training and test		
BNN	47.1%	69.1%
Quantize weights and activations during training and test		
QNN 4-bit	66.5%	83.4%
Quantize activation,weights and gradients during training and test		
QNN 6-bit	66.4%	83.1%
No Quantization (standard results)		
GoogleNet - our implementation	71.6%	91.2%

Table 3: Classification test error rates of the GoogleNet model trained on the ImageNet 1000 classification task. No unsupervised pre-training or data augmentation was used.

4.5 Language Models

Recurrent neural networks (RNNs) are very demanding in memory and computational power in comparison to feed forward networks. There are a large variety of recurrent models with the Long Short Term Memory networks (LSTMs) introduced by Hochreiter and Schmidhuber (1997) are being the most popular model. LSTMs are a special kind of RNN, capable of learning long-term dependencies using unique gating mechanisms. Recently, Ott et al. (2016) tried to quantize the RNNs weight matrices using similar techniques as described in Section 2. They observed that the weight binarization methods do not work with RNNs. However, by using 2-bits (i.e., $-1, 0, 1$), they have been able to achieve similar and even higher accuracy on several datasets. Here we report on the first attempt to quantize both weights and activations by trying to evaluate the accuracy of quantized recurrent models

trained on the Penn Treebank dataset. The Penn Treebank Corpus (Marcus et al., 1993) contains 10K unique words. We followed the same setting as in (Mikolov and Zweig, 2012) which resulted in 18.55K words for training set, 14.5K and 16K words in the validation and test sets respectively. We experimented with both vanilla RNNs and LSTMs. For our vanilla RNN model we used one hidden layers of size 2048 and ReLU as the activation function. For our LSTM model we use 1 hidden layer of size 300. Our RNN implementation was constructed to predict the next character hence performance was measured using the bits-per-character (BPC) metric. In the LSTM model we tried to predict the next word so performance was measured using the perplexity per word (PPW) metric. Similar to (Ott et al., 2016), our preliminary results indicate that binarization of weight matrices lead to large accuracy degradation. However, as can be seen in Table 4.5, with 4-bits activations and weights we can achieve similar accuracies as their 32-bit floating point counterparts.

Model	Layers	Hidden Units	bits(weights)	bits(activation)	Accuracy
RNN	1	2048	3	3	1.81 BPC
RNN	1	2048	2	4	1.67 BPC
RNN	1	2048	3	4	1.11 BPC
RNN	1	2048	3	4	1.05 BPC
RNN	1	2048	FP	FP	1.05 BPC
LSTM	1	300	2	3	220 PPW
LSTM	1	300	3	4	110 PPW
LSTM	1	300	4	4	100 PPW
LSTM	1	900	4	4	97 PPW
LSTM	1	300	FP	FP	97 PPW

Table 4: Language Models results on Penn Treebank dataset. FP stands for 32-bit floating point

5. High Power Efficiency During The Forward Pass

Computer hardware, be it general-purpose or specialized, is composed of memories, arithmetic operators and control logic. During the forward pass (both at run-time and train-time), BNNs drastically reduce memory size and accesses, and replace most arithmetic operations with bit-wise operations, which might lead to vastly improved power-efficiency. Moreover, a binarized CNN can lead to binary convolution kernel repetitions, and we argue that dedicated hardware could reduce the time complexity by 60% .

5.0.1 Memory Size and Accesses

Improving computing performance has always been and remains a challenge. Over the last decade, power has been the main constraint on performance (Horowitz, 2014). This is why considerable research efforts have been devoted to reducing the energy consumption of

[3] First and last layers were not binarized (i.e., using 32-bit precision weights and activation.)

Operation	MUL	ADD
8-bit Integer	0.2pJ	0.03pJ
32-bit Integer	3.1pJ	0.1pJ
16-bit Floating Point	1.1pJ	0.4pJ
32-bit Floating Point	3.7pJ	0.9pJ

Table 5: Energy consumption of multiply- accumulations; see Horowitz (2014)

Memory size	64-bit Cache
8K	10pJ
32K	20pJ
1M	100pJ
DRAM	1.3-2.6nJ

Table 6: Energy consumption of memory accesses; see Horowitz (2014)

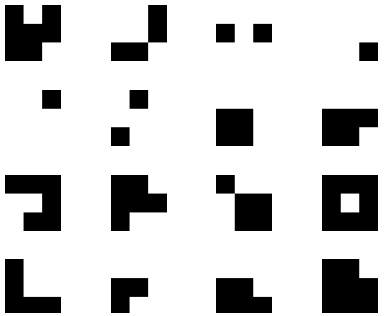


Figure 2: Binary weight filters, sampled from of the first convolution layer. Since we have only 2^{k^2} unique 2D filters (where k is the filter size), filter replication is very common. For instance, on our CIFAR-10 ConvNet, only 42% of the filters are unique.

neural networks. Horowitz (2014) provides rough numbers for the energy consumed by the computation

(the given numbers are for 45nm technology), as summarized in Tables 5 and 6. Importantly, we can see that memory accesses typically consume more energy than arithmetic operations, and *memory access cost increases with memory size*. In comparison with 32-bit DNNs, BNNs require 32 times smaller memory size *and* 32 times fewer memory accesses. This is expected to reduce energy consumption drastically (i.e., by a factor larger than 32).

5.0.2 XNOR-Count

Applying a DNN mainly involves convolutions and matrix multiplications. The key arithmetic operation of deep learning is thus the multiply-accumulate operation. Artificial neurons are basically multiply-accumulators computing weighted sums of their inputs. In BNNs, both the activations and the weights are constrained to either -1 or $+1$. As a result, most of the 32-bit floating point multiply-accumulations are replaced by 1-bit XNOR-count operations. We named this operation the XnorDotProduct kernel. This could have a big impact on dedicated deep learning hardware. For instance, a 32-bit floating point multiplier costs about 200 Xilinx FPGA slices (Govindu et al., 2004; Beauchamp et al., 2006), whereas a 1-bit XNOR gate only costs a single slice.

When using a ConvNet architecture with binary weights, the number of unique filters is bounded by the filter size. For example, in our implementation we use filters of size 3×3, so the maximum number of unique 2D filters is $2^9 = 512$. However, this should not prevent expanding the number of feature maps beyond this number, since the actual filter is a 3D matrix. Assuming we have M_ℓ filters in the ℓ convolutional layer, we have to store a 4D weight matrix of size $M_\ell \times M_{\ell-1} \times k \times k$. Consequently, the number of unique filters is $2^{k^2 M_{\ell-1}}$. When necessary, we apply each filter on the map and perform the required multiply-accumulate (MAC) operations (in our case, using XNOR and popcount operations). Since we now have binary filters, many 2D filters of size $k \times k$ repeat themselves. By using dedicated hardware/software, we can apply only the unique 2D filters on each feature map and sum the results to receive each 3D filter's convolutional result. Note that an inverse filter (i.e., [-1,1,-1] is the inverse of [1,-1,1]) can also be treated as a repetition; it is merely a multiplication of the original filter by -1. For example, in our ConvNet architecture trained on the CIFAR-10 benchmark, there are only 42% unique filters per layer on average. Hence we can reduce the number of the XNOR-popcount operations by 3.

QNNs complexity scale up linearly with the number of bits per weight/activation, since it requires the application of the XNOR kernel several times (see Section 3). As of now, QNNs still supply the best compression to accuracy ratio. Moreover, quantizing the gradients allows us to use the XNOR kernel for the backward pass, leading to fully fixed point layers with low bitwidth. By accelerating the training phase, QNNs can play an important role in future power demanding tasks.

6. Seven Times Faster on GPU at Run-Time

It is possible to speed up GPU implementations of QNNs, by using a method sometimes called SIMD (single instruction, multiple data) within a register (SWAR). The basic idea of SWAR is to *concatenate* groups of 32 binary variables into 32-bit registers, and thus obtain a 32-times speed-up on bitwise operations (e.g., XNOR). Using SWAR, it is possible to evaluate 32 connections with only 3 instructions:

$$a_1 += \text{popcount}(\text{xnor}(a_0^{32b}, w_1^{32b})), \tag{9}$$

where a_1 is the resulting weighted sum, and a_0^{32b} and w_1^{32b} are the concatenated inputs and weights. Those 3 instructions (accumulation, popcount, xnor) take $1+4+1 = 6$ clock cycles on recent Nvidia GPUs (and if they were to become a fused instruction, it would only take

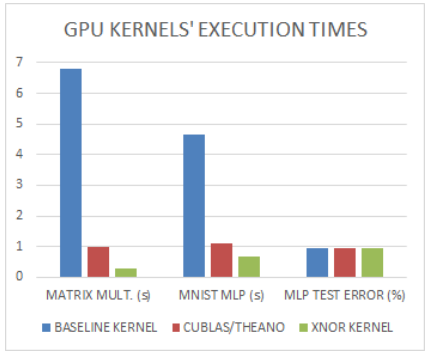
a single clock cycle). Consequently, we obtain a theoretical Nvidia GPU speed-up of factor of $32/6 \approx 5.3$ (a GPU core can preform 32 fused multiply and add (FMA) operations in 32 cycles). In practice, this speed-up is quite easy to obtain as the memory bandwidth to computation ratio is also increased 6 times.

In order to validate those theoretical results, we programmed two GPU kernels [4]:

- An unoptimized matrix multiplication kernel that serves as our baseline.
- The XNOR kernel, which is nearly identical to the baseline, except that it uses the SWAR method, as in Equation (9).

The two GPU kernels return identical outputs when their inputs are constrained to -1 or $+1$ (but not otherwise). The XNOR kernel is about *23 times faster than the baseline kernel* and *3.4 times faster than cuBLAS*, as shown in Figure 3. Our rather unoptimized binary matrix multiplication kernel can benefit from using better optimization techniques such as blocking, hence the XNOR kernel can be even faster. Last but not least, the MLP from Section 4 runs 7 times faster with the XNOR kernel than with the baseline kernel, without suffering any loss in classification accuracy (see Figure 3). As MNIST's images are not binary, the first layer's computations are always performed by the baseline kernel. The last three columns show that the MLP accuracy does not depend on which kernel is used.

Figure 3: The first 3 columns show the time it takes to perform a $8192 \times 8192 \times 8192$ (binary) matrix multiplication on a GTX750 Nvidia GPU, depending on which kernel is used. The next three columns show the time it takes to run the MLP from Section 3 on the full MNIST test set. The last three columns show that the MLP accuracy does not depend on the kernel



[4] Both kernels are block matrix multiplication based on the CUDA C programming guide: `http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html#shared-memory`

7. Discussion and Related Work

Until recently, the use of extremely low-precision networks (binary in the extreme case) was believed to substantially degrade the network performance (Courbariaux et al., 2014). Soudry et al. (2014) and Cheng et al. (2015) proved the contrary by showing that good performance could be achieved even if all neurons and weights are binarized to ± 1 . This was done using Expectation BackPropagation (EBP), a variational Bayesian approach, which infers networks with binary weights and neurons by updating the posterior distributions over the weights. These distributions are updated by differentiating their parameters (e.g., mean values) via the back propagation (BP) algorithm. Esser et al. (2015) implemented a fully binary network at run time using a very similar approach to EBP, showing significant improvement in energy efficiency. The drawback of EBP is that the binarized parameters are only used during inference.

The probabilistic idea behind EBP was extended in the BinaryConnect algorithm of Courbariaux et al. (2015). In BinaryConnect, the real-valued version of the weights is saved and used as a key reference for the binarization process. The binarization noise is independent between different weights, either by construction (by using stochastic quantization) or by assumption (a common simplification; see Spang and Schultheiss, 1962). The noise would have little effect on the next neuron's input because the input is a summation over many weighted neurons. Thus, the real-valued version could be updated using the back propagated error by simply ignoring the binarization noise in the update. With this method, Courbariaux et al. (2015) were the first to binarize weights in CNNs and achieved near state-of-the-art performance on several datasets. They also argued that noisy weights provide a form of regularization, which could help to improve generalization, as previously shown by Wan et al. (2013). This method binarized weights while still maintaining full precision neurons.

Lin et al. (2015) carried over the work of Courbariaux et al. (2015) to the back-propagation process by quantizing the representations at each layer of the network, to convert some of the remaining multiplications into binary shifts by restricting the neurons' values to be power-of-two integers. Lin et al. (2015)'s work and ours seem to share similar characteristics .However, their approach continues to use full precision weights during the test phase. Moreover, Lin et al. (2015) quantize the neurons only during the back propagation process, and not during forward propagation.

Other research (Baldassi et al., 2015) showed that full binary training and testing is possible in an array of committee machines with randomized input, where only one weight layer is being adjusted. Gong et al. (2014) aimed to compress a fully trained high precision network by using quantization or matrix factorization methods. These methods required training the network with full precision weights and neurons, thus requiring numerous MAC operations (which the proposed QNN algorithm avoids). Hwang and Sung (2014) focused on a fixed-point neural network design and achieved performance almost identical to that of the floating-point architecture. Kim and Smaragdis (2016) *retrained* neural networks with binary weights and activations.

As far as we know, before the first revision of this paper was published on arXive, no work succeeded in binarizing weights *and* neurons, at the inference phase *and* the entire training phase of a deep network. This was achieved in the present work. We relied on the

idea that binarization can be done stochastically, or be approximated as random noise. This was previously done for the weights by Courbariaux et al. (2015), but our BNNs extend this to the activations. Note that the binary activations are especially important for ConvNets, where there are typically many more neurons than free weights. This allows highly efficient operation of the binarized DNN at run time, and at the forward-propagation phase during training. Moreover, our training method has almost no multiplications, and therefore might be implemented efficiently in dedicated hardware. However, we have to save the value of the full precision weights. This is a remaining computational bottleneck during training, since it is an energy-consuming operation.

Shortly after the first version of this paper was posted on arXiv, several papers tried to improve and extend it. Rastegari et al. (2016) made a small modification to our algorithm (namely multiplying the binary weights and input by their L_1 norm) and published promising results on the ImageNet dataset. Note that their method, named Xnor-Net, requires additional multiplication by a different scaling factor for each patch in each sample (Rastegari et al., 2016) Section 3.2 Eq. 10 and figure 2). This in itself, requires many multiplications and prevents efficient implementation of XnorNet on known hardware designs. Moreover, (Rastegari et al., 2016) didn't quantize first and last layers, therefore XNOR-Net are only partially binarized NNs. Miyashita et al. (2016) suggested a more relaxed quantization (more than 1-bit) for both the weights and activation. Their idea was to quantize both and use shift operations as in our Eq. (4). They proposed to quantize the parameters in their non-uniform, base-2 logarithmic representation. This idea was inspired by the fact that the weights and activations in a trained network naturally have non-uniform distributions. They moreover showed that they can quantize the gradients as well to 6-bit without significant losses in performance (on the Cifar-10 dataset). Zhou et al. (2016) applied similar ideas to the ImageNet dataset and showed that by using 1-bit weights, 2-bit activations and 6-bit gradients they can achieve 46.1% top-1 accuracies using the AlexNet architecture. They named this method DoReFa net. Here we outperform DoReFa net and achieve 46.8% using a 1-2-6 bit quantization scheme (weight-activation-gradients) and 51% using a 1-2-32 quantization scheme. These results confirm that we can achieve comparable results even on a large dataset by applying the Xnor kernel several times. Merolla et al. (2016) showed that DNN can be robust to more than just weight binarization. They applied several different distortions to the weights, including additive and multiplicative noise, and a class of non-linear projections.This was shown to improve robustness to other distortions and even boost results. Zheng and Tang (2016) tried to apply our binarization scheme to recurrent neural network for language modeling and achieved comparable results as well. Andri et al. (2016) even created a hardware implementation to speed up BNNs.

8. Conclusion

We have introduced BNNs, which binarize deep neural networks and can lead to dramatic improvements in both power consumption and computation speed. During the forward pass (both at run-time and train-time), BNNs drastically reduce memory size and accesses, and replace most arithmetic operations with bit-wise operations. Our estimates indicate that power efficiency can be improved by more than one order of magnitude (see Section 5). In terms of speed, we programmed a binary matrix multiplication GPU kernel that enabled

running MLP over the MNIST datset 7 times faster (than with an unoptimized GPU kernel) without any loss of accuracy (see Section 6).

We have shown that BNNs can handle MNIST, CIFAR-10 and SVHN while achieving nearly state-of-the-art accuracy. While our results for the challenging ImageNet are not on par with the best results achievable with full precision networks, they significantly improve all previous attempts to compress ImageNet-capable architectures. Moreover, by quantizing the weights and activations to more than 1-bit (i.e., QNNs), we have been able to achieve comparable results to the 32-bit floating point architectures (see Section 4.4 and supplementary material - Appendix B). A major open research avenue would be to further improve our results on ImageNet. Substantial progress in this direction might go a long way towards facilitating DNN usability in low power instruments such as mobile phones.

Acknowledgments

We would like to express our appreciation to Elad Hoffer, for his technical assistance and constructive comments. We thank our fellow MILA lab members who took the time to read the article and give us some feedback. We thank the developers of Torch, (Collobert et al., 2011) a Lua based environment, and Theano (Bergstra et al., 2010; Bastien et al., 2012), a Python library that allowed us to easily develop fast and optimized code for GPU. We also thank the developers of Pylearn2 (Goodfellow et al., 2013a) and Lasagne (Dieleman et al., 2015), two deep learning libraries built on the top of Theano. We thank Yuxin Wu for helping us compare our GPU kernels with cuBLAS. We are also grateful for funding from NSERC, the Canada Research Chairs, Compute Canada, and CIFAR. We are also grateful for funding from CIFAR, NSERC, IBM, Samsung. This research was supported by The Israel Science Foundation (grant No. 1890/14)

Appendix A. Implementation Details

In this section we give full implementation details over our MNIST,SVHN, CIFAR-10 and ImageNet datasets.

A.1 MLP on MNIST (Theano)

MNIST is an image classification benchmark dataset (LeCun et al., 1998). It consists of a training set of 60K and a test set of 10K 28×28 gray-scale images representing digits ranging from 0 to 9. In order for this benchmark to remain a challenge, we did not use any convolution, data-augmentation, preprocessing or unsupervised learning. The Multi-Layer-Perceptron (MLP) we train on MNIST consists of 3 hidden layers of 4096 binary units and a L2-SVM output layer; L2-SVM has been shown to perform better than Softmax on several classification benchmarks (Tang, 2013; Lee et al., 2015). We regularize the model with Dropout (Srivastava et al., 2014). The square hinge loss is minimized with the ADAM adaptive learning rate method (Kingma and Ba, 2015). We use an exponentially decaying global learning rate, as per Algorithm 1, and also scale the learning rates of the weights with their initialization coefficients from (Glorot and Bengio, 2010), as suggested by Courbariaux et al. (2015). We use Batch Normalization with a minibatch of size 100 to speed up the training. As is typical, we use the last 10K samples of the training set as a validation set for early stopping and model selection. We report the test error rate associated with the best validation error rate after 1000 epochs (we do not retrain on the validation set).

A.2 MLP on MNIST (Torch7)

We use a similar architecture as in our Theano experiments, without dropout, and with 2048 binary units per layer instead of 4096. Additionally, we use the shift-base AdaMax and BN (with a minibatch of size 100) instead of the vanilla implementations, to reduce the number of multiplications. Likewise, we decay the learning rate by using a 1-bit right shift every 10 epochs.

A.3 ConvNet on CIFAR-10 (Theano)

CIFAR-10 is an image classification benchmark dataset. It consists of a training set of size 50K and a test set of size 10K, where instances are 32×32 color images representing airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships and trucks. We do not use data-augmentation (which can really be a game changer for this dataset; see Graham 2014). The architecture of our ConvNet is identical to that used by Courbariaux et al. (2015) except for the binarization of the activations. The Courbariaux et al. (2015) architecture is itself mainly inspired by VGG (Simonyan and Zisserman, 2015). The square hinge loss is minimized with ADAM. We use an exponentially decaying learning rate, as we did for MNIST. We scale the learning rates of the weights with their initialization coefficients from (Glorot and Bengio, 2010). We use Batch Normalization with a minibatch of size 50 to speed up the training. We use the last 5000 samples of the training set as a validation set. We report the test error rate associated with the best validation error rate after 500 training epochs (we do not retrain on the validation set).

Table 7: Architecture of our CIFAR-10 ConvNet. We only use "same" convolutions as in VGG (Simonyan and Zisserman, 2015).

CIFAR-10 ConvNet architecture
Input: 32×32 - RGB image
3×3 - 128 convolution layer
BatchNorm and Binarization layers
3×3 - 128 convolution and 2×2 max-pooling layers
BatchNorm and Binarization layers
3×3 - 256 convolution layer
BatchNorm and Binarization layers
3×3 - 256 convolution and 2×2 max-pooling layers
BatchNorm and Binarization layers
3×3 - 512 convolution layer
BatchNorm and Binarization layers
3×3 - 512 convolution and 2×2 max-pooling layers
BatchNorm and Binarization layers
1024 fully connected layer
BatchNorm and Binarization layers
1024 fully connected layer
BatchNorm and Binarization layers
10 fully connected layer
BatchNorm layer (no binarization)
Cost: Mean square hinge loss

A.4 ConvNet on CIFAR-10 (Torch7)

We use the same architecture as in our Theano experiments. We apply shift-based AdaMax and BN (with a minibatch of size 200) instead of the vanilla implementations to reduce the number of multiplications. Likewise, we decay the learning rate by using a 1-bit right shift every 50 epochs.

A.5 ConvNet on SVHN

SVHN is also an image classification benchmark dataset. It consists of a training set of size 604K examples and a test set of size 26K, where instances are 32×32 color images representing digits ranging from 0 to 9. In both sets of experiments, we follow the same procedure used for the CIFAR-10 experiments, with a few notable exceptions: we use half the number of units in the convolution layers, and we train for 200 epochs instead of 500 (because SVHN is a much larger dataset than CIFAR-10).

A.6 ConvNet on ImageNet

ImageNet classification task consists of a training set of size 1.2M samples and test set of size 50K. Each instance is labeled with one of 1000 categories including objects, animals, scenes, and even some abstract shapes.

A.6.1 AlexNet

Our AlexNet implementation ("One weird trick" - Krizhevsky, 2014) consists of 5 convolution layers followed by 3 fully connected layers (see Section 8). Additionally, we use Adam as our optimization method and batch-normalization layers (with a minibatch of size 512). Learning rate was set to 0.01 and decrease to 10^{-7} by dividing by 2 every 20 epochs. We did not use Glorot et al. (2011) weight initialization coefficients.

A.6.2 GoogleNet

Our GoogleNet implementation consist of 2 convolution layers followed by 10 inception layers, spatial-average-pooling and a fully connected classifier. We also used the 2 auxilary classifiers. Additionally, we use Adam (Kingma and Ba, 2015) as our optimization method and batch-normalization layers (with a minibatch of size 64). Learning rate was set to 0.1 and decrease to 10^{-7} by dividing by 2 every 10 epochs.

Table 8: Our AlexNet Architecture.

AlexNet ConvNet architecture
Input: 224×224 - RGB image
11×11 - 64 convolution layer and 3×3 max-pooling layers
BatchNorm and Binarization layers
5×5 - 192 convolution layer and 3×3 max-pooling layers
BatchNorm and Binarization layers
3×3 - 384 convolution layer
BatchNorm and Binarization layers
3×3 - 384 convolution layer
BatchNorm and Binarization layers
3×3 - 256 convolution layer
BatchNorm and Binarization layers
4096 fully connected layer
BatchNorm and Binarization layers
4096 fully connected layer
BatchNorm and Binarization layers
1000 fully connected layer
BatchNorm layer (no binarization)
SoftMax layer (no binarization)
Cost: Negative log likelihood

References

Renzo Andri, Lukas Cavigelli, Davide Rossi, and Luca Benini. Yodann: An ultra-low power convolutional neural network accelerator based on binary weights. In *VLSI (ISVLSI), 2016 IEEE Computer Society Annual Symposium on*, pages 236–241. IEEE, 2016.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

Carlo Baldassi, Alessandro Ingrosso, Carlo Lucibello, Luca Saglietti, and Riccardo Zecchina. Subdominant Dense Clusters Allow for Simple Learning and High Computational Performance in Neural Networks with Discrete Synapses. *Physical Review Letters*, 115(12): 1–5, 2015.

Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. Theano: new features and speed improvements. Workshop on Deep Learning and Unsupervised Feature Learning, Advances in Neural Information Processing Systems (NIPS), 2012.

Michael J Beauchamp, Scott Hauck, Keith D Underwood, and K Scott Hemmert. Embedded floating-point units in FPGAs. In *Proceedings of the 2006 ACM/SIGDA 14th international symposium on Field programmable gate arrays*, pages 12–20. ACM, 2006.

Yoshua Bengio. Estimating or propagating gradients through stochastic neurons. Technical Report arXiv:1305.2982, Universite de Montreal, 2013.

James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, 2010.

Tianshi Chen, Zidong Du, Ninghui Sun, Jia Wang, Chengyong Wu, Yunji Chen, and Olivier Temam. Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning. In *Proceedings of the 19th international conference on Architectural support for programming languages and operating systems*, pages 269–284. ACM, 2014a.

Wenlin Chen, James Wilson, Stephen Tyree, Kilian Weinberger, and Yixin Chen. Compressing neural networks with the hashing trick. In *International Conference on Machine Learning (ICML)*, pages 2285–2294, 2015.

Yunji Chen, Tao Luo, Shaoli Liu, Shijin Zhang, Liqiang He, Jia Wang, Ling Li, Tianshi Chen, Zhiwei Xu, Ninghui Sun, et al. Dadiannao: A machine-learning supercomputer. In *Microarchitecture (MICRO), 2014 47th Annual IEEE/ACM International Symposium on*, pages 609–622. IEEE, 2014b.

Zhiyong Cheng, Daniel Soudry, Zexi Mao, and Zhenzhong Lan. Training binary multilayer neural networks for image classification using expectation backpropgation. *arXiv preprint arXiv:1503.03562*, 2015.

Adam Coates, Brody Huval, Tao Wang, David Wu, Bryan Catanzaro, and Ng Andrew. Deep learning with COTS HPC systems. In *International conference on machine learning (ICML)*, pages 1337–1345, 2013.

Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. Torch7: A matlab-like environment for machine learning. In *Workshop on BigLearn, Advances in Neural Information Processing Systems (NIPS)*, 2011.

Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Training deep neural networks with low precision multiplications. *ArXiv e-prints*, abs/1412.7024, December 2014.

Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3123–3131, 2015.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1370–1380, 2014.

Sander Dieleman, Jan Schlüter, Colin Raffel, Eben Olson, Søren Kaae Sønderby, Daniel Nouri, Daniel Maturana, Martin Thoma, Eric Battenberg, Jack Kelly, et al. Lasagne: first release. *Zenodo: Geneva, Switzerland*, 3, 2015.

Steve K Esser, Rathinakumar Appuswamy, Paul Merolla, John V Arthur, and Dharmendra S Modha. Backpropagation for energy-efficient neuromorphic computing. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1117–1125, 2015.

Clément Farabet, Yann LeCun, Koray Kavukcuoglu, Eugenio Culurciello, Berin Martini, Polina Akselrod, and Selcuk Talay. Large-scale fpga-based convolutional networks. *Scaling up Machine Learning: Parallel and Distributed Approaches*, pages 399–419, 2011a.

Clément Farabet, Berin Martini, Benoit Corda, Polina Akselrod, Eugenio Culurciello, and Yann LeCun. Neuflow: A runtime reconfigurable dataflow processor for vision. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pages 109–116. IEEE, 2011b.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *JMLR W&CP: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, volume 9, pages 249–256, May 2010.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *International conference on machine learning (ICML)*, pages 513–520, 2011.

Yunchao Gong, Liu Liu, Ming Yang, and Lubomir Bourdev. Compressing deep convolutional networks using vector quantization. *arXiv preprint arXiv:1412.6115*, 2014.

Ian J. Goodfellow, David Warde-Farley, Pascal Lamblin, Vincent Dumoulin, Mehdi Mirza, Razvan Pascanu, James Bergstra, Frédéric Bastien, and Yoshua Bengio. Pylearn2: a machine learning research library. *arXiv preprint arXiv:1308.4214*, 2013a.

Ian J. Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. In *International Conference on Machine Learning (ICML)*, pages 1319–1327, 2013b.

Gokul Govindu, Ling Zhuo, Seonil Choi, and Viktor Prasanna. Analysis of high-performance floating-point arithmetic on FPGAs. In *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*, page 149. IEEE, 2004.

Benjamin Graham. Spatially-sparse convolutional neural networks. *arXiv preprint arXiv:1409.6070*, 2014.

Alex Graves. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2348–2356, 2011.

Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In *International Conference on Machine Learning (ICML)*, pages 1737–1746, 2015.

Philipp Gysel, Mohammad Motamedi, and Soheil Ghiasi. Hardware-oriented approximation of convolutional neural networks. *arXiv preprint arXiv:1604.03168*, 2016.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1135–1143, 2015.

Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations (ICLR)*, 2016.

Geoffrey Hinton. Neural networks for machine learning. Coursera, video lectures, 2012.

Geoffrey Hinton, Li Deng, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Mark Horowitz. Computing's Energy Problem (and what we can do about it). *IEEE Interational Solid State Circuits Conference*, pages 10–14, 2014.

Kyuyeon Hwang and Wonyong Sung. Fixed-point feedforward deep neural network design using weights+ 1, 0, and- 1. In *Signal Processing Systems (SiPS), 2014 IEEE Workshop*, pages 1–6. IEEE, 2014.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, pages 448–456, 2015.

Minje Kim and Paris Smaragdis. Bitwise neural networks. *arXiv preprint arXiv:1601.06071*, 2016.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems (NIPS)*, pages 1097–1105, 2012.

Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Artificial Intelligence and Statistics*, pages 562–570, 2015.

Chen-Yu Lee, Patrick W Gallagher, and Zhuowen Tu. Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. In *Artificial Intelligence and Statistics*, pages 464–472, 2016.

Zhouhan Lin, Matthieu Courbariaux, Roland Memisevic, and Yoshua Bengio. Neural networks with few multiplications. *ArXiv e-prints*, abs/1510.03009, October 2015.

Chris Lomont. Fast inverse square root. Technical report, Indiana: Purdue University, 2003.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

Paul Merolla, Rathinakumar Appuswamy, John Arthur, Steve K Esser, and Dharmendra Modha. Deep neural networks are robust to weight binarization and other non-linear distortions. *arXiv preprint arXiv:1606.01981*, 2016.

Tomas Mikolov and Geoffrey Zweig. Context dependent recurrent neural network language model. In *SLT*, pages 234–239, 2012.

Daisuke Miyashita, Edward H Lee, and Boris Murmann. Convolutional neural networks using logarithmic data representation. *arXiv preprint arXiv:1603.01025*, 2016.

Volodymyr Mnih, Koray Kavukcuoglo, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidgeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharsan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks. *Google Research Blog. Retrieved June*, 20:14, 2015.

Joachim Ott, Zhouhan Lin, Ying Zhang, Shih-Chii Liu, and Yoshua Bengio. Recurrent neural networks with limited numerical precision. *arXiv preprint arXiv:1608.06902*, 2016.

Phi-Hung Pham, Darko Jelaca, Clement Farabet, Berin Martini, Yann LeCun, and Eugenio Culurciello. Neuflow: dataflow vision processing system-on-a-chip. In *Circuits and Systems (MWSCAS), 2012 IEEE 55th International Midwest Symposium on*, pages 1044–1047. IEEE, 2012.

Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, pages 525–542. Springer, 2016.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

Tara Sainath, Abdel rahman Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran. Deep convolutional neural networks for LVCSR. In *ICASSP 2013*, 2013.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

Daniel Soudry, Itay Hubara, and Ron Meir. Expectation backpropagation: Parameter-free training of multilayer neural networks with continuous or discrete weights. In *Advances in Neural Information Processing Systems (NIPS)*, pages 963–971, 2014.

H Spang and P Schultheiss. Reduction of quantizing noise by use of feedback. *IRE Transactions on Communications Systems*, 10(4):373–380, 1962.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3104–3112, 2014.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

Yichuan Tang. Deep learning using linear support vector machines. 2013.

Naoya Torii, Hirotaka Kokubo, Dai Yamamoto, Kouichi Itoh, Masahiko Takenaka, and Tsutomu Matsumoto. Asic implementation of random number generators using sr latches and its evaluation. *EURASIP Journal on Information Security*, 2016(1):1–12, 2016.

Vincent Vanhoucke, Andrew Senior, and Mark Z Mao. Improving the speed of neural networks on CPUs. In *Workshop on Deep Learning and Unsupervised Feature Learning, Neural Information Processing Systems (NIPS)*, 2011.

Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International Conference on Machine Learning (ICML)*, pages 1058–1066, 2013.

Xiangyu Zhang, Jianhua Zou, Xiang Ming, Kaiming He, and Jian Sun. Efficient and accurate approximations of nonlinear convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1984–1992, 2015.

Weiyi Zheng and Yina Tang. Binarized neural networks for language modeling. Technical Report cs224d, Stanford University, 2016.

Shuchang Zhou, Zekun Ni, Xinyu Zhou, He Wen, Yuxin Wu, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016.

Journal of Machine Learning Research 18 (2018) 1-48 Submitted 7/17; Revised 12/17; Published 4/18

Significance-based community detection in weighted networks

John Palowitch* palojj@email.unc.edu
Shankar Bhamidi bhamidi@email.unc.edu
Andrew B. Nobel nobel@email.unc.edu
Department of Statistics and Operations Research
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599
**Now at Google*

Editor: Michael Mahoney

Abstract

Community detection is the process of grouping strongly connected nodes in a network. Many community detection methods for *un*-weighted networks have a theoretical basis in a null model. Communities discovered by these methods therefore have interpretations in terms of statistical significance. In this paper, we introduce a null for weighted networks called the continuous configuration model. First, we propose a community extraction algorithm for weighted networks which incorporates iterative hypothesis testing under the null. We prove a central limit theorem for edge-weight sums and asymptotic consistency of the algorithm under a weighted stochastic block model. We then incorporate the algorithm in a community detection method called CCME. To benchmark the method, we provide a simulation framework involving the null to plant "background" nodes in weighted networks with communities. We show that the empirical performance of CCME on these simulations is competitive with existing methods, particularly when overlapping communities and background nodes are present. To further validate the method, we present two real-world networks with potential background nodes and analyze them with CCME, yielding results that reveal macro-features of the corresponding systems.

Keywords: Community detection; Multiple testing; Network models; Weighted networks; Unsupervised learning

1. Introduction

For decades, the development of graph theory and network science has produced a wide array of quantitative tools for the study of complex systems. Network-based data analysis methods have driven advances in areas as diverse as social science, systems biology, life sciences, marketing, and computer science (cf. Palla et al., 2007; Barabasi and Oltvai, 2004; Lusseau and Newman, 2004; Guimera and Amaral, 2005; Reichardt and Bornholdt, 2007; Andersen et al., 2012). Thorough surveys of the network science and methodology literature have been provided by Newman (2003) and Jacobs and Clauset (2014), among others.

A common exploratory technique for networks called "community detection" involves the search for node subsets ("communities") that are somehow internally related. Usually, it is assumed that links or "edges" between nodes in the same community are more common than those between nodes in different communities. Thus, the specific goal of community detection is often to find groups of nodes that are strongly intraconnected and weakly interconnected (Newman, 2004b), which is known as "assortative" structure. Some approaches allow for "disassortative" structure, with *weak* intraconnection and *strong* interconnection (Karrer and Newman, 2011; Aicher et al., 2014). Regardless of one's broad notion of community structure, there are hundreds of distinct criteria and methods for its discovery. Community detection most often proceeds in an *unsupervised* fashion, so that communities are found through the network edge structure alone.

Since communities in real-world networks often arise due to some explanatory systematic mechanism, community detection has been an important part of both data-driven decision-making and scientific inquiry (Danon et al., 2005). For instance, community detection has been used to facilitate recommender systems in online social networks (e.g. Sahebi and Cohen, 2011; Xin et al., 2014), and has been used to "hone in" on regions of genomes (human and otherwise) for a variety of downstream analyses (e.g. Cabreros et al., 2016; Platig et al., 2015; Fan et al., 2012). Myriad examples of community detection applications can be found in Porter et al. (2009) and Fortunato (2010), and the references therein.

Many community detection methods are based on a null model, which in this context means a random network model without explicit community structure. For un-weighted networks the most common null is the configuration model (Bollobás, 1980; Bender, 1974) or a related model like that of Chung and Lu (2002a,b). Historically, the most common approach involving a null model is the use of a node partition score that is large when nodes within the cells of the partition are highly interconnected, relative to what is expected under the null (Fortunato, 2010; Newman, 2006). Arguably the most famous example of such a criterion is modularity, introduced by Newman and Girvan (2004). Various algorithms have been created to search directly for partitions of a network with large modularity (see Clauset et al., 2004; Blondel et al., 2008), while other approaches use modularity as an auxiliary criterion (see Langone et al., 2011). More recent approaches incorporate community-specific criteria which are large when the community is preferentially intra-connected, allowing for community *extraction* algorithms (e.g. Zhao et al., 2011; Lancichinetti et al., 2011; Wilson et al., 2014).

Generally speaking, communities found by null-based community detection methods can be said to have exhibited behavior strongly departing from the null. The results of these methods therefore carry a statistical *testing* interpretation unavailable to most methods without this foundation, like spectral clustering (White and Smyth, 2005; Zhang et al., 2007) or those based on random walks (e.g. Pons and Latapy, 2006). In particular, recent methods put forth by Lancichinetti et al. (2011) and Wilson et al. (2014) for binary networks use the theoretical properties of the configuration model to detect "background" nodes that are not significantly connected to any community. These methods incorporate tail behavior of node-wise graph statistics under the configuration model in a way that others do not. Note that another class of methods capable of assessing significance are focused on specifying *alternative* (non-null) models for communities. Once fit, the models can be compared with the null-partition via likelihood ratios (e.g. Yan et al., 2014) or Bayes factors (e.g. Peixoto, 2015).

A significant drawback of null-based community detection methodology is that no explicit null model exists for edge-weighted networks. Edge weights are commonplace in network data, and can provide information that improves community detection power and specificity (Newman, 2004a). While many existing community detection methods have been established for weighted and un-weighted networks alike, due to the absence of an appropriate weighted-network null model, these methods do not provide rigorous significance assessments of weighted-network communities. For instance, the aforementioned method from Lancichinetti et al. (2011), called OSLOM, can incorporate edge weights, but uses an exponential function to calculate nominal tail probabilities for edge weight sums, a testing approach which is not based on an explicit null. As a consequence, communities in *weighted* networks identified by OSLOM may in some cases be spurious or unreliable, especially when no "true" communities exist.

The key methodological contributions in this article are as follows: (i) we provide an explicit null model for networks with weighted edges, (ii) we present a community extraction method based on hypothesis tests under the null, and (iii) we analyze the consistency properties of the method's core algorithm with respect to a weighted stochastic block model. These contributions provide the beginnings of a rigorous statistical framework with which to study communities in weighted networks. Through extensive simulations, we show that the accuracy of our proposed extraction method is highly competitive with other community detection approaches on weighted networks with both disjoint and overlapping communities, and on weighted networks with background nodes. Importantly, the weighted stochastic block model employed (in both the theoretical and empirical studies) allows for arbitrary expected degree and weighted-degree distributions, reflecting degree heterogeneity observed in real-world networks. To further validate the method, we apply it to two real data sets with (arguably) potential overlapping communities and background nodes. We show that the proposed method recovers sensible features of the real data, in contrast to other methods.

1.1 Paper organization

The rest of the paper is organized as follows. We start by introducing general notation in Section 1.2. In Section 2 we motivate and state the continuous configuration model. In Section 3, we introduce a core algorithm to search for communities using multiple hypothesis testing under the model. In Section 4, we prove both a central limit theorem and a consistency result for the primary test statistic in the core algorithm. We describe the implementation and application of the core algorithm in Section 5, and evaluate its empirical efficacy on simulations and real data in Section 6 and 7 (respectively). We close with a discussion in Section 8.

1.2 Notation and terminology

We denote an undirected weighted network on n nodes by a triple $\mathcal{G} := (N, A, W)$, where $N := \{1, \ldots, n\}$ is the node set with u, v as general elements, A is the adjacency matrix with $A_{uv} = 1$ if and only if there is an edge between u and v, and W is the weight matrix with non-negative entries W_{uv} containing edge weights between nodes u and v. Note that $A_{uv} = 0$ implies $W_{uv} = 0$, but W_{uv} may be zero even when $A_{uv} = 1$. This allows for networks with potentially zero edge weights; for instance, an online social network from which friendship

links are edges and message counts are edge weights. The degree of a node u is defined by $d(u) := \sum_{v \in N} A_{uv}$, and we denote the vector of node degrees by $\mathbf{d} = (d_1, \ldots, d_n)$. In an analogous fashion, we define the *strength* of a node by $s(u) := \sum_{v \in N} W_{uv}$, and the strength vector of the network by $\mathbf{s} = (s(1), \ldots, s(n))$. The total degree and strength of $\mathcal{G}$ are given by $d_T := \sum_{v \in N} d(v)$ and $s_T := \sum_{v \in N} s(v)$, respectively.

2. The continuous configuration model

To motivate the null model, we first explain the intuition behind the binary configuration model for unweighted networks. The binary configuration model for an n-node network is based on a given degree vector $\mathbf{d}$ corresponding to the nodes. Studied originally in Bollobás (1980) and Bender (1974), the model is equivalent to a process in which each node u receives $d(u)$ half-edges, which are paired uniformly-at-random without replacement until no half-edges remain (Molloy and Reed, 1995). In other words, the model guarantees a graph with degrees $\mathbf{d}$ but otherwise uniformly distributed edges. Therefore, given an observed network with degrees $\mathbf{d}$, a typical draw from the configuration model under $\mathbf{d}$ represents that network without any community structure.

Due to these characteristics, many community detection methods proceed by identifying node sets having intra-connectivity significantly beyond what is expected under the model. For instance, the modularity measure, introduced by Newman and Girvan (2004), scores node partitions of binary networks according to the observed versus configuration model-expected edge densities of the communities. The methods OSLOM (Lancichinetti et al., 2011) and ESSC (Wilson et al., 2014) use the configuration model to assess the statistical significance of the deviations graph statistics from their configuration model-expected values. Methods based on the configuration model are able to properly distinguish true community structure from "false positive" community structure arising from inherently high connectivity. For instance, Karrer and Newman (2011) showed that on a political blogs network, modularity maximization found communities that agreed with political affiliation, whereas the same algorithm without node degree input found a naíve split of the network into more active and less active blogs.

The degrees $\mathbf{d}$ of the configuration model can be thought of as the nodes' relative propensities to form ties. Chung and Lu made this notion explicit by defining a Bernoulli-based model for a n-node unweighted network with a given expected degree sequence (Chung and Lu, 2002b). Under this model, the probability of nodes u and v sharing an edge is exactly $d(u)d(v)/d_T$. As null models for community detection, the Chung-Lu and configuration are often interchangeable (Durak et al., 2013). Indeed, for sparse graphs it can be shown that the probability of an edge between u and v under the configuration model is approximately the Chung-Lu probability. The *continuous* configuration model, introduced below, extends the spirit of the configuration and Chung-Lu models by taking both observed degrees $\mathbf{d}$ and strengths $\mathbf{s}$ as node propensities for (respectively) edge connection and edge weight.

In the next section, we use the following notation to concisely express the model. Given a vector $\mathbf{x}$ of dimension n, we define for any indices $u, v \in N$ the ratio

$$r_{uv}(\mathbf{x}) := \frac{x(u)x(v)}{\sum_{w \in N} x(w)} \tag{1}$$

Define $\tilde{r}_{uv}(\mathbf{x}) := \min\{1, r_{uv}(\mathbf{x})\}$. Note that when $\mathbf{x}$ is a degree sequence $\mathbf{d}$, $r_{uv}(\mathbf{d})$ is the Chung-Lu probability of an edge between nodes u and v. Finally, for a vector $\mathbf{y}$ of dimension n, define $f_{uv}(\mathbf{x}, \mathbf{y}) := r_{uv}(\mathbf{y})/\tilde{r}_{uv}(\mathbf{x})$.

2.1 Model statement

The continuous configuration model on n nodes has the parameter triple $\theta := (\mathbf{d}, \mathbf{s}, \kappa)$, where $\mathbf{d} \in \{1, 2, 3, \ldots\}^n$ is a degree vector, $\mathbf{s} \in [0, \infty)^n$ is a strength vector, and $\kappa > 0$ is a variance parameter. Let F be a distribution on the non-negative real line with mean one and variance κ. The model specifies a random weighted graph $\mathcal{G} := (N, A, W)$ on n nodes as follows:

1. $\mathbb{P}(A_{uv} = 1) = \tilde{r}_{uv}(\mathbf{d})$ independently for all node pairs $u, v \in N$
2. For each node pair u, v, generate an independent random variable ξ_{uv} according to F, and assign edge weights by: $W_{uv} = \begin{cases} f_{uv}(\mathbf{d}, \mathbf{s})\xi_{uv}, & A_{uv} = 1 \\ 0, & A_{uv} = 0 \end{cases}$

The edge generation defined by step 1 is equivalent to the Chung-Lu model: edge indicators are Bernoulli, with probabilities adjusted by the propensities $\mathbf{d}$. The weight generation in step 2 mirrors this process. Edge weights follow the distribution F, with means adjusted by the propensities $\mathbf{s}$, through $f(\mathbf{d}, \mathbf{s})$. Note that $\tilde{r}_{uv}(\mathbf{d})$ differs from $r_{uv}(\mathbf{d})$ only when the degree distribution is extremely right-skewed or highly dispersed, and then only when one of u or v has a large degree. Truncating edge probabilities is standard in approaches based on the Chung-Lu model (e.g. Zhao et al., 2012).

It is easily derived from the model that

$$P(A_{uv} = 1) = \frac{d(u)d(v)}{d_T} \quad \text{and} \quad \mathbb{E}(W_{uv}) = \frac{s(u)s(v)}{s_T}, \tag{2}$$

equations which extend the binary-network notion of null behavior to edge weights. (The first equation will be approximate for any u, v for which $r_{uv}(\mathbf{d}) > 1$.) The equations in (2) imply that

$$\mathbb{E}(D(u)) = d(u) \quad \text{and} \quad \mathbb{E}(S(u)) = s(u) \ \text{ for all } \ u \in N. \tag{3}$$

where $D(u)$ and $S(u)$ are the (random) degree and strength of u under the model. Thus, the continuous configuration model can be thought of as null weighted network with given expected degrees and given expected strengths.

2.2 Null model parameter specification

When the *binary* configuration model is used for community detection, the degree parameter of the model is set to the observed degree distribution of the network. In a sense, this is an *estimation* of the nodes' connection propensities under the null. Similarly, to use the continuous configuration model in practice, we derive the parameter θ from the data at hand. Given an observed network $\mathcal{G}$, we straightforwardly use the observed degrees and strengths $\mathbf{d}$ and $\mathbf{s}$ as the first two parameters of the model. The third parameter of the continuous configuration model, κ, is also computed from the $\mathcal{G}$, and meant to capture its

observed average edge-weight variance. We use the following method-of-moments estimator to specify κ:

$$\hat{\kappa}(\mathbf{d}, \mathbf{s}) := \sum_{u,v:A_{uv}=1} (W_{uv} - f_{uv}(\mathbf{d}, \mathbf{s}))^2 \Big/ \sum_{u,v:A_{uv}=1} f_{uv}(\mathbf{d}, \mathbf{s})^2 \tag{4}$$

This estimator is derived as follows. Under the continuous configuration model with $\mathbf{d}$ and $\mathbf{s}$,

$$\text{Var}\left(W_{uv} \,\middle|\, A_{uv} = 1\right) \;=\; f_{uv}(\mathbf{d}, \mathbf{s})^2 \,\text{Var}\left(\xi_{uv}\right) \;=\; f_{uv}(\mathbf{d}, \mathbf{s})^2 \,\kappa. \tag{5}$$

Therefore

$$\begin{aligned} \mathbb{E}\left\{ \sum_{u,v:A_{uv}=1} (W_{uv} - f_{uv}(\mathbf{d}, \mathbf{s}))^2 \;\middle|\; A \right\} &= \sum_{u,v:A_{uv}=1} \text{Var}\left(W_{uv} \mid A_{uv} = 1\right) \\ &= \kappa \sum_{u,v:A_{uv}=1} f_{uv}(\mathbf{d}, \mathbf{s})^2, \end{aligned}$$

Dividing through by $\sum_{u,v:A_{uv}=1} f_{uv}(\mathbf{d}, \mathbf{s})$ motivates equation 4.

Strictly speaking, the distribution F is also a parameter of the model. However, for testing purposes we do not require a null specification of F. As we discuss in the next section, p-values from the model will be based on a central limit theorem that requires only a third-moment assumption on F. While estimating F could improve the model's efficacy as a null, in general this would require potentially costly computational procedures, and additional theoretical assumptions that might be difficult to support or verify in practice. The specification of F will be most useful for applications of the model that involve simulations or likelihood-based analyses.

In subsequent sections, we describe a procedure for testing nodal connectivity against the continuous configuration model, with parameters derived from the real data (as above). These parameters pertain only to a *null* model, and not to any particular community-laden generative network model. Our procedure (to be introduced) is built upon hypothesis testing under this estimated null. Note that this is in contrast to a broad class of community detection methods which are all based on estimating a "Stochastic Block Model" (SBM), a generative network model with parameters controlling the presence and strength of communities (e.g. Nowicki and Snijders, 2001; Karrer and Newman, 2011; Aicher et al., 2014; Peixoto, 2017). In particular, the work of Peixoto (2017) introduces a weighted-edge SBM estimation procedure which automatically chooses the number of communities, and allows for a notion of community overlap. We feature this method in our simulation experiments, introduced in Section 6.

3. Test statistic and update algorithm

In this section we introduce a core testing-based community detection algorithm based on the continuous configuration model. The algorithm *updates* an input node set via node-to-set hypothesis testing. We describe the repeated application of the algorithm as part of an iterative set-update approach to community detection, following some recently-introduced methods (e.g. Lancichinetti et al., 2011; Wilson et al., 2014).

Formally, we define the core algorithm operation as a set-update map $U_\alpha(\,\cdot\,, \mathcal{G}) : 2^N \mapsto 2^N$, indexed by a parameter $\alpha \in (0,1)$. Given a weighted network $\mathcal{G}$ and candidate set $B \subseteq N$, the update $U_\alpha(B, \mathcal{G})$ outputs a new set B' formed by the nodes from N that have statistically significant association to B at level α, after a multiple-testing correction. The node-wise hypothesis tests are based on the summary statistic

$$S(u, B, \mathcal{G}) := \sum_{v \in B} W_{uv}, \tag{6}$$

which is the sum of all weights on edges incident with u and B. When the observed value of $S(u, B, \mathcal{G})$ is much larger than its expectation under the continuous configuration model, there is evidence to support an association between u and B resulting from some form of "ground-truth" community structure in the network. We assess the strength of evidence, that is, the significance of $S(u, B, \mathcal{G})$, with the p-value

$$p(u, B, \mathcal{G}) := \mathbb{P}\left(S(u, B, \widetilde{\mathcal{G}}) > S(u, B, \mathcal{G})\right), \tag{7}$$

where $\widetilde{\mathcal{G}}$ is random with respect to $\mathbb{P}$, the distribution of the continuous configuration model with parameters $\mathbf{d}, \mathbf{s}$, and $\hat{\kappa}(\mathbf{d}, \mathbf{s})$ (see Section 2.2). The update U_α is then:

Core update U_α

1. Given: graph $\mathcal{G}$ with nodes N and input set $B \subseteq N$
2. Calculate p-values $\mathbf{p} := \{p(u, B, \mathcal{G}) : u \in N\}$
3. Obtain threshold $\tau(\mathbf{p})$ from a multiple-testing procedure
4. Output set $B' = \{u : p(u, B, \mathcal{G}) \leqslant \tau(\mathbf{p})\}$

Many methods to compute a multiple-testing threshold $\tau(\mathbf{p})$ are available, the most stringent being the well-known Bonferroni correction. The correction we employ is the false discovery rate (FDR) control procedure of Benjamini and Hochberg (1995). Given a set of p-values $\mathbf{p} := \{p_u\}_{u \in N}$ corresponding to n hypothesis tests and a target FDR $\alpha \in (0,1)$, each p-value $p_u \in \mathbf{p}$ is associated with an adjusted p-value $p_u^* := n\, p_u \,/\, j(u)$ where $j(u)$ is the rank of p_u in $\mathbf{p}$, and $\tau(\mathbf{p}) := \max\{p_u : p_u^* \leqslant \alpha\}$. Benjamini and Hochberg show that, if the p-values corresponding to true null hypotheses are independent, the threshold $\tau(\mathbf{p})$ bounds the expected number of false discoveries at α.

The update U_α is an exploratory tool for moving an input set B closer to a "target" community. Consider that, if the initial set B has a majority group of nodes from some strongly-connected community C, the statistic $S(u, B, \mathcal{G})$ will be large for $u \in C$, and small otherwise. In this case, U_α applied to B will often return many nodes in C, and few nodes in C^c. Note that U_α can *remove* nodes from B, if they are not significantly connected, as well as add them.

If C is a community with strong enough signal, we should expect $U_\alpha(C, \mathbf{D})$ to return C. This reasoning motivates an algorithm that searches for "stable communities" C satisfying $U_\alpha(C, \mathbf{D}) = C$. By definition, all interior nodes of a stable community C are significantly

connected to C, and exterior nodes are not. We define a stable community search procedure, which iteratively applies U_α until convergence:

Stable community search (SCS) algorithm

1. Given weighted graph $\mathcal{G}$ with nodes N and initial set $B_0 \subseteq N$; $t \leftarrow 0$
2. Set $B_{t+1} \leftarrow U_\alpha(B_t, \mathcal{G})$ and $t \leftarrow t + 1$.
3. If $B_t = B_{t'}$ for some $t' < t$, terminate. Else, return to step 2.

Since the number of possible node subsets B_t is finite, SCS is guaranteed to terminate. There are some technicalities regarding use of this algorithm, like how to obtain B_0, and when in rare cases $t' < t - 1$. We relegate resolution of these issues to Section 5. For now, the update U_α and SCS raise two theoretical questions:

1. Is the p-value $p(u, B, \mathcal{G})$ analytically tractable? If not, is there a useful distributional approximation based on the continuous configuration model?

2. Consistency: with what power can SCS detect ground-truth community structure?

These questions are the focus of the next section.

4. Theoretical Results

We now address the theoretical questions raised at the end of the previous section by analyzing the distribution of the test statistic $S(u, B, \mathcal{G})$ under the continuous configuration model (for question 1) and an appropriate alternative model with planted community structure (for question 2). Both analyses have an asymptotic setting consisting of a sequence of random weighted networks. Denote this sequence by $\{\mathcal{G}_n\}_{n>1}$. If $\mathcal{G}_n$ is a continuous configuration model with parameters $\theta := (\mathbf{s}, \mathbf{d}, \kappa)$, the following proposition gives general expressions for the mean and standard deviation of $S(u, B, \mathcal{G}_n)$:

Proposition 1 *Let $\mathcal{G} = (N, A, W)$ be a random network generated by the continuous configuration model with parameters $\theta = (\mathbf{s}, \mathbf{d}, \kappa)$. For any $(u, B) \in N \times 2^N$, let $\mu(u, B|\theta)$ and $\sigma(u, B|\theta)$ be, respectively, the mean and standard deviation of $S(u, B, \mathcal{G})$ under $\mathcal{G}$. Then*

$$\mu(u, B|\theta) \equiv \mu(u, B|\mathbf{s}) = \sum_{v \in B} r_{uv}(\mathbf{s}) \tag{8}$$

and

$$\sigma(u, B|\theta)^2 = \sum_{v \in B} r_{uv}(\mathbf{s}) f_{uv}(\mathbf{d}, \mathbf{s})\left(1 - \tilde{r}_{uv}(\mathbf{d}) + \kappa\right) \tag{9}$$

The proof, given in Appendix A, follows from easy calculations with the model's generating procedure (see Section 2.1). All theoretical results will make use of the expressions defined in equations 8 and 9.

4.1 Asymptotic Normality of $S(u, B, \mathcal{G})$

A central limit theorem under the null model is now established for $S(u, B, \mathcal{G})$, yielding a closed-form approximation for the p-value in equation (7). This result is motivated by the fact that, under most non-trivial null parameter specifications, the distribution of $S(u, B, \mathcal{G})$ is not analytically tractable.

In the setting of the theorem, for any $n > 1$, a random network $\mathcal{G}_n$ is generated by a continuous configuration model with parameter $\theta_n := (\mathbf{d}_n, \mathbf{s}_n, \kappa_n)$ and common weight distribution F. The following regularity conditions are required on the sequence $\{\theta_n\}_{n>1}$. Let λ_n denote the average entry of $\mathbf{d}_n$, (which is the average expected degree of $\mathcal{G}_n$). For each $r \geqslant 0$ let $L_{n,r} := n^{-1} \sum_{u \in N} (d_n(u)/\lambda_n)^r$ be the normalized r^{th}-moment of $\mathbf{d}_n$. Note that $L_{n,1} = 1$. The regularity conditions are then as follows:

Assumption 1 *Define $e_n(u|\beta) := s_n(u)/d_n(u)^{1+\beta}$. There exists $\beta > 0$ such that*

$$0 < \liminf_{n\to\infty} \min_{u \in N} e_n(u|\beta) \quad \textit{and} \quad \limsup_{n\to\infty} \max_{u \in N} e_n(u|\beta) < \infty.$$

Assumption 2 *Let β be as in Assumption 1. There exists $\varepsilon > 0$ such that, for both $r = 4\beta + 2$ and $r = 4\beta + 2 + \varepsilon$,*

$$0 < \liminf_{n\to\infty} L_{n,r} \quad \textit{and} \quad \limsup_{n\to\infty} L_{n,r} < \infty$$

Assumption 3 $\limsup_{n\to\infty} \sup_{u,v \in N} r_{uv}(\mathbf{d}_n) < \infty$.

Assumption 4 *The sequence $\{\kappa_n\}_{n\geqslant 1}$ is bounded away from zero and infinity, and F has finite third moment.*

Assumption 1 reflects the common relationship between strengths and degrees in real-world weighted networks (Barrat et al., 2004; Clauset et al., 2009). Assumptions 2-3 are needed to control the extremal behavior of the degree distribution. They exclude, for instance, cases with a few nodes having $d_n(u) \asymp n$ and the remaining nodes having $d_n(u) = O(1)$. We note that the Assumption 2 becomes more stringent as β increases, since as β increases the strength-degree power law becomes more severe.

Theorem 2 *For each $n > 1$, let $\mathcal{G}_n$ be generated by the continuous configuration model with parameter θ_n and weight distribution F. Suppose $\{\theta_n\}_{n\geqslant 1}$ and F satisfy Assumptions 1-4. Fix a node sequence $\{u_n\}_{n\geqslant 1}$ with $u_n \in N$ and a positive integer sequence $\{b_n\}_{n\geqslant 1}$ with $b_n \leqslant n$. Suppose $d_n(u_n)b_n/n \to \infty$ as $n \to \infty$. Let $B_n \subseteq N$ be a node set chosen independently of $\mathcal{G}_n$ according to the uniform distribution on all sets of size b_n. Then*

$$\frac{S(u_n, B_n, \mathcal{G}_n) - \mu_n(u_n, B_n|\theta_n)}{\sigma_n(u_n, B_n|\theta_n)} \Rightarrow \mathcal{N}(0, 1) \quad \textit{as} \quad n \to \infty \tag{10}$$

The proof is given in Appendix B. Essentially, Theorem 2 says that $S(u, B, \mathcal{G})$ is asymptotically Normal provided that B is typical (i.e. a uniform draw) and that $d(u)$ and B are sufficiently large. The theorem justifies the following approximation of the p-value in (7):

$$p(u, B) \approx 1 - \Phi\left(\frac{S(u, B, \mathcal{G}) - \mu(u, B|\theta)}{\sigma(u, B|\theta)}\right) \tag{11}$$

Above, $\theta = (\mathbf{d}, \mathbf{s}, \hat{\kappa}(\mathbf{d}, \mathbf{s}))$ is computed from $\mathcal{G}$, as described in Section 2.2. Note that the numerator $S(u, B, \mathcal{G}) - \mu(u, B|\theta)$ is the contribution of B to the standard modularity score, as will be discussed in more detail in Sections 4.2.2 and 4.2.3. The variance $\sigma(u, B|\theta)$ thus standardizes the magnitude of B's modularity, providing an appropriate test statistic for community effect.

4.2 Consistency of SCS

In this section, we evaluate the ability of the SCS algorithm to identify true communities in a planted-community model. Explicitly, we consider a sequence of networks $\{\mathcal{G}_n\}_{n>1}$ where each network in the sequence is generated by a weighted stochastic block model (WSBM). The WSBM we employ is similar to that presented in Aicher et al. (2014), but is generalized to include node-specific weight parameters. In other words, it is "strength-corrected" as well as degree-corrected, in a manner analogous to the original degree-corrected SBM (Coja-Oghlan and Lanka, 2009). The proofs of Theorem 4 and Theorem 5 are given in Appendix C.

4.2.1 The weighted stochastic block model

For fixed $K > 1$, we define a K-block WSBM on $n > 1$ nodes as follows. Let $\mathbf{c}_n$ be a community partition vector with $c_n(u) \in \{1, \ldots, K\}$ giving the community index of u. Denote community i by $C_{i,n} := \{u : c_n(u) = i\}$. Define $\pi_{i,n} := n^{-1}|C_{i,n}|$ with $\boldsymbol{\pi}_n$ the associated vector. Let $\mathbf{P}$ and $\mathbf{M}$ be fixed $K \times K$ symmetric matrices with non-negative entries encoding intra- and inter-community baseline edge probabilities and edge weight expectations, respectively. Let ϕ_n and ψ_n be arbitrary n-vectors with positive entries, which are parameters giving nodes individual propensities to form edges and assign weight (separately from $\mathbf{P}$ and $\mathbf{M}$). To ensure proper edge probabilities, we assume that $\max(\phi_n)^2 \max(\mathbf{P}) \leqslant 1$. For identifiability, we assume the vectors ϕ_n and ψ_n sum to n. Finally, let F be a distribution on the non-negative real line with mean 1 and variance $\sigma^2 \geqslant 0$. The WSBM can then be specified as follows:

1. Place an edge between nodes u and v with probability $\mathbb{P}_n(A_{uv} = 1) = r_{uv}(\phi_n)\mathbf{P}_{c_n(u)c_n(v)}$, independently across node pairs.
2. For node pair u, v with $A_{uv} = 1$, generate an independent random variable ξ_{uv} according to F. Determine edge weight W_{uv} by:

$$W_{uv} = \begin{cases} f_{uv}(\psi_n, \phi_n)\mathbf{M}_{c_n(u)c_n(v)}\xi_{uv}, & A_{uv} = 1 \\ 0, & A_{uv} = 0 \end{cases}$$

The many parameters involved with this model allow for node heterogeneity and community structure. When $\mathbf{P}$ and $\mathbf{M}$ are proportional to a $K \times K$ matrix of ones, the WSBM

reduces to the continuous configuration model with parameters $\mathbf{d} \propto \phi$, $\mathbf{s} \propto \psi$, and $\kappa = \sigma^2$. Community structure is introduced in the network by allowing the diagonal entries of $\mathbf{P}$ and $\mathbf{M}$ to be arbitrarily larger than the off-diagonals.

4.2.2 Consistency theorem

The consistency analysis of SCS involves a sequence of random networks $\{\mathcal{G}_n\}_{n>1}$, where $\mathcal{G}_n$ is generated by a K-community WSBM. In this setting, we incorporate an additional parameter ρ_n, and let $\mathbf{P}_n := \rho_n \mathbf{P}$ replace $\mathbf{P}$ for each $n > 1$. This lets us distinguish the role of the asymptotic order of the average expected degree, defined $\lambda_n := n\rho_n$, from the profile of edge densities within and between communities ($\mathbf{P}$). Importantly, our results require only that $\lambda_n / \log n \to \infty$, reflecting the sparsity of real-world networks. Throughout this section, we denote the vector of (random) strengths from $\mathcal{G}_n$ by $\mathbf{S}_n$.

We now define an explicit notion of consistency in terms of the SCS algorithm. Recall from Section 3 that for fixed FDR $\alpha \in (0,1)$, a stable community in a network $\mathcal{G}_n$ is defined as a node set $C \subseteq N$ satisfying $U_\alpha(C, \mathcal{G}_n) = C$.

Definition 3 *We say that SCS is consistent for a sequence of WSBM random networks $\{\mathcal{G}_n\}_{n>1}$ if for any FDR level $\alpha \in (0,1)$, the probability that the true communities $C_{1,n}, \ldots, C_{K,n}$ are stable approaches 1 as $n \to \infty$.*

To assess the conditions that allow a target set C to be a stable community, we seek more general conditions under which the update $U_\alpha(\,\cdot\,, \mathcal{G})$ outputs C given any initial set B. If $U_\alpha(B, \mathcal{G}_n) = C$, all nodes $u \in C$ must have significant connectivity to B, as judged by the p-value approximation defined in 11. It is clear from that p-value expression that, for the update to return C, the test statistic $S(u, B, \mathcal{G}_n)$ must be significantly larger than $\mu(u, B|\mathbf{S}_n)$, its expected value under the continuous configuration model. Therefore, our first result hinges on asymptotic analysis of that deviation, which we denote by

$$A(u, B, \mathcal{G}_n) := S(u, B, \mathcal{G}_n) - \mu_n(u, B|\mathbf{S}_n). \tag{12}$$

The asymptotics of $A(u, B, \mathcal{G}_n)$ depend on its *population* version, in which all random quantities are replaced with their expected values under the WSBM. Let $\mathbf{s}_n$ be the expected value of $\mathbf{S}_n$ under $\mathcal{G}_n$. We define the (normalized) population version of $A(u, B, \mathcal{G}_n)$ by

$$\tilde{a}_n(u, B) := \lambda_n^{-1} \left(\mathbb{E}S(u, B, \mathcal{G}_n) - \mu_n(u, B|\bar{\mathbf{s}}_n) \right), \tag{13}$$

where λ_n is the order of the average expected degree. The value $\tilde{a}_n(u, B)$ is crucial to the primary condition of Theorem 4. Given a sequence of initial sets $\{B_n\}_{n>1}$ and target sets $\{C_n\}_{n>1}$, Theorem 4 establishes that $U_\alpha(B_n, \mathcal{G}_n) = C_n$ with probability approaching 1 if $\tilde{a}_n(u, B)$ is bounded away from zero, and is positive if and only if $u \in C_n$. The theorem requires the following two assumptions:

Assumption 5 *There exist constants $m_+ > m_- > 0$ such that, for all $n > 1$, the entries of ϕ_n, ψ_n, $\mathbf{P}$, $\mathbf{M}$, and $\boldsymbol{\pi}_n$ are all bounded in the interval $[m_-, m_+]$.*

Assumption 6 *F is independent of n and has support $(0, \eta)$ with $\eta < \infty$.*

Assumption 5 is standard in consistency analyses involving block models (e.g. Zhao et al., 2012; Bickel and Chen, 2009). We note that this assumption requires the minimum community size to scale with n. Assumption 6 allows the use of Bernstein's inequality throughout the proof, but may be relaxed if there are constraints on the moments of F allowing the use of a similar inequality. We now state Theorem 4, the proof of which is given in Appendix C.

Theorem 4 *Fix $K > 1$. For each $n > 1$, let $\mathcal{G}_n$ be a n-node random network generated by a K-community WSBM with parameters satisfying Assumptions 5 - 6. Suppose $\lambda_n / \log n \to \infty$. Let $\{B_n\}_{n>1}$, $\{C_n\}_{n>1}$ be sequences of node sets satisfying the following: there exist constants $q \in (0,1]$ and $\Delta > 0$ such that for all n sufficiently large, $|B_n|, |C_n| \geqslant qn$, and*

$$\tilde{a}_n(u, B_n) \geqslant \Delta, \;\; u \in C_n, \quad \text{and} \quad \tilde{a}_n(u, B_n) \leqslant -\Delta, \;\; u \notin C_n. \tag{14}$$

Then if the update U_α uses the p-value approximation given in Equation (11),

$$\mathbb{P}_n\big(U_\alpha(B_n, \mathcal{G}_n) = C_n\big) \to 1 \text{ as } n \to \infty.$$

To prove the consistency of SCS, we show that condition 14, when it involves the community sequence, is guaranteed by a concise condition on the model parameters. Let $\tilde{\pi}_{i,n} := \sum_{v \in C_{i,n}} \psi_n(v)$, and let $\tilde{\boldsymbol{\pi}}_n$ be the vector of $\tilde{\pi}_{i,n}$'s. The consistency theorem requires the following additional assumption, an analog to which can be found in Zhao et al. (2012) for consistency of modularity under the degree-corrected SBM:

Assumption 7 *$\tilde{\boldsymbol{\pi}}_n \equiv \tilde{\boldsymbol{\pi}}$ does not depend on n.*

Assumption 7 is made mainly for clarity. Without it, the condition in (15) of Theorem 5 (below) must hold for sufficiently large n, something which is inconsequential to the proof. Define $\mathbf{H} := \mathbf{P} \cdot \mathbf{M}$, the entry-wise product, which is symmetric because $\mathbf{P}$ and $\mathbf{M}$ are each symmetric. Note that when ϕ and ψ are proportional to 1-vectors, $\mathbb{E}(W_{uv}) = \mathbf{H}_{c(u)c(v)}$ for all $u, v \in N$. Thus, the interpretation of $\mathbf{H}$ is as the baseline inter/intra-community weight expectations after integrating out edge presence. Defining $\tilde{\boldsymbol{\Pi}} := \tilde{\boldsymbol{\pi}} \tilde{\boldsymbol{\pi}}^t$, we state the consistency theorem:

Theorem 5 *Fix $K > 0$. Let $\{\mathcal{G}_n\}_{n>1}$ be a sequence of networks generated by a K-community WSBM satisfying Assumptions 5-7. Suppose that the matrix*

$$\mathcal{M} := \mathbf{H} - \frac{\mathbf{H}\tilde{\boldsymbol{\Pi}}\mathbf{H}}{\tilde{\boldsymbol{\pi}}^t \mathbf{H} \tilde{\boldsymbol{\pi}}} \tag{15}$$

has positive diagonal entries and negative off-diagonal entries. If $\lambda_n / \log n \to \infty$, SCS is consistent for $\{\mathcal{G}_n\}_{n>1}$.

The proof of Theorem 5 is given in Appendix C. Some understanding of condition 15 can be had by considering when $K = 2$, when it reduces to the requirement $\mathbf{H}_{11}\mathbf{H}_{22} > \mathbf{H}_{12}^2$. More generally, and broadly speaking, the matrix $\mathcal{M}$ reveals whether or not appropriate signal exists in the model, with respect to the continuous configuration null. Notice that this signal need not be present in both $\mathbf{P}$ and $\mathbf{M}$. For instance, the condition can be satisfied

even if $\mathbf{H}$ is a scalar multiple of $\mathbf{M}$, that is, if $\mathbf{P}$ is proportional to the $\mathbf{1}$-matrix. This entails that SCS is consistent even when the edge structure of $\mathcal{G}_n$ is Erdős-Renyi, as long as the edge weight signal (encoded in $\mathbf{M}$) is properly assortative. Of course, the opposite also holds, namely that SCS is consistent even when assortative community signal is only present in $\mathbf{P}$. In fact, one matrix may even be *disassortative* (some or all off-diagonal entries larger than diagonals) as long as the other is sufficiently assortative.

4.2.3 Connection to weighted modularity and related work

The conditions of Theorem 4 and Theorem 5 have a deep relationship to the modularity measure, discussed in Section 2. Explicitly, let the *weighted* modularity (WM) be the modularity metric with degrees replaced by strengths, as introduced in (Newman, 2004a). For fixed $n > 1$, let $\mathbf{c}$ be any partition of N. Define $K := \max\{\mathbf{c}\}$ and $C_u := \{v : c(v) = c(u)\}$. Then the (random) WM of $\mathbf{c}$ on $\mathcal{G}_n$ can be written

$$\begin{aligned} Q^w(\mathbf{c}, \mathcal{G}_n) &:= \frac{1}{S_{n,T}} \sum_{uv \in N} \{W_{uv} - r_{uv}(\mathbf{S}_n)\}\, \mathbb{1}\{c(u) = c(v)\} \\ &= \frac{1}{S_{n,T}} \sum_{i=1}^{K} \sum_{c(u)=c(v)} W_{uv} - r_{uv}(\mathbf{S}_n) = \frac{1}{S_{n,T}} \sum_{u \in N} \sum_{v \in C_u} W_{uv} - r_{uv}(\mathbf{S}_n) \\ &= \frac{1}{S_{n,T}} \sum_{u \in N} S(u, C_u, \mathcal{G}_n) - \mu_n(u, C_u | \mathbf{S}_n) = \frac{1}{S_{n,T}} \sum_{u \in N} A(u, C_u, \mathcal{G}_n) \end{aligned}$$

Thus, the contribution of u to WM with its assignment C_u is precisely the random association from u to C_u. Writing the population WM as $\bar{q}_n^w(\mathcal{C}) := n^{-1} \sum_u \tilde{a}_n(u, C_u)$, it is easily shown that condition (15) implies q_n^w is maximized by $\mathcal{C}_n$, the true community partition.

The consistency analysis of the (binary) modularity metric under the degree-corrected SBM, provided by Zhao et al. (2012), similarly hinges on maximization of population modularity. It is unsurprising, then, that the parameter condition for their result can be (analogously) expressed as a fixed $K \times K$ matrix having positive diagonals and negative off-diagonals. In fact, if the WSBM parameter $\mathbf{M}$ is proportional to a matrix of 1s, and the parameter ψ is a scalar multiple of ϕ, condition 15 in Theorem 5 is equivalent to the parameter assumptions on modularity consistency in Zhao et al. (2012). Furthermore, their analysis also requires that $\lambda_n / \log n \to \infty$. However, both the definition of consistency and proof approach for the theorems in this section are entirely novel.

5. The Continuous Configuration Model Extraction method

In the previous section, we established an asymptotic result showing that ground-truth communities are, with high probability, fixed points of the SCS algorithm. This result demonstrates the in-principle sensibility of the algorithm. In practice, we must rely on local, heuristic algorithms for initialization and termination, as with other exploratory methods. For instance, k-means is often used to initialize the EM algorithm, and modularity can be locally maximized through agglomerative pairing (Clauset et al., 2004). We incorporate SCS in a general community detection method for weighted networks entitled Continuous Configuration Model Extraction (CCME), written in loose detail as follows:

The CCME Community Detection Method for Weighted Networks

1. Given an observed weighted network $\mathcal{G}$, obtain initial node sets $\mathcal{B}_1 \subseteq 2^N$.
2. Apply SCS to each node set in $\mathcal{B}_0$, resulting in fixed points $\mathcal{C}$.
3. Remove sets from $\mathcal{C}$ that are empty or redundant.

These steps are described in more detail below. Importantly, the method has no connection to any graph-partition criteria. It proceeds solely by the SCS algorithm, which assesses communities independently. This allows CCME to adaptively return communities that share nodes ("overlap"), and, through the multiple testing procedure, ignore nodes not significantly connected to any stable communities ("background").

5.1 Step 1: Initialization

Just as mixture-models can be initialized with heuristic methods like k-means, it is possible to initialize CCME with partition-based community detection method. However, we have observed this approach to perform somewhat poorly in practice. Instead, we initialize with a novel search procedure based on the continuous configuration model. For fixed nodes $u, v \in N$, we define

$$z_u(v) := \max\left\{ \frac{W_{uv} - f_{uv}(\mathbf{s}, \mathbf{d})}{\sqrt{\theta} f_{uv}(\mathbf{s}, \mathbf{d})},\ 0 \right\}$$

The measure $z_u(v)$ acts like a truncated z-statistic, quantifying the extremity of the weight W_{uv}. The initial node set corresponding to u is formed by sampling $d(u)$ nodes with replacement from N with probability proportional to $z_u(v)$. The intuition behind this procedure is that if u is part of a highly-connected node set C, then $z_u(v)$ for nodes $v \in C$ will be larger (on average) than for other nodes.

5.2 Step 2: Application of SCS

Recall that, given an initial set B_1, SCS proceeds (via the update U_α) along a sequence of sets $B_2, B_3, \ldots, B_t, \ldots$ until $B_t = B_{t'}$ for some $t' < t$. Since the number of possible node subsets is finite, SCS is guaranteed to terminate in one of two states:

1. A stable community C, satisfying $U_\alpha(C, \mathcal{G}) = C$.
2. A stable sequence of communities $C_1, \ldots, C_J$ satisfying

$$U_\alpha(C_1, \mathcal{G}) = U_\alpha(C_2, \mathcal{G}) = \ldots = U_\alpha(C_J, \mathcal{G}) = U_\alpha(C_1, \mathcal{G}).$$

In practice, on empirical and simulated data, case 1 is the majority. Furthermore, empirically speaking, stable sequences (case 2) usually consist only of two communities that overlap. Though case 2 does not result in a clear-cut community, a stable sequence may still be of practical interest if the constituent sets have high overlap. In Appendix D.2, we give a routine to either re-initialize or terminate SCS when it encounters a stable sequence.

5.3 Step 3: Filtering of $\mathcal{C}$

The CCME community detection method returns a final collection of communities $\mathcal{C}$, containing the results of the SCS algorithm for each initial set in $\mathcal{B}_0$. By default, we remove any empty or duplicate sets from $\mathcal{C}$. In some applications, pairs of sets in $\mathcal{C}$ will have high Jaccard similarity. In Appendix D.3, we detail a method of pruning these near-duplicates from $\mathcal{C}$. Additionally, in Appendix D.3, we describe routines to suppress the application of SCS to initial sets that are "weakly" intra-connected, or with high overlap to already-extracted communities. These routines greatly reduce the runtime of CCME, and, on some simulated networks, improve accuracy.

5.4 Remarks

1. The parameter α, used in the set update operation U_α, must be specified by the user of CCME. Having a natural interpretation as the false-discovery rate for each update, α was set to 0.05 for all simulations and real data analyses introduced in this paper. We found that $\alpha = 0.05$ was a universally effective default setting, and that CCME's results change negligibly for other values of α within a reasonable window.

2. Multiple testing corrections computed within CCME are valid only for the input network. If other edge-weighted networks with the same node set are available, say, from an alternate data source or different set of node covariates, then it is necessary to apply a multiple-testing correction across communities discovered by the separate runs of CCME. The community-wise z-statistic introduced in Appendix D.3 will be useful to this end.

3. CCME can be extended to directed networks, as described in Appendix E.

6. Simulations

This section contains a performance analysis of CCME and existing methods on a benchmarking simulation framework. Simulated networks were generated from the Weighted Stochastic Block Model (see Section 4.2.1), with slight modifications to include overlapping communities and background nodes, when necessary. The performance measures, competing methods, simulation settings, and results are described below.

6.1 Performance measures and competing methods

We used three community detection method performance metrics:

1. **Overlapping Normalized Mutual Information (oNMI):** Introduced by Lancichinetti et al. (2009), oNMI is an information-based measure between 0 and 1 that approaches 1 as two covers of the same node set become similar and equals 1 when they are the same. From a method's results, we calculate oNMI with respect to the true communities *only* for the nodes the method placed into communities.

2. **Community nodes in background (%C.I.B.):** The percentage of true community nodes incorrectly assigned to background.

3. **Background nodes in communities (%B.I.C.):** The percentage of true background nodes (if present) incorrectly placed into communities.

In addition to CCME, three other weighted-network methods capable of identifying overlapping nodes were applied to our simulation ensemble. One of these was OSLOM (Lancichinetti et al., 2011), described in Section 1. Another was SLPAw, a weighted-network version of an overlapping label propagation algorithm (Xie et al., 2011). Finally, we applied the weighted Stochastic Block Model fitting procedure introduced by Peixoto (2017), which we will call "WeightedGT" after it's implementation's package ("graph-tool"). Also included were four commonly used score-based methods implemented in the `R` package `igraph` (Csárdi and Nepusz, 2006): Fast-Greedy, which performs approximate modularity optimization via a hierarchical agglomeration (Clauset et al., 2004); Louvain, an approximate modularity optimizer that proceeds through node membership swaps (Blondel et al., 2008); Walktrap, an agglomerative algorithm that locally maximizes a score based on random walk theory (Pons and Latapy, 2006); Infomap, an information-flow mapping algorithm that uses random walk transition probabilities (Rosvall and Bergstrom, 2008).

Remark. Being extraction methods, only CCME and OSLOM naturally specify background nodes, via testing. As such, we will often make direct comparative comments between OSLOM and CCME with respect to background node handling. For other methods, we take as background any nodes in singleton communities. However, these methods almost never returned singleton communities, even when the simulation had weak or non-existent signal.

6.2 Simulation settings and results

We now give an overview of the simulation procedure for the benchmarking framework. A complete account is given in Appendix F. We first describe "default" parameter settings of the WSBM; in the simulation settings below, individual parameters are toggled around their default values, to reveal the dependence of the methods to those parameters. At each unique parameter setting, 20 random networks were simulated. The points in each plot from Figure 1 show the average performance measure of the methods over the 20 repetitions.

The default WSBM setting has the number of nodes at $n = 5,000$. The community memberships were set by obtaining community sizes from a power law, then assigning nodes uniformly at random. This process produced approximately 3 to 7 communities per network. Many of these simulation features (as well as those described in what follows) mirror those used in existing frameworks (e.g. Lancichinetti et al., 2011); full details are provided in Appendix F. Recall the parameters $\mathbf{P}$ and $\mathbf{M}$, which induce baseline intra- and inter-community edge and weight signal. In the default setting, these matrices have off-diagonals equal to 1 and diagonals equal to constants $s_e = 3$ and $s_w = 3$ (respectively). In some simulation settings, overlapping and background nodes are added (as described later in this section), but the default setting includes neither overlap nor background.

Common parameter settings. For all simulated networks (regardless of the setting), the node-wise edge parameters ϕ were drawn from a power law to induce degree heterogeneity. The parameter ϕ is scaled so that the expected average degree of each network was equal to $\sqrt{n}$, which induces sparsity in the network. The parameter ψ is set by the for-

mula $\psi = \phi^{1.5}$ to ensure a non-trivial relationship between expected degrees and expected strengths (see Appendix F).

6.2.1 Networks with varying signal levels

The first simulation setting tested the methods' dependence on s_e and s_w. These values were moved along an even grid on the range $[1, 3]$. Plots A-1 and B-1 in Figure 1 show the performance measure results when s_w is fixed at 3, plots A-2 and B-2 show results when $s_e = 3$, and plots A-3 and B-3 show results when s_e and s_w are moved along $[1, 3]$ together. Many methods had large oNMI scores in this simulation setting. We transformed the oNMI scores using the function

$$\text{t-oNMI}_a(x) := (\tfrac{1}{1-x+a} - \tfrac{1}{1+a})/(\tfrac{1}{a} - \tfrac{1}{1+a})$$

with $a = 0.05$. This is a monotonic, one-to-one transformation from $[0, 1]$ to itself, which stretches the region close to 1, allowing a clearer comparison between the methods' performances. CCME consistently out-performed all competing methods, especially when either the edge or weight signal was completely absent. (The t-oNMI transformation masks the trends in lower-performing methods, so we show these plots with un-transformed oNMI in Appendix F.3.)

The plots in row B show that when either s_e or s_w were near 1, OSLOM and CCME assigned many background nodes. This is consistent with these methods' unique abilities to leave nodes unassigned when they are not significantly connected to communities. That said, %C.I.B. can be seen as a measure of sensitivity, since ideally no nodes would be assigned to background when any signal is present. In this regard, CCME outperformed OSLOM across the range of model parameters.

6.2.2 Networks with overlapping communities

The second setting involved networks with overlapping nodes. To add overlapping nodes to the default network, two parameters were introduced: o_n, the number of overlapping nodes, and o_m, the number of memberships for each overlapping node. The particular overlapping nodes and community memberships were chosen uniformly-at-random. Plots C-1 and C-2 show performance results from the setting with o_n moving away from 0 and $o_m = 2$. Plot C-3 shows results from the setting with $o_n = 500$ and $o_m \in \{1, \ldots, 4\}$. We find that CCME consistently outperforms all methods in terms of accuracy (oNMI), and outperforms OSLOM in terms of sensitivity (%C.I.B.).

6.2.3 Networks with overlapping communities and background nodes

The final simulation setting involved networks with both overlap and background nodes. The number of background nodes was fixed at 1,000, and number of community nodes varied from $n = 500$ to $n = 5,000$. For each network, $o_n = n/4$ nodes were randomly chosen to overlap $o_m = 2$ communities (also chosen at random). Background nodes were created by first simulating the n-node community sub-network, and then generating the 1,000-node background sub-network according to the continuous configuration model, using empirical degrees and strengths from the community sub-network. The complete details of this procedure are given in Appendix F.

The results of this simulation setting are shown in row D from Figure 1. From plot D-1, we see that OSLOM and CCME had the highest oNMI scores, favoring OSLOM when the number of community nodes decreased. Because this simulation setting involved background nodes, the %B.I.C. metric is relevant, and can be taken as a measure of specificity: ideally, nodes from the background sub-network should be excluded from communities. From plot D-2, we see that methods incapable of assigning background had %B.I.C. equal to 1. We found that CCME correctly ignored background nodes as the network size increased, whereas OSLOM became increasingly *anti*-conservative for larger networks. Furthermore, CCME again had lower %C.I.B. than OSLOM.

7. Applications

In this section, we discuss applications of CCME, OSLOM, and SLPAw (the methods capable of returning overlapping communities) to two real data sets.

7.1 U.S. airport network data

The first application involves commercial airline flight data, obtained from the Bureau of Transportation Statistics (www.transtats.bts.gov). For each month from January to July of 2015, we created a weighted network with U.S. airports as nodes, edges connecting airports that exchanged flights, and edges weighted by aggregate passenger count. We also constructed a year-aggregated network, formed simply by taking the union of the month-wise edge sets, and adding the month-wise weights. In Figure 2, we display the methods' results when applied to the June and year-aggregated data sets from 2015. Each discovered community (within-method) has a unique color and shape. Each overlapping node is plotted multiple times, one for each community in which it was placed. To allow for a clearer visualization of communities, background nodes are not shown.

Overall, the CCME results, in contrast to results from OSLOM and SLPAw, suggest that many airports in the U.S. airport system may not participate in meaningful community behavior. The fact that CCME performs multiple testing against an explicit null model gives this result some validity. Furthermore, airports in significant communities tend to be located near large hubs or in geographically isolated areas. We also see that, with the monthly data, OSLOM and CCME tended to find communities consistent with geography, whereas SLPAw placed most of the network into one community. With the year-aggregated data, OSLOM also agglomerated most airports, whereas CCME continued to respect the geography. Since the aggregated data is much more edge-dense, this suggests the performance of OSLOM and SLPA may suffer on weighted graphs with high or homogeneous edge-density, whereas CCME is able to detect proper community structure from the weights alone. This aligns with the simulation results described in Section 6.2.1.

7.2 ENRON email network

An email corpus from the company ENRON was made available in 2009. The unweighted network formed by linking communicating email addresses is well-studied; see www.cs.cmu.edu/~./enron for references and Leskovec et al. (2010) for the data. For the purposes of this paper, we derived a *weighted* network from the original corpus, using

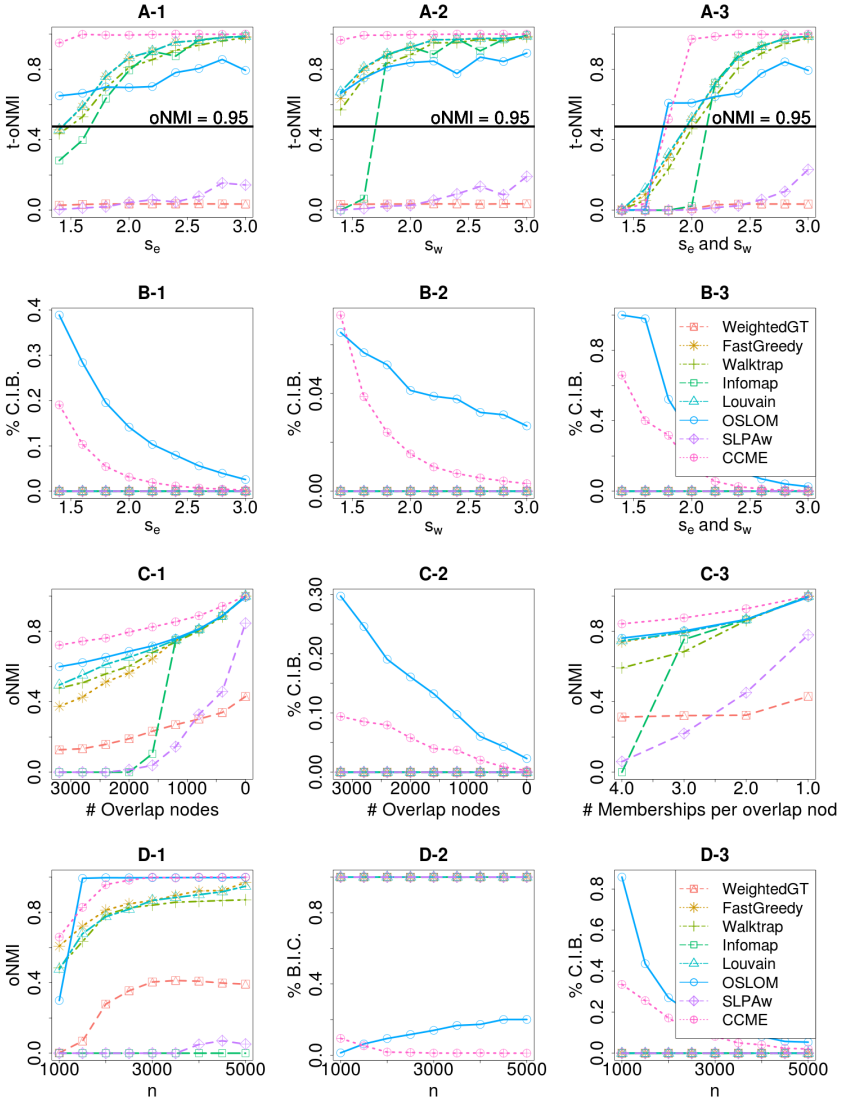


Figure 1: Simulation results described in Sections 6.2.1-6.2.3. Legends refer to all plots.

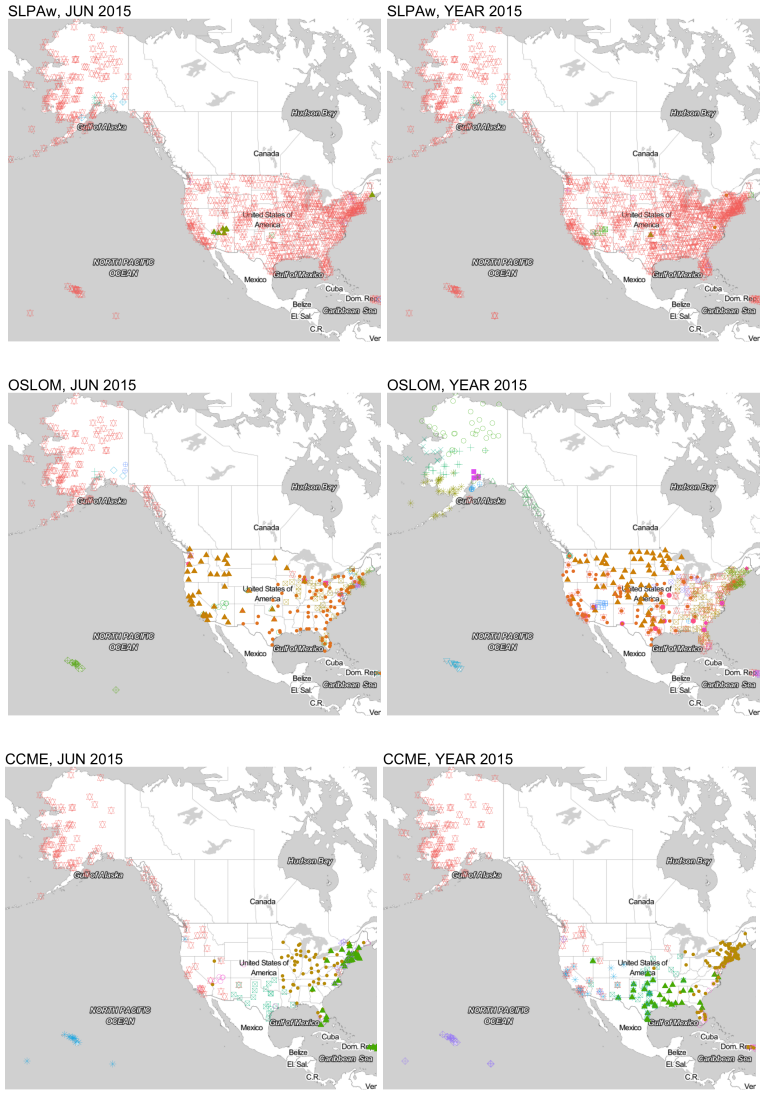


Figure 2: SLPAw, OSLOM, and CCME results from June 2015 and 2015-year-aggregated U.S. airport networks. Maps created with `ggmap` (Kahle and Wickham, 2013).

message count between addresses as edge weights. Though the corpus was formed from email folders of 150 ENRON executives, we made the network from addresses found in *any* message. This full network has 80,702 unique addresses, 47,481 of which were external to the company. Thus, the network likely has many spam or irrelevant senders, i.e. "true" background nodes. We applied CCME, OSLOM, and SLPAw to the network to see which methods best focused on company-specific areas of the data.

Tables 1 and 2 give basic summaries of the results, which show noticeable differences between the outputs of the methods. CCME placed far fewer into nodes into communities, but detected larger communities with more overlapping nodes. Notably, CCME had the highest percentage of ENRON addresses among nodes it placed into communities (see Table 3). These results suggest that CCME was more sensitive to critical relationships in the network.

Table 1: Metrics from methods' results on ENRON network: number of communities, min. comm. size, median comm. size, max. comm. size, # nodes in communities.

	Num.Comms	Min.size	Med.size	Max.size	Num.Nodes
CCME	185	2	687	5416	14552
OSLOM	405	2	19	770	17635
SLPAw	2138	2	4	4793	79316

Table 2: Metrics from methods' results on ENRON network: number of overlapping nodes, minimum # of memberships, median # of mem'ships, max. # of mem'ships.

	Num.OL.Nodes	Min.mships	Med.mships	Max.mships
CCME	8104	2	9	78
OSLOM	462	2	2	8
SLPAw	3860	2	2	4

Table 3: Top domains associated with community nodes from each method, by proportion.

CCME.Domains	Prop.
enron.com	0.784
aol.com	0.008
cpuc.ca.gov	0.006
pge.com	0.004
socalgas.com	0.003
dynegy.com	0.003

OSLOM.Domains	Prop.
enron.com	0.529
aol.com	0.029
haas.berkeley.edu	0.016
hotmail.com	0.015
yahoo.com	0.009
jmbm.com	0.005

SLPAw.Domains	Prop.
enron.com	0.423
aol.com	0.039
hotmail.com	0.023
yahoo.com	0.016
haas.berkeley.edu	0.007
msn.com	0.006

8. Discussion

In this paper, we introduced the continuous configuration model, which is, to the best of our knowledge, the first null model for community detection on weighted networks. The explicit generative form of the null model allowed the specification of CCME, a community

extraction method based on sequential significance testing. We showed that a standardized statistic for the tests is asymptotically normal, a result which enables an analytic approximation to p-values used in the method. We also proved asymptotic consistency under a weighted stochastic block model for the core algorithm of the method.

On simulated networks the proposed method CCME is competitive with commonly-used community detection methods. CCME was the dominant method for simulated networks with large numbers of overlapping nodes. Furthermore, on networks with background nodes, CCME was the only method to correctly label true background nodes while maintaining high detection power and accuracy for nodes belonging to communities. On real data, CCME gave results that were both interpretable and revelatory with respect to the natural system under study.

We expect that the continuous configuration model will have applications outside the setting of this paper, just as the binary configuration model has been studied in diverse contexts. One may investigate the distributional properties of many different graph-based statistics under the model, as a means of assessing statistical significance in practice. For instance, an appropriate theoretical analysis could yield an approach to the assessment of statistical significance of weighted modularity. The proof approach used for Theorem 2 may serve as a basis for this endeavor. Another benefit of an explicit null for weighted networks is the potential for simulation. Using the continuous configuration model, and parts of the framework presented in this paper, one can generate weighted networks having true background nodes with arbitrary expected degree and strength distributions.

8.1 Acknowledgements and Remarks

The authors wish to thank three anonymous referees for their thoughtful suggestions, and Dr. Peter J. Mucha for helpful discussions. The `R` code for the CCME method is available in the github repository 'jpalowitch/CCME'. The code for reproducing the analyses in Sections 6 and 7 is available at the github repository 'jpalowitch/CCME_analyses'. The work of JP and ABN was supported in part by NIH grant MH101819-01. The work of SB and ABN was supported in part by NSF grant DMS-1310002. SB was also supported in part by NSF grants DMS-1105581, DMS-161307, and DMS-160683, SES grant 1357622, and ARO grant W911NF-17-1-0010. ABN was also supported in part by NSF DMS-1613072 and NIH HG009125-01.

Appendix A. Proof of Proposition 1

Equation 8 follows immediately from the observation in equation 3 and the definition of $r_{uv}(\mathbf{s})$. Next, note that

$$\mathbb{E}\left(W_{uv}|A_{uv}\right) = f_{uv}(\mathbf{d},\mathbf{s})A_{uv}, \text{ and } \mathrm{Var}(W_{uv}|A_{uv}) = \kappa f_{uv}(\mathbf{d},\mathbf{s})^2 A_{uv}.$$

Thus, using the law of total variance,

$$\begin{aligned}\mathrm{Var}(W_{uv}) &= f_{uv}(\mathbf{d},\mathbf{s})^2\mathrm{Var}(A_{uv}) + \kappa f_{uv}(\mathbf{d},\mathbf{s})^2\mathbb{E}(A_{uv}) \\ &= f_{uv}(\mathbf{d},\mathbf{s})^2\tilde{r}_{uv}(\mathbf{d})(1-\tilde{r}_{uv}(\mathbf{d})) + \kappa f_{uv}(\mathbf{d},\mathbf{s})^2\tilde{r}_{uv}(\mathbf{d}) \\ &= r_{uv}(\mathbf{s})f_{uv}(\mathbf{d},\mathbf{s})\left(1-\tilde{r}_{uv}(\mathbf{d}) + \kappa\right)\end{aligned}$$

Summing over $v \in B$ gives equation 9. ■

Appendix B. Proof of Theorem 2 and supporting lemmas.

Here we give the proof of Theorem 2 in Section 4.1. We start with supporting lemmas. Recall the definition of the average degree parameter λ_n, the normalized r^{th}-moment $L_{n,r}$, and other associated definitions from Section 4.1. For the purposes of the results below, we define the following generalization of $L_{n,r}$, given a node set $B_n \subseteq N$ with $b_n := |B_n|$:

$$L_{n,r}(B_n) := b_n^{-1} \sum_{u \in B_n} \{d_n(u)/\lambda_n\}^r$$

Note that $L_{n,r}(N) = L_{n,r}$. Recall that in the setting of Theorem 2, the node set B_n is chosen uniformly from the node set N. The first result involves a *deterministic* sequence $\{B_n\}_{n \geqslant 1}$:

Lemma 6 *For each $n > 1$, let $\mathcal{G}_n$ be generated by the continuous configuration model with parameters $\theta_n = (\mathbf{d}_n, \mathbf{s}_n, \kappa_n)$ and common weight distribution F. Fix a node sequence $\{u_n\}_{n>1}$ with $u_n \in N$ and a positive integer sequence $\{b_n\}_{n>1}$ with $b_n \leqslant n$. Suppose the parameter sequence $\{d_n(u_n)\}_{n \geqslant 1}$ satisfies*

$$\frac{d_n(u_n) b_n}{n} \to \infty \text{ as } n \to \infty$$

Fix $\varepsilon > 0$ as in Assumption 2, and choose $\delta \in (0,1)$ such that $2\beta\delta < \varepsilon$. Fix a sequence of sets $\{B_n\}_{n>1}$ with $|B_n| = b_n$ for all n, and suppose that for $r = 2\beta + 1$ and $r = \beta(2+\delta)+1$, the sequence $\{L_{n,r}(B_n)\}_{n>1}$ is bounded away from zero and infinity. Then

$$\frac{S(u_n, B_n, \mathcal{G}_n) - \mu_n(u_n, B_n|\theta_n)}{\sigma_n(u_n, B_n|\theta_n)} \Rightarrow \mathcal{N}(0,1) \quad as \quad n \to \infty$$

Proof In what follows, the functions r_{uv} and $\tilde{r}_{uv}$ from Section 1.2 will be used extensively. Note that for any nodes u, v, $\mathbb{E}W_{uv} = r_{uv}(\mathbf{s})$. Thus by the classical Lyapunov central limit theorem it suffices to show that

$$\frac{\sum_{v \in B_n} \mathbb{E}|W_{u_n,v} - r_{u_n v}(\mathbf{s}_n)|^{2+\delta}}{\left(\sqrt{\sum_{v \in B_n} \mathbb{E}\left\{(W_{u_n,v} - r_{u_n v}(\mathbf{s}_n))^2\right\}}\right)^{2+\delta}} \to 0 \tag{16}$$

as n tends to infinity. The following derivations hold for any fixed $n > 1$, so we suppress dependence on n from u_n, and B_n, and similar expressions. For the numerator of (16), we have

$$\begin{aligned}
\mathbb{E}|W_{u,v} - r_{uv}(\mathbf{s})|^{2+\delta} &= \left(\frac{r_{uv}(\mathbf{s})}{\tilde{r}_{uv}(\mathbf{d})}\right)^{2+\delta} \mathbb{E}\left(|\xi_{uv} A_{uv} - \tilde{r}_{uv}(\mathbf{d})|^{2+\delta}\right) \\
&= f_{uv}(\mathbf{d}, \mathbf{s})^{2+\delta} \cdot \mathbb{E}\left(|\xi_{uv} A_{uv} - \tilde{r}_{uv}(\mathbf{d})|^{2+\delta}\right), \qquad (17)
\end{aligned}$$

by definition of the model in Section 2.1. Moreover, by the law of total variance,

$$\begin{aligned}
\mathbb{E}(|\xi_{uv} A_{uv} - \tilde{r}_{uv}(\mathbf{d})|^{2+\delta}) &= (1 - \tilde{r}_{uv}(\mathbf{d}))\tilde{r}_{uv}(\mathbf{d})^{2+\delta} + \tilde{r}_{uv}(\mathbf{d})\mathbb{E}|\xi_{uv} - \tilde{r}_{uv}(\mathbf{d})|^{2+\delta} \\
&= \left\{(1 - \tilde{r}_{uv}(\mathbf{d}))\tilde{r}_{uv}(\mathbf{d})^{1+\delta} + \mathbb{E}|\xi_{uv} - \tilde{r}_{uv}(\mathbf{d})|^{2+\delta}\right\} \cdot \tilde{r}_{uv}(\mathbf{d}) \\
&\leqslant C \cdot \tilde{r}_{uv}(\mathbf{d}) \qquad (18)
\end{aligned}$$

for some positive constant C, by Assumption 4. Next, we note that by Assumption 1, there exist positive constants $a < c$ such that for all $v \in N$,

$$a \cdot d_n(v)^\beta \leqslant \frac{s_n(v)}{d_n(v)} \leqslant c \cdot d_n(v)^\beta,$$

for n sufficiently large. Thus, if $r_{uv}(\mathbf{d}) \leqslant 1$, $\tilde{r}_{uv}(\mathbf{d}) = r_{uv}(\mathbf{d})$, and

$$f_{uv}(\mathbf{d}, \mathbf{s}) = \frac{r_{uv}(\mathbf{s})}{\tilde{r}_{uv}(\mathbf{d})} = \left(\frac{d_T}{s_T}\right) \frac{s(u)s(v)}{d(u)d(v)} \leqslant c \cdot \left(\frac{d_T}{s_T}\right) \{d(u)d(v)\}^\beta. \tag{19}$$

If $r_{uv}(\mathbf{d}) > 1$, $\tilde{r}_{uv}(\mathbf{d}) = 1$, and by Assumption 3 there exists c' such that

$$\begin{aligned}
f_{uv}(\mathbf{d}, \mathbf{s}) = \frac{s(u)s(v)}{s_T} &\leqslant c \cdot \left(\frac{d(u)d(v)}{s_T}\right) \{d(u)d(v)\}^\beta \\
&= c \cdot \left(\frac{d_T}{s_T}\right) r_{uv}(\mathbf{d}) \{d(u)d(v)\}^\beta \leqslant c' \cdot \left(\frac{d_T}{s_T}\right) \{d(u)d(v)\}^\beta. \qquad (20)
\end{aligned}$$

Therefore, combining (18)-(20) with (17), there exists $C > 0$ such that

$$\begin{aligned}
\mathbb{E}|W_{u,v} - r_{uv}(\mathbf{s})|^{2+\delta} &\leqslant C \left(\frac{d_T}{s_T}\right)^{2+\delta} \cdot \{d(u)d(v)\}^{\beta(2+\delta)} \tilde{r}_{uv}(\mathbf{d}) \\
&= C \left(\frac{d_T}{s_T}\right)^{2+\delta} \cdot \{d(u)d(v)\}^{\beta(2+\delta)} \frac{d(u)d(v)}{d_T} \\
&\leqslant C \cdot d_T^{1+\delta} s_T^{-(2+\delta)} \cdot \{d(u)d(v)\}^{\beta(2+\delta)+1} \qquad (21)
\end{aligned}$$

A similar analysis of the summands in the denominator of (16) gives

$$\mathbb{E}\left\{(W_{u,v} - r_{uv}(\mathbf{s}))^2\right\} \geqslant C' \cdot d_T s_T^{-2} \cdot \{d(u)d(v)\}^{2\beta+1} \tag{22}$$

for appropriately chosen C'. Let $b = |B|$. Combining (21) and (22), with some algebra, we find that the left side of (16) is (up to a constant) less than

$$\begin{aligned}
&\left(\frac{d(u)}{d_T}\right)^{-\delta/2} \cdot \frac{\sum_{v \in B} d(v)^{\beta(2+\delta)+1}}{\left(\sum_{v \in B} d(v)^{2\beta+1}\right)^{1+\delta/2}} \\
&= \left(\frac{d(u)}{d_T} b\lambda\right)^{-\delta/2} \cdot \frac{b^{-1} \sum_{v \in B} (d(u)/\lambda)^{\beta(2+\delta)+1}}{\left\{b^{-1} \sum_{v \in B} (d(u)/\lambda)^{2\beta+1}\right\}^{1+\delta/2}} \\
&= \left(\frac{d(u)}{d_T} b\lambda\right)^{-\delta/2} \cdot \frac{L_{n,\beta(2+\delta)+1}(B)}{(L_{n,2\beta+1}(B))^{1+\delta/2}} = O\left\{\left(\frac{d(u)}{d_T} b\lambda\right)^{-\delta/2}\right\} \qquad (23)
\end{aligned}$$

where the final term follows from our assumptions on $L_{n,\beta(2+\delta)+1}(B_n)$ and $L_{n,2\beta+1}(B_n)$. By definition, $d_{n,T} = n\lambda_n$, so the final expression above is $O\left\{(d_n(u_n)b_n/n)^{-\delta/2}\right\} = o(1)$ by assumption. Thus (16) holds and the result follows. ■

We now proceed with the proof of Theorem 2. Proposition 6 yields the CLT for $S(u_n, B_n, \mathcal{G}_n)$ for a deterministic sequence of vertex sets $\{B_n\}_{n\geqslant 1}$ satisfying regularity properties. The remainder of the argument shows that if B_n is selected uniformly at random then, under the assumptions of Theorem 2, these regularity properties are satisfied with high probability. We begin with a few preliminary definitions and results.

Definition 7 *A sequence of random variables $\{X_n\}_{n\geqslant 1}$ is said to be* asymptotically uniformly integrable *if*

$$\lim_{M\to\infty} \limsup_{n\to\infty} \mathbb{E}\left\{|X_n|\mathbb{1}(|X_n| > M)\right\} = 0$$

Theorem 8 *Let $f : \mathbb{R}^k \mapsto \mathbb{R}^k$ be measurable and continuous at every point in a set C. Suppose $X_n \xrightarrow{w} X$ where X takes its values in an interval C. Then $\mathbb{E}f(X_n) \to \mathbb{E}f(X)$ if and only if the sequence of random variables $f(X_n)$ is asymptotically uniformly integrable.*

Proof See Asymptotic Statistics (Van der Vaart 2000), page 17. ■

We now give a technical lemma (needed for a subsequent result) which uses Theorem 8:

Lemma 9 *Let $X_1, X_2, \ldots$ be non-negative random variables and let $s, \varepsilon > 0$. If the sequences $\{\mathbb{E}X_n^s\}_{n\geqslant 1}$ and $\{\mathbb{E}X_n^{s+\varepsilon}\}_{n\geqslant 1}$ are bounded away from zero and infinity, then $\{\mathbb{E}X_n^r\}_{n\geqslant 1}$ is bounded away from zero and infinity for every $r \in (0, s+\varepsilon)$.*

Proof Suppose by way of contradiction that there exists $t \in (0, s+\varepsilon)$ such that $\liminf_n \mathbb{E}X_n^t = 0$. Then $\lim_k \mathbb{E}X_{n_k}^t = 0$ along a subsequence $\{n_k\}$. As the random variables $X_{n_k}^t$ are non-negative, $X_{n_k}^t \xrightarrow{d} 0$, and it follows from the continuous mapping theorem that $X_{n_k} \xrightarrow{w} 0$. As $M^{\varepsilon/s} X_n^s \mathbb{1}(X_n^s > M) \leqslant X_n^{s+\varepsilon}$, we find that

$$\lim_{M\to\infty} \limsup_{k\to\infty} \mathbb{E}\{X_{n_k}^s \mathbb{1}(X_{n_k}^s > M)\} \;\leqslant\; \lim_{M\to\infty} M^{-\varepsilon/s} \limsup_{k\to\infty} \mathbb{E}(X_{n_k}^{s+\varepsilon}) = 0$$

as $\mathbb{E}(X_n^{s+\varepsilon})$ is bounded by assumption. It then follows from Theorem 8 and the fact that $X_{n_k}^s \xrightarrow{w} 0$ that $\mathbb{E}X_{n_k}^s \to 0$ as $k \to \infty$, violating our assumption that $\mathbb{E}X_n^s$ is bounded away from zero. We conclude that $\mathbb{E}X_n^r$ is bounded away from zero for $r \in (0, s+\varepsilon)$. On the other hand, if $r \in (0, s+\varepsilon)$ then for each $n \geqslant 1$

$$\mathbb{E}\{X_n^r \mathbb{1}(X_n > 1)\} \;\leqslant\; \mathbb{E}\{X_n^{s+\varepsilon} \mathbb{1}(X_n > 1)\} \;\leqslant\; \sup_n \mathbb{E}\{X_n^{s+\varepsilon}\}$$

As the last term is finite by assumption and $\mathbb{E}\{X_n^r \mathbb{1}(X_n \leqslant 1)\}$ is at most one, it follows that $\mathbb{E}(X_n^r)$ is bounded. ■

Lemma 10 *Suppose a degree parameter sequence $\{\mathbf{d}_n\}_{n\geqslant 1}$ satisfies Assumption 2 from Section 4.1. For each n, let B_n be a randomly chosen subset of N of size b_n, where $b_n \to \infty$. Fix $\varepsilon > 0$ as in Assumption 2, and choose δ so that $2\beta\delta < \varepsilon$. Then for every $r \in (0,\ \beta(2+\delta)+1]$, there exists an interval $I_r = (a_r, b_r)$ with $0 < a_r < b_r < \infty$ such that $\mathbb{P}\{L_{n,r}(B_n) \in I_r\} \to 1$ as $n \to \infty$.*

Remark: Note that the function $L_{n,r}(\cdot)$ is non-random. The probability appearing in the conclusion of Lemma 10 depends only on the random choice of the vertex set B_n.

Proof Let D_n and D'_n be drawn uniformly-at-random from $\mathbf{d}_n$ without replacement, and fix $r \in (0,\ \beta(2+\delta)]$. A routine calculation gives

$$\mathrm{Var}\{L_{n,r}(B_n)\} = b_n^{-1}\lambda_n^{-2r}\left[\mathrm{Var}\{D_n^r\} + \{b_n - 1\}\mathrm{Cov}\{D_n^r, (D'_n)^r\}\right].$$

Note that $\mathbb{E}(D_n^r) = \lambda_n^r L_{n,r}$ and $\mathbb{E}(D_n^{2r}) = \lambda_n^{2r} L_{n,2r}$, so $\mathrm{Var}(D_n^r) = \lambda_n^{2r}(L_{n,2r} - L_{n,r})$. Furthermore, a simple calculation shows that $\mathrm{Cov}\{D_n^r, (D'_n)^r\}$ is negative for every r, and therefore $\mathrm{Var}\{L_{n,r}(B_n)\} \leqslant b_n^{-1}(L_{n,2r} - L_{n,r})$. Our choice of δ ensures that $2r < 4\beta + 2 + \varepsilon$, and it then follows from Lemma 9 and Assumption 2 that $L_{n,2r}$ and $L_{n,r}$ are bounded. Thus $\mathrm{Var}\{L_{n,r}(B_n)\} = O(b_n^{-1})$. Define $\Delta := \liminf_n\ L_{n,r}/2$, which is positive by Assumption 2, and let

$$I_r := \left(\liminf_{n\to\infty} L_{n,r} - \Delta,\ \limsup_{n\to\infty} L_{n,r} + \Delta\right) \tag{24}$$

As $\mathbb{E}\{L_{n,r}(B_n)\} = L_{n,r}$, an application of Chebyshev's inequality yields the bound

$$\begin{aligned}\mathbb{P}\{L_{n,r}(B_n) \notin I_r\} \;&\leqslant\; \mathbb{P}\{|L_{n,r}(B_n) - \mathbb{E}[L_{n,r}(B_n)]| > \Delta/2\} \\ &\leqslant\; \frac{4\mathrm{Var}\{L_{n,r}(B_n)\}}{\Delta^2} \;=\; O(b_n^{-1}).\end{aligned}$$

As b_n tends to infinity with n, the result follows. ■

B.1 Completing the proof of Theorem 2.

Let ε and δ be as in Proposition 6 and Lemma 10. Note that since $d_n(u_n) \leqslant n$ for all n, our assumption that $b_n d_n(u_n)/n \to \infty$ implies $|B_n| = b_n \to \infty$. Hence by lemma 10, we have that for both $r = \beta(2+\delta)+1$ and $r = 2\beta+1$, there exists a positive, finite interval I_r such that $\mathbb{P}\{L_{n,r}(B_n) \in I_r\} \to 1$ as $n \to \infty$. Thus given any subsequence $\{n_k\}_{k\geqslant 1}$ we can find a further subsequence $\{n'_k\}_{k\geqslant 1}$ such that $L_{n'_k,r}(B_{n'_k}) \in I_r$ almost surely as $k \to \infty$, which means this sequence is bounded away from zero and infinity in k. Now using Proposition 6, for almost every ω we have

$$\frac{S_{n'_k}(u_{n'_k}, B_{n'_k}, \mathcal{G}_{n'_k}) - \mu_{n'_k}(u_{n'_k}, B_{n'_k}|\theta_{n'_k})}{\sigma_{n'_k}(u_{n'_k}, B_{n'_k}|\theta_{n'_k})} \Rightarrow \mathcal{N}(0,1) \;\text{ as }\; k \to \infty$$

Applying the subsequence principle completes the proof. ■

Appendix C. Proof of Theorems 4-5 and supporting lemmas.

Throughout this section, notation and conventions from Section 4.2.1 will be used, though we suppress dependence on n for convenience. Further recall functions r and f from Section 1.2. The following additional notation will be used throughout this section:

- Define $\phi_T := \sum_{v\in N}\phi(v)$ and $\psi_T := \sum_{v\in N}\psi(v)$. For each $K \geqslant j \geqslant 1$, define $\tilde{\pi}_j^0 := \sum_{v\in\mathcal{C}_j}\phi(v)/\phi_T$ and $\tilde{\pi}_j := \sum_{v\in\mathcal{C}_j}\psi(v)/\psi_T$. Let $\tilde{\boldsymbol{\pi}}^0$ and $\tilde{\boldsymbol{\pi}}$ be the associated vectors.
- Let $\langle\cdot,\cdot\rangle$ denote the vector dot-product. For a general symmetric matrix $\mathbf{A}$, let $\mathbf{A}_{ij}$ be the i,j-th entry, and $\mathbf{A}_i$ the i-th column. Define $\mathbf{H} := \mathbf{P}\cdot\mathbf{M}$, the entry-wise product.
- Let $D(u), S(u)$ be the random degree, strength of node $u \in N$, let $\bar{d}(u)$, $\bar{s}(u)$ be the corresponding expectations, and let $\mathbf{D}, \mathbf{S}, \bar{\mathbf{d}}, \bar{\mathbf{s}}$ be the associated n-vectors. Define $\bar{s}_T := \sum_{v\in N}\bar{s}(v)$ and $\bar{d}_T := \sum_{v\in N}\bar{d}(v)$.

We now define a *empirical* population version of the variance estimate:

Definition 11 *Fix $n > 1$ and let A and W be the edge and weight matrices from $\mathcal{G}_n$, the n-th random weighted network from the sequence in the setting of Theorem 4. Let $\mathbf{x}, \mathbf{y}$ be arbitrary n-vectors with positive entries. For nodes $u, v \in N$, define*

$$V_{uv}(\mathbf{x},\mathbf{y}) := (W_{uv} - f_{uv}(\mathbf{x},\mathbf{y}))^2, \qquad v_{uv}(\mathbf{x},\mathbf{y}) := \mathbb{E}\left\{V_{uv}(\mathbf{x},\mathbf{y})\middle|A_{uv}=1\right\}.$$

Define the empirical *population variance estimator as follows:*

$$\kappa_*(\mathbf{x},\mathbf{y}) := \frac{\sum_{u,v:A_{uv}=1} v_{uv}(\mathbf{x},\mathbf{y})}{\sum_{u,v:A_{uv}=1} f_{uv}(\mathbf{x},\mathbf{y})^2}$$

The estimator $\kappa_*(\mathbf{x},\mathbf{y})$ is called "empirical" because it depends on the random edge set E. Despite this, it has a deterministic bound, a fact which is part of Lemma 12. Throughout the remaining results, denote $\Theta := (\mathbf{D}, \mathbf{S}, \hat{\kappa}(\mathbf{D},\mathbf{S}))$ and $\theta_* := (\bar{\mathbf{d}}, \bar{\mathbf{s}}, \kappa_*(\bar{\mathbf{d}}, \bar{\mathbf{s}}))$, where the estimator $\hat{\kappa}$ is the estimator from Section 2.2.

Recall the definition of the asymptotic order of the average degree $\lambda_n := n\rho_n$, from Section 4.2.2 in the main text. With this and the conventions above, Lemma 12 establishes basic facts about the WSBM:

Lemma 12 *Fix $n > 1$, and let $\mathcal{G}_n$ be a random network generated by a WSBM. For all nodes $u, v \in N$, under Assumptions 5 and 6,*

(1) $\bar{d}(u) = \lambda_n\phi(u)\langle\tilde{\boldsymbol{\pi}}^0, \mathbf{P}[c(u)]\rangle$ *and* $\bar{s}(u) = \lambda_n\psi(u)\langle\tilde{\boldsymbol{\pi}}, \mathbf{H}[c(u)]\rangle$

(2) $m_-^2 \leqslant \bar{d}(u)/\lambda_n \leqslant m_+^2$ *and* $m_-^3 \leqslant \bar{s}(u)/\lambda_n \leqslant m_+^3$

(3) $m_- \leqslant \bar{d}_T/n\lambda_n \leqslant m_+$ *and* $m_-^2 \leqslant \bar{s}_T/n\lambda_n \leqslant m_+^2$

(4) $m_-^4/m_+^1 \leqslant r_{uv}(\bar{\mathbf{d}})/\rho_n \leqslant m_+^4/m_-^1$ *and* $m_-^6/m_+^2 \leqslant r_{uv}(\bar{\mathbf{s}})/\rho_n \leqslant m_+^6/m_-^2$

(5) $m_-^2/m_+^2 \leqslant f_{uv}(\phi,\psi) \leqslant m_+^2/m_-^2$ *and* $m_-^{10}/m_+^3 \leqslant f_{uv}(\bar{\mathbf{d}},\bar{\mathbf{s}}) \leqslant m_+^{10}/m_-^3$

(6) $0 \leqslant V_{uv}(\bar{\mathbf{d}},\bar{\mathbf{s}}) \leqslant (\eta m_+^2/m_-^2 + m_+^{10}/m_-^3)^2$

(7) $0 \leqslant \kappa_*(\bar{\mathbf{d}},\bar{\mathbf{s}}) \leqslant g(\eta, m_-, m_+)$ *where g is a deterministic function.*

(8) There exist global constants $0 < m_1 < m_2 < \infty$ independent of n such that for any node set $B \subseteq N$,

$$m_1|B|\rho_n \;\leqslant\; \mu(u,B|\bar{\mathbf{s}}),\; \sigma(u,B|\theta_*)^2 \;\leqslant\; m_2|B|\rho_n$$

Proof For (1), we have

$$\begin{aligned}\bar{s}(u) := \mathbb{E}S(u) &= \sum_{j=1}^K\sum_{v\in\mathcal{C}_j}\mathbb{E}W_{uv} = \sum_{j=1}^K\sum_{v\in\mathcal{C}_j}\rho_n r_{uv}(\phi)\mathbf{H}_{c(u)j}\\ &= \rho_n\sum_{j=1}^K\phi(u)n\tilde{\pi}_j\mathbf{H}_{c(u)j} = \lambda_n\phi(u)\langle\tilde{\boldsymbol{\pi}}, \mathbf{H}_{c(u)}\rangle\end{aligned}$$

An identical calculation yields the expression for $\bar{d}(u)$. The inequalities in (2) then follow from Assumption 5. For (3), we again apply Assumption 5 to the equation

$$\bar{s}_T = \sum_{i=1}^K\sum_{v\in\mathcal{C}_i}\bar{s}(u) = \sum_{i=1}^K n\lambda_n\phi(u)\langle\tilde{\boldsymbol{\pi}}, \mathbf{H}_i\rangle = n\lambda_n\tilde{\boldsymbol{\pi}}^T\mathbf{H}\tilde{\boldsymbol{\pi}}$$

An identical equation yields the inequality for $\bar{d}_T$. (2) and (3) directly yield the inequalities in (4). Note that Assumption 5 implies $m_-^2 \leqslant nr_{uv}(\phi), nr_{uv}(\psi) \leqslant m_+^2$, which yields the first inequality of (5). The second inequality of (5) follows from (4). For part (6), note that by Assumption 6 and the first inequality in (5), we have

$$W_{uv} := f_{uv}(\phi,\psi)\xi_{uv} \leqslant (m_+^2/m_-^2)\eta \tag{25}$$

The second inequality in (5) then yields (6). For part (7), recalling the definition of $\kappa_*(\bar{\mathbf{d}},\bar{\mathbf{s}})$ from Definition 11, note first that, by (6), $0 \leqslant v_{uv}(\bar{\mathbf{d}},\bar{\mathbf{s}}) \leqslant (\eta m_+^2/m_-^2 + m_+^{10}/m_-^3)^2$. Thus, by the second inequality (5),

$$0 \leqslant \kappa_*(\bar{\mathbf{d}},\bar{\mathbf{s}}) := \frac{\sum_{u,v:A_{uv}=1} v_{uv}(\bar{\mathbf{d}},\bar{\mathbf{s}})}{\sum_{u,v:A_{uv}=1} f_{uv}(\bar{\mathbf{d}},\bar{\mathbf{s}})} \leqslant \frac{(\eta m_+^2/m_-^2 + m_+^{10}/m_-^3)^2}{m_-^{10}/m_+^3}$$

For part (8), recall that

$$\mu(u,B|\bar{\mathbf{s}}) := \sum_{v\in B} r_{uv}(\bar{\mathbf{s}})$$

The first inequality in (8) follows from applying the second inequality in (4). Similarly,

$$\sigma(u,B|\theta_*)^2 := \sum_{v\in B} r_{uv}(\bar{\mathbf{s}})f_{uv}(\bar{\mathbf{d}},\bar{\mathbf{s}})(1 - \tilde{r}_{uv}(\bar{\mathbf{d}}) + \kappa_*(\bar{\mathbf{d}},\bar{\mathbf{s}}))$$

The second inequality in part (8) follows from parts (4), (5), and (7). ■

The next lemma shows that, if the degrees and strengths of $\mathcal{G}_n$ are bounded around their expected values, the empirical estimate of variance is bounded around the conditional population estimate, and the coefficient of variation of $S_n(u,B)$ is bounded around its population value. Define $D_T := \sum_{u\in N} D(u)$ as the (random) total degree. Recall that λ_n is the asymptotic order the average of the *expected* degrees $\bar{d}_T$.

Lemma 13 *Fix $n > 1$. Suppose Assumption 5 holds. Define*

$$M(\mathbf{D}, \mathbf{S}) := \max_{u \in N} \left\{ |S(u) - \bar{s}(u)|, |D(u) - \bar{d}(u)| \right\}. \tag{26}$$

Then the following statements hold:

(1) There exists small enough $t > 0$ such that if $M(\mathbf{D}, \mathbf{S}) \leqslant \lambda_n t$,

$$\left| \hat{\kappa}(\mathbf{D}, \mathbf{S}) - \kappa_*(\bar{\mathbf{d}}, \bar{\mathbf{s}}) \right| = \left| \frac{\sum_{u,v:A_{uv}=1} V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) - v_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})}{\sum_{u,v:A_{uv}=1} f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})^2 + D_T \rho_n O(t)} \right| + \rho_n O(t)$$

(2) Fix a constant $\varepsilon > 0$ independent of n. Assume $|\hat{\kappa}(\mathbf{D}, \mathbf{S}) - \kappa_*(\bar{\mathbf{d}}, \bar{\mathbf{s}})| \leqslant \varepsilon$. Then then there exists small enough $t > 0$ (not depending on ε) such that if $M(\mathbf{D}, \mathbf{S}) \leqslant t$, for all $B \subseteq N$, we have

$$\left| \frac{\mu(u, B|\Theta)}{\sigma(u, B|\Theta)} - \frac{\mu(u, B|\theta_*)}{\sigma(u, B|\theta_*)} \right| = \sqrt{|B|\rho_n} O(t)$$

Proof $M(\mathbf{D}, \mathbf{S}) \leqslant \lambda_n t$ implies there exists a n-vector $\mathbf{a}_t$ with components in the interval $[-1, 1]$ such that $S(u) = \bar{s}(u) + \lambda_n t a_t(u)$. Therefore, defining $\bar{a}_t := n^{-1} \sum_v a_t(v)$,

$$\begin{aligned}
r_{uv}(\mathbf{S}) - r_{uv}(\bar{\mathbf{s}}) &= \frac{\{\bar{s}(u) + \lambda_n a_t(u) t\}\{\bar{s}(v) + \lambda_n a_t(v) t\}}{\bar{s}_T + n\lambda_n \bar{a}_t t} - \frac{\bar{s}(u)\bar{s}(v)}{\bar{s}_T} \\
&= \frac{\bar{s}_T\{\bar{s}(u)a_t(v) + \bar{s}(v)a_t(u) + \lambda_n a_t(u)a_t(v)t\}\lambda_n t - \bar{s}(u)\bar{s}(v)n\lambda_n \bar{a}_t t}{\bar{s}_T\{\bar{s}_T + n\lambda_n \bar{a}_t t\}} \\
&= \left\{ \frac{\bar{s}(u)a_t(v) + \bar{s}(v)a_t(u) + \lambda_n a_t(u)a_t(v)t - r_{uv}(\bar{s})n\bar{a}_t}{\bar{s}_T + n\lambda_n \bar{a}_t t} \right\} \lambda_n t
\end{aligned}$$

Using parts (2)-(4) of Lemma 12, for sufficiently small t we have

$$\left| r_{uv}(\mathbf{S}) - r_{uv}(\bar{\mathbf{s}}) \right| \leqslant \frac{2\lambda_n m_+^3 + \lambda_n t + \rho_n (m_+^6/m_-^2) n}{n\lambda_n m_-^2 - n\lambda_n t} \lambda_n t = \frac{2m_+^3 + t + (m_+^6/m_-^2)}{m_-^2 - t} \rho_n t$$

Therefore,

$$|r_{uv}(\mathbf{S}) - r_{uv}(\bar{\mathbf{s}})| = \rho_n O(t) \tag{27}$$

as $t \to 0$. By a similar argument, $|r_{uv}(\mathbf{D}) - r_{uv}(\bar{\mathbf{d}})| = \rho_n O(t)$. It follows that

$$|f_{uv}(\mathbf{D}, \mathbf{S}) - f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})| = \rho_n O(t). \tag{28}$$

Therefore, using Equations 27-28 and part (7) of Lemma 12,

$$\begin{aligned}
V_{uv}(\mathbf{D}, \mathbf{S}) &:= (W_{uv} - f_{uv}(\mathbf{D}, \mathbf{S}))^2 \\
&= (W_{uv} - f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}))^2 + 2(W_{uv} - f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}))(f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) - f_{uv}(\mathbf{D}, \mathbf{S})) \\
&\quad + (f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) - f_{uv}(\mathbf{D}, \mathbf{S}))^2 \\
&= V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})^2 + 2V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})(f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) - f_{uv}(\mathbf{D}, \mathbf{S})) + (f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) - f_{uv}(\mathbf{D}, \mathbf{S}))^2 \\
&\leqslant V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})^2 + \rho_n O(t) + \rho_n^2 O(t^2) = V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})^2 + \rho_n O(t)
\end{aligned}$$

Define the following:

$$V_T := \sum_{u,v:A_{uv}=1} V_{uv}(\mathbf{D}, \mathbf{S}), \quad \bar{V}_T := \sum_{u,v:A_{uv}=1} V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}).$$

Since $D_T := \sum_{u \in N} D(u) = \sum_{u,v:A_{uv}=1} 1$, the above inequality implies that $V_T = \bar{V}_T + D_T \rho_n O(t)$. Define similarly:

$$g_T := \sum_{u,v:A_{uv}=1} f_{uv}(\mathbf{D}, \mathbf{S})^2, \quad \bar{g}_T := \sum_{u,v:A_{uv}=1} f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})^2.$$

Similar logic gives $g_T = \bar{g}_T + D_T \rho_n O(t)$. Finally, define $\bar{v}_T := \sum_{u,v:A_{uv}=1} v_{uv}(\bar{\mathbf{s}}, \bar{\mathbf{d}})$. Then

$$\begin{aligned}
\left| \hat{\kappa}(\mathbf{D}, \mathbf{S}) - \kappa_*(\bar{\mathbf{d}}, \bar{\mathbf{s}}) \right| &= \left| \frac{V_T}{g_T} - \frac{\bar{v}_T}{\bar{g}_T} \right| = \left| \frac{\bar{V}_T + D_T \rho_n O(t)}{\bar{g}_T + D_T \rho_n O(t)} - \frac{\bar{v}_T}{\bar{g}_T} \right| \\
&= \left| \frac{\bar{V}_T + D_T \rho_n O(t) - \frac{\bar{v}_T}{\bar{g}_T} \{\bar{g}_T + D_T \rho_n O(t)\}}{\bar{g}_T + D_T \rho_n O(t)} \right| \\
&\leqslant \left| \frac{\bar{V}_T - \bar{v}_T}{\bar{g}_T + D_T \rho_n O(t)} \right| + \left| \frac{D_T \rho_n O(t) - \frac{\bar{v}_T}{\bar{g}_T} D_T \rho_n O(t)}{\bar{g}_T + D_T \rho_n O(t)} \right|
\end{aligned}$$

Note that $\bar{v}_T / D_T$ and $\bar{g}_T / D_T$ are, each, by parts (5) and (6) of Lemma 12, bounded above and below by constants independent of A, t, and n. Therefore, dividing through by D_T,

$$\left| \frac{D_T \rho_n O(t) - \frac{\bar{v}_T}{\bar{g}_T} D_T \rho_n O(t)}{\bar{g}_T + D_T \rho_n O(t)} \right| \leqslant \frac{\rho_n O(t)}{\bar{g}_T / D_T + \rho_n O(t)} = \rho_n O(t)$$

This proves part 1. For part 2, first recall that $\mu(u, B|\Theta) \equiv \mu(u, B|\mathbf{S}) := \sum_{v \in B} r_{uv}(\mathbf{S})$. Therefore by Equation 27, we have

$$|\mu(u, B|\Theta) - \mu(u, B|\theta_*)| = \left| \sum_{v \in B} r_{uv}(\mathbf{S}) - r_{uv}(\bar{\mathbf{s}}) \right| = |B| \rho_n O(t) \tag{29}$$

Recall further that

$$\sigma(u, B|\Theta)^2 := \sum_{v \in B} r_{uv}(\mathbf{S}) f_{uv}(\mathbf{D}, \mathbf{S}) \left(1 - r_{uv}(\mathbf{D}) + \hat{\kappa}(\mathbf{D}, \mathbf{S})\right)$$

Using some straightforward algebra and applying Equations 27-28, we have

$$\begin{aligned}
\left| \sigma(u, B|\Theta)^2 - \sigma(u, B|\theta_*)^2 \right| &= |B| \left(1 + \left| \hat{\kappa}(\mathbf{D}, \mathbf{S}) - \kappa_*(\bar{\mathbf{d}}, \bar{\mathbf{s}}) \right| \right) \rho_n O(t) \\
&= |B| \rho_n O(t)
\end{aligned} \tag{30}$$

where the second line follows from the assumption that $|\hat{\kappa}(\mathbf{D}, \mathbf{S}) - \kappa_*(\bar{\mathbf{d}}, \bar{\mathbf{s}})| \leqslant \varepsilon$. We will now bound $\sigma(u, B|\Theta)$ close to $\sigma(u, B|\theta_*)$ using Equation 30 and a Taylor expansion. Define the function $h(x, \sigma) := \sqrt{\sigma^2 + x}$. For fixed σ, a Taylor expansion around $x = 0$ gives

$h(x,\sigma) = \sigma + \sum_{k=1}^{\infty}(-1)^k \frac{x^k}{k!\sigma^{2k-1}}$. Setting $x = \sigma(u,B|\Theta)^2 - \sigma(u,B|\theta_*)^2$ and $\sigma = \sigma(u,B|\theta_*)$ and applying Equation 30, we obtain

$$\begin{aligned}\sigma(u,B|\Theta) &= h(x, \sigma(u,B|\theta_*)) \\ &= \sigma(u,B|\theta_*) + \sum_{k=1}^{\infty}(-1)^k \frac{|B|^k \rho_n^k O(t^k)}{k!\sigma(u,B|\theta_*)^{2k-1}} \end{aligned} \tag{31}$$

Part (8) of Lemma 12 implies that $\sigma(u,B|\theta_*) \asymp \sqrt{|B|\rho_n}$. Equation 31 therefore gives

$$\sigma(u,B|\Theta) = \sigma(u,B|\theta_*) + \sqrt{|B|\rho_n}O(t) \tag{32}$$

using Equations 29 and 32, we write

$$\left|\frac{\mu(u,B|\Theta)}{\sigma(u,B|\Theta)} - \frac{\mu(u,B|\theta_*)}{\sigma(u,B|\theta_*)}\right| = \left|\frac{\mu(u,B|\theta_*) + |B|\rho_n O(t)}{\sigma(u,B|\theta_*) + \sqrt{|B|\rho_n}O(t)} - \frac{\mu(u,B|\theta_*)}{\sigma(u,B|\theta_*)}\right| \tag{33}$$

As shorthands, define $\bar{\mu}_n := \mu(u,B|\theta_*)/|B|\rho_n$ and $\bar{\sigma}_n := \sigma(u,B|\theta_*)/\sqrt{|B|\rho_n}$. Part (8) of Lemma 12 implies that $\bar{\mu}_n, \bar{\sigma}_n \asymp 1$. Thus, using Equation 33 and dividing through by the appropriate factors,

$$\begin{aligned}\left|\frac{\mu(u,B|\Theta)}{\sigma(u,B|\Theta)} - \frac{\mu(u,B|\theta_*)}{\sigma(u,B|\theta_*)}\right| &= \sqrt{|B|\rho_n}\left|\frac{\bar{\mu}_n + O(t)}{\bar{\sigma}_n + O(t)} - \frac{\bar{\mu}_n}{\bar{\sigma}_n}\right| \\ &= \sqrt{|B|\rho_n}O(t)\end{aligned}$$

This completes part 2. ■

The proof of Lemma 4 from the main text (below) makes use of Lemma 13 by showing that its assumption holds with high probability, for appropriate t.

C.1 Proof of Theorem 4

Throughout, we will sometimes suppress dependence on n for notational convenience. Recall that $A(u,B,\mathcal{G}) := S(u,B,\mathcal{G}) - \mu(u,B|\mathbf{S})$, the deviation of the CCME test statistic from its expected value under the continuous configuration model. Recalling that $\Theta := (\mathbf{D},\mathbf{S},\hat{\kappa}(\mathbf{D},\mathbf{S}))$, define also the random Z-statistic

$$Z(u,B,\mathcal{G}|\Theta) := \frac{A(u,B,\mathcal{G})}{\sigma(u,B|\Theta)}. \tag{34}$$

Define the random p-value

$$P(u,B,\mathcal{G}|\Theta) := 1 - \Phi(Z(u,B,\mathcal{G}|\Theta)). \tag{35}$$

The random variable $P(u,B,\mathcal{G}|\Theta)$ is the random version of the p-value $p(u,B_n|\theta)$ obtained from the approximation in Equation (11). As a consequence of the Benjamini-Hochberg procedure, the event $\{U_\alpha(B_n,\mathcal{G}) = C_n\}$ will occur if

$$\begin{aligned}&P(u,B_n,\mathcal{G}_n|\Theta) \leqslant q\alpha, \quad \text{for all } u \in C_n, \quad \text{and} \\ &P(u,B_n,\mathcal{G}_n|\Theta) > q\alpha, \quad \text{for all } u \notin C_n,\end{aligned} \tag{36}$$

since by assumption $|C_n| > qn$. Let h be the density function of a standard-Normal. By a well-known inequality for the CDF of a standard-Normal, if $Z(u,B_n,\mathcal{G}_n|\Theta) > 0$,

$$P(u,B_n,\mathcal{G}_n|\Theta) \leqslant \frac{1}{Z(u,B_n,\mathcal{G}_n|\Theta)}h(Z(u,B_n,\mathcal{G}_n|\Theta)). \tag{37}$$

By symmetry, if $Z(u,B_n,\mathcal{G}_n|\Theta) < 0$, then

$$P(u,B_n,\mathcal{G}_n|\Theta) \geqslant 1 + \frac{1}{Z(u,B_n,\mathcal{G}_n|\Theta)}h(Z(u,B_n,\mathcal{G}_n|\Theta)). \tag{38}$$

We therefore analyze the concentration properties of $Z(u,B_n,\mathcal{G}_n|\Theta)$ and apply Inequalities 37 and 38 to show that for sufficiently large n, the event in Equation 36 occurs with high probability. We will focus on the first line of 36 first; the second is shown similarly. Recall that θ_* is the empirical population null parameters of $\mathcal{G}_n$, defined after Definition 11. For the derivation below we use the following shorthands: $Y \equiv S(u,B_n,\mathcal{G}_n)$, $\mu \equiv \mu(u,B_n|\mathbf{S}_n)$, $\sigma := \sigma(u,B_n|\Theta)$, $\bar{y} \equiv \mathbb{E}Y$, $\bar{\mu} \equiv \mu(u,B_n|\theta_*)$, and $\bar{\sigma} := \sigma(u,B_n|\theta_*)$. Note

$$\begin{aligned}Z(u,B_n,\mathcal{G}_n|\Theta) := \frac{Y-\mu}{\sigma} = \frac{Y-\bar{\mu}}{\bar{\sigma}} - \left(\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}\right) = \frac{\bar{y}-\bar{\mu}}{\bar{\sigma}} + \frac{Y-\bar{y}}{\bar{\sigma}} - \left(\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}\right) \\ \geqslant \frac{\bar{y}-\bar{\mu}}{\bar{\sigma}} - \left|\frac{Y-\bar{y}}{\bar{\sigma}}\right| - \left|\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}\right|\end{aligned} \tag{39}$$

Define

$$\bar{z}(u,B_n|\theta_*) := \frac{\bar{y}-\bar{\mu}}{\bar{\sigma}} = \lambda_n \frac{\tilde{a}(u,B_n|\bar{\mathbf{s}})}{\sigma(u,B_n|\theta_*)}$$

where $\tilde{a}(u,B_n|\bar{\mathbf{s}})$ is the normalized population version of $A(u,B_n|\mathbf{S})$, as defined in Equation 13 from the main text. The definition above works with Equation 39 to produce the illustrative inequality

$$Z(u,B_n,\mathcal{G}_n|\Theta) \geqslant \bar{z}(u,B_n|\theta_*) - \left|\frac{Y-\bar{y}}{\bar{\sigma}}\right| - \left|\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}\right|. \tag{40}$$

Inequality 40 exemplifies that, if the right-hand terms vanish, $Z(u,B_n,\mathcal{G}_n|\Theta)$ can be approximated by a population version. Our analysis therefore reduces to bounding the right-hand order terms in probability.

Explicitly, consider that by part (8) of Lemma 12, there exists $m_2 > 0$ such that $\sigma(u,B_n|\theta_*)^2 \leqslant m_2 n\rho_n = m_2\lambda_n$. Combining this with the crucial assumption on $\tilde{a}(u,B_n)$ from line 14 from the main text, we have that for all $u \in C_n$,

$$\bar{z}(u,B_n|\theta_*) = \lambda_n \frac{\tilde{a}(u,B_n|\bar{\mathbf{s}})}{\sigma(u,B_n|\theta_*)} \geqslant \sqrt{\lambda_n}\frac{\Delta}{\sqrt{m_2}} \tag{41}$$

Therefore, the rest of the proof is mainly dedicated to showing that the final two terms in line (40) are $o_P(\sqrt{\lambda_n})$. This will imply that $Z(u,B_n,\mathcal{G}_n|\Theta) = \Omega_P(\sqrt{\lambda_n})$ and, using Inequality 37, that $\{P(u,B_n,\mathcal{G}_n|\Theta) \leqslant q\alpha,\ \forall\ u \in C_n\}$ has probability approaching 1.

Step 1: $|\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}| = O_P(\sqrt{\log n})$

For $t > 0$, define the event

$$\mathcal{E}_1(t) := \left\{ \max_{u \in N} |S(u) - \bar{s}(u)|, \max_{u \in N} \left|D(u) - \bar{d}(u)\right| \leqslant \lambda_n t \right\} \tag{42}$$

Fix arbitrary $b > 0$ independent of all other quantities and define $t_n(b) := \sqrt{\frac{b \log n}{\lambda_n}}$. Note that $t_n(b) \to 0$ for any b, by the assumptions of the Theorem. Recall that $D_T := \sum_{u \in N} D(u)$, the (random) total degree. For notational convenience, let $E := \{\text{pairs } u, v : A_{uv} = 1\}$. By part 1 of Lemma 13, the event $\mathcal{E}_1(t_n(b))$ implies

$$\left|\hat{\kappa}(\mathbf{D}, \mathbf{S}) - \kappa_*(\bar{\mathbf{d}}, \bar{\mathbf{s}})\right| = \left| \frac{\sum_E V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) - v_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})}{\sum_E f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})^2 + D_T \rho_n O(t_n(b))} \right| + \rho_n O(t_n(b)) \tag{43}$$

By Lemma 12 part (5),

$$0 \leqslant V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) \leqslant (\eta m_+^2 / m_-^2 + m_+^{10} / m_-^3)^2.$$

Recall that $v_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) := \mathbb{E} V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})$, and that the edge weights that comprise the (upper-triangle of the) weight matrix W are independent. For a fixed adjacency matrix A, Bernstein's Inequality therefore gives

$$\mathbb{P}\left(\left| \sum_E V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) - v_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) \right| > \sqrt{b \log n} \;\middle|\; A \right) \leqslant 2 \exp\left\{ \frac{-2b \log n}{2a_1 + \frac{2}{3} a_2 \sqrt{b \log n}} \right\} \tag{44}$$

Now by Lemma 12 part (6), $\sum_E f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})^2 \geqslant D_T \frac{m_-^{10}}{m_+^3}$. Thus

$$\sum_E f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})^2 + D_T \rho_n O(t_n(b)) \geqslant D_T \frac{m_-^{10}}{m_+^3} / 2$$

for large enough n, since $\rho_n t_n(b) \to 0$. Therefore there exist constants $a_1, a_2 > 0$ depending only on m_+, m_-, and η such that

$$\mathbb{P}\left(\left| \frac{\sum_E V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) - v_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})}{\sum_E f_{uv}(\bar{\mathbf{s}}, \bar{\mathbf{d}})^2 + D_T \rho_n O(t_n(b))} \right| > \sqrt{\tfrac{b \log n}{D_T}} \;\middle|\; A \right) \leqslant 2 \exp\left\{ \frac{-2b \log n}{2a_1 + \frac{2}{3} a_2 \sqrt{\frac{b \log n}{D_T}}} \right\} \tag{45}$$

The above expression is conditional on a fixed adjacency matrix A. We now bound in probability the functionals of A on which the expression depends. It is easily derivable from the statement of the WSBM and Assumption 5 that there exist constants a_3, a_4 depending on m_+ and m_- such that $\mathbb{E}(D_T) = a_3 n \lambda_n$ and $\mathrm{Var}(D_T) = a_4 n \lambda_n$. Therefore, by another application of Bernstein's Inequality,

$$\mathbb{P}\left(|D_T - a_3 n \lambda_n| > \sqrt{n \lambda_n b \log n} \right) \leqslant 2 \exp\left\{ \frac{-2b \log n}{2a_4 + \frac{2}{3} \sqrt{\frac{b \log n}{n \lambda_n}}} \right\} \tag{46}$$

Applying this to inequality (45), the law of total probability gives

$$\begin{aligned}
&\mathbb{P}\left(\left| \frac{\sum_E V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) - v_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})}{\sum_E f_{uv}(\bar{\mathbf{s}}, \bar{\mathbf{d}})^2 + D_T \rho_n O(t_n(b))} \right| > \sqrt{\frac{b \log n}{a_3 n \lambda_n - \sqrt{n \lambda_n b \log n}}} \right) \\
&\leqslant 2 \exp\left\{ \frac{-2b \log n}{2a_1 + \frac{2}{3} a_2 \sqrt{\frac{b \log n}{a_3 n \lambda_n - \sqrt{n \lambda_n b \log n}}}} \right\} + 2 \exp\left\{ \frac{-2b \log n}{2a_4 + \frac{2}{3} \sqrt{\frac{b \log n}{n \lambda_n}}} \right\} = O(n^{-b})
\end{aligned} \tag{47}$$

for sufficiently large n. Along with Equation (43), this implies there exists a constant A_0 depending on parameter constraints such that

$$\mathbb{P}\left\{ \left|\hat{\kappa}(\mathbf{D}, \mathbf{S}) - \kappa_*(\bar{\mathbf{d}}, \bar{\mathbf{s}})\right| \leqslant A_0 \left(\sqrt{\frac{b \log n}{n \lambda_n}} + \rho_n t_n(b) \right) \right\} \geqslant \mathbb{P}(\mathcal{E}_1(t_n(b))) - O(n^{-b}) \tag{48}$$

for sufficiently large n. We now assess $\mathbb{P}(\mathcal{E}_1(t_n(b)))$. Note that for all $u \in N$, $\mathrm{Var}(S(u)) = O(\lambda_n)$. Furthermore, recall from Inequality 25 (in the proof of Lemma 12) that $W_{uv} \leqslant m_+^2 \eta / m_-^2$ for all $u, v \in N$. For fixed $b > 0$, Bernstein's Inequality therefore gives, for any $u \in N$,

$$\mathbb{P}\left(|S(u) - \bar{s}(u)| > \sqrt{b \log n \lambda_n} \right) \leqslant 2 \exp\left\{ \frac{-2a_0 b \log n}{2 + \frac{2}{3} \sqrt{\frac{b \log n}{\lambda_n}}} \right\}, \tag{49}$$

where a_0 is a constant independent of n. The constant a_0 may be chosen so that, similarly,

$$\mathbb{P}\left(\left|D(u) - \bar{d}(u)\right| > \sqrt{b \log n \lambda_n} \right) \leqslant 2 \exp\left\{ \frac{-2a_0 b \log n}{2 + \frac{2}{3} \sqrt{\frac{b \log n}{\lambda_n}}} \right\} \tag{50}$$

Applying a union bound, equations (49) and (50) give

$$\begin{aligned}
\mathbb{P}(\mathcal{E}_1(t_n(b))) &\geqslant 1 - 2n \exp\left\{ \frac{-2a_0 b \log n}{2 + \frac{2}{3} \sqrt{\frac{b \log n}{\lambda_n}}} \right\} - 2n \exp\left\{ \frac{-2a_0 b \log n}{2 + \frac{2}{3} \sqrt{\frac{b \log n}{\lambda_n}}} \right\} \\
&= 1 - O(n^{-b+1})
\end{aligned} \tag{51}$$

for sufficiently large n. Returning to the inequality in (48), we therefore have

$$\begin{aligned}
\mathbb{P}\left\{ \left|\hat{\kappa}(\mathbf{D}, \mathbf{S}) - \kappa_*(\bar{\mathbf{d}}, \bar{\mathbf{s}})\right| \leqslant A_0 \left(\sqrt{\tfrac{b \log n}{n \lambda_n}} + \rho_n t_n(b) \right) \right\} &\geqslant \mathbb{P}(\mathcal{E}_1(t_n(b))) - O(n^{-b}) \\
&\geqslant 1 - O(n^{-b+1})
\end{aligned} \tag{52}$$

for sufficiently large n. Recall that by assumption, $\lambda_n / \log n \to \infty$. Thus $t_n(b) \to 0$, and

$$\sqrt{\tfrac{b \log n}{n \lambda_n}} + \rho_n t_n(b) = t_n(b) / \sqrt{n} + \rho_n t_n(b) \leqslant 1 / \sqrt{n} = o(1).$$

Thus, Inequality 52 implies that

$$\mathbb{P}\left(\left|\hat{\kappa}(\mathbf{D}, \mathbf{S}) - \kappa_*(\bar{\mathbf{d}}, \bar{\mathbf{s}})\right| \leqslant \varepsilon \right) \geqslant 1 - O(n^{-b+1}), \tag{53}$$

for sufficiently large n. For $\varepsilon > 0$, define the event $\mathcal{E}_2(\varepsilon) := \left\{\left|\hat{\kappa}(\mathbf{D},\mathbf{S}) - \kappa(\bar{\mathbf{d}},\bar{\mathbf{s}})\right| \leqslant \varepsilon\right\}$. By part 2 of Lemma 3, the event $\mathcal{E}_1(t_n(b)) \cap \mathcal{E}_2(\varepsilon)$ implies

$$\begin{aligned}\left|\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}\right| := \left|\frac{\mu(u,B_n|\mathbf{S})}{\sigma(u,B_n|\Theta)} - \frac{\mu(u,B_n|\bar{\mathbf{s}})}{\sigma(u,B_n|\theta_*)}\right| &= \sqrt{|B_n|\rho_n}O(t_n(b)) \\ &\leqslant \sqrt{\lambda_n}O(t_n(b)). \\ &= O(\sqrt{b\log n}) \end{aligned} \tag{54}$$

Therefore, there exists a constant $A_2 > 0$ such that, by Inequalities 51 and 53,

$$\mathbb{P}\left(\left|\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}\right| \leqslant A_2\sqrt{b\log n}\right) = 1 - O(n^{-b+1}) \tag{55}$$

for sufficiently large n. This completes Step 1.

Step 2: $|\frac{Y-\bar{y}}{\bar{\sigma}}| = O_P(\sqrt{\log n})$.

Note that, as for Inequality 49, Bernstein's Inequality gives

$$\mathbb{P}\left(|S(u,B_n,\mathcal{G}_n) - \mathbb{E}S(u,B_n,\mathcal{G}_n)| > \sqrt{b\log n\lambda_n}\right) \leqslant 2\exp\left\{\frac{-2a_0 b\log n}{2 + \frac{2}{3}\sqrt{\frac{b\log n}{\lambda_n}}}\right\} \tag{56}$$

By Lemma 12 part (8), there exists $m_2 > 0$ such that $\sigma(u,B_n|\theta_*)^2 \leqslant m_2\lambda_n$. Thus,

$$\left|\frac{Y-\bar{y}}{\bar{\sigma}}\right| := \left|\frac{S(u,B_n,\mathcal{G}_n) - \mathbb{E}S(u,B_n,\mathcal{G}_n)}{\sigma(u,B_n|\theta_*)}\right| \geqslant \left|\frac{S(u,B_n,\mathcal{G}_n) - \mathbb{E}S(u,B_n,\mathcal{G}_n)}{m_2\sqrt{\lambda_n}}\right|,$$

so by Inequality 56, we have for sufficiently large n that

$$\mathbb{P}\left(\left|\frac{Y-\bar{y}}{\bar{\sigma}}\right| \leqslant \sqrt{\frac{b\log n}{m_2}}\right) \geqslant 1 - O(n^{-b}). \tag{57}$$

This completes Step 2.

We now recall inequality 40:

$$Z(u,B_n,\mathcal{G}_n|\Theta) \geqslant \bar{z}(u,B_n|\theta_*) - \left|\frac{Y-\bar{y}}{\bar{\sigma}}\right| - \left|\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}\right|.$$

In step 1, we showed that there exists a constant A_2 depending only on the fixed WSBM model parameters such that for any fixed $b > 1$, for large enough n, $\left|\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}\right| \leqslant A_2\sqrt{b\log n}$ with probability $1 - O(n^{-b+1})$. In step 2, we showed that there exists a constant m_2 depending only on the fixed WSBM model parameters such that for any fixed $b > 1$, for large enough n, $|\frac{Y-\bar{y}}{\bar{\sigma}}| \leqslant \sqrt{b\log n/m_2}$ with probability $1 - O(n^{-b})$. Recall furthermore from inequality 41 that $\bar{z}(u,B_n|\theta_*) \geqslant \Delta\sqrt{\lambda_n/m_2}$, where Δ is from condition 14 in the statement of the Theorem. We can therefore write that for any fixed $b > 1$, for large enough n,

$$Z(u,B_n,\mathcal{G}_n|\Theta) \geqslant \Delta\sqrt{\lambda_n/m_2} - \sqrt{b\log n/m_2} - A_2\sqrt{b\log n} = A_3\sqrt{\lambda_n} - A_4\sqrt{b\log n}$$

with probability at least $1 - O(n^{-b+1})$. Now, by assumption, $|C_n| \geqslant qn$. Therefore, using Inequality 37 and a union bound, we can write that for any fixed $b > 1$, for large enough n,

$$\max_{u\in C_n} P(u,B_n,\mathcal{G}_n|\Theta) \leqslant \exp\{-(A_3\sqrt{\lambda_n} - A_4\sqrt{b\log n})^2\} \tag{58}$$

with probability at least $1 - O(n^{-b+2})$. Note that for any fixed b, the right-hand-side of inequality 58 vanishes, due to the assumption that $\lambda_n/\log n \to \infty$. Thus, for $b > 2$, inequality 58 implies that for large enough n (now depending on choice of b), the event $\{P(u,B_n,\mathcal{G}_n|\Theta) \leqslant q\alpha,\ \forall\ u \in C_n\}$ has probability $1 - O(n^{-b+2}) \to 1$.

It can be similarly shown that the second half of the event in (36) has probability approaching 1. Instead of Inequality 40 we (similarly) derive

$$Z(u,B_n,\mathcal{G}_n|\Theta) \leqslant \bar{z}(u,B_n|\theta_*) + \left|\frac{Y-\bar{y}}{\bar{\sigma}}\right| + \left|\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}\right| \tag{59}$$

This is useful because if $u \notin C_n$, assumption (14) ensures that $\tilde{a}_n(u,B_n)\bar{\mathbf{s}} < -\Delta$, and hence

$$\bar{z}(u,B_n|\theta_*) := \frac{\bar{y}-\bar{\mu}}{\bar{\sigma}} = \lambda_n\frac{\tilde{a}(u,B_n|\bar{\mathbf{s}})}{\sigma(u,B_n|\theta_*)} \leqslant \lambda_n\frac{-\Delta}{\sigma(u,B_n|\theta_*)} \leqslant \sqrt{\lambda_n}\frac{-\Delta}{\sqrt{m_1}}$$

where the last inequality follows from part (8) of Lemma 12. Steps 1 and 2 therefore work to show that for any fixed $b > 1$, for large enough n,

$$Z(u,B_n,\mathcal{G}_n|\Theta) \leqslant -\Delta\sqrt{\lambda_n/m_2} + \sqrt{b\log n/m_2} + A_2\sqrt{b\log n} = A_3\sqrt{\lambda_n} - A_4\sqrt{b\log n}$$

With probability $1 - O(n^{-b+1})$. Inequality 38 then implies that

$$\mathbb{P}\left(\max_{u\notin C_n} P(u,B_n,\mathcal{G}_n|\Theta) \geqslant 1 - \exp\{-(A_3\sqrt{\lambda_n} - A_4\sqrt{b\log n})^2\}\right) \geqslant 1 - O(n^{-b+2}) \tag{60}$$

With reasoning identical to the result for $u \in C_n$, this implies that for any $b > 2$, for large enough $n(b)$, the event $\{P(u,B_n,\mathcal{G}_n|\Theta) > q\alpha,\ \forall\ u \notin C_n\}$ has probability at least $1 - O(n^{-b+2}) \to 1$. Applying a union bound to the event in (36) completes the proof. ■

C.2 Proof of Theorem 5

We will show that if the condition in (15) holds, then the condition in (14) from Theorem 4 holds when $B_n = C_n = C_{j,n}$ simultaneously across all $j \in \{1,2,\ldots,K\}$. This involves representing (14) in terms of the model parameters when $B_n = C_n = C_{j,n}$. Specifically, we derive the normalized population deviation $\tilde{a}(u,C_{j,n}|\bar{\mathbf{s}}) := (\mathbb{E}S(u,C_{j,n},\mathcal{G}_n) - \mu(u,C_{j,n}|\bar{\mathbf{s}}))/\lambda_n$. First, note that for any fixed $j \leqslant K$, part (1) of Lemma 12 gives

$$\sum_{v\in C_{j,n}} \bar{s}(v) \ = \ \lambda_n\langle\tilde{\boldsymbol{\pi}},\mathbf{H}_j\rangle \cdot \sum_{v\in C_{j,n}} \psi(u) \ = \ n\lambda_n\langle\tilde{\boldsymbol{\pi}},\mathbf{H}_j\rangle\tilde{\pi}_j$$

and thus

$$\bar{s}_T := \sum_{v\in N}\bar{s}(v) = \sum_{j=1}^{K}\sum_{v\in C_{j,n}}\bar{s}(v) = n\lambda_n\sum_{j=1}^{K}\langle\tilde{\boldsymbol{\pi}},\mathbf{H}_j\rangle\tilde{\pi}_j = n\lambda_n\tilde{\boldsymbol{\pi}}^t\mathbf{H}\boldsymbol{\pi}.$$

Therefore, again applying part (1) of Lemma 12,

$$\mu(u, C_{j,n}|\bar{\mathbf{s}}) := \sum_{v \in C_{j,n}} r_{uv}(\bar{\mathbf{s}}) = \bar{s}(u) \sum_{v \in C_{j,n}} \frac{\bar{s}(v)}{\bar{s}_T} = \bar{s}(u) \frac{\langle \tilde{\boldsymbol{\pi}}, \mathbf{H}_j \rangle \tilde{\pi}_j}{\tilde{\boldsymbol{\pi}}^t \mathbf{H} \tilde{\boldsymbol{\pi}}}$$
$$= \lambda_n \psi(u) \frac{\langle \tilde{\boldsymbol{\pi}}, \mathbf{H}_{c(u)} \rangle \langle \tilde{\boldsymbol{\pi}}, \mathbf{H}_j \rangle \tilde{\pi}_j}{\tilde{\boldsymbol{\pi}}^t \mathbf{H} \tilde{\boldsymbol{\pi}}}.$$

Secondly,

$$\mathbb{E}S(u, C_{j,n}, \mathcal{G}_n) = \sum_{v \in C_{j,n}} \mathbb{E}W_{uv} = \sum_{v \in C_{j,n}} \rho_n r_{uv}(\psi) \mathbf{H}_{c(u)j} = \lambda_n \psi(u) \mathbf{H}_{c(u)j} \tilde{\pi}_j.$$

Thus,

$$\tilde{a}(u, C_{j,n}|\bar{\mathbf{s}}) := \frac{\mathbb{E}S(u, C_{j,n}, \mathcal{G}_n) - \mu(u, C_{j,n}|\bar{\mathbf{s}})}{\lambda_n}$$
$$= \psi(u) \tilde{\pi}_j \left(\mathbf{H}_{c(u)j} - \frac{\langle \tilde{\boldsymbol{\pi}}, \mathbf{H}_{c(u)} \rangle \langle \tilde{\boldsymbol{\pi}}, \mathbf{H}_j \rangle}{\tilde{\boldsymbol{\pi}}^t \mathbf{H} \tilde{\boldsymbol{\pi}}} \right). \tag{61}$$

If $u \in C_{i,n}$, the expression in the parentheses from the right-hand-side of (61) is the i,j-th element of the matrix $\mathbf{H} - \mathbf{H}\tilde{\mathbf{\Pi}}\mathbf{H}/\tilde{\boldsymbol{\pi}}^t \mathbf{H} \tilde{\boldsymbol{\pi}}$, with $\tilde{\mathbf{\Pi}} := \tilde{\boldsymbol{\pi}} \tilde{\boldsymbol{\pi}}^t$. By Assumption 5, $\psi(u) \geqslant m_-$ for all $u \in N$ and $i \leqslant K$, and $\tilde{\pi}_j$ is fixed. Thus, (15) ensures that (14) holds when $C_n = C_{j,n}$, simultaneously for $j \leqslant K$. Assumption 5 also ensures that there exists $q > 0$ such that for all $j \leqslant K$ and $n > 1$, $|C_{j,n}| > qn$. This allows us to apply Theorem 4 to the sequences $B_n = C_n = C_{j,n}$, for each $j \leqslant K$. A union bound proves the result. ■

Appendix D. Further discussion of the CCME methodology

D.1 Variance in results due to initialization

Since CCME's initialization step is stochastic, it is natural to investigate the variance of results across many applications of CCME to the same network. To do so, we (randomly) selected five replications from each fixed parameter setting of the third experiment described in 6.2.1, in which both edge and edge weight signal levels are increased from low values. At each replication, we ran CCME thirty times, and recorded the pairwise Overlapping Normalized Mutual Information (oNMI, see Section 6) scores between the thirty resulting community sets. In Figure 3, we plot the median and 5%-95% quantile bars for the oNMI scores for each replication, and jitter the lines around the corresponding parameter value. It is clear from the plot that CCME results become more stable, and eventually non-varying, as community signal becomes stronger. This follows the straightforward intuition that when in-community edge/weight signal is weak, CCME finds spurious or imperfect communities, and will find *different* such communities when it is initialized with different random starting sets.

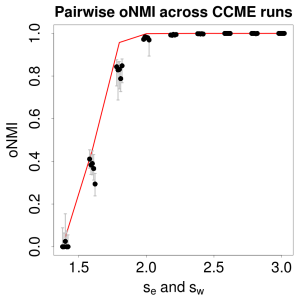


Figure 3: Stability tests for CCME.

D.2 Cycles in Fixed Point Search

As remarked in Section 5.2, it is possible for the SCS algorithm to reach a stable sequence $C_1, \ldots, C_J$ that is traversed by the update $U_\alpha(\cdot, \mathcal{G})$. If this happens, we apply the following routine to re-start the algorithm, or return the union of the sequence:

1. If $C_i \cap C_{i+1} = \phi$ for any $i \leqslant J$, or if $C_J \cap C_1 = \phi$, terminate the iterations and do not extract a community.

2. Otherwise, define $C^* = \cup_{i=1}^J C_i$, and:

   (a) If C^* has been visited previously by SCS, extract C^* into $\mathcal{C}$.

   (b) Otherwise, re-initialize with C^*.

D.3 Filtering of $\mathcal{B}_0$ and $\mathcal{C}$

To filter through $\mathcal{B}_0$ and $\mathcal{C}$, we use an inference procedure based on a set-wise z-statistic, analogous to the node-set z-statistic presented in Section 4. Define $S(B) := \sum_{v \in B} S(v, B)$. Note that $S(B)$ has an easily derivable expectation and standard deviation under the continuous configuration model, which we denote (respectively) by $\mu(B|\theta)$ and $\sigma(B|\theta)$. We define the corresponding z-statistic and an approximate p-value by

$$z(B|\theta) := \frac{S(B) - \mu(B|\theta)}{\sigma(B|\theta)}, \qquad p(B|\theta) := 1 - \Phi(z(B|\theta))$$

Before initializing the SCS algorithm on sets in $\mathcal{B}_0$, we compute the p-value above for each member set, and remove any that are not significant at FDR level $\alpha = 0.05$. This greatly reduces the number of extractions CCME must perform, and reduces the probability of convergence on small, spurious communities.

We also use $z(B|\theta)$ to filter near-matches in $\mathcal{C}$, once all SCS extractions have terminated and empty sets have been removed. To do so, we require an overlap "tolerance" parameter $\tau \in [0, 1]$. First, we create a (non-symmetric) $|\mathcal{C}| \times |\mathcal{C}|$ matrix O with general element $O_{ij} := |C_i \cap C_j|/|C_i|$, which measures the proportional overlap of C_i into C_j. After setting the diagonal of O to zero, the filtering proceeds as follows:

1. Find indices $i \neq j$ corresponding to the maximum entry of O.

2. If $O_{ij} < \tau$, terminate filtering.

3. Remove either C_i or C_j from $\mathcal{C}$, whichever has the smaller $z(B|\theta)$.

4. Re-compute O, set its diagonal to zero, and return to step 1.

For all simulations and real-data analyses in this paper, we employed this algorithm with $\tau = 0.9$. To further decrease the computation time of CCME, as we proceed through $\mathcal{B}_0$, we skip sets that were formed from nodes that have already been extracted into $\mathcal{C}$. We find that, in practice, none of these adjustments harm CCME's ability to find statistically significant overlapping communities. Indeed, the simulation results mentioned in Section 6.2.2 show that CCME outperforms competing methods with overlap capabilities.

Appendix E. Extension to directed networks

Directed networks are networks for which edges and weights carry directionality, in the sense that nodes give and receive different quantities links and edge weight. Preserving directionality can in some applications be important for appropriate analysis of a network data set. The extension of the CCME method to directed networks is straightforward. First, both "in" and "out" observed degrees and strengths, denoted $\mathbf{d}^{\rightarrow} := \{\mathbf{d}_{in}, \mathbf{d}_{out}\}$ and $\mathbf{s}^{\rightarrow} := \{\mathbf{s}_{in}, \mathbf{s}_{out}\}$, are used in the null model. Let $A^{\rightarrow}_{uv}$ be the *ordered* entry of the adjacency matrix, corresponding to the edge from u *into* v, and similarly for weights $W^{\rightarrow}_{uv}$. Let $\mathbf{d}_{T,in}$ and $\mathbf{s}_{T,in}$ be the total in-degree and in-strength. Define

$$r^{\rightarrow}_{uv}(\mathbf{d}^{\rightarrow}) := \frac{d_{out}(u)d_{in}(v)}{d_{T,in}} \tag{62}$$

Define $\tilde{r}^{\rightarrow}_{uv}(\mathbf{d}^{\rightarrow}) := \min(1, r^{\rightarrow}_{uv}(\mathbf{d}^{\rightarrow}))$ and $f^{\rightarrow}_{uv}(\mathbf{d}^{\rightarrow}, \mathbf{s}^{\rightarrow}) := r^{\rightarrow}_{uv}(\mathbf{s}^{\rightarrow})/\tilde{r}^{\rightarrow}_{uv}(\mathbf{d}^{\rightarrow})$. Then the directed continuous configuration model is as follows:

1. $\mathbb{P}(A^{\rightarrow}_{uv} = 1) = \tilde{r}^{\rightarrow}_{uv}$ independently for all ordered node pairs $(u, v) \in N \times N$
2. For each ordered node pair (u, v), generate an independent random variable ξ_{uv} according to F, and assign edge weights by:

$$W^{\rightarrow}_{uv} = \begin{cases} f^{\rightarrow}_{uv}(\mathbf{s}^{\rightarrow}, \mathbf{d}^{\rightarrow})\xi_{uv}, & A_{uv} = 1 \\ 0, & A_{uv} = 0 \end{cases}$$

For directed networks, the formula for the variance parameter is

$$\hat{\kappa}^{\rightarrow}(\mathbf{d}^{\rightarrow}, \mathbf{s}^{\rightarrow}) := \sum_{(u,v):A_{uv}=1} (W^{\rightarrow}_{uv} - f^{\rightarrow}_{uv}(\mathbf{d}^{\rightarrow}, \mathbf{s}^{\rightarrow}))^2 \Big/ \sum_{(u,v):A_{uv}=1} f^{\rightarrow}_{uv}(\mathbf{d}^{\rightarrow}, \mathbf{s}^{\rightarrow})^2, \tag{63}$$

an equation that follows from the same method-of-moments logic presented in Section 2.2. Initialization, set updates, and overlap filtering proceed through use and testing of *out*-weights and *out*-weight sums. The proof of the corresponding Central Limit Theorem proceeds similarly to the proof of Theorem 2 with trivial modifications.

Appendix F. Simulation framework

Here we describe the benchmarking simulation framework used in Section 6. In Table 4, we list and name parameters controlling the network model:

Table 4: Simulation model parameters

n: Number of nodes in communities	n_b: Number of nodes in background
$m_{\max}$: Max community size	$m_{\min}$: Min community size
τ_1: Power-law for degree parameters	τ_2: Power-law for community sizes
k: Mean of degree parameter power-law	$k_{\max}$: Maximum degree parameter
s_e: Within-community edge signal	s_w: Within-community weight signal
o_n: Number of nodes in multiple communities	o_m: Number of memberships for overlap nodes
F: Distributions of edge weights	σ^2: Variance parameter for F
β: Power-law for strength parameters	

F.1 Simulation of community nodes

The framework is capable of simulating networks with or without background nodes. We first describe the simulation procedure without background nodes, i.e. with $n_b = 0$. Later, we describe how to simulate a network with background nodes, which involves a slight modification to the procedure in this subsection. Regardless of the presence of background nodes, the first step is to determine community sizes and node memberships.

F.1.1 Community structure and node degree/strength parameters

Here we describe how to obtain a cover $\mathcal{C} := \{C_1, \dots C_K\}$ of n nodes. The following steps to obtain $\mathcal{C}$ are almost exactly as those from the LFR benchmark in Lancichinetti and Fortunato (2009), used extensively in Lancichinetti et al. (2011) and Xie et al. (2013):

1. Each of the o_n overlapping nodes will have o_m memberships. Let $n_m := n + o_n(o_m - 1)$ be the number of node *memberships* present in the network.
2. Draw community sizes from a power law with maximum value $m_{\max}$, minimum value $m_{\min}$, and exponent $-\tau_2$, until the sum of community sizes is greater than or equal to n_m. If the sum is greater than n_m, we reduce the sizes of the communities proportionally until the sum is equal to n_m.
3. Form a bipartite graph of community markers on one side and node markers on the other. Each community marker has number of empty node slots given by step (b), and each node has a number of memberships given by step (a). Sequentially pair node memberships and community node slots uniformly at random, without replacement, until every node membership is paired with a community.

With the community assignments in hand, simulation of the network proceeds according to the Weighted Stochastic Block Model as outlined in Section 6. We describe choices for particular components of this model in the following subsection.

F.1.2 Simulation of edges and weights

As described in Section 6, we set the $\mathbf{P}$ and $\mathbf{M}$ matrices to have diagonals equal to s_e and s_w (respectively, see Table 4), and off-diagonals equal to 1. We note that this homogeneity facilitates creating networks with overlapping communities. With variance in the diagonal of $\mathbf{P}$, for example, it would not be obvious with what probability to connect overlapping nodes that overlap to two of the same communities, simultaneously. It remains

to obtain the strength and degree propensity parameters ψ and ϕ; we do so analogously to the simulation framework in Lancichinetti et al. (2011). We first draw ϕ from a power law with exponent τ_1, mean k, and maximum $k_{\max}$ (see Table 4). Next we set ψ by the formula $\psi(u) = \phi(u)^{\beta+1}$.

It is worth noting here that, under the model given below, the expected degree of node u is *approximately* $\phi(u)$ and the expected strength *approximately* $\psi(u)$. Therefore, heterogeneity/skewness in ϕ and ψ induce heterogeneity/skewness in the degrees and strengths of the simulated networks. However, by scaling ϕ and ψ, we can force the total expected degree and total expected strength of the simulated networks to exactly match ϕ_T and ψ_T, respectively. The scaling constants depend on $\mathbf{P}$ and $\mathbf{M}$ and are easily derivable from the model's generative algorithm (described in Section 4.2.1).

F.1.3 Parameter settings

Here we list the "default" settings of the simulation model, mentioned in Section 6. The following choices for parameters were made regardless of the simulation setting: $\tau_2 = -2$, $k = \sqrt{n}$, $k_{\max} = 3k$ (three settings which make the degree/strength distributions skewed and the network sparse), $\beta = 0.5$ (to induce a non-trivial power law between strengths and degrees), $\tau_1 = -1$, $m_{min} = n/5$, $m_{max} = 3m_{max}/2$ (settings which produce between about 3 and 7 communities per network with skewed size distribution), and $\sigma^2 = 1/2$. Other parameter choices are specific to the simulation settings described in Section 6.

F.2 Background node simulation

If $n_b > 0$, we generate a network with n community nodes, and then add n_b background nodes, generating all remaining edges and weights according to the continuous configuration null model introduced in the main text. First, we obtain node-wise parameters for all $n+n_b$ nodes, yielding vectors ϕ and ψ as in subsection F.1. In a simulated network without background, $\phi(u)$ and $\psi(u)$ are approximately $\mathbb{E}[d(u)]$ and $\mathbb{E}[s(u)]$, respectively. To ensure that this remains the case in a network for which background nodes are added after the simulation of community nodes, we must split up each $\phi(u)$ and $\psi(u)$ into community and background portions. A few other adjustments must also be made after the simulation of community nodes. To this end, define

- $N_C := \{1, \ldots, n\}$; $N_B := \{n+1, \ldots, n+n_b\}$ (community and background node sets)
- $\phi_{C,T} := \sum_{N_C} \phi(u)$; $\phi_{B,T} := \sum_{N_B} \phi(u)$ (target total degrees of community and background nodes)
- $\phi_C(u) := \frac{\phi_{C,T}}{\phi_T}\phi(u)$; $\phi_B(u) := \frac{\phi_{B,T}}{\phi_T}\phi(u)$ (target edge-counts between u and the community and background nodes)
- $\phi_{1,T} := \sum_{N_C} \phi_C(u)$; $\phi_{2,T} := \sum_{N_B} \phi_B(u)$ (target total degrees of community and background *subnetworks*)
- $d^o_C(u) := \sum_{v \in N_C} A_{uv}$; $d^o_B(u) := \sum_{v \in N_B} A_{uv}$ (observed edge-counts between u and the community and background nodes)

The above definitions exist analogously for the strength parameters ψ (replacing "d" with "s" where appropriate). The word "target" above indicates that we will set up the background simulation model so that these values are the approximate expected values of the graph statistics they represent.

F.2.1 Adjusted community-node simulation model

The only adjustment to be made to the simulation of community nodes, described in subsection F.1.2, is that the degree and strength parameters are set to a certain *fraction* of their original values. This accounts for the eventual addition of background nodes, where the remaining (random) part of each nodes degree and strength is to be simulated. So, the community-node simulation (if background nodes are to be added later) follows the process described in subsection F.1 with degree parameters $\{\phi_C(1), \ldots, \phi_C(n)\}$ and strength parameters $\{\psi_C(1) \ldots \psi_C(n)\}$.

F.2.2 Edges and weights for background

For the simulation of the background nodes (following the community nodes) our goal is to specify adjusted degree/strength parameters ϕ' and ψ' given the observed edge-sums $\{d^o_C(1), \ldots, d^o_C(n)\}$ and weight-sums $\{s^o_C(1), \ldots, s^o_C(n)\}$ from the community nodes. In what follows we describe this specification for ϕ' only; the specification for ψ' is exactly analogous. We first represent ϕ'_T, which we have yet to determine, into community and background totals:

$$\phi'_T = \phi'_{C,T} + \phi'_{B,T}$$

Since the background subnetwork has not yet been generated, we make the specification $\phi'(u) := \phi(u)$ for all $u \in N_B$, and hence $\phi'_{B,T} = \phi_{B,T}$ is known. To address $\phi'_{C,T}$, note that for each community node $u \in N_C$, $\phi'(u)$ may be represented similarly:

$$\phi'(u) = \phi'_C(u) + \phi'_B(u)$$

This reduces the problem of specifying $\phi'(u)$ to specifying $\phi'_C(u)$ and $\phi'_B(u)$. Since the community node subnetwork has already been generated, we set $\phi'_C(u) \leftarrow d^o_C(u)$. Next, recalling that $\phi_B(u) := \frac{\phi_{B,T}}{\phi_T}\phi(u)$, we make the specification $\phi'_B(u) := \frac{\phi_{B,T}}{\phi'_T}\phi(u)$ (which must be solved for via ϕ'_T, in the following). So, in total, we have

$$\phi'(u) = \begin{cases} d^o_C(u) + \frac{\phi_{B,T}}{\phi'_T}\phi(u), & u \in N_C \\ \phi(u), & u \in N_B \end{cases}$$

Therefore we can solve for ϕ'_T with the equation

$$\begin{aligned} \phi'_T &:= \sum_{u \in N_C \cup N_B} \phi'(u) \\ &= \sum_{u \in N_C} \left[d^o_C(u) + \frac{\phi_{B,T}}{\phi'_T}\phi(u) \right] + \sum_{u \in N_B} \phi(u) \\ &= d^o_{C,T} + \frac{\phi_{B,T}}{\phi'_T}\phi_{C,T} + \phi_{B,T} \end{aligned}$$

Where $d^o_{C,T} := \sum_{u \in N_C} d^o_C(u)$. The solution for ϕ'_T from this quadratic is

$$\phi'_T = \frac{\phi_{B,T} + d^o_{C,T}}{2} + \sqrt{\frac{(\phi_{B,T} + d^o_{C,T})^2}{4} + \phi_{C,T}\phi_{B,T}} \tag{64}$$

which then immediately gives the full vector ϕ'. We can now simulate the remaining edges in the network. Specifically, for each $u \in N_B$ and each $v \in N_C \cup N_B$, we simulate an edge according to

$$\mathbb{P}(A_{uv} = 1) = \frac{\phi'(u)\phi'(v)}{\phi'_T} \text{ independent across node pairs} \tag{65}$$

We solve for ψ' analogously. Then for each $u \in N_B$ and each $v \in N$, we simulate an edge weight according to

$$W_{uv} = \begin{cases} f_{uv}(\phi', \psi')\xi_{uv}, & A_{uv} = 1 \\ 0, & A_{uv} = 1 \end{cases}$$

where $\xi \sim F$, is as it was for the generation of the community node subnetwork.

The above simulation steps correspond precisely to the continuous configuration model with parameters $(\phi', \psi', F, \sigma)$. Some basic computational trials have shown that, for large networks, the solution for ϕ'_T is quite close to ϕ_T. Therefore, for each $u \in N_B$, $\mathbb{E}(d(u))$ is almost exactly $\phi(u)$, i.e. what it would be under the model in F.1.2, without background nodes. The same holds for the strengths and expected strengths. Together with equation 65, this implies the background nodes are behaving according to the continuous configuration model, even as they are a sub-network within a larger network with communities.

To illustrate these points, we simulated a sample network from the default framework with parameters $n = 5,000$, $n_b = 1,000$, $s_e = s_w = 3$, disjoint communities, and other parameters specified by F.1.3. These settings are akin to what was used in subsection 6 of the main text. First we plotted ϕ' and ψ' against the empirical strengths and degrees with lowess curves to check the match. Figure 4 shows the fit is essentially linear. Second, for

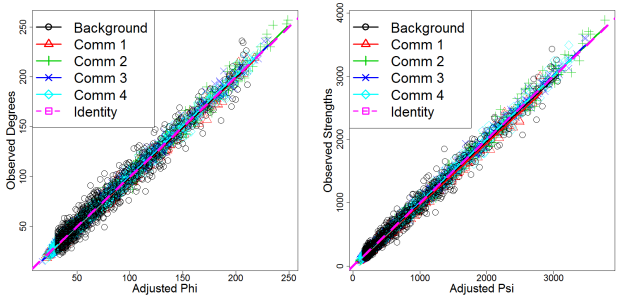


Figure 4: Empirical degrees/strengths vs. adjusted parameters for the example network

each node $u \in N$ and for each node block B (either a true community or the background

node set) we may calculate an empirical z-score for $S(u, B, \mathcal{G})$, as described in subsection 4.1 of the main text. The z-score for $S(u, B, \mathcal{G})$ is a measure of connection significance, with respect to the continuous configuration model (and also modularity, see Section 4.2.3) between u and B. Let K be the number of true communities in the network. For each $i, j = 1, \ldots, K+1$, where $K+1$ is the index of the background node block, we computed the empirical average of z-statistics between nodes u from node block i the node block B corresponding to index j. Theses empirical averages can be arranged in a $(K+1) \times (K+1)$ matrix showing the average inter-block connectivities of the network. In Figure 5 we display a visualization of this matrix, which shows preferential connection within communities, and roughly null connection between the background nodes and all blocks.

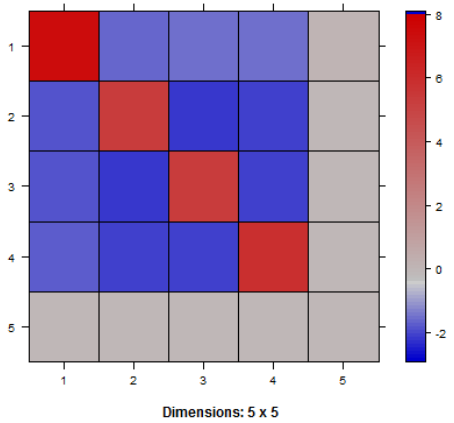


Figure 5: Average empirical z-statistics between nodes and node blocks

F.3 Additional simulation result displays

Here we display the results from our first simulation setting (described in Section 6.2.1) in plots with un-transformed y-axes. Figure 6 reveals that SLPAw seems to respond to increasing community signal, whereas WeightedGT seems to plateau, at least on this range of our parameters.

References

Christopher Aicher, Abigail Z Jacobs, and Aaron Clauset. Learning latent block structure in weighted networks. *Journal of Complex Networks*, page cnu026, 2014.

Reid Andersen, David F Gleich, and Vahab Mirrokni. Overlapping clusters for distributed computation. In *Proceedings of the Fifth ACM International Conference on Web Search*

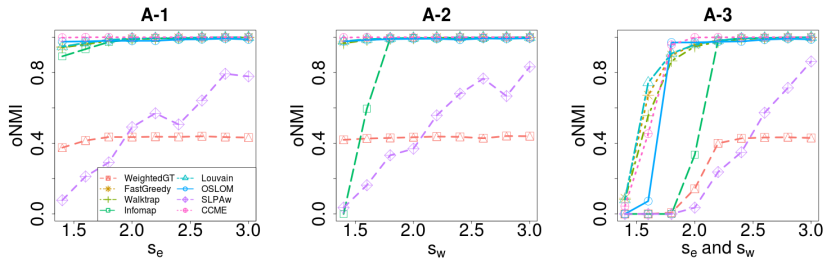


Figure 6: Simulation results described in Section 6.2.1, with un-transformed oNMI scores. Legend refers to all plots.

and Data Mining, pages 273–282. ACM, 2012.

Albert-Laszlo Barabasi and Zoltan N Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113, 2004.

Alain Barrat, Marc Barthelemy, Romualdo Pastor-Satorras, and Alessandro Vespignani. The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(11):3747–3752, 2004.

Edward A Bender. The asymptotic number of non-negative integer matrices with given row and column sums. *Discrete Mathematics*, 10(2):217–223, 1974.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B (Methodological)*, pages 289–300, 1995.

Peter J Bickel and Aiyou Chen. A nonparametric view of network models and newman–girvan and other modularities. *Proceedings of the National Academy of Sciences*, 106(50): 21068–21073, 2009.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

Béla Bollobás. A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *European Journal of Combinatorics*, 1(4):311–316, 1980.

Irineo Cabreros, Emmanuel Abbe, and Aristotelis Tsirigos. Detecting community structures in hi-c genomic data. In *Information Science and Systems (CISS), 2016 Annual Conference on*, pages 584–589. IEEE, 2016.

Fan Chung and Linyuan Lu. The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences*, 99(25):15879–15882, 2002a.

Fan Chung and Linyuan Lu. Connected components in random graphs with given expected degree sequences. *Annals of Combinatorics*, 6(2):125–145, 2002b.

Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, 70(6):066111, 2004.

Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009.

Amin Coja-Oghlan and André Lanka. Finding planted partitions in random graphs with general degree distributions. *SIAM Journal on Discrete Mathematics*, 23(4):1682–1714, 2009.

Gabor Csárdi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

Leon Danon, Albert Diaz-Guilera, Jordi Duch, and Alex Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005 (09):P09008, 2005.

Nurcan Durak, Tamara G Kolda, Ali Pinar, and C Seshadhri. A scalable null model for directed graphs matching all degree distributions: In, out, and reciprocal. In *Network Science Workshop (NSW), 2013 IEEE 2nd*, pages 23–30. IEEE, 2013.

Ming Fan, Ka-Chun Wong, Taewoo Ryu, Timothy Ravasi, and Xin Gao. Secom: A novel hash seed and community detection based-approach for genome-scale protein domain identification. *PLoS ONE*, 7:e39475, 06 2012.

Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

Roger Guimera and Luis A Nunes Amaral. Functional cartography of complex metabolic networks. *Nature*, 433(7028):895–900, 2005.

Abigail Z Jacobs and Aaron Clauset. A unified view of generative models for networks: models, methods, opportunities, and challenges. *arXiv:1411.4070*, 2014.

David Kahle and Hadley Wickham. ggmap: Spatial visualization with ggplot2. *The R Journal*, 5(1):144–161, 2013. URL `http://journal.r-project.org/archive/2013-1/kahle-wickham.pdf`.

Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107, 2011.

Andrea Lancichinetti and Santo Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E*, 80(1):016118, 2009.

Andrea Lancichinetti, Santo Fortunato, and János Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3): 033015, 2009.

Andrea Lancichinetti, Filippo Radicchi, José J Ramasco, Santo Fortunato, et al. Finding statistically significant communities in networks. *PloS One*, 6(4):e18961, 2011.

Rocco Langone, Carlos Alzate, and Johan AK Suykens. Modularity-based model selection for kernel spectral clustering. In *The 2011 International Joint Conference on Neural Networks*, pages 1849–1856. IEEE, 2011.

Jure Leskovec et al. Stanford network analysis project, 2010. URL `http://snap.stanford.edu`.

David Lusseau and Mark EJ Newman. Identifying the role that animals play in their social networks. *Proceedings of the Royal Society of London B: Biological Sciences*, 271(Suppl 6):S477–S481, 2004.

Michael Molloy and Bruce Reed. A critical point for random graphs with a given degree sequence. *Random structures & algorithms*, 6(2-3):161–180, 1995.

Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2): 167–256, 2003.

Mark EJ Newman. Analysis of weighted networks. *Physical Review E*, 70(5):056131, 2004a.

Mark EJ Newman. Detecting community structure in networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 38(2):321–330, 2004b.

Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

Krzysztof Nowicki and Tom A B Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087, 2001.

Gergely Palla, Albert-László Barabási, and Tamás Vicsek. Quantifying social group evolution. *Nature*, 446(7136):664–667, 2007.

Tiago P Peixoto. Model selection and hypothesis testing for large-scale network models with overlapping groups. *Physical Review X*, 5(1):011033, 2015.

Tiago P Peixoto. Nonparametric weighted stochastic block models. *arXiv:1708.01432*, 2017.

John Platig, Peter Castaldi, Dawn DeMeo, and John Quackenbush. Bipartite community structure of eqtls. *arXiv:1509.02816*, 2015.

Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. *Journal of Graph Algorithms Applications*, 10(2):191–218, 2006.

Mason A Porter, Jukka-Pekka Onnela, and Peter J Mucha. Communities in networks. *Notices of the AMS*, 56(9):1082–1097, 2009.

Jörg Reichardt and Stefan Bornholdt. Clustering of sparse data via network communitiesa prototype study of a large online market. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(06):P06016, 2007.

Martin Rosvall and Carl T Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.

Shaghayegh Sahebi and William W Cohen. Community-based recommendations: a solution to the cold start problem. In *Workshop on Recommender Systems and the Social Web, RSWEB*, 2011.

Scott White and Padhraic Smyth. A spectral clustering approach to finding communities in graph. In *SDM*, volume 5, pages 76–84. SIAM, 2005.

James D Wilson, Simi Wang, Peter J Mucha, Shankar Bhamidi, Andrew B Nobel, et al. A testing based extraction algorithm for identifying significant communities in networks. *The Annals of Applied Statistics*, 8(3):1853–1891, 2014.

Jierui Xie, Boleslaw K Szymanski, and Xiaoming Liu. Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In *IEEE 11th International Conference on Data Mining*, pages 344–349. IEEE, 2011.

Jierui Xie, Stephen Kelley, and Boleslaw K Szymanski. Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Computing Surveys*, 45 (4):43, 2013.

Liu Xin, E Haihong, Junde Song, Meina Song, and Junjie Tong. Book recommendation based on community detection. In *Pervasive Computing and the Networked World*, pages 364–373. Springer, 2014.

Xiaoran Yan, Cosma Shalizi, Jacob E Jensen, Florent Krzakala, Cristopher Moore, Lenka Zdeborová, Pan Zhang, and Yaojia Zhu. Model selection for degree-corrected block models. *Journal of Statistical Mechanics: Theory and Experiment*, 2014(5):P05007, 2014.

Shihua Zhang, Rui-Sheng Wang, and Xiang-Sun Zhang. Identification of overlapping community structure in complex networks using fuzzy c-means clustering. *Physica A: Statistical Mechanics and its Applications*, 374(1):483–490, 2007.

Yunpeng Zhao, Elizaveta Levina, and Ji Zhu. Community extraction for social networks. *Proceedings of the National Academy of Sciences*, 108(18):7321–7326, 2011.

Yunpeng Zhao, Elizaveta Levina, and Ji Zhu. Consistency of community detection in networks under degree-corrected stochastic block models. *The Annals of Statistics*, pages 2266–2292, 2012.

Journal of Machine Learning Research 18 (2018) 1-41 Submitted 6/17; Revised 1/18; Published 4/18

Kernel Method for Persistence Diagrams via Kernel Embedding and Weight Factor

Genki Kusano GENKI.KUSANO.R5@DC.TOHOKU.AC.JP
Graduate School of Science
Tohoku University
Sendai, Miyagi 980-8578, Japan

Kenji Fukumizu FUKUMIZU@ISM.AC.JP
The Institute of Statistical Mathematics
Tachikawa, Tokyo 190-8562, Japan

Yasuaki Hiraoka HIRAOKA@TOHOKU.AC.JP
Advanced Institute for Materials Research
Tohoku University
Sendai, Miyagi 980-0811, Japan

Editor: Arthur Gretton

Abstract

Topological data analysis (TDA) is an emerging mathematical concept for characterizing shapes in complicated data. In TDA, persistence diagrams are widely recognized as a useful descriptor of data, distinguishing robust and noisy topological properties. This paper introduces a kernel method for persistence diagrams to develop a statistical framework in TDA. The proposed kernel is stable under perturbation of data, enables one to explicitly control the effect of persistence by a weight function, and allows an efficient and accurate approximate computation. The method is applied into practical data on granular systems, oxide glasses and proteins, showing advantages of our method compared to other relevant methods for persistence diagrams.

Keywords: topological data analysis, persistence diagrams, kernel method, kernel embedding, persistence weighted Gaussian kernel

1. Introduction

Recent years have witnessed an increasing interest in utilizing methods of algebraic topology for statistical data analysis. In terms of algebraic topology, conventional clustering methods are regarded as charactering 0-dimensional topological features which mean connected components of data. Furthermore, higher dimensional topological features also represent informative shape of data, such as rings (1-dimension) and cavities (2-dimension). The research analyzing these topological features in data is called *topological data analysis* (TDA) (Carlsson, 2009), which has been successfully applied to various areas including information science (Carlsson et al., 2008; de Silva and Ghrist, 2007), biology (Kasson et al., 2007; Xia and Wei, 2014), brain science (Lee et al., 2011; Petri et al., 2014; Singh et al., 2008), biochemistry (Gameiro et al., 2015), material science (Hiraoka et al., 2016; Nakamura et al., 2015; Saadatfar et al., 2017), and so on. In many of these applications, data have complicated geometric structures, and thus it is important to extract informative topological features from the data.

A *persistent homology* (Edelsbrunner et al., 2002), which is a key mathematical tool in TDA, extracts robust topological information from data, and it has a compact expression called a *persistence diagram*. While it is applied to various problems such as the ones listed above, statistical or machine learning methods for analysis on persistence diagrams are still limited. In TDA, analysts often elaborate only single persistence diagram and, in particular, methods for handling many persistence diagrams, which can contain randomness from the data, are at the beginning stage (see the end of this section for related works). Hence, developing a framework of statistical data analysis on persistence diagrams is a significant issue for further success of TDA and, to this goal, this paper discusses kernel methods for persistence diagrams.

1.1 Topological Descriptor

In order to provide some intuitions for the persistent homology, let us consider a typical way of constructing persistent homology from data points in a Euclidean space, assuming that the point set lies on a submanifold. The aim is to make inference on the topology of the underlying manifold from finite data points. We consider the r-balls (balls with radius r) to recover the topology of the manifold, as popularly employed in constructing an r-neighbor graph in many manifold learning algorithms. While it is expected that, with an appropriate choice of r, the r-ball model can represent the underlying topological structures of the manifold, it is also known that the result is sensitive to the choice of r. If r is too small (resp. large), the union of r-balls consists simply of the disjoint r-balls (resp. a contractible space). Then, by considering not one specific r but all r, the persistent homology gives robust topological features of the point set.

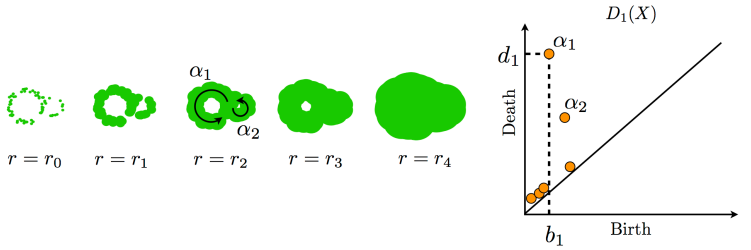


Figure 1: Unions of r-balls at data points (left) and its 1-st persistence diagram (right). The point (b_1, d_1) in the persistence diagram represents the ring α_1, which appears at $r = b_1$ and disappears at $r = d_1$. The noisy rings are plotted as the points close to the diagonal.

As a useful representation of persistent homology, a persistence diagram is often used in topological data analysis. The persistence diagram is given in the form of a multiset $D = \{(b_i, d_i) \in \mathbb{R}^2 \mid i \in I,\ b_i < d_i\}$ (Figure 1). Every point $(b_i, d_i) \in D$, called a *generator* of the persistent homology, represents a topological property (e.g., connected components, rings, cavities, etc.) which appears at $r = b_i$ and disappears at $r = d_i$ in the ball model. Then, the *persistence* $d_i - b_i$ of the generator shows the robustness of the topological property under the radius parameter. A generator with large persistence can be regarded as a reliable structure, while that with small persistence (points close to the diagonal) is likely to be a structure caused by noise. In this way, persistence diagrams encode topological and geometric information of data points.

1.2 Contributions

Since a persistence diagram is a point set of variable size, it is not straightforward to apply standard methods of statistical data analysis, which typically assume vectorial data. To vectorize persistence diagrams, we employ the framework of kernel embedding of (probability and more general) measures into reproducing kernel Hilbert spaces (RKHS). This framework has recently been developed and leading various new methods for nonparametric inference (Muandet et al., 2017; Smola et al., 2007; Song et al., 2013; Lopez-Paz et al., 2015; Szabó et al., 2016). It is known (Sriperumbudur et al., 2011) that, with an appropriate choice of kernels, a signed Radon measure can be uniquely represented by the Bochner integral of the feature vectors with respect to the measure. Since a persistence diagram can be regarded as a sum of Dirac delta measures, it can be embedded into an RKHS by the Bochner integral. Once such a vector representation is obtained, we can introduce any kernel methods for persistence diagrams systematically (see Figure 2).

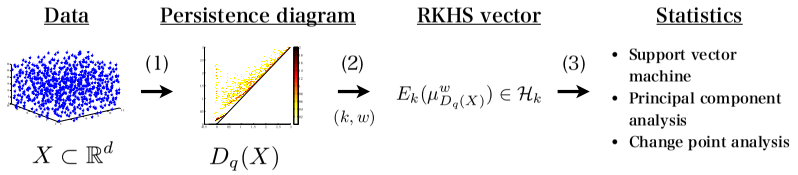


Figure 2: (1) A data set X is transformed into a persistence diagram $D_q(X)$ (Section 2.1). (2) The persistence diagram $D_q(X)$ is mapped to an RKHS vector $E_k(\mu^w_{D_q(X)})$, where k is a positive definite kernel and w is a weight function controlling the effect of persistence (Section 3.1). (3) Statistical methods are applied to those vector representations of persistence diagrams (Section 4).

Furthermore, since each generator in a persistence diagram is equipped with a persistence which indicates a robustness of the topological features, we will utilize it as a weight on the generator. For embedding persistence diagrams in an RKHS, we propose a useful class of positive definite kernels, called *persistence weighted Gaussian kernel* (PWGK). The

advantages of this kernel are as follows: (i) We can explicitly control the effect of persistence by a weight function, and hence discount the noisy generators appropriately in statistical analysis. (ii) As a theoretical contribution, the distance defined by the RKHS norm for the PWGK satisfies the stability property, which ensures the continuity from data to the vector representation of the persistence diagram. (iii) The PWGK allows efficient computation by using the random Fourier features (Rahimi and Recht, 2007), and thus it is applicable to persistence diagrams with a large number of generators.

We demonstrate the performance of the proposed kernel method with synthesized and real-world data, including granular systems (taken by X-ray Computed Tomography on granular experiments), oxide glasses (taken by molecular dynamics simulations) and protein data set (taken by NMR and X-ray crystallography experiments). We remark that these real-world problems have physical and biochemical significance in their own right, as detailed in Section 4.

1.3 Related Works

There are already some relevant works on statistical approaches to persistence diagrams. Some studies discuss how to transform a persistence diagram to a vector (Bubenik, 2015; Reininghaus et al., 2015; Adams et al., 2017; Cang et al., 2015; Robins and Turner, 2016; Donatini et al., 1998). In these methods, a transformed vector is typically expressed in a Euclidean space $\mathbb{R}^k$ or a function space L^p, and simple and ad-hoc summary statistics like means and variances are used for data analysis such as principal component analysis (PCA) and support vector machines (SVMs). In this paper, we will compare the performance among the PWGK, the persistence scale-space kernel (Reininghaus et al., 2015), the persistence landscape (Bubenik, 2015), the persistence image (Adams et al., 2017), and the molecular topological fingerprint (Cang et al., 2015) in several machine learning tasks. Furthermore, we show that our vectorization is a generalization of the persistence scale-space kernel and the persistence image although the constructions are different. We also remark that there are some works discussing statistical properties of persistence diagrams for random data points: Chazal et al. (2015) show convergence rates of persistence diagram estimation, and Fasy et al. (2014) discuss confidence sets in a persistence diagram. These works consider a different but important direction to the statistical methods for persistence diagrams.

The remaining of this paper is organized as follows: In Section 2, we review some basics on persistence diagrams and kernel embedding methods. In Section 3, the PWGK is proposed, and some theoretical and computational issues are discussed. Section 4 shows experimental results and compares the proposed kernel method with other methods.

This paper is an extended version of our ICML paper (Kusano et al., 2016). The difference from this conference version is as follows: (i) Comparisons with other relevant methods, in particular, persistence landscapes and persistence images, have been added to this version. (ii) New experimental results in comparison with other relevant methods. (iii) The theoretical results have been generalized to a class of kernels that satisfy the assumption (K) and weight functions that satisfy assumption (W1) (Propositions 8 and 10). The proofs have been modified accordingly, while the basic line of ideas are the same.

2. Backgrounds

We review the concepts of persistence diagrams and kernel methods. For readers who are not familiar with algebraic topology, we give a brief summary in Appendix A. See also Hatcher (2002) as an accessible introduction to algebraic topology.

2.1 Persistence Diagram

In order to define a persistence diagram, we transform a data set X into a filtration $\mathbb{F}\mathrm{ilt}(X)$ and compute its persistent homology $H_q(\mathbb{F}\mathrm{ilt}(X))$. In this section, we will first introduce this mathematical framework of persistence diagrams. Then, by using a ball model filtration, we will intuitively explain geometrical meanings of persistence diagrams. The ball model filtrations can be generalized toward two constructions using Čech complexes and sub-level sets. The former construction is useful for computations of persistence diagrams and the latter is useful to discuss theoretical properties.

2.1.1 Mathematical Framework of Persistence Diagrams

Let K be a coefficient field of homology.[1] Let $\mathbb{F}\mathrm{ilt} = \{F_a \mid a \in \mathbb{R}\}$ be a (right continuous) *filtration* of simplicial complexes, i.e., F_a is a subcomplex of F_b for $a \leq b$ and $F_a = \bigcap_{a<b} F_b$. Alternatively, $\mathbb{F}\mathrm{ilt}$ may be a filtration of topological spaces: in this case F_a is a subspace of F_b with the same condition as above. For $a \leq b$, the K-linear map induced from the inclusion $F_a \hookrightarrow F_b$ is denoted by $\rho_a^b : H_q(F_a) \to H_q(F_b)$, where $H_q(F_a)$ is the q-th homology of F_a. The q-th *persistent homology* $H_q(\mathbb{F}\mathrm{ilt}) = (H_q(F_a), \rho_a^b)$ of $\mathbb{F}\mathrm{ilt}$ is defined by the family of homology $\{H_q(F_a) \mid a \in \mathbb{R}\}$ and the induced linear maps $\{\rho_a^b \mid a \leq b\}$.

A *homological critical value* of $H_q(\mathbb{F}\mathrm{ilt})$ is the number $a \in \mathbb{R}$ such that the linear map $\rho_{a-\varepsilon}^{a+\varepsilon} : H_q(F_{a-\varepsilon}) \to H_q(F_{a+\varepsilon})$ is not isomorphic for any sufficiently small $\varepsilon > 0$. The persistent homology $H_q(\mathbb{F}\mathrm{ilt})$ is called *tame* if $\dim_K H_q(F_a) < \infty$ for any $a \in \mathbb{R}$ and the number of homological critical values is finite. A tame persistent homology $H_q(\mathbb{F}\mathrm{ilt})$ has a nice decomposition property:

Theorem 1 (Zomorodian and Carlsson (2005)) *A tame persistent homology can be uniquely expressed by*

$$H_q(\mathbb{F}\mathrm{ilt}) \cong \bigoplus_{i \in I} \mathbb{I}[b_i, d_i], \tag{1}$$

where $\mathbb{I}[b_i, d_i] = (U_a, \iota_a^b)$ *consists of a family of* K*-vector spaces*

$$U_a = \begin{cases} K, & b_i \leq a < d_i \\ 0, & \text{otherwise} \end{cases},$$

and $\iota_a^b = \mathrm{id}_K$ *for* $b_i \leq a \leq b < d_i$.[2]

1. In this setting, all homology are K-vector spaces. You may simply consider the case $K = \mathbb{R}$, but the theory is built with an arbitrary field.
2. To be more precise, a persistent homology can be seen as an object of the functor category from the poset category defined by $(\mathbb{R}, \leq)$ to the category of finite dimensional vector spaces. The symbols $\cong$ and $\oplus$ represent the isomorphism and the direct sum in the functor category. It is far beyond the scope of this paper to provide precise definitions of these notions. Interested readers can see Bubenik and Scott (2014) for more details.

Each summand $\mathbb{I}[b_i, d_i]$ means a topological feature in $\mathbb{F}\mathrm{ilt}$ that appears at $a = b_i$ and disappears at $a = d_i$. The birth-death pair $x = (b_i, d_i)$ is called a *generator* of the persistent homology, and $\mathrm{pers}(x) := d_i - b_i$ a *persistence* of x. We note that, when $\dim_K H_q(F_a) \neq 0$ for any $a < 0$ (resp. for any $a > 0$), the decomposition (1) should be understood in the sense that some b_i takes the value $-\infty$ (resp. $d_i = \infty$), where $-\infty, \infty$ are the elements in the extended real $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$. Through the decomposition in Theorem 1, a persistent homology $H_q(\mathbb{F}\mathrm{ilt})$, which is an algebraic object and is not suitable to be analyzed by statistical methods, is transformed into a multi-set of 2-dimensional vectors

$$D_q(\mathbb{F}\mathrm{ilt}) = \left\{ (b_i, d_i) \in \overline{\mathbb{R}}^2 \;\middle|\; i \in I \right\}$$

and we call it the *persistence diagram* of $\mathbb{F}\mathrm{ilt}$.[3]

In this paper, we assume that all persistence diagrams have finite cardinality because a tame persistent homology defines a finite persistence diagram. Moreover, we also assume that all birth-death pairs are bounded, that is, all elements in a persistence diagram take neither ∞ nor $-\infty$.[4] In the following, we also use abstract persistence diagrams (denoted by D or E) given by finite multi-sets above the diagonal $\mathbb{R}^2_{\mathrm{ad}} := \{(b, d) \in \mathbb{R}^2 \mid b < d\}$.

2.1.2 Ball Model Filtrations

The example used in Figure 1 can be expressed as follows. Let $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ be a finite subset in a metric space (M, d_M) and $X_a := \bigcup_{i=1}^n B(\boldsymbol{x}_i; a)$ be a union of balls $B(\boldsymbol{x}_i; a) = \{\boldsymbol{x} \in M \mid d_M(\boldsymbol{x}_i, \boldsymbol{x}) \leq a\}$ with radius $a \geq 0$. For convenience, let $X_a := \emptyset$ $(a < 0)$. Since $\mathbb{X} = \{X_a \mid a \in \mathbb{R}\}$ is a right-continuous filtration of topological spaces and X is a finite set, $H_q(\mathbb{X})$ is tame and the persistence diagram $D_q(\mathbb{X})$ is well-defined. For notational simplicity, the persistence diagram of this ball model filtration is denoted by $D_q(X)$.

We remark that, in this model, there is only one generator in $D_0(X)$ that does not disappear in the filtration; its lifetime is ∞. From now on, we deal with $D_0(X)$ by removing this infinite lifetime generator.[5] Let $\mathrm{diam}(X)$ be the diameter of X defined by $\max_{\boldsymbol{x}_i, \boldsymbol{x}_j \in X} d_M(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Then, all generators appear after $a = 0$ and disappear before $a = \mathrm{diam}(X)$ because $X_{\mathrm{diam}(X)}$ becomes a contractible space. Thus, for any dimension q, all birth-death pairs of $D_q(X)$ have finite values.

2.1.3 Geometric Complexes

We review some standard methods of constructing a filtration from finite sets in a metric space. See also Chazal et al. (2014) for more details.

Let (M, d_M) be a metric space and $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ be a finite subset in M. For a fixed $a \geq 0$, we form a q-simplex $[\boldsymbol{x}_{i_0} \cdots \boldsymbol{x}_{i_q}]$ as a subset $\{\boldsymbol{x}_{i_0}, \ldots, \boldsymbol{x}_{i_q}\}$ of X whenever there exists $\bar{\boldsymbol{x}} \in M$ such that $d_M(\boldsymbol{x}_{i_j}, \bar{\boldsymbol{x}}) \leq a$ for all $j = 0, \ldots, q$, or equivalently, $\cap_{j=0}^q B(\boldsymbol{x}_{i_j}; a) \neq \emptyset$. The set of these simplices forms a simplicial complex, called the *Čech complex* of X with parameter a, denoted by $\check{\mathrm{C}}\mathrm{ech}(X; a)$. For $a < 0$, we define $\check{\mathrm{C}}\mathrm{ech}(X; a)$ as an empty set.

3. A *multi-set* is a set with multiplicity of each point. We regard a persistence diagram as a multi-set, since several generators can have the same birth-death pairs.
4. This assumption will be justified in Section 2.1.2.
5. This is called the *reduced persistence diagram*.

Since there is a natural inclusion $\check{\mathrm{C}}\mathrm{ech}(X;a) \hookrightarrow \check{\mathrm{C}}\mathrm{ech}(X;b)$ whenever $a \leq b$, $\check{\mathrm{C}}\mathrm{ech}(X) = \left\{\check{\mathrm{C}}\mathrm{ech}(X;a) \;\middle|\; a \in \mathbb{R}\right\}$ is a filtration. When M is a subspace of $\mathbb{R}^d$, from the nerve lemma (Hatcher, 2002), it is known that the topology of $\check{\mathrm{C}}\mathrm{ech}(X;a)$ is the same as X_a (Figure 3)[6], and hence $D_q(\check{\mathbb{C}}\mathrm{ech}(X)) = D_q(X)$.

As a similar concept to the Čech complex, the Rips complex (or Vietoris-Rips complex) is also often used in TDA. While the Rips complex gives different topology from the Čech complex, it can be computed much more efficiently; the Rips complex needs only pairwise distances, while the Čech complex needs all the $(q+1)$-combinations among n points for q-th homology, which easily becomes infeasible for large n. For a fixed $a \geq 0$, we form a q-simplex $[\boldsymbol{x}_{i_0} \cdots \boldsymbol{x}_{i_q}]$ as a subset $\left\{\boldsymbol{x}_{i_0}, \ldots, \boldsymbol{x}_{i_q}\right\}$ of X that satisfies $d_M(\boldsymbol{x}_{i_j}, \boldsymbol{x}_{i_k}) \leq 2a$ for all $j, k = 0, \ldots, q$. The set of these simplices forms a simplicial complex, called the *Rips complex* of X with parameter a, denoted by $\mathrm{Rips}(X;a)$. Similarly, the Rips complex also forms a filtration $\mathbb{R}\mathrm{ips}(X)$. In general, $D_q(\mathbb{R}\mathrm{ips}(X))$ is not the same as $D_q(X)$ (see Figure 3). In experiments in this paper, all persistence diagrams are computed by a ball model filtration, which is equivalent to the Čech complex filtration, and we do not use the Rips complex filtration. We remark that, however, there are also applications of Rips complexes (e.g., sensor networks (de Silva and Ghrist, 2007)), and our kernel method and stability results shown in Proposition 8 and Proposition 10 can be applied not only the ball model filtration but also any filtrations including the Rips complex filtration.

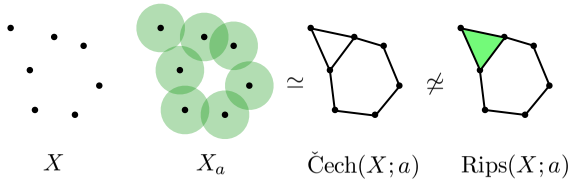


Figure 3: A point set X, the union of balls X_a, the Čech complex $\check{\mathrm{C}}\mathrm{ech}(X;a)$ and the Rips complex $\mathrm{Rips}(X;a)$. There are two rings in X_a and $\check{\mathrm{C}}\mathrm{ech}(X;a)$. However, $\mathrm{Rips}(X;a)$ has only one ring because there is a 2-simplex.

2.1.4 Sub-level Sets

Another popular way of constructing a filtration is to use sub-level sets. This is useful when data is given in the from of a function like a gray-scale image on a two dimensional region. Let M be a topological space and $f : M \to \mathbb{R}$ be a continuous map. Then, we define a *sub-level set* by $\mathrm{Sub}(f;a) := f^{-1}((-\infty, a])$ for $a \in \mathbb{R}$ and its filtration by $\mathbb{S}\mathrm{ub}(f) := \{\mathrm{Sub}(f;a) \mid a \in \mathbb{R}\}$. Here, $f : M \to \mathbb{R}$ is said to be *tame* if $H_q(\mathbb{S}\mathrm{ub}(f))$ is tame.

6. Precisely, they are *homotopy equivalent*.

For a finite set $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ in a metric space (M, d_M), we define the distance function $\mathrm{dist}_X : M \to \mathbb{R}$ by

$$\mathrm{dist}_X(\boldsymbol{x}) := \min_{\boldsymbol{x}_i \in X} d_M(\boldsymbol{x}, \boldsymbol{x}_i).$$

Then, we can see $\mathrm{Sub}(\mathrm{dist}_X; a) = \bigcup_{\boldsymbol{x}_i \in X} B(\boldsymbol{x}_i; a)$ and $D_q(\mathbb{S}\mathrm{ub}(\mathrm{dist}_X)) = D_q(X)$. This means that the ball model is a special case of the sub-level set, and the Čech complex and the sub-level set with the distance function dist_X give the same persistence diagram.

2.2 Stability of Persistence Diagrams

When we consider data analysis based on persistence diagrams, it is useful to introduce a distance measure among persistence diagrams for describing their relations. In introducing a distance measure, it is desirable that, as a representation of data, the mapping from data to a persistence diagram is continuous with respect to the distance. In many cases, data involve noise or stochasticity, and thus the persistence diagrams should be stable under perturbation of data.

The *bottleneck distance* d_{W_∞} between two persistence diagrams D and E is defined by

$$d_{\mathrm{W}_\infty}(D, E) := \inf_{\gamma} \sup_{x \in D \cup \Delta} \|x - \gamma(x)\|_\infty ,$$

where $\Delta := \{(a, a) \mid a \in \mathbb{R}\}$ is the diagonal set with infinite multiplicity and γ ranges over all multi-bijections from $D \cup \Delta$ to $E \cup \Delta$.[7] Here, for $z = (z_1, z_2) \in \mathbb{R}^2$, $\|z\|_\infty$ denotes $\max\{|z_1|, |z_2|\}$. We note that there always exists such a multi-bijection γ because the cardinalities of $D \cup \Delta$ and $E \cup \Delta$ are equal by considering the diagonal set Δ with infinite multiplicity. For sets X and Y in a metric space (M, d_M), let us recall the *Hausdorff distance* d_{H} given by

$$d_{\mathrm{H}}(X, Y) := \max\left\{ \sup_{\boldsymbol{x} \in X} \inf_{\boldsymbol{y} \in Y} d_M(\boldsymbol{x}, \boldsymbol{y}), \sup_{\boldsymbol{y} \in Y} \inf_{\boldsymbol{x} \in X} d_M(\boldsymbol{x}, \boldsymbol{y}) \right\}.$$

Then, the bottleneck distance satisfies the following stability property.

Proposition 2 (Chazal et al. (2014); Cohen-Steiner et al. (2007)) *Let X and Y be finite subsets in a metric space (M, d_M). Then the persistence diagrams satisfy*

$$d_{\mathrm{W}_\infty}(D_q(X), D_q(Y)) \leq d_{\mathrm{H}}(X, Y).$$

Proposition 2 provides a geometric intuition of the stability of persistence diagrams. Assume that two point sets X and Y are close to each other with $\varepsilon = d_{\mathrm{H}}(X, Y)$. If there is a generator $(b, d) \in D_q(Y)$, then we can find at least one generator in X which is born in $(b - \varepsilon, b + \varepsilon)$ and dies in $(d - \varepsilon, d + \varepsilon)$ (see Figure 4). Thus, the stability guarantees the similarity of two persistence diagrams, and hence we can infer the true topological features from the persistence diagrams given by noisy observation (see also Fasy et al. (2014)).

7. A *multi-bijection* is a bijective map between two multi-sets counted with their multiplicity.

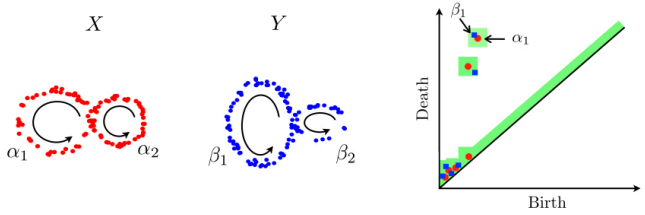


Figure 4: Two point sets X and Y (left) and their persistence diagrams (right). The green region is an ε-neighborhood of $D_q(Y)$ and all generators in $D_q(X)$ are in the ε-neighborhood.

For $1 \leq p < \infty$, the *p-Wasserstein distance* d_{W_p}, which is also used as a distance between two persistence diagrams D and E, is defined by

$$d_{\mathrm{W}_p}(D, E) = \inf_{\gamma} \left(\sum_{x \in D \cup \Delta} \|x - \gamma(x)\|_\infty^p \right)^{\frac{1}{p}},$$

where γ ranges over all multi-bijections from $D \cup \Delta$ to $E \cup \Delta$. Here, we define the *degree-p total persistence* of D by $\mathrm{Pers}_p(D) := \sum_{x \in D} \mathrm{pers}(x)^p$ for $1 \leq p < \infty$.

Proposition 3 (Cohen-Steiner et al. (2010)) *Let $1 \leq p' \leq p < \infty$, and D and E be persistence diagrams whose degree-p' total persistences are bounded from above. Then,*

$$d_{\mathrm{W}_p}(D, E) \leq \left(\frac{\mathrm{Pers}_{p'}(D) + \mathrm{Pers}_{p'}(E)}{2} \right)^{\frac{1}{p}} d_{\mathrm{W}_\infty}(D, E)^{1 - \frac{p'}{p}}.$$

For a persistence diagram D, its degree-p total persistence is bounded from above by $\mathrm{card}\,(D) \times \max_{x \in D} \mathrm{pers}(x)^p$, where $\mathrm{card}\,(D)$ denotes the number of generators in D. However, this bound may be weak because, in general, $\mathrm{card}\,(D)$ cannot be bounded from above. In particular, if data set has noise, the persistence diagram often has many generators close to the diagonal. Thus, it is desirable that the total persistence is bounded from above independently of $\mathrm{card}\,(D)$. In the case of persistence diagrams obtained from a ball model filtration, we have the following upper bound (see Appendix B for the proof):

Lemma 4 *Let M be a triangulable compact subspace in $\mathbb{R}^d$, X be a finite subset of M, and $p > d$. Then,*

$$\mathrm{Pers}_p(D_q(X)) \leq \frac{p}{p-d} C_M \mathrm{diam}(M)^{p-d},$$

where C_M is a constant depending only on M.

Hence, we have the following by combining Proposition 3 and Lemma 4.

Corollary 5 *Let M be a triangulable compact subspace in $\mathbb{R}^d$, X, Y be finite subsets of M, and $p \geq p' > d$. Then*

$$\begin{aligned} d_{\mathrm{W}_p}(D_q(X), D_q(Y)) &\leq \left(\frac{p'}{p'-d} C_M \mathrm{diam}(M)^{p'-d} \right)^{\frac{1}{p}} d_{\mathrm{W}_\infty}(D_q(X), D_q(Y))^{1 - \frac{p'}{p}} \\ &\leq \left(\frac{p'}{p'-d} C_M \mathrm{diam}(M)^{p'-d} \right)^{\frac{1}{p}} d_{\mathrm{H}}(X, Y)^{1 - \frac{p'}{p}} \end{aligned}$$

where C_M is a constant depending only on M.

2.3 Kernel Methods for Representing Signed Measures

As a preliminary to our proposal of vector representation for persistence diagrams, we briefly summarize a method for embedding signed measures with a positive definite kernel.

Let Ω be a set and $k : \Omega \times \Omega \to \mathbb{R}$ be a *positive definite kernel* on Ω, i.e., k is symmetric, and for any number of points $x_1, \ldots, x_n$ in Ω, the Gram matrix $(k(x_i, x_j))_{i,j=1,\ldots,n}$ is nonnegative definite. A popular example of positive definite kernel on $\mathbb{R}^d$ is the Gaussian kernel $k_{\mathrm{G}}(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}$ $(\sigma > 0)$, where $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^d$. From the Moore-Aronszajn theorem, it is also known that every positive definite kernel k on Ω is uniquely associated with a reproducing kernel Hilbert space $\mathcal{H}_k$ (RKHS).

We use a positive definite kernel to represent persistence diagrams by following the idea of the kernel mean embedding of distributions (Muandet et al., 2017; Smola et al., 2007; Sriperumbudur et al., 2011). Let Ω be a locally compact Hausdorff space, $M_{\mathrm{b}}(\Omega)$ be the space of all finite signed Radon measures on Ω, and k be a bounded measurable kernel on Ω.[8] Since $\int \|k(\cdot, x)\|_{\mathcal{H}_k} d\mu(x)$ is finite, the integral $\int k(\cdot, x) d\mu(x)$ is well-defined as the Bochner integral (Diestel and Uhl Jr, 1977). Here, we define a mapping from $M_{\mathrm{b}}(\Omega)$ to $\mathcal{H}_k$ by

$$E_k : M_{\mathrm{b}}(\Omega) \to \mathcal{H}_k, \quad \mu \mapsto \int k(\cdot, x) d\mu(x). \tag{2}$$

For a locally compact Hausdorff space Ω, let $C_0(\Omega)$ denote the space of continuous functions vanishing at infinity.[9] A kernel k on Ω is said to be C_0-kernel if $k(\cdot, x) \in C_0(\Omega)$ for any $x \in \Omega$. If k is C_0-kernel, the associated RKHS $\mathcal{H}_k$ is a subspace of $C_0(\Omega)$. A C_0-kernel k is called *C_0-universal* if $\mathcal{H}_k$ is dense in $C_0(\Omega)$. It is known that the Gaussian kernel k_{G} is C_0-universal on $\mathbb{R}^d$ (Sriperumbudur et al., 2011). When k is C_0-universal, the vector $E_k(\mu)$ in the RKHS uniquely determines the finite signed measure μ, and thus serves as a representation of μ. We summarize the property as follows:

Proposition 6 (Sriperumbudur et al. (2011)) *Let Ω be a locally compact Hausdorff space. If k is C_0-universal on Ω, the mapping E_k is injective. Thus,*

$$d_k(\mu, \nu) = \|E_k(\mu) - E_k(\nu)\|_{\mathcal{H}_k}$$

defines a distance on $M_{\mathrm{b}}(\Omega)$.

8. A *Radon measure* μ on Ω is a Borel measure on Ω satisfying (i) $\mu(C) < \infty$ for any compact subset $C \subset \Omega$, and (ii) $\mu(B) = \sup\{\mu(C) \mid C \subset B,\ C\text{:compact}\}$ for any B in the Borel σ-algebra of Ω.
9. A function f is said to *vanish at infinity* if for any $\varepsilon > 0$ there is a compact set $K \subset \Omega$ such that $\sup_{x \in K^c} |f(x)| \leq \varepsilon$.

3. Kernel Methods for Persistence Diagrams

We propose a positive definite kernel for persistence diagrams, called the persistence weighted Gaussian kernel (PWGK), to embed the persistence diagrams into an RKHS. This vectorization of persistence diagrams enables us to apply any kernel methods to persistence diagrams and explicitly control the effect of persistence. We show the stability theorem with respect to the distance defined by the embedding and discuss the efficient and precise approximate computation of the PWGK.

3.1 Vectorization of Persistence Diagrams

We propose a method for vectorizing persistence diagrams using the kernel embedding (2) by regarding a persistence diagram as a discrete measure. In vectorizing persistence diagrams, it is desirable to have flexibility to discount the effect of generators close to the diagonal, since they often tend to be caused by noise. To this goal, we explain slightly different two ways of embeddings, which turn out to give the same inner product for two persistence diagrams.

First, for a persistence diagram D, we introduce a measure $\mu_D^w := \sum_{x\in D} w(x)\delta_x$ with a weight $w(x) > 0$ for each generator $x \in D$ (Figure 5), where δ_x is the Dirac delta measure at x. By appropriately choosing $w(x)$, the measure μ_D^w can discount the effect of generators close to the diagonal. A concrete choice of $w(x)$ will be discussed later.

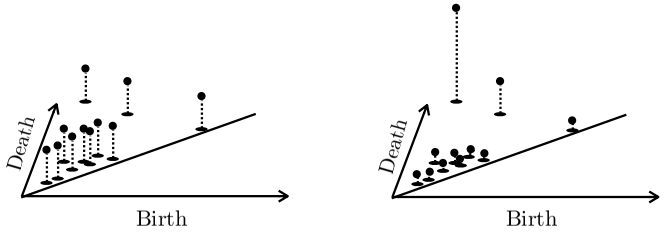


Figure 5: Unweighted (left) and weighted (right) measures.

As discussed in Section 2.3, given a C_0-universal kernel k above the diagonal $\mathbb{R}^2_{\mathrm{ad}} = \{(b,d) \in \mathbb{R}^2 \mid b < d\}$, the measure μ_D^w can be embedded as an element of the RKHS $\mathcal{H}_k$ via

$$\mu_D^w \mapsto E_k(\mu_D^w) := \sum_{x\in D} w(x)k(\cdot, x). \tag{3}$$

From the injectivity in Proposition 6, this mapping identifies a persistence diagram; in other words, it does not lose any information about persistence diagrams. Hence, $E_k(\mu_D^w) \in \mathcal{H}_k$ serves as a vector representation of the persistence diagram.

As the second construction, let

$$k^w(x,y) := w(x)w(y)k(x,y)$$

be the weighted kernel with the same weight function as above.[10] Then the mapping

$$E_{k^w} : \mu_D \mapsto \sum_{x\in D} w(x)w(\cdot)k(\cdot, x) \in \mathcal{H}_{k^w} \tag{4}$$

also defines a vectorization of persistence diagrams. The first construction may be more intuitive by directly weighting a measure, while the second one is also practically useful since all the parameter tuning is reduced to kernel choice. We note that the inner products introduced by two RKHS vectors (3) and (4) are the same:

$$\langle E_k(\mu_D^w), E_k(\mu_E^w)\rangle_{\mathcal{H}_k} = \langle E_{k^w}(\mu_D), E_{k^w}(\mu_E)\rangle_{\mathcal{H}_{k^w}}.$$

In addition, these two RKHS vectors (3) and (4) are essentially equivalent, as seen from the next proposition:

Proposition 7 *Let k be C_0-universal on $\mathbb{R}^2_{\mathrm{ad}}$ and w be a positive function on $\mathbb{R}^2_{\mathrm{ad}}$. Then the following mapping*

$$\mathcal{H}_k \to \mathcal{H}_{k^w}, \quad f \mapsto wf$$

defines an isomorphism between the RKHSs. Under this isomorphism, $E_k(\mu_D^w)$ and $E_{k^w}(\mu_D)$ are identified.

Proof Let $\tilde{\mathcal{H}} := \{wf : \mathbb{R}^2_{\mathrm{ad}} \to \mathbb{R} \mid f \in \mathcal{H}_k\}$ and define its inner product by

$$\langle wf, wg\rangle_{\tilde{\mathcal{H}}} := \langle f, g\rangle_{\mathcal{H}_k}.$$

Then, it is easy to see that $\tilde{\mathcal{H}}$ is a Hilbert space and the mapping $f \mapsto wf$ gives an isomorphism between $\tilde{\mathcal{H}}$ and $\mathcal{H}_k$ of the Hilbert spaces. In fact, we can show that $\tilde{\mathcal{H}}$ is the same as $\mathcal{H}_{k^w}$. To see this, it is sufficient to check that k^w is a reproducing kernel of $\tilde{\mathcal{H}}$ from the uniqueness property of a reproducing kernel for an RKHS. The reproducing property is proven from

$$\langle wf, k^w(\cdot, x)\rangle_{\tilde{\mathcal{H}}} = \langle f, w(x)k(\cdot, x)\rangle_{\mathcal{H}_k} = w(x)f(x) = (wf)(x).$$

The second assertion is obvious. ■

3.2 Stability with respect to the Kernel Embedding

Given a data set X, we compute the persistence diagram $D_q(X)$ and vectorize it as an element $E_k(\mu^w_{D_q(X)})$ of the RKHS. Then, for practical applications, this map $X \mapsto E_k(\mu^w_{D_q(X)})$ should be stable with respect to perturbations to the data as discussed in Section 2.2.

Let D and E be persistence diagrams and $\gamma : D \cup \Delta \to E \cup \Delta$ be any multi-bijection. Here, we partition D (resp. Δ) into D_1 and D_2 (resp. Δ_1 and Δ_2) such as

$$\gamma(D_1) \subset \mathbb{R}^2_{\mathrm{ad}},\ \gamma(D_2) \subset \Delta,\ \gamma(\Delta_1) \subset \mathbb{R}^2_{\mathrm{ad}},\ \gamma(\Delta_2) \subset \Delta.$$

10. From the facts that the product of positive definite kernels are also a positive definite kernel and $f(x,y) = w(x)w(y)$ is a positive definite kernel, k^w is actually a positive definite kernel.

Then $D_1 \cup \Delta_1$ and E are bijective under γ. Now, let a weight function w be zero on the diagonal Δ. Then, the norm of the difference between RKHS vectors is calculated as follows:

$$\begin{aligned}
&\|E_k(\mu_D^w) - E_k(\mu_E^w)\|_{\mathcal{H}_k} \\
&= \left\| \sum_{x\in D} w(x)k(\cdot,x) - \sum_{y\in E} w(y)k(\cdot,y) \right\|_{\mathcal{H}_k} \\
&= \left\| \sum_{x\in D} w(x)k(\cdot,x) - \sum_{x\in D_1\cup\Delta_1} w(\gamma(x))k(\cdot,\gamma(x)) \right\|_{\mathcal{H}_k} \\
&= \left\| \sum_{x\in D\cup\Delta_1} \Big(w(x)k(\cdot,x) - w(\gamma(x))k(\cdot,\gamma(x)) \Big) + \sum_{x\in D_2} w(\gamma(x))k(\cdot,\gamma(x)) \right\|_{\mathcal{H}_k} \\
&= \left\| \sum_{x\in D\cup\Delta_1} \Big(w(x)k(\cdot,x) - w(\gamma(x))k(\cdot,\gamma(x)) \Big) \right\|_{\mathcal{H}_k} \\
&= \left\| \sum_{x\in D} w(x) \Big(k(\cdot,x) - k(\cdot,\gamma(x)) \Big) + \sum_{x\in D\cup\Delta_1} \Big(w(x) - w(\gamma(x)) \Big) k(\cdot,\gamma(x)) \right\|_{\mathcal{H}_k} \\
&\le \sum_{x\in D} w(x) \|k(\cdot,x) - k(\cdot,\gamma(x))\|_{\mathcal{H}_k} + \sum_{x\in D\cup\Delta_1} |w(x) - w(\gamma(x))| \, \|k(\cdot,\gamma(x))\|_{\mathcal{H}_k} .
\end{aligned}$$

Here, let k be a C_0-universal kernel and satisfy the following:

(K) There exist constants $B_k, L_k > 0$ such that

$$\|k(\cdot,x)\|_{\mathcal{H}_k} \le B_k, \quad \|k(\cdot,x) - k(\cdot,y)\|_{\mathcal{H}_k} \le L_k \|x-y\|_\infty \quad (x,y\in\mathbb{R}^2).$$

Then, we have

$$\|E_k(\mu_D^w) - E_k(\mu_E^w)\|_{\mathcal{H}_k} \le L_k \sum_{x\in D} w(x) \|x-\gamma(x)\|_\infty + B_k \sum_{x\in D\cup\Delta_1} |w(x) - w(\gamma(x))| . \quad (5)$$

In this sequel, we consider the Gaussian kernel $k_{\mathrm{G}}(x,y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}$ $(\sigma > 0)$ for a C_0-universal kernel satisfying (K) by $B_{k_{\mathrm{G}}} = 1$ and $L_{k_{\mathrm{G}}} = \frac{\sqrt{2}}{\sigma}$ (Lemma 16 in Appendix C). Note that the Laplace kernel $k(x,y) = e^{-\alpha\sum_i |x_i - y_i|}$ $(\alpha > 0)$ also satisfies (K) by $B_k = 1$ and $L_k = 4\alpha$.

For a weight function, we consider the following assumption:

(W1) For any persistence diagrams D and E, and any multi-bijection $\gamma : D\cup\Delta \to E\cup\Delta$, there exist constants $B_1, L_1 > 0$ such that

$$\sum_{x\in D} |w(x)| \le B_1, \quad \sum_{x\in D\cup\Delta} |w(x) - w(\gamma(x))| \le L_1 \sup_{x\in D\cup\Delta} \|x-\gamma(x)\|_\infty . \quad (6)$$

If the weight function w satisfies (W1), from Equation (5), we have

$$\|E_k(\mu_D^w) - E_k(\mu_E^w)\|_{\mathcal{H}_k} \le (L_k B_1 + B_k L_1) \sup_{x\in D\cup\Delta} \|x-\gamma(x)\|_\infty .$$

Since this inequality holds for any multi-bijection γ, we obtain the bottleneck stability.

Proposition 8 *Let D and E be persistence diagrams, a C_0-universal kernel k satisfy (K), and a weight function w satisfy (W1). Then,*

$$\|E_k(\mu_D^w) - E_k(\mu_E^w)\|_{\mathcal{H}_k} \le (L_k B_1 + B_k L_1)\, d_{\mathrm{W}_\infty}(D,E).$$

In this paper, among many choices, we propose to use a weight function

$$w_{\mathrm{arc}}(x) = \arctan(C\mathrm{pers}(x)^p) \quad (C>0,\ p\in\mathbb{Z}_{>0}).$$

This is a bounded and increasing function of $\mathrm{pers}(x)$. The corresponding positive definite kernel is

$$k_{\mathrm{PWG}}(x,y) = w_{\mathrm{arc}}(x) w_{\mathrm{arc}}(y) e^{-\frac{\|x-y\|^2}{2\sigma^2}}. \quad (7)$$

We call it *persistence weighted Gaussian kernel* (PWGK). This function w_{arc} gives a small (resp. large) weight on a noisy (resp. essential) generator. In addition, by appropriately adjusting the parameters C and p in w_{arc}, we can control the effect of the persistence. In order to check whether w_{arc} satisfies (W1), we first have

$$\sum_{x\in D} |w_{\mathrm{arc}}(x)| \le C\mathrm{Pers}_p(D) \quad (8)$$

from the fact $w_{\mathrm{arc}}(x) \le C\mathrm{pers}(x)^p$ $(x\in\mathbb{R}^2)$, and

$$\begin{aligned}
&\sum_{x\in D\cup\Delta} |w_{\mathrm{arc}}(x) - w_{\mathrm{arc}}(\gamma(x))| \\
&\le 2pC \sum_{x\in D\cup\Delta_1} \max\{\mathrm{pers}(x)^{p-1}, \mathrm{pers}(\gamma(x))^{p-1}\} \|x-\gamma(x)\|_\infty \ \text{(Lemma 18 in Appendix C)} \\
&\le 2pC \left(\mathrm{Pers}_{p-1}(D\cup\Delta) + \mathrm{Pers}_{p-1}(\gamma(D\cup\Delta))\right) \sup_{x\in D\cup\Delta} \|x-\gamma(x)\|_\infty \\
&\le 2pC \left(\mathrm{Pers}_{p-1}(D) + \mathrm{Pers}_{p-1}(E)\right) \sup_{x\in D\cup\Delta} \|x-\gamma(x)\|_\infty . \quad (9)
\end{aligned}$$

Although total persistences in Equation (8) and Equation (9) are not constant, by restricting a class of persistence diagrams to that of a ball model filtration, w_{arc} satisfies (W1). Therefore, we obtain the bottleneck stability for PWGK:

Theorem 9 *Let M be a triangulable compact subspace in $\mathbb{R}^d$, $X, Y \subset M$ be finite subsets, $p > d+1$, and a C_0-universal kernel k satisfy (K). Then,*

$$\left\| E_k(\mu_{D_q(X)}^{w_{\mathrm{arc}}}) - E_k(\mu_{D_q(Y)}^{w_{\mathrm{arc}}}) \right\|_{\mathcal{H}_k} \le L_{k,\mathrm{arc}} d_{\mathrm{W}_\infty}(D_q(X), D_q(Y)),$$

where $L_{k,\mathrm{arc}}$ is a constant independent of X and Y.

Proof From Lemma 4, for $p-1>d$, there exists a constant $C_M>0$ such that

$$\mathrm{Pers}_p(D_q(X)) \le \frac{p}{p-d} C_M \mathrm{diam}(M)^{p-d},$$
$$\mathrm{Pers}_{p-1}(D_q(X)),\ \mathrm{Pers}_{p-1}(D_q(Y)) \le \frac{p-1}{p-1-d} C_M \mathrm{diam}(M)^{p-1-d}.$$

Thus, from Equation (8) and Equation (9), we obtain the constants in (W1) as

$$B_1 := \frac{p}{p-d} C C_M \mathrm{diam}(M)^{p-d}, \quad L_1 := \frac{4p(p-1)}{p-1-d} C C_M \mathrm{diam}(M)^{p-1-d},$$

and the statement is proven from Proposition 8. Note that

$$\begin{aligned} L_{k,\mathrm{arc}} &:= L_k B_1 + B_k L_1 \\ &= \left(\frac{\pi L_k}{2} \frac{p}{p-d} \mathrm{diam}(M) + B_k \frac{4p(p-1)}{p-1-d} \right) C C_M \mathrm{diam}(M)^{p-1-d}, \end{aligned}$$

is actually a constant independent of X and Y. ■

Let $\mathcal{P}_{\mathrm{finite}}(M)$ be the set of finite subsets in a triangulable compact subspace $M \subset \mathbb{R}^d$. Since the constant $L_{k_{\mathrm{G}},\mathrm{arc}}$ is independent of X and Y, Proposition 2 and Theorem 9 conclude that the map

$$\mathcal{P}_{\mathrm{finite}}(M) \to \mathcal{H}_{k_{\mathrm{G}}}, \quad X \mapsto E_{k_{\mathrm{G}}}(\mu^{w_{\mathrm{arc}}}_{D_q(X)})$$

is Lipschitz continuous. Note again that this implies a desirable stability property of the PWGK with the ball model: small perturbation of data points in terms of the Hausdorff distance causes only small perturbation of the persistence diagrams in terms of the RKHS distance with the PWGK. Note also that the RKHS of the PWGK is infinite dimensional. This can be seen from Proposition 7 and the fact that the Gaussian kernel defines an infinite dimensional RKHS on $\mathbb{R}^2_{\mathrm{ad}}$.

As the most relevant work to the PWGK, the persistence scale-space kernel (PSSK, Reininghaus et al. (2015)) provides another kernel method for vectorization of persistence diagrams and its stability result is shown with respect to 1-Wasserstein distance.[11] However, to the best of our knowledge, 1-Wasserstein stability with respect to the Hausdorff distance is not shown, that is, for point sets X and Y, $d_{\mathrm{W}_1}(D_q(X), D_q(Y))$ is not estimated by $d_{\mathrm{H}}(X,Y)$ such as Proposition 2 or Corollary 5. Furthermore, it is shown (Reininghaus et al., 2015, Theorem 3) that the PSSK does not satisfy the stability with respect to p-Wasserstein distance for $p>1$, including the bottleneck distance d_{W_∞}, and hence it is not ensured that results obtained from the PSSK are stable under perturbation of data points in terms of the Hausdorff distance. On the other hand, since the PWGK has the desirable stability (Theorem 9), it is one of the advantages of our method over the previous research.

For completeness of theoretical discussions, we will show some mathematical results on the the stability with respect to 1-Wasserstein distance for PWGK along the line of Reininghaus et al. (2015). Now, we consider the following assumption (W2) which is weaker than (W1).

11. See Section 4.1.1.

(W2) For any $x, y \in \mathbb{R}^2$, there exist constants $B_2, L_2 > 0$ such that

$$|w(x)| \le B_2, \quad |w(x) - w(y)| \le L_2 \|x-y\|_\infty . \tag{10}$$

Proposition 10 *Let D and E be persistence diagrams, a C_0-universal kernel k satisfy (K), and a weight function w satisfy (W2). Then,*

$$\|E_k(\mu^w_D) - E_k(\mu^w_E)\|_{\mathcal{H}_k} \le (L_k B_2 + B_k L_2)\, d_{\mathrm{W}_1}(D,E).$$

Proof From Equation (5), we have

$$\begin{aligned} \|E_k(\mu^w_D) - E_k(\mu^w_E)\|_{\mathcal{H}_k} &\le L_k \sum_{x\in D} w(x) \|x - \gamma(x)\|_\infty + B_k \sum_{x \in D \cup \Delta_1} |w(x) - w(\gamma(x))| \\ &\le L_k B_2 \sum_{x\in D} \|x-\gamma(x)\|_\infty + B_k L_2 \sum_{x \in D\cup\Delta_1} \|x-\gamma(x)\|_\infty \end{aligned}$$

Since this inequality holds for any multi-bijection γ, the statement is proven. ■

Here, we remark the relation between a weight function and stability. As a weight function, we also consider the following two natural weight functions

$$w_{\mathrm{pers}}(x) := \begin{cases} 0 & (\mathrm{pers}(x) < 0) \\ \frac{1}{L}\mathrm{pers}(x) & (0 \le \mathrm{pers}(x) \le L) \\ 1 & (\mathrm{pers}(x) > L) \end{cases}, \tag{11}$$
$$w_{\mathrm{one}}(x) \equiv 1,$$

where $L>0$ is a parameter. Similar to w_{arc}, a piecewise linear weighting function w_{pers} gives a weight to a generator dependent on its persistence, but it does not satisfy satisfies (W1) since $\sum_{x\in D} w_{\mathrm{pers}}(x) = \frac{1}{L}\mathrm{Pers}_1(D)$, which is not a constant. For an unweighted function w_{one}, it also does not satisfy (W1) since $\sum_{x\in D} w_{\mathrm{one}}(x) = \mathrm{card}\,(D)$. Thus, it is still unknown whether the bottleneck distance stability holds for w_{pers} or w_{one}. On the other hand, since w_{pers} and w_{one} satisfy (W2), the 1-Wasserstein stability holds for these weight functions.[12] Regarding w_{arc}, we proposed it to satisfy (W1) with restriction to the class of persistence diagrams, and obtained the bottleneck stability. For $p=1$, w_{arc} satisfies (W2) by $B_2 = \frac{\pi}{2}$ and $L_2 = 2C$ without any assumptions on persistence diagrams.

Corollary 11 *Let D and E be persistence diagrams and a C_0-universal kernel k satisfy (K). Then,*

$$\left\|E_k(\mu^{w_{\mathrm{pers}}}_D) - E_k(\mu^{w_{\mathrm{pers}}}_E)\right\|_{\mathcal{H}_k} \le \left(L_k + \frac{2B_k}{L}\right) d_{\mathrm{W}_1}(D,E),$$
$$\left\|E_k(\mu^{w_{\mathrm{one}}}_D) - E_k(\mu^{w_{\mathrm{one}}}_E)\right\|_{\mathcal{H}_k} \le L_k d_{\mathrm{W}_1}(D,E),$$
$$\left\|E_k(\mu^{w_{\mathrm{arc}}}_D) - E_k(\mu^{w_{\mathrm{arc}}}_E)\right\|_{\mathcal{H}_k} \le \left(\frac{\pi L_k}{2} + 2B_k C\right) d_{\mathrm{W}_1}(D,E) \quad (p=1 \text{ in } w_{\mathrm{arc}}).$$

12. Regarding w_{pers}, L_2 in (W2) is given by $\frac{2}{L}$. See Lemma 17 in Appendix C

For $p > 1$, in general, w_{arc} does not satisfy (W2) since Ct^p is not Lipschitz continuous with respect to $t \in \mathbb{R}$. Similar to Theorem 9, by restricting to the class of persistence diagrams, we have the 1-Wasserstein stability:

Corollary 12 *Let M be a triangulable compact subspace in $\mathbb{R}^d$, $X, Y \subset M$ be finite subsets, $p > d+1$, and a C_0-universal kernel k satisfy (K). Then,*

$$\left\| E_k(\mu^{w_{\mathrm{arc}}}_{D_q(X)}) - E_k(\mu^{w_{\mathrm{arc}}}_{D_q(Y)}) \right\|_{\mathcal{H}_k} \leq \left(\frac{\pi L_k}{2} + B_k \frac{4p(p-1)}{p-1-d} C C_M \mathrm{diam}(M)^{p-1-d} \right) d_{\mathrm{W}_1}(D_q(X), D_q(Y)),$$

for some constant $C_M > 0$.

Proof For any multi-bijection $\gamma : D_q(X) \cup \Delta \to D_q(Y) \cup \Delta$, we have

$$\begin{aligned} &\sum_{x \in D_q(X) \cup \Delta} |w_{\mathrm{arc}}(x) - w_{\mathrm{arc}}(\gamma(x))| \\ &\leq 2pC \left(\mathrm{Pers}_{p-1}(D_q(X)) + \mathrm{Pers}_{p-1}(D_q(Y)) \right) \sup_{x \in D_q(X) \cup \Delta} \|x - \gamma(x)\|_\infty \quad \text{(from Equation (8))} \\ &\leq \frac{4p(p-1)}{p-1-d} C C_M \mathrm{diam}(M)^{p-1-d} \sum_{x \in D_q(X) \cup \Delta} \|x - \gamma(x)\|_\infty \end{aligned}$$

From Equation (5) and $\arctan(t) \leq \frac{\pi}{2}$ $(t \in \mathbb{R})$, the statement is proven. ■

3.3 Kernel Methods on RKHS

Once persistence diagrams are represented as RKHS vectors, we can apply any kernel methods to those vectors by defining a kernel over the vector representation. In a similar way to the standard vectors, the simplest choice is to consider the inner product as a linear kernel

$$K_{\mathrm{L}}(D, E; k, w) := \langle E_k(\mu^w_D), E_k(\mu^w_E) \rangle_{\mathcal{H}_k} = \sum_{x \in D} \sum_{y \in E} w(x) w(y) k(x, y) \tag{12}$$

on the RKHS and we call it the *(k, w)-linear kernel*.

If k is a C_0-universal kernel and w is strictly positive on $\mathbb{R}^2_{\mathrm{ad}}$, from Proposition 6, $\|E_k(\mu^w_D) - E_k(\mu^w_E)\|_{\mathcal{H}_k}$ defines a distance on the persistence diagrams and it is computed as

$$\sqrt{K_{\mathrm{L}}(D, D; k, w) + K_{\mathrm{L}}(E, E; k, w) - 2K_{\mathrm{L}}(D, E; k, w)}.$$

Then, we can also consider a nonlinear kernel

$$K_{\mathrm{G}}(D, E; k, w) = \exp\left(-\frac{1}{2\tau^2} \|E_k(\mu^w_D) - E_k(\mu^w_E)\|^2_{\mathcal{H}_k} \right) \quad (\tau > 0) \tag{13}$$

on the RKHS and we call it the *(k, w)-Gaussian kernel*.

In this paper, if there is no confusion, we also refer to the $(k_{\mathrm{G}}, w_{\mathrm{arc}})$-Gaussian kernel as the PWGK. Muandet et al. (2012) observed better performance with nonlinear kernels for some complex tasks and this is one of the reasons that we will use the Gaussian kernel on the RKHS.

3.4 Computation of Gram Matrix

Let $\mathcal{D} = \{D_\ell \mid \ell = 1, \ldots, n\}$ be a collection of persistence diagrams. In many practical applications, the number of generators in a persistence diagram can be large, while n is often relatively small; in Section 4.4, for example, the number of generators is about 30000, while $n = 80$.

If the persistence diagrams contain at most m points, each element of the Gram matrix $(K_{\mathrm{G}}(D_i, D_j; k_{\mathrm{G}}, w))_{i,j=1,\ldots,n}$ involves $O(m^2)$ evaluations of $e^{-\frac{\|x-y\|^2}{2\sigma^2}}$, resulting the complexity $O(m^2 n^2)$ for obtaining the Gram matrix. Hence, reducing computational cost with respect to m is an important issue.

We solve this computational issue by using the random Fourier features (Rahimi and Recht, 2007). To be more precise, let $z_1, \ldots, z_{M_{\mathrm{rff}}}$ be random variables from the 2-dimensional normal distribution $N((0,0), \sigma^{-2} I)$ where I is the identity matrix. This method approximates $e^{-\frac{\|x-y\|^2}{2\sigma^2}}$ by $\frac{1}{M_{\mathrm{rff}}} \sum_{a=1}^{M_{\mathrm{rff}}} e^{\sqrt{-1} z_a^T x} (e^{\sqrt{-1} z_a^T y})^*$, where $*$ denotes the complex conjugate. Then, $\sum_{x \in D_i} \sum_{y \in D_j} w(x) w(y) k_{\mathrm{G}}(x, y)$ is approximated by $\frac{1}{M_{\mathrm{rff}}} \sum_{a=1}^{M_{\mathrm{rff}}} B^a_i (B^a_j)^*$, where $B^a_\ell = \sum_{x \in D_\ell} w(x) e^{\sqrt{-1} z_a^T x}$. As a result, the computational complexity of the approximated Gram matrix is $O(mn M_{\mathrm{rff}} + n^2 M_{\mathrm{rff}})$, which is linear to m. In Section 4.3 and Section 4.4, we set $M_{\mathrm{rff}} = 10^5$. For the convergence rate of this approximation with respect to M_{rff}, please see Appendix D.

We note that the approximation by the random Fourier features can be sensitive to the choice of σ. If σ is much smaller than $\|x - y\|$, the relative error can be large. For example, in the case of $x = (1, 2), y = (1, 2.1)$ and $\sigma = 0.01$, $e^{-\frac{\|x-y\|^2}{2\sigma^2}}$ is about 10^{-22} while we observed the approximated value can be about 10^{-4} with $M_{\mathrm{rff}} = 10^5$. As a whole, these m^2 errors may cause a critical error to the statistical analysis. Moreover, if σ is largely deviated from the ensemble $\|x - y\|$ for $x \in D_i, y \in D_j$, then most values $e^{-\frac{\|x-y\|^2}{2\sigma^2}}$ become close to 0 or 1.

In order to obtain a good approximation and extract meaningful values, the choice of parameters is important. For unsupervised case, we follow the heuristics proposed in Gretton et al. (2007) and set

$$\sigma = \mathrm{median}\{\sigma(D_\ell) \mid \ell = 1, \ldots, n\}, \text{ where } \sigma(D) = \mathrm{median}\{\|x_i - x_j\| \mid x_i, x_j \in D,\ i < j\},$$

so that σ takes close values to many $\|x - y\|$. For the parameter C, we also set

$$C = (\mathrm{median}\{\mathrm{pers}(D_\ell) \mid \ell = 1, \ldots, n\})^{-p}, \text{ where } \mathrm{pers}(D) = \mathrm{median}\{\mathrm{pers}(x_i) \mid x_i \in D\}.$$

Similarly, the parameter τ in the (k, w)-Gaussian kernel is defined by

$$\mathrm{median}\left\{ \left\| E_k(\mu^w_{D_i}) - E_k(\mu^w_{D_j}) \right\|_{\mathcal{H}_k} \;\middle|\; 1 \leq i < j \leq n \right\}. \tag{14}$$

For supervised learning such as SVM, we use the cross-validation (CV) approach and do not use the random Fourier features in Section 4.2 and Section 4.5.

4. Experiments

In this section, we apply the kernel method of the PWGK to synthesized and real data, and compare the performance between the PWGK and other statistical methods of persistence diagrams. All persistence diagrams are obtained from the ball model filtrations and computed by CGAL (Da et al., 2015) and PHAT (Bauer et al., 2014). With respect to the dimension of persistence diagrams, we use 2-dimensional persistence diagrams in Section 4.3 and 1-dimensional ones in other parts.

4.1 Comparisons to Previous Works

4.1.1 Persistence Scale-space Kernel

The most relevant work to our method is the one proposed by Reininghaus et al. (2015). Inspired by the heat equation, they propose a positive definite kernel called *persistence scale-space kernel* (PSSK) K_{PSS} on the persistence diagrams:

$$K_{\mathrm{PSS}}(D,E) = \langle \Phi_t(D), \Phi_t(E) \rangle_{L^2(\mathbb{R}^2)} = \frac{1}{8\pi t} \sum_{x\in D} \sum_{y\in E} e^{-\frac{\|x-y\|^2}{8t}} - e^{-\frac{\|x-\bar{y}\|^2}{8t}}, \tag{15}$$

where $\Phi_t(D)(x) = \frac{1}{4\pi t} \sum_{y\in D} e^{-\frac{\|x-y\|^2}{4t}} - e^{-\frac{\|x-\bar{y}\|^2}{4t}}$ and $\bar{y} := (y^2, y^1)$ for $y = (y^1, y^2)$. We note that $\Phi_t(D)$ also takes zero on the diagonal by subtracting the Gaussian kernels for y and $\bar{y}$.

In fact, we can verify that the (k, w)-linear kernel contains the PSSK. Let $\tilde{D} := D \cup D^*$ where $D^* = \{(d, b) \in \mathbb{R}^2 \mid (b, d) \in D\}$. Then, $\Phi_t(D)$ can also be expressed as

$$\Phi_t(D) = \frac{1}{4\pi t} \sum_{y\in \tilde{D}} w_{\mathrm{PSS}}(y) k_{\mathrm{G}}(\cdot, y) \quad \text{where} \quad w_{\mathrm{PSS}}(y) = \begin{cases} 1, & y^2 > y^1 \\ 0, & y \in \Delta \\ -1, & y^2 < y^1 \end{cases},$$

which is equal to $\frac{1}{4\pi t} E_{k_{\mathrm{G}}}(\mu_{\tilde{D}}^{w_{\mathrm{PSS}}})$. Furthermore, the inner product in $\mathcal{H}_{k_{\mathrm{G}}}$ is

$$K_{\mathrm{L}}(\tilde{D}, \tilde{E}; k_{\mathrm{G}}, w_{\mathrm{PSS}}) = \langle E_{k_{\mathrm{G}}}(\mu_{\tilde{D}}^{w_{\mathrm{PSS}}}), E_{k_{\mathrm{G}}}(\mu_{\tilde{E}}^{w_{\mathrm{PSS}}}) \rangle_{\mathcal{H}_{k_{\mathrm{G}}}} = 2 \sum_{x\in D} \sum_{y\in E} k_{\mathrm{G}}(x, y) - k_{\mathrm{G}}(x, \bar{y}). \tag{16}$$

By scaling the variance parameter σ in the Gaussian kernel k_{G} and multiplying by an appropriate scalar, Equation (15) is the same as Equation (16). Thus, the PSSK can also be approximated by the random Fourier features. When we apply the random Fourier features for the PSSK, we set $\tilde{\sigma} = \mathrm{median}\{\sigma(\tilde{D}_\ell) \mid \ell = 1, \cdots, n\}$ as before and $t = \frac{\tilde{\sigma}^2}{4}$.

While both methods discount noisy generators, the PWGK has the following advantages over the PSSK. (i) The PWGK can control the effect of the persistence by C and p in w_{arc} independently of the bandwidth parameter σ in the Gaussian factor, while in the PSSK only one parameter t cannot adjust the global bandwidth and the effect of persistence simultaneously. (ii) The PSSK does not satisfy the stability with respect to the bottleneck distance (see also remarks after Theorem 9).

4.1.2 Persistence Landscape

The *persistence landscape* (Bubenik, 2015) is a well-known approach in TDA for vectorization of persistence diagrams. For a persistence diagram D, the persistence landscape λ_D is defined by

$$\lambda_D(k, t) = k\text{-th largest value of } \min\{t - b_i, d_i - t\}_+,$$

where c_+ denotes $\max\{c, 0\}$, and it is a vector in the Hilbert space $L^2(\mathbb{N} \times \mathbb{R})$. Here, we define a positive definite kernel of persistence landscapes as a linear kernel on $L^2(\mathbb{N} \times \mathbb{R})$:

$$K_{\mathrm{PL}}(D, E) := \langle \lambda_D, \lambda_E \rangle_{L^2(\mathbb{N}\times\mathbb{R})} = \int_{\mathbb{R}} \sum_{k=1} \lambda_D(k, t) \lambda_E(k, t) dt. \tag{17}$$

Since a persistence landscape does not have any parameters, we do not need to consider the parameter tuning. However, the integral computation is required and it causes much computational time. Let $\mathcal{D} = \{D_\ell \mid \ell = 1, \ldots, n\}$ be a collection of persistence diagrams which contain at most m points. Since $\lambda_{D_i}(k, t) \equiv 0$ for any $k > m$, $t \in \mathbb{R}$, $i = 1, \cdots, n$, calculating $\{\lambda_{D_i}(k, t) \mid k \in \mathbb{Z}_{\geq 0}\}$, which needs sorting, is in $O(m \log m)$ (see also Bubenik and Dłotko (2017)). For a fixed t, we can calculate $(\sum_{k=1} \lambda_{D_i}(k, t)\lambda_{D_j}(k, t))_{i,j=1,\cdots,n}$ in $O(nm \log m + n^2 m)$, and the Gram matrix $(K_{\mathrm{PL}}(D_i, D_j))_{i,j=1,\cdots,n}$ in $O(M_{\mathrm{int}}(nm \log m + n^2 m))$, where M_{int} is the number of partitions in the integral interval. Theoretically speaking, this implies that it takes more time to calculate the Gram matrix of K_{PL} than the PWGK and the PSSK by the random Fourier features.

4.1.3 Persistence Image

As a finite dimensional vector representation of a persistence diagram, a *persistence image* is proposed in Adams et al. (2017). First, we prepare a differentiable probability density function $\phi_x : \mathbb{R}^2 \to \mathbb{R}$ with mean x and a weight function $w : \mathbb{R}^2_{\mathrm{ad}} \to \mathbb{R}$. For a persistence diagram D, the *corresponding persistence surface* is defined by

$$\rho_D(z) := \sum_{x\in D} w(x) \phi_x(z). \tag{18}$$

Then, for fixed points $a_0 < \cdots < a_M$ ($a_i \in \mathbb{R}$), the *persistence image*[13] $\mathrm{PI}(D)$ is defined by an $M \times M$ matrix whose (i, j)-element is assigned to the integral of ρ_D over the pixel $P_{i,j} := (a_{i-1}, a_i] \times (a_{j-1}, a_j]$, i.e.,

$$\mathrm{PI}(D)_{i,j} := \int_{P_{i,j}} \rho_D(z) dz.$$

Since the persistence image can be regarded as an M^2-dimensional vector, we define a vector $\mathrm{PIV}(D) \in \mathbb{R}^{M^2}$ by

$$\mathrm{PIV}(D)_{i+M(j-1)} := \mathrm{PI}(D)_{i,j}, \tag{19}$$

13. Adams et al. (2017) use a persistence diagram in birth-persistence coordinates. That is, by a linear transformation $T(b, d) = (b, d - b)$, a persistence diagram D is transformed into $T(D)$. In this paper, in order to compare with the persistence image and the PWGK, we use birth-death coordinates.

and, in this paper, call it the persistence image vector.

In Adams et al. (2017), they use the 2-dimensional Gaussian distribution $\frac{1}{2\pi\sigma^2}k_{\mathrm{G}}(x,z)$ as $\phi_x(z)$ and a piecewise linear weighting function $w_{\mathrm{pers}}(x)$. In this paper, for a collection of persistence diagrams $\mathcal{D} = \{D_\ell \mid \ell = 1,\dots,n\}$, we set a parameter L in Equation (11) as

$$L = \max\{L(D_\ell) \mid \ell = 1,\cdots,n\}, \text{ where } L(D) = \max\{d_i \mid (b_i,d_i) \in D\}.$$

For points $a_0 < \cdots < a_M$ of a pixel $P_{i,j} = (a_{i-1},a_i] \times (a_{j-1},a_j]$, we set $a_M = L$ and $a_i = \frac{i}{M}a_M$ for $0 \le i \le M$.[14]

Here, by choosing ϕ_x and w in the proposed way, we define a positive definite kernel of persistence image vector as a linear kernel on $\mathbb{R}^{M^2}$:

$$\begin{aligned} K_{\mathrm{PI}}(D,E) &:= \langle \mathrm{PIV}(D), \mathrm{PIV}(E)\rangle_{\mathbb{R}^{M^2}} \\ &= \sum_{i,j=1}^{M} \mathrm{PI}(D)_{i,j}\mathrm{PI}(E)_{i,j} \\ &= \frac{1}{(2\pi\sigma^2)^2}\sum_{x\in D}\sum_{y\in E} w_{\mathrm{pers}}(x)w_{\mathrm{pers}}(y)\sum_{i,j=1}^{M}\int_{P_{i,j}} k_{\mathrm{G}}(x,z)dz\int_{P_{i,j}} k_{\mathrm{G}}(y,z)dz. \quad (20) \end{aligned}$$

If we choose $\phi_x(z)$ as a (normalized) positive definite kernel $k(x,z)$, the corresponding persistence surface ρ_D (18) is the same as the RKHS vector $E_k(\mu_D^w)$. Thus, it may be expected that the persistence image and the PWGK show similar performance for data analysis. However, there are several differences between the persistence image and the PWGK. (i) Underlying vector spaces are different: the PWGK vector $E_k(\mu_D^w)$ is always in the RKHS and the corresponding persistence surface ρ_D is in $L^p(\mathbb{R}^2)$ with appropriate conditions. Hence, the inner product structures are also different.[15] (ii) Regarding the mapping from a persistence diagram to the corresponding persistence surface, the injectivity is not discussed in the original paper (Adams et al., 2017). On the other hand, from Proposition 6, we can easily check the injectivity of the RKHS vector $E_k(\mu_D^w)$ due to its construction based on kernel method. (iii) It is also shown that the persistence image has a stability result with respect to 1-Wasserstein distance, but it does not satisfy the bottleneck stability (Remark 1 in Adams et al. (2017)) or the Haussdorff stability as noted after Theorem 9. This instability is considered to be caused by the norm of the persistence image, which is different from the RKHS. (iv) The computational complexity of a persistence image does not depend on the number of generators in a persistence diagram, but instead, it depends on the number of pixels M^2. Precisely, the Gram matrix $(K_{\mathrm{PI}}(D_i,D_j))_{i,j=1,\cdots,n}$ is calculated in $O(n^2M^2)$. We can reduce the computational time of the persistence image by choosing a small mesh size M. However, some situations need a fine mesh (i.e., a large mesh size), and thus, we have to be careful with the choice of mesh size. In Section 4.2.2, we will discuss the effect of the mesh size on the classification performance of the persistence image.

14. Here, we set $a_0 = 0$ because all generators in the ball model filtrations are born after $b = 0$.

15. Since the persistence image vector $\mathrm{PIV}(D)$ (19) is a discretization of ρ_D, the inner product (20) can be also seen as a discretization of L^2 inner product of the corresponding persistence surfaces

$$\langle \rho_D, \rho_E\rangle_{L^2(\mathbb{R}^2)} = \frac{1}{(2\pi\sigma^2)^2}\sum_{x\in D}\sum_{y\in E} w_{\mathrm{pers}}(x)w_{\mathrm{pers}}(y)\int_{\mathbb{R}^2} k_{\mathrm{G}}(x,z)k_{\mathrm{G}}(y,z)dz.$$

Furthermore, since $\int_{\mathbb{R}^2} e^{-\frac{\|x-z\|^2}{2\sigma^2}}e^{-\frac{\|y-z\|^2}{2\sigma^2}}dz \propto e^{-\frac{\|x-y\|^2}{4\sigma^2}}$, $K_{\mathrm{PI}}(D,E)$ is also a discretization of the inner product of the RKHS vectors $K_{\mathrm{L}}(D,E;k_{\mathrm{G}},w_{\mathrm{arc}})$ by scaling the variance parameter σ in k_{G}. However, this is a special case, and it is not always to be true for any positive definite kernel.

4.2 Classification with Synthesized Data

We compare the performance among the PWGK, the PSSK, the persistence landscape, and the persistence image for a simple binary classification task with SVMs.

4.2.1 Synthesized Data

In this experiment, we design data sets so that important generators close to the diagonal must be taken into account to solve the classification task.

Let $S^1(x,y,r,N)$ be a set composed of N points sampled with equal distance from a circle in 2-dimensional Euclidean space with radius r centered at (x,y). When we compute the persistence diagram of $S^1(x,y,r,N)$ for $N > 3$, there always exists a generator whose birth time is approximately $\frac{\pi r}{N}$ (here we use $\sin\theta \approx \theta$ for small θ) and death time is r (Figure 6).

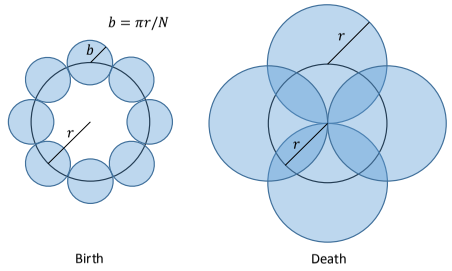


Figure 6: Birth and death of the generator for $S^1(x,y,r,N)$.

In order to add randomness on $S^1(x,y,r,N)$, we extend it into $\mathbb{R}^3$ and change $S^1(x,y,r,N)$ to $S^1_z(x,y,r,N)$ and $\tilde{S}^1_z(x,y,r,N)$ as follows:

$$\begin{aligned} S^1_z(x,y,r,N) &:= \{(z_1,z_2,z_3) \mid (z_1,z_2) \in S^1(x,y,r,N),\ z_3 \text{ is uniformly sampled from } [0,0.01]\} \\ \tilde{S}^1_z(x,y,r,N) &:= S^1_z(x+W_x^2, y+W_y^2, r+W_r^2, \lceil N+2W_N\rceil), \end{aligned}$$

where $W_x, W_y \sim N(0,2)$, $W_r, W_N \sim N(0,1)$ and $\lceil c\rceil$ is the smallest integer greater than or equal to c.[16] Then, we add $S_2 := S^1_z(x_2,y_2,r_2,N_2)$ to $S_1 := \tilde{S}^1_z(x_1,y_1,r_1,N_1)$ with probability 0.5 and use it as the synthesized data.

16. $N(\mu,\sigma^2)$ is the 1-dimensional normal distribution with mean μ and variance σ^2.

In this paper, we choose parameters by

$$r_1 = 1 + 8W^2 \quad (W \sim N(0,1)),$$
$$x_1 = y_1 = 1.5 r_1,$$
$$N_1 \ : \text{ a random integer with equal probability in } (\lceil \frac{\pi r}{2} \rceil, 4\pi r),$$

and set (x_2, y_2, r_2, N_2) as $(0, 0, 0.2, 10)$ (Figure 7).

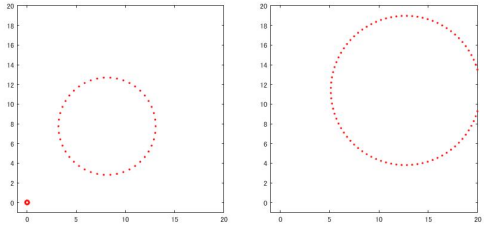


Figure 7: Examples of synthesized data. Left: S_2 exists. Right: S_2 does not exist.

For the binary classification, we introduce the following labels:

$z_0 = 1$ if there exists a generator (b, d) in the persistence diagram such that $b \leq 1$ and $d \geq 4$.

$z_1 = 1$ if S_2 exists.

The class label of the data set is then given by $\mathbf{XOR}(z_0, z_1)$. By this construction, identifying z_0 requires relatively smooth function in the area of long lifetimes, while classifying the existing of z_1 needs delicate control of the resolution around the diagonal.

4.2.2 SVM Results

SVMs are trained from persistence diagrams given by 100 data sets, and evaluated with 100 independent test data sets. As a positive definite kernel k, we choose the Gaussian kernel k_{G} and the linear kernel $k_{\mathrm{L}}(x, y) := \langle x, y \rangle_{\mathbb{R}^2}$. For a weight function w, we use the proposed function $w_{\mathrm{arc}}(x) = \arctan(C\mathrm{pers}(x)^p)$, the piecewise linear weighting function $w_{\mathrm{pers}}(x)$, and an unweighted function $w_{\mathrm{one}}(x) \equiv 1$. The hyper-parameters (σ, C) in the PWGK and t in the PSSK are chosen by the 10-fold cross-validation, and the degree p in $w_{\mathrm{arc}}(x)$ is set as $1, 5, 10$. For K_{PSS} and K_{PL}, while they originally consider only the inner product, we also apply the Gaussian kernels on RKHS following Equation (13). Since K_{PI} can be seen as a discretization of the $(k_{\mathrm{G}}, w_{\mathrm{pers}})$-linear kernel, we also construct another kernel of persistence image by replacing w_{pers} with w_{arc}, which is considered as a discretization of the PWGK. In order to check whether the persistence image with w_{arc} is an appropriate discretization of the PWGK, we try several mesh size $M = 20, 50, 100$.

In Table 1, we can see that the PWGK $\triangle$ and the Gaussian kernel on the persistence image with w_{arc} and large mesh size $\square_{100}$ show higher classification rates (85% accuracy)

		Linear	Gaussian
PWGK			
kernel	**weight**		
k_{G}	w_{arc} $(p = 1)$	75.7 ± 2.31	85.8 ± 5.19 (PWGK)
	w_{arc} $(p = 5)$	75.8 ± 2.47 ($\triangle$)	85.6 ± 5.01 (PWGK, $\square$)
	w_{arc} $(p = 10)$	76.0 ± 2.39	86.0 ± 4.98 (PWGK)
	w_{pers}	49.3 ± 2.72	52.3 ± 6.60
	w_{one}	53.8 ± 4.76	55.1 ± 8.42
k_{L}	w_{arc} $(p = 5)$	49.3 ± 6.92	51.8 ± 3.52
	w_{pers}	51.0 ± 6.84	55.7 ± 8.68
	w_{one}	50.5 ± 6.90	53.0 ± 4.89
PWGK with Persistence image			
$M = 20$	w_{arc} $(p = 5)$	48.8 ± 3.75 ($\triangle_{20}$)	52.0 ± 5.65 ($\square_{20}$)
$M = 50$	w_{arc} $(p = 5)$	49.2 ± 5.77 ($\triangle_{50}$)	51.8 ± 7.23 ($\square_{50}$)
$M = 100$	w_{arc} $(p = 5)$	75.0 ± 2.20 ($\triangle_{100}$)	85.8 ± 4.15 ($\square_{100}$)
PSSK		50.5 ± 5.60 (K_{PSS})	53.6 ± 6.69
Persistence landscape		50.6 ± 5.92 (K_{PL})	48.8 ± 4.25
Persistence image			
$M = 20$	w_{pers}	51.1 ± 4.38 (K_{PI})	51.7 ± 6.86
$M = 50$	w_{pers}	49.0 ± 6.14 (K_{PI})	52.3 ± 7.21
$M = 100$	w_{pers}	54.5 ± 8.76 (K_{PI})	52.1 ± 6.70

Table 1: Results of SVMs with the (k, w)-linear/Gaussian kernel, the PSSK, the persistence landscape, and the persistence image. Average classification rates (%) and standard deviations for 100 test data sets are shown.

than the other methods (K_{PSS} : 50%, K_{PL} : 50%, and K_{PI} : 55%). Although the $(k_{\mathrm{G}}, w_{\mathrm{pers}})$-Gaussian kernel and the persistence image with the original weight w_{pers} discount noisy generators, the classification rates are close the chance level. These unfavorable results must be caused by the difficulty in handling the local and global locations of generators simultaneously. While the result of the persistence image with a large mesh size is similar to that of the PWGK (e.g., $\square$ and $\square_{100}$), a small mesh size gives bad approximation results (e.g., $\square$ and $\square_{50}$). The reason is because a small mesh size makes rough pixels, and S_2 itself and noisy generators are treated in some rough pixel. On the other hand, we remark that a large mesh size M needs much computational time.

We observe that the classification accuracies are not sensitive to p. Thus, in the rest of this paper, we set $p = 5$ because the assumption $p > d + 1$ in Theorem 9 ensures the continuity in the kernel embedding of persistence diagrams and all data points are obtained from $\mathbb{R}^3$.

4.3 Analysis of Granular System

We apply the PWGK, the PSSK, the persistence landscape, and the persistence image to persistence diagrams obtained by experimental data in a granular packing system (Francois et al., 2013). In this example, a partially crystallized packing with 150,000 monosized beads (diameter = 1mm, polydispersity = 0.025mm) in a container is obtained by experiments, where the configuration of the beads is imaged by means of X-ray Computed Tomography. One of the fundamental interests in the study of granular packings is to understand the transition from random packings to crystallized packings. In particular, the maximum packing density ϕ_* that random packings can attain is still a controversial issue (e.g., see Torquato et al. (2000)). Here, we apply the change point analysis to detect ϕ_*.

In oder to observe configurations of various densities, we divide the original full system into 35 cubical subsets containing approximately 4000 beads. The data are provided by the authors of the paper (Francois et al., 2013). The packing densities of the subsets range from $\phi = 0.590$ to $\phi = 0.730$. Saadatfar et al. (2017) computed a persistence diagram for each subset by taking the beads configuration as a finite subset in $\mathbb{R}^3$, and found that the persistence diagrams characterize different configurations in random packings (small ϕ) and crystallized packings (large ϕ). Hence, it is expected that the change point analysis applied to these persistence diagrams can detect the maximum packing density ϕ_* as a transition from the random to crystallized packings.

Our strategy is to regard the maximum packing density as the change point and detect it from a collection $\mathcal{D} = \{D_\ell \mid \ell = 1, \ldots, n\}$ $(n = 35)$ of persistence diagrams made by beads configurations of granular systems, where ℓ is the index of the packing densities listed in the increasing order. As a statistical quantity for the change point detection, we use the kernel Fisher discriminant ratio (Harchaoui et al., 2009) defined by

$$\mathrm{KFDR}_{n,\ell,\gamma}(\mathcal{D}) = \frac{\ell(n-\ell)}{n} \left\| \left(\frac{\ell}{n} \hat{\Sigma}_{1:\ell} + \frac{n-\ell}{n} \hat{\Sigma}_{\ell+1:n} + \gamma I \right)^{-\frac{1}{2}} (\hat{\mu}_{\ell+1:n} - \hat{\mu}_{1:\ell}) \right\|_{\mathcal{H}_K}, \qquad (21)$$

where the empirical mean element $\hat{\mu}_{i:j}$ and the empirical covariance operator $\hat{\Sigma}_{i:j}$ with data D_i through D_j $(i < j)$ are given by

$$\hat{\mu}_{i:j} = \frac{1}{j-i+1} \sum_{\ell=i}^{j} K(\cdot, D_\ell),$$

$$\hat{\Sigma}_{i:j} = \frac{1}{j-i+1} \sum_{\ell=i}^{j} (K(\cdot, D_\ell) - \hat{\mu}_{i:j}) \otimes (K(\cdot, D_\ell) - \hat{\mu}_{i:j})$$

respectively, and γ is a regularization parameter (in this paper we set $\gamma = 10^{-3}$). The index ℓ achieving the maximum of $\mathrm{KFDR}_{n,\ell,\gamma}(\mathcal{D})$ corresponds to the estimated change point. In Figure 8, all the four methods detect $\ell = 23$ as the sharp maximizer of the KFDR. This result indicates that the maximum packing density ϕ_* exists in the interval $[0.604, 0.653]$ and supports the traditional observation $\phi_* \approx 0.636$ (Anonymous, 1972).

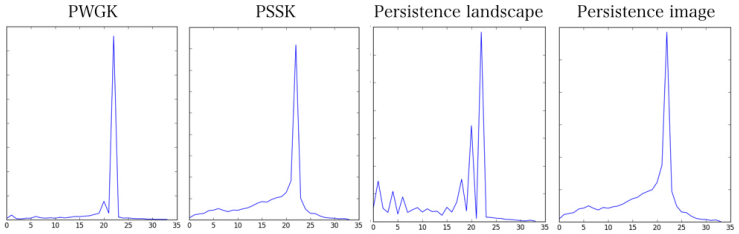


Figure 8: The KFDR graphs of the PWGK, the PSSK, the persistence landscape, and the persistence image.

We also apply kernel principal component analysis (KPCA) to the same collection of the 35 persistence diagrams. Figure 9 shows the 2-dimensional KPCA plots where each blue cross (resp. red circle) indicates the persistence diagram of random packing (resp. crystallized packing). We can see clear two-cluster structure corresponding to two physical states.

4.4 Analysis of SiO_2

When we rapidly cool down the liquid state of SiO_2, it avoids the usual crystallization and changes into a glass state. Understanding the liquid-glass transition is an important issue for the current physics and industrial applications (Greaves and Sen, 2007). Glass is an amorphous solid, which does not have a clear structure in the configuration of molecules, but it is also known that the medium distance structure such as rings have important influence on the physical properties of the material. It is thus promising to apply the persistent homology to express the topological and geometrical structure of the glass configuration. For estimating the glass transition temperature by simulations, a traditional physical method is

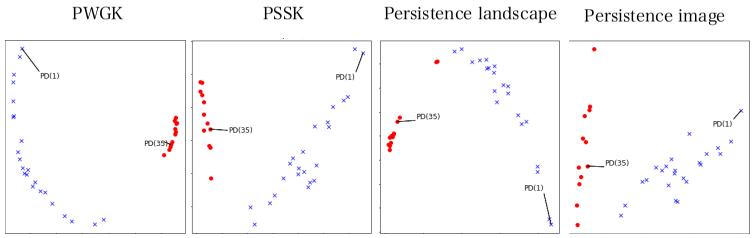


Figure 9: The KPCA plots of the PWGK (contribution rate: 92.9%), the PSSK (99.7%), the persistence landscape (83.8%), and the persistence image (98.7%).

to prepare atomic configurations of SiO_2 for a certain range of temperatures by molecular dynamics simulations, and then draw the temperature-enthalpy graph. The graph consists of two lines in high and low temperatures with slightly different slopes which correspond to the liquid and the glass states, respectively, and the glass transition temperature is conventionally estimated as an interval of the transient region combining these two lines (e.g., see Elliott (1990)). However, since the slopes of two lines are close to each other, determining the interval is a subtle problem. Usually only the rough estimate of the interval is available. Hence, we apply our framework of topological data analysis with kernels to detect the glass transition temperature.

Let $\{D_\ell \mid \ell = 1, \ldots, 80\}$ be a collection of the persistence diagrams made by atomic configurations of SiO_2 and sorted by the decreasing order of the temperature. The same data was used in the previous works by Hiraoka et al. (2016); Nakamura et al. (2015). The interval of the glass transition temperature T estimated by the conventional method explained above is $2000K \leq T \leq 3500K$, which corresponds to $35 \leq \ell \leq 50$.

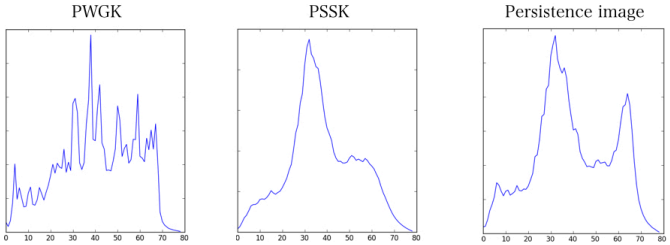


Figure 10: The KFDR graphs of the PWGK (left), the PSSK (center) and the persistence image (right).

In Figure 10, the KFDR plots show that the change point is estimated as $\ell = 39$ by the PWGK, $\ell = 33$ by the PSSK, and $\ell = 33$ by the persistence image. For the persistence landscape, we cannot obtain the KFDR or the KPCA results with reasonable computational time.

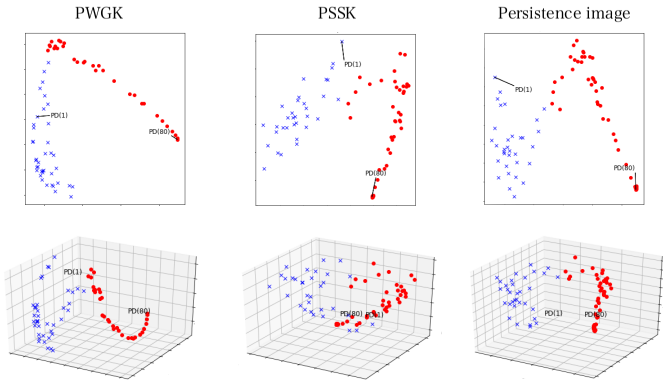


Figure 11: The 2-dimensional and 3-dimensional KPCA plots of the PWGK (contribution rates for 2-dimension: 81.7%, 3-dimension: 92.1%), the PSSK (97.2%, 99.3%) and the persistence image (99.9%, 99.9%).

As we see from the 2-dimensional plots given by KPCA (Figure 11), the PWGK presents sharp change of the gradients between before (blue cross) and after (red circle) the change point determined by the KFDR. This matches with the analysis in physics that expects a sharp change of slope in the temperature-enthalpy plane. This strongly suggests that the glass transition occurs at the detected change point. On the other hand, in the results of PSSK and persistence images we cannot observe a sharp change of the gradients at the boundary of the estimated two phases. We also remark that clearer structures are observed in the 3-dimensional KPCA plots of the PWGK.

4.5 Protein Classification

We apply the PWGK to two classification tasks studied in Cang et al. (2015). They introduced the molecular topological fingerprint (MTF) as a feature vector constructed from the persistent homology, and used it for the input to the SVM. The MTF is given by the 13-dimensional vector whose elements consist of the persistences of some specific generators

	Protein-Drug	Hemoglobin
PWGK	100	88.90
MTF	(nbd) 93.91 / (bd) 98.31	84.50

Table 2: CV classification rates (%) of SVMs with the PWGK and the MTF (cited from Cang et al. (2015)).

in persistence diagrams.[17] We compare the performance between the PWGK and the MTF method under the same setting of the SVM reported in Cang et al. (2015).

The first task is a protein-drug binding problem, where the binding and non-binding of drug to the M2 channel protein of the influenza A virus is to be classified. For each of the two forms, 15 data were obtained by NMR experiments, and 10 data are used for training and the remaining for testing. We randomly generate 100 ways of partitions and calculate the average classification rates.

In the second problem, the taut and relaxed forms of hemoglobin are to be classified. For each form, 9 data were collected by the X-ray crystallography. We select one data from each class for testing and use the remaining for training. All the 81 combinations are performed to calculate the CV classification rates.

The results of the two problems are shown in Table 4.5. We can see that the PWGK achieves better performance than the MTF in both problems.

5. Conclusion and Discussions

One of the contributions of this paper is to introduce a kernel framework to topological data analysis with persistence diagrams. We applied the kernel embedding approach to vectorize the persistence diagrams, which enables us to utilize any standard kernel methods for data analysis. Another contribution is to propose a kernel specific to persistence diagrams, that is called persistence weighted Gaussian kernel (PWGK). As a significant advantage, our kernel enables one to control the effect of persistence in data analysis. We have also proven the stability property with respect to the distance in the Hilbert space. Furthermore, we have analyzed the synthesized and real data by using the proposed kernel. The change point detection, the principal component analysis, and the support vector machine derived meaningful results for the tasks. From the viewpoint of computations, our kernel can utilize an efficient approximation to compute the Gram matrix.

One of the main theoretical results of this paper is the bottleneck stability of the PWGK (Theorem 9). It is obtained by restricting the class of persistence diagrams to that obtained from ball model filtrations. The reason of this restriction is because the total persistence can be bounded from above independent of the persistence diagram. Thus, one direction to extend this work is to examine the boundedness condition about the total persistence of other persistence diagrams, for example obtained from Rips complexes or sub-level sets.

17. The MTF method is not a general method for persistence diagrams because some elements of the MTF vector are specialized for protein data, e.g., the ninth element of the MTF vector is defined by the number of Betti 1 bars that locate at $[4.5, 5.5]$Å, divided by the number of atoms. For the details, see Cang et al. (2015).

Another direction to extend this work is to generalize the class of weight functions. The reason of the choice of w_{arc} is mainly for the stability property, but in principle, we can apply any weight function to data analysis. Even if we do not concern about stability properties, which weight function is practically good for data analysis? Suppose generators close to the diagonal are sometimes seen as important features. Then, our statistical framework can treat such small generators as significant ones by a weight function which has large weight close to the diagonal, while other statistical methods for persistence diagrams always see small generators as noisy ones. In addition, the weight function becomes better when it is constructed to satisfy the assumption (W1) or (W2), which implies the stability property.

Acknowledgement

We thank Ulrich Bauer for giving us useful comments in Section 4.1.1, Mohammad Saadatfar and Takenobu Nakamura for providing experimental and simulation data used in Section 4.3 and 4.4, and the anonymous referees for their valuable comments and suggestions This work is partially supported by JST CREST Mathematics (15656429), JSPS KAKENHI Grant Number 26540016, Structural Materials for Innovation Strategic Innovation Promotion Program D72, Materials research by Information Integration Initiative ($\mathrm{MI}^2\mathrm{I}$) project of the Support Program for Starting Up, Innovation Hub from JST, and JSPS Research Fellow (17J02401).

Appendix A. Topological tools

This section summarizes some topological tools used in the paper. To study topological properties algebraically, simplicial complexes are often considered as basic objects. We start with a brief explanation of simplicial complexes, and gradually increase the generality from simplicial homology to singular and persistent homology. For more details, see Hatcher (2002).

A.1 Simplicial complex

We first introduce a combinatorial geometric model called simplicial complex to define homology. Let $P = \{1, \dots, n\}$ be a finite set (not necessarily points in a metric space). A *simplicial complex* with the vertex set P is defined by a collection S of subsets in P satisfying the following properties:

1. $\{i\} \in S$ for $i = 1, \dots, n$, and
2. if $\sigma \in S$ and $\tau \subset \sigma$, then $\tau \in S$.

Each subset σ with $q+1$ vertices is called a q-simplex. We denote the set of q-simplices by S_q. A subcollection $T \subset S$ which also becomes a simplicial complex (with possibly less vertices) is called a subcomplex of S.

We can visually deal with a simplicial complex S as a polyhedron by pasting simplices in S into a Euclidean space. The simplicial complex obtained in this way is called a geometric realization, and its polyhedron is denoted by $|S|$. In this context, the simplices with small q correspond to points ($q = 0$), edges ($q = 1$), triangles ($q = 2$), and tetrahedra ($q = 3$).

Example 1 Figure 12 shows two polyhedra of simplicial complexes

$$S = \{\{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\},$$
$$T = \{\{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}\}.$$

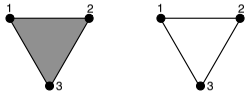


Figure 12: The polyhedra of the simplicial complexes S (left) and T (right).

A.2 Homology

A.2.1 Simplicial homology

The procedure to define homology is summarized as follows:

1. Given a simplicial complex S, build a chain complex $C_*(S)$. This is an algebraization of S characterizing the boundary.

2. Define homology by quotienting out certain subspaces in $C_*(S)$ characterized by the boundary.

We begin with the procedure 1 by assigning orderings on simplices. When we deal with a q-simplex $\sigma = \{i_0, \dots, i_q\}$ as an ordered set, there are $(q+1)!$ orderings on σ. For $q > 0$, we define an equivalence relation $i_{j_0}, \dots, i_{j_q} \sim i_{\ell_0}, \dots, i_{\ell_q}$ on two orderings of σ such that they are mapped to each other by even permutations. By definition, two equivalence classes exist, and each of them is called an oriented simplex. An oriented simplex is denoted by $\langle i_{j_0}, \dots, i_{j_q}\rangle$, and its opposite orientation is expressed by adding the minus $-\langle i_{j_0}, \dots, i_{j_q}\rangle$. We write $\langle\sigma\rangle = \langle i_{j_0}, \dots, i_{j_q}\rangle$ for the equivalence class including $i_{j_0} < \cdots < i_{j_q}$. For $q = 0$, we suppose that we have only one orientation for each vertex.

Let K be a field. We construct a K-vector space $C_q(S)$ as

$$C_q(S) = \mathrm{Span}_K\{\langle\sigma\rangle \mid \sigma \in S_q\}$$

for $S_q \neq \emptyset$ and $C_q(S) = 0$ for $S_q = \emptyset$. Here, $\mathrm{Span}_K(A)$ for a set A is a vector space over K such that the elements of A formally form a basis of the vector space. Furthermore, we define a linear map called the *boundary map* $\partial_q : C_q(S) \to C_{q-1}(S)$ by the linear extension of

$$\partial_q\langle i_0, \dots, i_q\rangle = \sum_{\ell=0}^{q} (-1)^\ell \langle i_0, \dots, \widehat{i_\ell}, \dots, i_q\rangle, \tag{22}$$

where $\widehat{i_\ell}$ means the removal of the vertex i_ℓ. We can regard the linear map ∂_q as algebraically capturing the $(q-1)$-dimensional boundary of a q-dimensional object. For example, the image of the linear map ∂_2 of a basis $\langle 1,2,3\rangle$ in the vector space $C_2(S)$ is given by

$$\partial_2\langle 1,2,3\rangle = \langle 2,3\rangle - \langle 1,3\rangle + \langle 1,2\rangle = \langle 1,2\rangle + \langle 2,3\rangle + \langle 3,1\rangle \tag{23}$$

as linear combinations of bases in the vector space $C_1(S)$. The above sentence is written only in the language of linear algebra and there is no meaning of $+$ or $-$ in Equation (23) except for its vector space structure. On the other hand, the geometric realization of $\partial_2\langle 1,2,3\rangle$ is a boundary of $\langle 1,2,3\rangle$ in a geometric sense (see Figure 12). In this way, we analyze geometric properties of a simplicial complex algebraically.

In practice, by arranging some orderings of the oriented q- and $(q-1)$- simplices, we can represent the boundary map as a matrix $M_q = (M_{\sigma,\tau})_{\sigma\in S_{q-1},\tau\in S_q}$ with the entry $M_{\sigma,\tau} = 0, \pm 1$ given by the coefficient in Equation (23). For the simplicial complex S in Example 1, the matrix representations M_1 and M_2 of the boundary maps are given by

$$M_2 = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}, \quad M_1 = \begin{bmatrix} -1 & 0 & -1 \\ 1 & -1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \tag{24}$$

Here, the 1-simplices (resp. 0-simplices) are ordered by $\langle 1,2\rangle, \langle 2,3\rangle, \langle 1,3\rangle$ (resp. $\langle 1\rangle$, $\langle 2\rangle$, $\langle 3\rangle$).

We call a sequence of the vector spaces and linear maps

$$\cdots \longrightarrow C_{q+1}(S) \xrightarrow{\partial_{q+1}} C_q(S) \xrightarrow{\partial_q} C_{q-1}(S) \longrightarrow \cdots$$

the *chain complex* of S. As an easy exercise, we can show $\partial_q \circ \partial_{q+1} = 0$. Hence, the subspaces $Z_q(S) = \ker\partial_q$ and $B_q(S) = \mathrm{im}\partial_{q+1}$ satisfy $B_q(S) \subset Z_q(S)$. Then, the q-th (simplicial) *homology* is defined by taking the quotient space

$$H_q(S) = Z_q(S)/B_q(S).$$

Note that $H_q(S)$ is a K-vector space, and the dimension can be considered in a standard way. Intuitively, the dimension of $H_q(S)$ counts the number of q-dimensional holes in S and each generator of the vector space $H_q(S)$ corresponds to these holes. We remark that the homology as a vector space is independent of the orientations of simplices. For $q = 0$, each generator of $H_0(S)$ corresponds to a path-connected component of S. This can be seen from the fact that any two vertices are in the same equivalence class modulo the boundary $B_0(S)$ if and only if they are connected by a path.

For a subcomplex T of S, the inclusion map $\rho : T \hookrightarrow S$ naturally induces a linear map in homology $\rho_q : H_q(T) \to H_q(S)$. Namely, an element $[c] \in H_q(T)$ is mapped to $[c] \in H_q(S)$, where the equivalence class $[c]$ is taken in each vector space.

For example, the simplicial complex S in Example 1 has

$$Z_1(S) = \mathrm{Span}_K[\ 1 \quad 1 \quad -1\ ]^T = B_1(S)$$

from (24). Hence $H_1(S) = 0$, meaning that there are no 1-dimensional hole (ring) in S. On the other hand, since $Z_1(T) = Z_1(S)$ and $B_1(T) = 0$, we have $H_1(T) \cong K$, meaning that T consists of one ring. Hence, the induced linear map $\rho_1 : H_1(T) \to H_1(S)$ means that the ring in T disappears in S under $T \hookrightarrow S$.

A topological space X is called *triangulable* if there exists a geometric realization of a simplicial complex S whose polyhedron is homeomorphic to X.[18] For such a triangulable topological space, the homology is defined by $H_q(X) = H_q(S)$. This is well-defined, since a different geometric realization provides an isomorphic homology.

A.2.2 Singular homology

We here extend the homology to general topological spaces. Let $e_0, \ldots, e_q$ be the standard basis of $\mathbb{R}^{q+1}$ (i.e., $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)$, 1 at $(i+1)$-th position, and 0 otherwise), and set

$$\Delta_q = \left\{ \sum_{i=0}^{q} \lambda_i e_i \;\middle|\; \sum_{i=0}^{q} \lambda_i = 1, \lambda_i \geq 0 \right\},$$
$$\Delta_q^\ell = \left\{ \sum_{i=0}^{q} \lambda_i e_i \;\middle|\; \sum_{i=0}^{q} \lambda_i = 1, \lambda_i \geq 0, \lambda_\ell = 0 \right\}.$$

We also denote the inclusion by $\iota_q^\ell : \Delta_q^\ell \hookrightarrow \Delta_q$.

For a topological space X, a continuous map $\sigma : \Delta_q \to X$ is called a singular q-simplex, and let X_q be the set of q-simplices. We construct a K-vector space $C_q(X)$ as

$$C_q(X) = \mathrm{Span}_K\{\sigma \mid \sigma \in X_q\}.$$

18. A continuous map $f : X \to Y$ is said to be *homeomorphic* if $f : X \to Y$ is bijective and the inverse $f^{-1} : Y \to X$ is also continuous.

The boundary map $\partial_q : C_q(X) \to C_{q-1}(X)$ is defined by the linear extension of

$$\partial_q \sigma = \sum_{\ell=0}^{q} (-1)^\ell \sigma \circ \iota_q^\ell.$$

Even in this setting, we can show that $\partial_q \circ \partial_{q+1} = 0$, and hence the subspaces $Z_q(X) = \ker\partial_q$ and $B_q(X) = \mathrm{im}\partial_{q+1}$ satisfy $B_q(X) \subset Z_q(X)$. Then, the q-th (singular) *homology* is similarly defined by

$$H_q(X) = Z_q(X)/B_q(X).$$

It is known that, for a triangulable topological space, the homology of this definition is isomorphic to that defined in A.2.1. From this reason, we hereafter identify simplicial and singular homology.

The induced linear map in homology for an inclusion pair of topological space $Y \subset X$ is similarly defined as in A.2.1.

Appendix B. Total persistence

Let (M, d_M) be a triangulable compact metric space. For a Lipschitz function $f : M \to \mathbb{R}$, we define the degree-p total persistence over t by

$$\mathrm{Pers}_p(D_q(\mathbb{S}\mathrm{ub}(f)), t) = \sum_{\substack{x \in D_q(\mathbb{S}\mathrm{ub}(f)) \\ \mathrm{pers}(x) > t}} \mathrm{pers}(x)^p$$

for $0 \leq t \leq \mathrm{Amp}(f)$, where $\mathrm{Amp}(f) := \max_{\boldsymbol{x} \in M} f(\boldsymbol{x}) - \min_{\boldsymbol{x} \in M} f(\boldsymbol{x})$ is the amplitude of f. Let S be a triangulated simplicial complex of M by a homeomorphism $\vartheta : |S| \to M$. The diameter of a simplex $\sigma \in S$ and the mesh of the triangulation S are defined by $\mathrm{diam}(\sigma) = \max_{\boldsymbol{x}, \boldsymbol{y} \in \sigma} d_M(\vartheta(\boldsymbol{x}), \vartheta(\boldsymbol{y}))$ and $\mathrm{mesh}(S) = \max_{\sigma \in S} \mathrm{diam}(\sigma)$, respectively. Furthermore, let us set $N(r) = \min_{\mathrm{mesh}(S) \leq r} \mathrm{card}\,(S)$. Then, the degree-$p$ total persistence over t is bounded from above as follows:

Lemma 13 (Cohen-Steiner et al. (2010)) *Let M be a triangulable compact metric space and $f : M \to \mathbb{R}$ be a tame Lipschitz function. Then, $\mathrm{Pers}_p(D_q(\mathbb{S}\mathrm{ub}(f)), t)$ is bounded from above by*

$$t^p N\left(\frac{t}{\mathrm{Lip}(f)}\right) + p \int_{\varepsilon=t}^{\mathrm{Amp}(f)} N\left(\frac{\varepsilon}{\mathrm{Lip}(f)}\right) \varepsilon^{p-1} d\varepsilon,$$

where $\mathrm{Lip}(f)$ is the Lipschitz constant of f.

For a compact triangulable subspace M in $\mathbb{R}^d$, the number of d-cubes with length $r > 0$ covering M is bounded from above by $O(\frac{1}{r^d})$, and hence there exists some constant C_M depending only on M such that $N(r) \leq \frac{C_M}{r^d}$.

For $p > d$, we can find the upper bounds for the both terms as follows:

$$t^p N\left(\frac{t}{\mathrm{Lip}(f)}\right) \leq t^p C_M \frac{\mathrm{Lip}(f)^d}{t^d} \;\to\; 0 \quad (t \to 0)$$

and

$$p\int_{\varepsilon=t}^{\mathrm{Amp}(f)} N\left(\frac{\varepsilon}{\mathrm{Lip}(f)}\right)\varepsilon^{p-1}d\varepsilon \le \frac{p}{p-d}C_M\mathrm{Lip}(f)^d\mathrm{Amp}(f)^{p-d}.$$

Then, the upper bound of the total persistence $\mathrm{Pers}_p(D_q(\mathbb{S}\mathrm{ub}(f))) = \mathrm{Pers}_p(D_q(\mathbb{S}\mathrm{ub}(f)),0)$ is given as follows:

Lemma 14 *Let M be a triangulable compact subspace in $\mathbb{R}^d$ and $p > d$. For any Lipschitz function $f : M \to \mathbb{R}$,*

$$\mathrm{Pers}_p(D_q(\mathbb{S}\mathrm{ub}(f))) \le \frac{p}{p-d}C_M\mathrm{Lip}(f)^d\mathrm{Amp}(f)^{p-d},$$

where C_M is a constant depending only on M.

In the case of a finite subset $X \subset \mathbb{R}^d$, there always exists an R-ball M containing X for some $R > 0$, which is a triangulable compact subspace in $\mathbb{R}^d$. Moreover, by estimating $\mathrm{Lip}(\mathrm{dist}_X)^d\mathrm{Amp}(\mathrm{dist}_X)^{p-d}$, we show Lemma 4 as a corollary of Lemma 14:

Proof [Lemma 4] The Lipschitz constant of dist_X is 1, because, for any $\boldsymbol{x}, \boldsymbol{y} \in M$,

$$\begin{aligned}\mathrm{dist}_X(\boldsymbol{x}) - \mathrm{dist}_X(\boldsymbol{y}) &= \min_{\boldsymbol{x}_i\in X} d_M(\boldsymbol{x},\boldsymbol{x}_i) - \min_{\boldsymbol{x}_i\in X} d_M(\boldsymbol{y},\boldsymbol{x}_i)\\ &\le \min_{\boldsymbol{x}_i\in X}(d_M(\boldsymbol{x},\boldsymbol{y}) + d_M(\boldsymbol{y},\boldsymbol{x}_i)) - \min_{\boldsymbol{x}_i\in X} d_M(\boldsymbol{y},\boldsymbol{x}_i)\\ &= d_M(\boldsymbol{x},\boldsymbol{y}).\end{aligned}$$

Moreover,

$$\mathrm{Amp}(\mathrm{dist}_X) \le \mathrm{diam}(M) := \max_{\boldsymbol{x}_i,\boldsymbol{x}_j\in M} d_M(\boldsymbol{x}_i,\boldsymbol{x}_j),$$

because $\min_{\boldsymbol{x}\in M}\mathrm{dist}_X(\boldsymbol{x}) = 0$ and $\max_{\boldsymbol{x}\in M}\mathrm{dist}_X(\boldsymbol{x}) \le \mathrm{diam}(M)$. Thus, for some constant C_M depending only on M, we have

$$\begin{aligned}\mathrm{Pers}_p(D_q(X)) &= \mathrm{Pers}_p(D_q(\mathbb{S}\mathrm{ub}(\mathrm{dist}_X)))\\ &\le \frac{p}{p-d}C_M\mathrm{Lip}(\mathrm{dist}_X)^d\mathrm{Amp}(\mathrm{dist}_X)^{p-d}\\ &\le \frac{p}{p-d}C_M\mathrm{diam}(M)^{p-d}.\end{aligned}$$

■

For a persistence diagram $D = \{x_1, \ldots, x_n\}$, we construct a n-dimensional vector

$$v(D) := (\mathrm{pers}(x_1), \ldots, \mathrm{pers}(x_n)).$$

Then, the degree-p total persistence is represented as

$$\mathrm{Pers}_p(D) = \|v(D)\|_p^p,$$

where $\|\cdot\|_p$ denotes the ℓ^p-norm of $\mathbb{R}^n$. Since $\|v\|_q \le \|v\|_p$ $(v \in \mathbb{R}^n,\ 1 \le p \le q < \infty)$, we have

$$\mathrm{Pers}_q(D)^{\frac{1}{q}} = \|v(D)\|_q \le \|v(D)\|_p = \mathrm{Pers}_p(D)^{\frac{1}{p}}.$$

Proposition 15 *If $1 \le p \le q < \infty$ and $\mathrm{Pers}_p(D)$ is bounded from above, $\mathrm{Pers}_q(D)$ is also bounded from above.*

Appendix C. Lemmata for Section 3.2

Lemma 16 *For any $x, y \in \mathbb{R}^2$, $\|k_{\mathrm{G}}(\cdot,x) - k_{\mathrm{G}}(\cdot,y)\|_{\mathcal{H}_{k_{\mathrm{G}}}} \le \frac{\sqrt{2}}{\sigma}\|x-y\|_\infty$.*

Proof

$$\begin{aligned}\|k_{\mathrm{G}}(\cdot,x) - k_{\mathrm{G}}(\cdot,y)\|^2_{\mathcal{H}_{k_{\mathrm{G}}}} &= k_{\mathrm{G}}(x,x) + k_{\mathrm{G}}(y,y) - 2k_{\mathrm{G}}(x,y)\\ &= 1 + 1 - 2e^{-\frac{\|x-y\|^2}{2\sigma^2}}\\ &= 2\left(1 - e^{-\frac{\|x-y\|^2}{2\sigma^2}}\right)\\ &\le \frac{1}{\sigma^2}\|x-y\|^2 \qquad (25)\\ &\le \frac{2}{\sigma^2}\|x-y\|_\infty^2. \qquad (26)\end{aligned}$$

We have used the fact $1 - e^{-t} \le t$ $(t \in \mathbb{R})$ in Equation (25) and $\|x\|^2 \le 2\|x\|_\infty^2$ $(x \in \mathbb{R}^2)$ in Equation (26). ■

Lemma 17 *For any $x, y \in \mathbb{R}^2$, the difference of persistences $|\mathrm{pers}(x) - \mathrm{pers}(y)|$ is less than or equal to $2\|x-y\|_\infty$.*

Proof For $x = (x_1, x_2), y = (y_1, y_2)$, we have

$$\begin{aligned}|\mathrm{pers}(x) - \mathrm{pers}(y)| &= |(x_2 - x_1) - (y_2 - y_1)|\\ &\le |x_2 - y_2| + |x_1 - y_1|\\ &\le 2\|x-y\|_\infty.\end{aligned}$$

■

Lemma 18 *For any $x, y \in \mathbb{R}^2$, we have*

$$|w_{\mathrm{arc}}(x) - w_{\mathrm{arc}}(y)| \le 2pC\max\{\mathrm{pers}(x)^{p-1}, \mathrm{pers}(y)^{p-1}\}\|x-y\|_\infty.$$

Proof

$$\begin{aligned}&|w_{\mathrm{arc}}(x) - w_{\mathrm{arc}}(y)|\\ &= |\arctan(C\mathrm{pers}(x)^p) - \arctan(C\mathrm{pers}(y)^p)| \qquad (27)\\ &\le C|\mathrm{pers}(x)^p - \mathrm{pers}(y)^p|\\ &\le C|\mathrm{pers}(x) - \mathrm{pers}(y)|\,p\max\{\mathrm{pers}(x)^{p-1}, \mathrm{pers}(y)^{p-1}\} \qquad (28)\\ &\le 2pC\max\{\mathrm{pers}(x)^{p-1}, \mathrm{pers}(y)^{p-1}\}\|x-y\|_\infty. \qquad (29)\end{aligned}$$

We have used the fact that the Lipschitz constant of arctan is 1 in Equation (27),

$$\begin{aligned} s^p - t^p &= (s-t)(s^{p-1} + s^{p-2}t + \cdots + t^{p-1}) \\ &\leq (s-t)p\max\{s^{p-1}, t^{p-1}\} \end{aligned}$$

for any $s, t > 0$ in Equation (28), and Lemma 17 in Equation (29). ■

Appendix D. The number of the random Fourier features

Let $\mathcal{M}$ be a compact subset of $\mathbb{R}^2$ and $\delta > 0$ be a positive number, then it is known (Rahimi and Recht, 2007) that

$$\sup_{x,y\in\mathcal{M}} \left| \mathrm{Re}\left(\frac{1}{M_{\mathrm{rff}}} \sum_{a=1}^{M_{\mathrm{rff}}} \xi_{z_a}(x)\xi_{z_a}(y)^* \right) - e^{-\frac{\|x-y\|^2}{2\sigma^2}} \right| \leq \delta$$

where $\xi_{z_a}(x) = e^{\sqrt{-1}z_a^T x}$ and $M_{\mathrm{rff}} = \Omega(\frac{2}{\delta^2}\log\frac{\sqrt{2}\mathrm{diam}(\mathcal{M})}{\sigma\delta})$. For a collection $\mathcal{D} = \{D_\ell \mid \ell = 1, \ldots, n\}$ of persistence diagrams, the absolute error between the (k_{G}, w)-linear kernel $K_{\mathrm{L}}(D_i, D_j; k_{\mathrm{G}}, w)$ and its random Fourier feature approximation is given by

$$\begin{aligned} &\left| \sum_{x\in D_i}\sum_{y\in D_j} w(x)w(y)\mathrm{Re}\left(\frac{1}{M_{\mathrm{rff}}} \sum_{a=1}^{M_{\mathrm{rff}}} \xi_{z_a}(x)\xi_{z_a}(y)^* \right) - \sum_{x\in D_i}\sum_{y\in D_j} w(x)w(y)e^{-\frac{\|x-y\|^2}{2\sigma^2}} \right| \\ &\leq \sum_{x\in D_i} w(x) \sum_{y\in D_j} w(y) \left| \mathrm{Re}\left(\frac{1}{M_{\mathrm{rff}}} \sum_{a=1}^{M_{\mathrm{rff}}} \xi_{z_a}(x)\xi_{z_a}(y)^* \right) - e^{-\frac{\|x-y\|^2}{2\sigma^2}} \right|. \end{aligned} \tag{30}$$

In order to make the Equation (30) bounded by an arbitrary $\varepsilon > 0$, M_{rff} is given by $\Omega(\frac{2W_{i,j}^2}{\varepsilon^2}\log\frac{\sqrt{2}W_{i,j}\mathrm{diam}(\mathcal{M})}{\sigma\varepsilon})$ where $W_{i,j} := \sum_{x\in D_i} w(x)\sum_{y\in D_j} w(y)$. In this case, we can define the subset $\mathcal{M}$ by $\bigcup_{\ell=1}^n D_\ell$. When we calculate several $K_{i,j} := K_{\mathrm{L}}(D_i, D_j; k_{\mathrm{G}}, w)$ of Section 4.3 without approximation, we observed $K_{i,j} \approx 10^8$.[19] Since the true values are huge, we consider 5% relative error for Equation (30) and set $\varepsilon := 0.05K_{i,j}$. Then,

$$\frac{2W_{i,j}^2}{\varepsilon^2}\log\frac{\sqrt{2}W_{i,j}\mathrm{diam}(\mathcal{M})}{\sigma\varepsilon} = \frac{800W_{i,j}^2}{K_{i,j}^2}\log\frac{20\sqrt{2}W_{i,j}\mathrm{diam}(\mathcal{M})}{\sigma K_{i,j}} =: M_{i,j}.$$

In Section 4.3 and Section 4.4, we observed $\frac{W_{i,j}}{K_{i,j}} \approx 2.5$ and $\frac{\mathrm{diam}(\mathcal{M})}{\sigma} \approx 10$ from several computation without approximation, and $M_{i,j} \approx 800 \cdot (2.5)^2\log(20\sqrt{2}\cdot 3\cdot 10) \approx 3\cdot 10^4$. Thus, the approximation (30) with $M_{\mathrm{rff}} = 10^5$, which is used in our experiments, gives 95% accuracy in the sense of the relative error.

19. Even though a single $K_{i,j}$ can be computed in several hours on MacBook Pro, 2.6 GHz Intel Core i5, 8 GB 1600 MHz DDR3, the total $\frac{1}{2}n(n-1)$ computations of $K_{i,j}$ for the Gram matrix cause huge computational time.

References

Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18(8):1–35, 2017.

Anonymous. What is random packing? *Nature*, 239:488–489, 1972.

Ulrich Bauer, Michael Kerber, Jan Reininghaus, and Hubert Wagner. *Mathematical Software – ICMS 2014: 4th International Congress, Seoul, South Korea, August 5-9, 2014. Proceedings*, chapter PHAT – Persistent Homology Algorithms Toolbox, pages 137–143. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

Peter Bubenik. Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 16(1):77–102, 2015.

Peter Bubenik and Paweł Dłotko. A persistence landscapes toolbox for topological statistics. *Journal of Symbolic Computation*, 78:91–114, 2017.

Peter Bubenik and Jonathan A Scott. Categorification of persistent homology. *Discrete & Computational Geometry*, 51(3):600–627, 2014.

Zixuan Cang, Lin Mu, Kedi Wu, Kristopher Opron, Kelin Xia, and Guo-Wei Wei. A topological approach for protein classification. *Molecular Based Mathematical Biology*, 3 (1), 2015.

Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46 (2):255–308, 2009.

Gunnar Carlsson, Tigran Ishkhanov, Vin de Silva, and Afra Zomorodian. On the local behavior of spaces of natural images. *International journal of computer vision*, 76(1): 1–12, 2008.

Frédéric Chazal, Vin de Silva, and Steve Oudot. Persistence stability for geometric complexes. *Geometriae Dedicata*, 173(1):193–214, 2014.

Frédéric Chazal, Marc Glisse, Catherine Labruère, and Bertrand Michel. Convergence rates for persistence diagram estimation in topological data analysis. *Journal of Machine Learning Research*, 16:3603–3635, 2015.

David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.

David Cohen-Steiner, Herbert Edelsbrunner, John Harer, and Yuriy Mileyko. Lipschitz functions have l_p-stable persistence. *Foundations of computational mathematics*, 10(2): 127–139, 2010.

Tran Kai Frank Da, Sébastien Loriot, and Mariette Yvinec. 3D alpha shapes. In *CGAL User and Reference Manual*. CGAL Editorial Board, 4.7 edition, 2015. URL `http://doc.cgal.org/4.7/Manual/packages.html#PkgAlphaShapes3Summary`.

Vin de Silva and Robert Ghrist. Coverage in sensor networks via persistent homology. *Algebraic & Geometric Topology*, 7(1):339–358, 2007.

Joseph Diestel and J Jerry Uhl Jr. Vector measures. with a foreword by bj pettis. mathematical surveys, no. 15. *American Mathematical Society, Providence, RI*, 56:12216, 1977.

Pietro Donatini, Patrizio Frosini, and Alberto Lovato. Size functions for signature recognition. In *Vision Geometry VII*, volume 3454, pages 178–184. International Society for Optics and Photonics, 1998.

Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. *Discrete and Computational Geometry*, 28(4):511–533, 2002.

Stephen R. Elliott. *Physics of amorphous materials (2nd)*. Longman London; New York, 1990.

Brittany Terese Fasy, Fabrizio Lecci, Alessandro Rinaldo, Larry Wasserman, Sivaraman Balakrishnan, and Aarti Singh. Confidence sets for persistence diagrams. *The Annals of Statistics*, 42(6):2301–2339, 2014.

Nicolas Francois, Mohammad Saadatfar, R Cruikshank, and A Sheppard. Geometrical frustration in amorphous and partially crystallized packings of spheres. *Physical review letters*, 111(14):148001, 2013.

Marcio Gameiro, Yasuaki Hiraoka, Shunsuke Izumi, Miroslav Kramar, Konstantin Mischaikow, and Vidit Nanda. A topological measurement of protein compressibility. *Japan Journal of Industrial and Applied Mathematics*, 32(1):1–17, 2015.

Neville G Greaves and Sabyasachi Sen. Inorganic glasses, glass-forming liquids and amorphizing solids. *Advances in Physics*, 56(1):1–166, 2007.

Arthur Gretton, Kenji Fukumizu, Choon H. Teo, Le Song, Bernhard Schölkopf, and Alex J Smola. A kernel statistical test of independence. In *Advances in Neural Information Processing Systems*, pages 585–592, 2007.

Zaid Harchaoui, Eric Moulines, and Francis R. Bach. Kernel change-point analysis. In *Advances in Neural Information Processing Systems*, pages 609–616, 2009.

Allen Hatcher. *Algebraic topology*. Cambridge University Press, 2002.

Yasuaki Hiraoka, Takenobu Nakamura, Akihiko Hirata, Emerson G Escolar, Kaname Matsue, and Yasumasa Nishiura. Hierarchical structures of amorphous solids characterized by persistent homology. *Proceedings of the National Academy of Sciences*, 113(26):7035–7040, 2016.

Peter M Kasson, Afra Zomorodian, Sanghyun Park, Nina Singhal, Leonidas J Guibas, and Vijay S Pande. Persistent voids: a new structural metric for membrane fusion. *Bioinformatics*, 23(14):1753–1759, 2007.

Genki Kusano, Yasuaki Hiraoka, and Kenji Fukumizu. Persistence weighted Gaussian kernel for topological data analysis. In *International Conference on Machine Learning*, pages 2004–2013, 2016.

Hyekyoung Lee, Moo K Chung, Hyejin Kang, Bung-Nyun Kim, and Dong Soo Lee. Discriminative persistent homology of brain networks. In *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pages 841–844. IEEE, 2011.

David Lopez-Paz, Krikamol Muandet, and Benjamin Recht. The randomized causation coefficient. *Journal of Machine Learning Research*, 16(16):2901–2907, 2015.

Krikamol Muandet, Kenji Fukumizu, Francesco Dinuzzo, and Bernhard Schölkopf. Learning from distributions via support measure machines. In *Advances in neural information processing systems*, pages 10–18, 2012.

Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Schölkopf. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends in Machine Learning*, 10(1-2):1–141, 2017.

Takenobu Nakamura, Yasuaki Hiraoka, Akihiko Hirata, Emerson G Escolar, and Yasumasa Nishiura. Persistent homology and many-body atomic structure for medium-range order in the glass. *Nanotechnology*, 26(304001), 2015.

Giovanni Petri, Paul Expert, Federico Turkheimer, Robin Carhart-Harris, David Nutt, Peter J Hellyer, and Francesco Vaccarino. Homological scaffolds of brain functional networks. *Journal of The Royal Society Interface*, 11(101):20140873, 2014.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2007.

Jan Reininghaus, Stefan Huber, Ulrich Bauer, and Roland Kwitt. A stable multi-scale kernel for topological machine learning. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 4741–4748, 2015.

Vanessa Robins and Katharine Turner. Principal component analysis of persistent homology rank functions with case studies of spatial point patterns, sphere packing and colloids. *Physica D: Nonlinear Phenomena*, 334:99–117, 2016.

Mohammad Saadatfar, Hiroshi Takeuchi, Vanessa Robins, Nicolas Francois, and Yasuaki Hiraoka. Pore configuration landscape of granular crystallization. *Nature Communications*, 8:15082 EP, 2017.

Gurjeet Singh, Facundo Memoli, Tigran Ishkhanov, Guillermo Sapiro, Gunnar Carlsson, and Dario L Ringach. Topological analysis of population activity in visual cortex. *Journal of vision*, 8(8):11, 2008.

Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A Hilbert space embedding for distributions. In *Algorithmic Learning Theory: 18th International Conference*, pages 13–31. Springer, 2007.

Le Song, Kenji Fukumizu, and Arthur Gretton. Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models. *IEEE Signal Processing Magazine*, 30(4):98–111, 2013.

Bharath K. Sriperumbudur, Kenji Fukumizu, and Gert R. G. Lanckriet. Universality, characteristic kernels and RKHS embedding of measures. *Journal of Machine Learning Research*, 12:2389–2410, 2011.

Zoltán Szabó, Bharath Sriperumbudur, Barnabás Póczos, and Arthur Gretton. Learning theory for distribution regression. *Journal of Machine Learning Research*, 17(152):1–40, 2016.

Salvatore Torquato, Thomas M Truskett, and Pablo G Debenedetti. Is random close packing of spheres well defined? *Physical review letters*, 84(10):2064, 2000.

Kelin Xia and Guo-Wei Wei. Persistent homology analysis of protein structure, flexibility, and folding. *International journal for numerical methods in biomedical engineering*, 30 (8):814–844, 2014.

Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.

Journal of Machine Learning Research 18 (2018) 1-5 Submitted 4/17; Revised 3/18; Published 4/18

Pycobra: A Python Toolbox for Ensemble Learning and Visualisation

Benjamin Guedj benjamin.guedj@inria.fr
Modal project-team, Lille - Nord Europe research center
Inria, France

Bhargav Srinivasa Desikan bhargav.srinivasa-desikan@inria.fr
Modal project-team, Lille - Nord Europe research center
Inria, France

Editor: Geoff Holmes

Abstract

We introduce `pycobra`, a Python library devoted to ensemble learning (regression and classification) and visualisation. Its main assets are the implementation of several ensemble learning algorithms, a flexible and generic interface to compare and blend any existing machine learning algorithm available in Python libraries (as long as a `predict` method is given), and visualisation tools such as Voronoi tessellations. `pycobra` is fully `scikit-learn` compatible and is released under the MIT open-source license. `pycobra` can be downloaded from the Python Package Index (PyPi) and Machine Learning Open Source Software (MLOSS). The current version (along with Jupyter notebooks, extensive documentation, and continuous integration tests) is available at https://github.com/bhargavvader/pycobra and official documentation website is https://modal.lille.inria.fr/pycobra.

Keywords: ensemble methods, machine learning, Voronoi tesselation, Python, open source software

1. Introduction

Combined statistical procedures, also known as ensemble or aggregation methods, are very popular in Machine Learning competitions – the celebrated Netflix Challenge (as are repeatedly Kaggle competitions) was won by an ensemble technique (see for example Bell and Koren, 2007, for a discussion). In a nutshell, ensemble methods combine different predictors (each trained on the same dataset) to improve performance. `scikit-learn` offers implementations of some ensemble methods but it does not ship with a specific analysis toolbox for aggregation. `pycobra` attempts to fill this gap by providing to the `scikit-learn` environment a toolbox to analyse and visualise the different preliminary predictors (also called machines hereafter), in addition to providing pythonic implementations of several ensemble algorithms (the COBRA algorithm introduced by Biau et al., 2016; the Exponential Weighted Aggregate (EWA) introduced by Vovk, 1990; a COBRA-flavored majority vote inspired by Mojirsheibani, 1999). All these algorithms are supported by oracle bounds which prove that the risk of the aggregated predictor is upper-bounded by the smallest risk of the initial pool of predictors, up to a remainder term which decays to zero (typically at a rate $\mathcal{O}(1/\sqrt{n})$ or faster).

COBRA (standing for COmBined Regression Alternative) is a nonlinear ensemble method designed for regression problems. To predict the response for a new data point, COBRA operates in two steps. First, it retains data points for which the prediction made by preliminary machines is close (in the Euclidean sense) to the prediction made for the new point. This is called the *consensus step*. Second, the final prediction is then formed by averaging responses over the retained points' indices. COBRA outputs a predictor which outperforms any combination of the preliminary predictors (as shown by see Theorem 2.1 in Biau et al., 2016). We describe the COBRA algorithm as implemented in `pycobra` in Algorithm 1. Exponential weights have been used for a long time in statistical learning theory (Vovk, 1990). The EWA algorithm implemented in Pycobra is inspired by the description of Dalalyan and Tsybakov (2007). EWA amounts to forming an exponentially weighted average of preliminary predictors, where the weights include a measure of each predictor's performance. COBRA has been inspired by the work of Mojirsheibani (1999) in classification, and therefore `pycobra` includes a classification version of COBRA called `ClassifierCobra`. The consensus step shares a similar philosophy to what is done in COBRA regression but the final prediction is delivered by a majority vote among the labels of the retained data points.

`pycobra` allows the user to gauge the performance of the preliminary predictors used in the aggregation, with built-in methods to easily plot boxplots and QQ-plots. A salient feature of `pycobra` is using Voronoi tessellations for generic visualisation. By implementing ensemble algorithms which were not represented in the Python machine learning community, and providing a variety of tools to visualise and analyse their behavior and performance, we present to the machine learning open source community a toolbox designed for ensemble learning and visualisation.

2. The `pycobra` library

Our toolbox is written in Python and uses `NumPy` (Walt et al., 2011) and `scikit-learn` (Pedregosa et al., 2011) for computation and linear algebra operations. `Matplotlib` (Hunter, 2007) is used for visualisation purposes, and `SciPy` (Jones et al., 2001) is used to help create Voronoi tessellations. Tutorials and examples are created using Jupyter IPython notebooks (Pérez and Granger, 2007). Currently `pycobra` supports any machine learning algorithm with a `predict` method, casting our procedure into a very flexible and generic framework. By default, `pycobra` relies on `scikit-learn` implementations of Lasso, Random Forest, Decision Trees and Ridge regression for the `EWA` and `COBRA` implementations. As for `ClassifierCobra`, `scikit-learn` implementations of SVM classifier, KNN, Decision Tree classifier, and a classifier which implements regularised linear models with SGD (SGDClassifier object). `pycobra` itself and all software it relies on is open source, with no dependence on proprietary software. Algorithm 1 presents the pseudo-code of the COBRA implementation. All the pycobra estimators are `scikit-learn` compatible and can be used as part of the existing `scikit-learn` ecosystem, such as `[GridSearchCV]` and `[Pipeline]`. While hyperparameter initialisation is systematically done using `scikit-learn`'s `GridSearchCV`, `pycobra`'s `Diagnostics` class allows us to compare between different combinations of the constituent predictors and data-splitting, among other basic parameters.

Algorithm 1: The original COBRA algorithm from Biau et al. (2016).

Data: input vector $\mathbf{X}$, `epsilon`, `alpha`, `basic-machines`, `training-set-responses`, `training-set`
`training-set` is the set composed of all `data_point` and the responses. `training-set-responses` is the set composed of the responses.
Result: prediction $\mathbf{Y}$
for *machine j in basic-machines* **do**
  `machine_set` = ;
  # *machine_set is a dictionary mapping each machine M to a set* `machine_set`$[M]$;
  `pred` $= r_j(\mathbf{X})$; # *where $r_j(x)$ denotes the prediction made by machine j at point x*;
  for *response in training-set-responses* **do**
    if $|$`data_point` $-$ `pred`$| \leq$ `epsilon`, collect `response` in `machine_set`$[M]$;
  end
end
array = [ ] ;
for *data_point in training-set* **do**
  if at least `alpha` machines out of M have collected `data_point` in their `set`, store point in array ;
end
result = average(array) ;

The `visualisation` class allows the user to compare all the machines used to create aggregates, as well as visualise the results, for all `pycobra` estimators. `pycobra` ships with [a notebook on visualisation] to illustrate this.

QQ-plots and boxplots. Once all the basic machines are initialized and trained, the user can easily compare their performance with boxplots and QQ-plots. All the plotting details are handled by both the `diagnostics` and `visualisation` classes. An example is given in Figure 1.

Visualisation through Voronoi tessellations. Voronoi tessellations can be used to visualise the selection of optimal machines for COBRA, as well as visualising outputs from clustering algorithms, as illustrated by the [tutorial notebook]. An example is reproduced Figure 2.

3. Project Focus

Community-based development. We intend `pycobra` to be an ongoing collaborative project. To that matter, it is referenced on [Python Package Index (PyPi)] and [Machine Learning Open Source Software]. Moreover, `pycobra` is under active development and available on [GitHub] to promote collaborative programming, issue tracking and idea discussions.

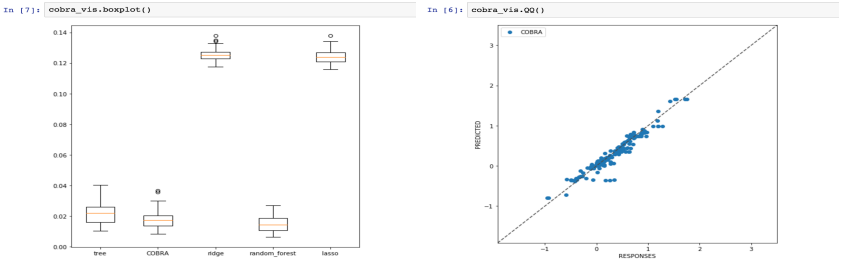


Figure 1: Assessing the performance of regression machines and COBRA.

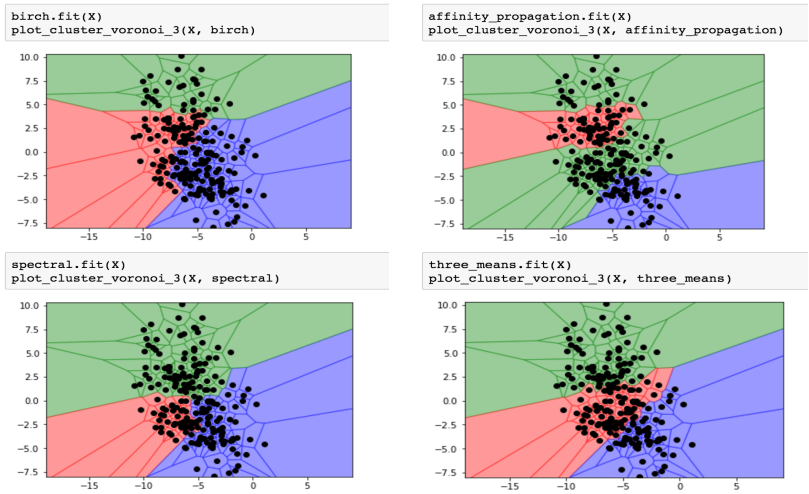


Figure 2: Visualising clustering through Voronoi tessellations.

Documentation and Jupyter Notebooks. A consistent API documentation is provided, along with an additional installation guide and examples. The documentation website is available at https://modal.lille.inria.fr/pycobra. The [notebooks directory] contains Jupyter notebooks which serve as both a documentation tool and a tutorial resource. These notebooks cover use-cases of Pycobra, from solving regression and classification problems to using Voronoi tesselations.

Ease of use and quality assurance. Ensemble learning with `pycobra` is as simple as loading trained `scikit-learn` machines – or any machine which has a `predict` method. Visualising involves little to no parameters, and after loading machines it is straightforward to analyse their performance on a particular dataset. In order to ensure code quality, a set of unit tests is provided for all classes in `pycobra`, and continuous integration via [Travis CI] ensures all commits are tested. The package follows the PEP8 convention, keeping the code easy to read and contribute to.

4. Conclusion and Future Work

The future of `pycobra` would be to grow its user base by adding new ways to add predictors, as well as further implement ensemble learning techniques and visualisation tools. Statistical aggregation and ensemble learning are an important part of the machine learning literature and are widely used by practitioners, yet it seems under-represented in the machine learning open source community. By creating `pycobra` and releasing it to the community, we intend to enrich the existing ecosystem with ensemble algorithms, a generic toolbox for ensemble learning on-the-go, along with analysis and visualisation tools.

References

Robert M. Bell and Yehuda Koren. Lessons from the Netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2):75–79, 2007.

Gérard Biau, Aurélie Fischer, Benjamin Guedj, and James D. Malley. COBRA: A combined regression strategy. *Journal of Multivariate Analysis*, 146:18–28, 2016.

Arnak Dalalyan and Alexandre Tsybakov. Aggregation by exponential weighting and sharp oracle inequalities. In *Computational Learning theory (COLT)*, pages 97–111. Springer, 2007.

John D. Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001. URL http://www.scipy.org/.

Majid Mojirsheibani. Combining classifiers via discretization. *Journal of the American Statistical Association*, 94(446):600–609, 1999.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

Fernando Pérez and Brian E. Granger. IPython: a system for interactive scientific computing. *Computing in Science & Engineering*, 9(3):21–29, May 2007. ISSN 1521-9615. doi: 10.1109/MCSE.2007.53. URL http://ipython.org.

Volodimir G. Vovk. Aggregating strategies. In *Computational Learning theory (COLT)*, pages 371–386. Morgan Kaufmann Publishers Inc., 1990.

Stéfan van der Walt, S. Chris Colbert, and Gaël Varoquaux. The NumPy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

Journal of Machine Learning Research 18 (2018) 1-5 Submitted 2/16; Revised 10/17; Published 4/18

KeLP: a Kernel-based Learning Platform

Simone Filice filice@info.uniroma2.it
DICII, University of Roma, Tor Vergata, Italy
Giuseppe Castellucci castellucci@ing.uniroma2.it
DIE, University of Roma, Tor Vergata, Italy
Giovanni Da San Martino gmartino@hbku.edu.qa
Qatar Computing Research Institute, HKBU, Qatar
Alessandro Moschitti* amosch@amazon.com
Amazon
Danilo Croce croce@info.uniroma2.it
Roberto Basili basili@info.uniroma2.it
DII, University of Roma, Tor Vergata, Italy

Editor: Cheng Soon Ong

Abstract

KeLP is a Java framework that enables fast and easy implementation of kernel functions over discrete data, such as strings, trees or graphs and their combination with standard vectorial kernels. Additionally, it provides several kernel-based algorithms, e.g., online and batch kernel machines for classification, regression and clustering, and a Java environment for easy implementation of new algorithms. KeLP is a versatile toolkit, very appealing both to experts and practitioners of machine learning and Java language programming, who can find extensive documentation, tutorials and examples of increasing complexity on the accompanying website. Interestingly, KeLP can be also used without any knowledge of Java programming through command line tools and JSON/XML interfaces enabling the declaration and instantiation of articulated learning models using simple templates. Finally, the extensive use of modularity and interfaces in KeLP enables developers to easily extend it with their own kernels and algorithms.

Keywords: Kernel Machines, Structured Data and Kernels, Java Framework.

1. Introduction

Kernel methods for discrete structures (Shawe-Taylor and Cristianini, 2004) are popular and effective techniques for the design of learning algorithms on non-vectorial data, such as strings (Lodhi et al., 2002), trees (Collins and Duffy, 2002; Moschitti, 2006; Aiolli et al., 2009; Croce et al., 2011; Annesi et al., 2014) and graphs (Gärtner, 2003; Borgwardt and Kriegel, 2005; Shervashidze, 2011). These kernels are very valuable to model complex relations in real-world applications, where data naturally has a structured form, e.g., strings and graphs are used to represent DNA and chemical compounds, or parse trees can encode syntactic and semantic information expressed in text.

However, current software for structural kernels is mainly limited to specific research, and is often not made publicly available or easily adaptable to new application domains. SVM-Light-TK toolkit by Moschitti (2006) is one of few exceptions that provides the user with different string and tree kernels but no graph kernels. It is written in C language thus extending it with new kernels can be costly, especially when new data structures are required. This may also prevent non programmers to use it for their specific applications.

In designing KeLP, we have capitalized on our previous experience with SVM-Light-TK and other toolkits to foster the reuse of previous software and models as well as their extendibility. We provide a software platform for learning on structured data, which is both easy to use for unexperienced users and easily extendable for developers. KeLP includes many standard kernel algorithms for classification, regression and clustering as well as popular kernel functions for strings, trees and graphs. Additionally, it includes kernel functions for modeling relations between pairs of objects, which are, e.g., required in paraphrasing detection, textual entailment and question answering (Moschitti and Zanzotto, 2007; Filice et al., 2015; Tymoshenko and Moschitti, 2015). Most importantly, new data structures, models, algorithms and kernels can be easily added on top of the previous code, facilitating and promoting the development of a library of kernel-based algorithms for structured data.

The KeLP source code is distributed under the terms of Apache 2.0 License. No additional software is required to be installed in order to use it, the Apache Maven project management tool resolves all module dependencies automatically. We also provide and maintain a website with updated tutorials and documentation.

2. The KeLP Framework: an Overview

KeLP is written in Java and uses three different Maven projects to logically separate its three main components: (*i*) the framework backbone implements classification, regression and clustering algorithms operating on vector-based kernels. These core modules along with SVMs[1] are always part of any framework instantiation. (*ii*) Additional-algorithm packages, e.g., online kernel machines, Nyström method (Williams and Seeger, 2001) and label sequence learning (Altun et al., 2003), and (*iii*) additional-kernel packages, which include kernel functions for sequences, trees and graphs. A complete and up-to-date list of algorithms and kernel functions, a full Javadoc API documentation in PDF, and tutorials for both end-users and developers are hosted on the KeLP website, `http://www.kelp-ml.org`.

2.1 Machine Learning Algorithms

Learning algorithms in KeLP are implemented following *implementation contracts* provided by specific Java interfaces for different scenarios, i.e., classification, regression and clustering, according to two main learning paradigms, i.e., batch and online. New learning algorithms can implement these interfaces, thus becoming fully integrated with the other library functions. More in detail: (*i*) The `ClassificationLearningAlgorithm` interface supports the definition of classification learning methods, such as SVMs (Chang and Lin, 2011) or the Dual Coordinate Descent (Hsieh et al., 2008). (*ii*) The `RegressionLearningAlgorithm` interface supports the definition of regressors, such as ϵ-SVR (Chang and Lin, 2011). (*iii*) The `ClusteringAlgorithm` interface enables the implementation of clustering algorithms, such as (Kulis et al., 2005). (*iv*) The `OnlineLearningAlgorithm` interface supports the definition of online learning algorithms, e.g., Passive Aggressive (Crammer et al., 2006), or the Soft Confidence Weighted (Wang et al., 2012) algorithms. Finally, (*v*) the `MetaLearningAlgorithm` interface enables the design of committees, such as the multi-classification schemas, e.g., One-VS-One and One-VS-All.

*. Professor at the University of Trento, Italy.

1. We include it because of its wide use.

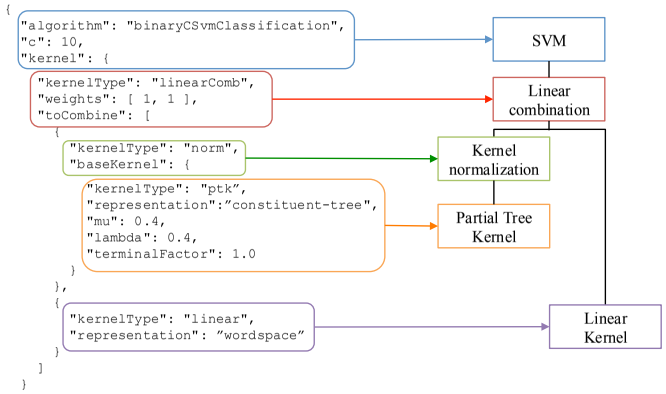


Figure 1: A JSON description of a SVM classifier.

2.2 Data Representation

In KeLP, data is represented by the `Example` class, which is constituted by (*i*) a set of `Labels` and (*ii*) a set of `Representations`. The former enables the design of single or multi-label classifiers and multi-variate regressors. The latter model examples in terms of vectors (e.g., `DenseVector` and `SparseVector`) or structures (e.g., `SequenceRepresentation`, `TreeRepresentation` or `GraphRepresentation`). In particular, kernels can be defined over examples encoded by multiple representations (e.g., multiple parse trees, strings, graphs and feature vectors). This makes the experimentation with multiple kernel combinations easy, just requiring negligible changes in the code or the JSON description (see Section 2.4), without the need of modifying the input data sets. Additionally, the examples can be combined in more complex structures, e.g., `ExamplePair`, useful to learn relations between objects, e.g., pairs representing question and answer text in QA, or text and hypothesis in textual entailment tasks. Building other types of data format is extremely simple, e.g., KeLP includes the SVM-Light-TK input format for trees and provides many scripts to use the popular gspan format for graphs (and indirectly for the 111 openBabel formats[2]).

2.3 Building Kernels from Kernels

KeLP enables (*i*) kernel composition, i.e., $K_{ab}(s_1, s_2) = (\phi_a \circ \phi_b)(s_1) \cdot (\phi_a \circ \phi_b)(s_2)$ from $K_a(s_1, s_2) = \phi_a(s_1) \cdot \phi_a(s_2)$ and $K_b(s_1, s_2) = \phi_b(s_1) \cdot \phi_b(s_2)$; and (*ii*) kernel combinations, e.g., $\lambda_1 K_a(s_1, s_2) + \lambda_2 K_b(s_1, s_2) \times K_a(s_1, s_2)$. These operations are coded using three abstractions of the `Kernel` class: (*i*) `DirectKernel` directly operates on a specified `Representation` object, derived from the `Example` object (e.g., implementing kernels for vectors, sequences, trees and graphs). (*ii*) The `KernelComposition` class composes `Kernel` objects, e.g., `PolynomialKernel`, `RBFKernel` and `NormalizationKernel`. (*iii*) `KernelCombination` class enables the combination of different `Kernels`, e.g., the `LinearKernelCombination` class applies a weighted kernel sum. (*iv*) `KernelOnPair` class operates on `ExamplePair`, e.g., to learn similarity functions between sentences (Filice et al., 2015) or to implement ranking algorithms with the `PreferenceKernel` class.

2. `http://openbabel.org`


```java
public static void run(String trainPath, String testPath, String learningAlgoPath){
        //Define (load) the learning algorithm  (see the JSON in Fig. 1)
        JacksonSerializerWrapper serializer = new JacksonSerializerWrapper();
        ClassificationLearningAlgorithm learningAlgo;
        learningAlgo = serializer.readValue(new File(learningAlgoPath),
                                        ClassificationLearningAlgorithm.class);
        //Load the datasets
        SimpleDataset trainDataset = new SimpleDataset();
        trainDataset.populate(trainPath);
        SimpleDataset testDataset = new SimpleDataset();
        testDataset.populate(testPath);
        //Learn the classifier
        List<Label> classes = trainDataset.getClassificationLabels();
        learningAlgo.setLabels(classes);
        learningAlgo.learn(trainDataset);
        //Classify and Evaluate
        Classifier classifier = learningAlgo.getPredictionFunction();
        Evaluator evaluator = new MulticlassClassificationEvaluator(classes);
        for(Example ex: testDataset.getExamples()){
                evaluator.addCount(ex, classifier.predict(ex));
        }
        System.out.println("ACC:" + evaluator.getPerformanceMeasure("accuracy"));
}
```

Listing 1: The Java instantiation (and evaluation) of the SVM classifier specified in Figure 1.

2.4 A User-friendly Interfacing with JSON

Each object, kernel function or algorithm, is serializable in JSON or XML. Thus, new algorithms can be implemented with a JSON description exploiting already implemented building blocks. The JSON interpreter of KeLP instantiates the corresponding objects without requiring any Java coding. Note that once a new kernel or learning algorithm is coded in Java, it will also be automatically available in the JSON format. Thus, it can be combined and composed with any kernel and algorithm available in KeLP by simply using JSON specifications. For example, Figure 1 provides a JSON description of an SVM classifier using a linear combination of a tree kernel with a linear kernel. The procedure for training and evaluating such classifier can be written in less than 20 code lines, as shown in Listing 1. Additionally, new kernels can be designed by combining JSON files and used in the framework by executing terminal commands (runnable jars). This enables experimenting with most KeLP features without writing any Java code.

3. Related Work and Conclusions

Most kernel-based software assumes data is represented by feature vectors (Hall et al., 2009; Chang and Lin, 2011; Abeel et al., 2009). Notable exceptions are SVM-Light-TK (Moschitti, 2006) and JKernelMachines (Picard et al., 2013). SVM-Light-TK is entirely written in C language and its main feature is the high computation speed. Unfortunately, C does not allow for fast prototyping of new kernel functions and machines. In contrast, KeLP enables fast and easy implementation of new kernel methods. JKernelMachines is a Java based package primarily designed to deal with custom kernels that cannot be easily found in standard libraries. However, many features offered by KeLP are not available in JKernelMachines, e.g., tree and graph kernels and regression algorithms. Moreover, KeLP supports easier composition and combination of kernels and learning algorithms.

References

Thomas Abeel, Yves Van de Peer, and Yvan Saeys. Java-ml: A machine learning library. *Journal of Machine Learning Research*, 10:931–934, 2009.

Fabio Aiolli, Giovanni Da San Martino, and Alessandro Sperduti. Route Kernels for Trees. In *Proceedings of ICML 09*, pages 17–24, Montreal, Quebec, Canada, 2009. ACM Press.

Yasemin Altun, Ioannis Tsochantaridis, and Thomas Hofmann. Hidden markov support vector machines. In *Proceedings of ICML 2003*, pages 4–11, Menlo Park, CA, USA, August 2003.

Paolo Annesi, Danilo Croce, and Roberto Basili. Semantic compositionality in tree kernels. In *Proc. of CIKM 2014*, pages 1029–1038, New York, NY, USA, 2014. ACM.

Karsten M Borgwardt and Hans-Peter Kriegel. Shortest-Path Kernels on Graphs. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 74–81, 2005.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

Michael Collins and Nigel Duffy. Convolution kernels for natural language. In *Advances in Neural Information Processing Systems*, pages 625–632. MIT Press, 2002.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *JMLR*, 7:551–585, December 2006.

Danilo Croce, Alessandro Moschitti, and Roberto Basili. Structured lexical similarity via convolution kernels on dependency trees. In *EMNLP*, pages 1034–1046, 2011.

Simone Filice, Giovanni Da San Martino, and Alessandro Moschitti. Structural representations for learning relations between pairs of texts. In *Proc. of ACL*, 2015.

Thomas Gärtner. A survey of kernels for structured data. *ACM SIGKDD Explorations Newsletter*, 5(1):49–58, 2003.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *sigkdd explor.*, 11(1), 2009.

Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S. Sathiya Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear svm. In *Proc. of ICML*, pages 408–415. ACM, 2008.

Brian Kulis, Sugato Basu, Inderjit Dhillon, and Raymond Mooney. Semi-supervised graph clustering: A kernel approach. In *Proc. of ICML*, pages 457–464. ACM, 2005.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, , and Chris Watkins. Text classification using string kernels. *journal of Machine Learning Research*, pages 419–444, 2002.

Alessandro Moschitti. Efficient convolution kernels for dependency and constituent syntactic trees. In *Proceedings of ECML'06*, pages 318–329, Berlin, Germany, 2006.

Alessandro Moschitti and Fabio Massimo Zanzotto. Fast and effective kernels for relational learning from texts. In *ICML'07*, pages 649–656. ACM, 2007.

David Picard, Nicolas Thome, and Matthieu Cord. Jkernelmachines: A simple framework for kernel machines. *Journal of Machine Learning Research*, 14:1417–1421, 2013.

John Shawe-Taylor and Nello Cristianini. *LaTeX User's Guide and Document Reference Manual*. Kernel Methods for Pattern Analysis, Cambridge University Press, 2004.

Nino Shervashidze. Weisfeiler-lehman graph kernels. *The Journal of Machine Learning*, 12:2539–2561, 2011. URL `http://dl.acm.org/citation.cfm?id=2078187`.

Kateryna Tymoshenko and Alessandro Moschitti. Assessing the impact of syntactic and semantic structures for answer passages reranking. In *Proc. of CIKM*, pages 1451–1460. ACM, 2015.

Jialei Wang, Peilin Zhao, and Steven C. Hoi. Exact soft confidence-weighted learning. In John Langford and Joelle Pineau, editors, *Proceedings of ICML*, pages 121–128, 2012.

Christopher K. I. Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *Proceedings of NIPS 2000*, 2001.

Journal of Machine Learning Research 18 (2018) 1-28 Submitted 5/17; Revised 9/17; Published 4/18

Uncovering Causality from Multivariate Hawkes Integrated Cumulants

Massil Achab MASSIL.ACHAB@M4X.ORG
Centre de Mathématiques Appliquées, Ecole polytechnique, Palaiseau, France

Emmanuel Bacry BACRY@CEREMADE.DAUPHINE.FR
Centre de Recherche en Mathématique de la Décision, Université Paris-Dauphine, Paris, France
Centre de Mathématiques Appliquées, Ecole polytechnique, Palaiseau, France

Stéphane Gaïffas GAIFFAS@MATH.UNIV-PARIS-DIDEROT.FR
Laboratoire de Probabilités et Modèles Aléatoires, Université Paris-Diderot, Paris, France

Iacopo Mastromatteo IACOPO.MASTROMATTEO@CFM.FR
Research-Execution, Capital Fund Management, Paris, France

Jean-François Muzy MUZY@UNIV-CORSE.FR
Laboratoire Sciences Pour l'Environnement, Université de Corse, Corte, France

Editor: Edoardo M. Airoldi

Abstract

We design a new nonparametric method that allows one to estimate the matrix of integrated kernels of a multivariate Hawkes process. This matrix not only encodes the mutual influences of each node of the process, but also disentangles the causality relationships between them. Our approach is the first that leads to an estimation of this matrix *without any parametric modeling and estimation of the kernels themselves*. As a consequence, it can give an estimation of causality relationships between nodes (or users), based on their activity timestamps (on a social network for instance), without knowing or estimating the shape of the activities lifetime. For that purpose, we introduce a moment matching method that fits the second-order and the third-order integrated cumulants of the process. A theoretical analysis allows us to prove that this new estimation technique is consistent. Moreover, we show, on numerical experiments, that our approach is indeed very robust with respect to the shape of the kernels and gives appealing results on the MemeTracker database and on financial order book data.

Keywords: Hawkes Process, Causality Inference, Cumulants, Generalized Method of Moments

1. Introduction

In many applications, one needs to deal with data containing a very large number of irregular timestamped events that are recorded in continuous time. These events can reflect, for instance, the activity of users on a social network, see Subrahmanian et al. (2016), the high-frequency variations of signals in finance, see Bacry et al. (2015), the earthquakes and aftershocks in geophysics, see Ogata (1998), the crime activity, see Mohler et al. (2011) or the position of genes in genomics, see Reynaud-Bouret and Schbath (2010). The succession of the precise timestamps carries a great deal of information about the dynamics of the underlying systems. In this context, multidimensional counting processes based models play a paramount role. Within this framework, an important task is to recover the mutual influence of the nodes (i.e., the different components of the counting process), by leveraging on their timestamp patterns, see, for instance, Bacry and Muzy (2016); Lemonnier and Vayatis (2014); Lewis and Mohler (2011); Zhou et al. (2013a); Gomez-Rodriguez et al. (2013); Farajtabar et al. (2015); Xu et al. (2016).

Consider a set of nodes $I = \{1, \dots, d\}$. For each $i \in I$, we observe a set Z^i of *events*, where each $\tau \in Z^i$ labels the occurrence time of an event related to the activity of i. The events of all nodes can be represented as a vector of counting processes $\boldsymbol{N}_t = [N_t^1 \cdots N_t^d]^\top$, where N_t^i counts the number of events of node i until time $t \in \mathbb{R}^+$, namely $N_t^i = \sum_{\tau \in Z^i} \mathbb{1}_{\{\tau \leq t\}}$. The vector of stochastic intensities $\boldsymbol{\lambda}_t = [\lambda_t^1 \cdots \lambda_t^d]^\top$ associated with the multivariate counting process $\boldsymbol{N}_t$ is defined as

$$\lambda_t^i = \lim_{dt \to 0} \frac{\mathbb{P}(N_{t+dt}^i - N_t^i = 1 | \mathcal{F}_t)}{dt}$$

for $i \in I$, where the filtration $\mathcal{F}_t$ encodes the information available up to time t. The coordinate λ_t^i gives the expected instantaneous rate of event occurrence at time t for node i. The vector $\boldsymbol{\lambda}_t$ characterizes the distribution of $\boldsymbol{N}_t$, see Daley and Vere-Jones (2003), and patterns in the events time-series can be captured by structuring these intensities.

The Hawkes process introduced in Hawkes (1971) corresponds to an autoregressive structure of the intensities in order to capture self-excitation and cross-excitation of nodes, which is a phenomenon typically observed, for instance, in social networks, see for instance Crane and Sornette (2008). Namely, $\boldsymbol{N}_t$ is called a *Hawkes point process* if the stochastic intensities can be written as

$$\lambda_t^i = \mu^i + \sum_{j=1}^d \int_0^t \phi^{ij}(t - t') dN_{t'}^j,$$

where $\mu^i \in \mathbb{R}^+$ is an exogenous intensity and ϕ^{ij} are positive, integrable and causal *i.e.* with support in $\mathbb{R}_+$ (so that effects don't happen before their cause) functions called *kernels* encoding the impact of an action by node j on the activity of node i. Note that when all kernels are zero, the process is a simple homogeneous multivariate Poisson process.

Most of the literature uses a parametric approach for estimating the kernels. Without a doubt, the most popular parametrization form is the exponential kernel $\phi^{ij}(t) = \alpha_{ij}\beta_{ij}e^{-\beta_{ij}t}$ because it definitely simplifies the inference algorithm (e.g., the complexity needed for computing the likelihood is much smaller). When d is large, in order to reduce the number of parameters, some authors choose to arbitrarily share the kernel shapes across the different nodes. Thus, for instance, in Yang and Zha (2013); Zhou et al. (2013b); Farajtabar et al. (2015), they choose $\phi^{ij}(t) = \alpha_{ij}h(t)$ with $\alpha_{ij} \in \mathbb{R}^+$ quantifies the intensity of the influence of j on i and $h(t)$ a (normalized) function that characterizes the time-profile of this influence and that is *shared* by all couples of nodes (i, j) (most often, it is chosen to be either exponential $h(t) = \beta e^{-\beta t}$ or power law $h(t) = \beta t^{-(\beta+1)}$). Both approaches are, most of the time, non-realistic. On the one hand there is a priori no reason for assuming that the time-profile of the influence of a node j on a node i does not depend on the pair (i, j). On the other

hand, assuming an exponential shape or a power law shape for a kernel arbitrarily imposes an event impact that is always instantly maximal and that can only decrease with time, while in practice, there may exist a latency between an event and its maximal impact.

In order to have more flexibility on the shape of the kernels, nonparametric estimation can be considered. Expectation-Maximization algorithms can be found in Lewis and Mohler (2011) (for $d = 1$) or in Zhou et al. (2013a) ($d > 1$). An alternative method is proposed in Bacry and Muzy (2016) where the nonparametric estimation is formulated as a numerical solving of a Wiener-Hopf equation. Another nonparametric strategy considers a decomposition of kernels on a dictionary of function $h_1, \dots, h_K$, namely $\phi^{ij}(t) = \sum_{k=1}^{K} a_k^{ij} h_k(t)$, where the coefficients a_k^{ij} are estimated, see Hansen et al. (2015); Lemonnier and Vayatis (2014) and Xu et al. (2016), where group-lasso is used to induce a sparsity pattern on the coefficients a_k^{ij} that is shared across $k = 1, \dots, K$.

Such methods are computationally-intensive when d is large, since they rely on likelihood maximization or least squares minimization within an over-parametrized space in order to gain flexibility on the shape of the kernels. This is problematic, since the original motivation for the use of Hawkes processes is to estimate the influence and causality of nodes, the knowledge of the full parametrization of the model being of little interest for causality purpose.

Our paper solves this problem with a different and more direct approach. Instead of trying to estimate the kernels ϕ^{ij}, we focus on the direct estimation of their *integrals*. Namely, we want to estimate the matrix $\boldsymbol{G} = [g^{ij}]$ where

$$g^{ij} = \int_0^{+\infty} \phi^{ij}(u)\, du \geq 0 \;\text{ for }\; 1 \leq i, j \leq d. \tag{1}$$

As it can be seen from the cluster representation of Hawkes processes (Hawkes and Oakes (1974)), this integral represents the mean total number of events of type i directly triggered by an event of type j, and then encodes a notion of *causality*. Actually, as detailed below (see Section 2.1), such integral can be related to the Granger causality (Granger (1969)).

The main idea of the method we developed in this paper is to estimate the matrix $\boldsymbol{G}$ directly using a matching cumulants (or moments) method. Apart from the mean, we shall use second and third-order cumulants which correspond respectively to centered second and third-order moments. We first compute an estimation $\widetilde{\boldsymbol{M}}$ of these centered moments $M(\boldsymbol{G})$ (they are uniquely defined by $\boldsymbol{G}$). Then, we look for a matrix $\widehat{\boldsymbol{G}}$ that minimizes the L^2 error $\|M(\widehat{\boldsymbol{G}}) - \widetilde{\boldsymbol{M}}\|^2$. Thus the integral matrix $\widehat{\boldsymbol{G}}$ is directly estimated without making hardly any assumptions on the shape the involved kernels. As it will be shown, this approach turns out to be particularly robust to the kernel shapes, which is not the case of all previous Hawkes-based approaches that aim causality recovery. We call this method NPHC (Non Parametric Hawkes Cumulant), since our approach is of nonparametric nature. We provide a theoretical analysis that proves the consistency of the NPHC estimator. Our proof is based on ideas from the theory of Generalized Method of Moments (GMM) but requires an original technical trick since our setting strongly departs from the standard parametric statistics with i.i.d observations. Note that moment and cumulant matching techniques proved particularly powerful for latent topic models, in particular Latent Dirichlet Allocation, see Podosinnikova et al. (2015). A small set of previous works, namely Da Fonseca and Zaatour (2014); Aït-Sahalia et al. (2010), already used method of moments with Hawkes processes, but only in a parametric setting. Our work

is the first to consider such an approach for a nonparametric counting processes framework.

The paper is organized as follows: in Section 2, we provide the background on the integrated kernels and the integrated cumulants of the Hawkes process. We then introduce the method, investigate its complexity and explain the consistency result we prove. In Section 3, we estimate the matrix of Hawkes kernels' integrals for various simulated datasets and for real datasets, namely the MemeTracker database and financial order book data. We then provide in Appendix B the technical details skipped in the previous parts and the proof of our consistency result. Section 4 contains concluding remarks.

2. NPHC: The Non Parametric Hawkes Cumulant method

In this Section, we provide the background on integrals of Hawkes kernels and integrals of Hawkes cumulants. We then explain how the NPHC method enables estimating $\boldsymbol{G}$.

2.1 Branching structure and Granger causality

From the definition of Hawkes process as a Poisson cluster process, see Jovanović et al. (2015) or Hawkes and Oakes (1974), g^{ij} can be simply interpreted as the average total number of events of node i whose *direct* ancestor is a given event of node j (by direct we mean that interactions mediated by any other intermediate event are not counted). In that respect, $\boldsymbol{G}$ not only describes the mutual influences between nodes, but it also quantifies their *direct causal* relationships. Namely, introducing the counting function $N_t^{i \leftarrow j}$ that counts the number of events of i whose direct ancestor is an event of j, we know from Bacry et al. (2015) that

$$\mathbb{E}[dN_t^{i \leftarrow j}] = g^{ij} \mathbb{E}[dN_t^j] = g^{ij} \Lambda^j dt, \tag{2}$$

where we introduced Λ^i as the intensity expectation, namely satisfying $\mathbb{E}[dN_t^i] = \Lambda^i dt$. Note that Λ^i does not depend on time by stationarity of $\boldsymbol{N}_t$, which is known to hold under the *stability condition* $\|\boldsymbol{G}\| < 1$, where $\|\boldsymbol{G}\|$ stands for the spectral norm of $\boldsymbol{G}$. In particular, this condition implies the non-singularity of $\boldsymbol{I_d} - \boldsymbol{G}$.

Since the question of a *real causality* is too complex in general, see Imbens and Rubin (2015); Pearl (2009), most econometricians agreed on the simpler definition of Granger causality Granger (1969). Its mathematical formulation is a statistical hypothesis test: X causes Y *in the sense of Granger causality* if forecasting future values of Y is more successful while taking X past values into account.

Definition 1 (Granger causality for time series) *Given two time series X and Y, we denote $\mathcal{H}(t)$ the set of all information available prior to t, $\mathcal{H}_{-X}(t)$ the previous set in which information coming from X is excluded, and A an arbitrary non-empty set. We say that X Granger-causes Y if*

$$\mathbb{P}[Y(t+1) \in A | \mathcal{H}(t)] \neq \mathbb{P}[Y(t+1) \in A | \mathcal{H}_{-X}(t)].$$

Existing works mainly focus on learning Granger causality for time series, see Arnold et al. (2007); Eichler (2012); Basu et al. (2015), such as vector autoregressive models (VAR), where Granger causality is formulated as a statistical test of the VAR coefficients. In Eichler et al. (2016), the authors extend the definition of Granger (non-)causality to the case of Hawkes processes.

Definition 2 (Granger causality for Hawkes processes) *For $\boldsymbol{N}_t$ a multivariate Hawkes process, N_t^j does not Granger-cause N_t^i w.r.t $\boldsymbol{N}_t$ if and only if $\phi^{ij}(u) = 0$ for $u \in \mathbb{R}^+$.*

Since the kernels take positive values, the latter condition is equivalent to $\int_0^\infty \phi^{ij}(u)du = 0$. In the following, we'll refer to *learning the kernels' integrals* as *uncovering causality* since each integral encodes the notion of Granger causality, and is also linked to the number of events directly caused from a node to another node, as described above at Eq. (2).

2.2 Integrated cumulants of the Hawkes process

A general formula for the integral of the cumulants of a multivariate Hawkes process is provided in Jovanović et al. (2015). As explained below, for the purpose of our method, we only need to consider cumulants up to the third order. Given $1 \leq i, j, k \leq d$, the first three integrated cumulants of the Hawkes process can be defined as follows thanks to stationarity:

$$\Lambda^i dt = \mathbb{E}(dN_t^i) \tag{3}$$

$$C^{ij}dt = \int_{\tau\in\mathbb{R}} \Big(\mathbb{E}(dN_t^i dN_{t+\tau}^j) - \mathbb{E}(dN_t^i)\mathbb{E}(dN_{t+\tau}^j)\Big) \tag{4}$$

$$\begin{aligned} K^{ijk}dt = \iint_{\tau,\tau'\in\mathbb{R}^2} \Big(&\mathbb{E}(dN_t^i dN_{t+\tau}^j dN_{t+\tau'}^k) + 2\mathbb{E}(dN_t^i)\mathbb{E}(dN_{t+\tau}^j)\mathbb{E}(dN_{t+\tau'}^k) \\ &- \mathbb{E}(dN_t^i dN_{t+\tau}^j)\mathbb{E}(dN_{t+\tau'}^k) - \mathbb{E}(dN_t^i dN_{t+\tau'}^k)\mathbb{E}(dN_{t+\tau}^j) - \mathbb{E}(dN_{t+\tau}^j dN_{t+\tau'}^k)\mathbb{E}(dN_t^i)\Big), \end{aligned} \tag{5}$$

where Eq. (3) is the mean intensity of the Hawkes process, the second-order cumulant (4) refers to the integrated covariance density matrix and the third-order cumulant (5) measures the skewness of $\boldsymbol{N}_t$. Using the martingale representation from Bacry and Muzy (2016) or the Poisson cluster process representation from Jovanović et al. (2015), one can obtain an explicit relationship between these integrated cumulants and the matrix $\boldsymbol{G}$. If one sets

$$\boldsymbol{R} = (\boldsymbol{I_d} - \boldsymbol{G})^{-1}, \tag{6}$$

straightforward computations (see Appendix B) lead to the following identities:

$$\Lambda^i = \sum_{m=1}^d R^{im}\mu^m \tag{7}$$

$$C^{ij} = \sum_{m=1}^d \Lambda^m R^{im} R^{jm} \tag{8}$$

$$K^{ijk} = \sum_{m=1}^d (R^{im}R^{jm}C^{km} + R^{im}C^{jm}R^{km} + C^{im}R^{jm}R^{km} - 2\Lambda^m R^{im}R^{jm}R^{km}). \tag{9}$$

Equations (8) and (9) are proved in Appendix B. Our strategy is to use a convenient subset of Eqs. (3), (4) and (5) to define $\boldsymbol{M}$, while we use Eqs. (7), (8) and (9) in order to construct the operator that maps a candidate matrix $\boldsymbol{R}$ to the corresponding cumulants $M(\boldsymbol{R})$. By looking for $\widehat{\boldsymbol{R}}$ that minimizes $\boldsymbol{R} \mapsto \|M(\boldsymbol{R}) - \widehat{\boldsymbol{M}}\|^2$, we obtain, as illustrated below, good recovery of the ground truth matrix $\boldsymbol{G}$ using Equation (6).

The simplest case $d = 1$ has been considered in Hardiman and Bouchaud (2014), where it is shown that one can choose $M = \{C^{11}\}$ in order to compute the kernel integral. Eq. (8) then reduces to a simple second-order equation that has a unique solution in $\boldsymbol{R}$ (and consequently a unique $\boldsymbol{G}$) that accounts for the stability condition ($\|\boldsymbol{G}\| < 1$).

Unfortunately, for $d > 1$, the choice $M = \{C^{ij}\}_{1\leq i\leq j\leq d}$ is not sufficient to uniquely determine the kernels integrals. In fact, the integrated covariance matrix provides $d(d+1)/2$ independent coefficients, while d^2 parameters are needed. It is straightforward to show that the remaining $d(d-1)/2$ conditions can be encoded in an orthogonal matrix $\boldsymbol{O}$, reflecting the fact that Eq. (8) is invariant under the change $\boldsymbol{R} \to \boldsymbol{OR}$, so that the system is under-determined.

Our approach relies on using the third order cumulant tensor $\boldsymbol{K} = [K^{ijk}]$ which contains $(d^3 + 3d^2 + 2d)/6 > d^2$ independent coefficients that are sufficient to uniquely fix the matrix $\boldsymbol{G}$. This can be justified intuitively as follows: while the integrated covariance only contains symmetric information, and is thus unable to provide causal information, *the skewness given by the third order cumulant in the estimation procedure can break the symmetry between past and future so as to uniquely fix $\boldsymbol{G}$*. Thus, our algorithm consists of selecting d^2 third-order cumulant components, namely $M = \{K^{iij}\}_{1\leq i,j\leq d}$. Note that the choice of K^{iij} is arbitrary, our method and the theory developed below is unchanged for any choice of d^2 distinct components among the d^3 third-order cumulant. In particular, we define the estimator of $\boldsymbol{R}$ as $\widehat{\boldsymbol{R}} \in \operatorname{argmin}_{\boldsymbol{R}} \mathcal{L}(\boldsymbol{R})$, where

$$\mathcal{L}(\boldsymbol{R}) = (1-\kappa)\|\boldsymbol{K^c}(\boldsymbol{R}) - \widehat{\boldsymbol{K^c}}\|_2^2 + \kappa\|\boldsymbol{C}(\boldsymbol{R}) - \widehat{\boldsymbol{C}}\|_2^2, \tag{10}$$

where $\|\cdot\|_2$ stands for the Frobenius norm, $\boldsymbol{K^c} = \{K^{iij}\}_{1\leq i,j\leq d}$ is the matrix obtained by the contraction of the tensor $\boldsymbol{K}$ to d^2 indices, $\boldsymbol{C}$ is the covariance matrix, while $\widehat{\boldsymbol{K^c}}$ and $\widehat{\boldsymbol{C}}$ are their respective estimators, see Equations (12), (13) below. It is noteworthy that the above mean square error approach can be seen as a peculiar Generalized Method of Moments (GMM), see Hall (2005). This framework allows us to determine the optimal weighting matrix involved in the loss function. However, this approach is unusable in practice, since the associated complexity is too high. Indeed, since we have d^2 parameters, this matrix has d^4 coefficients and GMM calls for computing its inverse leading to a $O(d^6)$ complexity. In this work, we use the coefficient κ to scale the two terms, as

$$\kappa = \frac{\|\widehat{\boldsymbol{K^c}}\|_2^2}{\|\widehat{\boldsymbol{K^c}}\|_2^2 + \|\widehat{\boldsymbol{C}}\|_2^2},$$

see Section B.4 for an explanation about the link between κ and the weighting matrix. Finally, the estimator of $\boldsymbol{G}$ is straightforwardly obtained as

$$\widehat{\boldsymbol{G}} = \boldsymbol{I_d} - \widehat{\boldsymbol{R}}^{-1},$$

from the inversion of Eq. (6). Let us mention an important point: the matrix inversion in the previous formula is not the bottleneck of the algorithm. Indeed, its has a complexity $O(d^3)$ that is cheap compared to the computation of the cumulants when $n = \max_i |Z^i| \gg d$, which is the typical scaling satisfied in applications. Solving the considered problem on a larger scale, say $d \gg 10^3$, is an open question, even with state-of-the-art parametric and nonparametric approaches, see for instance Zhou et al. (2013a); Xu et al. (2016); Zhou et al. (2013b); Bacry and Muzy (2016), where the number of components d in experiments is always around 100 or smaller. Note that, actually, our approach leads to a *much faster* algorithm than the considered state-of-the-art baselines, see Tables 1–4 from Section 3 below.

2.3 Estimation of the integrated cumulants

In this section we present explicit formulas to estimate the three moment-based quantities listed in the previous section, namely, $\mathbf{\Lambda}$, $\boldsymbol{C}$ and $\boldsymbol{K}$. We first assume there exists $H > 0$ such that the truncation from $(-\infty, +\infty)$ to $[-H, H]$ of the domain of integration of the quantities appearing in Eqs. (4) and (5), introduces only a small error. In practice, this amounts to neglecting border effects in the covariance density and in the skewness density that is a good approximation if the support of the kernel $\phi^{ij}(t)$ is smaller than H and the spectral norm $\|\boldsymbol{G}\|$ satisfies $\|\boldsymbol{G}\| < 1$.
In this case, given a realization of a stationary Hawkes process $\{\boldsymbol{N}_t : t \in [0, T]\}$, as shown in Appendix B, we can write the estimators of the first three cumulants (3), (4) and (5) as

$$\widehat{\Lambda}^i = \frac{1}{T} \sum_{\tau \in Z^i} 1 = \frac{N_T^i}{T} \tag{11}$$

$$\widehat{C}^{ij} = \frac{1}{T} \sum_{\tau \in Z^i} \left(N_{\tau+H}^j - N_{\tau-H}^j - 2H\widehat{\Lambda}^j \right) \tag{12}$$

$$\begin{aligned} \widehat{K}^{ijk} = & \frac{1}{T} \sum_{\tau \in Z^i} \left(N_{\tau+H}^j - N_{\tau-H}^j - 2H\widehat{\Lambda}^j \right) \cdot \left(N_{\tau+H}^k - N_{\tau-H}^k - 2H\widehat{\Lambda}^k \right) \\ & - \frac{\widehat{\Lambda}^i}{T} \sum_{\tau \in Z^j} \sum_{\tau' \in Z^k} (2H - |\tau' - \tau|)^+ + 4H^2 \widehat{\Lambda}^i \widehat{\Lambda}^j \widehat{\Lambda}^k. \end{aligned} \tag{13}$$

Let us mention the following facts.

Bias. While the first cumulant $\hat{\Lambda}^i$ is an unbiased estimator of Λ^i, the other estimators $\widehat{C}^{ij}$ and $\widehat{K}^{ijk}$ introduce a bias. However, as we will show, in practice this bias is small and hardly affects numerical estimations (see Section 3). This is confirmed by our theoretical analysis, which proves that if H does not grow too fast compared to T, then these estimated cumulants are consistent estimators of the theoretical cumulants (see Section 2.6).

Complexity. The computations of all the estimators of the first, second and third-order cumulants have complexity respectively $O(nd)$, $O(nd^2)$ and $O(nd^3)$, where $n = \max_i |Z^i|$. However, our algorithm requires a lot less than that: it computes only d^2 third-order terms, of the form $\widehat{K}^{iij}$, leaving us with only $O(nd^2)$ operations to perform.

Symmetry. While the values of Λ^i, C^{ij} and K^{ijk} are symmetric under permutation of the indices, their estimators are generally not symmetric. We have thus chosen to symmetrize the estimators by averaging their values over permutations of the indices. Worst case is for the estimator of $\boldsymbol{K^c}$, which involves only an extra factor of 2 in the complexity.

2.4 The NPHC algorithm

The objective to minimize in Equation (10) is non-convex. More precisely, the loss function is a polynomial of $\boldsymbol{R}$ of degree 6. However, the expectations of cumulants $\mathbf{\Lambda}$ and $\boldsymbol{C}$ defined in Eq. (4) and (5) that appear in the definition of $\mathcal{L}(\boldsymbol{R})$ are unknown and should be replaced with $\widehat{\mathbf{\Lambda}}$ and $\widehat{\boldsymbol{C}}$. We denote $\widetilde{\mathcal{L}}(\boldsymbol{R})$ the objective function, where the expectations of cumulants Λ^i and C^{ij} have been replaced with their estimators in the right-hand side of Eqs. (8) and (9):

$$\widetilde{\mathcal{L}}(\boldsymbol{R}) = (1-\kappa)\|\boldsymbol{R}^{\odot 2}\widehat{\boldsymbol{C}}^\top + 2[\boldsymbol{R} \odot (\widehat{\boldsymbol{C}} - \boldsymbol{R}\widehat{\boldsymbol{L}})]\boldsymbol{R}^\top - \widehat{\boldsymbol{K^c}}\|_2^2 + \kappa\|\boldsymbol{R}\widehat{\boldsymbol{L}}\boldsymbol{R}^\top - \widehat{\boldsymbol{C}}\|_2^2 \tag{14}$$

As explained in Choromanska et al. (2015), the loss function of a typical multilayer neural network with simple nonlinearities can be expressed as a polynomial function of the weights in the network, whose degree is the number of layers. Since the loss function of NPHC writes as a polynomial of degree 6, we expect good results using optimization methods designed to train deep multilayer neural networks. We use AdaGrad from Duchi et al. (2011), a variant of the Stochastic Gradient Descent with adaptive learning rates. AdaGrad scales the learning rates coordinate-wise using the online variance of the previous gradients, in order to incorporate second-order information during training. As detailed in Section 2.5, the optimization step is negligible compared to the computation of the cumulants whenever $n = \max_i |Z^i| \gg d$, which is the typical scaling in applications. The NPHC method is summarized schematically in Algorithm 1.

Algorithm 1 Non Parametric Hawkes Cumulant method

Input: $\boldsymbol{N}_t$
Output: $\widehat{\boldsymbol{G}}$

1: Estimate $\widehat{\Lambda}^i$, $\widehat{C}^{ij}$, $\widehat{K}^{iij}$ from Eqs. (11, 12, 13)
2: Design $\widetilde{\mathcal{L}}(\boldsymbol{R})$ using the computed estimators.
3: Minimize numerically $\widetilde{\mathcal{L}}(\boldsymbol{R})$ so as to obtain $\widehat{\boldsymbol{R}}$
4: Return $\widehat{\boldsymbol{G}} = \boldsymbol{I_d} - \widehat{\boldsymbol{R}}^{-1}$.

Our problem being non-convex, the choice of the starting point has a major effect on the convergence. Here, the key is to notice that the matrices $\boldsymbol{R}$ that match Equation (8) write $\boldsymbol{C}^{1/2}\boldsymbol{O}\boldsymbol{L}^{-1/2}$, with $\boldsymbol{L} = \mathrm{diag}(\mathbf{\Lambda})$ and $\boldsymbol{O}$ an orthogonal matrix. Our starting point is then simply chosen by setting $\boldsymbol{O} = \boldsymbol{I_d}$ in the previous formula, leading to nice convergence results. Even though our main concern is to retrieve the matrix $\boldsymbol{G}$, let us notice we can also obtain an estimation of the baseline intensities' from Eq. (3), which leads to $\widehat{\boldsymbol{\mu}} = \widehat{\boldsymbol{R}}^{-1}\widehat{\mathbf{\Lambda}}$. An efficient implementation of this algorithm with TensorFlow, see Abadi et al. (2016), is available on GitHub: `https://github.com/achab/nphc`.

The optimization problem can be regularized by adding to the function $\mathcal{L}(\boldsymbol{R})$ a regularizing term of the form $\lambda N(\boldsymbol{G})$ that encodes a prior assumption on the structure of $\boldsymbol{G}$. As long as $\boldsymbol{R}$ matches Equation (8) the penalty term can be written as a function of $\boldsymbol{R}$ since $\lambda N(\boldsymbol{G}) = \lambda N(\boldsymbol{I_d} - \boldsymbol{C}^{-1}\boldsymbol{L}\boldsymbol{R}^\top)$. Since the algorithms we compare our method to optimize different objective functions (negative log-likelihood, least-squares, etc.), adding $\lambda N(\boldsymbol{G})$ with the same λ to these functions would trigger different behaviors. Then, in the rest of the document we focus on the unregularized problem, for which we prove the convergence's consistency in the Section 2.6.

2.5 Complexity of the algorithm

Compared with existing state-of-the-art methods to estimate the kernel functions, e.g., the ordinary differential equations-based (ODE) algorithm in Zhou et al. (2013a), the Granger Causality-based algorithm in Xu et al. (2016), the ADM4 algorithm in Zhou et al. (2013b), and the Wiener-Hopf-based algorithm in Bacry and Muzy (2016), our method has a very competitive complexity. This can be understood by the fact that those methods estimate the kernel functions, while in NPHC we only estimate their integrals. The ODE-based algorithm is an EM algorithm that parametrizes the kernel function with M basis functions, each being discretized to L points. The basis functions are updated after solving M Euler-Lagrange equations. If n denotes the maximum number of events per component (i.e. $n = \max_{1 \le i \le d} |Z^i|$) then the complexity of one iteration of the algorithm is

$O(Mn^3d^2 + ML(nd + n^2))$. The Granger Causality-based algorithm is similar to the previous one, without the update of the basis functions, that are Gaussian kernels. The complexity per iteration is $O(Mn^3d^2)$. The algorithm ADM4 is similar to the two algorithms above, as EM algorithm as well, with only one exponential kernel as basis function. The complexity per iteration is then $O(n^3d^2)$. The Wiener-Hopf-based algorithm is not iterative, on the contrary to the previous ones. It first computes the empirical conditional laws on many points, and then invert the Wiener-Hopf system, leading to a $O(nd^2L + d^4L^3)$ computation. Similarly, our method first computes the integrated cumulants, then minimize the objective function with N_{iter} iterations, and invert the resulting matrix $\widehat{\boldsymbol{R}}$ to obtain $\widehat{\boldsymbol{G}}$. In the end, the complexity of the NPHC method is $O(nd^2 + N_{\text{iter}}d^3)$. According to this analysis, summarized in Table 1 below, one can see that in the regime $n \gg d$, the NPHC method outperforms all the other ones.

Table 1: Complexity of state-of-the-art methods. NPHC's complexity is very low, especially in the regime $n \gg d$.

Method	Total complexity
ODE Zhou et al. (2013a)	$O(N_{\text{iter}}M(n^3d^2 + L(nd + n^2)))$
GC Xu et al. (2016)	$O(N_{\text{iter}}Mn^3d^2)$
ADM4 Zhou et al. (2013b)	$O(N_{\text{iter}}n^3d^2)$
WH Bacry and Muzy (2016)	$O(nd^2L + d^4L^3)$
NPHC	$O(nd^2 + N_{\text{iter}}d^3)$

2.6 Theoretical guarantee: consistency

The NPHC method can be phrased using the framework of the Generalized Method of Moments (GMM). GMM is a generic method for estimating parameters in statistical models. In order to apply GMM, we have to find a vector-valued function $g(X, \theta)$ of the data, where X is distributed with respect to a distribution $\mathbb{P}_{\theta_0}$, which satisfies the *moment condition*: $\mathbb{E}[g(X, \theta)] = 0$ if and only if $\theta = \theta_0$, where θ_0 is the "ground truth" value of the parameter. Based on i.i.d. observed copies $x_1, \dots, x_n$ of X, the GMM method minimizes the norm of the empirical mean over n samples, $\|\frac{1}{n}\sum_{i=1}^n g(x_i, \theta)\|$, as a function of θ, to obtain an estimate of θ_0.

In the theoretical analysis of NPHC, we use ideas from the consistency proof of the GMM, but the proof actually relies on very different arguments. Indeed, the integrated cumulants estimators used in NPHC are not unbiased, as the theory of GMM requires, but asymptotically unbiased. Moreover, the setting considered here, where data consists of a single realization $\{\boldsymbol{N}_t\}$ of a Hawkes process strongly departs from the standard i.i.d setting. Our approach is therefore based on the GMM idea but the proof is actually not using the theory of GMM.

In the following, we use the subscript T to refer to quantities that only depend on the process $(\boldsymbol{N_t})$ in the interval $[0, T]$ (e.g., the truncation term H_T, the estimated integrated covariance $\widehat{\boldsymbol{C}}_T$ or the estimated kernel norm matrix $\widehat{\boldsymbol{G}}_T$). In the next equation, $\odot$ stands for the Hadamard product and $\odot 2$ stands for the entrywise square of a matrix. We denote $\boldsymbol{G}_0 = \boldsymbol{I_d} - \boldsymbol{R}_0^{-1}$ the true value of $\boldsymbol{G}$, and

the $\mathbb{R}^{2d\times d}$ valued vector functions

$$g_0(\boldsymbol{R}) = \begin{bmatrix} \boldsymbol{C} - \boldsymbol{R}\boldsymbol{L}\boldsymbol{R}^\top \\ \boldsymbol{K^c} - \boldsymbol{R}^{\odot 2}\boldsymbol{C}^\top - 2[\boldsymbol{R} \odot (\boldsymbol{C} - \boldsymbol{R}\boldsymbol{L})]\boldsymbol{R}^\top \end{bmatrix}$$

$$\widehat{g}_T(\boldsymbol{R}) = \begin{bmatrix} \widehat{\boldsymbol{C}}_T - \boldsymbol{R}\widehat{\boldsymbol{L}}_T\boldsymbol{R}^\top \\ \widehat{\boldsymbol{K^c_T}} - \boldsymbol{R}^{\odot 2}\widehat{\boldsymbol{C}}_T^\top - 2[\boldsymbol{R} \odot (\widehat{\boldsymbol{C}}_T - \boldsymbol{R}\widehat{\boldsymbol{L}}_T)]\boldsymbol{R}^\top \end{bmatrix}.$$

Using these notations, $\widetilde{\mathcal{L}}_T(\boldsymbol{R})$ can be seen as the weighted squared Frobenius norm of $\widehat{g}_T(\boldsymbol{R})$. Moreover, when $T \to +\infty$, one has $\widehat{g}_T(\boldsymbol{R}) \xrightarrow{\mathbb{P}} g_0(\boldsymbol{R})$ under the conditions of the following theorem, where $\xrightarrow{\mathbb{P}}$ stands for convergence in probability.

Theorem 3 (Consistency of NPHC) *Suppose that $(\boldsymbol{N_t})$ is observed on $\mathbb{R}^+$ and assume that*

1. *$g_0(\boldsymbol{R}) = 0$ if and only if $\boldsymbol{R} = \boldsymbol{R}_0$;*
2. *$\boldsymbol{R} \in \Theta$, where Θ is a compact set;*
3. *the spectral radius of the kernel norm matrix satisfies $\|\boldsymbol{G}_0\| < 1$;*
4. *$H_T \to \infty$ and $H_T^2/T \to 0$.*

Then

$$\widehat{\boldsymbol{G}}_T = \boldsymbol{I_d} - \left(\arg\min_{\boldsymbol{R}\in\Theta} \widetilde{\mathcal{L}}_T(\boldsymbol{R})\right)^{-1} \xrightarrow{\mathbb{P}} \boldsymbol{G}_0.$$

The proof of the Theorem is given in the subsection B.5 below. Assumption 3 is mandatory for stability of the Hawkes process, and Assumptions 3 and 4 are sufficient to prove that the estimators of the integrated cumulants defined in Equations (11), (12) and (13) are asymptotically consistent. Assumption 2 is a very mild standard technical assumption allowing to prove consistency for estimators based on moments. Assumption 1 is a standard asymptotic moment condition, that allows to identity parameters from the integrated cumulants.

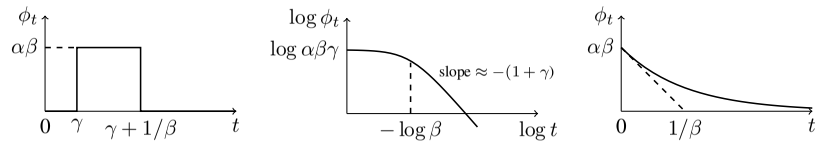
3. Numerical Experiments

In this Section, we provide a comparison of NPHC with the state-of-the art, on simulated datasets with different kernel shapes, the MemeTracker dataset (social networks) and the order book dynamics dataset (finance).

Simulated datasets. We simulated several datasets with Ogata's Thinning algorithm Ogata (1981) using the open-source library `tick`[1], each corresponding to a shape of kernel: rectangular, exponential or power law kernel, see Figure 1 below.

The integral of each kernel on its support equals α, $1/\beta$ can be regarded as a characteristic time-scale and γ is the scaling exponent for the power law distribution and a delay parameter for the rectangular one. We consider a non-symmetric block-matrix $\boldsymbol{G}$ to show that our method can effectively uncover causality between the nodes, see Figure 3. The matrix $\boldsymbol{G}$ has constant entries α on the three blocks - $\alpha = g^{ij} = 1/6$ for dimension 10 and $\alpha = g^{ij} = 1/10$ for dimension 100 -,

1. `https://github.com/X-DataInitiative/tick`



(a) Rectangular kernel $\phi_t = \alpha\beta \mathbb{1}_{[0,1/\beta]}(t-\gamma)$

(b) Power law kernel on log-log scale $\phi_t = \alpha\beta\gamma(1+\beta t)^{-(1+\gamma)}$

(c) Exponential kernel $\phi_t = \alpha\beta \exp(-\beta t)$

Figure 1: The three different kernels used to simulate the datasets.

and zero outside. The two other parameters' values are the same for dimensions 10 and 100. The parameter γ is set to $1/2$ on the three blocks as well, but we set three very different β_0, β_1 and β_2 from one block to the other, with ratio $\beta_{i+1}/\beta_i = 10$ and $\beta_0 = 0.1$. The number of events is roughly equal to 10^5 on average over the nodes. We ran the algorithm on three simulated datasets: a 10-dimensional process with rectangular kernels named Rect10, a 10-dimensional process with power law kernels named PLaw10 and a 100-dimensional process with exponential kernels named Exp100.

MemeTracker dataset. We use events of the most active sites from the MemeTracker dataset[2]. This dataset contains the publication times of articles in many websites/blogs from August 2008 to April 2009, and hyperlinks between posts. We extract the top 100 and the top 200 media sites with the largest number of documents, with about 7 million of events. We name MemeTracker100 the 100-dimensional dataset, and MemeTracker200 the 200-dimensional one. We use the links to trace the flow of information and establish an estimated ground truth for the matrix $\boldsymbol{G}$. Indeed, when an hyperlink j appears in a post in website i, the link j can be regarded as a direct ancestor of the event. Then, Eq. (2) shows g^{ij} can be estimated by $N_T^{i\leftarrow j}/N_T^j = \#\{\text{links } j \to i\}/N_T^j$.

Order book dynamics. We apply our method to financial data, in order to understand the self and cross-influencing dynamics of all event types in an order book. An order book is a list of buy and sell orders for a specific financial instrument, the list being updated in real-time throughout the day. This model has first been introduced in Bacry et al. (2016), and models the order book via the following 8-dimensional point process: $N_t = (P_t^{(a)}, P_t^{(b)}, T_t^{(a)}, T_t^{(b)}, L_t^{(a)}, L_t^{(b)}, C_t^{(a)}, C_t^{(b)})$, where $P^{(a)}$ (resp. $P^{(b)}$) counts the number of upward (resp. downward) price moves, $T^{(a)}$ (resp. $T^{(b)}$) counts the number of market orders at the ask[3] (resp. at the bid) that do not move the price, $L^{(a)}$ (resp. $L^{(b)}$) counts the number of limit orders at the ask[4] (resp. at the bid) that do not move the price, and $C^{(a)}$ (resp. $C^{(b)}$) counts the number of cancel orders at the ask[5] (resp. at the bid) that do not move the price. The financial data has been provided by QuantHouse EUROPE/ASIA, and consists of DAX future contracts between 01/01/2014 and 03/01/2014.

2. https://www.memetracker.org/data.html
3. That is buy orders that are executed and removed from the list.
4. That is buy orders added to the list.
5. That is the number of times a limit order at the ask is canceled: in our dataset, almost 95% of limit orders are canceled before execution.

Baselines. We compare NPHC to state-of-the art baselines: the ODE-based algorithm (ODE) by Zhou et al. (2013a), the Granger Causality-based algorithm (GC) by Xu et al. (2016), the ADM4 algorithm (ADM4) by Zhou et al. (2013b), and the Wiener-Hopf-based algorithm (WH) by Bacry and Muzy (2016).

Metrics. We evaluate the performance of the proposed methods using the computing time, the Relative Error

$$\text{RelErr}(\boldsymbol{A}, \boldsymbol{B}) = \frac{1}{d^2} \sum_{i,j} \frac{|a^{ij} - b^{ij}|}{|a^{ij}|} \mathbb{1}_{\{a^{ij} \neq 0\}} + |b^{ij}| \mathbb{1}_{\{a^{ij}=0\}}$$

and the Mean Kendall Rank Correlation

$$\text{MRankCorr}(\boldsymbol{A}, \boldsymbol{B}) = \frac{1}{d} \sum_{i=1}^{d} \text{RankCorr}([a^{i\bullet}], [b^{i\bullet}]),$$

where $\text{RankCorr}(x, y) = \frac{2}{d(d-1)}(N_{\text{concordant}}(x, y) - N_{\text{discordant}}(x, y))$ with $N_{\text{concordant}}(x, y)$ the number of pairs (i, j) satisfying $x_i > x_j$ and $y_i > y_j$ or $x_i < x_j$ and $y_i < y_j$ and $N_{\text{discordant}}(x, y)$ the number of pairs (i, j) for which the same condition is not satisfied.

Note that RankCorr score is a value between -1 and 1, representing rank matching, but can take smaller values (in absolute value) if the entries of the vectors are not distinct.

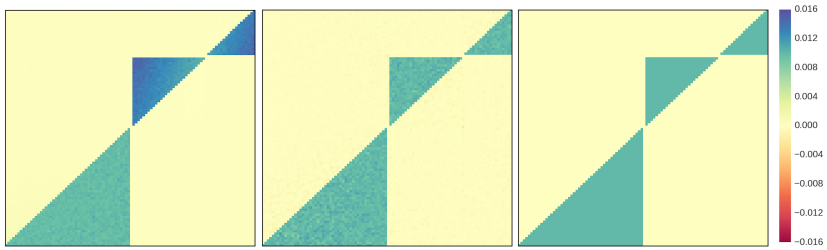


Figure 2: On Exp100 dataset, estimated $\widehat{\boldsymbol{G}}$ with ADM4 (left), with NPHC (middle) and the ground-truth matrix $\boldsymbol{G}$ (right). Both ADM4 and NPHC estimates recover the three blocks. However, ADM4 overestimates the integrals on two of the three blocks, while NPHC gives the same value on each blocks.

Discussion. We perform the ADM4 estimation, with exponential kernel, by giving the exact value $\beta = \beta_0$ of one block. Let us stress that this helps a lot this baseline, in comparison to NPHC where nothing is specified on the shape of the kernel functions. We used $M = 10$ basis functions for both ODE and GC algorithms, and $L = 50$ quadrature points for WH. We did not run WH on the 100-dimensional datasets, for computing time reasons, because its complexity scales with d^4. We ran multi-processed versions of the baseline methods on 56 cores, to decrease the computing time.

Our method consistently performs better than all baselines, on the three synthetic datasets, on MemeTracker and on the financial dataset, both in terms of Kendall rank correlation and estimation

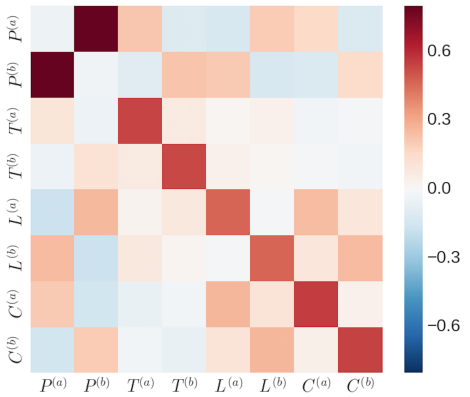


Figure 3: Estimated $\widehat{\boldsymbol{G}}$ via NPHC on DAX order book data.

error. Moreover, we observe that our algorithm is roughly 50 times faster than all the considered baselines.

On Rect10, PLaw10 and Exp100 our method gives very impressive results, despite the fact that it does not uses any prior shape on the kernel functions, while for instance the ADM4 baseline do. On Figure 3, we observe that the matrix $\widehat{\boldsymbol{G}}$ estimated with ADM4 recovers well the block for which $\beta = \beta_0$, i.e. the value we gave to the method, but does not perform well on the two other blocks, while the matrix $\widehat{\boldsymbol{G}}$ estimated with NPHC approximately reaches the true value for each of the three blocks. On these simulated datasets, NPHC obtains a comparable or slightly better Kendall rank correlation, but improves a lot the relative error.

On MemeTracker100, the baseline methods obtain a high relative error between 9% and 19% while our method achieves a relative error of 7% which is a strong improvement. Moreover, NPHC reaches a much better Kendall rank correlation, which proves that it leads to a much better recovery of the relative order of estimated influences than all the baselines. On MemeTracker200, NPHC outperforms again the baselines, with smaller Kendall rank correlation. The comparison of the computation times for both experiments shows that NPHC scales better than other methods. Plus, it has been shown in Zhou et al. (2013a) that kernels of MemeTracker data are not exponential, nor power law. This partly explains why our approach behaves better.

On the financial data, the estimated kernel norm matrix obtained via NPHC, see Figure 3, gave some interpretable results (see also Bacry et al. (2016)):

1. Any 2×2 sub-matrix with same kind of inputs (i.e. Prices changes, Trades, Limits or Cancels) is symmetric. This shows empirically that ask and bid have symmetric roles.

2. The prices are mostly cross-excited, which means that a price increase is very likely to be followed by a price decrease, and conversely. This is consistent with the wavy prices we observe on financial markets.

3. The market, limit and cancel orders are strongly self-excited. This can be explained by the persistence of order flows, and by the splitting of meta-orders into sequences of smaller orders. Moreover, we observe that orders impact the price without changing it. For example, the increase of cancel orders at the bid causes downward price moves.

4. Conclusion

In this paper, we introduce a simple nonparametric method (the NPHC algorithm) that leads to a fast and robust estimation of the matrix $\boldsymbol{G}$ of the kernel integrals of a Multivariate Hawkes process that encodes Granger causality between nodes. This method relies on the matching of the integrated order 2 and order 3 empirical cumulants, which represent the simplest set of global observables containing sufficient information to recover the matrix $\boldsymbol{G}$. Since this matrix fully accounts for the self- and cross- influences of the process nodes (that can represent agents or users in applications), our approach can naturally be used to quantify the degree of endogeneity of a system and to uncover the causality structure of a network.

By performing numerical experiments involving very different kernel shapes, we show that the baselines, involving either parametric or non-parametric approaches are very sensible to model misspecification, do not lead to accurate estimation, and are numerically expensive, while NPHC provides fast, robust and reliable results. This is confirmed on the MemeTracker database, where we show that NPHC outperforms classical approaches based on EM algorithms or the Wiener-Hopf equations. Finally, the NPHC algorithm provided very satisfying results on financial data, that are consistent with well-known stylized facts in finance.

Acknowledgments

This work benefited from the support of the chair "Changing markets", CMAP École Polytechnique and École Polytechnique fund raising - Data Science Initiative. The authors want to thank Marcello Rambaldi for fruitful discussions on order book data's experiments.

Table 2: Metrics on Rect10: comparable rank correlation, strong improvement for relative error and computing time.

Method	ODE	GC	ADM4	WH	NPHC
RelErr	0.007	0.15	0.10	0.005	**0.001**
MRankCorr	0.33	0.02	0.21	**0.34**	**0.34**
Time (s)	846	768	709	933	**20**

Table 3: Metrics on PLaw10: comparable rank correlation, strong improvement for relative error and computing time.

Method	ODE	GC	ADM4	WH	NPHC
RelErr	0.011	0.09	0.053	0.009	**0.0048**
MRankCorr	0.31	0.26	0.24	**0.34**	0.33
Time (s)	870	781	717	946	**18**

Table 4: Metrics on Exp100: comparable rank correlation, strong improvement for relative error and computing time.

Method	ODE	GC	ADM4	NPHC
RelErr	0.092	0.112	0.079	**0.008**
MRankCorr	0.032	0.009	**0.049**	0.041
Time (s)	3215	2950	2411	**47**

Appendix A. Additional experiments

We propose below additional experiments and technical details on the theoretical study of the NPHC procedure.

A.1 Convergence curves versus dimension

The NPHC procedure is divided into two parts: the first is the computation of the integrated cumulants' estimators, the second is the minimization of the loss function. As highlighted in the Section 2.5, the bottleneck of the algorithm is the computation of the cumulants' estimators. However, we can still wonder how fast the optimization algorithm converges with respect to the dimensionality of the point process. To answer that question, we simulated ten datasets corresponding to dimensions $d \in \{10, 20, \dots, 100\}$ with $\mu_i = 0.01$ for $i \in [d]$, $g^{ij} = 0.9/d$ for $(i, j) \in [d]^2$ (such that $\|\boldsymbol{G}\| = 0.9 < 1$), and $T = 10^7$. We then ran the NPHC method and recorded the loss function's evolution over the iterations, rescaled between 0 and 1. One can see on Figure 4 how the dimension influences the convergence curve: using the same hyperparameters for AdaGrad Duchi et al. (2011), the higher the dimension the more oscillating the convergence curve. Plus, the loss function seems to be flatter in lower dimension ($d = 10$ for instance) since AdaGrad needs more iterations to reach a minimum compared to higher-dimensional cases ($d = 70$ or $d = 100$ for instance).

A.2 Relative error versus number of events

The estimation of the integrated cumulants becomes more accurate when the amount of training data increases. The consistency of the estimators given in Equations (7), (8) and (9) is indeed proved in the theorem's proof in Appendix B. A natural question that arises is then to evaluate the precision of the parameter's estimation when the number of points increases, all other things remaining equal. To quantify this effect, we simulated several datasets similar to Rect10 - described in Section 3 - with

Table 5: Metrics on MemeTracker100: strong improvement in relative error, rank correlation and computing time.

Method	ODE	GC	ADM4	NPHC
RelErr	0.162	0.19	0.092	**0.071**
MRankCorr	0.07	0.053	0.081	**0.095**
Time (s)	2944	2780	2217	**38**

Table 6: Metrics on MemeTracker200: strong improvement in relative error, rank correlation and computing time.

Method	ODE	GC	ADM4	NPHC
RelErr	0.173	0.212	0.109	**0.084**
MRankCorr	0.062	0.048	0.077	**0.085**
Time (s)	11786	11210	8903	**164**

different simulation's durations. The only difference between those datasets is then the number of points per node. We ran NPHC ten times per dataset to loosely evaluate the variance of the estimate.

We only focus on the relative error metric which gives more interpretable results. The results summarized on the Figure 5 show the decrease of the relative error as the average (over the ten dimensions) number of points per node becomes larger. This decrease comes along with a variance decrease of the estimates.

A.3 Random choice of d^2 third integrated cumulant's entries

The NPHC method arbitrarily computes the d^2 *iij* entries of the third integrated cumulant, among the d^3 entries available, and then minimizes the distance between theoretical and empirical cumulants. We numerically show in this subsection than the computation of d^2 random entries followed by the minimization of the loss function reachs the same performance than NPHC's method. We sampled three sets of random d^2 indices ijk, and ran the methods on the dataset Rect10 introduced in Section 3.

Table 7: Metrics on Rect10: similar relative errors and rank correlations for the different methods.

Set of indices	Random set 1	Random set 2	Random set 3	*iij* (NPHC)
RelErr	0.0013	0.0014	0.0013	0.0013
MRankCorr	0.34	0.34	0.33	0.34

The results summarized on the Table 7 show that, to our knowledge, the procedure is not very sensitive to the selection of the d^2 entries from the third integrated cumulant tensor.

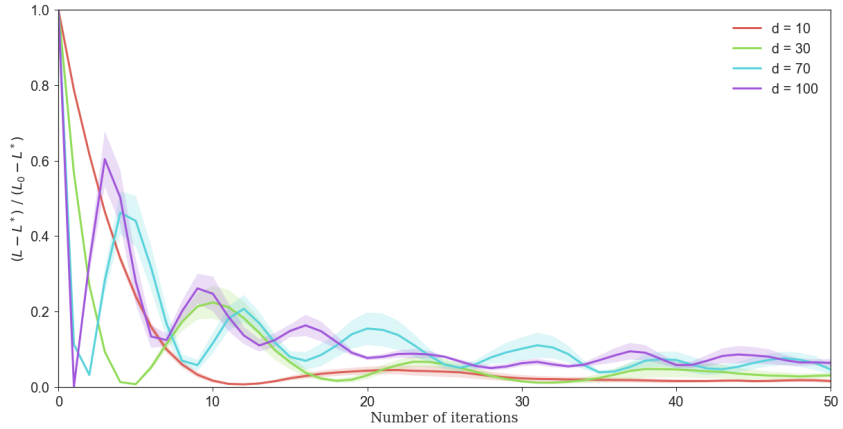


Figure 4: Convergence curves versus dimension.

Appendix B. Technical details

We show in this section how to obtain the equations stated above, the estimators of the integrated cumulants and the scaling coefficient κ that appears in the objective function. We then prove the theorem of the paper.

B.1 Proof of Equation (8)

We denote $\boldsymbol{\nu}(z)$ the matrix

$$\nu^{ij}(z) = \mathcal{L}_z\Big(t \to \frac{\mathbb{E}(dN_u^i dN_{u+t}^j)}{dudt} - \Lambda^i \Lambda^j\Big),$$

where $\mathcal{L}_z(f)$ is the Laplace transform of f, and $\psi_t = \sum_{n\geq 1} \phi_t^{(\star n)}$, where $\phi_t^{(\star n)}$ refers to the n^{th} auto-convolution of ϕ_t. Then we use the characterization of second-order statistics, first formulated in Hawkes (1971) and fully generalized in Bacry and Muzy (2016),

$$\boldsymbol{\nu}(z) = (\boldsymbol{I_d} + \mathcal{L}_{-z}(\boldsymbol{\Psi}))\boldsymbol{L}(\boldsymbol{I_d} + \mathcal{L}_z(\boldsymbol{\Psi}))^\top,$$

where $\boldsymbol{L}^{ij} = \Lambda^i \delta^{ij}$ with δ^{ij} the Kronecker symbol. Since $\boldsymbol{I_d} + \mathcal{L}_z(\boldsymbol{\Psi}) = (\boldsymbol{I_d} - \mathcal{L}_z(\boldsymbol{\Phi}))^{-1}$, taking $z = 0$ in the previous equation gives

$$\begin{aligned}\boldsymbol{\nu}(0) &= (\boldsymbol{I_d} - \boldsymbol{G})^{-1}\boldsymbol{L}(\boldsymbol{I_d} - \boldsymbol{G}^\top)^{-1},\\ \boldsymbol{C} &= \boldsymbol{RLR}^\top,\end{aligned}$$

which gives us the result since the entry (i,j) of the last equation gives $C^{ij} = \sum_m \Lambda^m R_{im} R_{jm}$.

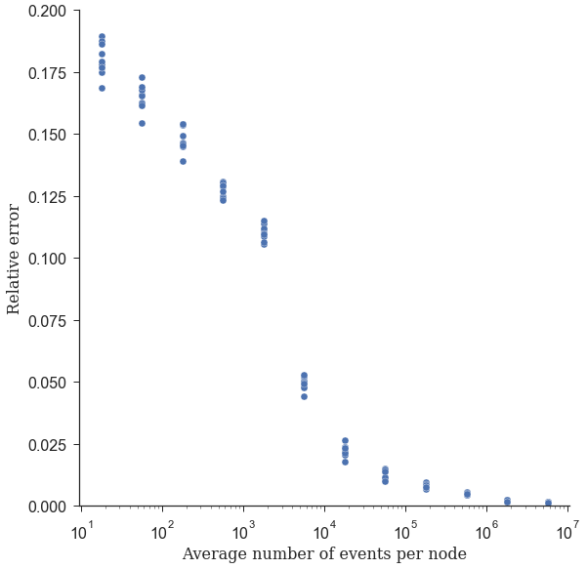


Figure 5: Relative error versus number of events.

B.2 Proof of Equation (9)

We start from Jovanović et al. (2015), cf. Eqs. (48) to (51), and group some terms:

$$\begin{aligned}K^{ijk} = &\sum_m \Lambda^m R_{im} R_{jm} R_{km}\\ &+ \sum_m R_{im} R_{jm} \sum_n \Lambda^n R_{kn} \mathcal{L}_0(\psi^{mn})\\ &+ \sum_m R_{im} R_{km} \sum_n \Lambda^n R_{jn} \mathcal{L}_0(\psi^{mn})\\ &+ \sum_m R_{jm} R_{km} \sum_n \Lambda^n R_{in} \mathcal{L}_0(\psi^{mn}).\end{aligned}$$

Using the relations $\mathcal{L}_0(\psi^{mn}) = R_{mn} - \delta^{mn}$ and $C^{ij} = \sum_m \Lambda^m R_{im} R_{jm}$, proves Equation (9).

B.3 Integrated cumulants estimators

For $H > 0$ let us denote $\Delta_H N_t^i = N_{t+H}^i - N_{t-H}^i$. Let us first remark that, if one restricts the integration domain to $(-H, H)$ in Eqs. (4) and (5), one gets by permuting integrals and expectations:

$$\begin{aligned}
\Lambda^i dt &= \mathbb{E}(dN_t^i) \\
C^{ij} dt &= \mathbb{E}\left(dN_t^i(\Delta_H N_t^j - 2H\Lambda^j)\right) \\
K^{ijk} dt &= \mathbb{E}\left(dN_t^i(\Delta_H N_t^j - 2H\Lambda^j)(\Delta_H N_t^k - 2H\Lambda^k)\right) \\
&\quad - dt\Lambda^i \mathbb{E}\left((\Delta_H N_t^j - 2H\Lambda^j)(\Delta_H N_t^k - 2H\Lambda^k)\right).
\end{aligned}$$

The estimators (11) and (12) are then naturally obtained by replacing the expectations by their empirical counterparts, notably

$$\frac{\mathbb{E}(dN_t^i f(t))}{dt} \to \frac{1}{T} \sum_{\tau \in Z^i} f(\tau).$$

For the estimator (13), we shall also notice that

$$\begin{aligned}
&\mathbb{E}((\Delta_H N_t^j - 2H\Lambda^j)(\Delta_H N_t^k - 2H\Lambda^k)) \\
&\quad = \int\int \mathbb{1}_{[-H,H]}(t) \mathbb{1}_{[-H,H]}(t') C_{t-t'}^{jk} dt dt' \\
&\quad = \int (2H - |t|)^+ C_t^{jk} dt.
\end{aligned}$$

We estimate the last integral with the remark above.

B.4 Choice of the scaling coefficient κ

Following the theory of GMM, we denote $m(X, \theta)$ a function of the data, where X is distributed with respect to a distribution $\mathbb{P}_{\theta_0}$, which satisfies the *moment conditions* $g(\theta) = \mathbb{E}[m(X, \theta)] = 0$ if and only if $\theta = \theta_0$, the parameter θ_0 being the *ground truth*. For $x_1, \dots, x_N$ observed copies of X, we denote $\widehat{g}_i(\theta) = m(x_i, \theta)$, the usual choice of weighting matrix is $\widehat{W}_N(\theta) = \frac{1}{N} \sum_{i=1}^N \widehat{g}_i(\theta) \widehat{g}_i(\theta)^\top$, and the objective to minimize is then

$$\left(\frac{1}{N} \sum_{i=1}^N \widehat{g}_i(\theta)\right) \left(\widehat{W}_N(\theta_1)\right)^{-1} \left(\frac{1}{N} \sum_{i=1}^N \widehat{g}_i(\theta)\right), \tag{15}$$

where θ_1 is a constant vector. Instead of computing the inverse weighting matrix, we rather use its projection on $\{\alpha \boldsymbol{I_d} : \alpha \in \mathbb{R}\}$. It can be shown that the projection choses α as the mean eigenvalue of $\widehat{W}_N(\theta_1)$. We can easily compute the sum of its eigenvalues:

$$\mathrm{Tr}(\widehat{W}_N(\theta_1)) = \frac{1}{N} \sum_{i=1}^N \mathrm{Tr}(\widehat{g}_i(\theta_1) \widehat{g}_i(\theta_1)^\top) = \frac{1}{N} \sum_{i=1}^N \mathrm{Tr}(\widehat{g}_i(\theta_1)^\top \widehat{g}_i(\theta_1)) = \frac{1}{N} \sum_{i=1}^N ||\widehat{g}_i(\theta_1)||_2^2.$$

In our case, $\widehat{g}(\boldsymbol{R}) = \left[\mathbf{vec}[\widehat{\boldsymbol{K^c}} - \boldsymbol{K^c}(\boldsymbol{R})], \mathbf{vec}[\widehat{\boldsymbol{C}} - \boldsymbol{C}(\boldsymbol{R})]\right]^\top \in \mathbb{R}^{2d^2}$. Considering a block-wise weighting matrix, one block for $\widehat{\boldsymbol{K^c}} - \boldsymbol{K^c}(\boldsymbol{R})$ and the other for $\widehat{\boldsymbol{C}} - \boldsymbol{C}(\boldsymbol{R})$, the sum of the eigenvalues of the first block becomes $\|\widehat{\boldsymbol{K^c}} - \boldsymbol{K^c}(\boldsymbol{R})\|_2^2$, and $\|\widehat{\boldsymbol{C}} - \boldsymbol{C}(\boldsymbol{R})\|_2^2$ for the second. We compute the previous terms with $\boldsymbol{R}_1 = 0$. All together, the objective function to minimize is

$$\frac{1}{\|\widehat{\boldsymbol{K^c}}\|_2^2} \|\boldsymbol{K^c}(\boldsymbol{R}) - \widehat{\boldsymbol{K^c}}\|_2^2 + \frac{1}{\|\widehat{\boldsymbol{C}}\|_2^2} \|\boldsymbol{C}(\boldsymbol{R}) - \widehat{\boldsymbol{C}}\|_2^2. \tag{16}$$

Dividing this function by $\left(1/\|\widehat{\boldsymbol{K^c}}\|_2^2 + 1/\|\widehat{\boldsymbol{C}}\|_2^2\right)^{-1}$, and setting $\kappa = \|\widehat{\boldsymbol{K^c}}\|_2^2 / (\|\widehat{\boldsymbol{K^c}}\|_2^2 + \|\widehat{\boldsymbol{C}}\|_2^2)$, we obtaind the loss function given in Equation (10).

B.5 Proof of the Theorem

The main difference with the usual Generalized Method of Moments, see Hansen (1982), relies in the relaxation of the moment conditions, since we have $\mathbb{E}[\widehat{g}_T(\theta_0)] = m_T \neq 0$. We adapt the proof of consistency given in Newey and McFadden (1994).

We can relate the integral of the Hawkes process's kernels to the integrals of the cumulant densities, from Jovanović et al. (2015). Our cumulant matching method would fall into the usual GMM framework if we could estimate - without bias - the integral of the covariance on $\mathbb{R}$, and the integral of the skewness on $\mathbb{R}^2$. Unfortunately, we can't do that easily. We can however estimate without bias $\int f_t^T C_t^{ij} dt$ and $\int f_t^T K_t^{ijk} dt$ with f^T a compact supported function on $[-H_T, H_T]$ that weakly converges to 1, with $H_T \longrightarrow \infty$. In most cases we will take $f_t^T = \mathbb{1}_{[-H_T, H_T]}(t)$. Denoting $\widehat{C}^{ij,(T)}$ the estimator of $\int f_t^T C_t^{ij} dt$, the term $|\mathbb{E}[\widehat{C}^{ij,(T)}] - C^{ij}| = |\int f_t^T C_t^{ij} dt - C^{ij}|$ can be considered a proxy to the *distance to the classical GMM*. This distance has to go to zero to make the rest of GMM's proof work: the estimator $\widehat{C}^{ij,(T)}$ is then asymptotically unbiased towards C^{ij} when T goes to infinity.

B.5.1 NOTATIONS

We observe the multivariate point process $(\boldsymbol{N}_t)$ on $\mathbb{R}^+$, with Z^i the events of the i^{th} component. We will often write covariance / skewness instead of integrated covariance / skewness. In the rest of the document, we use the following notations.

Hawkes kernels' integrals $\boldsymbol{G}^{\text{true}} = \int \boldsymbol{\Phi}_t dt = (\int \phi_t^{ij} dt)_{ij} = \boldsymbol{I_d} - (\boldsymbol{R}^{\text{true}})^{-1}$

Theoretical mean matrix $\boldsymbol{L} = \mathrm{diag}(\Lambda^1, \dots, \Lambda^d)$

Theoretical covariance $\boldsymbol{C} = \boldsymbol{R}^{\text{true}} \boldsymbol{L} (\boldsymbol{R}^{\text{true}})^\top$

Theoretical skewness $\boldsymbol{K^c} = (K^{iij})_{ij} = (\boldsymbol{R}^{\text{true}})^{\odot 2} \boldsymbol{C}^\top + 2[\boldsymbol{R}^{\text{true}} \odot (\boldsymbol{C} - \boldsymbol{R}^{\text{true}} \boldsymbol{L})](\boldsymbol{R}^{\text{true}})^\top$

Filtering function $f^T \geq 0 \qquad \mathrm{supp}(f^T) \subset [-H_T, H_T] \qquad F^T = \int f_s^T ds \qquad \widetilde{f}_t^T = f_{-t}^T$

Events sets $Z^{i,T,1} = Z^i \cap [H_T, T + H_T] \qquad Z^{j,T,2} = Z^j \cap [0, T + 2H_T]$

Estimators of the mean $\widehat{\Lambda}^i = \frac{N_{T+H_T}^i - N_{H_T}^i}{T} \qquad \widetilde{\Lambda}^j = \frac{N_{T+2H_T}^j}{T+2H_T}$

Estimator of the covariance $\widehat{C}^{ij,(T)} = \frac{1}{T}\sum_{\tau\in Z^{i,T,1}}\left(\sum_{\tau'\in Z^{j,T,2}} f_{\tau'-\tau} - \widetilde{\Lambda}^j F^T\right)$

Estimator of the skewness[6]

$$\widehat{K}^{ijk,(T)} = \frac{1}{T}\sum_{\tau\in Z^{i,T,1}}\left(\sum_{\tau'\in Z^{j,T,2}} f_{\tau'-\tau} - \widetilde{\Lambda}^j F^T\right)\left(\sum_{\tau''\in Z^{k,T,2}} f_{\tau'-\tau} - \widetilde{\Lambda}^k F^T\right) - \frac{\widehat{\Lambda}^i}{T+2H_T}\sum_{\tau'\in Z^{j,T,2}}\left(\sum_{\tau''\in Z^{k,T,2}} (f^T\star\widetilde{f}^T)_{\tau'-\tau''} - \widetilde{\Lambda}^k (F^T)^2\right)$$

GMM RELATED NOTATIONS

$$\begin{aligned}
\theta &= \boldsymbol{R} \quad \text{and} \quad \theta_0 = \boldsymbol{R}^{\text{true}}\\
g_0(\theta) &= \textbf{vec}\begin{bmatrix}\boldsymbol{C} - \boldsymbol{R}\boldsymbol{L}\boldsymbol{R}^\top\\ \boldsymbol{K^c} - \boldsymbol{R}^{\odot^2}\boldsymbol{C}^\top - 2[\boldsymbol{R}\odot(\boldsymbol{C}-\boldsymbol{R}\boldsymbol{L})]\boldsymbol{R}^\top\end{bmatrix} \in \mathbb{R}^{2d^2}\\
\widehat{g}_T(\theta) &= \textbf{vec}\begin{bmatrix}\widehat{\boldsymbol{C}}^{(T)} - \boldsymbol{R}\widehat{\boldsymbol{L}}\boldsymbol{R}^\top\\ \widehat{\boldsymbol{K}^{\boldsymbol{c}}}^{(T)} - \boldsymbol{R}^{\odot^2}(\widehat{\boldsymbol{C}}^{(T)})^\top - 2[\boldsymbol{R}\odot(\widehat{\boldsymbol{C}}^{(T)} - \boldsymbol{R}\widehat{\boldsymbol{L}})]\boldsymbol{R}^\top\end{bmatrix} \in \mathbb{R}^{2d^2}\\
Q_0(\theta) &= g_0(\theta)^\top W g_0(\theta)\\
\widehat{Q}_T(\theta) &= \widehat{g}_T(\theta)^\top \widehat{W}_T \widehat{g}_T(\theta)
\end{aligned}$$

B.5.2 CONSISTENCY

First, let's remind a useful theorem for consistency in GMM from Newey and McFadden (1994).

Theorem 4 *If there is a function $Q_0(\theta)$ such that (i) $Q_0(\theta)$ is uniquely maximized at θ_0; (ii) Θ is compact; (iii) $Q_0(\theta)$ is continuous; (iv) $\widehat{Q}_T(\theta)$ converges uniformly in probability to $Q_0(\theta)$, then $\widehat{\theta}_T = \arg\max \widehat{Q}_T(\theta) \xrightarrow{\mathbb{P}} \theta_0$.*

We can now prove the consistency of our estimator.

Theorem 5 *Suppose that $(\boldsymbol{N_t})$ is observed on $\mathbb{R}^+$, $\widehat{W}_T \xrightarrow{\mathbb{P}} W$, and*

1. *W is positive semi-definite and $Wg_0(\theta) = 0$ if and only if $\theta = \theta_0$,*
2. *$\theta \in \Theta$, which is compact,*
3. *the spectral radius of the kernel norm matrix satisfies $||\boldsymbol{\Phi}||_* < 1$,*
4. *$\forall i,j,k \in [d]$, $\int f_u^T C_u^{ij} du \to \int C_u^{ij} du$ and $\int f_u^T f_v^T K_{u,v}^{ijk} du dv \to \int K_{u,v}^{ijk} du dv$,*
5. *$(F^T)^2/T \xrightarrow{\mathbb{P}} 0$ and $||f||_\infty = O(1)$.*

6. When $f_t^T = \mathbb{1}_{[-H_T,H_T]}(t)$, we remind that $(f^T\star\widetilde{f}^T)_t = (2H_T - |t|)^+$. This leads to the estimator we showed in the article.

Then

$$\widehat{\theta}_T \xrightarrow{\mathbb{P}} \theta_0.$$

Remark 6 *In practice, we use a constant sequence of weighting matrices: $\widehat{W}_T = \boldsymbol{I_d}$.*

Proof Proceed by verifying the hypotheses of Theorem 2.1 from Newey and McFadden (1994). Condition 2.1(i) follows by (i) and by $Q_0(\theta) = [W^{1/2}g_0(\theta)]^\top[W^{1/2}g_0(\theta)] > 0 = Q_0(\theta_0)$. Indeed, there exists a neighborhood N of θ_0 such that $\theta \in N\backslash\{\theta_0\}$ and $g_0(\theta) \neq 0$ since $g_0(\theta)$ is a polynom. Condition 2.1(ii) follows by (ii). Condition 2.1(iii) is satisfied since $Q_0(\theta)$ is a polynom. Condition 2.1(iv) is harder to prove. First, since $\widehat{g}_T(\theta)$ is a polynom of θ, we prove easily that $\mathbb{E}[\sup_{\theta\in\Theta}|\widehat{g}_T(\theta)|] < \infty$. Then, by Θ compact, $g_0(\theta)$ is bounded on Θ, and by the triangle and Cauchy-Schwarz inequalities,

$$\begin{aligned}
&\left|\widehat{Q}_T(\theta) - Q_0(\theta)\right|\\
&\quad\le \left|(\widehat{g}_T(\theta) - g_0(\theta))^\top \widehat{W}_T(\widehat{g}_T(\theta) - g_0(\theta))\right|\\
&\quad+ \left|g_0(\theta)^\top(\widehat{W}_T + \widehat{W}_T^\top)(\widehat{g}_T(\theta) - g_0(\theta))\right| + \left|g_0(\theta)^\top(\widehat{W}_T - W)g_0(\theta)\right|\\
&\quad\le \|\widehat{g}_T(\theta) - g_0(\theta)\|^2\|\widehat{W}_T\| + 2\|g_0(\theta)\|\|\widehat{g}_T(\theta) - g_0(\theta)\|\|\widehat{W}_T\| + \|g_0(\theta)\|^2\|\widehat{W}_T - W\|.
\end{aligned}$$

To prove $\sup_{\theta\in\Theta}\left|\widehat{Q}_T(\theta) - Q_0(\theta)\right| \xrightarrow{\mathbb{P}} 0$, we should now prove that $\sup_{\theta\in\Theta}\|\widehat{g}_T(\theta) - g_0(\theta)\| \xrightarrow{\mathbb{P}} 0$. By Θ compact, it is sufficient to prove that $\|\widehat{\boldsymbol{L}} - \boldsymbol{L}\| \xrightarrow{\mathbb{P}} 0$, $\|\widehat{\boldsymbol{C}}^{(T)} - \boldsymbol{C}\| \xrightarrow{\mathbb{P}} 0$, and $\|\widehat{\boldsymbol{K^c}}^{(T)} - \boldsymbol{K^c}\| \xrightarrow{\mathbb{P}} 0$.

PROOF THAT $\|\widehat{\boldsymbol{L}} - \boldsymbol{L}\| \xrightarrow{\mathbb{P}} 0$

The estimator of $\boldsymbol{L}$ is unbiased so let's focus on the variance of $\widehat{\boldsymbol{L}}$.

$$\begin{aligned}
\mathbb{E}[(\widehat{\Lambda}^i - \Lambda^i)^2] &= \mathbb{E}\left[\left(\frac{1}{T}\int_{H_T}^{T+H_T}(dN_t^i - \Lambda^i dt)\right)^2\right]\\
&= \frac{1}{T^2}\int_{H_T}^{T+H_T}\int_{H_T}^{T+H_T}\mathbb{E}[(dN_t^i - \Lambda^i dt)(dN_{t'}^i - \Lambda^i dt')]\\
&= \frac{1}{T^2}\int_{H_T}^{T+H_T}\int_{H_T}^{T+H_T} C_{t'-t}^{ii} dt dt'\\
&\le \frac{1}{T^2}\int_{H_T}^{T+H_T} C^{ii} dt = \frac{C^{ii}}{T} \longrightarrow 0
\end{aligned}$$

By Markov inequality, we have just proved that $\|\widehat{\boldsymbol{L}} - \boldsymbol{L}\| \xrightarrow{\mathbb{P}} 0$.

PROOF THAT $\|\widehat{\boldsymbol{C}}^{(T)} - \boldsymbol{C}\| \xrightarrow{\mathbb{P}} 0$

First, let's remind that $\mathbb{E}(\widehat{\boldsymbol{C}}^{(T)}) \neq \boldsymbol{C}$. Indeed,

$$\begin{aligned}
\mathbb{E}\left(\widehat{C}^{ij,(T)}\right) &= \mathbb{E}\left(\frac{1}{T}\int_{H_T}^{T+H_T} dN_t^i \int_0^{T+2H_T} dN_{t'}^j f_{t'-t} - \widehat{\Lambda}^i\widetilde{\Lambda}^j F^T\right) \\
&= \mathbb{E}\left(\frac{1}{T}\int_{H_T}^{T+H_T} dN_t^i \int_{-t}^{T+2H_T-t} dN_{t+s}^j f_s - \Lambda^i\Lambda^j F^T\right) + \epsilon^{ij,T,H_T}F^T \\
&= \frac{1}{T}\int_{H_T}^{T+H_T}\int_{-H_T}^{H_T} f_s \mathbb{E}\left(dN_t^i dN_{t+s}^j - \Lambda^i\Lambda^j ds\right) + \epsilon^{ij,T,H_T}F^T \\
&= \int f_s C_s^{ij} ds + \epsilon^{ij,T,H_T}F^T
\end{aligned}$$

Now,

$$\begin{aligned}
\epsilon^{ij,T,H_T} &= \mathbb{E}\left(\Lambda^i\Lambda^j - \widehat{\Lambda}^i\widetilde{\Lambda}^j\right) \\
&= -\frac{1}{T^2}\int_{H_T}^{T+H_T}\int_0^{T+2H_T} \mathbb{E}\left(dN_t^i dN_{t'}^j - \Lambda^i\Lambda^j dt dt'\right) \\
&= -\frac{1}{T^2}\int_{H_T}^{T+H_T}\int_0^{T+2H_T} C_{t-t'}^{ij} dt dt' \\
&= -\frac{1}{T}\int\left(1 + \left(\frac{H_T - |t|}{T}\right)^-\right)^+ C_t^{ij} dt
\end{aligned}$$

Since f satisfies $F^T = o(T)$, we have $\mathbb{E}(\widehat{\boldsymbol{C}}^{(T)}) \longrightarrow \boldsymbol{C}$. It remains now to prove that $\|\widehat{\boldsymbol{C}}^{(T)} - \mathbb{E}(\widehat{\boldsymbol{C}}^{(T)})\| \xrightarrow{\mathbb{P}} 0$.
Let's now focus on the variance of $\widehat{C}^{ij,(T)}$: $\quad \mathbb{V}(\widehat{C}^{ij,(T)}) = \mathbb{E}\left((\widehat{C}^{ij,(T)})^2\right) - \mathbb{E}(\widehat{C}^{ij,(T)})^2$.
Now,

$$\begin{aligned}
&\mathbb{E}\left((\widehat{C}^{ij,(T)})^2\right) \\
&= \mathbb{E}\left(\frac{1}{T^2}\sum_{(\tau,\eta,\tau',\eta')\in(Z^{i,T,1})^2\times(Z^{j,T,2})^2}(f_{\tau'-\tau} - F^T/(T+2H_T))(f_{\eta'-\eta} - F^T/(T+2H_T))\right) \\
&= \mathbb{E}\left(\frac{1}{T^2}\int_{t,s\in[H_T,T+H_T]}\int_{t',s'} dN_t^i dN_{t'}^j dN_s^i dN_{s'}^j (f_{t'-t} - F^T/(T+2H_T))(f_{s'-s} - F^T/(T+2H_T))\right) \\
&= \frac{1}{T^2}\int_{t,s\in[H_T,T+H_T]}\int_{t',s'\in[0,T+2H_T]} \mathbb{E}\left(dN_t^i dN_{t'}^j dN_s^i dN_{s'}^j\right) \\
&\qquad\qquad \cdot (f_{t'-t} - F^T/(T+2H_T))(f_{s'-s} - F^T/(T+2H_T))
\end{aligned}$$

And,

$$\begin{aligned}
&\mathbb{E}(\widehat{C}^{ij,(T)})^2 \\
&= \frac{1}{T^2}\int_{t,s\in[H_T,T+H_T]}\int_{t',s'\in[0,T+2H_T]} \mathbb{E}\left(dN_t^i dN_{t'}^j\right)\mathbb{E}\left(dN_s^i dN_{s'}^j\right) \\
&\qquad\qquad \cdot (f_{t'-t} - F^T/(T+2H_T))(f_{s'-s} - F^T/(T+2H_T))
\end{aligned}$$

Then, the variance involves the integration towards the difference of moments $\mu^{r,s,t,u} - \mu^{r,s}\mu^{t,u}$.
Let's write it as a sum of cumulants, since cumulants density are integrable.

$$\begin{aligned}
\mu^{r,s,t,u} - \mu^{r,s}\mu^{t,u} &= \kappa^{r,s,t,u} + \kappa^{r,s,t}\kappa^u[4] + \kappa^{r,s}\kappa^{t,u}[3] + \kappa^{r,s}\kappa^t\kappa^u[6] + \kappa^r\kappa^s\kappa^t\kappa^u - (\kappa^{r,s} + \kappa^r\kappa^s)(\kappa^{t,u} + \kappa^t\kappa^u) \\
&= \kappa^{r,s,t,u} \\
&\quad + \kappa^{r,s,t}\kappa^u + \kappa^{u,r,s}\kappa^t + \kappa^{t,u,r}\kappa^s + \kappa^{s,t,u}\kappa^r \\
&\quad + \kappa^{r,t}\kappa^{s,u} + \kappa^{r,u}\kappa^{s,t} \\
&\quad + \kappa^{r,t}\kappa^s\kappa^u + \kappa^{r,u}\kappa^s\kappa^t + \kappa^{s,t}\kappa^r\kappa^u + \kappa^{s,t}\kappa^r\kappa^u
\end{aligned}$$

In the rest of the proof, we denote $a_t = \mathbb{1}_{t\in[H_T,T+H_T]}$, $b_t = \mathbb{1}_{t\in[0,T+2H_T]}$, $c_t = \mathbb{1}_{t\in[-H_T,H_T]}$, $g_t = f_t - \frac{1}{T+2H_T}F^T$
Before starting the integration of each term, let's remark that:

1. $\boldsymbol{\Psi}_t = \sum_{n\geq 1}\boldsymbol{\Phi}_t^{(\star n)} \geq 0$ since $\boldsymbol{\Phi}_t \geq 0$.
2. The regular parts of C_u^{ij}, $K_{u,v}^{ijk}$ (skewness density) and $M_{u,v,w}^{ijkl}$ (fourth cumulant density) are positive as polynoms of integrals of $\psi_\cdot^{ab}$ with positive coefficients. The integrals of the singular parts are positive as well.
3. (a) $\int a_t b_{t'} f_{t'-t} dt dt' = TF^T$
   (b) $\int a_t b_{t'} g_{t'-t} dt dt' = 0$
   (c) $\int a_t b_{t'} |g_{t'-t}| dt dt' \leq 2TF^T$
4. $\forall t \in \mathbb{R}, a_t(b \star \widetilde{g})_t = 0$, where $\widetilde{g}_s = g_{-s}$.

Fourth cumulant We want here to compute $\int \kappa_{t,t',s,s'}^{i,j,i,j} a_t b_{t'} a_s b_{s'} g_{t'-t} g_{s'-s} dt dt' ds ds'$.
We remark that $|g_{t'-t}g_{s'-s}| \leq (||f||_\infty(1 + 2H_T/T))^2 \leq 4||f||_\infty^2$.

$$\begin{aligned}
\left|\frac{1}{T^2}\int \kappa_{t,t',s,s'}^{i,j,i,j} a_t b_{t'} a_s b_{s'} g_{t'-t} g_{s'-s} dt dt' ds ds'\right| &\leq \left(\frac{2||f||_\infty}{T}\right)^2 \int dt a_t \int dt' b_{t'} \int ds a_s \int ds' b_{s'} M_{t'-t,s-t,s'-t}^{ijij} \\
&\leq \left(\frac{2||f||_\infty}{T}\right)^2 \int dt a_t \int dt' b_{t'} \int ds a_s \int dw M_{t'-t,s-t,w}^{ijij} \\
&\leq \left(\frac{2||f||_\infty}{T}\right)^2 \int dt a_t \int M_{u,v,w}^{ijij} du dv dw \\
&\leq \frac{4||f||_\infty^2}{T} M^{ijij} \underset{T\to\infty}{\longrightarrow} 0
\end{aligned}$$

Third $\times$ First We have four terms, but only two different forms since the roles of (s, s') and (t, t') are symmetric.
First form

$$\begin{aligned}\int \kappa^{i,j,i}_{t,t',s}\Lambda^j G_{\boldsymbol{t}} d\boldsymbol{t} &= \frac{\Lambda^j}{T^2}\int \kappa^{i,j,i}_{t,t',s} a_t b_{t'} a_s b_{s'} g_{t'-t} g_{s'-s} dt dt' ds ds' \\ &= \frac{\Lambda^j}{T^2}\int \kappa^{i,j,i}_{t,t',s} a_t b_{t'} a_s (b \star \widetilde{g})_s g_{t'-t} dt dt' ds \\ &= 0 \qquad \text{since } a_s(b\star\widetilde{g})_s = 0\end{aligned}$$

Second form

$$\begin{aligned}\Big|\int \kappa^{i,j,j}_{t,t',s'}\Lambda^i G_{\boldsymbol{t}} d\boldsymbol{t}\Big| &= \Big|\frac{\Lambda^i}{T^2}\int \kappa^{i,j,j}_{t,t',s'} a_t b_{t'} a_s b_{s'} g_{t'-t} g_{s'-s} dt dt' ds ds'\Big| \\ &= \Big|\frac{\Lambda^i}{T^2}\int \kappa^{i,j,j}_{t,t',s'} a_t b_{t'} g_{t'-t} b_{s'} (a\star g)_{s'} dt dt' ds'\Big| \\ &\leq \frac{\Lambda^i}{T^2} 2||f||_\infty \int ds' b_{s'}(a\star|g|)_{s'} \int dt a_t \int dt' b_{t'} K^{ijj}_{t'-s',t-s'} \\ &\leq 4||f||_\infty K^{ijj}\Lambda^i \frac{F^T}{T} \underset{T\to\infty}{\longrightarrow} 0\end{aligned}$$

Second $\times$ Second
First form

$$\begin{aligned}\Big|\int \kappa^{i,i}_{t,s}\kappa^{j,j}_{t',s'} G_{\boldsymbol{t}} d\boldsymbol{t}\Big| &\leq \frac{2||f||_\infty}{T^2}\int C^{ii}_{t-s} C^{jj}_{t'-s'} a_t b_{t'} |g_{t'-t}| a_s b_{s'} dt dt' ds ds' \\ &\leq \frac{2||f||_\infty}{T^2} C^{ii} C^{jj} \int a_t b_{t'} |g_{t'-t}| dt dt' \\ &\leq 4||f||_\infty C^{ii} C^{jj} \frac{F^T}{T} \underset{T\to\infty}{\longrightarrow} 0\end{aligned}$$

Second form

$$\Big|\int \kappa^{i,j}_{t,s'}\kappa^{i,j}_{t',s} G_{\boldsymbol{t}} d\boldsymbol{t}\Big| \leq 4||f||_\infty (C^{ij})^2 \frac{F^T}{T} \underset{T\to\infty}{\longrightarrow} 0$$

Second $\times$ First $\times$ First
First form

$$\int \kappa^{i,j}_{t,t'}\Lambda^i\Lambda^j G_{\boldsymbol{t}} d\boldsymbol{t} = \frac{\Lambda^i\Lambda^j}{T^2}\int \kappa^{i,j}_{t,t'} a_t b_{t'} g_{t'-t} dt dt' \int a_s b_{s'} g_{s'-s} ds ds' = 0$$

Second form

$$\int \kappa^{i,i}_{t,s}\Lambda^j\Lambda^j G_{\boldsymbol{t}} d\boldsymbol{t} = \left(\frac{\Lambda^j}{T}\right)^2 \int \kappa^{i,i}_{t,s} a_t b_{t'} g_{t'-t} a_s (b\star\widetilde{g})_s dt dt' ds = 0$$

We have just proved that $\mathbb{V}(\widehat{\boldsymbol{C}}^{(T)}) \xrightarrow{\mathbb{P}} 0$. By Markov inequality, it ensures us that $\|\widehat{\boldsymbol{C}}^{(T)} - \mathbb{E}(\widehat{\boldsymbol{C}}^{(T)})\| \xrightarrow{\mathbb{P}} 0$, and finally that $\|\widehat{\boldsymbol{C}}^{(T)} - \boldsymbol{C}\| \xrightarrow{\mathbb{P}} 0$. ■

PROOF THAT $\|\widehat{\boldsymbol{K}^{\boldsymbol{c}}}^{(T)} - \boldsymbol{K}^{\boldsymbol{c}}\| \xrightarrow{\mathbb{P}} 0$

The scheme of the proof is similar to the previous one. The upper bounds of the integrals involve the same kind of terms, plus the new term $(F^T)^2/T$ that goes to zero thanks to the assumption 5 of the theorem.

References

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

Y. Aït-Sahalia, J. Cacho-Diaz, and R. JA Laeven. Modeling financial contagion using mutually exciting jump processes. Technical report, National Bureau of Economic Research, 2010.

A. Arnold, Y. Liu, and N. Abe. Temporal causal modeling with graphical granger methods. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 66–75. ACM, 2007.

E. Bacry and J.-F. Muzy. First- and second-order statistics characterization of hawkes processes and non-parametric estimation. *IEEE Transactions on Information Theory*, 62(4):2184–2202, 2016.

E. Bacry, I. Mastromatteo, and J.-F. Muzy. Hawkes processes in finance. *Market Microstructure and Liquidity*, 1(01):1550005, 2015.

E. Bacry, T. Jaisson, and J.-F. Muzy. Estimation of slowly decreasing hawkes kernels: application to high-frequency order book dynamics. *Quantitative Finance*, pages 1–23, 2016.

S. Basu, A. Shojaie, and G. Michailidis. Network granger causality with inherent grouping structure. *The Journal of Machine Learning Research*, 16(1):417–453, 2015.

A. Choromanska, M. Henaff, M. Mathieu, G. Ben Arous, and Y. LeCun. The loss surfaces of multilayer networks. In *AISTATS*, 2015.

R. Crane and D. Sornette. Robust dynamic classes revealed by measuring the response function of a social system. *Proceedings of the National Academy of Sciences*, 105(41), 2008.

J. Da Fonseca and R. Zaatour. Hawkes process: Fast calibration, application to trade clustering, and diffusive limit. *Journal of Futures Markets*, 34(6):548–579, 2014.

D. J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes Volume I: Elementary Theory and Methods*. Springer Science & Business Media, 2003.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

M. Eichler. Graphical modelling of multivariate time series. *Probability Theory and Related Fields*, 153(1-2):233–268, 2012.

M. Eichler, R. Dahlhaus, and J. Dueck. Graphical modeling for multivariate hawkes processes with nonparametric link functions. *Journal of Time Series Analysis*, 2016. ISSN 1467-9892.

M. Farajtabar, Y. Wang, M. Rodriguez, S. Li, H. Zha, and L. Song. Coevolve: A joint point process model for information diffusion and network co-evolution. In *Advances in Neural Information Processing Systems*, pages 1945–1953, 2015.

M. Gomez-Rodriguez, J. Leskovec, and B. Schölkopf. Modeling information propagation with survival theory. *Proceedings of the International Conference on Machine Learning*, 2013.

C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, 1969. ISSN 00129682, 14680262.

A. R. Hall. *Generalized Method of Moments*. Oxford university press, 2005.

L. P. Hansen. Large sample properties of generalized method of moments estimators. *Econometrica: Journal of the Econometric Society*, pages 1029–1054, 1982.

N. R. Hansen, P. Reynaud-Bouret, and V. Rivoirard. Lasso and probabilistic inequalities for multivariate point processes. *Bernoulli*, 21(1):83–143, 2015.

S. J. Hardiman and J.-P. Bouchaud. Branching-ratio approximation for the self-exciting Hawkes process. *Phys. Rev. E*, 90(6):062807, December 2014.

A. G. Hawkes. Point spectra of some mutually exciting point processes. *Journal of the Royal Statistical Society. Series B (Methodological)*, 33(3):438–443, 1971. ISSN 00359246.

A. G. Hawkes and D. Oakes. A cluster process representation of a self-exciting process. *Journal of Applied Probability*, pages 493–503, 1974.

G. W. Imbens and D. B. Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

S. Jovanović, J. Hertz, and S. Rotter. Cumulants of Hawkes point processes. *Phys. Rev. E*, 91(4): 042802, April 2015.

R. Lemonnier and N. Vayatis. Nonparametric markovian learning of triggering kernels for mutually exciting and mutually inhibiting multivariate hawkes processes. In *Machine Learning and Knowledge Discovery in Databases*, pages 161–176. Springer, 2014.

E. Lewis and G. Mohler. A nonparametric em algorithm for multiscale hawkes processes. *Journal of Nonparametric Statistics*, 2011.

G. O. Mohler, M. B. Short, P. J. Brantingham, F. P. Schoenberg, and G. E. Tita. Self-exciting point process modeling of crime. *Journal of the American Statistical Association*, 2011.

W. K Newey and D. McFadden. Large sample estimation and hypothesis testing. *Handbook of econometrics*, 4:2111–2245, 1994.

Y. Ogata. On lewis' simulation method for point processes. *Information Theory, IEEE Transactions on*, 27(1):23–31, 1981.

Y. Ogata. Space-time point-process models for earthquake occurrences. *Annals of the Institute of Statistical Mathematics*, 50(2):379–402, 1998.

J. Pearl. *Causality*. Cambridge university press, 2009.

A. Podosinnikova, F. Bach, and S. Lacoste-Julien. Rethinking lda: moment matching for discrete ica. In *Advances in Neural Information Processing Systems*, pages 514–522, 2015.

P. Reynaud-Bouret and S. Schbath. Adaptive estimation for hawkes processes; application to genome analysis. *The Annals of Statistics*, 38(5):2781–2822, 2010.

V.S. Subrahmanian, A. Azaria, S. Durst, V. Kagan, A. Galstyan, K. Lerman, L. Zhu, E. Ferrara, A. Flammini, and F. Menczer. The darpa twitter bot challenge. *Computer*, 49(6):38–46, 2016.

H. Xu, M. Farajtabar, and H. Zha. Learning granger causality for hawkes processes. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1717–1726, 2016.

S.-H. Yang and H. Zha. Mixture of mutually exciting processes for viral diffusion. In *Proceedings of the International Conference on Machine Learning*, 2013.

K. Zhou, H. Zha, and L. Song. Learning triggering kernels for multi-dimensional hawkes processes. In *Proceedings of the International Conference on Machine Learning*, pages 1301–1309, 2013a.

K. Zhou, H. Zha, and L. Song. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. *AISTATS*, 2013b.

Journal of Machine Learning Research 18 (2018) 1–46 Submitted 04/17; Revised 04/18; Published 05/18

Making Better Use of the Crowd: How Crowdsourcing Can Advance Machine Learning Research

Jennifer Wortman Vaughan jenn@microsoft.com
Microsoft Research
641 Avenue of the Americas, 7th Floor
New York, NY 10011

Editor: Qiang Liu

Abstract

This survey provides a comprehensive overview of the landscape of crowdsourcing research, targeted at the machine learning community. We begin with an overview of the ways in which crowdsourcing can be used to advance machine learning research, focusing on four application areas: 1) data generation, 2) evaluation and debugging of models, 3) hybrid intelligence systems that leverage the complementary strengths of humans and machines to expand the capabilities of AI, and 4) crowdsourced behavioral experiments that improve our understanding of how humans interact with machine learning systems and technology more broadly. We next review the extensive literature on the behavior of crowdworkers themselves. This research, which explores the prevalence of dishonesty among crowdworkers, how workers respond to both monetary incentives and intrinsic forms of motivation, and how crowdworkers interact with each other, has immediate implications that we distill into best practices that researchers should follow when using crowdsourcing in their own research. We conclude with a discussion of additional tips and best practices that are crucial to the success of any project that uses crowdsourcing, but rarely mentioned in the literature.

Keywords: crowdsourcing, data generation, model evaluation, hybrid intelligence, behavioral experiments, incentives, mechanical turk

1. Introduction

Crowdsourcing allows us to harness the power of human computation to solve tasks that are notoriously difficult to solve with computers alone, such as determining whether or not an image contains a tree, rating the quality of a website, or verifying the phone number of a business. The machine learning community was early to embrace crowdsourcing as a tool for quickly and inexpensively obtaining the vast quantities of labeled data needed to train machine learning systems. Crowdsourcing has been used to generate the image annotations that are needed to train computer vision systems (Deng et al., 2009; Patterson and Hays, 2012; Raykar et al., 2010; Wah et al., 2011), provide the linguistic annotations needed for common natural language processing tasks (Callison-Burch and Dredze, 2010; Snow et al., 2008), and collect the relevance judgments needed to optimize search engines (Alonso, 2013; Alonso et al., 2008). This simple idea—that crowds could be used to generate training data for machine learning algorithms—inspired a flurry of algorithmic work on how to best elicit and aggregate potentially noisy labels (e.g., Ghosh et al., 2011; Karger et al., 2011; Khetan and Oh, 2016; Liu et al., 2012a; Sheng et al., 2008; Welinder et al., 2010; Zhang et al., 2016b; Zhou et al., 2012), still an active area of research (Zheng et al., 2017). Meanwhile, machine learning researchers have begun to put crowdsourcing to use in other ways, most commonly as a tool to evaluate and debug machine learning models (Chang et al., 2009; Ribeiro et al., 2016).

Crowdsourcing has flourished as a research tool outside of the machine learning community as well. In human-computer interaction and related fields, researchers are building "hybrid intelligence systems" with the goal of expanding the capabilities of current AI technology by incorporating humans in the loop (e.g., Bernstein et al., 2010; Kamar, 2016; Lasecki et al., 2017; Zhang et al., 2012). And psychologists and social scientists have increasingly moved experiments that traditionally would have been run in physical labs onto crowdsourcing platforms (Buhrmester et al., 2011; Mason and Suri, 2012). While these bodies of research are less well known within the machine learning community, there are countless opportunities for machine learning research to both influence and benefit from these lines of work. For example, human-in-the-loop clustering algorithms have been designed that produce better clusters by drawing on the common sense knowledge and experience of the crowd (Gomes et al., 2011; Heikinheimo and Ukkonen, 2013; Tamuz et al., 2011), while behavioral experiments run on the crowd offer insight about how to encourage human trust in algorithmic predictions (Dietvorst et al., 2015, 2016; Dzindolet et al., 2002).

The first goal of this survey is to expand the horizons of how machine learning researchers think about crowdsourcing, providing a broad overview of ways in which crowdsourcing can benefit (and sometimes benefit from) machine learning research. Unlike other surveys, which go into greater depth on algorithms for aggregating crowdsourced labels (Zhang et al., 2016a; Zheng et al., 2017), we address the label aggregation problem only briefly, devoting relatively more attention and detail to applications that are less well known within the machine learning community, in the hope of inspiring new connections and directions of research. We break the applications into four broad categories:

- **Data generation.** (Section 2) Crowdsourcing platforms are well suited to generating data, but challenges arise since the data supplied by crowdworkers can be prone to errors. We start with a brief review of two lines of research aimed at improving the quality of crowdsourced labels. The first assumes that data points are redundantly assigned to multiple workers and seeks algorithms for aggregating workers' responses that take into account the quality of individual workers (e.g., Zhang et al., 2016a; Zheng et al., 2017). Though there are many notable exceptions, much of this work builds on the influential model and expectation-maximization framework of Dawid and Skene (1979). The second line of work focuses on developing incentive schemes to motivate high quality responses. Much of this work builds on the literature on peer prediction, a framework in which crowdworkers' payments are a function of their own reported labels and the labels of other workers (Jurca and Faltings, 2009; Miller et al., 2005; Radanovic et al., 2016). We then review ways in which crowdsourcing has been applied to generate other forms of data, including transcriptions of printed text (von Ahn et al., 2008), translations of sentences from one language to another (Zaidan and Callison-Burch, 2011), and image annotations (Kovashka et al., 2016).

- **Evaluating and debugging models.** (Section 3) Crowdsourcing is also commonly used to evaluate or debug models, including unsupervised learning models, which can

be difficult to evaluate objectively since there is often no clear notion of ground truth. Along these lines, we discuss the use of crowdsourcing to evaluate the coherence of topic models, generative models used to discover and explore the thematic topics discussed in a set of documents (Chang et al., 2009). We also explore ways in which crowdsourcing has been used to evaluate the interpretability of explanations of predictions in supervised learning settings (Ribeiro et al., 2016) and to debug the components of a pipeline in a complex computer vision system (Mottaghi et al., 2013, 2016; Parikh and Zitnick, 2011).

- **Hybrid intelligence systems.** (Section 4) Hybrid intelligence or "human-in-the-loop" systems advance the capabilities of current AI technology by leveraging the complementary strengths of humans and machines. We explore several compelling examples of hybrid systems that suggest their great potential: hybrid systems for clustering data points (Gomes et al., 2011; Heikinheimo and Ukkonen, 2013; Tamuz et al., 2011), transcribing speech in real time (Kushalnagar et al., 2012; Lasecki and Bigham, 2012; Lasecki et al., 2012, 2013, 2017; Naim et al., 2013), scheduling conference sessions (André et al., 2013; Bhardwaj et al., 2014; Chilton et al., 2014; Kim et al., 2013), and forecasting geopolitical or economic events (Atanasov et al., 2017; Baron et al., 2014; Mellers et al., 2015b). These systems are able to achieve more than would be possible with state-of-the-art machine learning or AI systems alone because they can make use of people's common sense knowledge, life experience, subjective beliefs, and flexible reasoning skills.

- **Behavioral studies to inform machine learning research.** (Section 5) As machine learning becomes a larger part of people's everyday lives, interest in understanding how real people interact with machine learning systems continues to grow. There has been a surge of recent research on questions like how to design machine learning models that are more interpretable (Doshi-Velez and Kim, 2017; Lipton, 2016) or how to understand the ways that algorithmic decisions impact people's lives (Angwin et al., 2016; Barocas and Selbst, 2016). These questions are interdisciplinary in nature and require gaining a better understanding of the underlying principles behind humans' interactions with machine learning and other technological systems. At the same time, psychologists and social scientists have started using crowdsourcing platforms as a fast and easy way to gain access to large pools of subjects for behavioral experiments. This presents a natural opportunity for researchers to conduct studies on crowdsourcing platforms that improve our understanding of how humans interact with technology broadly and machine learning algorithms in particular. Rather than evaluating the way in which people interact with one particular algorithm, this line of research aims to develop an understanding of components of human behavior that could inform the use and design of machine learning systems more broadly. We walk through illustrative examples of behavioral studies aimed at understanding user trust in algorithmic predictions (Dietvorst et al., 2015, 2016; Dzindolet et al., 2002; Poursabzi-Sangdeh et al., 2018) and how users react to online advertising (Goldstein et al., 2013, 2014). The latter study, while not directly motivated by a machine learning question, could provide insights that inform the design of better machine learning algorithms for ad pricing and display.

Having explored these applications and motivated the use of crowdsourcing in machine learning research, the remainder of the survey addresses our second goal: to provide the reader with best practices for crowdsourcing by drawing deeply on the vast and cross-disciplinary literature aimed at studying and understanding the behavior of the crowd itself. In Section 6, we describe several studies that quantify and measure the prevalence of dishonest or spammy behavior on crowdsourcing platforms (Chandler and Paolacci, 2017; Suri et al., 2011; Wessling et al., 2017). We discuss how to set payments for tasks in light of both ethical considerations (Williamson, 2016) and a body of research that explores how the quality and quantity of crowdwork are impacted by monetary incentives (e.g., Buhrmester et al., 2011; Ho et al., 2015; Mason and Watts, 2009; Shaw et al., 2011). We examine how workers react to intrinsic sources of motivation, such as gamification (von Ahn and Dabbish, 2008) or the satisfaction of performing meaningful work (Chandler and Kapelner, 2013; Rogstadius et al., 2011), and how intrinsically motivating workers can increase workers' willingness to complete tasks. Finally, we explore communication and collaboration patterns of workers and see that crowdworkers are not independent and isolated, but rather part of a rich communication network (Gray et al., 2016; Yin et al., 2016). We discuss the implications of this work on the practice of using crowdsourcing for research, outlining best practices that should be followed whether one wishes to use crowdsourcing for data generation, model evaluation, building hybrid systems, running behavioral studies, or beyond.

Section 7 concludes with a discussion of additional tips and tricks that are crucial for the success of any crowdsourcing-based project, but rarely discussed in the literature.

1.1 A Note on Scope

As described above, this survey was written with two overarching goals in mind. The first is to inspire machine learning researchers to discover unexpected applications of crowdsourcing in their own research. Because of this, relatively more space is devoted to applications that may be less familiar to the machine learning community, such as hybrid intelligence systems and behavioral studies of users interacting with technology, compared with more familiar applications like data generation and model evaluation. Full surveys have been written just on the problem of aggregating noisy labels from a crowd; readers who are most interested in this problem are encouraged to look at Zheng et al. (2017), Zhang et al. (2016a), or the comprehensive summary of related work in Zhang et al. (2016b).

The second goal is to introduce machine learning researchers to the extensive cross-disciplinary literature on crowd behavior and motivate why understanding this literature is crucial for successfully employing crowdsourcing in research. Because of this, Section 6 draws heavily on work from outside the machine learning community, and emphasizes experimental work over pure theory.

Several other crowdsourcing surveys have been published, each focused on a different set of themes and problems. Readers interested in gaining a broader perspective on crowdsourcing may be interested in recent surveys and guides from the computer vision community (Kovashka et al., 2016), the databases community (Li et al., 2016), and the marketing community (Goodman and Paolacci, 2017), or the older but still widely cited and applicable how-to guides on crowdsourcing user studies (Kittur et al., 2008) and behavioral research (Mason and Suri, 2012).

1.2 Background on Mechanical Turk and Other Crowdsourcing Platforms

In this survey, we use the term crowdsourcing very generally to encompass both paid and volunteer crowdwork, done by experts or nonexperts, on any general or specialized crowdsourcing platform. Regardless, the majority of research covered was conducted using paid crowdworkers, and the majority of that on one particular platform, Amazon Mechanical Turk.

Amazon Mechanical Turk[1] is the most commonly used crowdsourcing platform among researchers. It is designed for crowdsourcing relatively small "microtasks" (often referred to as "HITs," for human intelligence tasks) such as labeling a set of images or completing a survey, though it can also be used for more complex short-term tasks, such as participating in behavioral experiments (Mason and Suri, 2012) or even writing fiction (Kim et al., 2017). Task requesters come to Mechanical Turk to post their tasks, stating up front the amount of money that they are willing to pay to have their tasks completed. Requesters can also specify certain criteria that a crowdworker must meet to be eligible for their task, such as having an approval rate of more than a particular amount (say, 97%) on previous tasks or being located in a particular country. Crowdworkers can then browse the set of tasks available and choose the tasks they would like to work on. After a crowdworker completes a task, the requester approves her work and a payment is made.

Although Mechanical Turk is used broadly in the research community, it is not the right choice for everyone. In particular, it can be difficult to use from outside of the United States. Luckily, many alternatives are available. For example:

- CrowdFlower[2] is a crowdsourcing platform widely used in both industry and research. CrowdFlower offers specialized enterprise solutions for businesses with artificial intelligence and data science needs including search relevance evaluation, sentiment analysis, and data classification.
- ClickWorker[3] is a German crowdsourcing platform that attracts European workers. It provides support for specialized tasks such as translation, web research, and web content generation. It also provides tools for mobile crowdsourcing.
- Prolific Academic[4] is a UK-based crowdsourcing platform focused primarily on connecting researchers with participants for behavioral or user studies.
- Upwork[5] is an online freelancer marketplace focused not on microtasks but rather on larger scale jobs such as writing an article or designing a website.
- TopCoder[6] hosts crowdsourcing contests (Chawla et al., 2015; DiPalantino and Vojnovic, 2009; Gao et al., 2012) in which coders design and develop software in response to specific challenges and compete for prizes. Unlike on the other sites mentioned, only those who create the top-judged submissions are paid.

1. https://www.mturk.com
2. https://www.crowdflower.com
3. https://www.clickworker.com
4. https://www.prolific.ac
5. https://www.upwork.com
6. https://www.topcoder.com

Several researchers have compared commercially available platforms along different dimensions. Vakharia and Lease (2015) provided a thorough qualitative content analysis of seven platforms: ClickWorker, CrowdComputing Systems (now WorkFusion), CloudFactory, CrowdFlower, CrowdSource, MobileWorks (now LeadGenius), and oDesk (now Upwork), examining factors like infrastructure and tools, support for fraud protection, and quality of work. Peera et al. (2017) provided a detailed experimental comparison of three platforms: Mechanical Turk, CrowdFlower, and Prolific Academic. They compared dropout rates, response rates, workers' performance on attention-check questions, workers' reliability, workers' familiarity with common psychology studies (which would signal an overused population of subjects), and the ability to replicate classic psychology studies on each platform. They found, for example, that workers on both CrowdFlower and Prolific Academic were less familiar with common psychology studies and less dishonest than workers on Mechanical Turk, with Prolific Academic producing higher quality data than CrowdFlower.

2. Data Generation

Perhaps the most common application of crowdsourcing within the machine learning community is data generation. We first describe techniques for crowdsourcing binary or categorical labels, reviewing the literature on how to improve label quality through redundancy and incentives. We then discuss several examples of ways in which crowdsourcing has been used to generate more complex forms of data, such as image annotations and translations of text. As mentioned above, entire surveys could be written on the topic of crowdsourced data generation alone, and indeed some have (Zhang et al., 2016a; Zheng et al., 2017). Our treatment of this topic is comparatively brief, intended only to give the reader the flavor of this work with pointers to the literature for readers who wish to learn more.

2.1 Generating Binary or Categorical Labels

We start with the setting in which crowdworkers are presented with unlabeled data instances (for instance, websites) and are asked to supply labels (for instance, a binary label indicating whether or not the website contains profanity). The main challenge arises from the fact that the supplied labels are often noisy or inaccurate, either because workers are imperfect or because workers are unmotivated to put high effort into the labeling task. There are two primary lines of work aimed at improving the quality of crowdsourced labels. The first assumes that each instance is presented to multiple crowdworkers and explores algorithmic techniques for aggregating the workers' responses (e.g., Abraham et al., 2016; Aydin et al., 2014; Demartini et al., 2012; Fan et al., 2015; Gao et al., 2016; Ghosh et al., 2011; Ho et al., 2013; Karger et al., 2011, 2014; Khetan and Oh, 2016; Kim and Ghahramani, 2012; Li et al., 2014a,b; Liu et al., 2012a,b; Ma et al., 2015; Raykar et al., 2010; Shah and Lee, 2018; Sheng et al., 2008; Tian and Zhu, 2015; Venanzi et al., 2014; Welinder et al., 2010; Whitehill et al., 2009; Zhang et al., 2016b; Zhou et al., 2012). The second explores the use of well-designed incentives to encourage higher quality work (e.g., Dasgupta and Ghosh, 2013; Ho et al., 2016; Kamar and Horvitz, 2012; Kamble et al., 2015; Radanovic et al., 2016; Shah et al., 2015; Shah and Zhou, 2016a,b).

Much of the research on label aggregation builds on the model proposed in the seminal paper of Dawid and Skene (1979). The basic Dawid-Skene model assumes that instances are

homogeneous. A worker's probability of labeling any given instance correctly is controlled by one or more worker-specific quality parameters. In the original model, each worker's quality is governed by a latent confusion matrix that specifies the worker's probability of choosing each possible label conditioned on the true label of the instance. A variety of extensions have been studied. For example, several papers have explored models in which difficulty varies by instance (Khetan and Oh, 2016; Ma et al., 2015; Whitehill et al., 2009; Zhou et al., 2012) or workers have diverse sets of skills (Fan et al., 2015; Ho et al., 2013; Welinder et al., 2010). Most of this work assumes a fixed assignment of instances to workers, but some assumes that workers arrive online and are assigned to instances upon arrival (Abraham et al., 2016; Ho and Vaughan, 2012; Ho et al., 2013; Karger et al., 2011, 2014). Some work requires the existence of "gold-standard" or "control" tasks (Le et al., 2010; Liu et al., 2013; Oleson et al., 2011), data instances for which ground truth labels are known a priori, while some does not. Extensions of the Dawid-Skene model have also been applied to the problem of aggregating ordinal labels or rankings (Zhou et al., 2014).

While there are exceptions, much of this research builds on the general expectation-maximization (EM) algorithmic framework that Dawid and Skene (1979) proposed. This framework is outlined in Algorithm 1. The basic approach involves iteratively updating estimates of both workers' quality parameters and the labels of each instance. At each step, quality parameters are updated treating the current label estimates as ground truth. The instance labels are then updated to their most likely values treating the quality parameters as ground truth. The details of these updates vary. Zheng et al. (2017) provide a thorough survey and empirical comparison of seventeen algorithms that are based on this general framework, characterizing them in terms of the way in which instances and workers are modeled as well as the specifics of how the calculations of quality parameters and label assignments are made (through what they call direct computation, using optimization methods, or using probabilistic graphical models). While their experiments reveal that different algorithms perform best on different data sets, they show that the original Dawid-Skene algorithm is itself fairly robust in practice.

Algorithm 1 The basic EM framework of Dawid and Skene (1979).

Input: Sets of worker-generated labels for each instance
Initialize each instance's label based on a simple majority vote
repeat
  for all Workers w **do**
    Calculate w's quality parameter(s), treating each instance's current label as ground truth
  end for
  for all Instances i **do**
    Calculate the most likely label for i, treating each worker's approximated quality parameter(s) as ground truth
  end for
until Label assignments have converged
Output: The current label assignments for each instance

One broadly studied class of incentive schemes for crowdsourcing is built on the literature on peer prediction (e.g., Dasgupta and Ghosh, 2013; Jurca and Faltings, 2009; Kamar and Horvitz, 2012; Kamble et al., 2015; Miller et al., 2005; Prelec, 2004; Prelec et al., 2017; Radanovic and Faltings, 2013; Waggoner and Chen, 2014; Witkowski and Parkes, 2012), a framework in which workers are rewarded based on a function of their own reported labels and the reports of others. Under many peer prediction mechanisms, higher rewards are given to reports that are "surprisingly common" among workers, where "surprise" could be measured, for example, in terms of a common prior or the frequency of a label. As just one example, to determine the payment for a worker w who labels a particular instance as x, the mechanism proposed by Radanovic et al. (2016) selects another worker w' at random and compares the label provided by w to the label provided by w'. If w' reports x (so the two workers' labels agree), then w receives payment $1/f$ where f is the empirical frequency with which other workers have reported the label x over all instances. If w' does not report x, then w receives no payment. Radanovic et al. (2016) showed that under some assumptions on workers' beliefs, truthful reporting (i.e., each worker reporting the label they think is most likely) is an equilibrium under this mechanism.

The primary benefit of peer prediction style methods is that there is no need to know ground truth labels in order to calculate payments. One major drawback is that most such methods leave open the opportunity for workers to benefit by coordinating and colluding on incorrect reports, behavior that has been observed when these methods were tested on real workers (Gao et al., 2014). There is an active line of research on developing peer prediction techniques for which such undesirable equilibria do not exist, or at least are less profitable than truth telling (Agarwal et al., 2017; Dasgupta and Ghosh, 2013; Shnayder et al., 2016).

When gold-standard labels are available for some instances, workers can be rewarded directly for the quality of labels they supply for these instances, using either simple bonuses for accuracy (Ho et al., 2015; Shaw et al., 2011) or more complex incentive schemes (Ho et al., 2016; Shah et al., 2015; Shah and Zhou, 2016a,b). See Section 6.3 for a more detailed discussion of how workers respond to monetary incentives in practice.

2.2 Generating Transcriptions, Translations, and Image Annotations

Crowdsourcing is also used to generate more complex and free-form labels, such as transcriptions, translations of language, or image annotations.

Perhaps the best known example of a crowdsourcing system for transcription is reCAPTCHA (von Ahn et al., 2008). CAPTCHAs (or Completely Automated Public Turing tests to tell Computers and Humans Apart) are security tools designed to prevent bots from accessing online services (von Ahn et al., 2003, 2004). People attempting to access a website or create an account are asked to perform a task that is difficult for computers to perform but that humans find easy, such as reading and transcribing distorted characters. CAPTCHAs are used to stop ticket scalpers from using bots to buy out popular shows and to prevent spammers from opening arbitrary numbers of email accounts.

Von Ahn et al. (2008) found a way to put the vast quantities of human effort exerted on solving CAPTCHAs to use, harnessing this effort to digitize old books that current optical character recognition (OCR) systems were unable to handle. Their reCAPTCHA system presents two images of words, both taken from scanned text on which state-of-the-

art OCR systems have failed. One of these images is a gold-standard data point for which the correct transcription is already known. This is the image used to test whether or not the transcriber is human. The true label of the second image is unknown. By completing the CAPTCHA, the human is essentially entering a label for this data. Since reCAPTCHA was acquired by Google, similar techniques have been used to annotate images and build other large-scale machine learning data sets.[7]

Within the natural language processing community, crowdsourcing has been successfully used to generate translations of sentences from one language to another (Ambati et al., 2012; Callison-Burch and Dredze, 2010; Pavlick et al., 2014; Post et al., 2012; Zaidan and Callison-Burch, 2011; Zbib et al., 2012). This approach is especially effective for language pairs for which not much data exists. As one example, Zaidan and Callison-Burch (2011) used crowdsourcing to generate translations of sentences from Urdu to English. They assigned crowdworkers different jobs such as translating a sentence, editing other workers' translations to make them more fluent and grammatical, or ranking the quality of a translation. Finally, they used machine learning methods to predict the highest quality translation based on sentence-level features, worker-level features, and worker-generated ranks. The crowd-generated translations collected by this and other systems can then be used as training data for machine translation tasks.

Within the computer vision community, crowdsourcing is commonly used to collect human-generated labels and annotations for images and video (e.g., Deng et al., 2009; Gebru et al., 2017; Patterson and Hays, 2012; Patterson et al., 2014; Raykar et al., 2010; Russakovsky et al., 2015b; Sorokin and Forsyth, 2008; Su et al., 2012; Vijayanarasimhan and Grauman, 2009; Welinder et al., 2010). For example, the widely used ImageNet database[8] was constructed by leveraging workers on Mechanical Turk to perform tasks such as verifying image annotations and generating bounding boxes (Deng et al., 2009; Russakovsky et al., 2015a; Su et al., 2012). Other data generation and labeling tasks appropriate for crowdsourcing include object classification, attribute (or feature) generation, and image segmentation. A full taxonomy of applications of crowdsourcing to computer visions tasks is beyond the scope of this paper; see instead the comprehensive survey of Kovashka et al. (2016).

3. Evaluating and Debugging Models

Aside from data generation, the most common use of crowdsourcing within the machine learning community is to evaluate or debug models. Crowdsourced evaluation is especially common for unsupervised models, which generally cannot be evaluated in terms of simple metrics like accuracy or precision because there is no objective notion of ground truth. More recently, crowdsourcing has been used to evaluate human-centric properties of supervised models, such as model interpretability. In this section, we review several examples of applications of crowdsourcing to model evaluation and debugging. This list of examples is not intended to be exhaustive, but to give a flavor of different ways in which crowdsourcing can be used in this context.

7. `https://www.google.com/recaptcha`
8. `http://www.image-net.org`

3.1 Evaluating Unsupervised Models

It is increasingly common to see crowdsourcing used to evaluate unsupervised models, such as topic models (e.g., Chang et al., 2009; Hu et al., 2014; Newman et al., 2011; Paul and Dredze, 2014). Topic models are widely used to discover thematic topics from a set of documents such as New York Times articles from the past year or transcripts of Supreme Court hearings (Blei and Lafferty, 2009; Blei et al., 2003; Boyd-Graber et al., 2017). In this context, a topic is a distribution over words in a vocabulary. Every word in the vocabulary occurs in every topic, but with a different probability or weight. For example, a topic model might produce a food topic that places high weight on `cheese`, `kale`, and `bread`, or a politics topic that places high weight on `election`, `senate`, and `bill`. Each document is then represented as a distribution over topics.

Topic models are often used for data exploration and summarization, especially in the social sciences (Boyd-Graber et al., 2017). In order to be useful in these contexts, the inferred topics must be meaningful to end users. For example, if the set of words that appear with high weight in an individual topic are not coherent, the topic will not be useful to end users trying to understand the content of their documents. However, "meaningfulness" is hard to measure analytically, leading many researchers to instead evaluate topic models in terms of easier to quantify criteria, such as predictive power. To address this problem, Chang et al. (2009) proposed using crowdsourcing to measure the quality of a set of topics. The researchers designed a word intrusion task in which a crowdworker is presented with a randomly ordered list of the most common words from a topic. Included in that list is one intruder word that has low weight for the topic but high weight for another topic. The worker is then asked to identify the intruder. If the topic is coherent, then picking out the intruder should be easy (think {`cheese`, `bread`, `steak`, `election`, `mushroom`, `kale`}). If a topic is incoherent, identifying the intruder would be harder. The average error that crowdworkers make on this task can thus be used as a proxy for how coherent topics are. The researchers found that previous measures of success like high log likelihood of held out data do not necessarily imply coherence, illustrating the value of crowd-based evaluation over other techniques.

3.2 Evaluating Model Interpretability

In supervised learning, models are often evaluated in terms of objective performance metrics such as accuracy, precision, or recall. However, even if a model performs well in terms of these criteria, users may hesitate to rely on the model if they do not understand the model's predictions, especially in critical domains like health or criminal justice. Because of this, there is now wide interest in developing models that are human-interpretable (e.g., Doshi-Velez and Kim, 2017; Jung et al., 2017; Koh and Liang, 2017; Lipton, 2016; Lou et al., 2012, 2013; Paul, 2016; Ribeiro et al., 2016; Ustun and Rudin, 2016). Because of the subjective and inherently human-centric nature of interpretability, it is natural to use crowdsourcing to evaluate the interpretability of models.

As one example, Ribeiro et al. (2016) proposed an algorithm, Local Interpretable Model-agnostic Explanations, or LIME, that generates simple, locally faithful explanations for individual predictions made by potentially complex black-box models. They used crowdsourcing to test how how these explanations impact people's ability to perform tasks such as as-

sessing the classifier's quality or determining instances on which the classifier will make a mistake. In one study, the researchers presented crowdworkers with predictions from two SVM classifiers trained on different data sets along with explanations for their predictions. Workers were asked to select the classifier they believed would have better performance. The researchers found that explanations improved workers' ability to choose the best classifier and that LIME produced more effective explanations than a greedy technique. In another study, the researchers used crowdworkers to test which explanations best helped people find flaws in a trained model by identifying features used in the explanation that are irrelevant for the prediction task.

In this example, the researchers compared the effectiveness of specific techniques for generating explanations of predictions. In Section 5.1, we contrast this with crowdsourcing approaches that can be used to gain a better general understanding of how various properties associated with the "interpretability" of a model impact different aspects of human behavior in different scenarios.

3.3 Debugging Components of a Pipeline

In fields like computer vision, speech recognition, translation, and natural language processing, systems often consist of several discrete components linked together to perform a complex task. For example, consider the problem of semantic segmentation, which involves partitioning an image into semantically meaningful parts and labeling each part with a class. There are promising approaches to this problem that use machine learning models such as conditional random fields (CRFs) to integrate feedback from independent components that perform various scene understanding tasks like object detection, scene recognition, and segmentation. If a system designer wants to improve performance, it is not always clear which component to focus attention on.

To solve this problem, Parikh and Zitnick (2011) proposed the idea of "human debugging," in which humans are used to uncover bottlenecks in AI systems. The goal of human debugging is to identify which component in a system is the "weakest link." The basic idea is simple. To quantify how much potential improvements to a particular component would benefit the system as a whole, we could imagine replacing the component with something (close to) perfectly accurate and testing how much the system improves. Since for many vision and language tasks human performance is an upper bound on what we might expect from a machine, we can instead replace the component with a human.

Mottaghi et al. (2013, 2016) applied this idea in order to analyze the performance of a CRF that has been used in the computer vision community for scene understanding (Yao et al., 2012). They replaced each component with crowdworkers from Mechanical Turk and measured the change in performance of both the component in isolation and the system as a whole. One of their most interesting findings was that humans are actually less accurate than machines at one particular task (classifying super-pixels), yet when human classifications were plugged into the CRF, the system performance improved. One interpretation of this result is that perhaps making fewer mistakes classifying super-pixels is not enough. Rather it may be more important that the classifier makes the right kind of mistakes—the kind made by humans. This kind of feedback helps designers know where to focus their effort.

Recently, Nushi et al. (2017) took this idea one step further, allowing crowdworkers to propose targeted fixes to the machine components of a larger system and then evaluating the effect of various component fixes on the overall system performance.

4. Hybrid Intelligence Systems

Despite the current hype around AI and the great technological advances that have been made in recent years, AI systems are still far from perfect. In some cases, AI systems can benefit by involving humans in the loop to perform tasks that rely on life experience, judgment, or domain knowledge (Kamar, 2016). Such hybrid intelligence systems can leverage the complementary strengths of humans and machines to accomplish more than would be possible using humans or machines alone.

Hybrid systems have been designed to perform tasks from grading students' work (Kulkarni et al., 2014; Wright et al., 2015) to writing essays or novels (Bernstein et al., 2010; Kim et al., 2017; Kittur et al., 2011; Salehi et al., 2017; Teevan et al., 2016) to building better topic models (Hu et al., 2014). In this section, we walk through four illustrative examples: hybrid systems for clustering data points (Gomes et al., 2011; Heikinheimo and Ukkonen, 2013; Tamuz et al., 2011), transcribing speech in real time (Kushalnagar et al., 2012; Lasecki and Bigham, 2012; Lasecki et al., 2012, 2013, 2017; Naim et al., 2013), scheduling conference sessions (André et al., 2013; Bhardwaj et al., 2014; Chilton et al., 2014; Kim et al., 2013), and forecasting geopolitical or economic events (Atanasov et al., 2017; Baron et al., 2014; Mellers et al., 2015b).

4.1 Hybrid Clustering

Hybrid intelligence systems can be put to use to solve traditional machine learning problems like clustering in scenarios in which data points are easier for humans to understand and categorize than they are for machines. For example, given a data set of celebrity images, a human could potentially use their life experience and background knowledge to categorize them into sets like "actors" or "politicians," while a machine without access to this knowledge and experience could not.

Many hybrid clustering techniques have been proposed (Davidson et al., 2014; Gomes et al., 2011; Karaletsos et al., 2016; Mazumdar and Saha, 2016; Tamuz et al., 2011; Vesdapunt et al., 2014; Vinayak and Hassibi, 2016; Vinayak et al., 2014; Wah et al., 2014; Wang et al., 2012b; Wauthier et al., 2012; Yi et al., 2012a,b), the majority of which solicit human judgments or comparisons in order to actively generate a similarity matrix or other similarity function. Approaches vary in terms of the types of queries given to human judges, the algorithms used to aggregate their responses, and whether or not additional features of each object are available to the algorithm. Some researchers focus primarily on the entity resolution setting, in which the goal is to cluster together objects that refer to the same entity (e.g., Marcus et al., 2011; Mazumdar and Saha, 2016; Vesdapunt et al., 2014; Wang et al., 2012b), while others consider clustering more broadly.

Tamuz et al. (2011) designed an adaptive algorithm that estimates a similarity matrix from human judgments based on comparisons of triples ("*Is object A more similar to object B or object C?*"). Their approach requires only a relatively small number of human judgments to obtain a good approximation. Using this approach, they were able to answer questions

like which necktie would be a good substitute for another, a task that would perhaps be difficult for a machine without specialized human knowledge. As a complement to this algorithmic work, Wilber et al. (2014) outlined and studied several user interface techniques that allow high quality comparisons of triples to be collected faster.

Gomes et al. (2011) proposed a crowd-clustering approach in which each member of a crowd is presented a relatively small set of objects and asked to cluster just these objects. The sets of objects presented to different workers are distinct but overlap. The partial clusterings generated by the workers are then aggregated into one full clustering of all objects using an algorithm based on Bayesian inference.

Heikinheimo and Ukkonen (2013) took a different approach to hybrid clustering, proposing a crowd-powered version of the k-means algorithm. Running the standard k-means algorithm (Lloyd, 1982) requires the ability to perform two main operations:

1. Given a set S of objects and one new object x, find the object in S that is closest to x.
2. Find the center of a set of objects S, that is, the object in S with the lowest sum of distances to other objects in S.

Heikinheimo and Ukkonen (2013) showed how to perform each of these operations with a crowd. The first operation is straight-forward, assuming the size of the set S (which, in this case, is the number of clusters k since S is the set of cluster representatives) is sufficiently small: they simply present a worker with S and x and ask the worker which object in S is closest to x. (Noisy responses from multiple workers can be combined using standard techniques.) To perform the second operation, the researchers proposed the following simple algorithm, which they call `crowd-median`. Each crowdworker is presented with a set of three objects from the set S and asked which object is the outlier. After many such triples have been presented to workers, the object chosen least often as the outlier is selected as the center. The researchers gave both theoretical and empirical evidence that the output of `crowd-median` coincides with existing definitions of a centroid, and empirical evidence that the resulting crowd-powered k-means algorithms produces coherent clusters.

4.2 Hybrid Speech Recognition

Quickly and reliably converting speech to text requires a level of contextual understanding beyond the capabilities of machines and a level of speed beyond the capabilities of most humans. Closed captioning systems that rely on automatic speech recognition work very well under ideal circumstances (for example, when the voice recording is high quality and the system has been trained on data from the particular speaker), but can fail when presented with low quality audio, speakers with novel accents, or language with technical jargon that falls outside the vocabulary on which the system was trained. In these scenarios, professional stenographers produce the best results, but high quality stenographers can be prohibitively expensive and are not available on demand.

With this in mind, Lasecki et al. developed Scribe (Kushalnagar et al., 2012; Lasecki and Bigham, 2012; Lasecki et al., 2012, 2013, 2017; Naim et al., 2013), a hybrid speech recognition system that provides relatively inexpensive, real-time, and on-demand closed captioning to deaf or hard of hearing users to help them understand lectures, meetings, or other day-to-day conversations. As illustrated in Figure 1, Scribe combines algorithmic

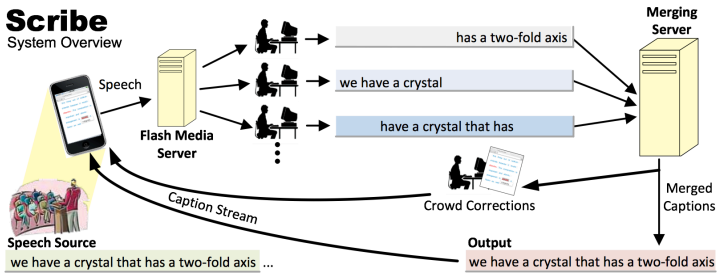


Figure 1: The architecture of Scribe. Image originally appeared in Lasecki et al. (2012).

techniques with the power of the crowd. As soon as a user starts recording, the recorded audio is sent simultaneously to several crowdworkers. These workers are not expected to fully transcribe the speech, which is generally not possible without a specialized stenotype keyboard. Instead, each worker transcribes sentence fragments. Scribe adjusts the speed and volume of the speech adaptively for each worker in order to focus workers' attention on distinct, overlapping components. It then uses multiple sequence alignment techniques (Edgar and Batzoglou, 2006; Lermen and Reinert, 2000; Naim et al., 2013) to combine the workers' text into one complete and coherent stream that is delivered back to the user with a delay of only a few seconds.

More recently, Gaur et al. (2015, 2016) developed an approach in which a single crowdworker is used to correct mistakes made by an automatic speech recognition system in real time. The crowdworker views the transcription output by a speech recognition system while listening to the corresponding audio. She then types out corrections to mistakes in the transcription as she notices them. These corrections are automatically incorporated into the appropriate spot in the transcription. Initial experiments showed that when this system was run with crowdworkers from Mechanical Turk, the word error rate improved. However, only 30% of possible corrections were made, leaving significant room for improvement in future work (Gaur et al., 2015).

As new breakthroughs continue to improve automatic speech recognition (Yu and Deng, 2014), this hybrid approach may eventually become unnecessary. However, the ability for speech recognition systems to achieve true human parity under nonideal conditions likely remains far in the future (Bigham, 2017). This example shows that crowdsourcing can be an effective way of compensating for a lack of sufficient machine learning or AI solutions until a time when the technology improves.

4.3 Hybrid Scheduling

Several researchers have explored the potential use of hybrid intelligence systems to solve complex tasks with global constraints or consistency requirements. Examples of such tasks include itinerary planning (Zhang et al., 2012), taxonomy creation (Bragg et al., 2013;

Chilton et al., 2013), and writing (Bernstein et al., 2010; Kim et al., 2017; Kittur et al., 2011; Salehi et al., 2017; Teevan et al., 2016). In this section we describe Cobi[9] (André et al., 2013; Bhardwaj et al., 2014; Chilton et al., 2014; Kim et al., 2013), a hybrid conference scheduling system. Rather than enlisting the help of anonymous crowdworkers, Cobi is based on the idea of "communitysourcing." It draws on the specialized expertise of people within a research community.

The problem of scheduling conference sessions can be viewed as a constrained optimization problem in which the solver has no direct access to the constraints. The goal of conference organizers is to group similar talks together in sessions while minimizing conflicts between talks that are scheduled at the same time, but conference organizers generally do not know which sets of talks attendees want to see. This optimization problem can be large; for example, Cobi was used to map out the schedule of the Conference on Human Factors in Computing Systems (CHI) in 2013, which required scheduling 400 talks in 16 parallel tracks.

Cobi was designed to efficiently collect information about attendee preferences from the community and use this information to optimize the conference schedule. (It is worth noting that the conference chairs always retain control and can choose to overwrite the optimized schedule.) As one example, their "authorsourcing" component presents authors with papers that are potentially similar to their own and asks which ones would be a good fit to appear in the same session. In order to generate the lists of potentially similar papers in the first place, their "committeesourcing" component makes use of hybrid clustering techniques like those discussed in Section 4.1 (André et al., 2013).

Communitysourcing can be especially effective because the same people who are asked to provide information also benefit from high quality end results. Although participants were not directly compensated, when Cobi was first deployed at CHI, the authors of 87% of accepted submissions opted to engage with the system.

Cobi was subsequently deployed several times for scheduling at both CHI and the ACM Conference on Computer-Supported Cooperative Work and Social Computing (CSCW). Anecdotally, users of the system found that the constraints generated by the authorsourcing component were crucial for producing a schedule of coherent and nonconflicting sessions; without this input the scheduler performed poorly.

4.4 Hybrid Forecasting

Significant resources are devoted to producing forecasts about geopolitical events and economic indicators. Humans are flexible in their ability to reason about arbitrary events, but human forecasts can be limited by cognitive biases or the inability to digest and process information at scale. Statistical and data-driven models, on the other hand, are able to take advantage of vast quantities of available data, but are difficult to design and train for one-of-a-kind events. Hybrid forecasting systems aim to combine the computational power of machines with the flexibility of humans to produce accurate forecasts.

In the most basic hybrid forecasting systems, algorithmic techniques are used primarily as a way to elicit and aggregate human-generated forecasts. One common example is a prediction market (Berg et al., 2008; Wolfers and Zitzewitz, 2004), a financial market in which traders can buy or sell securities with payoffs that are linked to future events. For

9. http://projectcobi.com

example, in an election market, traders might buy or sell a security that is worth \$1 if the incumbent candidate wins and nothing otherwise. If a trader believes that the probability of this candidate winning is p and wants to maximize her expected payoff, then she should be willing to buy this security at any price less than \$$p$, since with probability p she would get \$1. Similarly, she should be willing to sell at any price greater than \$$p$. For this reason, we can think of the current market price of this security as capturing traders' collective beliefs about how likely it is that the incumbent will win. Prediction markets can be operated as continuous double auctions, much like the stock market, requiring very little algorithmic ingenuity. However, when the level of trade is low, there can be advantages to operating prediction markets using algorithmic market makers that automatically set prices based on the history of trade (Chen and Pennock, 2007; Hanson, 2003). Prediction markets have recently gained more attention in the machine learning community due to the discovery of strong mathematical connections between these algorithmic market makers and no-regret online learning algorithms (Abernethy et al., 2013; Chen and Vaughan, 2010).

As part of the Good Judgment Project (Schoemaker and Tetlock, 2016; Tetlock et al., 2017; Ungar et al., 2012), a large-scale project funded by U.S. Intelligence, researchers evaluated and compared a wide range of algorithmic techniques for eliciting and aggregating human forecasts, including prediction markets and prediction polls, in which forecasters are asked to directly provide a probability estimate about the likelihood of an event. Through a series of randomized controlled trials, they found that prediction markets produce more accurate forecasts than those obtained by simply averaging forecasts from prediction polls. However, even higher accuracy could be obtained by aggregating prediction poll forecasts using more clever statistical methods (Atanasov et al., 2017) and extremizing aggregated forecasts (Baron et al., 2014). They also studied the importance of identifying and teaming up the top-performing individual forecasters (Mellers et al., 2015b) as well as the psychological traits shared by these top forecasters (Mellers et al., 2015a).

Building on lessons learned from the Good Judgment Project, the U.S. Office of the Director of National Intelligence has recently launched a new program aimed at producing hybrid forecasting systems that more comprehensively integrate modern data-driven approaches with human forecasting capabilities.[10]

4.5 Hybrid Intelligence Systems in Industry

So far we have focused on hybrid intelligence systems that have come out of the research community, but it is worth mentioning that human-in-the-loop systems are widely used in industry as well. To name just a few examples, Stitch Fix, which provides personalized style recommendations, trains machine learning algorithms to suggest items that a user might like and then sends the output of these algorithms to a human expert to curate and pare down.[11] Twitter employs contract workers to interpret search terms that suddenly spike in meaning, often due to recent mentions in the media or pop culture that an algorithm trained on stale data would miss.[12] PatternEx uses machine learning to identify suspicious activity that could indicate a security threat. This suspicious activity is then examined by a

10. https://www.iarpa.gov/index.php/research-programs/hfc
11. http://multithreaded.stitchfix.com/blog/2016/03/29/hcomp1/
12. http://nyti.ms/2gzOo4K

human, who determines whether there is a real attack, and the human's feedback is used to improve the system.[13] And even search engines like Google and Bing can be viewed as hybrid intelligence systems, since the relevance judgments that are used to adaptively improve the systems are generated by humans (Alonso, 2013; Alonso et al., 2008; Kazai, 2011).

5. Behavioral Studies to Inform Machine Learning Research

Within the past few years, there has been an increased interest, both from within and outside the machine learning community, in understanding how humans interact with machine learning systems. Researchers are striving to make machine learning models human-interpretable (Doshi-Velez and Kim, 2017; Jung et al., 2017; Koh and Liang, 2017; Lipton, 2016; Lou et al., 2012, 2013; Paul, 2016; Ribeiro et al., 2016; Ustun and Rudin, 2016), to make machine learning tools easier to use (Brooks et al., 2015; Patel et al., 2010; Simard et al., 2017), and to understand how algorithmic decisions impact people's lives (Angwin et al., 2016; Barocas and Selbst, 2016; Campolo et al., 2017; Chouldechova, 2017; Corbett-Davies et al., 2017; Sweeney, 2013).

During the same time period, psychologists and social scientists have increasingly turned to crowdsourcing platforms to run behavioral experiments that traditionally would have been conducted on undergraduates in a physical lab. One of the papers frequently credited with introducing Mechanical Turk to the psychology community, Buhrmester et al.'s "Amazon's Mechanical Turk: A New Source of Inexpensive, Yet High-Quality, Data?" (Buhrmester et al., 2011), has been cited almost 5500 times according to Google Scholar, and this number continues to rise. Crowdsourcing platforms provide fast and easy access to large pools of subjects, and promote faster iteration between the development of new theories and experimentation, allowing researchers to speed up their overall research process (Mason and Suri, 2012). Additionally, studies have shown that classic psychology and behavioral economics results can be replicated using crowdworkers (Horton et al., 2011; Paolacci et al., 2010; Simons and Chabris, 2012; Suri and Watts, 2011). On the down side, there are some concerns about overuse of subjects (Chandler et al., 2014; Peera et al., 2017), with the same crowdworkers participating in many variants of the same standard psychology experiments, with which they have become familiar.

These developments open up the opportunity for interdisciplinary research that uses behavioral experiments conducted on crowdsourcing platforms to improve our understanding of how humans interact with machine learning and AI systems. This line of work goes beyond the idea of using crowdsourcing to evaluate one particular machine learning model or user interface, as discussed in Section 3, and instead seeks a general understanding of the components of human behavior that could inform the use and design of machine learning systems more broadly. Such experiments can help us develop better models of human behavior that could be used, in turn, to develop better algorithms and interfaces (Chen et al., 2016).

It is worth noting that behavioral experiments and other crowdsourced user studies could benefit other subfields of computer science as well, and there are already examples where it has. For example, Mechanical Turk experiments have been used to study lay people's perceptions of password security (Ur et al., 2016) and perceptions of graphics and visualizations (Heer and Bostock, 2010).

13. http://tek.io/1Vy01KB

Compared with the bodies of work on crowdsourcing for data generation, model evaluation, and hybrid intelligence, there is relatively little research in this area to date. In this section, we describe a few examples of behavioral studies using crowdsourcing platforms that have the potential to change the way we think about applications of machine learning in the hope that these examples will inspire additional research.

5.1 Understanding Trust in Predictive Models

There is a large body of work cutting across psychology, management, and other research communities that studies human trust in algorithmic predictions and models (Dietvorst et al., 2015, 2016; Dzindolet et al., 2002; Logg, 2017; Promberger and Baron, 2006; Sinha and Swearingen, 2001; Yeomans et al., 2017). While the earlier work was performed in traditional labs, the newer studies primarily rely on crowdworkers or a mix of crowdworkers and in-person participants (Dietvorst et al., 2015, 2016; Logg, 2017; Yeomans et al., 2017). This body of work provides invaluable insights about how real end users interact with machine learning systems in practice that can be put to use immediately in the design of models and user interfaces.

As one example, there is significant evidence in the literature of algorithm aversion, a phenomenon in which people fail to trust an algorithm once they have seen the algorithm make a mistake, even if the algorithm outperforms human predictors (Dietvorst et al., 2015; Dzindolet et al., 2002). This is worrisome since essentially all machine learning models make errors at least occasionally. Dietvorst et al. (2016) ran a sequence of experiments, some on crowdworkers and some on students, to examine the question of whether algorithm aversion can be overcome using simple interventions. In particular, the researchers asked whether people would choose to use an algorithm more if they were given the ability to intervene and make minor adjustments to the algorithm's prediction when they believed it to be wrong.

In one study, they asked crowdworkers to predict students' test scores on a standardized test using nine features such as the student's favorite subject in school and the region of the country the student lives in. Workers would be paid based on the accuracy of their predictions. The workers were told that analysts had trained a model to make the same predictions and were given the average error of the model (and thus explicitly told that the model was not perfect). They were then asked to decide whether they wanted to use the model or not. The workers were assigned to one of four randomized conditions. In the first, workers were told that if they chose to use the model they would not be allowed to adjust the predictions of the model at all. In the other three, workers were told that they would be given the ability to adjust each prediction of the model by up to 2 points, 5 points, or 10 points respectively. The researchers found that participants were indeed significantly more likely to choose to use the model if they were given the ability to intervene and adjust the model's prediction. In both the adjust-by-5 and adjust-by-10 conditions, 71% of participants chose to use the model, along with 68% in the adjust-by-2 condition. On the other hand, those who would not have the opportunity to intervene only chose to use the model 47% of the time.

Participants in the conditions in which adjustments were allowed also had significantly better predictive accuracy than those who were not allowed to adjust the model. This is not because their adjustments helped; on the contrary, participants would have done better

by always following the model exactly. They were more accurate because they were more willing to rely on the model.

This finding has immediate implications about which strategies might help gain user trust in machine learning models. In particular, users might be more willing to trust a model if they have the ability to intervene. Even if this human intervention leads to a worse prediction, allowing the intervention may still be beneficial because the user will be more likely to use the model in the first place.

It is natural to imagine that similar ideas could be applied to study model interpretability. As one very recent step in this direction, Poursabzi-Sangdeh et al. (2018) ran a large-scale human-subject experiment on Mechanical Turk that was designed to test how two different attributes of a model (the number of features and whether the details of the model are presented to the user or the model is presented as a black box) impact either users' understanding of the model or users' trust in the model. Unlike the experiments described in Section 3.2, in which crowdworkers were used to evaluate the interpretability of one specific model, Poursabzi-Sangdeh et al. (2018) tried to uncover how different factors of a model influence different aspects of interpretability more broadly, with the goal of providing general guidance on how to develop future interpretable models.

5.2 Understanding Reactions to Ads

Internet advertising is a huge business. Revenues from internet advertising in the U.S. alone hit $72.5 billion in 2016, up 22% from the previous year.[14] Naturally, significant effort is devoted to developing and optimizing machine learning algorithms for online advertising, especially in industry. New algorithms are generally put through rigorous A-B testing before deployment to quantify and measure their impact when run on real users. However, one might ask whether it is possible to design better algorithms using general insights about human behavior that transcend any one particular algorithm.

Goldstein et al. (2013; 2014) set out to quantify the impact of "annoying" display ads on user behavior, under the hypothesis that web publishers might be losing money and driving away users by displaying annoying ads. They designed a two-stage experiment run on Mechanical Turk. The first stage was designed to identify examples of "good" and "bad" ads. To do this, they presented crowdworkers with overlapping sets of display ads and asked for judgments of how annoying the ads were, a standard data labeling task as discussed in Section 2.1. By aggregating these judgments across workers, they produced lists of the most and least annoying banner ads.

In the second stage, Goldstein et al. (2013; 2014) ran a behavioral experiment aimed at estimating how much monetary value web users get from not being subjected to annoying ads. They posted another Mechanical Turk task in which workers were asked to label emails from the Enron email database as either spam or not spam. Workers were randomly assigned to different experimental conditions. Some workers saw good ads (as determined in stage 1) next to each email, while others saw bad, annoying ads. A third group saw no ads at all. The researchers also randomly varied how much users were paid for labeling each email. Workers were free to label as many emails as they wanted. By comparing the number of emails that workers chose to classify in each experimental condition, the

14. https://read.bi/2Ks0StR

researchers produced an estimate of the amount of extra money it would be necessary to pay workers who were exposed to bad ads in order to get them to perform the same number of email classification tasks as those exposed to good ads or no ads at all. From that, they were able to measure people's annoyance at bad ads.

They found that people were willing to classify almost as many emails when shown good ads compared with no ads at all, suggesting that displaying good ads does not hurt publishers. The same was not true for annoying ads. The researchers estimated that they would have to pay approximately $1 extra to generate 1000 views using a bad ad compared with a good ad or no ad. This is a significant amount as a typical banner ad might cost $1-$5 per 1000 views. In other words, publishers may very well be losing money by displaying annoying ads unless they charge significantly more per view.

While there are some limitations to the applicability of these results (for example, it is plausible that people react differently to ads when performing classification tasks than they would when browsing the news), it is a valuable step towards a model of user reactions to annoying ads. It is easy to imagine that such a model would prove useful when designing machine learning algorithms for pricing and displaying banner ads.

6. Understanding the Crowd

In the previous section, we argued that crowdsourced studies of human behavior can be valuable for understanding how lay people interact with machine learning systems. In this section, we argue that such studies are also useful for understanding the behavior of the crowd itself. This understanding helps us better model the crowd and allows us to define concrete recommendations of best practices that can be put to use whether using the crowd for data generation, model evaluation, hybrid intelligence systems, behavioral research, or any other purpose.

The studies described in this section help us understand how real crowdworkers respond to incentives, yielding immediately applicable guidelines for setting payments in addition to more accurate ways of modeling crowd behavior in theoretical work on incentive design. They tell us how to most effectively gamify crowdwork and provide other sources of intrinsic motivation for workers. They help us get a grip on the question of how widespread dishonesty is on crowdsourcing platforms, and how dishonest behavior can be mitigated. And they show us that crowdworkers are not independent and isolated workers, but have a rich social network.

6.1 Crowdworker Demographics

Over the years there have been several studies published that examine the demographics of workers on Mechanical Turk. We mention only a few key statistics that help paint a picture of the worker pool. These are based on a November 2016 snapshot from MTurk Tracker,[15] a project aimed at tracking the demographics of Mechanical Turk over time by continually releasing tasks containing demographic surveys on Mechanical Turk to obtain up-to-date information about workers (Difallah et al., 2015).

15. http://www.behind-the-enemy-lines.com/2015/04/demographics-of-mechanical-turk-now.html

These statistics should be taken with a grain of salt for several reasons. First, worker demographics change over time. For example, there is evidence that the worker pool on Mechanical Turk is shifting to be more heavily composed of American workers due to changes in Amazon's rules and regulations (Silberman et al., 2015). Second, not all workers on Mechanical Turk choose to work on surveys, so these statistics perhaps better reflect the population of workers who do survey work. That said, these demographics are more or less in line with those reported in other contemporaneous studies.

According to the MTurk Tracker data:

- About 70-80% of Mechanical Turk workers are from the United States, while about 10-20% are from India, but the breakdown of workers varies significantly throughout the day. The prevalence of workers from the U.S. and India makes sense because Mechanical Turk offers payment only in U.S. dollars, Indian rupees, or Amazon credit.
- The breakdown between male and female workers is fairly close to even, though it varies a bit by country.
- For crowdworkers in the U.S., the (self-reported) median household income is in the range of $40K-$60K, which is in line with the median U.S. household income. The median for Indian workers is less than $15K, with many Indian workers reporting a household income of less than $10K per year.

There is evidence that other crowdsourcing platforms, such as CrowdFlower and Prolific Academic, attract more European workers and lower income workers than Mechanical Turk (Peera et al., 2017).

Goodman and Paolacci (2017) provide a nice overview of the similarities and differences between the population of Mechanical Turk workers and the U.S. population as a whole, as well as the populations traditionally used for consumer studies.

The demographics of available workers vary widely based on the time of day and, to a lesser extent, day of the week. Through a large study of intertemporal demographic differences on Mechanical Turk, Casey et al. (2017) found that, even restricting attention to workers from the U.S., the demographics of available workers change dramatically over the course of a day. For example, they found that workers who completed their task at night were more likely to be single than those who completed it in the morning, and more likely to be completing the task on a smartphone.

6.2 Dishonesty Among Workers

Researchers may be deterred from using crowdsourcing due to concerns about the prevalence of dishonest workers. In this section, we discuss the results of behavioral experiments aimed at quantifying the presence of dishonest behavior on Mechanical Turk.

Suri et al. (2011) borrowed a trick from a lab study run by Fischbacher and Föllmi-Heusi (2013) that allowed them to measure how trustworthy crowdworkers are on the whole without the ability to detect individual lies. They posted a task on Mechanical Turk in which each worker was asked to roll a die (or simulate rolling a die on an external website) and report the value of her roll, a random number between 1 and 6. For completing this task, the worker received a base payment of $0.25 plus an additional bonus of $0.25 times her reported roll. For example, if a worker reported rolling a 4, she would receive a total

payment of $1.25. Thus the workers, who knew that there was no way for the outcomes of their rolls to be verified, were given a direct incentive to lie.

If all workers honestly reported the outcome of a die roll, the mean of the reported rolls of the 175 participants in the study would be close to 3.5. The researchers instead observed a mean reported roll of 3.91. On the whole, workers had a tendency to overreport rolls of 5 and 6 and underreport rolls of 1 and 2. In fact, it is possible that roughly a third of participants who rolled a 1 or 2 chose to misreport. (It is also possible that some participants ignored the instructions to roll a die at all, choosing to report an arbitrary number.)

Suri et al. (2011) next examined whether increasing the ability of the task requester to detect cheating would lead to higher levels of honesty. They conducted a variant of the die rolling experiment in which each worker was asked to report the results of a sequence of thirty die rolls instead of just one. Workers in this experiment received a base payment of $0.25 plus the sum of their reported die rolls, for a total payment between $0.55 and $2.05. This time, the mean reported role of the 232 participants was 3.57—still statistically significantly higher than the expected mean of 3.5, but practically much closer. Only three of the 232 experiment participants were found to have an average report significantly greater than what would be expected if they reported their rolls honestly, and only one reported the payoff-maximizing sequence of thirty rolls of 6. This suggests that dishonest behavior may be less prevalent when there is a chance that this behavior will be detected (though the results of the two experiments are not directly comparable since they were performed at different times on potentially different populations).

Two recent contemporaneous studies (Chandler and Paolacci, 2017; Wessling et al., 2017) examined whether crowdworkers misrepresent themselves in order to satisfy the eligibility criteria of tasks. Chandler and Paolacci (2017) asked over two thousand crowdworkers to report whether they are the parent or guardian of a child with autism. Workers were randomly assigned to one of two experimental conditions. In one, workers were told that this question was part of a prescreening test for a subsequent study, while in another (the control), they were not. 7.8% of workers in the prescreening condition reported being the parent or guardian of a child with autism, compared with 4.3% in the control, suggesting that at least 3.5% of workers in the prescreening condition provided a fraudulent response. While 3.5% of workers may seem small, note that due to the low number of workers who would truly qualify for a follow-up study based on this question, this level of dishonesty would lead to a high prevalence of impostors (45%) in a follow-up study. In a separate study, the same researchers found that when payments were sufficiently high, 16% of participants made a second attempt to pass a prescreening survey, identifying themselves as a different gender after initially being blocked. Similarly, Wessling et al. (2017) provided additional evidence of misrepresentation in prescreening tests.

Based on these findings, both papers include detailed discussions of best practices that can be followed to mitigate dishonesty in prescreening tests. One recommendation is to collect screening data as part of a stand-alone task ahead of time and call back eligible participants later. As an alternative, when costs are not prohibitive, all workers can be allowed to perform a task and those who do not meet the screening criteria can be filtered out after all data have been collected.

All of this work suggests that the rate of dishonest behavior and spam depends on the particular task and the motivation to lie. There is some anecdotal evidence that spammers

are especially drawn to surveys (Mason and Suri, 2012) and multiple-choice questions (Rao and Michel, 2016) since these are especially easy tasks to complete, so rates of spam may be even higher on these questions.

It is worth noting that requesters on crowdsourcing platforms can be dishonest or otherwise malicious too. For example, requesters may ask workers to create social media accounts to post specific content or engage in other dodgy Internet marketing practices. It fact, there are now entire crowdsourcing platforms in China specifically devoted to these "crowdturfing" tasks (Wang et al., 2012a).

6.3 Monetary Incentives

One of the first questions that many researchers have when they decide to incorporate crowdsourcing into their work is how much to pay per task. When crowdsourcing first began to gain popularity among researchers, part of the appeal was the ability to generate data or run experiments cheaply. For example, Snow et al. (2008) boasted that they were able to offer workers $0.02 to complete a set of 30 annotations, obtaining 1500 annotations per dollar. However, over the last decade, the views of the community have shifted as the ethics of crowdsourcing have received more attention (Kittur et al., 2013; Salehi et al., 2015; Silberman et al., 2010; Williamson, 2016). Although crowdworkers are legally considered contractors, making minimum wage laws inapplicable, guidelines put forth by the Dynamo project[16] (Salehi et al., 2015) recommend paying the equivalent of the current U.S. federal minimum wage or more. An effective and widely used method of setting the payment for a task is to first estimate the time it takes to complete the task (for example, by asking colleagues or students to try out the task, or by posting a small test batch of tasks) and then use that estimate to ensure that the per-hour payment is higher than U.S. minimum wage.

A natural question is whether paying higher wages increases the quality of work. There is evidence that, at least in some scenarios, the answer is no. Several behavioral studies examining the impact of payments on quality found that setting higher payments increased the quantity of work that crowdworkers were willing to do, but not the quality (Buhrmester et al., 2011; Litman et al., 2014; Mason and Watts, 2009; Rogstadius et al., 2011). Mason and Watts (2009) conjectured that this might be due, at least in part, to an anchoring effect: workers' opinions about fair payment rates are anchored by the payments they are offered. In one of their experiments, workers were asked to sort sets of images and were randomly assigned to experimental conditions in which they received $0.01 per task, $0.05 per task, or $0.10 per task respectively. (All tasks were advertised at a rate of $0.01 and additional payments were made via bonuses to avoid selection effects.) In a post-hoc survey, workers who were paid $0.01 felt they should have received $0.05 for the task on average, while workers who were paid $0.05 felt they should have received $0.08, and workers who were paid $0.10 felt they should have received $0.13.

On the other hand, two recent studies have shown that in other scenarios, higher payments do increase work quality (Ho et al., 2015; Ye et al., 2017). There are several possible reasons for this discrepancy. One is that the type of task being performed might impact the effectiveness of increased payments. Ho et al. (2015) and Ye et al. (2017) both used "effort-responsive tasks"—tasks for which workers are able to improve their output by spending

16. `http://wiki.wearedynamo.org/index.php/Guidelines_for_Academic_Requesters`

more time or effort—in their studies, whereas the study mentioned above used a task that was fairly easy to complete without much effort. Mason and Watts (2009) did use an effort-responsive task (solving word puzzles) in a second study, but found that for this particular task, worker quality was more correlated with enjoyment of word puzzles than the rate of pay offered, an argument that intrinsic motivation (the topic of Section 6.4) can overpower monetary incentives. Another possibility is that in some studies, payments were so low that the differences between conditions were not salient. Buhrmester et al. (2011) offered either $0.02, $0.10, or $0.50 for up to 30 minutes of work, while Ho et al. (2015) and Ye et al. (2017) aimed to pay at least minimum wage and had larger per-hour gaps between payments in different experimental conditions.

While this research is somewhat inconclusive, it does appear that paying higher wages consistently increases the number of tasks that crowdworkers are willing to perform and therefore speeds up the rate at which a requester's tasks are completed.

Another body of work is aimed at answering the question of whether the quality of crowdwork can be improved through the use of performance-based payments, payment schemes that explicitly reward crowdworkers for higher quality work (Harris, 2011; Ho et al., 2015; Shaw et al., 2011; Yin et al., 2013, 2014). While in theory, such payment schemes could be arbitrarily complex (Ho et al., 2016), in practice they are generally implemented in the form of a bonus payment awarded for exceeding a particular quality threshold. Bonus payments of this form are common on Mechanical Turk.

Here, too, results have been mixed. Harris (2011) asked workers to evaluate the relevance of resumes and found that performance-based payments increased both quality and the amount of time that workers spent on the task. On the other hand, Shaw et al. (2011) compared fourteen incentives schemes, including four that involved performance-based payments, and did not find significant increases in quality, and Yin et al. (2013) varied the bonus sizes offered to workers and found no significant difference in quality between experimental conditions.

The most comprehensive experimental study of performance-based payments was performed by Ho et al. (2015). They showed that performance-based payments can improve quality for particular tasks (again, those that are effort-responsive). They found that the effectiveness of performance-based payments is not heavily dependent on the precise quality threshold chosen. They also tested how sensitive their results were to the size of the bonus offered and found that as long as the bonus offered was big enough, quality improved. Offering a very small bonus (say, $0.05 on a $0.50 base payment) actually led to a small apparent decrease in performance, though this decrease is not statistically significant. This may explain the negative results of Shaw et al. (2011), since their bonus payments were very small ($0.03 on a base payment of $0.30). It may also explain the results of Yin et al. (2013), who considered only bonuses that were relatively large compared with the base, a regime in which Ho et al. (2015) also saw no statistically significant differences in quality when varying the bonus size.

It is not always immediately apparent whether or not a task is effort-responsive. As an example, Ho et al. (2015) were surprised to find that handwriting recognition is not. When they gave workers a handwriting recognition task, the majority of workers were able to identify most words with little effort. When workers could not make out a word, additional time spent did not help. The researchers suggest that to determine whether or not a task

is effort-responsive, a requester should run a pilot experiment in which they ask workers to complete the task for a fixed payment and examine the relationship between time spent and quality of results. The results of such a pilot could be used to determine an appropriate payment scheme for the task.

6.4 Intrinsic Motivation

In addition to monetary incentives, researchers have also explored the ways in which intrinsic sources of motivation, such as having fun or doing meaningful work, affect the quantity and quality of work that crowdworkers perform. This research is important for informing the design of volunteer-based or citizen science platforms, like the Zooniverse[17] and Science at Home[18], as well as paid crowdsourcing platforms like Mechanical Turk.

In one study, Chandler and Kapelner (2013) found that crowdworkers are more active when the tasks they are asked to perform are framed as meaningful. The researchers recruited workers on Mechanical Turk to label medical images. In one experimental condition, workers were told that they were labeling tumor cells and that the results of their work would be used to assist medical researchers. In the control condition, they were given no context for the task at all. In a third condition, they were given no context and additionally told that the labels they generated would not be recorded; that is, all of their work would be discarded. The researchers found that when workers were told their work would benefit medical research, the quantity of work that they produced increased compared with the control, but their work was not significantly more accurate. On the other hand, when workers were told their work would be discarded, the quality of their work was worse than the control, but the quantity of work produced was similar. Similar effects were observed by Rogstadius et al. (2011), who compared the behavior of workers who were told they were performing work for a nonprofit organization "dedicated to saving lives by improving health throughout the world" with workers who were told they were working for a for-profit pharmaceutical manufacturer. This work suggests that when the task being performed has the potential to do good in the world, it is worth emphasizing this to the workers.

Gamification is an another effective way of increasing crowdworkers' motivation that can be applied in both paid and unpaid crowdsourcing settings (Feyisetan et al., 2015; Lee et al., 2013; von Ahn and Dabbish, 2008). Von Ahn and Dabbish (2008) pioneered the study of "games with a purpose," computer games in which players, as a side effect of their play, accomplish tasks that are difficult for machines, such as generating training data for machine learning algorithms. For example, in the ESP Game (von Ahn and Dabbish, 2004) (now the Google Image Labeler), randomly matched pairs of players are presented with an image and asked to type in words that describe it (i.e., labels), as in Figure 2. The players receive points if both members of the pair produce the same word. Since players are unable to communicate with each other, their best strategy for producing the same word is to type in words that are relevant to the image, thus providing useful labels. As another example, Verbosity (von Ahn et al., 2006) is a game designed to collect common-sense facts about objects. Players are again randomly paired, with one partner serving as a describer and the other as a guesser. The describer is given an object (such as a sock) and asked to list

17. https://www.zooniverse.org
18. https://www.scienceathome.org

Figure 2: A screenshot of the ESP Game. Image originally appeared in von Ahn and Dabbish (2008).

facts about the object ("it is a kind of clothing" or "it is related to feet"). The guesser is shown these facts and must try to guess the object as quickly as possible. To win, it is in the describer's best interest to come up with facts that succinctly and accurately describe the object, which then become a part of a database of common knowledge statements.

Von Ahn and Dabbish (2008) credit the success of these games in part to following known game-design principles, such as incorporating extra challenges through game features like timed responses, score keeping and leaderboards, and randomness (Malone, 1980, 1982). However, there are some caveats of gamification. Hamari et al. (2014) recently conducted an extensive survey of the literature on gamification and found that it is more successful with some types of users than others, and may generally be less effective in scenarios in which people are prone towards exhibiting rational behavior, which may include paid crowdsourcing sites since at least some crowdworkers are already be in the mindset of maximizing pay.

Law et al. (2016) examined the possibility of appealing to workers' curiosity as a source of intrinsic motivation. Their work was inspired by the information gap theory of curiosity, which suggests that when people are made aware that there is a gap in their knowledge, they actively seek out the information needed to fill in this gap. They suggested several "curiosity interventions" aimed at stoking workers' curiosity and showed that these interventions led to workers completing more tasks and performing better. Curiosity interventions are especially effective on tasks that are inherently less interesting.

6.5 Crowdworker Communication and Forum Usage

Finally, several recent papers have explored the questions of how, why, and to what extent crowdworkers communicate with each other.

Gray et al. (2016) conducted a mixed-method study to understand communication among crowdworkers on four crowdsourcing platforms. Through extensive ethnographic fieldwork and in-person interviews with 118 crowdworkers in India, the researchers uncov-

ered three common categories of communication among workers. First, workers help each other with administrative overhead in order to reduce their costs. For example, a crowdworker might seek help figuring out how to receive her payments from a crowdsourcing platform, which can be nontrivial for Indian workers. Second, crowdworkers share information about good tasks and reputable (or irreputable) task requesters. Third, crowdworkers help each other complete specific tasks. More generally, crowdworkers seek each other out to recreate the social connections and support structures that are common in most traditional jobs but missing from crowdwork. This idea that crowdworkers communicate and collaborate with each other is supported by the work of Gupta et al. (2014), who also conducted interviews of crowdworkers in India.

This line of work suggests that crowdworkers are not independent, but rather that there is a hidden communication network among workers. Yin et al. (2016) attempted to quantify and map the hidden network of workers on Mechanical Turk in order to better understand its scale and structure as well as how is it used. To do this, the researchers designed and launched a task on Mechanical Turk. When a worker accepted the task, she was first asked to create a nickname for herself. She then filled out a brief demographic survey and was asked to answer two free-form questions about her experiences on Mechanical Turk: why she started Turking and what motivated her to keep Turking. (These questions were selected based on the results of a pilot study in which crowdworkers were asked what they most wanted to know about other crowdworkers, with the goal of generating interesting questions.) The worker was then asked to pause and swap nicknames with other workers she knows who had already completed the task or might be interested in completing it. If she entered another worker's nickname, she was asked several questions about their communication patterns, and an edge was created between them in the network. Finally, the worker was given the chance to explore the partially constructed network, viewing basic information on all workers (including their answers to the two free-form questions above), and more extensive information about those workers with whom she had exchanged nicknames. She was also given a personalized link she could use to return to the network later and add more connections.

The resulting communication network is shown in Figure 3. Over a period of several weeks, 10,354 workers completed the task. (Stewart et al. 2015 estimated the number of active workers on Mechanical Turk to be only 7,300, so it is likely that this task was completed by a large fraction of active workers, mitigating potential issues with sample bias.) These 10,354 workers reported a total of 5,268 connections. Since workers were not financially incentivized to add connections, this number is probably an underestimate of the true number of connected pairs. Roughly 13% of workers were connected to at least one other worker. On average these workers had 7.6 connections, and the maximum degree of any worker was 321. The largest connected component contained 994 workers, or about 72% of connected workers.

While many different methods of communicating were reported, by far the most common was through online forums. In fact, 90% of all edges were between pairs of workers who communicate via forums. This finding is in line with other work that examined the important role that forums play in crowdsourcing (Martin et al., 2014). Different online forums create different but overlapping subcommunities in the network, as illustrated in Figure 4. The researchers' analysis showed that these subcommunities differ in terms of topo-

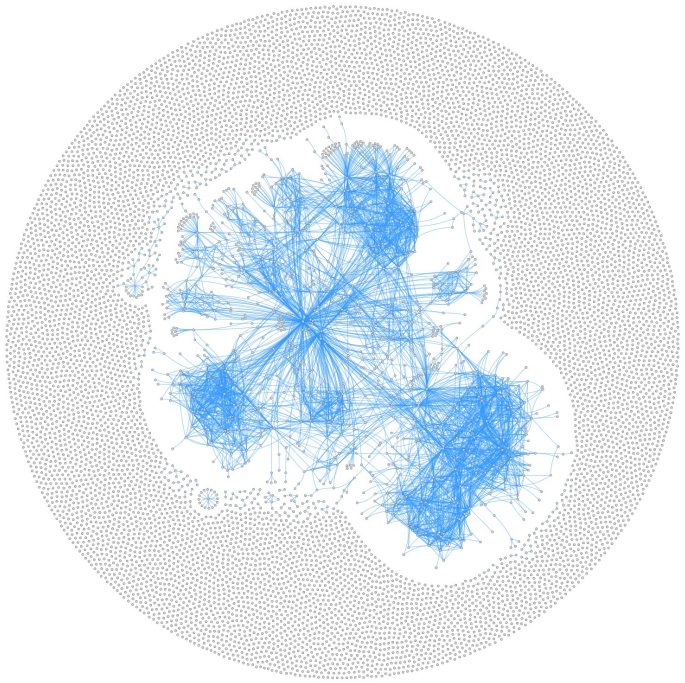
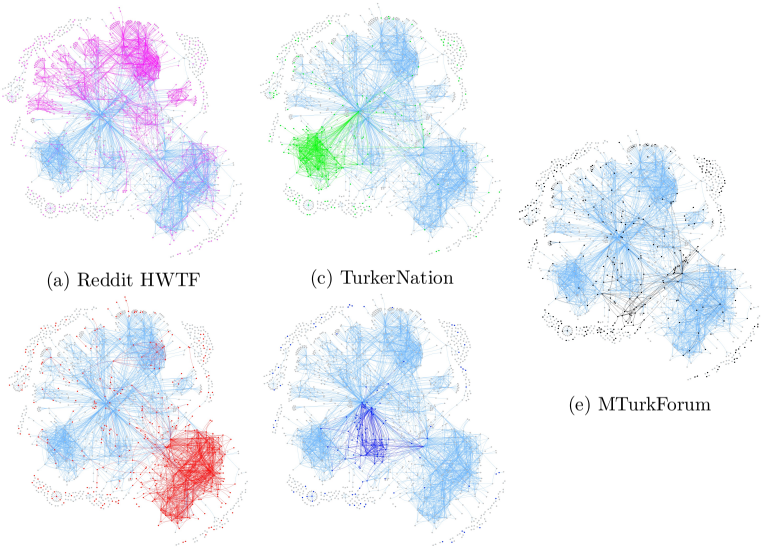


Figure 3: The communication network among Amazon Mechanical Turk workers. Image originally appeared in Yin et al. (2016).

logical structure, dynamics, and the content of communication, with some acting more as social communities and others more like broadcasting platforms.

Although it is impossible to establish causal claims on the basis of their study, Yin et al. (2016) did find correlations between a worker's connectivity and different measures of success on Mechanical Turk. In particular, they found that connected workers tended to find their task faster, were more likely to have been active on Mechanical Turk longer, were more likely to have achieved Mechanical Turk's "Master" qualification, and had a higher approval rate on average.

The high level of communication among crowdworkers and widespread use of forums have several immediate implications for researchers. First, researchers should keep in mind that the set of crowdworkers who choose to do a task may not be an independent sample of the worker pool since workers often share good tasks. Second, it can be in a researcher's best



(a) Reddit HWTF (c) TurkerNation (e) MTurkForum

(b) MTurkGrind (d) Facebook

Figure 4: Subnetworks for Reddit HWTF (magenta), MTurkGrind (red), TurkerNation (green), Facebook (blue), and MTurkForum (black). Images originally appeared in Yin et al. (2016).

interest to monitor popular forums such as Turker Nation[19], MTurk Forum[20], and Reddit HITs Worth Turking For[21] while their tasks are running to be aware of any potential issues that workers are discussing.

7. Discussion and Additional Best Practices

We have explored examples of four different ways in which machine learning researchers can put crowdsourcing to use in their own research: to generate data, to evaluate and debug models, to build hybrid intelligence systems, and to run behavioral experiments that inform the design of future machine learning systems. We have also reviewed the results of a variety of behavioral and user studies aimed at understanding the crowd and have discussed the implications of these studies for researchers who use crowdsourcing in their own research.

We conclude this survey with a discussion of additional crowdsourcing best practices that are rarely mentioned in the literature, despite their importance to the success of a project.

19. http://turkernation.com
20. http://www.mturkforum.com
21. https://www.reddit.com/r/HITsWorthTurkingFor/

7.1 Maintain a Good Relationship With Crowdworkers

There are several reasons why it can be valuable for a researcher to build a relationship with the community of crowdworkers and maintain a good reputation among them. As discussed in Section 6.5, workers share information about both tasks and requesters among themselves, especially through forums. Workers will be discouraged from accepting a task if other workers have complained about bugs, slow payments, or other issues. Experienced Mechanical Turk workers also commonly use tools like TurkOpticon[22] that allow them to view requester ratings broken down by communicativity, generosity, fairness, and promptness (Irani and Silberman, 2013). There are also tools available that allow workers to be notified when a favorite requester posts a new task. Being known as a good requester therefore brings more attention to one's tasks, while being known as a bad requester can deter experienced workers.

To maintain a good reputation, researchers should actively monitor and respond to any email or questions from workers so that any potential problems or bugs can be caught quickly. It is worth planning in advance to make sure that someone will be available to communicate with workers while a new task is running.

Researchers should pay fair wages (at least the equivalent of U.S. federal minimum wage, as discussed in Section 6.3) and approve work quickly since on many platforms workers are not paid until their work is approved. It is also good practice to avoid rejecting work. On Mechanical Turk, for example, a lowered approval rate can have a damaging effect on a worker's ability to find new work. Rejecting the work of a well-meaning worker who makes a mistake can therefore harm that worker's future income (Mason and Suri, 2012).

Finally, researchers should strive to be a ethical requesters. A good way to start is by reviewing the Dynamo project's guidelines for academic requesters[23] (Salehi et al., 2015).

7.2 Good Task Design Matters

Crowdworkers cannot be expected to excel at a task with unclear instructions or a confusing user interface. Any ambiguity will increase the rate of errors since it is difficult and time consuming for workers to ask clarifying questions (Rao and Michel, 2016). Both the instructions and UI should be piloted on a small batch of workers to make sure that they are clear before launching a task at a large scale. Some evidence suggests that including examples in a task's description or instructions is correlated with lower levels of confusion among workers and with workers more quickly accepting the task (Jain et al., 2017). Including quiz questions can also be an effective way to check for clarity. In addition to increased clarity, an attractive and easy-to-use UI may help keep workers engaged.

7.3 Pilot, Pilot, Pilot

Last but not least, the effective use of pilots is crucial to a project's success. Crowdsourcing platforms allow for quick and easy iteration of task design and experimentation (Mason and Suri, 2012). One of the primary benefits of crowdsourcing platforms is the ability to quickly pilot new or modified task designs on small batches of crowdworkers before making a commitment. As is the case with any software system, tasks often have bugs that are

22. https://turkopticon.ucsd.edu
23. http://wiki.wearedynamo.org/index.php/Guidelines_for_Academic_Requesters

hard to catch until they are deployed on real users. It is often best to launch a task slowly, iterating as many times as necessary to ensure that the task is clear and bug-free.

Acknowledgments

This survey grew out of lecture notes that were originally prepared to accompany my tutorial "Crowdsourcing: Beyond Label Generation" at NIPS 2016. Thanks to all of the people—far too many to name—who sent pointers and suggestions of research to include. Thanks to Dan Goldstein, Chien-Ju Ho, Jake Hofman, Andrew Mao, Roozbeh Mottaghi, Sid Suri, Jaime Teevan, Ming Yin, and Haoqi Zhang for discussing their research and sending along slides and other material while I was preparing the tutorial. Thanks to the authors of Lasecki et al. (2012) and von Ahn and Dabbish (2004) and to my coauthors on Yin et al. (2016) for allowing me to include images from their papers. Thanks to Chien-Ju Ho, Andrew Mao, Joelle Pineau, Sid Suri, Hanna Wallach, and especially Ming Yin for extended discussions and valuable feedback on previous versions of this survey. Finally, huge thanks to the four anonymous JMLR reviewers who went above and beyond their duty to give countless valuable comments, suggestions, and references that greatly improved the breadth and structure of this survey. This was a crowdsourced effort.

References

Jacob Abernethy, Yiling Chen, and Jennifer Wortman Vaughan. Efficient market making via convex optimization, and a connection to online learning. *ACM Transactions on Economics and Computation*, 1(2):Article 12, 2013.

Ittai Abraham, Omar Alonso, Vasilis Kandylas, Rajesh Patel, Steven Shelford, and Aleksandrs Slivkins. How many workers to ask? Adaptive exploration for collecting high quality labels. In *SIGIR*, 2016.

Arpit Agarwal, Debmalya Mandal, David C. Parkes, and Nisarg Shah. Peer prediction with heterogeneous users. In *ACM EC*, 2017.

Omar Alonso. Implementing crowdsourcing-based relevance experimentation: An industrial perspective. *Information Retrieval*, 16(2):101–120, 2013.

Omar Alonso, Daniel E. Rose, and Benjamin Stewart. Crowdsourcing for relevance evaluation. *ACM SigIR Forum*, 42(2):9–15, 2008.

Vamshi Ambati, Stephan Vogel, and Jaime Carbonell. Collaborative workflow for crowdsourcing translation. In *CSCW*, 2012.

Paul André, Haoqi Zhang, Juho Kim, Lydia B. Chilton, Steven P. Dow, and Robert C. Miller. Community clustering: Leveraging an academic crowd to form coherent conference sessions. In *HCOMP*, 2013.

Julia Angwin, Je Larson, Surya Mattu, and Lauren Kirchner. Machine bias: There's software used across the country to predict future criminals and it's biased against blacks. ProPublica article accessed at `https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing`, 2016.

Pavel D. Atanasov, Phillip Rescober, Eric Stone, Samuel A. Swift, Emile Servan-Schreiber, Philip E. Tetlock, Lyle Ungar, and Barbara Mellers. Distilling the wisdom of crowds: Prediction markets versus prediction polls. *Management Science*, 63(3):691–706, 2017.

Bahadir Ismail Aydin, Yavuz Selim Yilmaz, Yaliang Li, Qi Li, Jing Gao, and Murat Demirbas. Crowdsourcing for multiple-choice question answering. In *AAAI*, 2014.

Solon Barocas and Andrew Selbst. Big data's disparate impact. *California Law Review*, 104, 2016.

Jonathan Baron, Barbara A. Mellers, Philip E. Tetlock, Eric Stone, and Lyle H. Ungar. Two reasons to make aggregated probability forecasts more extreme. *Decision Analysis*, 11(2):133–145, 2014.

Joyce Berg, Robert Forsythe, Forrest Nelson, and Thomas Rietz. Results from a dozen years of election futures markets research. *Handbook of experimental economics results*, 1:742–751, 2008.

Michael Bernstein, Greg Little, Rob Miller, Bjoern Hartmann, Mark Ackerman, David Karger, David Crowell, and Katrina Panovich. Soylent: A word processor with a crowd inside. In *UIST*, 2010.

Anant Bhardwaj, Juho Kim, Steven P. Dow, David Karger, Sam Madden, Robert C. Miller, and Haoqi Zhang. Attendee-sourcing: Exploring the design space of community-informed conference scheduling. In *HCOMP*, 2014.

Jeffrey P. Bigham. Reaching dubious parity with hamstrung humans. Blog post accessed at `http://jeffreybigham.com/blog/2017/reaching-dubious-parity-with-hamstrung-humans.html`, 2017.

David M. Blei and John D. Lafferty. Topic models. *Text mining: Classification, clustering, and applications*, 10(71):34, 2009.

David M. Blei, Andrew Y. Ng, , and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022, 2003.

Jordan Boyd-Graber, Yuening Hu, and David Mimno. Applications of topic models. *Foundations and Trends in Information Retrieval*, 11(2–3):143–296, 2017.

Jonathan Bragg, Mausam, and Daniel S. Weld. Crowdsourcing multi-label classification for taxonomy creation. In *HCOMP*, 2013.

Michael Brooks, Saleema Amershi, Bongshin Lee, Steven Drucker, Ashish Kapoor, and Patrice Simard. FeatureInsight: Visual support for error-driven feature ideation in text classification. In *IEEE VAST*, 2015.

Michael Buhrmester, Tracy Kwang, and Samuel D. Gosling. Amazon's Mechanical Turk: A new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science*, 6(1):3–5, 2011.

Chris Callison-Burch and Mark Dredze. Creating speech and language data with Amazon's Mechanical Turk. In *NAACL HLT Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, 2010.

Alex Campolo, Madelyn Sanfilippo, Meredith Whittaker, and Kate Crawford. AI Now 2017 Report. Accessed at `https://ainowinstitute.org/AI_Now_2017_Report.pdf`, 2017.

Logan Casey, Jesse Chandler, Adam Seth Levine, Andrew Proctor, and Dara Z. Strolovitch. Intertemporal differences among MTurk worker demographics. Working paper on PsyArXiv, 2017.

Dana Chandler and Adam Kapelner. Breaking monotony with meaning: Motivation in crowdsourcing markets. *Journal of Economic Behavior and Organization*, 90:123–133, 2013.

Jesse Chandler, Pam Mueller, and Gabriele Paolacci. Nonnaïveté among Amazon Mechanical Turk workers: Consequences and solutions for behavioral researchers. *Behavior Research Methods*, 46(1):112–130, 2014.

Jesse J. Chandler and Gabriele Paolacci. Lie for a dime: When most prescreening responses are honest but most study participants are imposters. *Social Psychological and Personality Science*, 8(5):500–508, 2017.

Jonathan Chang, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, and David M. Blei. Reading tea leaves: How humans interpret topic models. In *NIPS*, 2009.

Shuchi Chawla, Jason D. Hartline, and Balasubramanian Sivan. Optimal crowdsourcing contests. *Games and Economic Behavior*, 2015.

Yiling Chen and David M. Pennock. A utility framework for bounded-loss market makers. In *UAI*, 2007.

Yiling Chen and Jennifer Wortman Vaughan. A new understanding of prediction markets via no-regret learning. In *ACM EC*, 2010.

Yiling Chen, Arpita Ghosh, Michael Kearns, Tim Roughgarden, and Jennifer Wortman Vaughan. Mathematical foundations of social computing. *Communications of the ACM*, 59(12):102–108, December 2016.

Lydia Chilton, Juho Kim, Paul André, Felicia Cordeiro, James Landay, Dan Weld, Steven P. Dow, Robert C. Miller, and Haoqi Zhang. Frenzy: Collaborative data organization for creating conference sessions. In *CHI*, 2014.

Lydia B. Chilton, Greg Little, Darren Edge, Daniel S. Weld, and James A. Landay. Cascade: Crowdsourcing taxonomy creation. In *CHI*, 2013.

Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data, Special Issue on Social and Technical Trade-Offs*, 2017.

Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness. In *KDD*, 2017.

Anirban Dasgupta and Arpita Ghosh. Crowdsourced judgement elicitation with endogenous proficiency. In *WWW*, 2013.

Susan B. Davidson, Sanjeev Khanna, Tova Milo, and Sudeepa Roy. Top-k and clustering with noisy comparisons. *ACM Transactions on Database Systems*, 39(4):35:1–39, 2014.

Philip Dawid and Allan Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, 28(1):20–28, 1979.

Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. Zencrowd: Leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *WWW*, 2012.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

Berkeley J. Dietvorst, Joseph P. Simmons, and Cade Massey. Algorithm aversion: People erroneously avoid algorithms after seeing them err. *Journal of Experimental Psychology: General*, 144(1):114, 2015.

Berkeley J. Dietvorst, Joseph P. Simmons, and Cade Massey. Overcoming algorithm aversion: People will use imperfect algorithms if they can (even slightly) modify them. *Management Science*, 2016.

Djellel Eddine Difallah, Michele Catasta, Gianluca Demartini, Panagiotis G. Ipeirotis, and Philippe Cudré-Mauroux. The dynamics of micro-task crowdsourcing: The case of Amazon MTurk. In *WWW*, 2015.

Dominic DiPalantino and Milan Vojnovic. Crowdsourcing and all-pay auctions. In *ACM EC*, 2009.

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. CoRR arXiv:1702.08608, 2017.

Mary T. Dzindolet, Linda G. Pierce, Hall P. Beck, and Lloyd A. Dawe. The perceived utility of human and automated aids in a visual detection task. *Human Factors*, 44(1): 79–94, 2002.

Robert C. Edgar and Serafim Batzoglou. Multiple sequence alignment. *Current opinion in structural biology*, 16(3):368–373, 2006.

Ju Fan, Guoliang Li, Beng Chin Ooi, Kian-lee Tan, and Jianhua Feng. icrowd: An adaptive crowdsourcing framework. In *SIGMOD*, 2015.

Oluwaseyi Feyisetan, Elena Simperl, Max Van Kleek, and Nigel Shadbolt. Improving paid microtasks through gamification and adaptive furtherance incentives. In *WWW*, 2015.

Urs Fischbacher and Franziska Föllmi-Heusi. Lies in disguise: An experimental study on cheating. *Journal of the European Economic Association*, 11(3):525–547, 2013.

Chao Gao, Yu Lu, and Dengyong Zhou. Exact exponent in optimal rates for crowdsourcing. In *ICML*, 2016.

Xi Alice Gao, Yoram Bachrach, Peter Key, and Thore Graepel. Quality expectation-variance tradeoffs in crowdsourcing contests. In *AAAI*, 2012.

Xi Alice Gao, Andrew Mao, Yiling Chen, and Ryan Prescott Adams. Trick or treat: Putting peer prediction to the test. In *ACM EC*, 2014.

Yashesh Gaur, Florian Metze, Yajie Miao, and Jeffrey P. Bigham. Using keyword spotting to help humans correct captioning faster. In *INTERSPEECH*, 2015.

Yashesh Gaur, Florian Metze, and Jeffrey P. Bigham. Manipulating word lattices to incorporate human corrections. In *INTERSPEECH*, 2016.

Timnit Gebru, Jonathan Krause, Jia Deng, and Li Fei-Fei. Scalable annotation of fine-grained objects without experts. In *CHI*, 2017.

Arpita Ghosh, Satyen Kale, and Preston McAfee. Who moderates the moderators? Crowdsourcing abuse detection in user-generated content. In *ACM EC*, 2011.

Daniel G. Goldstein, R. Preston McAfee, and Siddharth Suri. The cost of annoying ads. In *WWW*, 2013.

Daniel G. Goldstein, Siddharth Suri, R. Preston McAfee, Matthew Ekstrand-Abueg, and Fernando Diaz. The economic and cognitive costs of annoying display advertisements. *Journal of Marketing Research*, 51(6):742—752, 2014.

Ryan Gomes, Peter Welinder, Andreas Krause, and Pietro Perona. Crowdclustering. In *NIPS*, 2011.

Joseph K. Goodman and Gabriele Paolacci. Crowdsourcing consumer research. *Journal of Consumer Research*, 44(1):196–210, 2017.

Mary L. Gray, Siddharth Suri, Syed Shoaib Ali, and Deepti Kulkarni. The crowd is a collaborative network. In *CSCW*, 2016.

Neha Gupta, David Martin, Benjamin V. Hanrahan, and Jacki O'Neil. Turk-life in India. In *the Intnernational Conference on Supporting Groupwork*, 2014.

Juho Hamari, Jonna Koivisto, and Harri Sarsa. Does gamification work? – A literature review of empirical studies on gamification. In *Hawaii International Conference on System Sciences*, 2014.

Robin Hanson. Combinatorial information market design. *Information Systems Frontiers*, 5(1):105–119, 2003.

Christopher G. Harris. You're hired! An examination of crowdsourcing incentive models in human resource tasks. In *WSDM Workshop on Crwdsourcing for Search and Data Mining*, 2011.

Jeffrey Heer and Michael Bostock. Crowdsourcing graphical perception: Using Mechanical Turk to assess visualization design. In *CHI*, 2010.

Hannes Heikinheimo and Antti Ukkonen. The crowd-median algorithm. In *HCOMP*, 2013.

Chien-Ju Ho and Jennifer Wortman Vaughan. Online task assignment in crowdsourcing markets. In *AAAI*, 2012.

Chien-Ju Ho, Shahin Jabbari, and Jennifer Wortman Vaughan. Adaptive task assignment for crowdsourced classification. In *ICML*, 2013.

Chien-Ju Ho, Aleksandrs Slivkins, Siddharth Suri, and Jennifer Wortman Vaughan. Incentivizing high quality crowdwork. In *WWW*, 2015.

Chien-Ju Ho, Aleksandrs Slivkins, and Jennifer Wortman Vaughan. Adaptive contract design for crowdsourcing markets: Bandit algorithms for repeated principal-agent problems. *Journal of Artificial Intelligence Research*, 55:317–359, 2016.

John J. Horton, David Rand, and Richard Zeckhauser. The online laboratory: Conducting experiments in a real labor market. *Experimental Economics*, 14(3):399–425, 2011.

Yuening Hu, Jordan Boyd-Graber, Brianna Satinoff, and Alison Smith. Interactive topic modeling. *Machine Learning*, 95:423–469, 2014.

Lilly C. Irani and M. Six Silberman. Turkopticon: Interrupting worker invisibility in Amazon Mechanical Turk. In *CHI*, 2013.

Ayush Jain, Akash Das Sarma, Aditya Parameswaran, and Jennifer Widom. Understanding workers, developing effective tasks, and enhancing marketplace dynamics: A study of a large crowdsourcing marketplace. *Proceedings of the VLDB Endowment*, 10(7):829–840, 2017.

Jongbin Jung, Connor Concannon, Ravi Shro, Sharad Goel, and Daniel G. Goldstein. Simple rules for complex decisions. CoRR arXiv:1702.04690, 2017.

Radu Jurca and Boi Faltings. Mechanisms for making crowds truthful. *Journal of Artificial Intelligence Research*, 34:209–253, 2009.

Ece Kamar. Directions in hybrid intelligence: Complementing AI systems with human intelligence. Abstract for IJCAI Early Career Spotlight Track Talk, 2016.

Ece Kamar and Eric Horvitz. Incentives for truthful reporting in crowdsourcing (short paper). In *AAMAS*, 2012.

Vijay Kamble, David Marn, Nihar Shah, Abhay Parekh, and Kannan Ramachandran. Truth serums for massively crowdsourced evaluation tasks. CoRR arXiv:1507.07045, 2015.

Theofanis Karaletsos, Serge Belongie, and Gunnar Rätsch. Bayesian representation learning with oracle constraints. In *ICLR*, 2016.

David Karger, Sewoong Oh, and Devavrat Shah. Iterative learning for reliable crowdsourcing systems. In *NIPS*, 2011.

David Karger, Sewoong Oh, and Devavrat Shah. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, 62:1–24, 2014.

Gabriella Kazai. In search of quality in crowdsourcing for search engine evaluation. In *ECIR*, 2011.

Ashish Khetan and Sewoong Oh. Achieving budget-optimality with adaptive schemes in crowdsourcing. In *NIPS*, 2016.

Hyun-Chul Kim and Zoubin Ghahramani. Bayesian classifier combination. In *AISTATS*, 2012.

Joy Kim, Sarah Sterman, Allegra Argent Beal Cohen, and Michael S. Bernstein. Mechanical novel: Crowdsourcing complex work through reflection and revision. In *CSCW*, 2017.

Juho Kim, Haoqi Zhang, Paul André, Lydia B. Chilton, Wendy Mackay, Michel Beaudouin-Lafon, Robert C. Miller, and Steven P. Dow. Cobi: A community-informed conference scheduling tool. In *UIST*, 2013.

Aniket Kittur, Ed Chi, and Bongwon Suh. Crowdsourcing user studies with Mechanical Turk. In *CHI*, 2008.

Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E. Kraut. Crowdforge: Crowdsourcing complex work. In *UIST*, 2011.

Aniket Kittur, Jeffrey V. Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. The future of crowd work. In *CSCW*, 2013.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *ICML*, 2017.

Adriana Kovashka, Olga Russakovsky, Li Fei-Fei, and Kristen Grauman. Crowdsourcing in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 10(3):177–243, 2016.

Chinmay E. Kulkarni, Richard Socher, Michael S. Bernstein, and Scott R. Klemmer. Scaling short-answer grading by combining peer assessment with algorithmic scoring. In *ACM Conference on Learning@scale*, 2014.

Raja S. Kushalnagar, Walter S. Lasecki, and Jeffrey P. Bigham. A readability evaluation of real-time crowd captions in the classroom. In *ASSETS*, 2012.

Walter S. Lasecki and Jeffrey P. Bigham. Online quality control for real-time crowd captioning. In *ASSETS*, 2012.

Walter S. Lasecki, Christopher D. Miller, Adam Sadilek, Andrew Abumoussa, Donato Borrello, Raja Kushalnagar, and Jeffrey P. Bigham. Real-time captioning by groups of non-experts. In *UIST*, 2012.

Walter S. Lasecki, Christopher D. Miller, and Jeffrey P. Bigham. Warping time for more effective real-time crowdsourcing. In *CHI*, 2013.

Walter S. Lasecki, Christopher D. Miller, Iftekhar Naim, Raja Kushalnagar, Adam Sadilek, Daniel Gildea, and Jeffrey P. Bigham. Scribe: Deep integration of human and machine intelligence to caption speech in real-time. *Communications of the ACM*, 60(11), 2017.

Edith Law, Ming Yin, Joslin Goh, Kevin Chen, Michael Terry, and Krzysztof Z. Gajos. Curiosity killed the cat, but makes crowdwork better. In *CHI*, 2016.

John Le, Andy Edmonds, Vaughn Hester, and Lukas Biewald. Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution. In *SIGIR Workshop on Crowdsourcing for Search Evaluation*, 2010.

Tak Yeon Lee, Casey Dugan, Werner Geyer, Tristan Ratchford, Jamie Rasmussen, N. Sadat Shami, and Stela Lupushor. Experiments on motivational feedback for crowdsourced workers. In *ICWSM*, 2013.

Martin Lermen and Knut Reinert. The practical use of the A* algorithm for exact multiple sequence alignment. *Journal of Computational Biology*, 7(5):655–671, 2000.

Guoliang Li, Jiannan Wang, Yudian Zheng, and Michael J. Franklin. Crowdsourced data management: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 28(9): 2296–2319, 2016.

Qi Li, Yaliang Li, Jing Gao, Lu Su, Bo Zhao, Murat Demirbas, Wei Fan, and Jiawei Han. A confidence-aware approach for truth discovery on long-tail data. *Proceedings of the VLDB Endowment*, 8(4):425–436, 2014a.

Qi Li, Yaliang Li, Jing Gao, Bo Zhao, Wei Fan, and Jiawei Han. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In *SIGMOD*, 2014b.

Zachary C. Lipton. The mythos of model interpretability. CoRR arXiv:1606.03490, 2016.

Leib Litman, Jonathan Robinson, and Cheskie Rosenzweig. The relationship between motivation, monetary compensation, and data quality among US- and India-based workers on Mechanical Turk. *Behavioral Research Methods*, 47(2):519–528, 2014.

Qiang Liu, Jian Peng, and Alexander Ihler. Variational inference for crowdsourcing. In *NIPS*, 2012a.

Qiang Liu, Mark Steyvers, and Alexander Ihler. Scoring workers in crowdsourcing: How many control questions are enough? In *NIPS*, 2013.

Xuan Liu, Meiyu Lu, Beng Chin Ooi, Yanyan Shen, Sai Wu, and Meihui Zhang. CDAS: A crowdsourcing data analytics system. *Proceedings of the VLDB Endowment*, 5(10):1040–1051, 2012b.

Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 2(28):129–137, 1982.

Jennifer M. Logg. Theory of machine: When do people rely on algorithms? Harvard Business School NOM Unit Working Paper No. 17-086, 2017.

Yin Lou, Rich Caruana, and Johannes Gehrke. Intelligible models for classification and regression. In *KDD*, 2012.

Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. Accurate intelligible models with pairwise interactions. In *KDD*, 2013.

Fenglong Ma, Yaliang Li, Qi Li, Minghui Qiu, Jing Gao, Shi Zhi, Lu Su, Bo Zhao, Heng Ji, and Jiawei Han. FaitCrowd: Fine grained truth discovery for crowdsourced data aggregation. In *SIGMOD*, 2015.

Thomas W. Malone. What makes things fun to learn? Heuristics for designing instructional computer games. In *ACM SIGSMALL Symposium and the First SIGPC Symposium on Small Systems*, 1980.

Thomas W. Malone. Heuristics for designing enjoyable user interfaces: Lessons from computer games. In *CHI*, 1982.

Adam Marcus, Eugene Wu, David Karger, Samuel Madden, and Robert Miller. Human-powered sorts and joins. *Proceedings of the VLDB Endowment*, 5(1):13–24, 2011.

David Martin, Benjamin V. Hanrahan, Jacki O'Neill, and Neha Gupta. Being a Turker. In *CSCW*, 2014.

Winter Mason and Siddharth Suri. Conducting behavioral research on Amazon's Mechanical Turk. *Behavior Research Methods*, 44(1):1–23, 2012.

Winter Mason and Duncan J. Watts. Financial incentives and the "performance of crowds". In *HCOMP*, 2009.

Arya Mazumdar and Barna Saha. Clustering via crowdsourcing. CoRR arXiv:1604.01839, 2016.

Barbara Mellers, Eric Stone, Pavel Atanasov, Nick Rohrbaugh, S. Emlen Metz, Lyle Ungar, Michael M. Bishop, Michael Horowitz, Ed Merkle, and Philip Tetlock. The psychology of intelligence analysis: Drivers of prediction accuracy in world politics. *Journal of Experimental Psychology: Applied*, 21(2):1–14, 2015a.

Barbara Mellers, Eric Stone, Terry Murray, Angela Minster, Nick Rohrbaugh, Michael Bishop, Eva Chen, Joshua Baker, Yuan Hou, Michael Horowitz, Lyle Ungar, and Philip Tetlock. Identifying and cultivating superforecasters as a method of improving probabilistic predictions. *Perspectives on Psychological Science*, 10(2):267–281, 2015b.

Nolan Miller, Paul Resnick, and Richard Zeckhauser. Eliciting informative feedback: The peer-prediction method. *Management Science*, 51(9):1359–1373, 2005.

Roozbeh Mottaghi, Sanja Fidler, Jian Yao, Raquel Urtasun, and Devi Parikh. Analyzing semantic segmentation using hybrid human-machine CRFs. In *CVPR*, 2013.

Roozbeh Mottaghi, Sanja Fidler, Alan Yuille, Raquel Urtasun, and Devi Parikh. Human-machine CRFs for identifying bottlenecks in scene understanding. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

Iftekhar Naim, Daniel Gildea, Walter Lasecki, and Jeffrey P. Bigham. Text alignment for real-time crowd captioning. In *NAACL*, 2013.

David Newman, Edwin V. Bonilla, and Wray Buntine. Improving topic coherence with regularized topic models. In *NIPS*, 2011.

Besmira Nushi, Ece Kamar, Donald Kossmann, and Eric Horvitz. On human intellect and machine failures: Troubleshooting integrative machine learning systems. In *AAAI*, 2017.

David Oleson, Alexander Sorokin, Greg Laughlin, Vaughn Hester, John Le, and Lukas Biewald. Programmatic gold: Targeted and scalable quality assurance in crowdsourcing. In *HCOMP*, 2011.

Gabriele Paolacci, Jesse Chandler, and Panagiotis G. Ipeirotis. Running experiments on Amazon Mechanical Turk. *Judgment and Decision Making*, 5:411–419, 2010.

Devi Parikh and C. Lawrence Zitnick. Human-debugging of machines. In *Second NIPS Workshop on Computational Social Science and the Wisdom of Crowds*, 2011.

Kayur Patel, Naomi Bancroft, Steven M. Drucker, James Fogarty, Andrew J. Ko, and James A. Landay. Gestalt: Integrated support for implementation and analysis in machine learning processes. In *UIST*, 2010.

Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, 2012.

Genevieve Patterson, Chen Xu, Hang Su, and James Hays. The sun attribute database: Beyond categories for deeper scene understanding. *International Journal of Computer Vision*, 108(1–2):59–81, 2014.

Michael J. Paul. Interpretable machine learning: Lessons from topic modeling. In *CHI Workshop on Human-Centered Machine Learning*, 2016.

Michael J. Paul and Mark Dredze. Discovering health topics in social media using topic models. *PloS one*, 9(8):e103408, 2014.

Ellie Pavlick, Matt Post, Ann Irvine, Dmitry Kachaev, and Chris Callison-Burch. The language demographics of Amazon Mechanical Turk. *Transactions of the Association for Computational Linguistics*, 2:79–92, 2014.

Eyal Peera, Laura Brandimarteb, Sonam Samatc, and Alessandro Acquistic. Beyond the Turk: An empirical comparison of alternative platforms for crowdsourcing online behavioral research. *Journal of Experimental Social Psychology*, 70:153–163, 2017.

Matt Post, Chris Callison-Burch, and Miles Osborne. Constructing parallel corpora for six Indian languages via crowdsourcing. In *Seventh Workshop on Statistical Machine Translation*, 2012.

Forough Poursabzi-Sangdeh, Daniel G. Goldstein, Jake Hofman, Jennifer Wortman Vaughan, and Hanna Wallach. Manipulating and measuring model interpretability. CoRR arXiv:1802.07810, 2018.

Dražen Prelec. A Bayesian truth serum for subjective data. *Science*, 306(5695):462–466, 2004.

Dražen Prelec, H. Sebastian Seung, and John McCoy. A solution to the single-question crowd wisdom problem. *Nature*, 541:532–535, 2017.

Marianne Promberger and Jonathan Baron. Do patients trust computers? *Journal of Behavioral Decision Making*, 19:455–468, 2006.

Goran Radanovic and Boi Faltings. A robust Bayesian truth serum for non-binary signals. In *AAAI*, 2013.

Goran Radanovic, Boi Faltings, and Radu Jurca. Incentives for effort in crowdsourcing using the peer truth serum. *ACM Transactions on Intelligent Systems and Technology*, 7(4):1–28, 2016.

Srinivas Rao and Amanda Michel. ProPublica's guide to Mechanical Turk. ProPublica article accessed at `https://www.propublica.org/article/propublicas-guide-to-mechanical-turk`, 2016.

Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11: 1297–1322, 2010.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should I trust you?: Explaining the predictions of any classifier. In *KDD*, 2016.

Jakob Rogstadius, Vassilis Kostakos, Aniket Kittur, Boris Smus, Jim Laredo, and Maja Vukovic. An assessment of intrinsic and extrinsic motivation on task performance in crowdsourcing markets. In *ICWSM*, 2011.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015a.

Olga Russakovsky, Li-Jia Li, and Li Fei-Fei. Best of both worlds: Human-machine collaboration for object annotation. In *CVPR*, 2015b.

Niloufar Salehi, Lilly Irani, Michael Bernstein, Ali Alkhatib, Eva Ogbe, Kristy Milland, and Clickhappier. We are dynamo: Overcoming stalling and friction in collective action for crowd workers. In *CHI*, 2015.

Niloufar Salehi, Jaime Teevan, Shamsi Iqbal, and Ece Kamar. Communicating context to the crowd for complex writing tasks. In *CSCW*, 2017.

Paul J. H. Schoemaker and Philip E. Tetlock. Superforecasting: How to upgrade your company's judgment. *Harvard Business Review*, 94:72–78, 2016.

Devavrat Shah and Christina E. Lee. Reducing crowdsourcing to graphon estimation, statistically. In *AISTATS*, 2018.

Nihar Shah, Dengyong Zhou, and Yuval Peres. Approval voting and incentives in crowdsourcing. In *ICML*, 2015.

Nihar B. Shah and Dengyong Zhou. Double or nothing: Multiplicative incentive mechanisms for crowdsourcing. *Journal of Machine Learning Research*, 17(165):1–52, 2016a.

Nihar B. Shah and Dengyong Zhou. No oops, you won't do it again: Mechanisms for self-correction in crowdsourcing. In *ICML*, 2016b.

Aaron D. Shaw, John J. Horton, and Daniel L. Chen. Designing incentives for inexpert human raters. In *CSCW*, 2011.

Victor Sheng, Foster Provost, and Panagiotis Ipeirotis. Get another label? Improving data quality using multiple, noisy labelers. In *KDD*, 2008.

Victor Shnayder, Arpit Agarwal, Rafael M. Frongillo, and David C. Parkes. Informed truthfulness in multi-task peer prediction. In *ACM EC*, 2016.

M. Six Silberman, Lilly Irani, and Joel Ross. Ethics and tactics of professional crowdwork. *XRDS: Crossroads, The ACM Magazine for Students*, 17(2):39–43, 2010.

M. Six Silberman, Kristy Milland, Rochelle LaPlant, Joel Ross, and Lilly Irani. Stop citing Ross et al. 2010, "Who are the crowdworkers?". Accessed at `https://medium.com/@silberman/stop-citing-ross-et-al-2010-who-are-the-crowdworkers-b3b9b1e8d300`, 2015.

Patrice Y. Simard, Saleema Amershi, David M. Chickering, Alicia Edelman Pelton, Soroush Ghorashi, Christopher Meek, Gonzalo Ramos, Jina Suh, Johan Verwey, Mo Wang, and John Wernsing. Machine teaching: A new paradigm for building machine learning systems. CoRR arXiv:1707.06742, 2017.

Daniel J. Simons and Christopher F. Chabris. Common (mis)beliefs about memory: A replication and comparison of telephone and Mechanical Turk survey methods. *PLoS ONE*, 7(12), 2012.

Rashmi R. Sinha and Kirsten Swearingen. Comparing recommendations made by online systems and friends. In *DELOS workshop: Personalisation and recommender systems in digital libraries*, 2001.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks. In *EMNLP*, 2008.

Alexander Sorokin and David Forsyth. Utility data annotation with Amazon Mechanical Turk. In *CVPRW*, June 2008.

Neil Stewart, Christoph Ungemach, Adam J. L. Harris, Daniel M. Bartels, Ben R. Newell, Gabriele Paolacci, and Jesse Chandler. The average laboratory samples a population of 7,300 Amazon Mechanical Turk workers. *Judgment and Decision Making*, September 2015.

Hao Su, Jia Deng, and Li Fei-Fei. Crowdsourcing annotations for visual object detection. In *HCOMP*, 2012.

Siddharth Suri and Duncan J. Watts. Cooperation and contagion in web-based, networked public goods experiments. *PLoS ONE*, 6(3), 2011.

Siddharth Suri, Daniel Goldstein, and Winter Mason. Honesty in an online labor market. In *HCOMP*, 2011.

Latanya Sweeney. Discrimination in online ad delivery. *Communications of the ACM*, 56 (5):44–54, 2013.

Omer Tamuz, Ce Liu, Serge Belongie, Ohad Shamir, and Adam Kalai. Adaptively learning the crowd kernel. In *ICML*, 2011.

Jaime Teevan, Shamsi Iqbal, and Curtis von Veh. Supporting collaborative writing with microtasks. In *CHI*, 2016.

Philip E. Tetlock, Barbara A. Mellers, and J. Peter Scoblic. Bringing probability judgments into policy debates via forecasting tournaments. *Science*, 355(6324):481–483, 2017.

Tian Tian and Jun Zhu. Max-margin majority voting for learning from crowds. In *NIPS*, 2015.

Lyle H. Ungar, Barbara A. Mellers, Ville Satopää, Philip Tetlock, and Jon Baron. The good judgment project: A large scale test of different methods of combining expert predictions. In *AAAI Fall Symposium: Machine Aggregation of Human Judgment*, 2012.

Blase Ur, Jonathan Bees, Sean M. Segreti, Lujo Bauer, and Lorrie Faith Cranor Nicolas Christin. Do users' perceptions of password security match reality? In *CHI*, 2016.

Berk Ustun and Cynthia Rudin. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning Journal*, 102(3):349–391, 2016.

Donna Vakharia and Matthew Lease. Beyond Mechanical Turk: An analysis of paid crowd work platforms. In *Proceedings of the iConference*, 2015.

Matteo Venanzi, John Guiver, Gabriella Kazai, Pushmeet Kohli, and Milad Shokouhi. Community-based Bayesian aggregation models for crowdsourcing. In *WWW*, 2014.

Norases Vesdapunt, Kedar Bellare, and Nilesh Dalvi. Crowdsourcing algorithms for entity resolution. *Proceedings of the VLDB Endowment*, 7(12):1071–1082, 2014.

Sudheendra Vijayanarasimhan and Kristen Grauman. What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In *CVPR*, 2009.

Ramya Korlakai Vinayak and Babak Hassibi. Crowdsourced clustering: Querying edges vs triangles. In *NIPS*, 2016.

Ramya Korlakai Vinayak, Samet Oymak, and Babak Hassibi. Graph clustering with missing data: Convex algorithms and analysis. In *NIPS*, 2014.

Luis von Ahn and Laura Dabbish. Labeling images with a computer game. In *CHI*, 2004.

Luis von Ahn and Laura Dabbish. General techniques for designing games with a purpose. *Communications of the ACM*, 51(8):58–67, August 2008.

Luis von Ahn, Manuel Blum, Nicholas Hopper, and John Langford. CAPTCHA: Using hard AI problems for security. In *EUROCRYPT*, 2003.

Luis von Ahn, Manuel Blum, and John Langford. Telling humans and computers apart automatically. *Communications of the ACM*, pages 56–60, 2004.

Luis von Ahn, Mihir Kedia, and Manuel Blum. Verbosity: A game for collecting common-sense knowledge. In *CHI Notes*, 2006.

Luis von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. reCAPTCHA: Human-based character recognition via web security measures. *Science*, 321(5895):1465–1468, 2008.

Bo Waggoner and Yiling Chen. Output agreement mechanisms and common knowledge. In *HCOMP*, 2014.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Catherine Wah, Grant Van Horn, Steve Branson, Subhransu Maji, Pietro Perona, and Serge Belongie. Similarity comparisons for interactive fine-grained categorization. In *CVPR*, 2014.

Gang Wang, Christo Wilson, Xiaohan Zhao, Yibo Zhu, Manish Mohanlal, Haitao Zheng, and Ben Y. Zhao. Serf and turf: Crowdturfing for fun and profit. In *WWW*, 2012a.

Jiannan Wang, Tim Kraska, Michael J. Franklin, and Jianhua Feng. CrowdER: Crowdsourcing entity resolution. *Proceedings of the VLDB Endowment*, 5(11):1483–1494, 2012b.

Fabian L. Wauthier, Nebojsa Jojic, and Michael I. Jordan. Active spectral clustering via iterative uncertainty reduction. In *KDD*, 2012.

Peter Welinder, Steve Branson, Serge Belongie, and Perona Pietro. The multidimensional wisdom of crowds. In *NIPS*, 2010.

Kathryn Sharpe Wessling, Joel Huber, and Oded Netzer. Character misrepresentation by Amazon Turk workers: Assessment and solutions. *Journal of Consumer Research*, 44(1): 211–230, 2017.

Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R. Movellan, and Paul L. Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*, 2009.

Michael Wilber, Sam Kwak, and Serge Belongie. Cost-effective hits for relative similarity comparisons. In *HCOMP*, 2014.

Vanessa Williamson. On the ethics of crowdsourced research. *Political Science & Politics*, 49(1):77–81, 2016.

Jens Witkowski and David C. Parkes. A robust Bayesian truth serum for small populations. 2012.

Justin Wolfers and Eric Zitzewitz. Prediction markets. *The Journal of Economic Perspectives*, 18(2):107–126, 2004.

James R. Wright, Chris Thornton, and Kevin Leyton-Brown. Mechanical TA: Partially automated high-stakes peer grading. In *ACM Technical Symposium on Computer Science Education*, 2015.

Jian Yao, Sanja Fidler, and Raquel Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *CVPR*, 2012.

Teng Ye, Sangseok You, and Lionel P. Robert Jr. When does more money work? Examining the role of perceived fairness in pay on the performance quality of crowdworkers. In *ICWSM*, 2017.

Michael Yeomans, Anuj K. Shah, Sendhil Mullainathan, and Jon Kleinberg. Making sense of recommendations. *Management Science*, 2017.

Jinfeng Yi, Rong Jin, Anil K. Jain, and Shaili Jain. Crowdclustering with sparse pairwise labels: A matrix completion approach. In *HCOMP*, 2012a.

Jinfeng Yi, Rong Jin, Anil K. Jain, Shaili Jain, and Tianbao Yang. Semi crowdsourced clustering: Generalizing crowd labeling by robust distance metric learning. In *NIPS*, 2012b.

Ming Yin, Yiling Chen, and Yu-An Sun. The effects of performance-contingent financial incentives in online labor markets. In *AAAI*, 2013.

Ming Yin, Yiling Chen, and Yu-An Sun. Monetary interventions in crowdsourcing task switching. In *HCOMP*, 2014.

Ming Yin, Mary L. Gray, Siddharth Suri, and Jennifer Wortman Vaughan. The communication network within the crowd. In *WWW*, 2016.

Dong Yu and Li Deng. *Automatic speech recognition: A deep learning approach*. Springer, 2014.

Omar Zaidan and Chris Callison-Burch. Crowdsourcing translation: Professional quality from non-professionals. In *ACL*, 2011.

Rabih Zbib, Erika Malchiodi, Jacob Devlin, David Stallard, Spyros Matsoukas, Richard Schwartz, John Makhoul, Omar F. Zaidan, and Chris Callison-Burch. Machine translation of Arabic dialects. In *NAACL*, 2012.

Haoqi Zhang, Edith Law, Krzysztof Gajos, Eric Horvitz, Rob Miller, and David Parkes. Human computation tasks with global constraints. In *CHI*, 2012.

Jing Zhang, Xindong Wu, and Victor S. Sheng. Learning from crowdsourced labeled data: A survey. *Artificial Intelligence Review*, 46(4):543–576, 2016a.

Yuchen Zhang, Xi Chen, Dengyong Zhou, and Michael I. Jordan. Spectral methods meet EM: A provably optimal algorithm for crowdsourcing. *Journal of Machine Learning Research*, 17(102):1–44, 2016b.

Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. Truth inference in crowdsourcing: Is the problem solved? *Proceedings of the VLDB Endowment*, 10(5): 541–552, 2017.

Dengyong Zhou, Sumit Basu, Yi Mao, and John Platt. Learning from the wisdom of crowds by minimax entropy. In *NIPS*, 2012.

Dengyong Zhou, Qiang Liu, John C. Platt, and Christopher Meek. Aggregating ordinal labels from crowds by minimax conditional entropy. In *ICML*, 2014.

Journal of Machine Learning Research 18 (2018) 1-23 Submitted 9/17; Revised 5/18; Published 6/18

Enhancing Identification of Causal Effects by Pruning

Santtu Tikka SANTTU.TIKKA@JYU.FI
Juha Karvanen JUHA.T.KARVANEN@JYU.FI
Department of Mathematics and Statistics
P.O.Box 35 (MaD) FI-40014 University of Jyvaskyla, Finland

Editor: Peter Spirtes

Abstract

Causal models communicate our assumptions about causes and effects in real-world phenomena. Often the interest lies in the identification of the effect of an action which means deriving an expression from the observed probability distribution for the interventional distribution resulting from the action. In many cases an identifiability algorithm may return a complicated expression that contains variables that are in fact unnecessary. In practice this can lead to additional computational burden and increased bias or inefficiency of estimates when dealing with measurement error or missing data. We present graphical criteria to detect variables which are redundant in identifying causal effects. We also provide an improved version of a well-known identifiability algorithm that implements these criteria.

Keywords: causal inference, identifiability, causal model, pruning, algorithm

1. Introduction

A formal framework for causal inference is provided by the probabilistic causal model (Pearl, 2009) that encodes our knowledge of the variables of interest and their mutual relationships. In observational studies experimentation is not available, but through the causal model framework we can still symbolically intervene on variables, forcing them to take certain values as if an experiment had taken place. The question is whether we can make inferences about the effect of the intervention in the post-intervention model using only the observed probability distribution of the variables in the model before the intervention took place. This question is formally defined as identifiability of causal effects, and it has received considerable attention in literature, including a number of algorithmic solutions (Huang and Valtorta, 2006; Shpitser and Pearl, 2006; Tian and Pearl, 2002).

A causal model can be associated with a directed acyclic graph (DAG) that represents the functional relationships of the variables included in the model. The graphical representation provides us with the concept of d-separation (Geiger et al., 1990), that can be used to infer conditional independences between variables from the graph. If the distribution of the variables implies no conditional independence statements other than those already encoded in the graph, we say that the distribution is faithful (Spirtes et al., 2000).

The use of d-separation in the post-intervention model is the basis of do-calculus (Pearl, 1995), which consists of a set of inference rules for manipulating interventional distributions. The purpose of do-calculus is to derive formulas for causal effects and other causal queries, and it has been shown to be complete with respect to the identifiability of causal effects (Huang and Valtorta, 2006; Shpitser and Pearl, 2006). The derived formulas provide recipes for estimating the causal effects from observational data.

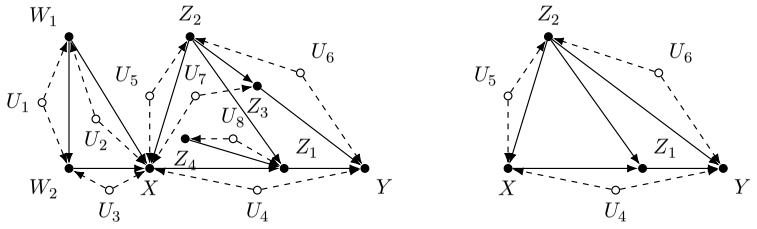
When computing causal effect formulas, we often apply an identifiability algorithm, such as the ID algorithm by Shpitser and Pearl (2006). Criteria for identifiability such as the back-door criterion and front-door criterion are available for manual derivations (Pearl, 2009) but the ID algorithm is more general and thus more suitable for automated processing. The ID algorithm splits the original problem into smaller subproblems which are then solved and aggregated as the final expression for the causal effect.

Complicated expressions are likely to arise in situations where we have included variables in our model that do not provide further benefit for the identification of the causal effect of interest. It is often the case that these variables nonetheless appear in the resulting formula, and deriving a simpler expression with the variable eliminated can be non-trivial. It is hard to specify what makes one expression simpler than another, but we can consider a number of criteria to evaluate simplicity. For example, we can compare the number of sums and fractions and the number of variables present in the expression.

In this paper we propose a number of graphical criteria to infer which variables in our causal model are in fact not necessary for identification. These criteria allow us to prune the graph, which in practice means removing specific vertices and considering identification in a latent projection. A significantly simpler expression can be obtained by pruning alone, but we may also combine pruning with simplification procedures that operate symbolically on the interventional distribution as presented in (Tikka and Karvanen, 2017b). Applying these methods in conjunction often provides additional benefits.

We present an identifiability algorithm that is able to recognize and eliminate unnecessary variables from the graph based on our criteria resulting in a simpler expression. When a large number of graphs and identifiability queries are processed, evaluating simpler expressions has apparent computational benefits. First, it is more efficient to evaluate a simpler expression repeatedly especially when some variables have been completely removed which further reduces the complexity of the task. Second, in practical applications that involve real-world data, variables often contain missing data or are affected by bias. Obtaining expression that do not involve such variables can be of great benefit in estimation. Third, a simpler expression is easier to communicate.

An introductory example motivates the use of the improved algorithm. We are interested in the causal effect of X on Y in graph G of Figure 1(a). Here, open circles denote unobserved variables. A more in-depth overview of graph theoretic concepts used in this paper is provided in Section 2. The causal effect is identifiable and the output of the ID algorithm is



(a) Graph G. (b) Graph G after pruning.

Figure 1: Graph G before and after pruning for the introductory example.

$$\sum_{z_2,z_4,z_3,z_1}\left(\sum_{w_1,w_2,x'}P(y|w_1,z_2,z_4,w_2,z_3,x',z_1)P(x'|w_1,z_2,z_4,w_2,z_3)\times P(z_3|w_1,z_2,z_4,w_2)P(w_2|w_1,z_2,z_4)P(z_2|w_1)P(w_1)\right)\Bigg/$$
$$\left(\sum_{w_1,w_2,x',y'}P(y'|w_1,z_2,z_4,w_2,z_3,x',z_1)P(x'|w_1,z_2,z_4,w_2,z_3)\times P(z_3|w_1,z_2,z_4,w_2)P(w_2|w_1,z_2,z_4)P(z_2|w_1)P(w_1)\right)\times$$
$$\left(\sum_{w_1,w_2,z_3',x',y'}P(y'|w_1,z_2,z_4,w_2,z_3',x',z_1)P(x'|w_1,z_2,z_4,w_2,z_3')\times P(z_3'|w_1,z_2,z_4,w_2)P(w_2|w_1,z_2,z_4)P(z_2|w_1)P(w_1)\right)\times$$
$$P(z_1|w_1,z_2,z_4,w_2,x)P(z_3|z_2)P(z_4).$$

This expression is very cumbersome and complicated. However, it turns out that a simpler expression exists for the causal effect. By exploiting the structure of the graph and using standard probability calculus the following expression can be obtained

$$\sum_{z_2,z_1}\left(\sum_{x'}P(y|z_2,z_1,x')P(x'|z_2)P(z_2)\right)P(z_1|z_2,x).$$

This expression is simpler in every regard compared to the original output. It contains fewer terms and no fractions. Also, we have completely removed the variables w_1, w_2 and z_4 from the expression. It can be shown that identifying the causal effect in the original graph is

equivalent to identifying it in the graph depicted in Figure 1(b). By running our improved algorithm we are able to prune the original graph and obtain this simpler expression directly. The algorithm works recursively and the pruning is carried out at each stage of the recursion. The recursive pruning provides significant benefits over pruning as a pre-processing step as demonstrated later.

The paper is structured as follows. In Section 2 we review crucial definitions and concepts related to graph theory and causal models. In Section 3 we focus on semi-Markovian causal models and present the original formulation of the ID algorithm. Our main results are presented in Section 4 and they are implemented into an improved identifiability algorithm in Section 5. Examples on the benefits of recursive pruning are provided in Section 6. Section 7 concludes with a discussion.

2. Definitions

We assume the reader to be familiar with a number of graph theoretic concepts and refer them to works such as (Koller and Friedman, 2009). We use capital letters to denote vertices and the respective variables, and small letters to denote their values. Bold letters are used to denote sets. A directed graph with a vertex set $\mathbf{V}$ and an edge set $\mathbf{E}$ is denoted by $\langle \mathbf{V}, \mathbf{E}\rangle$. For a graph $G = \langle \mathbf{V}, \mathbf{E}\rangle$ and a set of vertices $\mathbf{W} \subseteq \mathbf{V}$ the sets $\mathrm{Pa}(\mathbf{W})_G, \mathrm{Ch}(\mathbf{W})_G, \mathrm{An}(\mathbf{W})_G$ and $\mathrm{De}(\mathbf{W})_G$ denote a set that contains $\mathbf{W}$ in addition to its parents, children, ancestors and descendants in G, respectively. We also define the set $\mathrm{Co}(\mathbf{W})_G$ to denote the set of vertices that are connected to $\mathbf{W}$ in G via paths where the directionality of the edges is ignored, including $\mathbf{W}$. The root set of a graph G is the set of vertices without any descendants $\{X \in V \mid \mathrm{De}(X)_G \setminus \{X\} = \emptyset\}$, where $\setminus$ denotes the set difference. A subgraph of a graph $G = \langle \mathbf{V}, \mathbf{E}\rangle$ induced by a set of vertices $\mathbf{W} \subset \mathbf{V}$ is denoted by $G[\mathbf{W}]$. This subgraph retains all edges $V \to W$ of G such that $V, W \in \mathbf{W}$. The graph obtained from G by removing all incoming edges of $\mathbf{X}$ and all outgoing edges of $\mathbf{Z}$ is written as $G_{\overline{\mathbf{X}},\underline{\mathbf{Z}}}$. To facilitate analysis of causal effects we must first define the probabilistic causal model (Pearl, 2009).

Definition 1 (Probabilistic Causal Model) *A* probabilistic causal model *is a quadruple*

$$M = \langle \mathbf{U}, \mathbf{V}, \mathbf{F}, P(\mathbf{u})\rangle,$$

where

1. $\mathbf{U}$ *is a set of unobserved (exogenous) variables that are determined by factors outside the model.*
2. $\mathbf{V}$ *is a set* $\{V_1, V_2, \ldots, V_n\}$ *of observed (endogenous) variables that are determined by variables in* $\mathbf{U} \cup \mathbf{V}$.
3. $\mathbf{F}$ *is a set of functions* $\{f_{V_1}, f_{V_2}, \ldots, f_{V_n}\}$ *such that each* f_{V_i} *is a mapping from (the respective domains of)* $\mathbf{U} \cup (\mathbf{V} \setminus \{V_i\})$ *to* V_i*, and such that the entire set* $\mathbf{F}$ *forms a mapping from* $\mathbf{U}$ *to* $\mathbf{V}$.
4. $P(\mathbf{u})$ *is a joint probability distribution of the variables in the set* $\mathbf{U}$.

Each causal model induces a causal diagram which is a directed graph that provides a graphical means to convey our assumptions of the causal mechanisms involved. The induced graph is constructed by adding a vertex for each variable in $\mathbf{U} \cup \mathbf{V}$ and a directed edge from $V_i \in \mathbf{U} \cup \mathbf{V}$ into $V_j \in \mathbf{V}$ whenever f_{V_j} is defined in terms of V_i.

Causal inference often focuses on a sub-class of models that satisfy additional assumptions: each $U \in \mathbf{U}$ appears in at most two functions of $\mathbf{F}$, the variables in $\mathbf{U}$ are mutually independent and the induced graph of the model is acyclic. Models that satisfy these additional assumptions are called *semi-Markovian causal models*. A graph associated with a semi-Markovian model is called a *semi-Markovian graph* (SMG). In SMGs every $U \in \mathbf{U}$ has at most two children. When semi-Markovian models are considered it is common not to depict background variables in the induced graph explicitly. Unobserved variables with exactly two children are not denoted as $V_i \leftarrow U \rightarrow V_j$ but as a bidirected edge $V_i \leftrightarrow V_j$ instead. Furthermore, unobserved variables with only one or no children are omitted entirely. We also adopt these abbreviations. For SMGs the sets $\mathrm{Pa}(\cdot)_G, \mathrm{Ch}(\cdot)_G, \mathrm{An}(\cdot)_G, \mathrm{De}(\cdot)_G$ and $\mathrm{Co}(\cdot)_G$ contain only observed vertices. Additionally, a subgraph $G[\mathbf{W}]$ of an SMG G will also retain any bidirected edges between vertices in $\mathbf{W}$.

Any DAG can be associated with an SMG by constructing its *latent projection* (Verma, 1993).

Definition 2 (latent projection) *Let $G = \langle \mathbf{V} \cup \mathbf{L}, \mathbf{E} \rangle$ be a DAG such that the vertices in $\mathbf{V}$ are observed and the vertices in $\mathbf{L}$ are latent. The* latent projection *$L(G, \mathbf{V})$ is a DAG $\langle \mathbf{V}, \mathbf{E}_L \rangle$, where for every pair of distinct vertices $Z, W \in \mathbf{V}$ it holds that:*

1. *$L(G, \mathbf{V})$ contains an edge $Z \rightarrow W$ if there exists a directed path $Z \rightarrow \cdots \rightarrow W$ in G on which every vertex except Z and W is in $\mathbf{L}$.*
2. *$L(G, \mathbf{V})$ contains an edge $Z \leftrightarrow W$ if there exists a path from Z to W in G that does not contain the pattern $Z \rightarrow M \leftarrow W$ (a collider) and on which every vertex except Z and W is in $\mathbf{L}$ and the first edge has an arrowhead pointing into W and the last edge has an arrowhead pointing into Z.*

From the construction it is easy to see that a latent projection is in fact an SMG. The induced graph of a probabilistic causal model can also be used to derive conditional independences among the variables in the model using a concept known as d-separation. We provide a definition for d-separation (Shpitser and Pearl, 2008) which takes into account the presence of bidirected edges and is thus suitable for SMGs.

Definition 3 (d-separation) *A path P in an SMG G is said to be d-separated by a set $\mathbf{Z}$ if and only if either*

1. *P contains one of the following three patterns of edges: $I \rightarrow M \rightarrow J$, $I \leftrightarrow M \rightarrow J$ or $I \leftarrow M \rightarrow J$, such that $M \in \mathbf{Z}$, or*
2. *P contains one of the following three patterns of edges: $I \rightarrow M \leftarrow J$, $I \leftrightarrow M \leftarrow J$, $I \leftrightarrow M \leftrightarrow J$, such that $De(M)_G \cap \mathbf{Z} = \emptyset$.*

Disjoint sets $\mathbf{X}$ and $\mathbf{Y}$ are said to be d-separated by $\mathbf{Z}$ in G if every path from $\mathbf{X}$ to $\mathbf{Y}$ is d-separated by $\mathbf{Z}$ in G.

Whenever we can decompose the joint distribution of the observed variables $\mathbf{V}$ and the unobserved variables $\mathbf{U}$ as $P(\mathbf{v}, \mathbf{u}) = \prod_{W \in \mathbf{V} \cup \mathbf{U}} P(w|\mathrm{Pa}^*(w)_G)$, where $\mathrm{Pa}^*(\cdot)$ also contains the unobserved parents but not the argument itself, we say that G is an I-map of $P(\mathbf{v}, \mathbf{u})$ (Pearl, 2009). If sets $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in G, then $\mathbf{X}$ is independent of $\mathbf{Y}$ given $\mathbf{Z}$ in every P for which G is an I-map (Pearl, 1988). We use the notation of (Dawid, 1979) to denote this d-separation and conditional independence statement as $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z})_G$. It is clear that the graph induced by any semi-Markovian causal model is an I-map for the joint distribution $P(\mathbf{v}, \mathbf{u})$ induced by the model.

Our interest lies in the effects of actions imposing changes to the model. An action that forces $\mathbf{X}$ to take a specific value $\mathbf{x}$ is called an *intervention* and it is denoted by $\mathrm{do}(\mathbf{x})$ (Pearl, 2009). An intervention $\mathrm{do}(\mathbf{x})$ on a model M creates a new sub-model, denoted by $M_{\mathbf{x}}$, where the functions in $\mathbf{F}$ that determine the value of $\mathbf{X}$ have been replaced with constant functions. The *interventional distribution* of a set of variables $\mathbf{Y}$ in the model $M_{\mathbf{x}}$ is denoted by $P_{\mathbf{x}}(\mathbf{y})$. This distribution is also known as the *causal effect* of $\mathbf{X}$ on $\mathbf{Y}$.

Multiple causal models can share the same graph, and thus the same sub-model resulting from an intervention. The question is, are our assumptions encoded in the causal model sufficient to uniquely specify an interventional distribution of interest. This notion is captured by the following definition (Shpitser and Pearl, 2006).

Definition 4 (identifiability) *Let $G = \langle \mathbf{V}, \mathbf{E} \rangle$ be an SMG and let $\mathbf{X}$ and $\mathbf{Y}$ be disjoint sets of variables such that $\mathbf{X}, \mathbf{Y} \subset \mathbf{V}$. The causal effect of $\mathbf{X}$ on $\mathbf{Y}$ is said to be* identifiable *from P in G if $P_{\mathbf{x}}(\mathbf{y})$ is uniquely computable from $P(\mathbf{V})$ in any causal model that induces G.*

In order to show the identifiability of a given effect we have to express the interventional distribution in terms of observed probabilities only. The link between observed probabilities and interventional distributions is provided by three inference rules known as *do-calculus* (Pearl, 1995):

1. Insertion and deletion of observations:

$$P_{\mathbf{x}}(\mathbf{y}|\mathbf{z}, \mathbf{w}) = P_{\mathbf{x}}(\mathbf{y}|\mathbf{w}), \text{ if } (\mathbf{Y} \perp\!\!\!\perp \mathbf{Z}|\mathbf{X}, \mathbf{W})_{G_{\overline{\mathbf{X}}}}.$$

2. Exchanging actions and observations:

$$P_{\mathbf{x},\mathbf{z}}(\mathbf{y}|\mathbf{w}) = P_{\mathbf{x}}(\mathbf{y}|\mathbf{z}, \mathbf{w}), \text{ if } (\mathbf{Y} \perp\!\!\!\perp \mathbf{Z}|\mathbf{X}, \mathbf{W})_{G_{\overline{\mathbf{X}},\underline{\mathbf{Z}}}}.$$

3. Insertion and deletion of actions:

$$P_{\mathbf{x},\mathbf{z}}(\mathbf{y}|\mathbf{w}) = P_{\mathbf{x}}(\mathbf{y}|\mathbf{w}), \text{ if } (\mathbf{Y} \perp\!\!\!\perp \mathbf{Z}|\mathbf{X}, \mathbf{W})_{G_{\overline{\mathbf{X}},Z(\overline{\mathbf{W}})}},$$

   where $Z(\mathbf{W}) = \mathbf{Z} \setminus \mathrm{An}(\mathbf{W})_{G_{\overline{\mathbf{X}}}}$.

Completeness of do-calculus was established independently by Huang and Valtorta (2006) and Shpitser and Pearl (2006). In this paper we focus on the solution provided by Shpitser and Pearl (2006). They constructed an identifiability algorithm called ID, which in essence applies the rules of do-calculus and breaks the problem into smaller sub-problems repeatedly.

3. ID Algorithm

In order to present the ID algorithm, we first need some additional definitions that are used to construct the graphical criterion for non-identifiability (Shpitser and Pearl, 2006).

Definition 5 (C-component) *Let G be an SMG and let $C \subseteq G$. If every pair of vertices in C is connected by a bidirected path, that is a path consisting entirely of bidirected edges, then C is a* C-component *(confounded component). Furthermore, C is a* maximal C-component *if C contains every vertex connected to C via bidirected paths in G and C is an induced subgraph of G.*

No restrictions are imposed on the directed edges of a C-component. The same is not true for the maximal C-components (also known as districts) of an SMG G, which are assumed to be induced subgraphs of G. This requirement guarantees the uniqueness of the maximal C-components.

Maximal C-components are an important tool for identifying causal effects. The set of maximal C-components of a semi-Markovian graph G is denoted by $C(G)$. A result in (Tian, 2002) states that if $C = \langle \mathbf{C}, \mathbf{E} \rangle$ is a maximal C-component and $C \subset G$ then the causal effect $P_{\mathbf{v}\setminus\mathbf{c}}(\mathbf{c})$ is identifiable from P in G. A distribution P of a semi-Markovian model also factorizes with respect to the maximal C-components of the induced graph G such that $P(\mathbf{v}) = \prod_{\langle \mathbf{C},\mathbf{E}\rangle \in C(G)} P_{\mathbf{v}\setminus\mathbf{c}}(\mathbf{c})$ (Shpitser and Pearl, 2006). It is precisely this factorization that the ID algorithm takes advantage of. A specific type of C-component is used to characterize problematic structures for identifiability.

Definition 6 (C-forest) *Let G be an SMG and let $\mathbf{Y}$ be the root set of G. If G is a C-component and all observed vertices have at most one child, then G is a* $\mathbf{Y}-$rooted C-forest.

The complete criterion for non-identifiability uses a structure formed by two C-forests:

Definition 7 (hedge) *Let $\mathbf{X}, \mathbf{Y} \subset \mathbf{V}$ be disjoint sets of variables and let G be an SMG. Let $F = \langle \mathbf{V}_F, \mathbf{E}_F \rangle$ and $F' = \langle \mathbf{V}_{F'}, \mathbf{E}_{F'} \rangle$ be $\mathbf{R}$-rooted C-forests in G such that $\mathbf{V}_F \cap \mathbf{X} \neq \emptyset$, $\mathbf{V}_{F'} \cap \mathbf{X} = \emptyset$, $F' \subseteq F$, and $\mathbf{R} \subseteq An(\mathbf{Y})_{G_{\overline{\mathbf{X}}}}$. Then F and F' form a hedge for $P_{\mathbf{x}}(\mathbf{y})$ in G.*

Intuitively hedges are a difficult concept. Whenever a hedge is present, there exists two causal models with the same probability distribution over $\mathbf{V}$ but their interventional distributions do not agree. Observational data can not be used to estimate causal effects in this scenario. We are now ready to present the ID algorithm.

Shpitser and Pearl (2006) showed that whenever Algorithm 1 returns an expression for a causal effect, it is correct. Additionally whenever line 5 is triggered there exists a hedge for the causal effect currently being identified. This result establishes the completeness of the algorithm and also the completeness of do-calculus, since the soundness of each line of the algorithm can be shown with do-calculus and standard probability calculus alone.

Algorithm 1 The causal effect of intervention $do(\mathbf{X} = \mathbf{x})$ on $\mathbf{Y}$ (ID).

INPUT: Value assignments $\mathbf{x}$ and $\mathbf{y}$, joint distribution $P(\mathbf{v})$ and an SMG $G = \langle \mathbf{V}, \mathbf{E} \rangle$. G is an I-map of P.

OUTPUT: Expression for $P_{\mathbf{x}}(\mathbf{y})$ in terms of $P(\mathbf{v})$ or **FAIL**(F, F').

function ID$(\mathbf{y}, \mathbf{x}, P, G)$

1: **if** $\mathbf{x} = \emptyset$,
  return $\sum_{v \in \mathbf{v}\setminus\mathbf{y}} P(\mathbf{v})$.
2: **if** $\mathbf{V} \neq \mathrm{An}(\mathbf{Y})_G$,
  return ID$(\mathbf{y}, \mathbf{x} \cap \mathrm{An}(\mathbf{y})_G, P(\mathrm{An}(\mathbf{Y})_G), G[\mathrm{An}(\mathbf{Y})_G])$.
3: **let** $\mathbf{W} = (\mathbf{V} \setminus \mathbf{X}) \setminus \mathrm{An}(\mathbf{Y})_{G_{\overline{\mathbf{X}}}}$.
  if $\mathbf{W} \neq \emptyset$,
  return ID$(\mathbf{y}, \mathbf{x} \cup \mathbf{w}, P, G)$.
4: **if** $C(G[\mathbf{V} \setminus \mathbf{X}]) = \{G[\mathbf{S}_1], \ldots, G[\mathbf{S}_k]\}$,
  return $\sum_{v \in \mathbf{v}\setminus(\mathbf{y}\cup\mathbf{x})} \prod_{i=1}^{k}$ **ID**$(\mathbf{s}_i, \mathbf{v} \setminus \mathbf{s}_i, P, G)$.
  if $C(G[\mathbf{V} \setminus \mathbf{X}]) = \{G[\mathbf{S}]\}$,
5: **if** $C(G) = \{G\}$,
    throw FAIL$(G, G[\mathbf{S}])$.
6: **if** $G[\mathbf{S}] \in C(G)$,
    return $\sum_{v \in \mathbf{s}\setminus\mathbf{y}} \prod_{V_i \in \mathbf{S}} P(v_i | v_\pi^{(i-1)})$.
7: **if** $(\exists \mathbf{S}')\mathbf{S} \subset \mathbf{S}'$ **such that** $G[\mathbf{S}'] \in C(G)$,
    return ID$(\mathbf{y}, \mathbf{x} \cap \mathbf{s}', \prod_{V_i \in \mathbf{S}'} P(V_i | V_\pi^{(i-1)} \cap \mathbf{S}', v_\pi^{(i-1)} \setminus \mathbf{s}'), G[\mathbf{S}'])$.

4. Pruning of Variables

In this section we present a number of results that deal with variables that are not necessary for identification either by removing them from the graph or by considering them latent. When the causal effect $P_{\mathbf{x}}(\mathbf{y})$ is considered in an SMG we can present an outline of the pruning process:

1. Removal of non-ancestors of $\mathbf{Y}$.
2. Removal of ancestors of $\mathbf{X}$ that are connected to $\mathbf{Y}$ only via $\mathbf{X}$ under certain conditions.
3. Removal of vertices connected to other vertices only through a single vertex.
4. Identification in a latent projection under certain conditions.

Steps 2–4 are new and they are based on the results of this section. Step 1 is derived from a useful result by Shpitser and Pearl (2006) which states that for a causal effect $P_{\mathbf{x}}(\mathbf{y})$ we can always ignore non-ancestors of $\mathbf{Y}$.

Lemma 8 *Let $\mathbf{X}' = \mathbf{X} \cap An(\mathbf{Y})_G$. Then $P_{\mathbf{x}}(\mathbf{y})$ obtained from P in G is equal to $P'_{\mathbf{x}'}(\mathbf{y})$ obtained from $P' = P(An(\mathbf{Y})_G)$ in $G[An(\mathbf{Y})_G]$.*

Lemma 8 is implemented on line 2 of Algorithm 1. Not all ancestors of $\mathbf{Y}$ are always necessary for identification. The next result states that we may sometimes remove ancestors of $\mathbf{X}$ that are connected to $\mathbf{Y}$ only through $\mathbf{X}$.

Theorem 9 *Let G be an SMG and let $\mathbf{Z} \subset \mathbf{V}$ be the set of all vertices such that $\mathbf{X}$ intercepts all paths from $\mathbf{Z}$ to $\mathbf{Y}$. Then the causal effect $P_{\mathbf{x}}(\mathbf{y})$ obtained from P in G is equal to $P'_{\mathbf{x}}(\mathbf{y})$ obtained from $P' = P(\mathbf{V} \setminus \mathbf{Z})$ in $G[\mathbf{V} \setminus \mathbf{Z}]$ if $\mathbf{Z}$ contains no members of $\mathbf{X}$ and if $G[\mathbf{V} \setminus \mathbf{Z}] = L(G, \mathbf{V} \setminus \mathbf{Z})$.*

Proof Let $G' = G[\mathbf{V} \setminus \mathbf{Z}]$ and assume that $G' = L(G, \mathbf{V} \setminus \mathbf{Z})$. Let $\mathbf{U_Z}$, $\mathbf{U_{V \setminus Z}}$ and $\mathbf{U_X}$ be sets of unobserved variables such that for all $U \in \mathbf{U_Z}$ it holds that $\mathrm{Ch}(U)_{G_{\overline{\mathbf{X}}}} \subseteq \mathbf{Z}$, for all $U \in \mathbf{U_{V \setminus Z}}$ it holds that $\mathrm{Ch}(U)_{G_{\overline{\mathbf{X}}}} \subseteq \mathbf{V} \setminus \mathbf{Z}$ and for all $U \in \mathbf{U_X}$ it holds that $\mathrm{Ch}(U)_G \in \mathbf{X}$. The sets $\mathbf{U_Z}$, $\mathbf{U_{V \setminus Z}}$ and $\mathbf{U_X}$ partition $\mathbf{U}$ because $\mathbf{X}$ intercepts all paths from $\mathbf{Z}$ to $\mathbf{Y}$. According to the third rule of do-calculus $P_{\mathbf{x}}(\mathbf{y}) = P_{\mathbf{x},\mathbf{z}}(\mathbf{y})$ because the condition $(\mathbf{Y} \perp\!\!\!\perp \mathbf{Z}|\mathbf{X})_{G_{\overline{\mathbf{X}}}}$ holds as removing the edges incoming to $\mathbf{X}$ separates $\mathbf{X}$ from its ancestors. Applying the truncated factorization formula (Pearl, 2009) we have that

$$P_{\mathbf{x},\mathbf{z}}(\mathbf{y}) = \sum_{\mathbf{U}} \sum_{\mathbf{V} \setminus (\mathbf{Y} \cup \mathbf{X} \cup \mathbf{Z})} \prod_{\mathbf{V} \setminus (\mathbf{X} \cup \mathbf{Z})} P(v_i | \mathrm{Pa}(v_i)_G \setminus \{v_i\}) \prod_{\mathbf{U}} P(u_i).$$

Since variables in $\mathbf{U_Z}$ can only be parents of variables in $\mathbf{Z}$ or $\mathbf{X}$ in G, we can sum them out from the previous expression and obtain

$$P_{\mathbf{x},\mathbf{z}}(\mathbf{y}) = \sum_{\mathbf{U_{V \setminus Z}} \cup \mathbf{U_X}} \sum_{\mathbf{V} \setminus (\mathbf{Y} \cup \mathbf{X} \cup \mathbf{Z})} \prod_{\mathbf{V} \setminus (\mathbf{X} \cup \mathbf{Z})} P(v_i | \mathrm{Pa}(v_i)_G \setminus \{v_i\}) \prod_{\mathbf{U_{V \setminus Z}} \cup \mathbf{U_X}} P(u_i).$$

Similarly, variables in $\mathbf{U_X}$ can only be parents of variables in $\mathbf{X}$ in G, so we can also sum them out of the expression to obtain

$$P_{\mathbf{x},\mathbf{z}}(\mathbf{y}) = \sum_{\mathbf{U_{V \setminus Z}}} \sum_{\mathbf{V} \setminus (\mathbf{Y} \cup \mathbf{X} \cup \mathbf{Z})} \prod_{\mathbf{V} \setminus (\mathbf{X} \cup \mathbf{Z})} P(v_i | \mathrm{Pa}(v_i)_G \setminus \{v_i\}) \prod_{\mathbf{U_{V \setminus Z}}} P(u_i).$$

We let $\mathbf{V}' = \mathbf{V} \setminus \mathbf{Z}$. Verma (1993) showed that a graph and its latent projection have the same set of conditional independence relations among the observed variables. Because we have assumed that $G' = L(G, \mathbf{V} \setminus \mathbf{Z})$ every conditional independence between variables in $\mathbf{V}'$ and $\mathbf{U_{V'}}$ applies in both G and G'. We have that for all $V_i \in \mathbf{V}' \setminus \mathbf{X}$ it holds that $P(v_i|\mathrm{Pa}(v_i)_G \setminus \{v_i\}) = P'(v_i|\mathrm{Pa}(v_i)_{G'} \setminus \{v_i\})$ and for all $U_i \in \mathbf{U_{V'}}$ it holds that $P(u_i) = P'(u_i)$. Finally we obtain

$$P_{\mathbf{x},\mathbf{z}}(\mathbf{y}) = \sum_{\mathbf{U_{V'}}} \sum_{\mathbf{V}' \setminus (\mathbf{Y} \cup \mathbf{X})} \prod_{\mathbf{V}' \setminus \mathbf{X}} P'(v_i | \mathrm{Pa}(v_i)_{G'} \setminus \{v_i\}) \prod_{\mathbf{U_{V'}}} P'(u_i) = P'_{\mathbf{x}}(\mathbf{y}).$$

■

Theorem 9 can also be applied in a more general setting where a subset of $\mathbf{X}$ intercepts all paths from a set $\mathbf{Z}$ to $\mathbf{Y}$

Corollary 10 *Let G be an SMG and let $\mathbf{Z} \subset \mathbf{V}$ be the set of all vertices such that a set $\mathbf{W} \subseteq \mathbf{X}$ intercepts all paths from $\mathbf{Z}$ to $\mathbf{Y}$ and no member of $\mathbf{X} \setminus \mathbf{W}$ is a descendant of $\mathbf{W}$. Then the causal effect $P_{\mathbf{x}}(\mathbf{y})$ obtained from P in G is equal to $P'_{\mathbf{x}\setminus\mathbf{z}}(\mathbf{y})$ obtained from $P' = P(\mathbf{V} \setminus \mathbf{Z})$ in $G[\mathbf{V} \setminus \mathbf{Z}]$ if $\mathbf{Z}$ contains no members of $\mathbf{W}$ and if $G[\mathbf{V} \setminus \mathbf{Z}] = L(G, \mathbf{V} \setminus \mathbf{Z})$.*

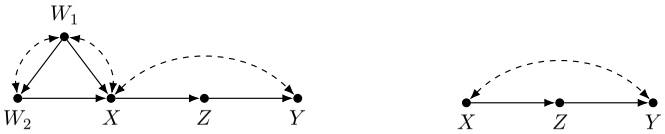
Proof Since $\mathbf{W}$ intercepts all paths from $\mathbf{Z}$ to $\mathbf{Y}$ and no member of $\mathbf{X} \setminus \mathbf{W}$ is a descendant of $\mathbf{W}$, it follows that no member of $\mathbf{W}$ is in $\mathbf{Z}$. According to the third rule of do-calculus we have that $(\mathbf{Y} \perp\!\!\!\perp \mathbf{Z}|\mathbf{X} \setminus \mathbf{Z})_{G_{\overline{\mathbf{X}\setminus\mathbf{Z}}}}$ and $P_{\mathbf{x}}(\mathbf{y}) = P_{\mathbf{x}\setminus\mathbf{z}}(\mathbf{y})$. The claim now follows by applying Theorem 9 to $P_{\mathbf{x}\setminus\mathbf{z}}(\mathbf{y})$. ■

Corollary 11 *When the causal effect $P_{\mathbf{x}}(\mathbf{y})$ is considered in graph G, a set of vertices $\mathbf{Z} = An(\mathbf{Y})_G \setminus Co(\mathbf{Y})_{G_{\overline{\mathbf{X}}}}$ can be removed from G if $G[\mathbf{V} \setminus \mathbf{Z}] = L(G, \mathbf{V} \setminus \mathbf{Z})$.*

Proof The set $\mathrm{An}(\mathbf{Y})_G$ contains $\mathbf{Y}$ in addition to the ancestors of $\mathbf{Y}$, and the set $\mathrm{Co}(\mathbf{Y})_{G_{\overline{\mathbf{X}}}}$ contains $\mathbf{Y}$ and all vertices that are connected to $\mathbf{Y}$ via a path that does not contain edges incoming to $\mathbf{X}$. Therefore, $\mathbf{Z}$ contains such ancestors of $\mathbf{Y}$ that all paths from $\mathbf{Z}$ to $\mathbf{Y}$ contain $\mathbf{X}$. The removal of $\mathbf{Z}$ from G is now licensed by Corollary 10. ■

Corollary 11 provides a constructive criterion for the set $\mathbf{Z}$ described in Corollary 10 when G consists only of $\mathbf{Y}$ and its ancestors. If a vertex Z_i is a member of $\mathrm{An}(\mathbf{Y})_G \setminus \mathrm{Co}(\mathbf{Y})_{G_{\overline{\mathbf{X}}}}$ then it must be connected to $\mathbf{Y}$ only through paths containing some $\mathbf{W}_{Z_i} \subseteq \mathbf{X}$. We can always choose the sets $\mathbf{W}_{Z_i}$ in such a way that the union $\mathbf{W} = \cup \mathbf{W}_{Z_i}$ over the members Z_i of $\mathrm{An}(\mathbf{Y})_G \setminus \mathrm{Co}(\mathbf{Y})_{G_{\overline{\mathbf{X}}}}$ has no descendants in $\mathbf{X} \setminus \mathbf{W}$. The set $\mathbf{W}$ intercepts all paths from $\mathbf{Z} = \cup \{Z_i\}$ to $\mathbf{Y}$. Conversely, if Z_i is a vertex such that a set $\mathbf{W} \subseteq \mathbf{X}$ intercepts all paths from Z_i to $\mathbf{Y}$, then Z_i cannot be connected to $\mathbf{Y}$ in $G_{\overline{\mathbf{X}}}$. If we assume that $G = G[\mathrm{An}(\mathbf{Y})_G]$ it follows that Z_i is a member of $\mathrm{An}(\mathbf{Y})_G \setminus \mathrm{Co}(\mathbf{Y})_{G_{\overline{\mathbf{X}}}}$.

We present a simple example to motivate the usefulness of Corollary 11. We apply the ID algorithm to identify the causal effect of X on Y in graph G of Figure 2(a). Applying



(a) Graph G. (b) Subgraph $G[\{X, Y, Z\}]$.

Figure 2: A graph for an example where Corollary 11 allows us to remove vertices W_1 and W_2 when the causal effect of X on Y is considered.

the ID algorithm results in the following expression for the causal effect

$$\sum_z P(z|w_1, w_2, x) \left(\sum_{w_1, w_2, x'} P(y|w_1, w_2, x', z) P(x'|w_1, w_2) P(w_2|w_1) P(w_1) \right).$$

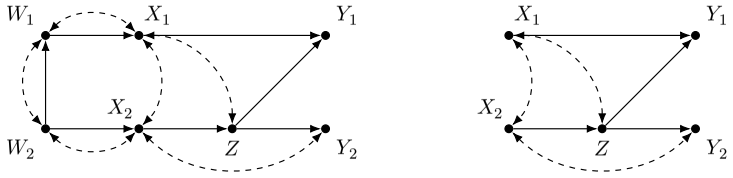
Applying Corollary 11 in this case would result in the removal of the vertices W_1 and W_2 from the graph, since they are ancestors of Y in G but not connected to Y in $G_{\overline{X}}$ and the corresponding latent projection is the subgraph $G[\{X, Y, Z\}]$ of Figure 2(b). Running the

ID algorithm in this subgraph provides us the following expression

$$\sum_z P(z|x)\left(\sum_{x'} P(y|x',z)P(x')\right).$$

We may consider this expression simpler compared to the previous output by noting that W_1 and W_2 do not appear in the expression and it has fewer unique terms. The same expression can also be obtained manually by applying the front-door criterion (Pearl, 2009).

Often the question of identifiability can not be answered directly by neither the back-door nor the front-door criterion which leads us to more general methods, such as the ID algorithm. We are interested in the causal effect of W_1, X_1 and X_2 on Y_1 and Y_2 in the graph of Figure 3(a).



(a) Graph G. (b) Subgraph $G[\{X_1, X_2, Z, Y_1, Y_2\}]$.

Figure 3: A graph for an example where the back-door and front-door criteria are unavailable, but Corollary 11 allows us to remove vertices W_1 and W_2 even when W_1 is part of the intervention.

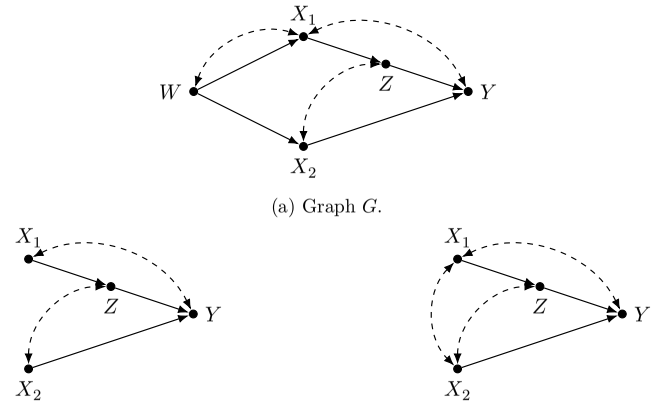
Direct application of the ID algorithm provides us with the following expression

$$\sum_z P(y_1|w_2,w_1,x_2,x_1,z)P(z|w_2,x_2)\left(\sum_{w_2,x_2'} P(y_2|w_2,x_2',z)P(x_2'|w_2)P(w_2)\right).$$

Corollary 11 licenses the removal of W_1 and W_2 from the graph. By running ID again in the resulting subgraph $G[\{X_1, X_2, Z, Y_1, Y_2\}]$ as shown in Figure 3(b) we obtain a simpler expression for the causal effect

$$\sum_z P(y_1|x_1,x_2,z)P(z|x_2)\left(\sum_{x_2'} P(y_2|x_2',z)P(x_2')\right).$$

The next example illustrates the necessity of the assumption $G[\mathbf{V}\setminus\mathbf{Z}] = L(G, \mathbf{V}\setminus\mathbf{Z})$ of Theorem 9 and Corollary 10. We are interested in the causal effect of X_1 and X_2 on Y in graph G of Figure 4(a). In this graph W is connected to Y only through X_1 and X_2, but the corresponding latent projection does not match the subgraph with W removed as seen in Figures 4(b) and 4(c). In G the causal effect is identifiable, but it is not identifiable in the latent projection $G' = L(G, \{X_1, X_2, Z, Y\})$. In this latent projection a bidirected edge exists between X_1 and X_2 and a hedge is formed by the C-forests G' and $G[\{Y\}]$.



(a) Graph G.

(b) Subgraph $G[\{X_1, X_2, Z, Y\}]$. (c) Latent projection $L(G, \{X_1, X_2, Z, Y\})$.

Figure 4: An example where the assumptions of Corollary 10 are not met because the subgraph in panel (b) and the latent projection in panel (c) differ from each other.

We may also remove sets of vertices that are connected to the rest of the graph only through a single vertex even when no intervention on the corresponding variable has taken place.

Theorem 12 *Let G be an SMG such that $G = G[An(\mathbf{Y})_G]$ for a set of vertices $\mathbf{Y}$ and let W be a vertex of G. If there exists a set $\mathbf{Z}$ such that $\mathbf{Z}\cap(\mathbf{Y}\cup\mathbf{X}) = \emptyset$ and $\mathbf{Z}$ is connected to $\mathbf{V}\setminus\mathbf{Z}$ only through W. Then the causal effect $P_{\mathbf{x}}(\mathbf{y})$ obtained from P in G is equal to $P'_{\mathbf{x}}(\mathbf{y})$ obtained from $P' = P(\mathbf{V}\setminus\mathbf{Z})$ in $G[\mathbf{V}\setminus\mathbf{Z}]$.*

Proof Let $G' = G[\mathbf{V}\setminus\mathbf{Z}]$ and let $\mathbf{U}_{\mathbf{Z}}$ and $\mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}$ be sets of unobserved variables such that for all $U\in\mathbf{U}_{\mathbf{Z}}$ it holds that $\mathrm{Ch}(U)_G \in \mathbf{Z}\cup\{W\}$ and for all $U\in\mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}$ it holds that $\mathrm{Ch}(U)_G \in (\mathbf{V}\setminus\mathbf{Z})\cup\{W\}$. Sets $\mathbf{U}_{\mathbf{Z}}$ and $\mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}$ partition $\mathbf{U}$ because $\mathbf{Z}$ is connected to $\mathbf{V}\setminus\mathbf{Z}$ only through W. Applying the truncated factorization formula yields

$$P_{\mathbf{x}}(\mathbf{y}) = \sum_{\mathbf{U}}\sum_{\mathbf{V}\setminus(\mathbf{Y}\cup\mathbf{X})} P(w|\mathrm{Pa}(w)_G\setminus\{w\}) \prod_{\mathbf{V}\setminus(\mathbf{X}\cup\{W\})} P(v_i|\mathrm{Pa}(v_i)_G\setminus\{v_i\}) \prod_{\mathbf{U}} P(u_i).$$

Since variables in $\mathbf{U}_{\mathbf{Z}}$ and $\mathbf{Z}$ can be connected to the other vertices of G only through W we can complete the marginalization over $\mathbf{Z}$ and $\mathbf{U}_{\mathbf{Z}}$

$$P_{\mathbf{x}}(\mathbf{y}) = \sum_{\mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}}\sum_{\mathbf{V}\setminus(\mathbf{Y}\cup\mathbf{X}\cup\mathbf{Z})} P(w|\mathrm{Pa}(w)_G\setminus(\mathbf{z}\cup\{w\})) \prod_{\mathbf{V}\setminus(\mathbf{X}\cup\mathbf{Z}\cup\{W\})} P(v_i|\mathrm{Pa}(v_i)_G\setminus\{v_i\}) \prod_{\mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}} P(u_i).$$

Because we have assumed that $\mathbf{Z}$ is disconnected from $\mathbf{V}\setminus\mathbf{Z}$ in $G_{\overline{W}}$ we have that $G[\mathbf{V}\setminus\mathbf{Z}] = L(G, \mathbf{V}\setminus\mathbf{Z})$. Therefore, just as in the proof of Theorem 9, we have that $P(v_i|\mathrm{Pa}(v_i)_G\setminus\{v_i\}) =$

$P'(v_i|\text{Pa}(v_i)_{G'} \setminus \{v_i\})$ for all $V_i \in \mathbf{V} \setminus (\mathbf{X} \cup \mathbf{Z} \cup \{W\})$ and $P(u_i) = P'(u_i)$ for all $U_i \in \mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}$. Additionally, we have $P(w|\text{Pa}(w)_G \setminus (\mathbf{z} \cup \{w\})) = P(w|Pa(w)_{G'} \setminus \{w\})$. Finally we obtain

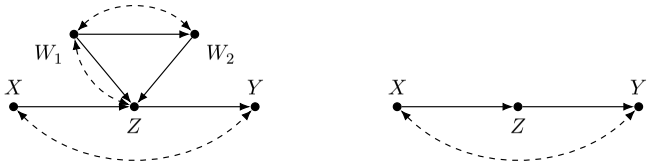
$$\begin{aligned} P_{\mathbf{x}}(\mathbf{y}) &= \sum_{\mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}} \sum_{\mathbf{V}\setminus(\mathbf{Y}\cup\mathbf{X}\cup\mathbf{Z})} P(w|\text{Pa}(w)_{G'} \setminus \{w\}) \prod_{\mathbf{V}\setminus(\mathbf{X}\cup\mathbf{Z}\cup\{W\})} P(v_i|\text{Pa}(v_i)_{G'} \setminus \{v_i\}) \prod_{\mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}} P(u_i) \\ &= \sum_{\mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}} \sum_{\mathbf{V}\setminus(\mathbf{Y}\cup\mathbf{X}\cup\mathbf{Z})} \prod_{\mathbf{V}\setminus(\mathbf{X}\cup\mathbf{Z})} P(v_i|\text{Pa}(v_i)_{G'} \setminus \{v_i\}) \prod_{\mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}} P(u_i) \\ &= P'_{\mathbf{x}}(\mathbf{y}). \end{aligned}$$

■

Corollary 13 *Let W be a vertex of an SMG G and let $\mathbf{R} = An(W)_{G_{\overline{\mathbf{X}}}} \setminus De(\mathbf{X})_G$. When the causal effect $P_{\mathbf{x}}(\mathbf{y})$ is considered in graph G, the set of vertices $\mathbf{T} = \mathbf{R} \setminus Co(\mathbf{V} \setminus \mathbf{R})_{G_{\overline{W}}}$ can be removed from the graph if $G = G[An(\mathbf{Y})_G]$.*

Proof No descendant of $\mathbf{X}$ can be removed via theorem 12 since they are connected to $\mathbf{X}$ and the set to be removed cannot itself contain $\mathbf{X}$. By removing descendants of $\mathbf{X}$ and $\mathbf{X}$ itself, and assuming that $G = G[\text{An}(\mathbf{Y})_G]$, we have that $\mathbf{T} \cap (\mathbf{X} \cup \mathbf{Y}) = \emptyset$. Thus it remains to remove those vertices from $\mathbf{R}$ that are connected to $\mathbf{V} \setminus \mathbf{R}$ through a path that does not contain W. Removal of the resulting set $\mathbf{T}$ from the graph is now licensed by theorem 12. ■

In the following example we consider the causal effect of X on Y in graph G of Figure 5(a) and show how Corollary 13 can be applied. The ID algorithm succeeds in identifying the



(a) Graph G. (b) Subgraph $G[\{X, Y, Z\}]$.

Figure 5: First example of Corollary 13.

causal effect and returns following expression for it

$$\sum_{w_1,w_2,z} P(z|x, w_1, w_2)P(w_2|x, w_1)P(w_1|x) \left(\sum_{x'} P(y|x', w_1, w_2, z)P(x') \right).$$

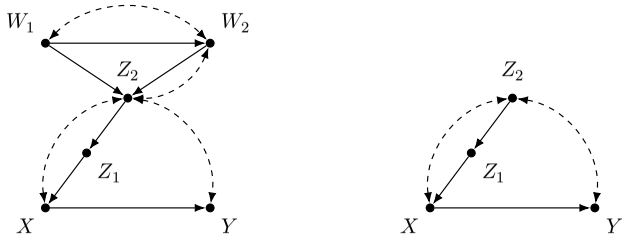
We apply Corollary 13 which allows us to remove W_1 and W_2 from the graph, since they are connected to other vertices of the graph through a single vertex Z. Applying the ID

algorithm in the resulting subgraph $G[\{X, Y, Z\}]$ of Figure 5(b) provides us the following expression

$$\sum_z P(z|x) \left(\sum_{x'} P(y|x', z)P(x') \right).$$

The same expression can be obtained manually by applying the front-door criterion.

We provide another example on Corollary 13 with a slightly more complicated graph. We are interested in the causal effect of X on Y in graph G of Figure 6(a) We obtain a



(a) Graph G. (b) Subgraph $G[\{X, Y, Z_1, Z_2\}$.

Figure 6: Second example of Corollary 13.

formula for the causal effect using the ID algorithm

$$\frac{\sum_{w_2,z_2} P(y|w_1, w_2, z_2, z_1, x)P(x|w_1, w_2, z_2, z_1)P(z_2|w_1, w_2)P(w_2|w_1)}{\left(\sum_{w_2,z_2,y} P(y|w_1, w_2, z_2, z_1, x)P(x|w_1, w_2, z_2, z_1)P(z_2|w_1, w_2)P(w_2|w_1)\right)}.$$

Vertices W_1 and W_2 are connected to other vertices only through Z_2 which allows us to remove them from the graph. We obtain a simpler formula for the causal effect from the subgraph $G[\{X, Y, Z_1, Z_2\}]$ of Figure 6(b)

$$\frac{\sum_{z_2} P(y|z_2, z_1, x)P(x|z_2, z_1)P(z_2)}{\sum_{z_2,y} P(y|z_2, z_1, x)P(x|z_2, z_1)P(z_2)}.$$

The previous results have allowed us to completely remove specific vertices from the graph. Next we will consider cases where a vertex is present in the graph, but it is not necessary to observe it. This means that instead of the original graph we may consider identifiability in the corresponding latent projection, as characterized by the following lemma.

Lemma 14 *Let $G = \langle \mathbf{V}, \mathbf{E} \rangle$ be an SMG and let $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$ be disjoint sets of variables. Let $P(\mathbf{V})$ be the joint distribution of $\mathbf{V}$. Then the causal effect of $\mathbf{X}$ on $\mathbf{Y}$ is identifiable from P' in the latent projection $L(G, \mathbf{V} \setminus \mathbf{Z})$ where $P' = P(\mathbf{V} \setminus \mathbf{Z})$, if and only if it is identifiable from P' in G.*

Proof Tian (2002) showed that the latent projection has the same topological relations over the observables and that it has the same set of maximal C-components. Thus if $P_{\mathbf{x}}(\mathbf{y})$

is identifiable from P' in G it is also identifiable from P' in $L(G, \mathbf{V} \setminus \mathbf{Z})$ with the same expression and vice versa. ■

In situations where $\mathbf{X}$ is a singleton we can exploit the following sufficient condition for identifiability by Tian and Pearl (2002).

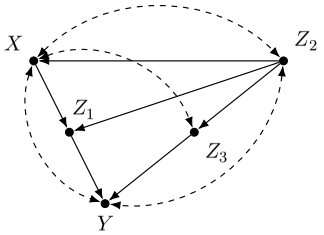
Theorem 15 *The causal effect $P_x(\mathbf{y})$ is identifiable if there is no bidirected path between X and any of its children in $G[An(\mathbf{Y})_G]$.*

We can regard any variable as latent when $\mathbf{X}$ is a singleton if the respective latent projection does not induce such a bidirected path.

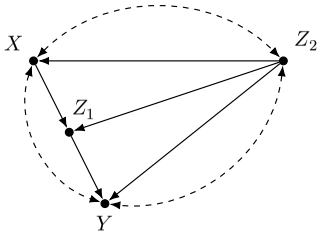
Corollary 16 *Let G be an SMG and let $\mathbf{Y}$ be a set of vertices. Let $X, W \in \mathbf{V}$ such that $X \neq W$, $W \notin \mathbf{Y}$ and $X, W \in An(\mathbf{Y})_G$. The causal effect $P_x(\mathbf{y})$ obtained from P in G is equal to $P'_x(\mathbf{y})$ obtained from $P' = P(An(\mathbf{Y})_G \setminus \{w\})$ and $G' = L(G[An(\mathbf{Y})_G], An(\mathbf{Y})_G \setminus \{W\})$ if there is no bidirected path from X to any of its children in G'.*

Proof By Theorem 15 the causal effect $P'_x(\mathbf{y})$ is identifiable from P' in G'. By Lemma 14 $P'_x(\mathbf{y})$ is now identifiable from P' in G. $P_x(y)$ obtained from P' in G is equal to $P^*_x(y)$ obtained from $P^* = P(\mathrm{An}(\mathbf{Y})_G)$ in $G[\mathrm{An}(\mathbf{Y})]$ since identifiability from P' implies identifiability from P. Finally, $P^*_x(\mathbf{y})$ obtained from $P^* = P(\mathrm{An}(\mathbf{Y})_G)$ in $G[\mathrm{An}(\mathbf{Y})_G]$ is equal to $P_x(y)$ obtained from P in G. ■

We continue with an example on how Corollary 16 can be applied in practice. We consider the causal effect of X on Y in the graph G of Figure 7(a). The causal effect is



(a) Graph G.



(b) Latent projection $L(G, \{X, Y, Z_1, Z_2\})$.

Figure 7: A graph for an example where Corollary 16 allows us to make variable Z_3 latent.

identifiable and the output of the ID algorithm is

$$\sum_{z_2,z_3,z_1} P(z_1|z_2,x)\frac{\left(\sum_{x'} P(y|z_2,x',z_3,z_1)P(z_3|z_2,x')P(x'|z_2)P(z_2)\right)}{\left(\sum_{x',y'} P(y'|z_2,x',z_3,z_1)P(z_3|z_2,x')P(x'|z_2)P(z_2)\right)}\times \left(\sum_{x',z_3,y'} P(y'|z_2,x',z_3,z_1)P(z_3|z_2,x')P(x'|z_2)P(z_2)\right)P(z_3|z_2).$$

We may apply Corollary 16 by noting that $G = G[\mathrm{An}(Y)_G]$ and that there is no bidirected path between X and its only child Z_1 in the latent projection $L(G, \mathbf{V} \setminus \{Z_3\})$ as depicted in Figure 7(b).

Running the ID algorithm in $L(G, \mathbf{V} \setminus \{Z_3\})$ results in the following expression

$$\sum_{z_2,z_1}\left(\sum_{x'} P(y|z_2,x',z_1)P(x'|z_2)P(z_2)\right)P(z_1|z_2,x).$$

5. Pruning Identifiability Algorithm

Corollaries 11, 13 and 16 can be implemented as additional steps for the ID algorithm. For Algorithm 2, line 3 implements Corollary 11, line 4 implements Corollary 13 and line 5 implements Corollary 16. Other lines are identical to the ID algorithm. This algorithm is provided by the R package *causaleffect* which implements various causal inference algorithms such as the original ID algorithm (Tikka and Karvanen, 2017a).

The ordering of the variables in the loop on line 4 has no effect on the resulting expression, but the ordering does matter on line 5. Choosing a different ordering may lead to a different expression. For example, when identifying the causal effect of X on Y in graph G of Figure 8 one may obtain either the back-door formula or the front-door formula by proceeding in either the topological ordering or the reverse-topological ordering of the vertices, respectively.

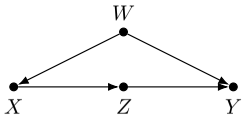


Figure 8: Graph G where different latent projections lead to different expressions for $P_x(y)$.

Algorithm 2 A pruning identifiability algorithm (PID) for causal effects.

INPUT: Value assignments $\mathbf{x}$ and $\mathbf{y}$, joint distribution $P(\mathbf{v})$ and an SMG $G = \langle \mathbf{V}, \mathbf{E} \rangle$. G is an I-map of P.

OUTPUT: Expression for $P_{\mathbf{x}}(\mathbf{y})$ in terms of $P(\mathbf{v})$ or **FAIL**(F, F').

function PID$(\mathbf{y}, \mathbf{x}, P, G)$
1: **if** $\mathbf{x} = \emptyset$,
  return $\sum_{v \in \mathbf{v} \setminus \mathbf{y}} P(\mathbf{v})$.
2: **if** $\mathbf{V} \neq \mathrm{An}(\mathbf{Y})_G$,
  return PID$(\mathbf{y}, \mathbf{x} \cap \mathrm{An}(\mathbf{y})_G, P(\mathrm{An}(\mathbf{Y})_G), G[\mathrm{An}(\mathbf{Y})_G])$
3: **let** $\mathbf{Z} = \mathrm{An}(\mathbf{Y})_G \setminus \mathrm{Co}(\mathbf{Y})_{G_{\overline{\mathbf{X}}}}$.
  if $\mathbf{Z} \neq \emptyset$ **and** $G[\mathbf{V} \setminus \mathbf{Z}] = L(G, \mathbf{V} \setminus \mathbf{Z})$,
  return PID$(\mathbf{y}, \mathbf{x} \setminus \mathbf{z}, P(\mathbf{V} \setminus \mathbf{Z}), G[\mathbf{V} \setminus \mathbf{Z}])$
4: **for** $W \in \mathbf{V} \setminus \mathbf{X}$ **do**
  let $\mathbf{R} = \mathrm{An}(W)_{G_{\overline{\mathbf{X}}}} \setminus \mathrm{De}(\mathbf{X})_G$.
  let $\mathbf{T} = \mathbf{T} \cup (\mathbf{R} \setminus \mathrm{Co}(\mathbf{V} \setminus \mathbf{R})_{G_{\overline{W}}})$.
  if $\mathbf{T} \neq \emptyset$,
  return PID$(\mathbf{y}, \mathbf{x}, P(\mathbf{V} \setminus \mathbf{T}), G[\mathbf{V} \setminus \mathbf{T}])$
5: **if** $\mathbf{X} = \{X\}$,
  let $G[\mathbf{S}_X] \in C(G), X \in \mathbf{S}_X$.
  if $\mathrm{Ch}(\mathbf{X})_{G[\mathbf{S}_X]} \setminus \mathbf{X} = \emptyset$,
    for $W \in \mathbf{V} \setminus (\mathbf{Y} \cup \mathbf{X})$ **do**
      let $G' = L(G, \mathbf{V} \setminus \{W\})$.
      let $G'[\mathbf{S}'_X] \in C(G'), X \in \mathbf{S}'_X$.
      if $\mathrm{Ch}(\mathbf{X})_{G'[\mathbf{S}'_X]} \setminus \mathbf{X} = \emptyset$,
        $P \leftarrow P(\mathbf{V} \setminus \{W\})$.
        $G \leftarrow G'$.
        $\mathbf{V} \leftarrow \mathbf{V} \setminus \{W\}$.
6: **let** $\mathbf{W} = (\mathbf{V} \setminus \mathbf{X}) \setminus \mathrm{An}(\mathbf{Y})_{G_{\overline{\mathbf{X}}}}$.
  if $\mathbf{W} \neq \emptyset$,
  return PID$(\mathbf{y}, \mathbf{x} \cup \mathbf{w}, P, G)$.
7: **if** $C(G[\mathbf{V} \setminus \mathbf{X}]) = \{G[\mathbf{S}_1], \ldots, G[\mathbf{S}_k]\}$,
  return $\sum_{v \in \mathbf{v} \setminus (\mathbf{y} \cup \mathbf{x})} \prod_{i=1}^{k}$ **PID**$(\mathbf{s}_i, \mathbf{v} \setminus \mathbf{s}_i, P, G)$.
  if $C(G[\mathbf{V} \setminus \mathbf{X}]) = \{G[\mathbf{S}]\}$,
8: **if** $C(G) = \{G\}$,
    throw FAIL$(G, G[\mathbf{S}])$.
9: **if** $G[\mathbf{S}] \in C(G)$,
    return $\sum_{v \in \mathbf{s} \setminus \mathbf{y}} \prod_{V_i \in \mathbf{S}} P(v_i | v_\pi^{(i-1)})$.
10: **if** $(\exists \mathbf{S}')\mathbf{S} \subset \mathbf{S}'$ **such that** $G[\mathbf{S}'] \in C(G)$,
    return PID$(\mathbf{y}, \mathbf{x} \cap \mathbf{s}', \prod_{V_i \in \mathbf{S}'} P(V_i | V_\pi^{(i-1)} \cap \mathbf{S}', v_\pi^{(i-1)} \setminus \mathbf{s}'), G[\mathbf{S}'])$.

On line 5 we first check whether the set $\mathbf{X}$ is a singleton. If so, we determine whether any children of X belong the same C-component as X. If no such children exist, we iterate over

the possible latent projections in an attempt to find one that does not induce a bidirected path between X and any of its children in the projection. After the new pruning steps have been carried out, we attempt identification using the original formulation of the ID algorithm.

We return to the example presented in the introduction and show how Algorithm 2 operates to derive the expression for $P_x(y)$ in the graph of Figure 1(a). We choose the topological ordering to be $Y > Z_1 > X > Z_3 > W_2 > Z_4 > Z_2 > W_1$. We begin on line 3, since $\mathbf{Z} = \mathrm{An}(Y)_G \setminus \mathrm{Co}(Y)_{G_{\overline{X}}} = \mathbf{V} \setminus \{W_1, W_2\}$ and continue by calling $\mathrm{PID}(y, x, P(\mathbf{V} \setminus \{W_1, W_2\}), G[\mathbf{V} \setminus \{W_1, W_2\}])$. In the presentation below, $\mathbf{V}$ and G refer to the set of vertices and graph in the current call of PID, respectively.

Next we enter the loop on line 4. When $W = Z_1$ we obtain $\mathbf{R} = \mathrm{An}(Z_1)_{G_{\overline{X}}} \setminus \mathrm{De}(X)_G = \{Z_2, Z_4\}$ and $\mathbf{R} \setminus \mathrm{Co}(\mathbf{V} \setminus \mathbf{R})_{G_{\overline{Z}_1}} = \{Z_2, Z_4\} \setminus \{Z_2\} = \{Z_4\}$ since Z_4 is an ancestor of Z_1 and it is disconnected from other vertices in $G_{\overline{Z}_1}$. Other choices of W result in an empty set. When the loop is completed we have $\mathbf{T} = \{Z_4\}$ and we continue by calling $\mathrm{PID}(y, x, P(\mathbf{V} \setminus \{Z_4\}), G[\mathbf{V} \setminus \{Z_4\}])$.

Since $\mathbf{X}$ is a singleton we end up on line 5 and find no children of X in the same C-component as X. We assume a reverse-topological ordering for the loop and begin with the latent projection $L(G, \mathbf{V} \setminus \{Z_2\})$. This projection creates a bidirected edge between X and Z_1 bringing them into the same C-component in the projection. Thus we continue with $L(G, \mathbf{V} \setminus \{Z_1\})$. This projection is also unsuitable, since Y is a child of X in the projection and there is a bidirected arc connecting them. We continue with $L(G, \mathbf{V} \setminus \{Z_3\})$ and find that X is not connected to its children via bidirected paths in the projection. Thus we set $P \leftarrow P(\mathbf{V} \setminus \{Z_3\})$, $G \leftarrow L(G, \mathbf{V} \setminus \{Z_3\})$ and $\mathbf{V} \leftarrow \mathbf{V} \setminus \{Z_3\}$. This projection is identical to the graph depicted in Figure 7(b). After these steps, only lines of the original ID algorithm are called, which results in the expression

$$\sum_{z_2, z_1} \left(\sum_{x'} P(y | z_2, z_1, x') P(x' | z_2) P(z_2) \right) P(z_1 | z_2, x).$$

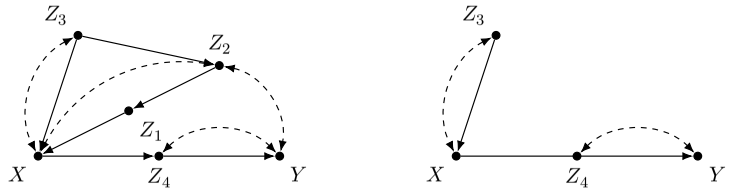
6. Examples on Recursive Pruning

Corollaries 10, 13 and 16 often provide direct benefits when applied before the ID algorithm. The following examples show why they are also useful as recursive steps as implemented in the PID algorithm.

We are tasked with identifying the causal effect of X on Y in graph G depicted in Figure 9(a)
Initially there are no vertices that are connected to other vertices only through X. As a recursive step of the ID algorithm, we are tasked with identifying $P'_{z_3,x}(y)$ from P', where

$$P' = \sum_{z_2} P(y | z_3, z_2, z_1, x) P(x | z_3, z_2, z_1) P(z_2 | z_3) P(z_3),$$

in the subgraph $G[\{Z_3, X, Y\}]$ shown in Figure 9(b). In this graph Z_3 is connected to other vertices only through X and it can be removed according to Corollary 11, since the corresponding latent projection is the subgraph $G[\{X, Y\}]$. Thus we sum out Z_3 from P'



(a) Graph G. (b) Subgraph $G[\{Z_3, X, Y\}]$.

Figure 9: A graph for the example of recursive application of Corollary 11 within the ID algorithm.

and the resulting expression for the causal effect is

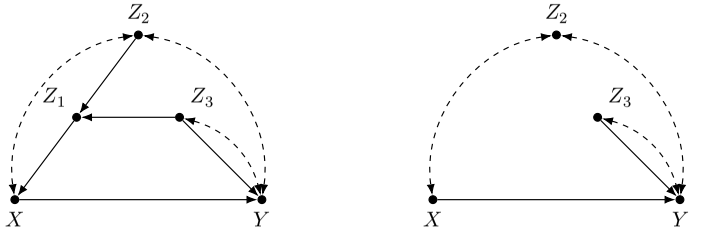
$$\sum_{z_4} \frac{\sum_{z_2,z_3} P(y|z_3,z_2,z_1,x,z_4)P(z_4|z_3,z_2,z_1,x)P(x|z_3,z_2,z_1)P(z_2|z_3)P(z_3)}{\sum_{z_2,z_3,y'} P(y'|z_3,z_2,z_1,x,z_4)P(z_4|z_3,z_2,z_1,x)P(x|z_3,z_2,z_1)P(z_2|z_3)P(z_3)}.$$

If Corollary 11 is not applied at this stage, the final expression is instead

$$\sum_{z_4} \left(\frac{\sum_{z_2} P(y|z_3,z_2,z_1,x,z_4)P(z_4|z_3,z_2,z_1,x)P(x|z_3,z_2,z_1)P(z_2|z_3)P(z_3)}{\sum_{z_2,y'} P(y'|z_3,z_2,z_1,x,z_4)P(z_4|z_3,z_2,z_1,x)P(x|z_3,z_2,z_1)P(z_2|z_3)P(z_3)} \times \frac{\sum_{z_2,y'} P(y'|z_3,z_2,z_1,x,z_4)P(z_4|z_3,z_2,z_1,x)P(x|z_3,z_2,z_1)P(z_2|z_3)P(z_3)}{\sum_{z_2,y',z_4} P(y'|z_3,z_2,z_1,x,z_4)P(z_4|z_3,z_2,z_1,x)P(x|z_3,z_2,z_1)P(z_2|z_3)P(z_3)} \right).$$

Using Corollary 11 provides us with a simpler expression in this situation by completely removing the second term from the product inside the summation.

Next we show how Corollary 13 can also be applied at a recursive step. Our interest lies in the causal effect of X on Y in graph G of Figure 10(a)



(a) Graph G. (b) Subgraph $G[\{Z_2, Z_3, X, Y\}]$.

Figure 10: A graph for the example of recursive application of Corollary 13 within the ID algorithm.

There are no vertices that are connected to other vertices in the graph via a single vertex. When the ID algorithm is applied we eventually reach a step where the causal effect of

$P'_{z_2,x}(y)$ is to be identified from P', where

$$P' = P(y|z_2,z_3,z_1,x)P(x|z_2,z_3,z_1)P(z_3|z_2)P(z_2),$$

in the subgraph $G[\{Z_2, Z_3, X, Y\}]$ which is depicted in Figure 10(b).

In this graph Z_3 can be removed according to Corollary 13, since Z_3 is connected to other vertices of the subgraph only through Y. The resulting distribution is obtained by summing out Z_3 from P'. The final expression for the causal effect is now

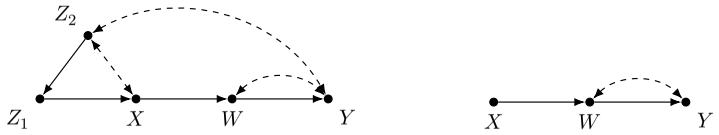
$$\frac{\sum_{z_2,z_3} P(y|z_2,z_3,z_1,x)P(x|z_2,z_3,z_1)P(z_3|z_2)P(z_2)}{\sum_{z_2,z_3,y'} P(y'|z_2,z_3,z_1,x)P(x|z_2,z_3,z_1)P(z_3|z_2)P(z_2)}. \quad (1)$$

If Corollary 13 is not applied, the resulting expression is instead

$$\sum_{z_3} \left(\frac{\sum_{z_2} P(y|z_2,z_3,z_1,x)P(x|z_2,z_3,z_1)P(z_3|z_2)P(z_2)}{\sum_{z_2,y'} P(y'|z_2,z_3,z_1,x)P(x|z_2,z_3,z_1)P(z_3|z_2)P(z_2)} \times \sum_{z_2,x,y'} P(y'|z_2,z_3,z_1,x)P(x|z_2,z_3,z_1)P(z_3|z_2)P(z_2) \right).$$

As in the previous example, the benefit of applying Corollary 13 is apparent.

We can also take advantage of latent projections recursively via Corollary 16 as shown in the next example. Our interest lies in the causal effect of X on Y in graph G of Figure 11.



(a) Graph G. (b) Subgraph $G[\{X, W, Y\}]$.

Figure 11: A graph for the example of recursive application of Corollary 16 within the ID algorithm.

Here X is connected to its child W via a bidirected path and thus they belong to the same C-component rendering Corollary 16 unusable at this time. However, as a recursive step of the ID algorithm we have to identify $P'_x(y)$ from P', where

$$P' = \sum_{z_2} P(y|z_2,z_1,x,w)P(w|z_2,z_1,x)P(x|z_2,z_1)P(z_2)$$

in the subgraph $G[\{X, W, Y\}]$. In this subgraph X is not connected to its child W via a bidirected path. We also find that the latent projection $L(G[\{X, W, Y\}], \{X, Y\})$ does not induce such a path between X and Y. We can now continue identification in this latent projection and sum out W from P'. The resulting expression for the causal effect is

$$\frac{\sum_{z_2,w} P(y|z_2,z_1,x,w)P(w|z_2,z_1,x)P(x|z_2,z_1)P(z_2)}{\sum_{z_2,w,y'} P(y'|z_2,z_1,x,w)P(w|z_2,z_1,x)P(x|z_2,z_1)P(z_2)},$$

whereas the expression without applying the corollary is instead

$$\sum_w \left(\frac{\sum_{z_2} P(y|z_2,z_1,x,w)P(w|z_2,z_1,x)P(x|z_2,z_1)P(z_2)}{\sum_{z_2,y'} P(y'|z_2,z_1,x,w)P(w|z_2,z_1,x)P(x|z_2,z_1)P(z_2)} \times \frac{\sum_{z_2,y'} P(y'|z_2,z_1,x,w)P(w|z_2,z_1,x)P(x|z_2,z_1)P(z_2)}{\sum_{z_2,w',y'} P(y'|z_2,z_1,x,w)P(w'|z_2,z_1,x)P(x|z_2,z_1)P(z_2)} \right).$$

Additional examples are provided as an R script (R Core Team, 2017) at the JMLR online paper repository. The script also includes all of the examples presented in this paper.

An interesting question is how pruning works together with simplification presented in (Tikka and Karvanen, 2017b). We return to the example on identifying the causal effect of X on Y in the graph of Figure 10. If we apply the ID algorithm without pruning and perform simplification as a post-processing step, then the resulting expression is

$$\sum_{z_3} \frac{\sum_{z_2} P(y|z_2,z_3,z_1,x)P(x|z_2,z_3,z_1)P(z_3|z_2)P(z_2)}{\sum_{z_2} P(x|z_2,z_3,z_1)P(z_3|z_2)P(z_2)} P(z_3|z_2). \tag{2}$$

This expression is in some aspects simpler than expression (1) obtained using pruning alone, but does contain a sum over Z_3 that was not originally present.

When pruning is introduced to the ID algorithm and simplification is again applied, the resulting expression is instead

$$\frac{\sum_{z_3,z_2} P(y|z_2,z_3,z_1,x)P(x|z_2,z_3,z_1)P(z_3|z_2)P(z_2)}{\sum_{z_2} P(x|z_2,z_1)P(z_2)},$$

which is noticeably simpler than expressions (1) and (2). This example shows, that when pruning methods are employed together with simplification, a simpler expression can be reached than what is possible with either pruning or simplification alone.

7. Discussion

We have presented criteria for removing variables from causal models that are not necessary to achieve identifiability for a given causal effect, and showed how these criteria can be applied in practice. We integrated our results into a new version of the ID algorithm called PID as presented in Algorithm 2 to facilitate automatic processing of identifiability queries. It should be noted that the ID algorithm already performs some pruning such as removing non-ancestors of $\mathbf{Y}$.

The pruning operations carried out by Algorithm 2 can significantly simplify the resulting expression compared to the traditional ID algorithm. Benefits of simplification can be realized in various settings. A simpler expression is easier to understand and to evaluate, since the dimensionality of the problem has been reduced. This is especially true for settings where the expression has to be evaluated repeatedly. Simplification can also help dealing with data where some variables are affected by bias or contain missing data. Obtaining an expression that does not contain these variables has clear advantages. It may also be of interest to obtain a different expression for the same causal effect.

The choice of variable ordering on line 5 of Algorithm 2 is not arbitrary. However, available external knowledge may guide our selection to prefer certain orderings. For example, in a situation where two latent projections are mutually exclusive, we may prefer an ordering where the variable that is associated with the smallest cost, or of which we have the most accurate measurements is not considered latent.

When PID is applied in conjunction with simplification methods described in (Tikka and Karvanen, 2017b), various situations that lead to complex expressions can be taken into account. These methods complement each other, since the results in this paper deal with completely removing variables from the resulting expression, whereas the simplification methods focus on symbolic summation of so-called atomic expressions, which are expression consisting of a single sum and a number of product terms. An expression for a causal effect may consist of multiple atomic expressions, some of which can be simplified and some of which can not.

We showed via examples that our improvements are not simply pre-processing steps to be carried out before calling the ID algorithm, but actually provide significant benefits when applied recursively. As the ID algorithm manipulates the original graph it often enables the application of our results as well. As hedges characterize identifiability, it is possible to consider latent projections in a more general manner, but this is not necessarily beneficial for simplification. One could construct an algorithm that performs a search over the possible subsets of $\mathbf{V}$, and checks whether identifiability is retained in the corresponding latent projection. However, as we have shown via examples, this may not be enough to obtain a simpler expression, and the recursive structure of the ID algorithm needs to be taken advantage of. Instead, we could consider a variant of the PID algorithm, where line 5 is replaced by this procedure. However, one must be careful when applying this method, so that the computation does not become intractable when the number of vertices increases due to the complexity of the search.

Acknowledgments

We wish to thank Professor Jukka Nyblom for his comments that greatly helped to improve this paper. We also thank the anonymous reviewers for their insightful feedback. The work belongs to the profiling area "Decision analytics utilizing causal models and multiobjective optimization" (DEMO) supported by Academy of Finland (grant number 311877).

References

A. P. Dawid. Conditional independence in statistical theory. *Journal of the Royal Statistical Society B*, 41:1–31, 1979.

D. Geiger, T. Verma, and J. Pearl. Identifying independence in Bayesian networks. *Networks*, 20(5):507–534, 1990.

Y. Huang and M. Valtorta. Pearl's calculus of intervention is complete. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*, pages 217–224. AUAI Press, 2006.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

J. Pearl. Causal diagrams for empirical research. *Biometrika*, 82:669–688, 1995.

J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2nd edition, 2009.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, 2017. URL `https://www.R-project.org/`.

I. Shpitser and J. Pearl. Identification of joint interventional distributions in recursive semi-Markovian causal models. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, pages 1219–1226. AAAI Press, 2006.

I. Shpitser and J. Pearl. Complete identification methods for the causal hierarchy. *Journal of Machine Learning Research*, 9:1941–1979, 2008. ISSN 1532-4435.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT press, 2nd edition, 2000.

J. Tian. *Studies in Causal Reasoning and Learning*. PhD thesis, Department of Computer Science, University of California, Los Angeles, 2002.

J. Tian and J. Pearl. A general identification condition for causal effects. In *Proceedings of the 18th National Conference on Artificial Intelligence*, pages 567–573. AAAI Press, 2002.

S. Tikka and J. Karvanen. Identifying causal effects with the R package causaleffect. *Journal of Statistical Software*, 76(12):1–30, 2017a.

S. Tikka and J. Karvanen. Simplifying probabilistic expressions in causal inference. *Journal of Machine Learning Research*, 18(36):1–30, 2017b. URL `http://jmlr.org/papers/v18/16-166.html`.

T. S. Verma. Graphical aspects of causal models. Technical report, Department of Computer Science, University of California, Los Angeles, 1993. R-191.

Journal of Machine Learning Research 18 (2018) 1-38 Submitted 10/16; Revised 12/17; Published 6/18

Active Nearest-Neighbor Learning in Metric Spaces

Aryeh Kontorovich KARYEH@CS.BGU.AC.IL
Sivan Sabato SABATOS@CS.BGU.AC.IL
Department of Computer Science
Ben-Gurion University of the Negev
Beer Sheva 8499000, Israel

Ruth Urner RUTH@EECS.YORKU.CA
Lassonde School of Engineeging, EECS Department
York University
Toronto, ON, Canada

Editor: Andreas Krause

Abstract

We propose a pool-based non-parametric active learning algorithm for general metric spaces, called MArgin Regularized Metric Active Nearest Neighbor (MARMANN), which outputs a nearest-neighbor classifier. We give prediction error guarantees that depend on the noisy-margin properties of the input sample, and are competitive with those obtained by previously proposed passive learners. We prove that the label complexity of MARMANN is significantly lower than that of any passive learner with similar error guarantees. MARMANN is based on a generalized sample compression scheme, and a new label-efficient active model-selection procedure.

Keywords: Nearest-neighbors, active learning, metric spaces, non-parametric learning

1. Introduction

Active learning is a framework for reducing the amount of label supervision for prediction tasks. While labeling large amounts of data can be expensive and time-consuming, unlabeled data is often much easier to come by. In this paper we propose a non-parametric pool-based active learning algorithm for general metric spaces, which outputs a nearest-neighbor classifier.

In pool-based active learning (McCallum and Nigam, 1998), a collection of random examples is provided, and the algorithm can interactively query an oracle to label some of the examples. The goal is good prediction accuracy, while keeping the label complexity — that is, the number of labels queried — low. Our algorithm, MArgin Regularized Metric Active Nearest Neighbor (MARMANN), receives a pool of unlabeled examples in a general metric space, and outputs a variant of the 1-nearest-neighbor classifier, implemented by a 1-nearest-neighbor rule. The algorithm obtains a prediction error guarantee that depends on a noisy-margin property of the input sample, and has a provably smaller label complexity than any passive learner with a similar guarantee.

Active learning has been mostly studied in a parametric setting, in which learning takes place with respect to a fixed hypothesis class with a bounded capacity. There has also been some work on analyzing non-parametric active learning strategies under certain distributional assumptions (see Section 1.1 for more discussion on this). However, the question of whether active querying strategies can yield label savings for non-parametric methods in a general setting, without distributional assumptions, had not been analyzed prior to this work. Here, we provide a first demonstration that this is indeed possible. We discuss related work in detail in Section 1.1 below.

Our contributions. MARMANN is a new non-parametric pool-based active learning algorithm, which obtains an error guarantee competitive with that of a noisy-margin-based passive learner. Additionally, it provably uses significantly fewer labels in nontrivial regimes. As far as the authors are aware, this is the first non-parametric active learner for general metric spaces, which achieves competitive prediction error guarantees to the passive learner, while provably improving label complexity. The guarantees of MARMANN are given in Theorem 4 in Section 3. We further provide a passive learning lower bound (Theorem 5), which together with Theorem 4 shows that MARMANN can have a significantly reduced label complexity compared to any passive learner. The passive lower bound is more general than previous lower bounds, relies on a novel technique, and may be of independent interest. Additionally, we give an active label complexity lower bound (Theorem 6), which holds for any active learner with similar error guarantees as MARMANN. The proof of this active lower bound relies on a new No-Free-Lunch type result, which holds for active learning algorithms.

Our approach. Previous passive learning approaches to classification using nearest-neighbor rules under noisy-margin assumptions (Gottlieb et al., 2014b, 2017) provide statistical guarantees using sample compression bounds (Graepel et al., 2005). Their finite-sample guarantees depend on the number of noisy labels relative to an optimal margin scale.

A central challenge in the active setting is performing model selection to select a margin scale with a low label complexity. A key insight that we exploit in this work is that by designing a new labeling scheme for the compression set, we can construct the compression set and estimate its error with label-efficient procedures. We obtain statistical guarantees for this approach using generalization bounds for sample compression with side information.

We derive a label-efficient, as well as computationally efficient, active model-selection procedure. This procedure finds a good scale by estimating the sample error for some scales, using a small number of active querying rounds. Crucially, unlike cross-validation, our model-selection procedure does not require a number of labels that depends on the worst possible scale, nor does it test many scales. This allows our label complexity bounds to be low, and to depend only on the final scale selected by the algorithm. Our error guarantee is a constant factor over the error guarantee of the passive learner of Gottlieb et al. (2017). An approach similar to Gottlieb et al. (2017), proposed in Gottlieb et al. (2014a), has been shown to be Bayes consistent (Kontorovich and Weiss, 2015). The Bayes-consistency of the passive version of our approach has recently been established (Kontorovich et al., 2017).

Paper structure. Related work is discussed in Section 1.1. We lay down the preliminaries in Section 2. In Section 3 we provide our main result: Theorem 4, which gives error and label complexity guarantees for MARMANN. Additionally we state the passive and active lower bounds, Theorem 5 and Theorem 6. The rest of the paper is devoted to the description and analysis of MARMANN, and proof of the main results. Section 4 shows how MARMANN defines the nearest neighbor rule for a given scale, and Section 5 describes the model selection procedure of MARMANN. Theorem 4 is proved in Section 6, based on a framework for compression with side information. The passive lower bound in Theorem 5 is proved in Section 7. The active lower bound Theorem 6 is proved in Section 8. We conclude with a discussion in Section 9.

1.1 Related Work

The theory of active learning has received considerable attention in the past decade (e.g., Dasgupta, 2004; Balcan et al., 2007, 2009; Hanneke, 2011; Hanneke and Yang, 2015). Active learning theory has been mostly studied in a parametric setting (that is, learning with respect to a fixed hypothesis class with a bounded capacity). Benefits and limitations of various active querying strategies have been proven in the realizable setting (Dasgupta, 2004; Balcan et al., 2007; Gonen et al., 2013) as well as in the agnostic case (Balcan et al., 2009; Hanneke, 2011; Awasthi et al., 2014). It has also been shown that active queries can also be beneficial for regression tasks (Castro et al., 2005; Sabato and Munos, 2014). Further, an active model selection procedure has been developed for the parametric setting (Balcan et al., 2010).

The potential benefits of active learning for non-parametric settings are less well understood. Practical Bayesian graph-based active learning methods (Zhu et al., 2003; Wei et al., 2015) rely on generative model assumptions, and therefore come without distribution-free performance guarantees. From a theoretical perspective, the label complexity of graph based active learning has mostly been analyzed in terms of combinatorial graph parameters (Cesa-Bianchi et al., 2010; Dasarathy et al., 2015). While the latter work provides some statistical guarantees under specific conditions on the data-generating distribution, this type of analysis does not yield distribution-free statistical performance guarantees.

Castro et al. (2005); Castro and Nowak (2008) analyze minimax rates for non-parametric regression and classification respectively, for a class of distributions in Euclidean space, characterized by decision boundary regularity and noise conditions with uniform marginals. The paradigm of cluster-based active learning (Dasgupta and Hsu, 2008) has been shown to provide label savings under some distributional clusterability assumptions (Urner et al., 2013; Kpotufe et al., 2015). Dasgupta and Hsu (2008) showed that a suitable cluster-tree can yield label savings in this framework, and papers following up (Urner et al., 2013; Kpotufe et al., 2015) quantified the label savings under distributional clusterability assumptions. However, no active non-parametric strategy has been proposed so far that has label complexity guarantees for i.i.d. data from general distributions and general metric spaces. Here, we provide the first such algorithm and guarantees.

The passive nearest-neighbor classifier, introduced by Fix and Hodges (1951, 1989), is popular among theorists and practitioners alike (Fix and Hodges, 1989; Cover and Hart, 1967; Stone, 1977; Kulkarni and Posner, 1995; Boiman et al., 2008). This paradigm is applicable in general metric spaces, and its simplicity is an attractive feature for both implementation and analysis. When appropriately regularized — either by taking a majority vote among the k nearest neighbors (Stone, 1977; Devroye and Györfi, 1985; Zhao, 1987), or by enforcing a *margin* separating the classes (von Luxburg and Bousquet, 2004; Gottlieb et al., 2014a; Kontorovich and Weiss, 2015; Kontorovich et al., 2017) — this type of learner can be made Bayes-consistent. Another desirable property of nearest-neighbor-based methods is their ability to generalize at a rate that scales with the intrinsic data dimension, which can be much lower than that of the ambient space (Kpotufe, 2011; Gottlieb et al., 2014a, 2016; Chaudhuri and Dasgupta, 2014). Furthermore, margin-based regularization makes nearest neighbor classifiers ideally suited for sample compression, which yields a compact representation, faster classification runtime, and improved generalization performance (Gottlieb et al., 2014b). The resulting error guarantees can be stated in terms of the sample's noisy-margin, which depends on the distances between differently-labeled examples in the input sample.

Active learning strategies specific to nearest neighbor classification have recently received attention. It has been shown that certain active querying rules maintain Bayes consistency for nearest neighbor classification, while other, seemingly natural, rules do not lead to a consistent algorithm (Dasgupta, 2012). A selective querying strategy has been shown to be beneficial for nearest neighbors under covariate shift (Berlind and Urner, 2015), where one needs to adapt to a change in the data generating process. However, the querying rule in that work is based solely on information in the unlabeled data, to account for a shift in the distribution over the covariates. It does not imply any label savings in the standard learning setting, where training and test distribution are identical. In contrast, our current work demonstrates how an active learner can take label information into account, to reduce the label complexity of a general nearest neighbor method in the standard setting.

1.2 A Remark on Bayes-Consistency

We remark on the Bayes-consistency of the margin-based passive 1-NN methods. In Gottlieb et al. (2014a), a PAC-style generalization bound was given. At a given scale t, the algorithm first ensured t-separation of the sample by solving solving a minimum vertex cover problem to eliminate the t-blocking pairs. Following that, the hypothesis was constructed as a Lipschitz extension from the remaining sample; the latter is computationally implemented as a nearest neighbor classifier. Structural Risk Minimization (SRM) was used to select the optimal scale t. A very close variant of this learner was shown to be Bayes-consistent by Kontorovich and Weiss (2015). The only difference between the two is that the former analyzed the hypothesis complexity in terms of fat-shattering dimension while the latter via Rademacher averages. Thus, a margin-regularized 1-NN classifier was shown to be Bayes-consistent; however, no compression was involved.

A compression-based alternative to Lipschitz extension was proposed in Gottlieb et al. (2014b). The idea is again to ensure t-separation via vertex cover and then compress the remaining sample down to a t-net. We conjecture that this latter algorithm is also Bayes-consistent, but currently have no proof. If instead one considers a compression-based passive learner implemented as in this paper (by taking majority vote in each Voronoi region rather than enforcing t-separation via vertex cover), the resulting classifier is indeed Bayes-consistent, as was recently shown by Kontorovich et al. (2017).

2. Preliminaries

In this section we lay down the necessary preliminaries. We formally define the setting and necessary notation in Section 2.1. We discuss nets in metric spaces in Section 2.2, and present the guarantees of the compression-based passive learner of Gottlieb et al. (2017) in Section 2.3.

2.1 Setting and Notation

For positive integers n, denote $[n] := \{1, \ldots, n\}$. We consider learning in a general metric space $(\mathcal{X}, \rho)$, where $\mathcal{X}$ is a set and ρ is the metric on $\mathcal{X}$. Our problem setting is that of classification of the instance space $\mathcal{X}$ into some finite label set $\mathcal{Y}$. Assume that there is some distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, and let $S \sim \mathcal{D}^m$ be a labeled sample of size m, where m is an integer. Denote the sequence of unlabeled points in S by $\mathbb{U}(S)$. We sometimes treat S and $\mathbb{U}(S)$ as multisets, since the order is

unimportant. For a labeled multiset $S \subseteq \mathcal{X} \times \mathcal{Y}$ and $y \in \mathcal{Y}$, denote $S^y := \{x \mid (x,y) \in S\}$; in particular, $\mathbb{U}(S) = \cup_{y \in \mathcal{Y}} S^y$.

The error of a classifier $h : \mathcal{X} \to \mathcal{Y}$ on $\mathcal{D}$, for any fixed h, is denoted

$$\mathrm{err}(h, \mathcal{D}) := \mathbb{P}[h(X) \neq Y],$$

where $(X,Y) \sim \mathcal{D}$. The empirical error on a labeled sample S instantiates to

$$\mathrm{err}(h, S) = \frac{1}{|S|} \sum \mathbb{I}[h(X) \neq Y].$$

A passive learner receives a labeled sample S_{in} as input. An active learner receives the unlabeled part of the sample $U_{\text{in}} := \mathbb{U}(S_{\text{in}})$ as input, and is allowed to interactively select examples from U_{in} and request their label from S_{in}. In other words, the active learner iteratively selects an example and requests its label, wherein all the labels requested so far can be used to make the next selection.

When either learner terminates, it outputs a classifier $\hat{h} : \mathcal{X} \to \mathcal{Y}$, with the goal of achieving a low $\mathrm{err}(\hat{h}, \mathcal{D})$. An additional goal of the active learner is to achieve a performance competitive with that of the passive learner, while querying considerably fewer labels.

The diameter of a set $A \subseteq \mathcal{X}$ is defined by

$$\mathsf{diam}(A) := \sup_{a,a' \in A} \rho(a,a').$$

For a finite set $U = \{u_1, \ldots, u_{|U|}\} \subseteq \mathcal{X}$ with some fixed numbering of its elements,[1] denote the index of the closest point in U to $x \in \mathcal{X}$ by

$$\kappa(x, U) := \operatorname*{argmin}_{i : x_i \in U} \rho(x, x_i).$$

We assume here and throughout this work that when there is more than one minimizer for $\rho(x, x_i)$, ties are broken arbitrarily (but in a consistent and deterministic fashion). Any labeled sample $S = ((x_i, y_i))_{i \in [k]}$ naturally induces the 1-nearest-neighbor classifier $h_S^{\text{nn}} : \mathcal{X} \to \mathcal{Y}$, via $h_S^{\text{nn}}(x) := y_{\kappa(x, \mathbb{U}(S))}$. For a set $Z \subseteq \mathcal{X}$, denote by

$$\kappa(Z, U) := \{\kappa(z, U) \mid z \in Z\}$$

the set of all the indices $\kappa(z, U)$, as defined above. For $x \in \mathcal{X}$, and $t > 0$, denote by $\mathsf{ball}(x,t)$ the (closed) ball of radius t around x:

$$\mathsf{ball}(x,t) := \left\{x' \in \mathcal{X} \mid \rho(x,x') \leq t\right\}.$$

2.2 Nets

A set $A \subseteq \mathcal{X}$ is *t-separated* if $\inf_{a,a' \in A : a \neq a'} \rho(a,a') \geq t$. For $A \subseteq B \subseteq \mathcal{X}$, the set A is a *t-net* of B if A is t-separated and $B \subseteq \bigcup_{a \in A} \mathsf{ball}(a,t)$. Thus, A is a t-net of B if it is both a t-covering and a t-packing.

1. Invoking the well-ordering principle, we may assume $\mathcal{X}$ to be well-ordered and then any $U \subseteq \mathcal{X}$ inherits the ordering of $\mathcal{X}$.

The size of a t-net of a metric space is strongly related to its *doubling dimension*. The doubling dimension is the effective dimension of the metric space, which controls generalization and runtime performance of nearest-neighbors (Kpotufe, 2011; Gottlieb et al., 2014a). It is defined as follows. Let $\lambda = \lambda(\mathcal{X})$ be the smallest number such that every ball in $\mathcal{X}$ can be covered by λ balls of half its radius, where all balls are centered at points of $\mathcal{X}$. Formally,

$$\lambda(\mathcal{X}) := \min\{\lambda \in \mathbb{N} : \forall x \in \mathcal{X}, r > 0, \quad \exists x_1, \ldots, x_\lambda \in \mathcal{X} : \mathsf{ball}(x,r) \subseteq \cup_{i=1}^{\lambda} \mathsf{ball}(x_i, r/2)\}.$$

Then the doubling dimension of $\mathcal{X}$ is defined by $\mathsf{ddim}(\mathcal{X}) := \log_2 \lambda$. In line with modern literature, we work in the low-dimension, large-sample regime, where the doubling dimension is assumed to be constant, and hence sample complexity and algorithmic runtime may depend on it exponentially. This exponential dependence is unavoidable, even under margin assumptions, as previous analyses (Kpotufe, 2011; Gottlieb et al., 2014a) indicate. Generalization bounds in terms of the doubling dimension of the hypothesis space were established in Bshouty et al. (2009), while runtime and generalization errors in terms of $\mathsf{ddim}(\mathcal{X})$ were given in Gottlieb et al. (2014a).

As shown in Gottlieb and Krauthgamer (2013), the doubling dimension is "almost hereditary" in the sense that for $A \subset \mathcal{X}$, we have $\mathsf{ddim}(A) \leq c\mathsf{ddim}(\mathcal{X})$ for some universal constant $c \leq 2$ (Feldmann et al., 2015, Lemma 6.6). In the works cited above, where generalization bounds are stated in terms of $\mathsf{ddim}(\mathcal{X})$, one can obtain tighter bounds in terms of $\mathsf{ddim}(\mathbb{U}(S))$ when the latter is substantially lower than that of the ambient space, and it is also possible to perform metric dimensionality reduction, as in Gottlieb et al. (2016).

Constructing a minimum size t-net for a general set B is NP-hard (Gottlieb and Krauthgamer, 2013). However, a simple greedy algorithm constructs a (not necessarily minimal) t-net in time $O(m^2)$ (Gottlieb et al., 2014b, Algorithm 1). There is also an algorithm for constructing a t-net in time $2^{O(\mathsf{ddim}(\mathcal{X}))} m \log(1/t)$ (Krauthgamer and Lee, 2004; Gottlieb et al., 2014b). The size of any t-net of a metric space $A \subseteq \mathcal{X}$ is at most

$$\lceil \mathsf{diam}(A)/t \rceil^{\mathsf{ddim}(\mathcal{X})+1} \tag{1}$$

(Krauthgamer and Lee, 2004). In addition, the size of any t-net is at most $2^{\mathsf{ddim}(A)+1}$ times the size of the minimal t-net, as the following easy lemma shows.

Lemma 1 (comparison of two nets) *Let $t > 0$ and suppose that M_1, M_2 are t-nets of $A \subseteq \mathcal{X}$. Then $|M_1| \leq 2^{\mathsf{ddim}(A)+1} |M_2|$.*

Proof Suppose that $|M_1| \geq k|M_2|$ for some positive integer k. Since $M_1 \subseteq \bigcup_{x \in M_2} \mathsf{ball}(x,t)$, it follows from the pigeonhole principle that at least one of the points in M_2 must cover at least k points in M_1. Thus, suppose that $x \in M_2$ covers the set $Z = \{z_1, \ldots, z_l\} \subseteq M_1$, meaning that $Z \subseteq \mathsf{ball}(x,t)$, where $l = |Z| \geq k$. By virtue of belonging to the t-net M_1, the set Z is t-separated. Therefore Z is a t-net of Z. Since Z is contained in a t-ball, we have $\mathsf{diam}(Z) \leq 2t$. It follows from Eq. (1) that $|Z| \leq 2^{\mathsf{ddim}(A)+1}$, whence the claim. ■

Throughout the paper, we fix a deterministic procedure for constructing a t-net, and denote its output for a multiset $U \subseteq \mathcal{X}$ by $\mathsf{Net}(U,t)$. Let $\mathsf{Par}(U,t)$ be a partition of $\mathcal{X}$ into regions induced by $\mathsf{Net}(U,t)$, that is: for $\mathsf{Net}(U,t) = \{x_1, \ldots, x_N\}$, define $\mathsf{Par}(U,t) := \{P_1, \ldots, P_N\}$, where

$$P_i = \{x \in \mathcal{X} \mid \kappa(x, \mathsf{Net}(U,t)) = i\}.$$

For $t > 0$, let $\mathcal{N}(t) := |\mathsf{Net}(U_{\text{in}}, t)|$ be the size of the t-net for the input sample.

2.3 Passive Compression-Based Nearest-Neighbors

Non-parametric binary classification admits performance guarantees that scale with the sample's noisy-margin (von Luxburg and Bousquet, 2004; Gottlieb et al., 2014a, 2017). The original margin-based methods of von Luxburg and Bousquet (2004) and Gottlieb et al. (2014a) analyzed the generalization performance via the technique of Lipschitz extension. Later, it was noticed in Gottlieb et al. (2014b) that the presence of a margin allows for compression — in fact, nearly optimally so.

We say that a labeled multiset S is (ν, t)-*separated*, for $\nu \in [0,1]$ and $t > 0$ (representing a margin t with noise ν), if one can remove a ν-fraction of the points in S, and in the resulting multiset, any pair of differently labeled points is separated by a distance of at least t. Formally, we have the following definition.

Definition 2 *S is (ν, t)-separated if there exists a subsample $\tilde{S} \subseteq S$ such that*

1. $|S \setminus \tilde{S}| \leq \nu |S|$ *and*
2. $\forall y_1 \neq y_2 \in \mathcal{Y}, a \in \tilde{S}^{y_1}, b \in \tilde{S}^{y_2}$, *we have* $\rho(a,b) \geq t$.

For a given labeled sample S, denote by $\nu(t)$ the smallest value ν such that S is (ν, t)-separated. Gottlieb et al. (2017) propose a passive learner with the following guarantees[2] as a function of the separation of S. Setting $\alpha := m/(m-N)$, define the following form of a generalization bound:

$$\mathrm{GB}(\epsilon, N, \delta, m, k) := \alpha\epsilon + \frac{2}{3}\frac{(N+1)\log(mk) + \log(\frac{1}{\delta})}{m-N} + \frac{3}{\sqrt{2}}\sqrt{\frac{\alpha\epsilon((N+1)\log(mk) + \log(\frac{1}{\delta}))}{m-N}}.$$

Further, for an integer m and $\delta \in (0,1)$, denote

$$G_{\min}(m, \delta) := \min_{t>0} \mathrm{GB}(\nu(t), \mathcal{N}(t), \delta, m, 1).$$

The quantity $G_{\min}(m,\delta)$ is small for datasets where we only need to remove few points to obtain a reasonably well separated subset. As an example, consider data generated by two well separated Gaussians (one generating the positively labeled points, and one generating the negatively labeled points). Then, most of the data points will be close to their respective means, but some will be farther, and may lie closer to the mean of the other Gaussian. Removing those few will result in a separated set.

Theorem 3 (Gottlieb et al. 2017) *Let m be an integer, $\delta \in (0,1)$. There exists a passive learning algorithm that returns a nearest-neighbor classifier $h^{\mathrm{nn}}_{S_{\mathrm{pas}}}$, where $S_{\mathrm{pas}} \subseteq S_{\mathrm{in}}$, such that, with probability $1-\delta$,*

$$\mathrm{err}(h^{\mathrm{nn}}_{S_{\mathrm{pas}}}, \mathcal{D}) \leq G_{\min}(m, \delta).$$

The bound above is data-dependent, meaning that the strength of the generalization guarantee depends on the quality of the random sample. Specifically, the passive algorithm of Gottlieb et al. (2017) generates S_{pas} of size approximately $\mathcal{N}(t)$ for the optimal scale $t > 0$ (found by searching over all scales), by removing the $|S_{\mathrm{in}}|\nu(t)$ points that obstruct the t-separation between different labels in S_{in}, and then selecting a subset of the remaining labeled examples to form S_{pas}, so that the examples are a t-net for S_{in} (not including the obstructing points). For the binary classification case ($|\mathcal{Y}| = 2$) an efficient algorithm is shown in Gottlieb et al. (2017). However, in the general multi-class case, it is not known how to find a minimal t-separation efficiently — a naive approach requires solving the NP-hard problem of vertex cover. Our approach, which we detail below, circumvents this issue, and provides an efficient algorithm also for the multiclass case.

2. The guarantees hold for the more general case of semimetrics.

3. Main Results

We propose a novel approach for generating a subset for a nearest-neighbor rule. This approach, detailed in the following sections, does not require finding and removing all the obstructing points in S_{in}, and can be implemented in an active setting using a small number of labels. The resulting active learning algorithm, MARMANN, has an error guarantee competitive with that of the passive learner, and a label complexity that can be significantly lower. We term the subset used by the nearest-neighbor rule a *compression set.*

Algorithm 1 MARMANN: MArgin Regularized Metric Active Nearest Neighbor

input Unlabeled sample U_{in} of size m, $\delta \in (0,1)$.
$\hat{t} \leftarrow \mathsf{SelectScale}(\delta)$. # SelectScale is given in Section 5, Alg. 4.
$\hat{S} \leftarrow \mathsf{GenerateNNSet}(\hat{t}, [\mathcal{N}(\hat{t})], \delta)$. # GenerateNNSet is given in Section 4, Alg. 2.
Output $h^{\mathrm{nn}}_{\hat{S}}$.

MARMANN, listed in Alg. 1, operates as follows. First, a scale $\hat{t} > 0$ is selected, by calling $\hat{t} \leftarrow \mathsf{SelectScale}(\delta)$, where SelectScale is our model selection procedure. SelectScale has access to U_{in}, and queries labels from S_{in} as necessary. It estimates the generalization error bound GB for several different scales, and executes a procedure similar to binary search to identify a good scale. The binary search keeps the number of estimations (and thus requested labels) small. Crucially, our estimation procedure is designed to prevent the search from spending a number of labels that depends on the net size of the smallest possible scale t, so that the total label complexity of MARMANN depends only on the error of the selected $\hat{t}$. Second, the selected scale $\hat{t}$ is used to generate the compression set by calling $\hat{S} \leftarrow \mathsf{GenerateNNSet}(\hat{t}, [\mathcal{N}(\hat{t})], \delta)$, where GenerateNNSet is our procedure for generating the compression set, and $[\mathcal{N}(\hat{t})] \equiv \{1, \ldots, \mathcal{N}(\hat{t})\}$. Our main result is the following guarantee for MARMANN.

Theorem 4 (Main result; Guarantee for MARMANN) *Let $S_{\mathrm{in}} \sim \mathcal{D}^m$, where $m \geq \max(6, |\mathcal{Y}|)$, $\delta \in (0, \frac{1}{4})$. Let $h^{\mathrm{nn}}_{\hat{S}}$ be the output of $\mathsf{MARMANN}(U_{\mathrm{in}}, \delta)$, where $\hat{S} \subseteq \mathcal{X} \times \mathcal{Y}$, and let $\hat{N} := |\hat{S}|$. Let $\hat{h} := h^{\mathrm{nn}}_{\hat{S}}$ and $\hat{\epsilon} := \mathrm{err}(\hat{h}, S_{\mathrm{in}})$, and denote $\hat{G} := \mathrm{GB}(\hat{\epsilon}, \hat{N}, \delta, m, 1)$. With a probability of $1-\delta$ over S_{in} and randomness of MARMANN,*

$$\mathrm{err}(\hat{h}, \mathcal{D}) \leq 2\hat{G} \leq O\left(G_{\min}(m, \delta)\right),$$

and the number of labels from S_{in} requested by MARMANN is at most

$$O\left(\log^3(\frac{m}{\delta})\left(\frac{1}{\hat{G}}\log(\frac{1}{\hat{G}}) + m\hat{G}\right)\right). \tag{2}$$

Here the $O(\cdot)$ notation hides only universal multiplicative constants.

Our error guarantee is thus a constant factor over the error guarantee of the passive learner of (Gottlieb et al., 2017), given in Theorem 3. The constant factor that we derive in our analysis is in the order of 2000 (this can be seen in the proof of Theorem 15). Note that we did not focus on optimizing it, opting instead for a more streamlined analysis. As the lower bound in Theorem 6 shows, the additive term $m\hat{G}$ in Eq. (2) is essentially unavoidable. Whether the dependence on $1/\hat{G}$ is indeed necessary is currently an open problem.

To observe the advantages of MARMANN over a passive learner, consider a scenario in which the upper bound $G_{\min}$ of Theorem 3, as well as the Bayes error of $\mathcal{D}$, are of order $\tilde{\Theta}(1/\sqrt{m})$. Then $\hat{G} = \Theta(1/\sqrt{m})$ as well. Therefore, MARMANN obtains a prediction error guarantee of $\tilde{\Theta}(1/\sqrt{m})$, similarly to the passive learner, but it uses only $\tilde{\Theta}(\sqrt{m})$ labels instead of m. In contrast, the following result shows that no learner that selects labels uniformly at random from S_{in} can compete with MARMANN: Theorem 5 below shows that for any passive learner that uses $\tilde{\Theta}(\sqrt{m})$ random labels from S_{in}, there exists a distribution $\mathcal{D}$ with the above properties, for which the prediction error of the passive learner in this case is $\tilde{\Omega}(m^{-1/4})$, a decay rate which is almost quadratically slower than the $\tilde{O}(1/\sqrt{m})$ rate achieved by MARMANN. Thus, the guarantees of MARMANN cannot be matched by any passive learner.

Theorem 5 (Passive lower bound) *Let $m > 0$ be an integer, and suppose that $(\mathcal{X}, \rho)$ is a metric space such that for some $\bar{t} > 0$, there is a $\bar{t}$-net T of $\mathcal{X}$ of size $\Theta(\sqrt{m})$. Consider any passive learning algorithm that maps i.i.d. samples $S_\ell \sim \mathcal{D}^\ell$ from some distribution $\mathcal{D}$ over $\mathcal{X} \times \{-1, 1\}$, to functions $\hat{h}_\ell : \mathcal{X} \to \{-1, 1\}$. For any such algorithm and any $\ell = \tilde{\Theta}(\sqrt{m})$, there exists a distribution $\mathcal{D}$ such that:*

i. The Bayes error of $\mathcal{D}$ is $\Theta(1/\sqrt{m})$;

ii. With at least a constant probability, both of the following events occur:

(a) The passive learner achieves error $\operatorname{err}(\hat{h}_\ell, \mathcal{D}) = \tilde{\Omega}(m^{-1/4})$,

(b) $G_{\min}(m, \delta) = \tilde{\Theta}(1/\sqrt{m})$.

Furthermore, i. and ii. continue to hold when the learning algorithm has access to the full marginal distribution over $\mathcal{X}$.

Thus, MARMANN even improves over a semi-supervised learner: its label savings stem from actively selecting labels, and are not achievable by merely exploiting information from unlabeled data or by randomly selecting examples to label.

We deduce Theorem 5 from a more general result, which might be of independent interest. Theorem 18, given in Section 7, improves existing passive learning sample complexity lower bounds. In particular, our result removes the restrictions of previous lower bounds on the relationship between the sample size, the VC-dimension, and the noise level, which render existing bounds inapplicable to our parameter regime. The proof of Theorem 5 is given thereafter in Section 7, as a consequence of Theorem 18.

We further provide a label complexity lower bound, in Theorem 6 below, which holds for any active learner that obtains similar guarantees to those of MARMANN. The lower bound shows that any active learning algorithm which guarantees a multiplicative accuracy over $G_{\min}(m, \delta)$ has a label complexity which is $\tilde{\Omega}(mG_{\min}(m, \delta))$, for a wide range of values of $G_{\min}(m, \delta)$ — essentially, as long as $G_{\min}(m, \delta)$ is not trivially large or trivially small. This implies that the term $m\hat{G}$ in the upper bound of the label complexity of MARMANN in Theorem 4 cannot be significantly improved.

Theorem 6 (Active lower bound) *Let $\mathcal{X} = \mathbb{R}$, $\delta \in (0, 1/14)$. Let $C \geq 1$, and let $\mathcal{A}$ be an active learning algorithm that outputs $\hat{h}$. Suppose that for any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, if the input unlabeled sample is of size m, then $\operatorname{err}(\hat{h}, \mathcal{D}) \leq CG_{\min}(m, \delta)$ with probability at least $1 - \delta$. Then for any $\alpha \in (\frac{\log(m)+\log(28)}{8\sqrt{2m}}, \frac{1}{240C})$ there exists a distribution $\mathcal{D}$ such with probability at least $\frac{1}{28}$ over $S \sim \mathcal{D}^m$ and the randomness of $\mathcal{A}$, both of the following events hold:*

1. *$\alpha \leq G_{\min}(m, \delta) \leq 30\alpha$*
2. *$\mathcal{A}$ queries at least $\frac{1}{2}\left\lfloor \frac{mG_{\min}(m,\delta) - \log(\frac{m}{\delta})}{30\log(m)} \right\rfloor \equiv \tilde{\Omega}(mG_{\min}(m, \delta))$ labels.*

The proof of this lower bound is provided in Section 8. In the rest of the paper, the components of MARMANN are described in detail, and the main results are proved.

4. Active Nearest-Neighbor at a Given Scale

A main challenge for active learning in our non-parametric setting is performing model selection, that is, selecting a good scale t similarly to the passive learner of Gottlieb et al. (2017). In the passive supervised setting, the approach developed in several previous works (Gottlieb et al., 2014b; Kontorovich and Weiss, 2014; Gottlieb et al., 2014a; Kontorovich and Weiss, 2015) performs model selection by solving a minimum vertex cover problem for each considered scale t, so as to eliminate all of the *t-blocking pairs* — i.e., pairs of differently labeled points within a distance t. The passive algorithm generates a compression set by first finding and removing from S_{in} all points that obstruct (ν, t)-separation at a given scale $t > 0$. This incurs a computational cost but no significant sample complexity increase, aside from the standard logarithmic factor that comes from stratifying over data-dependent hierarchies (Shawe-Taylor et al., 1998).

While this approach works for passive learning, in the active setting we face a crucial challenge: estimating the error of a nearest-neighbor rule at scale t using a small number of samples. A key insight that we exploit in this work is that instead of eliminating the blocking pairs, one may simply relabel some of the points in the compression set, and this would also generate a low-error nearest neighbor rule. This new approach enables estimation of the sample accuracy of a (possibly relabeled) t-net by label-efficient active sampling. In addition, this approach is significantly simpler than estimating the size of the minimum vertex cover of the t-blocking graph. Moreover, we gain improved algorithmic efficiency, by avoiding the relatively expensive vertex cover procedure.

A small technical difference, which will be evident below, is that in this new approach, examples in the compression set might have a different label than their original label in S_{in}. Standard sample compression analysis (e.g. Graepel et al., 2005) assumes that the classifier is determined by a small number of labeled examples from S_{in}. This does not allow the examples in the compression set to have a different label than their original label in S_{in}. Therefore, we require a slight generalization of previous compression analysis (following previous works on compression, see details in Section 6.1), which allows adding side information to the compression set. This side information will be used to set the label of each of the examples in the compression set. The generalization incurs a small statistical penalty, which we quantify in Section 6, as a preliminary to proving Theorem 4.

We now describe our approach to generating a compression set for a given scale $t > 0$. Recall that $\nu(t)$ is the smallest value for which S_{in} is (ν, t)-separated. We define two compression sets. The first one, denoted $S_{\text{a}}(t)$, represents an ideal compression set, constructed (solely for the sake of analysis) so that it induces an empirical error of at most $\nu(t)$. Calculating $S_{\text{a}}(t)$ might require many labels, thus it is only used for analysis purposes; the algorithm never constructs it. The second compression set, denoted $\hat{S}_{\text{a}}(t)$, represents an approximation to $S_{\text{a}}(t)$, which can be constructed using a small number of labels, and induces a sample error of at most $4\nu(t)$ with high probability.

We first define the ideal set $S_{\text{a}}(t) := \{(x_1, y_1), \ldots, (x_N, y_N)\}$. The examples in $S_{\text{a}}(t)$ are the points in $\mathsf{Net}(U_{\text{in}}, t/2)$, and the label of each example is the majority label, out of the labels of the examples in S_{in} to which x_i is closest. Formally, $\{x_1, \ldots, x_N\} := \mathsf{Net}(U_{\text{in}}, t/2)$, and for $i \in [N]$, $y_i := \operatorname{argmax}_{y \in \mathcal{Y}} |S^y \cap P_i|$, where $P_i = \{x \in \mathcal{X} \mid \kappa(x, \mathsf{Net}(U, t/2)) = i\} \in \mathsf{Par}(U_{\text{in}}, t/2)$.

Lemma 7 *Let S be a labeled sample of size m, and let $\{P_1, \ldots, P_N\}$ be a partition of $\mathbb{U}(S)$, with $\max_i \mathsf{diam}(P_i) \le t$ for some $t \ge 0$. For $i \in [N]$, let $\Lambda_i := S^{y_i} \cap P_i$. Then*

$$\nu(t) \ge 1 - \frac{1}{m} \sum_{i \in [N]} |\Lambda_i|.$$

Proof Let $\tilde{S} \subseteq S$ be a subsample that witnesses the $(\nu(t), t)$-separation of S, so that $|\tilde{S}| \ge m(1 - \nu(t))$, and for any two points $(x, y), (x', y') \in \tilde{S}$, if $\rho(x, x') \le t$ then $y = y'$. Denote $\tilde{U} := \mathbb{U}(\tilde{S})$. Since $\max_i \mathsf{diam}(P_i) \le t$, for any $i \in [N]$ all the points in $\tilde{U} \cap P_i$ must have the same label in $\tilde{S}$. Therefore,

$$\exists y \in \mathcal{Y} \text{ s.t. } \tilde{U} \cap P_i \subseteq \tilde{S}^y \cap P_i.$$

Hence $|\tilde{U} \cap P_i| \le |\Lambda_i|$. It follows

$$|S| - \sum_{i \in [N]} |\Lambda_i| \le |S| - \sum_{i \in [N]} |\tilde{U} \cap P_i| = |S| - |\tilde{S}| = m \cdot \nu(t).$$

Dividing by m we get the statement of the lemma. ■

From Lemma 7, we get Cor. 8, which upper bounds the empirical error of $h^{\text{nn}}_{S_{\text{a}}(t)}$ by $\nu(t)$.

Corollary 8 *For every $t > 0$, $\operatorname{err}(h^{\text{nn}}_{S_{\text{a}}(t)}, S_{\text{in}}) \le \nu(t)$.*

This corollary is immediate from Lemma 7, since for any $P_i \in \mathsf{Par}(U_{\text{in}}, t/2)$, $\mathsf{diam}(P_i) \le t$, and

$$\operatorname{err}(h^{\text{nn}}_{S_{\text{a}}(t)}, S_{\text{in}}) = 1 - \frac{1}{m} \sum_{i \in [N]} |\Lambda_i|.$$

Now, calculating $S_{\text{a}}(t)$ requires knowing most of the labels in S_{in}. MARMANN constructs instead an approximation $\hat{S}_{\text{a}}(t)$, in which the examples are the points in $\mathsf{Net}(U_{\text{in}}, t/2)$ (so that $\mathbb{U}(\hat{S}_{\text{a}}(t)) = \mathbb{U}(S_{\text{a}}(t))$), but the labels are determined using a bounded number of labels requested from S_{in}. The labels in $\hat{S}_{\text{a}}(t)$ are calculated by the simple procedure GenerateNNSet given in Alg. 2. The empirical error of the output of GenerateNNSet is bounded in Theorem 9 below.[3]

3. In the case of binary labels ($|\mathcal{Y}| = 2$), the problem of estimating $S_{\text{a}}(t)$ can be formulated as a special case of the benign noise setting for parametric active learning, for which tight lower and upper bounds are provided in Hanneke and Yang (2015). However, our case is both more general (as we allow multiclass labels) and more specific (as we are dealing with a specific "hypothesis class"). Thus we provide our own procedure and analysis.

A technicality in Alg. 2 requires explanation: In MARMANN, the generation of $\hat{S}_{\text{a}}(t)$ will be split into several calls to GenerateNNSet, so that different calls determine the labels of different points in $\hat{S}_{\text{a}}(t)$. Therefore GenerateNNSet has an additional argument I, which specifies the indices of the points in $\mathsf{Net}(U_{\text{in}}, t/2)$ for which the labels should be returned this time. Crucially, if during the run of MARMANN, GenerateNNSet is called again for the same scale t and the same point in $\mathsf{Net}(U_{\text{in}}, t/2)$, then GenerateNNSet returns the same label that it returned before, rather than recalculating it using fresh labels from S_{in}. This guarantees that despite the randomness in GenerateNNSet, the full $\hat{S}_{\text{a}}(t)$ is well-defined within any single run of MARMANN, and is distributed like the output of $\mathsf{GenerateNNSet}(t, [\mathcal{N}(t/2)], \delta)$, which is convenient for the analysis. Define

$$Q(m) := \lceil 18 \log(4m^3/\delta) \rceil . \tag{3}$$

Algorithm 2 $\mathsf{GenerateNNSet}(t, I, \delta)$

input Scale $t > 0$, a target set $I \subseteq [\mathcal{N}(t/2)]$, confidence $\delta \in (0, 1)$.
output A labeled set $S \subseteq \mathcal{X} \times \mathcal{Y}$ of size $|I|$
  $\{x_1, \ldots, x_N\} \leftarrow \mathsf{Net}(U_{\text{in}}, t/2)$, $\{P_1, \ldots, P_N\} \leftarrow \mathsf{Par}(U_{\text{in}}, t/2)$, $S \leftarrow ()$
  for $i \in I$ **do**
    if $\hat{y}_i$ has not already been calculated for U_{in} with this value of t **then**
      Draw $Q(m)$ points uniformly at random from P_i and query their labels.
      Let $\hat{y}_i$ be the majority label observed in these $Q(m)$ queries.
    end if
    $S \leftarrow S \cup \{(x_i, \hat{y}_i)\}$.
  end for
  Output S

Theorem 9 *Let $\hat{S}_{\text{a}}(t)$ be the output of $\mathsf{GenerateNNSet}(t, [\mathcal{N}(t/2)], \delta)$. With a probability at least $1 - \frac{\delta}{2m^2}$, the following event, which we denote by $E(t)$, holds:*

$$\operatorname{err}(h^{\text{nn}}_{\hat{S}_{\text{a}}(t)}, S_{\text{in}}) \le 4\nu(t).$$

Proof By Cor. 8, $\operatorname{err}(h^{\text{nn}}_{S_{\text{a}}(t)}, S_{\text{in}}) \le \nu(t)$. In $S_{\text{a}}(t)$, the labels assigned to each point in $\mathsf{Net}(U_{\text{in}}, t/2)$ are the majority labels (based on S_{in}) of the points in the regions in $\mathsf{Par}(U_{\text{in}}, t/2)$. As above, we denote the majority label for region P_i by $y_i := \operatorname{argmax}_{y \in \mathcal{Y}} |S^y \cap P_i|$. We now compare these labels to the labels $\hat{y}_i$ assigned by Alg. 2. Let $p(i) = |\Lambda_i|/|P_i|$ be the fraction of points in P_i which are labeled by the majority label y_i, where Λ_i is as defined in Lemma 7. Let $\hat{p}(i)$ be the fraction of labels equal to y_i out of those queried by Alg. 2 in round i. Let $\beta := 1/6$. By Hoeffding's inequality and union bounds, we have that with a probability of at least

$$1 - 2\mathcal{N}(t/2) \exp(-\frac{Q(m)}{18}) \ge 1 - \frac{\delta}{2m^2},$$

we have $\max_{i \in [\mathcal{N}(t/2)]} |\hat{p}(i) - p(i)| \le \beta$. Denote this "good" event by E'. We now prove that $E' \Rightarrow E(t)$. Let $J = \{i \in [\mathcal{N}(t/2)] \mid \hat{p}(i) > \frac{1}{2}\}$. It can be easily seen that $\hat{y}_i = y_i$ for all $i \in J$. Therefore, for all x such that $\kappa(x, \mathbb{U}(S_{\text{a}}(t))) \in J$, $h^{\text{nn}}_{\hat{S}_{\text{a}}(t)}(x) = h^{\text{nn}}_{S_{\text{a}}(t)}(x)$, and hence

$$\operatorname{err}(h^{\text{nn}}_{S'}, S_{\text{in}}) \le \mathbb{P}_{X \sim S_{\text{in}}}[\kappa(X, \mathbb{U}(S_{\text{a}}(t))) \notin J] + \operatorname{err}(h^{\text{nn}}_{S_{\text{a}}(t)}, U_{\text{in}}).$$

The second term is at most $\nu(t)$ by Cor. 8, and it remains to bound the first term, on the condition that E' holds. We have $\mathbb{P}_{X\sim U}[\kappa(X,\mathbb{U}(S_{\rm a}(t)))\notin J] = \frac{1}{m}\sum_{i\notin J}|P_i|$. If E' holds, then for any $i\notin J$, $p(i)\leq\frac{1}{2}+\beta$, therefore

$$|P_i|-|\Lambda_i| = (1-p(i))|P_i| \geq (\frac{1}{2}-\beta)|P_i|.$$

Recall that, by Lemma 7, $\nu(t)\geq 1-\frac{1}{m}\sum_{i\in[\mathcal{N}(t/2)]}|\Lambda_i|$. Therefore,

$$\begin{aligned}\nu(t) &\geq 1-\frac{1}{m}\sum_{i\in[\mathcal{N}(t/2)]}|\Lambda_i| \\ &= \frac{1}{m}\sum_{i\in[\mathcal{N}(t/2)]}(|P_i|-|\Lambda_i|) \\ &\geq \frac{1}{m}\sum_{i\notin J}(|P_i|-|\Lambda_i|) \\ &\geq \frac{1}{m}\sum_{i\notin J}(\frac{1}{2}-\beta)|P_i|.\end{aligned}$$

Thus, under E',

$$\mathbb{P}_{X\sim U}[\kappa(X,\mathbb{U}(S_{\rm a}(t)))\notin J] \leq \frac{\nu(t)}{\frac{1}{2}-\beta} = 3\nu(t).$$

It follows that under E', $\mathrm{err}(h^{\rm nn}_S, U_{\rm in})\leq 4\nu(t)$. ■

5. Model Selection

We now show how to select the scale $\hat{t}$ that will be used to generate the output nearest-neighbor rule. The main challenge is to do this with a low label complexity: Generating the full classification rule for scale t requires a number of labels that depends on $\mathcal{N}(t)$, which might be very large. We would like the label complexity of MARMANN to depend only on $\mathcal{N}(\hat{t})$ (where $\hat{t}$ is the selected scale), which is of the order $m\hat{G}$. Therefore, during model selection we can only invest a bounded number of labels in each tested scale. In addition, to keep the label complexity low, we would like to avoid testing all scales. In Section 5.1 we describe how we estimate the error on a given scale. In Section 5.2 we provide a search procedure, resembling binary search, which uses the estimation procedure to select a single scale $\hat{t}$.

5.1 Estimating the Error at a Given Scale

For $t>0$, let $\hat{S}_{\rm a}(t)$ be the compressed sample that MARMANN would generate if the selected scale were set to t. Our model selection procedure performs a search, similar to binary search, over the possible scales. For each tested scale t, the procedure estimates the empirical error $\epsilon(t) := \mathrm{err}(h^{\rm nn}_{\hat{S}_{\rm a}(t)}, S)$ within a certain accuracy, using an estimation procedure given below, called EstimateErr. EstimateErr outputs an estimate $\hat{\epsilon}(t)$ of $\epsilon(t)$, up to a given threshold $\theta>0$, using labels requested from $S_{\rm in}$.

To estimate the error, we sample random labeled examples from $S_{\rm in}$, and check the prediction error of $h^{\rm nn}_{\hat{S}_{\rm a}(t)}$ on these examples. The prediction error of any fixed hypothesis h on a random labeled example from $S_{\rm in}$ is an independent Bernoulli variable with expectation $\mathrm{err}(h, S_{\rm in})$. EstimateErr is implemented using the following procedure, EstBer, which adaptively estimates the expectation of a Bernoulli random variable to an accuracy specified by the parameter θ, using a small number of random independent Bernoulli experiments. Let $B_1, B_2, \ldots \in \{0,1\}$ be i.i.d. Bernoulli random variables. For an integer n, denote $\hat{p}_n = \frac{1}{n}\sum_{i=1}^n B_i$. The estimation procedure EstBer is given in Alg. 3. We prove a guarantee for this procedure in Theorem 10. Note that we assume that the threshold parameter is in $(0,1]$, since for $\theta\geq 1$ one can simply output 1 using zero random draws to satisfy Theorem 10.

Algorithm 3 EstBer(θ,β,δ)

input A threshold parameter $\theta\in(0,1]$, a budget parameter $\beta\geq 7$, confidence $\delta\in(0,1)$
  $S\leftarrow\{B_1,\ldots,B_4\}$
  $K\leftarrow\frac{4\beta}{\theta}\log(\frac{8\beta}{\delta\theta})$
  for $i=3:\lceil\log_2(\beta\log(2K/\delta)/\theta)\rceil$ **do**
    $n\leftarrow 2^i$
    $S\leftarrow S\cup\{B_{n/2+1},\ldots,B_n\}$.
    if $\hat{p}_n>\beta\log(2n/\delta)/n$ **then**
      break
    end if
  end for
  Output $\hat{p}_n$.

The following theorem states that Alg. 3 essentially estimates p, the expectation of the i.i.d. Bernoulli variables $B_1, B_2, \ldots$, up to a multiplicative constant, except if p is smaller than a value proportional to the threshold θ, in which case the algorithm simply returns a value at most θ. Moreover, the theorem shows that the number of random draws required by the algorithm is inversely proportional to the maximum of the threshold θ and the expectation p. Thus, if p is very small, the number of random draws does not increase without bound. The parameter β controls the trade-off between the accuracy of estimation and the number of random draws.

Theorem 10 *Let $\delta\in(0,1)$, $\theta\in(0,1]$, $\beta\geq 7$. Let $B_1, B_2, \ldots\in\{0,1\}$ be i.i.d Bernoulli random variables with expectation p. Let p_o be the output of* EstBer(θ,β,δ). *The following holds with a probability of $1-\delta$, where $f(\beta) := 1+\frac{8}{3\beta}+\sqrt{\frac{2}{\beta}}$.*

1. *If $p_o\leq\theta$, then $p\leq f(\beta)\theta$. Otherwise, $\frac{p}{f(\beta)}\leq p_o\leq\frac{p}{2-f(\beta)}$.*
2. *Let $\psi := \max(\theta, p/f(\beta))$. The number of random draws in* EstBer *is at most $\frac{4\beta\log(\frac{8\beta}{\delta\psi})}{\psi}$.*

Proof First, consider any single round i with $n=2^i$. By the empirical Bernstein bound (Maurer and Pontil, 2009, Theorem 4), with a probability of $1-\delta/n$, for $n\geq 8$, we have[4]

$$|\hat{p}_n - p| \leq \frac{8\log(2n/\delta)}{3n} + \sqrt{\frac{2\hat{p}_n\log(2n/\delta)}{n}}. \tag{4}$$

4. This follows from Theorem 4 of Maurer and Pontil (2009) since $\frac{7}{3(n-1)}\leq\frac{8}{3n}$ for $n\geq 8$.

Define $g := (\beta + 8/3 + \sqrt{2\beta})$, so that $f(\beta) = g/\beta$. Conditioned on Eq. (4), there are two cases:

(a) $\hat{p}_n \leq \beta \log(2n/\delta)/n$. In this case, $p \leq g \log(2n/\delta)/n$.

(b) $\hat{p}_n > \beta \log(2n/\delta)/n$. In this case, $n \geq \beta \log(2n/\delta)/\hat{p}_n$. Thus, by Eq. (4),

$$|\hat{p}_n - p| \leq \hat{p}_n(\frac{8}{3\beta} + \sqrt{2/\beta}) = \hat{p}_n(g/\beta - 1).$$

Therefore

$$\frac{\beta p}{g} \leq \hat{p}_n \leq \frac{p}{2 - g/\beta}.$$

Taking a union bound on all the rounds, we have that the guarantee holds for all rounds with a probability of at least $1 - \delta$.

Condition now on the event that these guarantees all hold. First, we prove the label complexity bound. Note that since $\beta \geq 7$, $K \geq 28$, thus we have $2\log(2K) > 8$, therefore there is always at least one round. Let n_o be the value of n in the last round that the algorithm runs, and let $p_o = \hat{p}_{n_o}$. Let i such that $n_o = 2^{i+1}$, thus the algorithm stops during round $i + 1$. This implies $\hat{p}_n \leq \beta \log(2n/\delta)/n$ for $n = 2^i$, therefore case (a) holds for n, which means $n \leq g \log(2n/\delta)/p$. It follows that $n \leq 2g \log(4g/\delta)/p$, therefore $n_o \leq 4g \log(4g/(\delta p))/p$. In addition, the number of random draws in the algorithm is n_0, which is bounded by

$$n_0 \leq 2^{\lceil \log_2(\beta \log(2K/\delta)/\theta) \rceil} \leq 2 \cdot 2^{\log_2(\beta \log(2K/\delta)/\theta)} \leq 2\beta \log(2K/\delta)/\theta.$$

Therefore we have the following bound on the number of random draws:

$$n_o \leq \min\left(\frac{2\beta \log(2K/\delta)}{\theta}, \frac{4g \log(4g/(\delta p))}{p}\right).$$

Plugging in the definition of K and substituting $\beta \cdot f(\beta)$ for g yields

$$\begin{aligned} n_0 \leq & \beta \min\left(\frac{2\log(\frac{8\beta}{\delta\theta}\log(\frac{8\beta}{\delta\theta}))}{\theta}, \frac{4f(\beta)\log(\frac{4\beta f(\beta)}{\delta p})}{p}\right) \leq \\ & \beta \min\left(\frac{4\log(\frac{8\beta}{\delta\theta})}{\theta}, \frac{4f(\beta)\log(\frac{4\beta f(\beta)}{\delta p})}{p}\right) \leq \\ & 4\beta \min\left(\frac{1}{\theta}(\log(\frac{8\beta}{\delta}) + \log(\frac{1}{\theta})), \frac{f(\beta)}{p}(\log(\frac{4\beta}{\delta}) + \log(\frac{f(\beta)}{p}))\right). \end{aligned}$$

Using the definition of ψ, we get that the number of draws is at most $\frac{4\beta \log(\frac{8\beta}{\delta\psi})}{\psi}$.

Next, we prove the accuracy of p_o (item 1 in the theorem statement) by considering two cases.

(I) If $p_o > \beta \log(2n_o/\delta)/n_o$, then case (b) above holds for n_o, thus

$$\frac{\beta p}{g} \leq p_o \leq \frac{p}{2 - g/\beta}.$$

In addition, if $p_o \leq \theta$, the LHS implies $p \leq f(\beta)\theta$. Thus item 1 in the theorem statement holds in this case.

(II) If $p_o \leq \beta \log(2n_o/\delta)/n_o$, then EstBer could not have ended by breaking out of the loop, thus it ran until the last round. Therefore $n_o \geq \beta \log(2K/\delta)/\theta$. In addition, case (a) holds for n_0, therefore

$$p \leq \frac{g \log(2n_o/\delta)}{n_o} \leq \frac{g\theta \log(2n_0/\delta)}{\beta \log(2K/\delta)}. \tag{5}$$

Now, for any possible value of n_o,

$$n_o \leq 2\beta \log(2K/\delta)/\theta \leq K.$$

The first inequality follows from the bound on i in EstBer, and the second inequality holds since, as defined in EstBer, $K \geq \frac{4\beta}{\theta}\log(\frac{8\beta}{\theta\delta})$. Since $n_0 \leq K$, Eq. (5) implies that

$$p \leq \frac{g\theta}{\beta} = f(\beta)\theta.$$

In addition, we have

$$p_o \leq \beta \log(2n_o/\delta)/n_o \leq \frac{\theta \log(2n_0/\delta)}{\log(2K/\delta)} \leq \theta.$$

Therefore in this case, necessarily $p_o \leq \theta$ and $p \leq f(\beta)\theta$, which satisfies item 1 in the theorem statement.

In both cases item 1 holds, thus the theorem is proved. ■

The procedure $\mathsf{EstimateErr}(t, \theta, \delta)$ is then implemented as follows:

- Call $\mathsf{EstBer}(\theta, 52, \delta/(2m^2))$, where the random variables B_i are independent copies of the Bernoulli variable
$$B := \mathbb{I}[h^{\mathrm{nn}}_{\hat{S}_{\mathrm{a}}(t)}(X) \neq Y]$$
and $(X, Y) \sim S_{\mathrm{in}}$.
- To draw a single B_i, sample a random pair (x', y') from S_{in}, set
$$i := \kappa(x', \mathsf{Net}(U_{\mathrm{in}}, t/2)),$$
and get $S \leftarrow \mathsf{GenerateNNSet}(t, \{i\}, \delta)$. This returns $S = ((x_i, \hat{y}_i))$ where $\hat{y}_i$ is the label of x_i in $\hat{S}_{\mathrm{a}}(t)$. Then $B_i := \mathbb{I}[\hat{y}_i \neq y']$. Note that B_i is indeed distributed like B, and $\mathbb{E}[B] = \epsilon(t)$. Note further that this call to $\mathsf{GenerateNNSet}(t, \{i\}, \delta)$ uses $Q(m)$ label queries. Therefore the overall label complexity of a single draw of a B_i is $Q(m) + 1$.

Cor. 11 gives a guarantee for the accuracy and label complexity of EstimateErr. The proof is immediate from Theorem 10, by setting $\beta = 52$, which implies $f(\beta) \leq 5/4$.

Corollary 11 *Let $t, \theta > 0$ and $\delta \in (0, 1)$, and let $\hat{\epsilon}(t) \leftarrow \mathsf{EstimateErr}(t, \theta, \delta)$. Let $Q(m)$ as defined in Eq. (3) The following properties hold with a probability of $1 - \frac{\delta}{2m^2}$ over the randomness of* EstimateErr *(and conditioned on $\hat{S}_{\mathrm{a}}(t)$).*

1. *If* $\hat{\epsilon}(t) \le \theta$, *then* $\epsilon(t) \le 5\theta/4$. *Otherwise,*

$$\frac{4\epsilon(t)}{5} \le \hat{\epsilon}(t) \le \frac{4\epsilon(t)}{3}.$$

2. *Let* $\psi' := \max(\theta, \epsilon(t))$. *The number of labels that* EstimateErr *requests is at most*

$$\frac{260(Q(m)+1)\log(\frac{1040m^2}{\delta\psi'})}{\psi'}.$$

To derive item 2. above from Theorem 10, note that for $\beta = 52$,

$$\psi' = \max(\theta, \epsilon(t)) \le f(\beta)\max(\theta, \epsilon(t)/f(\beta)) = f(\beta)\psi \le \frac{5}{4}\psi,$$

where ψ is as defined in Theorem 10. Below we denote the event that the two properties in Cor. 11 hold for t by $V(t)$.

5.2 Selecting a Scale

The model selection procedure SelectScale, given in Alg. 4, implements its search based on the guarantees in Cor. 11. First, we introduce some notation. We would like MARMANN to obtain a generalization guarantee that is competitive with $G_{\min}(m, \delta)$. Denote

$$\phi(t) := \frac{(\mathcal{N}(t)+1)\log(m) + \log(\frac{1}{\delta})}{m}, \tag{6}$$

and let

$$G(\epsilon, t) := \epsilon + \frac{2}{3}\phi(t) + \frac{3}{\sqrt{2}}\sqrt{\epsilon\phi(t)}.$$

Note that for all ϵ, t,

$$\mathrm{GB}(\epsilon, \mathcal{N}(t), \delta, m, 1) = \frac{m}{m - \mathcal{N}(t)}G(\epsilon, t).$$

When referring to $G(\nu(t), t), G(\epsilon(t), t)$, or $G(\hat{\epsilon}(t), t)$ we omit the second t for brevity.

Instead of directly optimizing $G(\nu(t))$, we will select a scale based on our estimate $G(\hat{\epsilon}(t))$ of $G(\epsilon(t))$. Let Dist denote the set of pairwise distances in the unlabeled dataset U_{in} (note that $|\text{Dist}| < \binom{m}{2}$). We remove from Dist some distances, so that the remaining distances have a net size $\mathcal{N}(t)$ that is monotone non-increasing in t. We also remove values with a very large net size. Concretely, define

$$\text{Dist}_{\text{mon}} := \text{Dist} \setminus \{t \mid \mathcal{N}(t) + 1 > m/2\} \setminus \{t \mid \exists t' \in \text{Dist}, t' < t \text{ and } \mathcal{N}(t') < \mathcal{N}(t)\}.$$

Then for all $t, t' \in \text{Dist}_{\text{mon}}$ such that $t' < t$, we have $\mathcal{N}(t') \ge \mathcal{N}(t)$. The output of SelectScale is always a value in Dist_{mon}. The following lemma shows that it suffices to consider these scales.

Lemma 12 *Assume* $m \ge 6$ *and let* $t^*_m \in \operatorname{argmin}_{t\in\text{Dist}} G(\nu(t))$. *If* $G_{\min}(m, \delta) \le 1/3$ *then* $t^*_m \in \text{Dist}_{\text{mon}}$.

Algorithm 4 SelectScale(δ)

input $\delta \in (0, 1)$
output Scale $\hat{t}$

$\mathcal{T} \leftarrow \text{Dist}_{\text{mon}}$, # $\mathcal{T}$ maintains the current set of possible scales
while $\mathcal{T} \neq \emptyset$ **do**
  $t \leftarrow$ the median value in $\mathcal{T}$ # break ties arbitrarily
  $\hat{\epsilon}(t) \leftarrow \mathsf{EstimateErr}(t, \phi(t), \delta)$.
  if $\hat{\epsilon}(t) < \phi(t)$ **then**
    $\mathcal{T} \leftarrow \mathcal{T} \setminus [0, t]$ # go right in the binary search
  else if $\hat{\epsilon}(t) > \frac{11}{10}\phi(t)$ **then**
    $\mathcal{T} \leftarrow \mathcal{T} \setminus [t, \infty)$ # go left in the binary search
  else
    $t_0 \leftarrow t, \mathcal{T}_0 \leftarrow \{t_0\}$.
    break from loop
  end if
end while
if $\mathcal{T}_0$ was not set yet **then**
  If the algorithm ever went to the right, let t_0 be the last value for which this happened, and let $\mathcal{T}_0 := \{t_0\}$. Otherwise, $\mathcal{T}_0 := \emptyset$.
end if
Let $\mathcal{T}_L$ be the set of all t that were tested and made the search go left
Output $\hat{t} := \operatorname{argmin}_{t\in\mathcal{T}_L\cup\mathcal{T}_0} G(\hat{\epsilon}(t))$

Proof Assume by way of contradiction that $t^*_m \in \text{Dist} \setminus \text{Dist}_{\text{mon}}$. First, since $G(\nu(t^*_m)) \le G_{\min}(m, \delta) \le 1/3$ we have

$$\frac{\mathcal{N}(t^*_m)+1}{m - \mathcal{N}(t^*_m)}\log(m) \le \frac{1}{2}.$$

Therefore, since $m \ge 6$, it is easy to verify $\mathcal{N}(t^*_m) + 1 \le m/2$. Therefore, by definition of Dist_{mon} there exists a $t \le t^*_m$ with $\phi(t) < \phi(t^*_m)$. Since $\nu(t)$ is monotone over all of $t \in \text{Dist}$, we also have $\nu(t) \le \nu(t^*_m)$. Now, $\phi(t) < \phi(t^*_m)$ and $\nu(t) \le \nu(t^*_m)$ together imply that $G(\nu(t)) < G(\nu(t^*_m))$, a contradiction. Hence, $t^*_m \in \text{Dist}_{\text{mon}}$. ■

SelectScale follows a search procedure similar to binary search, however the conditions for going right and for going left are not exhaustive, thus it is possible that neither condition holds. The search ends either when neither conditions hold, or when no additional scale should be tested. The final output of the algorithm is based on minimizing $G(\hat{\epsilon}(t))$ over some of the values tested during search.

For $c > 0$, define

$$\gamma(c) := 1 + \frac{2}{3c} + \frac{3}{\sqrt{2c}} \text{ and } \tilde{\gamma}(c) := \frac{1}{c} + \frac{2}{3} + \frac{3}{\sqrt{2c}}.$$

For all $t, \epsilon > 0$ we have the implications

$$\epsilon \ge c\phi(t) \;\Rightarrow\; \gamma(c)\epsilon \ge G(\epsilon, t) \quad \text{and} \quad \phi(t) \ge c\epsilon \;\Rightarrow\; \tilde{\gamma}(c)\phi(t) \ge G(\epsilon, t). \tag{7}$$

The following lemma uses Eq. (7) to show that the estimate $G(\hat{\epsilon}(t))$ is close to the true $G(\epsilon(t))$.

Lemma 13 *Let $t > 0$, $\delta \in (0,1)$, and suppose that* SelectScale *calls $\hat{\epsilon}(t) \leftarrow$ EstimateErr$(t, \phi(t), \delta)$. Suppose that $V(t)$ as defined in Cor. 11 holds. Then*

$$\frac{1}{6}G(\hat{\epsilon}(t)) \leq G(\epsilon(t)) \leq 6.5G(\hat{\epsilon}(t)).$$

Proof Under $V(t)$, we have that if $\hat{\epsilon}(t) < \phi(t)$ then $\epsilon(t) \leq \frac{5}{4}\phi(t)$. In this case,

$$G(\epsilon(t)) \leq \tilde{\gamma}(4/5)\phi(t) \leq 4.3\phi(t),$$

by Eq. (7). Therefore

$$G(\epsilon(t)) \leq \frac{3 \cdot 4.3}{2} G(\hat{\epsilon}(t)).$$

In addition, $G(\epsilon(t)) \geq \frac{2}{3}\phi(t)$ (from the definition of G), and by Eq. (7) and $\tilde{\gamma}(1) \leq 4$,

$$\phi(t) \geq \frac{1}{4}G(\hat{\epsilon}(t)).$$

Therefore $G(\epsilon(t)) \geq \frac{1}{6}G(\hat{\epsilon}(t))$. On the other hand, if $\hat{\epsilon}(t) \geq \phi(t)$, then by Cor. 11

$$\frac{4}{5}\epsilon(t) \leq \hat{\epsilon}(t) \leq \frac{4}{3}\epsilon(t).$$

Therefore $G(\hat{\epsilon}(t)) \leq \frac{4}{3}G(\epsilon(t))$ and $G(\epsilon(t)) \leq \frac{5}{4}G(\hat{\epsilon}(t))$. Taking the worst-case of both possibilities, we get the bounds in the lemma. ■

The next theorem bounds the label complexity of SelectScale. Let $\mathcal{T}_{\text{test}} \subseteq \text{Dist}_{\text{mon}}$ be the set of scales that are tested during SelectScale (that is, their $\hat{\epsilon}(t)$ was estimated).

Theorem 14 *Suppose that the event $V(t)$ defined in Cor. 11 holds for all $t \in \mathcal{T}_{\text{test}}$ for the calls $\hat{\epsilon}(t) \leftarrow$ EstimateErr$(t, \phi(t), \delta)$. If the output of* SelectScale *is $\hat{t}$, then the number of labels requested by* SelectScale *is at most*

$$9620|\mathcal{T}_{\text{test}}|(Q(m)+1)\frac{1}{G(\epsilon(\hat{t}))}\log(\frac{38480m^2}{\delta G(\epsilon(\hat{t}))}).$$

Proof The only labels used by the procedure are those used by calls to EstimateErr. Let $\psi_t := \max(\phi(t), \epsilon(t))$, and $\psi_{\min} := \min_{t \in \mathcal{T}_{\text{test}}} \psi_t$. Denote also $\hat{\psi}_t := \max(\phi(t), \hat{\epsilon}(t))$. From Cor. 11 we have that the total number of labels in all the calls to EstimateErr in SelectScale is at most

$$\sum_{t \in \mathcal{T}_{\text{test}}} \frac{260(Q(m)+1)\log(\frac{1040m^2}{\delta\psi_t})}{\psi_t} \leq |\mathcal{T}_{\text{test}}| \frac{260(Q(m)+1)\log(\frac{1040m^2}{\delta\psi_{\min}})}{\psi_{\min}}. \tag{8}$$

We now lower bound $\psi_{\min}$ using $G(\epsilon(\hat{t}))$. By Lemma 13 and the choice of $\hat{t}$,

$$G(\epsilon(\hat{t})) \leq 6.5G(\hat{\epsilon}(\hat{t})) = 6.5 \min_{t \in \mathcal{T}_L \cup T_0} G(\hat{\epsilon}(t)).$$

From the definition of G, for any $t > 0$,

$$G(\hat{\epsilon}(t)) \leq \gamma(1)\max(\phi(t), \hat{\epsilon}(t)) \leq 25\hat{\psi}_t.$$

Therefore

$$G(\epsilon(\hat{t})) \leq 25 \min_{t \in \mathcal{T}_L \cup \mathcal{T}_0} \hat{\psi}_t. \tag{9}$$

We will show a similar upper bound when minimizing over all of $\mathcal{T}_{\text{test}}$, not just over $\mathcal{T}_L \cup \mathcal{T}_0$. This is trivial if $\mathcal{T}_{\text{test}} = \mathcal{T}_L \cup \mathcal{T}_0$. Consider the case $\mathcal{T}_L \cup \mathcal{T}_0 \subsetneq \mathcal{T}_{\text{test}}$. For any $t \in \mathcal{T}_{\text{test}}$, we have one of:

- The search went left on t (step 8), hence $t \in \mathcal{T}_L$.
- The search went nowhere on t and the loop broke (step 11), hence $t = t_0 \in \mathcal{T}_0$.
- The search went right on t (step 6) and this was the last value for which this happened, hence $t = t_0 \in \mathcal{T}_0$.
- The search went right on t (step 6) and this was *not* the last value for which this happened. Hence $t \in \mathcal{T}_{\text{test}} \setminus (\mathcal{T}_L \cup \mathcal{T}_0)$.

Set some $t_1 \in \mathcal{T}_{\text{test}} \setminus (\mathcal{T}_L \cup \mathcal{T}_0)$. Since the search went right on t_1, then t_0 also exists, since the algorithm did go to the right for some t (see step 15). Since the binary search went right on t_1, we have $\hat{\epsilon}(t_1) \leq \phi(t_1)$. Since the binary search did *not* go left on t_0 (it either broke from the loop or went right), $\hat{\epsilon}(t_0) \leq \frac{11}{10}\phi(t_0)$.

In addition, $t_0 \geq t_1$ (since the search went right at t_1, and t_0 was tested later than t_1), thus $\phi(t_0) \leq \phi(t_1)$ (since $t_0, t_1 \in \text{Dist}_{\text{mon}}$). Therefore,

$$\hat{\psi}_{t_0} = \max(\phi(t_0), \hat{\epsilon}(t_0)) \leq \frac{11}{10}\phi(t_0) \leq \frac{11}{10}\phi(t_1) = \frac{11}{10}\max(\phi(t_1), \hat{\epsilon}(t_1)) = \hat{\psi}_{t_1}.$$

It follows that for any such t_1,

$$\min_{t \in \mathcal{T}_L \cup \mathcal{T}_0} \hat{\psi}_t \leq \frac{11}{10}\hat{\psi}_{t_1}.$$

Therefore

$$\min_{t \in \mathcal{T}_L \cup \mathcal{T}_0} \hat{\psi}_t \leq \frac{11}{10} \min_{t \in \mathcal{T}_{\text{test}}} \hat{\psi}_t.$$

Therefore, by Eq. (9)

$$G(\epsilon(\hat{t})) \leq 27.5 \min_{t \in \mathcal{T}_{\text{test}}} \hat{\psi}_t.$$

By Cor. 11, $\hat{\epsilon}(t) \leq \max(\phi(t), 4\epsilon(t)/3)$, therefore $\hat{\psi}_t \leq \frac{4}{3}\psi_t$. Therefore $G(\epsilon(\hat{t})) \leq 37\psi_{\min}$. Therefore, from Eq. (8), the total number of labels is at most

$$9620|\mathcal{T}_{\text{test}}|(Q(m)+1)\frac{1}{G(\epsilon(\hat{t}))}\log(\frac{38480m^2}{\delta G(\epsilon(\hat{t}))}).$$

■

The following theorem provides a competitive error guarantee for the selected scale $\hat{t}$.

Theorem 15 *Suppose that $V(t)$ and $E(t)$, defined in Cor. 11 and Theorem 9, hold for all values $t \in \mathcal{T}_{\text{test}}$, and that $G_{\min}(m,\delta) \le 1/3$. Then* SelectScale *outputs $\hat{t} \in \text{Dist}_{\text{mon}}$ such that*

$$\text{GB}(\epsilon(\hat{t}), \mathcal{N}(\hat{t}), \delta, m, 1) \le O(G_{\min}(m,\delta)),$$

Where the $O(\cdot)$ notation hides only universal multiplicative constants.

The full proof of this theorem is given below. The idea of the proof is as follows: First, we show (using Lemma 13) that it suffices to prove that $G(\nu(t_m^*)) \ge O(G(\hat{\epsilon}(\hat{t})))$ to derive the bound in the theorem. Now, SelectScale ends in one of two cases: either $\mathcal{T}_0$ is set within the loop, or $\mathcal{T} = \emptyset$ and $\mathcal{T}_0$ is set outside the loop. In the first case, neither of the conditions for turning left and turning right holds for t_0, so we have $\hat{\epsilon}(t_0) = \Theta(\phi(t_0))$ (where Θ hides numerical constants). We show that in this case, whether $t_m^* \ge t_0$ or $t_m^* \le t_0$, $G(\nu(t_m^*)) \ge O(G(\hat{\epsilon}(t_0)))$. In the second case, there exist (except for edge cases, which are also handled) two values $t_0 \in \mathcal{T}_0$ and $t_1 \in \mathcal{T}_L$ such that t_0 caused the binary search to go right, and t_1 caused it to go left, and also $t_0 \le t_1$, and $(t_0, t_1) \cap \text{Dist}_{\text{mon}} = \emptyset$. We use these facts to show that for $t_m^* \ge t_1$, $G(\nu(t_m^*)) \ge O(G(\hat{\epsilon}(t_1)))$, and for $t_m^* \le t_0$, $G(\nu(t_m^*)) \ge O(G(\hat{\epsilon}(t_0)))$. Since $\hat{t}$ minimizes over a set that includes t_0 and t_1, this gives $G(\nu(t_m^*)) \ge O(G(\hat{\epsilon}(\hat{t})))$ in all cases.

Proof First, note that it suffices to show that there is a constant C, such that for the output $\hat{t}$ of SelectScale, we have $G(\epsilon(\hat{t})) \le CG(\nu(t_m^*))$. This is because of the following argument: From Lemma 12 we have that if $G_{\min}(m,\delta) \le 1/3$, then $t_m^* \in \text{Dist}_{\text{mon}}$. Now

$$G_{\min}(m,\delta) = \frac{m}{m - \mathcal{N}(t_m^*)} G(\nu(t_m^*)) \ge G(\nu(t_m^*)).$$

And, if we have the guarantee on $G(\epsilon(\hat{t}))$ and $G_{\min}(m,\delta) \le 1/3$ we will have

$$\text{GB}(\epsilon(\hat{t}), \mathcal{N}(\hat{t}), \delta, m, 1) = \frac{m}{m - \mathcal{N}(\hat{t})} G(\epsilon(\hat{t})) \le 2G(\epsilon(\hat{t})) \le 2CG(\nu(t_m^*)) \le 2CG_{\min}(m,\delta). \tag{10}$$

We now prove the existence of such a guarantee and set C. Denote the two conditions checked in SelectScale during the binary search by Condition 1: $\hat{\epsilon}(t) < \phi(t)$ and Condition 2: $\hat{\epsilon}(t) > \frac{11}{10}\phi(t)$. The procedure ends in one of two ways: either $\mathcal{T}_0$ is set within the loop (Case 1), or $\mathcal{T} = \emptyset$ and $\mathcal{T}_0$ is set outside the loop (Case 2). We analyze each case separately.

In Case 1, none of the conditions 1 and 2 hold for t_0. Therefore

$$\phi(t_0) \le \hat{\epsilon}(t_0) \le \frac{11}{10}\phi(t_0).$$

Therefore, by Eq. (7),

$$\phi(t_0) \ge G(\hat{\epsilon}(t_0))/\tilde{\gamma}(\frac{10}{11}).$$

By Cor. 11, since $\hat{\epsilon}(t_0) > \phi(t_0)$,

$$\frac{3}{4}\phi(t_0) \le \frac{3}{4}\hat{\epsilon}(t_0) \le \epsilon(t_0) \le \frac{5}{4}\hat{\epsilon}(t_0) \le \frac{55}{40}\phi(t_0).$$

Suppose $t_m^* \ge t_0$, then

$$G(\nu(t_m^*)) \ge \nu(t_m^*) \ge \nu(t_0) \ge \frac{1}{4}\epsilon(t_0) \ge \frac{3}{16}\phi(t_0).$$

here we used $\epsilon(t_0) \le 4\nu(t_0)$ by Theorem 9. Therefore, from Eq. (7) and Lemma 13,

$$G(\nu(t_m^*)) \ge \frac{3}{16}\phi(t_0) \ge \frac{3}{16\tilde{\gamma}\left(\frac{40}{55}\right)} G(\epsilon(t_0)) \ge \frac{\frac{1}{2}}{16\tilde{\gamma}(\frac{40}{55})} G(\hat{\epsilon}(t_0)).$$

Now, suppose $t_m^* < t_0$, then

$$G(\nu(t_m^*)) \ge \frac{2}{3}\phi(t_m^*) \ge \frac{2}{3}\phi(t_0) \ge \frac{2}{3\tilde{\gamma}(\frac{10}{11})} G(\hat{\epsilon}(t_0)).$$

In this inequality we used the fact that $t_m^*, t_0 \in \text{Dist}_{\text{mon}}$, hence $\phi(t_m^*) \ge \phi(t_0)$. Combining the two possibilities for t_m^*, we have in Case 1,

$$G(\hat{\epsilon}(t_0)) \le \max(32\tilde{\gamma}(\frac{40}{55}), \frac{3\tilde{\gamma}(\frac{10}{11})}{2}) G(\nu(t_m^*)).$$

Since $\hat{t}$ minimizes $G(\hat{\epsilon}(t))$ on a set that includes t_0, we have, using Lemma 13

$$G(\epsilon(\hat{t})) \le 6.5 G(\hat{\epsilon}(\hat{t})) \le 6.5 G(\hat{\epsilon}(t_0)).$$

Therefore, in Case 1,

$$G(\epsilon(\hat{t})) \le 6.5 \max(32\tilde{\gamma}(\frac{40}{55}), \frac{3\tilde{\gamma}(\frac{10}{11})}{2}) G(\nu(t_m^*)). \tag{11}$$

In Case 2, the binary search halted without satisfying Condition 1 nor Condition 2 and with $\mathcal{T} = \emptyset$. Let t_0 be as defined in this case in SelectScale (if it exists), and let t_1 be the smallest value in $\mathcal{T}_L$ (if it exists). At least one of these values must exist. If both values exist, we have $t_0 \le t_1$ and $(t_0, t_1) \cap \text{Dist}_{\text{mon}} = \emptyset$.

If t_0 exists, it is the last value for which the search went right. We thus have $\hat{\epsilon}(t_0) < \phi(t_0)$. If $t_m^* \le t_0$, from condition 1 on t_0 and Eq. (7) with $\tilde{\gamma}(1) \le 4$,

$$G(\nu(t_m^*)) \ge \frac{2}{3}\phi(t_m^*) \ge \frac{2}{3}\phi(t_0) \ge \frac{1}{6} G(\hat{\epsilon}(t_0)).$$

Here we used the monotonicity of ϕ on $t_m^*, t_0 \in \text{Dist}_{\text{mon}}$, and Eq. (7) applied to condition 1 for t_0.

If t_1 exists, the search went left on t_1, thus $\hat{\epsilon}(t_1) > \frac{11}{10}\phi(t_1)$. By Cor. 11, it follows that $\hat{\epsilon}(t_1) \le \frac{4}{3}\epsilon(t_1)$. Therefore, if $t_m^* \ge t_1$,

$$G(\nu(t_m^*)) \ge \nu(t_m^*) \ge \nu(t_1) \ge \frac{1}{4}\epsilon(t_1) \ge \frac{3}{16}\hat{\epsilon}(t_1) \ge \frac{3}{16\gamma(11/10)} G(\hat{\epsilon}(t_1)).$$

Here we used $\epsilon(t_1) \le 4\nu(t_1)$ by Theorem 9 and Eq. (7). Combining the two cases for t_m^*, we get that if t_0 exists and $t_m^* \le t_0$, or t_1 exists and $t_m^* \ge t_1$,

$$G(\nu(t_m^*)) \ge \min(\frac{1}{6}, \frac{3}{16\gamma(11/10)}) \min_{t \in T_E} G(\hat{\epsilon}(t)).$$

where we define $T_E = \{t \in \{t_0, t_1\} \mid t \text{ exists}\}$. We now show that this covers all possible values for t_m^*: If both t_0, t_1 exist, then since $(t_0, t_1) \cap \text{Dist}_{\text{mon}} = \emptyset$, it is impossible to have $t_m^* \in (t_0, t_1)$.

If only t_0 exists, then the search never went left, which means $t_0 = \max(\mathrm{Dist}_{\mathrm{mon}})$, thus $t^*_m \le t_0$. If only t_1 exists, then the search never went right, which means $t_1 = \min(\mathrm{Dist}_{\mathrm{mon}})$, thus $t^*_m \ge t_1$.

Since $\hat{t}$ minimizes $G(\hat{\epsilon}(t))$ on a set that has T_E as a subset, we have, using Lemma 13 $G(\epsilon(\hat{t})) \le 6.5G(\hat{\epsilon}(\hat{t})) \le 6.5 \min_{t\in T_E} G(\hat{\epsilon}(t))$. Therefore in Case 2,

$$G(\nu(t^*_m)) \ge \frac{1}{6.5} \min\left(\frac{1}{6}, \frac{3}{16\gamma(11/10)}\right) G(\epsilon(\hat{t}). \tag{12}$$

From Eq. (11) and Eq. (12) we get that in both cases

$$G(\nu(t^*_m)) \ge \frac{1}{6.5} \min\left(\frac{1}{6}, \frac{3}{16\gamma(11/10)}, \frac{2}{3\tilde{\gamma}(10/11)}, \frac{1}{32\tilde{\gamma}(\frac{40}{55})}\right) G(\epsilon(\hat{t})) \ge G(\epsilon(\hat{t}))/865.$$

Combining this with Eq. (10) we get the statement of the theorem. ■

6. Bounding the Label Complexity of MARMANN

We are now almost ready to prove Theorem 4. Our last missing piece is quantifying the effect of side information on the generalization of sample compression schemes in Section 6.1. We then prove Theorem 4 in Section 6.2.

6.1 Sample Compression with Side Information

It appears that compression-based generalization bounds were independently discovered by Littlestone and Warmuth (1986) and Devroye et al. (1996); some background is given in Floyd and Warmuth (1995). As noted in Section 4, our algorithm relies on a generalized sample compression scheme, which requires side information. This side information is used to represent the labels of the sample points in the compression set. A similar idea appears in Floyd and Warmuth (1995) for hypotheses with short description length. Here we provide a generalization that is useful for the analysis of MARMANN.

Let Σ be a finite alphabet, and define a mapping $\mathsf{Rec}_N : (\mathcal{X} \times \mathcal{Y})^N \times \Sigma^N \to \mathcal{Y}^{\mathcal{X}}$.[5] This is a *reconstruction* function mapping a labeled sequence of size N with side information $T \in \Sigma^N$ to a classifier. For $I \subseteq [|S|]$, denote by $S[I]$ the subsequence of S indexed by I. For a labeled sample S, define the set of possible hypotheses reconstructed from a compression of S of size N with side information in Σ: $\mathcal{H}_N(S) := \left\{h : \mathcal{X} \to \mathcal{Y} \mid h = \mathsf{Rec}_N(S[I], T), I \in [m]^N, T \in \Sigma^N\right\}$. The following result closely follows the sample compression arguments in Graepel et al. (2005, Theorem 2), and Gottlieb et al. (2017, Theorem 6), but incorporates side information.

Theorem 16 *Let m be an integer and $\delta \in (0,1)$. Let $S \sim \mathcal{D}^m$. With probability at least $1-\delta$, if there exist $N < m$ and $h \in \mathcal{H}_N(S)$ with $\epsilon := \mathrm{err}(h,S) \le \frac{1}{2}$, then $\mathrm{err}(h,\mathcal{D}) \le \mathrm{GB}(\epsilon, N, \delta, m, |\Sigma|)$.*

Proof We recall a result of Dasgupta and Hsu (2008, Lemma 1): if $\hat{p} \sim \mathrm{Bin}(n,p)/n$ and $\delta > 0$, then the following holds with probability at least $1-\delta$:

$$p \le \hat{p} + \frac{2}{3n}\log\frac{1}{\delta} + \sqrt{\frac{9\hat{p}(1-\hat{p})}{2n}\log\frac{1}{\delta}}. \tag{13}$$

5. If $\mathcal{X}$ is infinite, replace $\mathcal{Y}^{\mathcal{X}}$ with the set of measurable functions from $\mathcal{X}$ to $\mathcal{Y}$.

Now fix $N < m$, and suppose that $h \in \mathcal{H}_{\mathcal{N}}(S)$ has $\hat{\epsilon} \le \frac{1}{2}$. Let $I \in [m]^N, T \in \Sigma^N$ such that $h = \mathsf{Rec}_N(S[I], T)$. We have $\mathrm{err}(h, S[[m] \setminus I]) \le \frac{\hat{\epsilon} m}{m-N} = \theta\hat{\epsilon}$. Substituting into (13) $p := \mathrm{err}(h,\mathcal{D})$, $n := m-N$ and $\hat{p} := \mathrm{err}(h, S[[m]\setminus I]) \le \theta\hat{\epsilon}$, yields that for a fixed $S[I]$ and a random $S[[m]\setminus I] \sim \mathcal{D}^{m-N}$, with probability at least $1-\delta$,

$$\mathrm{err}(h,\mathcal{D}) \le \theta\hat{\epsilon} + \frac{2}{3(m-N)}\log\frac{1}{\delta} + \sqrt{\frac{9\theta\hat{\epsilon}}{2(m-N)}\log\frac{1}{\delta}}. \tag{14}$$

To make (14) hold simultaneously for all $(I,T) \in [m]^N \times \Sigma^N$, divide δ by $(m|\Sigma|)^N$. To make the claim hold for all $N \in [m]$, stratify (as in Graepel et al. (2005, Lemma 1)) over the (fewer than) m possible choices of N, which amounts to dividing δ by an additional factor of m. ■

For MARMANN, we use the following sample compression scheme with $\Sigma = \mathcal{Y}$. Given a subsequence $S' := S[I] := (x'_1, \ldots, x'_N)$ and $T = (t_1, \ldots, t_N) \in \mathcal{Y}^N$, the reconstruction function $\mathsf{Rec}_N(S[I], T)$ generates the nearest-neighbor rule induced by the labeled sample $\psi(S', T) := ((x'_i, t_i))_{i\in[N]}$. Formally, $\mathsf{Rec}_N(S', T) = h^{\mathrm{nn}}_{\psi(S',T)}$. Note the slight abuse of notation: formally, the y_i in $S_{\mathrm{a}}(t)$ should be encoded as side information T, but for clarity, we have opted to "relabel" the examples $\{x_1, \ldots, x_N\}$ as dictated by the majority in each region. The following corollary is immediate from Theorem 16 and the construction above.

Theorem 17 *Let $m \ge |\mathcal{Y}|$ be an integer, $\delta \in (0, \frac{1}{4})$. Let $S_{\mathrm{in}} \sim \mathcal{D}^m$. With probability at least $1-\delta$, if there exist $N < m$ and $S \subseteq (\mathcal{X}\times\mathcal{Y})^N$ such that $\mathbb{U}(S) \subseteq U_{\mathrm{in}}$ and $\epsilon := \mathrm{err}(h^{\mathrm{nn}}_S, S_{\mathrm{in}}) \le \frac{1}{2}$, then $\mathrm{err}(h^{\mathrm{nn}}_S, \mathcal{D}) \le \mathrm{GB}(\epsilon, N, \delta, m, |\mathcal{Y}|) \le 2\mathrm{GB}(\epsilon, N, 2\delta, m, 1)$.*

If the compression set includes only the original labels, the compression analysis of Gottlieb et al. (2017) gives the bound $\mathrm{GB}(\epsilon, N, \delta, m, 1)$. Thus the effect of allowing the labels to change is only logarithmic in $|\mathcal{Y}|$, and does not appreciably degrade the prediction error.

6.2 Proof of Theorem 4

The proof of the main theorem, Theorem 4, which gives the guarantee for MARMANN, is almost immediate from Theorem 17, Theorem 9, Theorem 15 and Theorem 14.

Proof [of Theorem 4] We have $|\mathrm{Dist}_{\mathrm{mon}}| \le \binom{m}{2}$. By a union bound, the events $E(t)$ and $V(t)$ of Theorem 9 and Cor. 11 hold for all $t \in \mathcal{T}_{\mathrm{test}} \subseteq \mathrm{Dist}_{\mathrm{mon}}$ with a probability of at least $1-\delta/2$. Under these events, we have by Theorem 15 that if $G_{\min}(m,\delta) \le 1/3$,

$$\mathrm{GB}(\epsilon(\hat{t}), \mathcal{N}(\hat{t}), \delta, m, 1) \le O\left(\min_t \mathrm{GB}(\nu(t), \mathcal{N}(t), \delta, m, 1)\right).$$

By Theorem 17, with a probability at least $1-\delta/2$, if $\epsilon(\hat{t}) \le \frac{1}{2}$ then

$$\mathrm{err}(\hat{h}, \mathcal{D}) \le 2\mathrm{GB}(\epsilon(\hat{t}), \mathcal{N}(\hat{t}), \delta, m, 1).$$

The statement of the theorem follows. Note that the statement trivially holds for $G_{\min}(m,\delta) \ge 1/3$ and for $\epsilon(\hat{t}) \ge \frac{1}{2}$, thus these conditions can be removed. To bound the label complexity, note that the total number of labels used by MARMANN is at most the number of labels used by SelectScale plus the number of labels used by GenerateNNSet when the final compression set is generated.

By Theorem 14, since $Q(m) = O(\log(m/\delta))$, the number of labels used by SelectScale is at most

$$O\left(|\mathcal{T}_{\text{test}}|\frac{\log^2(m/\delta)}{G(\epsilon(\hat{t}))}\log\left(\frac{1}{G(\epsilon(\hat{t}))}\right)\right).$$

In addition,

$$G(\epsilon(\hat{t})) \geq \text{GB}(\epsilon(\hat{t}), \mathcal{N}(\hat{t}), \delta, m, 1) = \hat{G}.$$

The number of tested scales in SelectScale is bounded by

$$|\mathcal{T}_{\text{test}}| \leq \lfloor \log_2(|\text{Dist}_{\text{mon}}|) + 1 \rfloor \leq 2\log_2(m)$$

Therefore the number of labels used by SelectScale is

$$O\left(\frac{\log^3(m/\delta)}{\hat{G}}\log\left(\frac{1}{\hat{G}}\right)\right).$$

The number of labels used by GenerateNNSet is at most $Q(m)\mathcal{N}(\hat{t})$, where $Q(m) \leq O(\log(m/\delta)$, and from the definition of $\hat{G}$, $\mathcal{N}(\hat{t}) \leq O(m\hat{G}/\log(m))$. Summing up the number of labels used by SelectScale and the number used by GenerateNNSet, this gives the bound in the statement of the theorem. ■

7. Passive Learning Lower Bounds

Theorem 5 lower bounds the performance of a passive learner who observes a limited number ℓ of random labels from S_{in}. The number ℓ is chosen so that it is of the same order as the number of labels MARMANN observes for the case analyzed in Section 3. We deduce Theorem 5 from a more general result pertaining to the sample complexity of passive learning. The general result is given as Theorem 18 in Section 7.1. The proof of Theorem 5 is provided in Section 7.2.

We note that while the lower bounds below assume that the passive learner observes only the random labeled sample of size ℓ, in fact their proofs hold also if the algorithm has access to the full unlabeled sample of size m of which S_ℓ is sampled. This is because the lower bound is based on requiring the learner to distinguish between distributions that all have the same marginal. Under this scenario, access to unlabeled examples does not provide any additional information to the learner.

7.1 A General Lower Bound

In this section we show a general sample complexity lower bound for passive learning, which may be of independent interest. We are aware of two existing lower bounds for agnostic PAC with bounded Bayes error: Devroye et al. (1996, Theorem 14.5) and Audibert (2009, Theorem 8.8). Both place restrictions on the relationship between the sample size, VC-dimension, and Bayes error level, which render them inapplicable as stated to some parameter regimes, including the one needed for proving Theorem 5.

Let $\mathcal{H}$ be a hypothesis class with VC-dimension d and suppose that $\mathcal{L}$ is a passive learner[6] mapping labeled samples $S_\ell = (X_i, Y_i)_{i\in[\ell]}$ to hypotheses $\hat{h}_\ell \in \mathcal{H}$. For any distribution $\mathcal{D}$ over

6. We allow $\mathcal{L}$ access to an independent internal source of randomness.

$\mathcal{X} \times \{-1, 1\}$, define the *excess risk* of $\hat{h}_\ell$ by

$$\Delta(\hat{h}_\ell, \mathcal{D}) := \text{err}(\hat{h}_\ell, \mathcal{D}) - \inf_{h\in\mathcal{H}} \text{err}(h, \mathcal{D}).$$

Let $\mathscr{D}(\eta)$ be the collection of all *η-bounded agnostic error* distributions $\mathcal{D}$ over $\mathcal{X}\times\{-1,1\}$ that satisfy $\inf_{h\in\mathcal{H}} \text{err}(h, \mathcal{D}) \leq \eta$. We say that $Z \in \{-1, 1\}$ has Rademacher distribution with parameter $b \in [-1, 1]$, denoted $Z \sim \text{R}_b$, if

$$\mathbb{P}[Z = 1] = 1 - \mathbb{P}[Z = -1] = \frac{1}{2} + \frac{b}{2}.$$

All distributions on $\{-1, 1\}$ are of this form. For $k \in \mathbb{N}$ and $b \in [0, 1]$, define the function

$$\text{bayes}(k, b) = \tfrac{1}{2}\left(1 - \tfrac{1}{2}\left\|\text{R}_b^k - \text{R}_{-b}^k\right\|_1\right),$$

where $\text{R}_{\pm b}^k$ is the corresponding product distribution on $\{-1, 1\}^k$ and $\frac{1}{2}\|\cdot\|_1$ is the total variation norm. This expression previously appeared in Berend and Kontorovich (2015, Equation (25)) in the context of information-theoretic lower bounds; the current terminology was motivated in Kontorovich and Pinelis (2016), where various precise estimates on $\text{bayes}(\cdot)$ were provided. In particular, the function $\text{baÿes}(\kappa, b)$ was defined as follows: for each fixed $b \in [0, 1]$, $\text{baÿes}(\cdot, b)$ is the largest convex minorant on $[0, \infty)$ of the function $\text{bayes}(\cdot, b)$ on $\{0, 1, \dots\}$. It was shown in Kontorovich and Pinelis (2016, Proposition 2.8) that $\text{baÿes}(\cdot, b)$ is the linear interpolation of $\text{bayes}(\cdot, b)$ at the points $0, 1, 3, 5, \dots$.

Theorem 18 *Let $0 < \eta < \frac{1}{2}$, $\ell \geq 1$, and $\mathcal{H}$ be a hypothesis class with VC-dimension $d > 1$. Then, for all $0 < b, p < 1$ satisfying*

$$p\left(\frac{1}{2} - \frac{b}{2}\right) \leq \eta, \tag{15}$$

we have

$$\inf_{\hat{h}_\ell} \sup_{\mathcal{D}\in\mathscr{D}(\eta)} \mathop{\mathbb{E}}_{\mathcal{D}^\ell}\left[\Delta(\hat{h}_\ell, \mathcal{D})\right] \;\geq\; bp\,\text{baÿes}(\ell p/(d-1), b). \tag{16}$$

Furthermore, for $0 \leq u < 1$,

$$\inf_{\hat{h}_\ell} \sup_{\mathcal{D}\in\mathscr{D}(\eta)} \mathbb{P}\left[\Delta(\hat{h}_\ell, \mathcal{D}_{\sigma,b,p}) > bpu\right] \;>\; \text{baÿes}(\ell p/(d-1), b) - u. \tag{17}$$

Proof This proof uses ideas from Devroye et al. (1996, Theorem 14.5), Anthony and Bartlett (1999, Theorem 5.2) and Kontorovich and Pinelis (2016, Theorem 2.2).

We will construct adversarial distributions supported on a shattered subset of size d, and hence there is no loss of generality in taking $\mathcal{X} = [d]$ and $\mathcal{H} = \{-1, 1\}^{\mathcal{X}}$. A random distribution $\mathcal{D}_{\sigma,b,p}$ over $\mathcal{X} \times \{-1, 1\}$, parametrized by a random $\sigma \sim \text{Unif}(\{-1, 1\}^{d-1})$ and scalars $b, p \in (0, 1)$ to be specified later, is defined as follows. The point $x = d \in \mathcal{X}$ gets a marginal weight of $1 - p$ where p is a parameter to be set; the remaining $d - 1$ points each get a marginal weight of $p/(d - 1)$:

$$\mathop{\mathbb{P}}_{X\sim\mathcal{D}_{\sigma,b,p}}[X = d] = 1 - p, \qquad \mathop{\mathbb{P}}_{X\sim\mathcal{D}_{\sigma,b,p}}[X < d] = \frac{p}{d-1}. \tag{18}$$

The distribution of Y conditional on X is given by $\mathbb{P}_{(X,Y)\sim\mathcal{D}_{\sigma,b,p}}[Y=1\,|\,X=d]=1$ and

$$\mathop{\mathbb{P}}_{(X,Y)\sim\mathcal{D}_{\sigma,b,p}}[Y=\pm 1\,|\,X=j<d]=\frac{1}{2}\pm\frac{b\sigma_j}{2}. \tag{19}$$

Suppose that $(X_i,Y_i)_{i\in[\ell]}$ is a sample drawn from $\mathcal{D}^{\ell}_{\sigma,b,p}$. The assumption that $\mathcal{D}_{\sigma,b,p}\in\mathscr{D}(\eta)$ implies that b and p must satisfy the constraint (15).

A standard argument (e.g., Anthony and Bartlett (1999) p. 63 display (5.5)) shows that, for any hypothesis $\hat{h}_\ell$,

$$\begin{aligned}
\Delta(\hat{h}_\ell,\mathcal{D}_{\sigma,b,p}) &= \mathrm{err}(\hat{h}_\ell,\mathcal{D}_{\sigma,b,p})-\inf_{h\in\mathcal{H}}\mathrm{err}(h,\mathcal{D}_{\sigma,b,p})\\
&= \mathop{\mathbb{P}}_{X\sim\mathcal{D}_{\sigma,b,p}}[X=d,\hat{h}_\ell(X)\neq 1]+b\mathop{\mathbb{P}}_{X\sim\mathcal{D}_{\sigma,b,p}}[X<d,\hat{h}_\ell(X)\neq\sigma(X)]\\
&\geq b\mathop{\mathbb{P}}_{X\sim\mathcal{D}_{\sigma,b,p}}[X<d,\hat{h}_\ell(X)\neq\sigma(X)]\\
&= bp\mathop{\mathbb{P}}_{X\sim\mathcal{D}_{\sigma,b,p}}[\hat{h}_\ell(X)\neq\sigma(X)|X<d].
\end{aligned}\tag{20}$$

It follows from Kontorovich and Pinelis (2016, Theorems 2.1, 2.5) that

$$\begin{aligned}
\mathop{\mathbb{E}}_{\sigma}\mathop{\mathbb{P}}_{X\sim\mathcal{D}_{\sigma,b,p}}[\hat{h}_\ell(X)\neq\sigma(X)|X<d] &\geq \mathop{\mathbb{E}}_{N\sim\mathrm{Bin}(\ell,p/(d-1))}[\mathrm{bayes}(N,b)]\\
&\geq \mathop{\mathbb{E}}_{N\sim\mathrm{Bin}(\ell,p/(d-1))}[\mathrm{ba\check{y}es}(N,b)]\\
&\geq \mathrm{ba\check{y}es}(\mathbb{E}[N],b)=\mathrm{ba\check{y}es}(\ell p/(d-1),b),
\end{aligned}\tag{21}$$

where the second inequality holds because $\mathrm{ba\check{y}es}$ is, by definition, a convex minorant of bayes, and the third follows from Jensen's inequality. Combined with (20), this proves (16).

To show (17), define the random variable

$$Z=Z(\sigma,\mathcal{L})=\mathop{\mathbb{P}}_{X\sim\mathcal{D}_{\sigma,b,p}}[\hat{h}_\ell(X)\neq\sigma(X)|X<d].$$

Since $Z\in[0,1]$, Markov's inequality implies

$$\mathbb{P}[Z>u]\geq\frac{\mathbb{E}[Z]-u}{1-u}>\mathbb{E}[Z]-u,\qquad 0\leq u<1.$$

Now (20) implies that $\Delta(\hat{h}_\ell,\mathcal{D}_{\sigma,b,p})\geq bpZ$ and hence, for $0\leq u<1$,

$$\begin{aligned}
\inf_{\hat{h}_\ell}\sup_{\mathcal{D}\in\mathscr{D}(\eta)}\mathbb{P}\left[\Delta(\hat{h}_\ell,\mathcal{D}_{\sigma,b,p})>bpu\right] &= \inf_{\hat{h}_\ell}\sup_{\mathcal{D}\in\mathscr{D}(\eta)}\mathbb{P}[Z>u]\\
&> \inf_{\hat{h}_\ell}\sup_{\mathcal{D}\in\mathscr{D}(\eta)}\mathbb{E}[Z]-u\\
&\geq \frac{1}{bp}\inf_{\hat{h}_\ell}\sup_{\mathcal{D}\in\mathscr{D}(\eta)}\mathbb{E}[\Delta(\hat{h}_\ell,\mathcal{D}_{\sigma,b,p})]-u\\
&\geq \mathrm{ba\check{y}es}(\ell p/(d-1),b)-u.
\end{aligned}$$

■

7.2 Proof of Theorem 5

We break down the proof into several steps. For now, we assume that the labeled examples are sampled i.i.d. as per the classic PAC setup. At the end, we show how to extend the proof to the semi-supervised setting.

(i) Defining a family of adversarial distributions. Let T be a $\bar{t}$-net of $\mathcal{X}$ of size $\Theta(\sqrt{m})$ and $\eta=\Theta(1/\sqrt{m})$. For any passive learning algorithm mapping i.i.d. samples of size $\ell=\tilde{\Theta}(\sqrt{m})$ to hypotheses $\hat{h}_\ell:\mathcal{X}\to\{-1,1\}$, we construct a random adversarial distribution $\mathcal{D}_{\sigma,b,p}$ with agnostic error η. We accomplish this via the construction described in the proof of Theorem 18, with $|T|=d=\Theta(\sqrt{m})$. The marginal distribution over $T=\{x_1,\ldots,x_d\}$ puts a mass of $1-p$ on $x_d\in T$ and spreads the remaining mass uniformly over the other points, as in (18). The "heavy" point has a deterministic label and the remaining "light" points have noisy labels drawn from a random distribution with symmetric noise level b, as in (19). We proceed to choose b and p; namely,

$$p=\frac{d-1}{2\ell}\sqrt{\eta}=\tilde{\Theta}(m^{-1/4}),\qquad b=1-\frac{2\eta}{p}=1-\tilde{\Theta}(m^{-1/4}),$$

which makes the constraint in (15) hold with equality; this means that the agnostic error is exactly η and in particular, establishes (i).

(ii.a) Lower-bounding the passive learner's error. Our choice of p implies that $\ell p/(d-1)=\sqrt{\eta}/2=:\kappa<1$. For this range of κ, Kontorovich and Pinelis (2016, Proposition 2.8) implies that $\mathrm{ba\check{y}es}(\kappa,b)=\frac{1}{2}(1-\kappa b)=\Theta(1)$. Choosing $u=\frac{1}{4}(1-\kappa b)=\Theta(1)$ in (17), Theorem 18 implies that

$$\inf_{\hat{h}_\ell}\sup_{\mathcal{D}\in\mathscr{D}(\eta)}\mathbb{P}[\Delta(\hat{h}_\ell,\mathcal{D})>\tilde{\Omega}(m^{-1/4})]>\Omega(1).$$

In more formal terms, there exist constants $c_0,c_1>0$ such that

$$\inf_{\hat{h}_\ell}\sup_{\mathcal{D}\in\mathscr{D}(\eta)}\mathbb{P}[\Delta(\hat{h}_\ell,\mathcal{D})>c_0 p]>c_1. \tag{22}$$

(ii.b) Upper-bounding $\nu(\bar{t})$. To establish (ii.b), it suffices to show that for $(X_i,Y_i)_{i\in[\ell]}\sim\mathcal{D}^{\ell}_{\sigma,b,p}$, we will have $\nu(\bar{t})=O(m^{-1/2})$ with sufficiently high probability. Indeed, the latter condition implies the requisite upper bound on $\min_{t>0:\mathcal{N}(t)<m}\mathrm{GB}(\nu(t),\mathcal{N}(t),\delta,m,1)$, while (i) implies the lower bound, since the latter quantity cannot be asymptotically smaller than the Bayes error (which coincides with the agnostic error for $\mathcal{H}=\{-1,1\}^{\mathcal{X}}$).

Recall that the $\bar{t}$-net points $\{x_1,\ldots,x_{d-1}\}$ are the "light" ones (i.e., each has weight $p/(d-1)$) and define the random sets $J_j\subset[\ell]$ by

$$J_j=\{i\in[\ell]:X_i=x_j\},\qquad j\in[d-1].$$

In words, J_j consists of the indices i of the sample points for which X_i falls on the net point x_j. For $y\in\{-1,1\}$, put $\tau_j^y=\sum_{i\in J_j}\mathbb{I}[Y_i=y]$ and define the *minority count* ξ_j at the net point x_j by

$$\xi_j=\min_{y\in\{-1,1\}}\tau_j^y=\tfrac{1}{2}(|\tau_j^+-\tau_j^-|-|\tau_j^++\tau_j^-|).$$

Observe that by virtue of being a $\bar{t}$-net, T is $\bar{t}$-separated and hence the only contribution to $\nu(\bar{t})$ is from the minority counts (to which the "heavy" point x_d does not contribute due to its deterministic

label):

$$\nu(\bar{t}) = \frac{1}{\ell}\sum_{j=1}^{d-1}\xi_j.$$

Now

$$\mathbb{E}\,|\tau_j^+ + \tau_j^-| = \mathbb{E}\,|J_j| = \frac{\ell p}{d-1} = \Theta(m^{-1/4})$$

and

$$\begin{aligned}\mathbb{E}\,|\tau_j^+ - \tau_j^-| &= \mathop{\mathbb{E}}_{\sigma_j}\mathbb{E}[|\tau_j^+ - \tau_j^-|\,|\sigma_j] \\ &\geq \mathop{\mathbb{E}}_{\sigma_j}\left|\mathbb{E}[\tau_j^+ - \tau_j^-\,|\sigma_j]\right|.\end{aligned}$$

Computing

$$\mathbb{E}[\tau_j^+|\sigma_j = +1] = (\frac{1}{2}+\frac{b}{2})\frac{\ell p}{d-1}, \qquad \mathbb{E}[\tau_j^-|\sigma_j = +1] = (\frac{1}{2}-\frac{b}{2})\frac{\ell p}{d-1},$$

with an analogous calculation when conditioning on $\sigma_j = -1$, we get

$$\mathbb{E}\,|\tau_j^+ - \tau_j^-| \geq \frac{b\ell p}{d-1}$$

and hence

$$\begin{aligned}\mathbb{E}[\xi_j] &\leq \frac{1}{2}\left(\frac{\ell p}{d-1} - b\frac{\ell p}{d-1}\right) \\ &= (1-b)\frac{\ell p}{2(d-1)} = \frac{2\eta}{p}\cdot\frac{\ell p}{2(d-1)} = \frac{\eta\ell}{d-1}.\end{aligned}$$

It follows that

$$\begin{aligned}\mathbb{E}[\nu(\bar{t})] &= \frac{1}{\ell}\sum_{j=1}^{d-1}\mathbb{E}[\xi_j] \\ &\leq \frac{d-1}{\ell}\cdot\frac{\eta\ell}{d-1} = \eta = \Theta(m^{-1/2}).\end{aligned}$$

To give tail bounds on $\nu(\bar{t})$, we use Markov's inequality: for all $c_2 > 0$,

$$\mathbb{P}[\nu(\bar{t}) > c_2\,\mathbb{E}[\nu(\bar{t})] \leq \frac{1}{c_2}.$$

Choosing c_2 sufficiently large that $1 - 1/c_2 > c_1$ (the latter from (22)) implies the existence of constants $c_0, c_2, c_3 > 0$ such that

$$\inf_{\hat{h}_\ell}\sup_{\mathcal{D}\in\mathscr{D}(\eta)}\mathbb{P}\left[\left(\Delta(\hat{h}_\ell,\mathcal{D}) > c_0 p\right) \wedge (\nu(\bar{t}) \leq c_2\eta)\right] > c_3.$$

Since $p = \tilde{\Theta}(m^{-1/4})$ and $\eta = \Theta(m^{-1/2})$, this establishes (ii) and concludes the proof of Theorem 5.

(iii) Extending to the semi-supervised setting. Providing the learner with the exact weights of $\mathcal{X} = [d]$ under our adversarial distribution does not give it any additional power. Indeed, the information-theoretic excess risk lower bound in Eq. (21) hinges on the fact that to estimate $\sigma(x)$ with some desired certainty, the point x must be sampled some minimal number of times. The marginal probability of x does not enter that calculation, and hence knowing its value does not afford the learner an improved performance.

8. Active Learning Lower Bound

We now prove the active learning lower bound stated in Theorem 6. To prove the theorem, we first prove a result which is similar to the classical No-Free-Lunch theorem, except it holds for active learning algorithms. The proof follows closely the proof of the classical No-Free-Lunch theorem given in Shalev-Shwartz and Ben-David (2014, Theorem 5.1), with suitable modifications.

Theorem 19 *Let $\beta \in [0, \frac{1}{2})$, and m be an integer. Let $\mathcal{A}$ any active learning algorithm over a finite domain $\mathcal{X}$ which gets as input a random labeled sample $S \sim \mathcal{D}^m$ (with hidden labels) and outputs $\hat{h}$. If $\mathcal{A}$ queries fewer than $\mathcal{X}/2$ labels from S, then there exists a distribution $\mathcal{D}$ over $\mathcal{X}\times\{0,1\}$ such that*

- *Its marginal on $\mathcal{X}$ is uniform, and for each $x \in \mathcal{X}$, $\mathbb{P}[Y = 1 \mid X = x] \in \{\beta, 1-\beta\}$.*
- $\mathbb{E}[\mathrm{err}(\hat{h},\mathcal{D})] \geq \frac{1}{4}$.

Proof Let $\mathcal{F} = \{f_1,\ldots,f_T\}$ be the set of possible functions $f_i : \mathcal{X} \to \{0,1\}$. Let $\mathcal{D}_i$ to be a distribution with a uniform marginal over $\mathcal{X}$, and $\mathbb{P}_{(X,Y)\sim\mathcal{D}_i}[Y = 1 \mid X = x] = f_i(x)(1-\beta) + (1-f_i(x))\beta$. Consider the following random process: First, draw an unlabeled sample $X = (x_1,\ldots,x_m)$ i.i.d. from $\mathcal{D}_X^m$. Then, draw $B = (b_1,\ldots,b_m)$ independently from a Bernoulli distribution with $\mathbb{P}[b_i = 1] = \beta$. For $i \in [T]$, let $S^i(X,B) = ((x_1,y_1),\ldots,(x_m,y_m))$ such that x_i are set by X, and $y_i = f_i(x)$ if $b_i = 0$ and $1 - f_i(x)$ otherwise. Clearly, $S^i(X,B)$ is distributed according to $\mathcal{D}_i^m$. Let $h_i(S)$ be the output of $\mathcal{A}$ when the labeled sample is S. Denote by $\hat{h}_i$ the (random) output of $\mathcal{A}$ when the sample is drawn from $\mathcal{D}_i$. Clearly

$$\mathbb{E}[\mathrm{err}(\hat{h}_i,\mathcal{D}_i)] = \mathop{\mathbb{E}}_{X,B}[\mathrm{err}(h_i(S(X,B)),\mathcal{D}_i)].$$

Therefore (as in (5.4) in Shalev-Shwartz and Ben-David (2014)), for some j, X, B it holds that

$$\mathbb{E}[\mathrm{err}(\hat{h}_j,\mathcal{D}_j)] \geq \frac{1}{T}\sum_{i=1}^{T}\mathbb{E}[\mathrm{err}(\hat{h}_i,\mathcal{D}_i)] \geq \frac{1}{T}\sum_{i=1}^{T}\mathrm{err}(h_i(S(X,B)),\mathcal{D}_i). \tag{23}$$

Fix X, B, j as above, and denote for brevity $h_i := h_i(S(X,B))$. Let V_i be the set of examples $x \in \mathcal{X}$ for which that $\mathcal{A}$ does not observe their label if the labeled sample is $S^i(X,B)$ (this includes both examples that are not in the sample at all as well as examples that are in the sample but their label is not requested by $\mathcal{A}$). We have $|V_i| > |\mathcal{X}|/2$ by assumption. Then (as in Eq. (5.6) therein)

$$\frac{1}{T}\sum_{i=1}^{T}\mathrm{err}(h_i,\mathcal{D}_i) \geq \frac{1}{T}\sum_{i=1}^{T}\frac{1}{2|V_i|}\sum_{x\in V_i}\mathbb{I}[h_i(x) \neq f_i(x)]. \tag{24}$$

Since $\mathcal{A}$ is active, it selects which examples to request, which can depend on the labels observed by $\mathcal{A}$ so far. Therefore, V_i can be different for different i. However, an argument similar to that of the No-Free-Lunch theorem for the passive case still goes through, as follows.

Let i, i' such that $f_i(x) = f_{i'}(x)$ for all $x \notin V_i$, and $f_i(x) = 1 - f_{i'}(x)$ for all $x \in V_i$. Since X, B are fixed, $\mathcal{A}$ observes the same labels for all $x \notin V_i$ for both $S^{i'}(X,B)$ and $S^i(X,B)$, thus all its decisions and requests are identical for both samples, and so $V_i = V_{i'}$, and $h_i = h_{i'}$. Therefore, it is possible to partition T into $T/2$ pairs of indices i, i' such that for each such pair,

$$\begin{aligned}
&\frac{1}{2|V_i|}\sum_{x\in V_i}\mathbb{I}[h_i(x)\neq f_i(x)] + \frac{1}{2|V_{i'}|}\sum_{x\in V_{i'}}\mathbb{I}[h_{i'}(x)\neq f_{i'}(x)] \\
&= \frac{1}{2|V_i|}\sum_{x\in V_i}\mathbb{I}[h_i(x)\neq f_i(x)] + \mathbb{I}[h_i(x)\neq 1-f_i(x)] \\
&= \frac{1}{2}.
\end{aligned}$$

Therefore, $\frac{1}{T}\sum_{i=1}^{T}\mathrm{err}(h_i,\mathcal{D}_i) \geq \frac{1}{4}$. Therefore, from Eq. (24), $\frac{1}{T}\sum_{i=1}^{T}\mathrm{err}(h_i,\mathcal{D}_i) \geq \frac{1}{4}$. Combining this with Eq. (23), it follows that $\mathbb{E}[\mathrm{err}(\hat{h}_j,\mathcal{D}_j)] \geq \frac{1}{4}$. ■

We will also make use of the following simple lemma.

Lemma 20 *Let $\beta \in [0,1]$. Let $\mathcal{D}$ be a distribution over $\mathcal{X}\times\{0,1\}$ such that for $(X,Y)\sim\mathcal{D}$, for any x in the support of $\mathcal{D}$, $\mathbb{P}[Y=1 \mid X=x] \in \{\beta, 1-\beta\}$. Let N be the size of the support of $\mathcal{D}$. Let $S\sim\mathcal{D}^m$. Denote by n_x the number of sample pairs (x',y') in S where $x'=x$, and let n_x^+ be the number of sample pairs (x',y') where $x'=x$ and $y'=1$. Let $\hat{p}_x^+ = n_x^+/n_x$ (or zero if $n_x=0$). Then*

$$2\beta(1-\beta)(m-N) \leq \sum_{x\in\mathcal{X}}\mathbb{E}[2n_x\hat{p}_x^+(1-\hat{p}_x^+)] \leq 2\beta(1-\beta)m.$$

Proof We have

$$\mathbb{E}[2n_x\hat{p}_x^+(1-\hat{p}_x^+)] = \sum_{i=1}^{\infty}\mathbb{P}[n_x=i]\cdot i\cdot\mathbb{E}[2\hat{p}_x^+(1-\hat{p}_x^+)\mid n_x=i].$$

Note that $\mathbb{E}[2\hat{p}_x^+(1-\hat{p}_x^+)\mid n_x=1]=0$. For $i>1$, let $y_1,\ldots,y_i$ be the labels of the examples that are equal to x in S, then

$$\sum_{j,k\in[i]}\mathbb{I}[y_k\neq y_j] = 2n_x^+(i-n_x^+) = i^2\cdot 2\hat{p}_x^+(1-\hat{p}_x^+).$$

Therefore, letting $(X_1,Y_1),(X_2,Y_2)\sim\mathcal{D}^2$,

$$\begin{aligned}
\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[2\hat{p}_x^+(1-\hat{p}_x^+)\mid n_x=i] &= \frac{1}{i^2}\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[\sum_{j,k\in[n_x]}\mathbb{I}[y_k\neq y_j]\mid n_x=i] \\
&= \frac{i^2-i}{i^2}\,\mathbb{P}[Y_1\neq Y_2\mid X_1=X_2=x] \\
&= 2(1-\frac{1}{i})\beta(1-\beta),
\end{aligned}$$

Thus

$$\begin{aligned}
\mathbb{E}[2n_x\hat{p}_x^+(1-\hat{p}_x^+)] &= 2\beta(1-\beta)\sum_{i=2}^{\infty}(i-1)\,\mathbb{P}[n_x=i] \\
&= 2\beta(1-\beta)(\mathbb{E}[n_x]+\mathbb{P}[n_x=0]-1).
\end{aligned}$$

To complete the proof, sum over all x in the support of $\mathcal{D}$, and note that $\sum_x\mathbb{E}[n_x]=m$, and $\sum_x(\mathbb{P}[n_x=0]-1)\in[-N,0]$. ■

We now prove our lower bound, stated in Theorem 6, on the number of queries required by any active learning with competitive guarantees similar to ours.

Proof [of Theorem 6] Let $N = \left\lfloor \frac{m\alpha - \log(\frac{m}{\delta})}{\log(m)} \right\rfloor$. Let $\beta = 8\alpha \leq \frac{1}{2}$. Consider a marginal distribution $\mathcal{D}_X$ over $\mathcal{X}$ which is uniform over N points $1,\ldots,N\in\mathbb{R}$. Consider the following family of distributions: $\mathcal{D}$ such that its marginal over $\mathcal{X}$ is $\mathcal{D}_X$, and for each $x\in\mathcal{X}$, $\mathbb{P}[Y=1\mid X=x]\in\{\beta,1-\beta\}$. Thus the Bayes optimal error for each of these distributions is β.

Let $S\sim\mathcal{D}^m$. If one example in S is changed, $\nu(\frac{1}{2})$ changes by at most $1/m$. Hence, by McDiarmid's inequality (McDiarmid, 1989), with probability at least $1-\frac{1}{28}$, $\left|\nu(\frac{1}{2})-\mathbb{E}[\nu(\frac{1}{2})]\right| \leq \sqrt{\frac{\log(28)}{2m}}$. Denote the event that this holds E_M. Since $\beta = 8\alpha \geq \frac{\log(m)+\log(28)}{\sqrt{2m}}$, it follows that under E_M,

$$|\nu(1/2)-\mathbb{E}[\nu(1/2)]| \leq \beta/8. \tag{25}$$

We now bound $\mathbb{E}[\nu(\frac{1}{2})]$. Using the notation $\hat{p}_x^+, n_x, n_x^+$ as in Lemma 20, we have

$$\nu(1/2) = \frac{1}{m}\sum_{x\in\mathcal{X}}\min(n_x^+, n_x-n_x^+) = \frac{1}{m}\sum_{x\in\mathcal{X}}n_x\min(p_x^+,1-p_x^+)$$

Also, for all $p\in[0,1]$, $\min(p,1-p)\leq 2p(1-p)\leq 2\min(p,1-p)$. Therefore

$$\frac{1}{2m}\sum_{x\in\mathcal{X}}2n_xp_x^+(1-p_x^+) \leq \nu(1/2) \leq \frac{1}{m}\sum_{x\in\mathcal{X}}2n_xp_x^+(1-p_x^+).$$

By Lemma 20, it follows that

$$\frac{m-N}{m}\beta(1-\beta) \leq \mathbb{E}[\nu(1/2)] \leq 2\beta(1-\beta).$$

Since $N\leq m/2$ and $\beta\in[0,\frac{1}{2}]$, $\mathbb{E}[\nu(\frac{1}{2})]\in(\beta/4,2\beta)$. Combining this with Eq. (25), we get that under E_M, $\alpha=\beta/8\leq\nu(\frac{1}{2})\leq\frac{17}{8}\beta=17\alpha$.

Now, we bound $G_{\min}$ from above and below assuming E_M holds. Denote

$$G(t) := \mathrm{GB}(\nu(t),\mathcal{N}(t),\delta,m,1).$$

To establish a lower bound on $G_{\min}(m,\delta)$, note that $G_{\min}(m,\delta)=\min_{t>0}G(t)\geq\min_{t>0}\nu(t)$. For $t\in(0,\frac{1}{2})$, $\nu(t)=\nu(\frac{1}{2})$ (since the distances between any two distinct points in S is at least 1). In addition, since ν is monotonically increasing, we have $\nu(t)\geq\nu(\frac{1}{2})$ for $t\geq\frac{1}{2}$. Hence $\min_{t>0}\nu(t)\geq\nu(\frac{1}{2})\geq\beta/8=\alpha$.

To show an upper bound on $G_{\min}(m,\delta)$, we upper bound $G(\frac{1}{2})$. Note that $\mathcal{N}(\frac{1}{2}) \leq N$. Recall the definition of $\phi(t)$ in Eq. (6). We have

$$\phi(\frac{1}{2}) = \frac{(N+1)\log(m) + \log(\frac{1}{\delta})}{m} \leq \alpha.$$

Then, since $\nu(\frac{1}{2}) \leq 17\alpha$,

$$G(1/2) \leq \frac{m}{m-N}(\nu(\frac{1}{2}) + \frac{2}{3}\alpha + \frac{3}{\sqrt{2}}\sqrt{\nu(\frac{1}{2})\alpha}) \leq 30\alpha.$$

In the last inequality we used the fact that $\frac{m}{m-N} \leq 10/9$. So if E_M holds, $G_{\min}(m,\delta) \leq G(\frac{1}{2}) \leq 30\alpha$.

From the assumption on $\mathcal{A}$, with probability at least $1-\delta$, we have $\mathrm{err}(\hat{h},\mathcal{D}) \leq CG_{\min}(m,\delta) \leq 30C\alpha \leq 1/8$ (since $\alpha \leq \frac{1}{240C}$). Let $E_L(\mathcal{D})$ denote the event that $\mathcal{A}$ queries fewer than $N/2$ labels, where the probability is over the randomness of S and $\mathcal{A}$. Let h' be the output of an algorithm that behaves like $\mathcal{A}$ in cases where $E_L(\mathcal{D})$ holds, and queries at most $N/2$ otherwise. By Theorem 19, there exists some $\mathcal{D}$ in the family of distributions such that $\mathbb{E}[\mathrm{err}(h',\mathcal{D})] \geq \frac{1}{4}$. By Markov's inequality, $\mathbb{P}[\mathrm{err}(h',\mathcal{D}) \geq \frac{1}{8}] \geq 1/7$. Also, $\mathbb{P}[h' = \hat{h}] \geq \mathbb{P}[E_L(\mathcal{D})]$. Therefore

$$\mathbb{P}[\mathrm{err}(\hat{h},\mathcal{D}) \geq 1/8] \geq \mathbb{P}[\mathrm{err}(h',\mathcal{D}) \geq 1/8] + \mathbb{P}[E_L(\mathcal{D})] - 1 = \mathbb{P}[E_L(\mathcal{D})] - 6/7.$$

Therefore $\mathbb{P}[E_L(\mathcal{D})] - 6/7 \leq \mathbb{P}[\mathrm{err}(\hat{h},\mathcal{D}) \geq \frac{1}{8}] \leq \delta$. Since by assumption $\delta \leq 1/14$, it follows that $\mathbb{P}[E_L(\mathcal{D})] \leq 6/7 + \delta \leq 13/14$. It follows that with a probability of at least $1/14$, the negation of $E_L(\mathcal{D})$ holds. Since also E_M holds with probability at least $1 - \frac{1}{28}$, it follows that with a probability of at least $\frac{1}{28}$, both E_M and the negation of $E_L(\mathcal{D})$ hold. Now, as shown above, E_M implies the bounds on $G_{\min}(m,\delta)$ (item 1 in the theorem statement). In addition, the negation of $E_L(\mathcal{D})$ implies that $\mathcal{A}$ queries at least $N/2 = \frac{1}{2}\left\lfloor \frac{m\alpha - \log(\frac{m}{\delta})}{\log(m)} \right\rfloor$ labels (item 2 in the theorem statement). This completes the proof. ■

9. Discussion

We have presented an efficient fully empirical proximity-based non parametric active learner. Our approach provides competitive error guarantees for general distributions, in a general metric space, while keeping label complexity significantly lower than any passive learner with the same guarantees. MARMANN yields fully empirical error estimates, easily computable from finite samples. This is in contrast with classic techniques, that present bounds and rates that depend on unknown distribution-dependent quantities.

An interesting question is whether the guarantees can be related to the Bayes error of the distribution. Our error guarantees give a constant factor over the error guarantees of Gottlieb et al. (2017). A variant of this approach (Gottlieb et al., 2014a) was shown to be Bayes-consistent (Kontorovich and Weiss, 2015), and we conjecture that this holds also for the algorithm of Gottlieb et al. (2017). The passive component of our learning algorithm is indeed Bayes-consistent (Kontorovich et al., 2017). Since in our analysis MARMANN achieves a constant factor over the error of the passive

learner, Bayes-consistency of the active learner cannot be inferred from our present techniques; we leave this problem open for future research.

Another important issue is one of efficient implementation. We mentioned that the naive $O(m^2)$ runtime for constructing a t-net may be improved to $2^{O(\mathsf{ddim}(\mathcal{X}))}m\log(1/t)$, as shown in Krauthgamer and Lee (2004); Gottlieb et al. (2014b). The fast t-net construction was the algorithmic work-horse of Gottlieb et al. (2014a,b, 2017) and inspired the passive component of our learner. We note that implementing even this passive component efficiently is far from trivial; this formed the core of a Master's thesis (Korsunsky, 2017). The remaining obstacle to making our algorithm fully practical is the magnitude of some of the constants. We believe these to be artifacts of the proof and intend to bring them down to manageable values in future work.

Acknowledgments

Acknowledgements Sivan Sabato was partially supported by the Israel Science Foundation (grant No. 555/15). Aryeh Kontorovich was partially supported by the Israel Science Foundation (grants No. 1141/12 and 755/15) and Yahoo Faculty and Paypal awards. We thank Lee-Ad Gottlieb and Dana Ron for the helpful discussions, and the referees for carefully reading the manuscript and their helpful suggestions.

References

Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, Cambridge, 1999.

Jean-Yves Audibert. Fast learning rates in statistical inference through aggregation. *Ann. Statist.*, 37(4):1591–1646, 08 2009.

Pranjal Awasthi, Maria-Florina Balcan, and Philip M. Long. The power of localization for efficiently learning linear separators with noise. In *Symposium on Theory of Computing, STOC 2014*, pages 449–458, 2014.

Maria-Florina Balcan, Andrei Z. Broder, and Tong Zhang. Margin based active learning. In *Proceedings of the 20th Annual Conference on Learning Theory, COLT 2007*, pages 35–50, 2007.

Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. *J. Comput. Syst. Sci.*, 75(1), 2009.

Maria-Florina Balcan, Steve Hanneke, and Jennifer Wortman Vaughan. The true sample complexity of active learning. *Machine Learning*, 80(2-3):111–139, 2010.

Daniel Berend and Aryeh Kontorovich. A finite sample analysis of the naive bayes classifier. *Journal of Machine Learning Research*, 16:1519–1545, 2015.

Christopher Berlind and Ruth Urner. Active nearest neighbors in changing environments. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015*, pages 1870–1879, 2015.

Oren Boiman, Eli Shechtman, and Michal Irani. In defense of nearest-neighbor based image classification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2008*, 2008.

Nader H. Bshouty, Yi Li, and Philip M. Long. Using the doubling dimension to analyze the generalization of learning algorithms. *Journal of Computer and System Sciences*, 75(6):323 – 335, 2009.

Rui M Castro and Robert D Nowak. Minimax bounds for active learning. *IEEE Transactions on Information Theory*, 54(5):2339–2353, 2008.

Rui M. Castro, Rebecca Willett, and Robert D. Nowak. Faster rates in regression via active learning. In *Advances in Neural Information Processing Systems, NIPS 2005*, pages 179–186, 2005.

Nicolò Cesa-Bianchi, Claudio Gentile, Fabio Vitale, and Giovanni Zappella. Active learning on trees and graphs. In *Proceedings of rhe 23rd Conference on Learning Theory, COLT 2010*, pages 320–332, 2010.

Kamalika Chaudhuri and Sanjoy Dasgupta. Rates of convergence for nearest neighbor classification. In *Advances in Neural Information Processing Systems, NIPS 2014*, pages 3437–3445, 2014.

Thomas M. Cover and Peter E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13:21–27, 1967.

Gautam Dasarathy, Robert D. Nowak, and Xiaojin Zhu. S2: an efficient graph based active learning algorithm with application to nonparametric classification. In *Proceedings of the 28th Annual Conference on Learning Theory, COLT 2015*, pages 503–522, 2015.

Sanjoy Dasgupta. Analysis of a greedy active learning strategy. In *Advances in Neural Information Processing Systems, NIPS 2004*, pages 337–344, 2004.

Sanjoy Dasgupta. Consistency of nearest neighbor classification under selective sampling. In *Proceedings of the 25th Annual Conference on Learning Theory, COLT 2012*, pages 18.1–18.15, 2012.

Sanjoy Dasgupta and Daniel Hsu. Hierarchical sampling for active learning. In *Proceedings of the 25th International Conference on Machine Learning, ICML 2008*, pages 208–215, 2008.

Luc Devroye and László Györfi. *Nonparametric density estimation: the L_1 view*. Wiley Series in Probability and Mathematical Statistics: Tracts on Probability and Statistics. John Wiley & Sons, Inc., New York, 1985.

Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*, volume 31 of *Applications of Mathematics (New York)*. Springer-Verlag, New York, 1996.

Andreas Emil Feldmann, Wai Shing Fung, Jochen Könemann, and Ian Post. A $(1 + \epsilon)$-embedding of low highway dimension graphs into bounded treewidth graphs. *CoRR*, abs/1502.04588, 2015. URL `http://arxiv.org/abs/1502.04588`.

Evelyn Fix and Jr. Hodges, J. L. Report Number 4, Project Number 21-49-004, USAF School of Aviation Medicine, Randolph Field, Texas, 1951.

Evelyn Fix and Jr. Hodges, J. L. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review / Revue Internationale de Statistique*, 57(3):pp. 238–247, 1989.

Sally Floyd and Manfred K. Warmuth. Sample compression, learnability, and the vapnik-chervonenkis dimension. *Machine Learning*, 21(3):269–304, 1995.

Alon Gonen, Sivan Sabato, and Shai Shalev-Shwartz. Efficient active learning of halfspaces: an aggressive approach (extended abstract ICML 2013). *Journal of Machine Learning Research*, 14 (1):2583–2615, 2013.

Lee-Ad Gottlieb and Robert Krauthgamer. Proximity algorithms for nearly doubling spaces. *SIAM J. Discrete Math.*, 27(4):1759–1769, 2013.

Lee-Ad Gottlieb, Aryeh Kontorovich, and Robert Krauthgamer. Efficient classification for metric data (extended abstract COLT 2010). *IEEE Transactions on Information Theory*, 60(9):5750–5759, 2014a.

Lee-Ad Gottlieb, Aryeh Kontorovich, and Pinhas Nisnevitch. Near-optimal sample compression for nearest neighbors. In *Advances in Neural Information Processing Systems, NIPS 2014*, pages 370–378, 2014b.

Lee-Ad Gottlieb, Aryeh Kontorovich, and Robert Krauthgamer. Adaptive metric dimensionality reduction (extended abstract ALT 2013). *Theoretical Computer Science*, pages 105–118, 2016.

Lee-Ad Gottlieb, Aryeh Kontorovich, and Pinhas Nisnevitch. Nearly optimal classification for semimetrics (extended abstract AISTATS 2016). *Journal of Machine Learning Research*, 2017.

Thore Graepel, Ralf Herbrich, and John Shawe-Taylor. Pac-bayesian compression bounds on the prediction error of learning algorithms for classification. *Machine Learning*, 59(1-2):55–76, 2005.

Steve Hanneke. Rates of convergence in active learning. *The Annals of Statistics*, 39(1):333–361, 2011.

Steve Hanneke and Liu Yang. Minimax analysis of active learning. *Journal of Machine Learning Research*, 16:3487–3602, 2015.

Aryeh Kontorovich and Iosif Pinelis. Exact lower bounds for the agnostic probably-approximately-correct (PAC) machine learning model. *CoRR*, abs/1606.08920, 2016.

Aryeh Kontorovich and Roi Weiss. Maximum margin multiclass nearest neighbors. In *Proceedings of the 31st International Conference on Machine Learning, ICML 2014*, pages 892–900, 2014.

Aryeh Kontorovich and Roi Weiss. A bayes consistent 1-nn classifier. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2015*, 2015.

Aryeh Kontorovich, Sivan Sabato, and Roi Weiss. Nearest-neighbor sample compression: Efficiency, consistency, infinite dimensions. In *Advances in Neural Information Processing Systems, NIPS 2017*, pages 1572–1582, 2017.

Yevgeni Korsunsky. Classifying processes by their system call traces using 1-nearest-neighbor with compression. Master's thesis, Ben-Gurion University Computer Science Department, 2017. URL `https://tinyurl.com/Korsunsky-msc`.

Samory Kpotufe. k-nn regression adapts to local intrinsic dimension. In *Advances in Neural Information Processing Systems, NIPS 2011*, pages 729–737, 2011.

Samory Kpotufe, Ruth Urner, and Shai Ben-David. Hierarchical label queries with data-dependent partitions. In *Proceedings of the 28th Annual Conference on Learning Theory, COLT 2015*, pages 1176–1189, 2015.

Robert Krauthgamer and James R. Lee. Navigating nets: Simple algorithms for proximity search. In *15th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 791–801, January 2004.

Sanjeev R. Kulkarni and S. E. Posner. Rates of convergence of nearest neighbor estimation under arbitrary sampling. *IEEE Transactions on Information Theory*, 41(4):1028–1039, 1995.

Nick Littlestone and Manfred K. Warmuth. Relating data compression and learnability. unpublished, 1986.

Andreas Maurer and Massimiliano Pontil. Empirical Bernstein bounds and sample-variance penalization. In *Proceedings of the 22nd Annual Conference on Learning Theory, COLT 2009*, 2009.

Andrew McCallum and Kamal Nigam. Employing EM and pool-based active learning for text classification. In *Proceedings of the 15th International Conference on Machine Learning, ICML 1998*, pages 350–358, 1998.

Colin McDiarmid. On the method of bounded differences. In J. Siemons, editor, *Surveys in Combinatorics, volume 141 of LMS Lecture Notes Series*, pages 148–188. Morgan Kaufmann Publishers, San Mateo, CA, 1989.

Sivan Sabato and Rémi Munos. Active regression by stratification. In *Advances in Neural Information Processing Systems, NIPS 2014*, pages 469–477, 2014.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

John Shawe-Taylor, Peter L. Bartlett, Robert C. Williamson, and Martin Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 44(5): 1926–1940, 1998.

Charles J. Stone. Consistent nonparametric regression. *The Annals of Statistics*, 5(4):595–620, 1977.

Ruth Urner, Sharon Wulff, and Shai Ben-David. PLAL: cluster-based active learning. In *Proceedings of the 26th Annual Conference on Learning Theory, COLT 2013*, pages 376–397, 2013.

Ulrike von Luxburg and Olivier Bousquet. Distance-based classification with Lipschitz functions. *Journal of Machine Learning Research*, 5:669–695, 2004.

Kai Wei, Rishabh K. Iyer, and Jeff A. Bilmes. Submodularity in data subset selection and active learning. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015*, pages 1954–1963, 2015.

Lin Cheng Zhao. Exponential bounds of mean error for the nearest neighbor estimates of regression functions. *J. Multivariate Anal.*, 21(1):168–178, 1987.

Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *ICML 2003 workshop*, pages 58–65, 2003.

Journal of Machine Learning Research 18 (2018) 1-39 Submitted 2/17; Revised 3/18; Published 4/18

From Predictive Methods to Missing Data Imputation: An Optimization Approach

Dimitris Bertsimas DBERTSIM@MIT.EDU
Colin Pawlowski CPAWLOWS@MIT.EDU
Ying Daisy Zhuo ZHUO@MIT.EDU
Sloan School of Management and Operations Research Center
Massachusetts Institute of Technology
Cambridge, MA 02139

Editor: Francis Bach

Abstract

Missing data is a common problem in real-world settings and for this reason has attracted significant attention in the statistical literature. We propose a flexible framework based on formal optimization to impute missing data with mixed continuous and categorical variables. This framework can readily incorporate various predictive models including K-nearest neighbors, support vector machines, and decision tree based methods, and can be adapted for multiple imputation. We derive fast first-order methods that obtain high quality solutions in seconds following a general imputation algorithm `opt.impute` presented in this paper. We demonstrate that our proposed method improves out-of-sample accuracy in large-scale computational experiments across a sample of 84 data sets taken from the UCI Machine Learning Repository. In all scenarios of missing at random mechanisms and various missing percentages, `opt.impute` produces the best overall imputation in most data sets benchmarked against five other methods: mean impute, K-nearest neighbors, iterative knn, Bayesian PCA, and predictive-mean matching, with an average reduction in mean absolute error of 8.3% against the best cross-validated benchmark method. Moreover, `opt.impute` leads to improved out-of-sample performance of learning algorithms trained using the imputed data, demonstrated by computational experiments on 10 downstream tasks. For models trained using `opt.impute` single imputations with 50% data missing, the average out-of-sample R^2 is 0.339 in the regression tasks and the average out-of-sample accuracy is 86.1% in the classification tasks, compared to 0.315 and 84.4% for the best cross-validated benchmark method. In the multiple imputation setting, downstream models trained using `opt.impute` obtain a statistically significant improvement over models trained using multivariate imputation by chained equations (`mice`) in 8/10 missing data scenarios considered.

Keywords: missing data imputation, K-NN, SVM, optimal decision trees

1. Introduction

The missing data problem is arguably the most common issue encountered by machine learning practitioners when analyzing real-world data. In many applications ranging from gene expression in computational biology to survey responses in social sciences, missing data is present to various degrees. As many statistical models and machine learning algorithms rely on complete data sets, it is key to handle the missing data appropriately.

Method Name	Category	Software	Reference
Mean impute (`mean`)	Mean		Little and Rubin (1987)
Expectation-Maximization (`EM`)	EM		Dempster et al. (1977)
EM with Mixture of Gaussians and Multinomials	EM		Ghahramani and Jordan (1994)
EM with Bootstrapping	EM	`Amelia II`	Honaker et al. (2011)
K-Nearest Neighbors (`knn`)	K-NN	`impute`	Troyanskaya et al. (2001)
Sequential K-Nearest Neighbors	K-NN		Kim et al. (2004)
Iterative K-Nearest Neighbors	K-NN		Caruana (2001); Brás and Menezes (2007)
Support Vector Regression	SVR		Wang et al. (2006)
Predictive-Mean Matching (`pmm`)	LS	`MICE`	Buuren and Groothuis-Oudshoorn (2011)
Least Squares	LS		Bø et al. (2004)
Sequential Regression Multivariate Imputation	LS		Raghunathan et al. (2001)
Local-Least Squares	LS		Kim et al. (2005)
Sequential Local-Least Squares	LS		Zhang et al. (2008)
Iterative Local-Least Squares	LS		Cai et al. (2006)
Sequential Regression Trees	Tree	`MICE`	Burgette and Reiter (2010)
Sequential Random Forest	Tree	`missForest`	Stekhoven and Bühlmann (2012)
Singular Value Decomposition	SVD		Troyanskaya et al. (2001)
Bayesian Principal Component Analysis	SVD	`pcaMethods`	Oba et al. (2003); Mohamed et al. (2009)
Factor Analysis Model for Mixed Data	FA		Khan et al. (2010)

Table 1: List of Imputation Methods

In some cases, simple approaches may suffice to handle missing data. For example, complete-case analysis uses only the data that is fully known and omits all observations with missing values to conduct statistical analysis. This works well if only a few observations contain missing values, and when the data is missing completely at random, complete-case analysis does not lead to biased results (Little and Rubin, 1987). Alternately, some machine learning algorithms naturally account for missing data, and there is no need for preprocessing. For instance, CART and K-means have been adapted for problems with missing data (Breiman et al., 1984; Wagstaff, 2004).

In many other situations, missing values need to be imputed prior to running statistical analyses on the complete data set. The benefit of the latter approach is that once a set (or multiple sets) of complete data has been generated, practitioners can easily apply their own learning algorithms to the imputed data set. We focus on methods for missing data imputation in this paper.

Concretely, assume that we are given data $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ with missing entries $x_{id}, (i,d) \in \mathcal{M}$. The objective is to impute the values of the missing data that resemble the underlying complete data as closely as possible. This way, when one conducts statistical inference or pattern recognition using machine learning methods on the imputed data, the results should be similar to those obtained if full data were given. We outline some of the state-of-the-art methods for imputation in Table 1 and describe them briefly below. Part of the list is adapted from a review paper by Liew et al. (2011).

1.1 Related Work

The simplest method is mean impute, in which each missing value x_{id} is imputed as the mean of all observed values in dimension d. Mean impute underestimates the variance, ignores the correlation between the features, and thus often leads to poor imputation (Little and Rubin, 1987).

Joint modeling asserts some joint distribution on the entire data set. It assumes a parametric density function (e.g., multivariate normal) on the data given model parameters. In practice, model parameters are typically estimated using an Expectation-Maximization (EM) approach. It finds a solution (often non-optimal) of missing values and model parameters to maximize the likelihood function. Many software tools such as the R package `Amelia 2` implement the EM method with bootstrapping, assuming that the data is drawn from a multivariate normal distribution (Honaker et al., 2011). Joint modeling provides useful theoretical properties but lacks the flexibility for processing data types seen in many real applications (Van Buuren, 2007). For example, when the data includes continuous and categorical variable types, standard multivariate density functions often fail at modeling the complexity of mixed data types. However, under the assumption that the categorical variables are independent, we can use mixture models of Gaussians and Multinomials for imputation (Ghahramani and Jordan, 1994).

In contrast to joint modeling, fully conditional specification is a more flexible alternative where one specifies the conditional model for each variable; it is especially useful in mixed data types (Van Buuren, 2007). To generalize to multivariate settings, a chained equation process — initializing using random sampling and conducting univariate imputations sequentially until convergence — is typically used (Buuren and Groothuis-Oudshoorn, 2011). Each iteration is a Gibbs sampler that draws from the conditional distribution on the imputed values.

A simple example of conditional specification is based on regression. Least-Squares (LS) imputation constructs single univariate regressions, regressing features with missing values on all of the other dimensions in the data. Each missing value x_{id} is then imputed as the weighted average of these regression predictions (Bø et al., 2004; Raghunathan et al., 2001). Alternatively, in the Predictive-Mean Matching method (`pmm`), imputations are random samples drawn from a set of observed values close to regression predictions (Buuren and Groothuis-Oudshoorn, 2011). Imputation methods that use Support Vector Regression in place of LS for the regression step have also been explored (Wang et al., 2006).

When there is non-linear relationship between the variables, linear regression based imputation may perform poorly. Burgette and Reiter (2010) propose using Classification and Regression Trees (CART) as the conditional model for imputation. Extensions to random forests have also shown promising results (Stekhoven and Bühlmann, 2012). These decision tree based imputation methods are non-parametric approaches that do not rely upon distributional assumptions on the data.

One of the most commonly used non-parametric approaches is K-Nearest Neighbors (K-NN) based imputation. This method imputes each missing entry x_{id} as the mean of the dth dimension of the K-nearest neighbors that have observed values in dimension d (Troyanskaya et al., 2001). Some extensions of K-NN include sequential K-NN, which starts by imputing missing values from observations with the fewest missing dimensions and continues imputing the next unknown entries reusing the previously imputed values (Kim et al., 2004). Iterative K-NN uses an iterative process to refine the estimates and choose the nearest neighbors based on the estimates from the previous iteration (Caruana, 2001; Brás and Menezes, 2007). The Local-Least Squares method combines ideas from K-NN and LS, imputing each missing value x_{id} using regression models trained on the K-nearest neighbors of the point

$\mathbf{x}_i$ (Kim et al., 2005). Sequential and iterative variations of Local-Least Squares resemble their K-NN imputation counterparts (Zhang et al., 2008; Cai et al., 2006).

Low dimensional representation-based imputation assumes that the data represents a noisy observation of a linear combination of a small set of principal components or factor variables. In the basic method, singular value decomposition (SVD) is used on the entire data set to determine the principal eigenvectors. The missing values are imputed as a linear combination of these eigenvectors. This process is iteratively repeated until convergence (Troyanskaya et al., 2001; Mazumder et al., 2010). Bayesian Principal Component Analysis is similar to SVD imputation but extends the method to incorporate information from a prior distribution on the model parameters (Oba et al., 2003; Mohamed et al., 2009). Some recent development of a variant of the EM algorithm for factor analysis also provides a missing data imputation method for mixed data (Khan et al., 2010).

Thus far, we have only discussed methods for single imputation which generate one set of completed data that will be used for further statistical analyses. Multiple imputation, on the other hand, imputes multiple times (each set is possibly different), runs the statistical analyses on each, and pools the results (Little and Rubin, 1987). Such method is able to capture the variability in the missing data and therefore generate potentially more accurate estimates to the larger statistical problem. However, multiple imputation methods are slower and require pooling results, which may not be appropriate for certain applications.

Within the multiple imputation framework, the procedure for generating multiple estimates of missing values varies. Multivariate imputation by chained equations (`mice`), a popular multiple imputation method, generates estimates using: predictive mean matching, Bayesian linear regression, logistic regression, and others (Buuren and Groothuis-Oudshoorn, 2011). In all cases, the method initializes using random sampling and conducts univariate imputations sequentially until convergence. Each iteration is a Gibbs sampler that draws from the conditional distribution on the imputed values.

Because of its importance, missing data imputation remains an active research area. Although there are numerous methods, many of them have serious shortcomings. Joint modeling methods are not as effective when data sets violate normality assumptions, and a naïve implementation often crashes during the computation of a singular covariance matrix (Honaker et al., 2011). Some conditional specification methods such as `pmm` are practically reliable, but lack theoretical foundation and have no explicit formulation as an optimization problem. This stands in stark contrast to other areas of machine learning, where statistical models and optimization problems are deeply intertwined.

Evidence from recent literature suggests that recent advances in optimization have driven significant progress in machine learning. Integer and convex optimization have been applied successfully to median and sparse regression problems (Bertsimas and Van Parys, 2017; Bertsimas and Mazumder, 2014). Recent work on Optimal Decision Trees for classification leverages integer and robust optimization (Bertsimas and Dunn, 2017; Bertsimas et al., 2017). In this paper, we reconsider the missing data problem from this perspective, in order to develop optimization-based methods for imputation with improved out-of-sample performance.

1.2 Contributions

We summarize our contributions in this paper below:

1. We pose the missing data problem under a general optimization framework. The framework produces an optimization problem with a predictive model-based cost function that explicitly handles both continuous and categorical variables and can be used to generate multiple imputations. We present three cost functions derived from K-nearest neighbors, support vector machines, and optimal decision tree models. This optimization perspective provides fresh insight into the classical missing data problem and leads to new algorithms for more accurate data imputation.

2. For each imputation model, we derive first-order methods to find high-quality solutions to the missing data problem following a general imputation algorithm `opt.impute` presented in this paper. These methods easily scale to data sets with n in the 100,000s and p in the 1,000s on a standard desktop computer and converge within a few iterations. In addition, the first-order methods are robust and reliable for arbitrary missing patterns and mixed data types.

3. We evaluate the methods in computational experiments using 84 real-world data sets taken from the UCI Machine Learning Repository. Benchmarked against existing imputation methods including mean impute, K-nearest neighbors, iterative knn, Bayesian PCA, and predictive-mean matching, `opt.impute` produces the best overall imputation in more than 75.8% of all data sets, and results in an average reduction in mean absolute error of 8.3% against the best cross-validated benchmark method.

4. We demonstrate that the improved data imputations generated by `opt.impute` give rise to improved performance on 10 downstream classification and regression tasks. With 50% of missing data, classification models trained on data imputed via `opt.impute` have an average testing accuracy of 86.1% compared to 84.4% for the best cross-validated benchmark method. In addition, regression models trained on data imputed via `opt.impute` have an average out-of-sample R^2 value of 0.339 compared to 0.315 for the best cross-validated benchmark method. Finally, downstream models trained on multiple imputations produced by `opt.impute` significantly outperform multiple imputations produced by `mice` in 3/5 missing data scenarios for classification and 5/5 scenarios for regression.

The structure of the paper is as follows. In Section 2, we formulate the missing data imputation problem as an optimization problem, present a general first-order method `opt.impute` that can be used to find high-quality solutions, and derive the algorithms for each model: K-NN, SVM, and trees. We also discuss a cross-validation procedure and extensions of `opt.impute` to multiple imputation. In Section 3, we compare the imputation quality and performance on downstream tasks of `opt.impute` to benchmark imputation methods on a wide range of real data sets. In Section 4, we discuss the benefits from adopting such framework and suggest areas for future work. We conclude in Section 5.

2. Methods for Optimal Imputation

In this section, we pose the missing data problem as an optimization problem in which we optimize the missing values in all data points and dimensions simultaneously. We introduce a general imputation framework on mixed data (continuous and categorical) based upon first-order methods applied to this problem. Within this framework, we use K-nearest neighbors, SVM, and decision tree based imputation as examples to define three specific optimization problems. For each problem, we present two first-order methods used to find high-quality solutions: block coordinate descent (BCD) and coordinate descent (CD).

Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ be the data set given with p variables. Without loss of generality, we assume each data vector $\mathbf{x}_i$ contains continuous variables indexed by $d \in \{1, 2, \ldots, p_0\}$ and categorical variables indexed by $d \in \{p_0 + 1, \ldots, p_0 + p_1\}$ with $p_0 + p_1 = p$. As a pre-processing step, we transform all continuous variables to have unit standard deviation. We leave all categorical variables unchanged, and assume the dth categorical variable $d \in \{p_0 + 1, \ldots, p_0 + p_1\}$ takes values among k_d classes. Note that if all data is continuous $p_0 = 0$, while if all data is categorical $p_1 = 0$. The missing and known values are specified by the following sets:

$$\begin{aligned}
\mathcal{M}_0 &= \{(i,d) : \text{entry } x_{id} \text{ is missing, } 1 \le d \le p_0\},\\
\mathcal{N}_0 &= \{(i,d) : \text{entry } x_{id} \text{ is known, } 1 \le d \le p_0\},\\
\mathcal{M}_1 &= \{(i,d) : \text{entry } x_{id} \text{ is missing, } p_0 + 1 \le d \le p_0 + p_1\},\\
\mathcal{N}_1 &= \{(i,d) : \text{entry } x_{id} \text{ is known, } p_0 + 1 \le d \le p_0 + p_1\}.
\end{aligned}$$

We also refer to the full missing pattern as $\mathcal{M} := \mathcal{M}_0 \cup \mathcal{M}_1$. Let $\mathbf{W} \in \mathbb{R}^{n \times p_0}$ be the matrix of imputed continuous values, where w_{id} is the imputed value for entry x_{id}, $d \in \{1, \ldots, p_0\}$. Similarly, let $\mathbf{V} \in \{1, \ldots, k_1\} \times \ldots \times \{1, \ldots, k_{p_1}\}$ be the matrix of imputed categorical values, where v_{id} is the imputed value for entry x_{id}, $d \in \{p_0 + 1, \ldots, p_0 + p_1\}$. We refer to the full imputation for observation $\mathbf{x}_i$ as $(\mathbf{w}_i, \mathbf{v}_i)$ in the following sections.

2.1 General Problem Formulation

As the task is to impute the missing values, for each model the key decision variables are the imputed values $\{w_{id} : (i,d) \in \mathcal{M}_0\}$ and $\{v_{id} : (i,d) \in \mathcal{M}_1\}$. We also introduce auxiliary decision variables as well; denote these as $\mathbf{U}$. For instance, in a K-NN based approach, indicator variables $z_{ij}, 1 \le i, j \le n$ are introduced to identify the neighbor assignment for each pair of points $\mathbf{x}_i$, $\mathbf{x}_j$. For a given set of imputed values and a given model, there is a cost function $c(\cdot)$ associated with it. Our goal is to solve the following optimization problem:

$$\begin{aligned}
\min \quad & c(\mathbf{U}, \mathbf{W}, \mathbf{V}; \mathbf{X}) \\
\text{s.t.} \quad & w_{id} = x_{id} \quad (i,d) \in \mathcal{N}_0, \\
& v_{id} = x_{id} \quad (i,d) \in \mathcal{N}_1, \\
& (\mathbf{U}, \mathbf{W}, \mathbf{V}) \in \mathcal{U},
\end{aligned} \tag{1}$$

where $\mathcal{U}$ is the set of all feasible combinations $(\mathbf{U}, \mathbf{W}, \mathbf{V})$ of auxiliary vectors and imputations. For example, in a K-NN based approach, this includes the constraints that each

point has exactly K neighbors and the assignment variables are binary. We list the auxiliary variables and cost functions corresponding to each of the imputation models K-NN, SVM, and trees in Table 2. Note that the cost function can be different for continuous and categorical variables. We can introduce a parameter that controls the relative contribution to the cost between the continuous and categorical variables, or scale continuous variables appropriately. For the remainder of the paper the latter is assumed for simplicity of notation.

Model	U	$c(\mathbf{U}, \mathbf{W}, \mathbf{V}; \mathbf{X})$
K-NN	$\mathbf{Z}$	$\sum_{i\in\mathcal{I}}\sum_{j=1}^{n} z_{ij}\Big[\sum_{d=1}^{p_0}(w_{id}-w_{jd})^2 + \sum_{d=p_0+1}^{p_0+p_1}\mathbb{1}_{\{v_{id}\neq v_{jd}\}}\Big]$
SVM	$[\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\gamma}^*, \boldsymbol{\xi}]$	$\frac{1}{2}(\|\boldsymbol{\beta}\|_{\mathcal{H}}^2 + \|\boldsymbol{\theta}\|_{\mathcal{H}}^2) + C\sum_{i=1}^{n}\left(\sum_{d=1}^{p_0}(\gamma_{id}+\gamma_{id}^*) + \sum_{d=p_0+1}^{p_0+p_1}\xi_{id}\right)$
Trees	$\mathbf{T}$	$\sum_{i=1}^{n}\sum_{j=1}^{n}\Big[\sum_{d=1}^{p_0} t_{ij}^d(w_{id}-w_{jd})^2 + \sum_{d=p_0+1}^{p_0+p_1} t_{ij}^d\mathbb{1}_{\{v_{id}\neq v_{jd}\}}\Big]$

Table 2: Variables and cost functions for each imputation model. Variables for K-NN, SVM, and trees are defined in Sections 2.3, 2.4, and 2.5 respectively.

This problem is non-convex for K-NN, SVM, and tree models. To obtain a certifiable optimal solution, one can reformulate the problem with integer variables and solve it using a mixed integer solver. We ran computational experiments and found that solving such mixed integer problems requires a long time to reach a certifiably optimal solution. As a result, we present a general imputation algorithm `opt.impute` which approximates the solution to Problem (1) very fast using first-order methods.

2.2 First-Order Method for the General Problem

To obtain high-quality solutions to Problem (1), we can use first-order methods with random warm starts. In particular, we will focus on block coordinate descent (BCD) and coordinate descent (CD) (Bertsekas, 1999). Algorithm 1, which we refer to as `opt.impute`, implements BCD or CD for Problem (1). The variables $\mathbf{U}, \mathbf{W}, \mathbf{V}$, and $\mathbf{X}$ as well as the cost function $c(\cdot)$ are summarized in Table 2 for K-NN, SVM, and trees. The detailed solution methods for Problems (2), (3), (4), and (5) for K-NN, SVM, and tree imputation models are described in Sections 2.3-2.5, respectively.

By construction, the objective function value strictly decreases by at least δ_0 until termination. It follows that the number of steps needed for the algorithm to terminate is $\lceil \frac{1}{\delta_0} c(\mathbf{U}^0, \mathbf{W}^0, \mathbf{V}^0; \mathbf{X}) \rceil$, where $\mathbf{W}^0, \mathbf{V}^0$ are the initialization values, $\mathbf{X}$ is data, and $\mathbf{U}^0$ is the argmin in Equation (2). However, the algorithm is not guaranteed to find a global minimum for Problem (1) (Wright, 2015).

In the next sections, we discuss three example models and the optimization problem formulations. For each model and each first-order method, we derive the specific updates

Algorithm 1 `opt.impute`

Input: $\mathbf{X} \in \mathbb{R}^{n\times p_0} \times \{1,\ldots,k_1\} \times \ldots \times \{1,\ldots,k_{p_1}\}$, a data matrix with some missing entries $\mathcal{M} = \{(i,d) : x_{id} \text{ is missing}\}$, $\delta_0 > 0$, and warm start $\mathbf{W}^0 \in \mathbb{R}^{n\times p_0}$, $\mathbf{V}^0 \in \{1,\ldots,k_1\} \times \ldots \times \{1,\ldots,k_{p_1}\}$.

Output: $\mathbf{X}^{imp}$ a full matrix with imputed values.

Procedure:

Initialize $\delta \leftarrow \infty$, $\mathbf{W}^{old} \leftarrow \mathbf{W}^0$, $\mathbf{V}^{old} \leftarrow \mathbf{V}^0$.

while $\delta > \delta_0$ **do**

① Update $\mathbf{U}^*$ using model dependent information:

$$\begin{aligned} \mathbf{U}^* \leftarrow \arg\min_{\mathbf{U}} \;& c(\mathbf{U}, \mathbf{W}^{old}, \mathbf{V}^{old}; \mathbf{X}) \\ \text{s.t.} \;& (\mathbf{U}, \mathbf{W}^{old}, \mathbf{V}^{old}) \in \mathcal{U}. \end{aligned} \tag{2}$$

② Update the imputation $\mathbf{W}^*$, $\mathbf{V}^*$, following either:

②a block coordinate descent (BCD):

$$\begin{aligned} \mathbf{W}^*, \mathbf{V}^* \leftarrow \arg\min_{\mathbf{W},\mathbf{V}} \;& c(\mathbf{U}^*, \mathbf{W}, \mathbf{V}; \mathbf{X}) \\ \text{s.t.} \;& w_{id} = x_{id} \qquad (i,d) \in \mathcal{N}_0, \\ & v_{id} = x_{id} \qquad (i,d) \in \mathcal{N}_1, \\ & (\mathbf{U}^*, \mathbf{W}, \mathbf{V}) \in \mathcal{U}, \end{aligned} \tag{3}$$

or

②b coordinate descent (CD):

$$\begin{aligned} w_{jr}^* \leftarrow \arg\min_{w_{jr}} \;& c(\mathbf{U}^*, \mathbf{W}, \mathbf{V}; \mathbf{X}) \\ \text{s.t.} \;& w_{id} = x_{id} \qquad (i,d) \in \mathcal{N}_0, \\ & v_{id} = x_{id} \qquad (i,d) \in \mathcal{N}_1, \\ & w_{id} = w_{id}^* \qquad (i,d) \in \mathcal{M}_0 \backslash (j,r), \\ & v_{id} = v_{id}^* \qquad (i,d) \in \mathcal{M}_1, \\ & (\mathbf{U}^*, \mathbf{W}, \mathbf{V}) \in \mathcal{U}, \end{aligned} \tag{4}$$

$$\begin{aligned} v_{jr}^* \leftarrow \arg\min_{v_{jr}} \;& c(\mathbf{U}^*, \mathbf{W}, \mathbf{V}; \mathbf{X}) \\ \text{s.t.} \;& w_{id} = x_{id} \qquad (i,d) \in \mathcal{N}_0, \\ & v_{id} = x_{id} \qquad (i,d) \in \mathcal{N}_1, \\ & w_{id} = w_{id}^* \qquad (i,d) \in \mathcal{M}_0, \\ & v_{id} = v_{id}^* \qquad (i,d) \in \mathcal{M}_1 \backslash (j,r), \\ & (\mathbf{U}^*, \mathbf{W}, \mathbf{V}) \in \mathcal{U}. \end{aligned} \tag{5}$$

③ $\delta \leftarrow c(\mathbf{U}^*, \mathbf{W}^*, \mathbf{V}^*; \mathbf{X}) - c(\mathbf{U}^{old}, \mathbf{W}^{old}, \mathbf{V}^{old}; \mathbf{X})$.

④ $(\mathbf{U}^{old}, \mathbf{W}^{old}, \mathbf{V}^{old}) \leftarrow (\mathbf{U}^*, \mathbf{W}^*, \mathbf{V}^*)$.

end while

$\mathbf{X}^{imp} \leftarrow [\mathbf{W}^*; \mathbf{V}^*]$

for $\mathbf{U}, \mathbf{W}, \mathbf{V}$ that we use in our optimization-based imputation procedure. After, we describe a cross-validation procedure to select the specific model and parameters for the imputation.

2.3 K-NN Based Imputation

We first define a distance metric between rows $(\mathbf{w}_i, \mathbf{v}_i)$ and $(\mathbf{w}_j, \mathbf{v}_j)$ as

$$d_{ij} := \sum_{d=1}^{p_0} (w_{id} - w_{jd})^2 + \sum_{d=p_0+1}^{p_0+p_1} \mathbb{1}_{\{v_{id} \neq v_{jd}\}}. \tag{6}$$

Next, we introduce the binary variables:

$$z_{ij} = \begin{cases} 1, & \text{if } (\mathbf{w}_j, \mathbf{v}_j) \text{ is among the } K\text{-nearest neighbors of } (\mathbf{w}_i, \mathbf{v}_i) \\ & \text{with respect to distance metric (6)}, \\ 0, & \text{otherwise.} \end{cases}$$

We further define the set of indices $\mathcal{I} := \{i : \mathbf{x}_i \text{ has at least one missing coordinate}\}$. The optimization problem for the K-NN based imputation model is:

$$\begin{aligned} \min \quad & c(\mathbf{Z}, \mathbf{W}, \mathbf{V}; \mathbf{X}) := \sum_{i \in \mathcal{I}} \sum_{j=1}^{n} z_{ij} \Big[\sum_{d=1}^{p_0} (w_{id} - w_{jd})^2 + \sum_{d=p_0+1}^{p_0+p_1} \mathbb{1}_{\{v_{id} \neq v_{jd}\}} \Big] \\ \text{s.t.} \quad & w_{id} = x_{id} && (i,d) \in \mathcal{N}_0, \\ & v_{id} = x_{id} && (i,d) \in \mathcal{N}_1, \\ & z_{ii} = 0 && i \in \mathcal{I}, \\ & \sum_{j=1}^{n} z_{ij} = K && i \in \mathcal{I}, \\ & \mathbf{Z} \in \{0,1\}^{|\mathcal{I}| \times n}. \end{aligned} \tag{7}$$

By optimality, it follows that $z_{ij} = 1$ if and only if $(\mathbf{w}_j, \mathbf{v}_j)$ is among the K-nearest neighbors of $(\mathbf{w}_i, \mathbf{v}_i)$. Therefore, solving Problem (7) produces the missing value imputation which minimizes the sum of distances from each point $(\mathbf{w}_i, \mathbf{v}_i), i \in \mathcal{I}$ to its K-nearest neighbors. Note that the relation $\mathbb{1}_{\{v_{id} \neq v_{jd}\}}$ can be modeled with binary variables. Problem (7) is a nonconvex optimization problem with both continuous and binary variables. Correspondingly, it is difficult to solve to provable optimality, even if the data set contains continuous variables only.

Next, we describe the updates in Algorithm 1 for K-NN based imputation. We refer to this specific imputation method as `opt.knn`.

2.3.1 `OPT.KNN`

In step (1), to update the auxiliary variables $\mathbf{Z}$, first fix all imputed values $\mathbf{W}$, $\mathbf{V}$. Problem (2) decomposes by $i \in \mathcal{I}$ into the assignment problems:

$$\begin{aligned} \min_{\mathbf{z}_i} \quad & \sum_{j=1}^{n} z_{ij} d_{ij} \\ \text{s.t.} \quad & z_{ii} = 0, \\ & \sum_{j=1}^{n} z_{ij} = K, \\ & \mathbf{z}_i \in \{0,1\}^n. \end{aligned} \tag{8}$$

The optimal solution to Problem (8) can be found using a simple sorting procedure on the distances $\{d_{ij}\}_{j=1}^n$. For each $i \in \mathcal{I}$, we find the K-nearest neighbors of $(\mathbf{w}_i, \mathbf{v}_i)$ and set $z_{ij} = 1$ for these neighbors, $z_{ij} = 0$, otherwise.

Next, we fix $\mathbf{Z}$ and update the imputed values $\mathbf{W}, \mathbf{V}$ using either BCD or CD. In step (2a), the BCD update, Problem (3) decomposes by dimension $d = 1, \ldots, p$. For each continuous dimension $d = 1, \ldots, p_0$, we consider the following quadratic optimization problem:

$$\begin{aligned} \min_{\mathbf{w}^d} \quad & \sum_{i \in \mathcal{I}} \sum_{j=1}^{n} z_{ij} (w_{id} - w_{jd})^2 \\ \text{s.t.} \quad & w_{id} = x_{id} && (i,d) \in \mathcal{N}_0, \end{aligned}$$

where $\mathbf{w}^d \in \mathbb{R}^n$ are the imputed values in the dth dimension. Taking partial derivative of the objective function with respect to w_{id} for some missing entry $(i,d) \in \mathcal{M}_0$ and setting it to zero, we obtain after some simplifications:

$$(K + \sum_{j \in \mathcal{I}} z_{ji}) w_{id} - \sum_{(j,d) \in \mathcal{M}_0} (z_{ij} + z_{ji}) w_{jd} - \sum_{(j,d) \in \mathcal{N}_0} (z_{ij} + \mathbb{1}_{\{j \in \mathcal{I}\}} z_{ji}) x_{jd} = 0. \tag{9}$$

For each continuous dimension d, we have a system of equations of the form (9) which we can solve to determine the optimal imputed values $w_{id}, (i,d) \in \mathcal{M}_0$. To simplify notation, suppose that the missing values for dimension d are $\widetilde{\mathbf{w}} := (\widetilde{w}_{1d}, \ldots, \widetilde{w}_{ad})$ and the known values are $\widetilde{\mathbf{x}} := (\widetilde{x}_{(a+1)d}, \ldots, \widetilde{x}_{nd})$. Then, the set of optimal imputed missing values $\widetilde{\mathbf{w}}$ is the solution to the linear system $\mathbf{Q}\widetilde{\mathbf{w}} = \mathbf{R}\widetilde{\mathbf{x}}$, where

$$\mathbf{Q} = \begin{bmatrix} K + \sum_{j \in \mathcal{I}} z_{j1} - 2z_{11} & -z_{12} - z_{21} & \cdots & -z_{1a} - z_{a1} \\ -z_{21} - z_{12} & K + \sum_{j \in \mathcal{I}} z_{j2} - 2z_{22} & \cdots & -z_{2a} - z_{a2} \\ \vdots & & \ddots & \vdots \\ -z_{a1} - z_{1a} & -z_{a2} - z_{2a} & \cdots & K + \sum_{j \in \mathcal{I}} z_{ja} - 2z_{aa} \end{bmatrix},$$

$$\mathbf{R} = \begin{bmatrix} z_{1(a+1)} + \mathbb{1}_{\{(a+1)\in\mathcal{I}\}} z_{(a+1)1} & \cdots & z_{1n} + \mathbb{1}_{\{n\in\mathcal{I}\}} z_{n1} \\ \vdots & & \vdots \\ z_{a(a+1)} + \mathbb{1}_{\{(a+1)\in\mathcal{I}\}} z_{(a+1)a} & \cdots & z_{an} + \mathbb{1}_{\{n\in\mathcal{I}\}} z_{na} \end{bmatrix}.$$

Note that when K is sufficiently large, the matrix $\mathbf{Q}$ is positive semidefinite and therefore invertible. If $\mathbf{Q}$ is singular, then we may add a small positive perturbation to the diagonal of $\mathbf{Q}$ so that the matrix becomes positive semidefinite. Therefore, without loss of generality there is a closed-form solution $\widetilde{\mathbf{w}} = \mathbf{Q}^{-1}\mathbf{R}\widetilde{\mathbf{x}}$ to this system of equations for each continuous dimension d.

In order to update $\mathbf{V}$, we solve the following integer linear optimization problem for each categorical dimension $d = (p_0+1), \ldots, p$:

$$\begin{aligned} \min_{\mathbf{v}^d} \quad & \sum_{i\in\mathcal{I}} \sum_{j=1}^{n} z_{ij} y_{ij} \\ \text{s.t.} \quad & v_{id} = x_{id} && (i,d) \in \mathcal{N}_1, \\ & v_{id} - v_{jd} \le y_{ij} k_d && i = 1,\ldots,n, j = 1,\ldots,n, \\ & v_{jd} - v_{id} \le y_{ij} k_d && i = 1,\ldots,n, j = 1,\ldots,n, \\ & y_{ij} \in \{0,1\}^{n\times n}, \end{aligned}$$

where $\mathbf{v}^d \in \{1,\ldots,k_d\}^n$ are the imputed values for the dth dimension. Here, the indicator variables y_{ij} take values equal to $\mathbb{1}_{\{v_{jd}\neq v_{jd}\}}$ in the optimal solution.

In step (2b), following the CD method, we update the missing imputed values one at a time. Each $w_{id}, (i,d) \in \mathcal{M}_0$ is imputed as the minimizer of the following:

$$\min_{w_{id}} \quad \sum_{j=1}^{n} z_{ij}(w_{id} - w_{jd})^2 + \sum_{j\in\mathcal{I}} z_{ji}(w_{jd} - w_{id})^2.$$

Solving the above gives

$$w_{id} = \frac{\sum_{j=1}^{n} z_{ij} w_{jd} + \sum_{j\in\mathcal{I}} z_{ji} w_{jd}}{K + \sum_{j\in\mathcal{I}} z_{ji}}. \tag{10}$$

We can interpret the missing value imputation (10) as a weighted average of the K nearest neighbors of $\mathbf{x}_i$, along with all points $\mathbf{x}_j$ which include $\mathbf{x}_i$ as a neighbor. Similarly, each categorical variable $v_{id}, (i,d) \in \mathcal{M}_1$ is imputed as the minimizer of the following:

$$\min_{v_{id}} \quad \sum_{j=1}^{n} z_{ij} \mathbb{1}_{\{v_{id}\neq v_{jd}\}} + \sum_{j\in\mathcal{I}} z_{ji} \mathbb{1}_{\{v_{jd}\neq v_{jd}\}}.$$

The solution is

$$v_{id} = \text{mode}\big(\{\{v_{jd} : z_{ij} = 1\}, \{v_{jd} : z_{ji} = 1\}\}\big).$$

Here, we set v_{id} to be the highest frequency category among the K nearest neighbors of $\mathbf{x}_i$, along with all points $\mathbf{x}_j$ which include $\mathbf{x}_i$ as a nearest neighbor. In practice, we use this update for $v_{id}, (i,d) \in \mathcal{M}_1$ in place of the update for $\mathbf{V}$ in BCD because it is much faster computationally.

2.4 Mixed SVM Based Imputation

In this section, we consider a second model for imputation, based upon SVM regression for imputing continuous features and SVM classification for imputing categorical features. First, define $\widetilde{\mathbf{v}}_i \in \{-1,1\}^{p_2}$ to be a dummy encoded representation of $\mathbf{v}_i$, where $p_2 = \sum_{d=p_0+1}^{p_0+p_1} k_d - p_1$. Let $\widetilde{v}_{id}^{fixed}, (i,d) \in \mathcal{N}_2$ be the known dummy encoded values. For each continuous feature $d \in \{1,\ldots,p_0\}$, let $(\boldsymbol{\beta}_d, \beta_{d0}) \in \mathbb{R}^{p_0+p_2+1}$ be the coefficients for an SVM regression model regressing feature d on the other features with the dummy encoding. Let $(\boldsymbol{\theta}_d, \theta_{d0}) \in \mathbb{R}^{p_0+p_2+1}$ be the coefficients for an SVM classification model predicting dummy feature d based upon the other features. Note that it is also possible to use a multi-class SVM model to predict each categorical feature directly, as described by Crammer and Singer (2001), using parameters of the form $\mathbf{M} \in \mathbb{R}^{k_d\times(p_0+p_2+1)}$ for each feature $d \in \{p_0+1,\ldots,p_0+p_1\}$. In this case, we would keep the dummy encoded decision variables as covariates to predict the other features and add constraints relating $v_{id}, (i,d) \in \mathcal{M}_1$ and $\widetilde{v}_{id}, (i,d) \in \mathcal{M}_2$. For illustrative purposes and simplicity of notation, we present the formulation using binary SVM to predict each dummy variable d.

We consider the following optimization problem:

$$\begin{aligned} \min \quad & c([\boldsymbol{\beta},\boldsymbol{\theta}], \mathbf{W}, \widetilde{\mathbf{V}}; \mathbf{X}) := \frac{1}{2}\left(\|\boldsymbol{\theta}\|^2 + \|\boldsymbol{\beta}\|^2\right) + C\left(\sum_{i=1}^{n}\sum_{d=1}^{p_0}(\gamma_{id} + \gamma_{id}^*) + \sum_{i=1}^{n}\sum_{d=p_0+1}^{p_0+p_1}\xi_{id}\right) \\ \text{s.t.} \quad & x_{id} = w_{id} && (i,d) \in \mathcal{N}_0, \\ & \widetilde{v}_{id} = \widetilde{v}_{id}^{fixed} && (i,d) \in \mathcal{N}_2, \\ & \beta_{dd} = 0 && d = 1,\ldots,p_0, \\ & \theta_{dd} = 0 && d = 1,\ldots,p_2, \\ & \gamma_{id} \ge w_{id} - (\boldsymbol{\beta}_d^T \begin{bmatrix} \mathbf{w}_i \\ \widetilde{\mathbf{v}}_i \end{bmatrix} + \beta_{d0}) - \epsilon && d = 1,\ldots,p_0, i = 1\ldots,n, \\ & \gamma_{id}^* \ge (\boldsymbol{\beta}_d^T \begin{bmatrix} \mathbf{w}_i \\ \widetilde{\mathbf{v}}_i \end{bmatrix} + \beta_{d0}) - w_{id} - \epsilon && d = 1,\ldots,p_0, i = 1\ldots,n, \\ & \xi_{id} \ge 1 - \widetilde{v}_{id}(\boldsymbol{\theta}_d^T \begin{bmatrix} \mathbf{w}_i \\ \widetilde{\mathbf{v}}_i \end{bmatrix} + \theta_{d0}) && d = 1,\ldots,p_2, i = 1\ldots,n, \\ & \gamma_{id} \ge 0 && d = 1,\ldots,p_0, i = 1\ldots,n, \\ & \gamma_{id}^* \ge 0 && d = 1,\ldots,p_0, i = 1\ldots,n, \\ & \xi_{id} \ge 0 && d = 1,\ldots,p_2, i = 1\ldots,n, \\ & \widetilde{v}_{id} \in \{-1,1\} && d = 1,\ldots,p_2, i = 1\ldots,n. \end{aligned} \tag{11}$$

This formulation is based upon SVM with a linear kernel; however we can extend Problem (11) to arbitrary kernels, including the multi-class cases, using the modified objective function

$$c([\boldsymbol{\beta},\boldsymbol{\theta}], \mathbf{W}, \mathbf{V}; \mathbf{X}) := \frac{1}{2}(\|\boldsymbol{\beta}\|_{\mathcal{H}}^2 + \|\boldsymbol{\theta}\|_{\mathcal{H}}^2) + C\left(\sum_{i=1}^{n}\sum_{d=1}^{p_0}(\gamma_{id} + \gamma_{id}^*) + \sum_{i=1}^{n}\sum_{d=p_0+1}^{p_0+p_1}\xi_{id}\right),$$

where $\|\cdot\|_{\mathcal{H}}$ is the norm in a given Reproducing Kernel Hilbert Space $\mathcal{H}$.

Another important aspect of Problem (11) is the compound objective function, which is the summation of objective functions derived from both SVM regression and SVM classification methods. Observe that if we fix a single imputed entry w_{id} or $\widetilde{v}_{id}$, the contribution to the objective function scales linearly as $(\boldsymbol{\beta}_d^T \begin{bmatrix} \mathbf{w}_i \\ \widetilde{\mathbf{v}}_i \end{bmatrix} + \beta_{d0})$ if d is continuous or scales linearly as $(\boldsymbol{\theta}_d^T \begin{bmatrix} \mathbf{w}_i \\ \widetilde{\mathbf{v}}_i \end{bmatrix} + \theta_{d0})$ if d is categorical. This is desirable because we do not wish to weight continuous and categorical variables unequally in our imputation. Next, we describe the updates in Algorithm 1 for mixed SVM based imputation, which we refer to as `opt.svm`.

2.4.1 `opt.svm`

In step (1), we fix the imputed values $\mathbf{W}, \mathbf{V}$ and update the auxiliary variables $[\boldsymbol{\beta}, \boldsymbol{\beta}_0, \boldsymbol{\theta}, \boldsymbol{\theta}_0]$. Independent of the choice of kernel, Problem (2) decomposes by dimension p into p_0 SVM regression problems and p_2 SVM classification problems for the categorical variables. For each continuous feature $d \in \{1, \ldots, p_0\}$, we update $\boldsymbol{\beta}_d, \beta_{d0}$ by solving

$$\begin{aligned}
\min \quad & \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{n}(\gamma_{id} + \gamma_{id}^*) \\
\text{s.t.} \quad & \beta_{dd} = 0 \\
& \gamma_{id} \geq w_{id} - (\boldsymbol{\beta}_d^T \begin{bmatrix} \mathbf{w}_i \\ \widetilde{\mathbf{v}}_i \end{bmatrix} + \beta_{d0}) - \epsilon \quad i = 1 \ldots, n, \\
& \gamma_{id}^* \geq (\boldsymbol{\beta}_d^T \begin{bmatrix} \mathbf{w}_i \\ \widetilde{\mathbf{v}}_i \end{bmatrix} + \beta_{d0}) - w_{id} - \epsilon \quad i = 1 \ldots, n, \\
& \gamma_{id} \geq 0 \qquad\qquad\qquad\qquad i = 1, \ldots, n, \\
& \gamma_{id}^* \geq 0 \qquad\qquad\qquad\qquad i = 1, \ldots, n.
\end{aligned} \tag{12}$$

Similarly, for each dummy feature $d \in \{p_0 + 1, \ldots, p_0 + p_2\}$, we update $\boldsymbol{\theta}_d, \theta_{d0}$ by solving

$$\begin{aligned}
\min \quad & \frac{1}{2}\|\boldsymbol{\theta}\|^2 + C\sum_{i=1}^{n}\xi_{id} \\
\text{s.t.} \quad & \theta_{dd} = 0 \\
& \xi_{id} \geq 1 - \widetilde{v}_{id}(\boldsymbol{\theta}_d^T \begin{bmatrix} \mathbf{w}_i \\ \widetilde{\mathbf{v}}_i \end{bmatrix} + \theta_{d0}) \quad i = 1 \ldots, n, \\
& \xi_{id} \geq 0 \qquad\qquad\qquad\qquad i = 1, \ldots, n.
\end{aligned} \tag{13}$$

Taking the Lagrangian duals, both Problems (12) and (13) can be reformulated as quadratic optimization problems which can be solved efficiently (Cortes and Vapnik, 1995).

Next, we fix the auxiliary variables $[\boldsymbol{\beta}, \boldsymbol{\beta}_0, \boldsymbol{\theta}, \boldsymbol{\theta}_0]$ and update the imputed values $\mathbf{W}, \mathbf{V}$ using BCD or CD. In step (2a), Problem (2) decomposes by observation i into n nonlinear

integer optimization problems. For each i we solve

$$\begin{aligned}
\min_{\mathbf{w}_i, \widetilde{\mathbf{v}}_i} \quad & \sum_{d=1}^{p_0}(\gamma_{id} + \gamma_{id}^*) + \sum_{d=p_0+1}^{p_0+p_1}\xi_{id} \\
\text{s.t.} \quad & x_{id} = w_{id} \qquad\qquad\qquad\qquad (i,d) \in \mathcal{N}_0, \\
& \gamma_{id} \geq w_{id} - (\boldsymbol{\beta}_d^T \begin{bmatrix} \mathbf{w}_i \\ \widetilde{\mathbf{v}}_i \end{bmatrix} + \beta_{d0}) - \epsilon \quad d = 1, \ldots, p_0, \\
& \gamma_{id}^* \geq (\boldsymbol{\beta}_d^T \begin{bmatrix} \mathbf{w}_i \\ \widetilde{\mathbf{v}}_i \end{bmatrix} + \beta_{d0}) - w_{id} - \epsilon \quad d = 1, \ldots, p_0, \\
& \xi_{id} \geq 1 - \widetilde{v}_{id}(\boldsymbol{\theta}_d^T \begin{bmatrix} \mathbf{w}_i \\ \widetilde{\mathbf{v}}_i \end{bmatrix} + \theta_{d0}) \qquad d = 1, \ldots, p_2, \\
& \gamma_{id} \geq 0 \qquad\qquad\qquad\qquad d = 1, \ldots, p_0, \\
& \gamma_{id}^* \geq 0 \qquad\qquad\qquad\qquad d = 1, \ldots, p_0, \\
& \xi_{id} \geq 0 \qquad\qquad\qquad\qquad d = 1, \ldots, p_2,
\end{aligned} \tag{14}$$

where $(\mathbf{w}_i, \widetilde{\mathbf{v}}_i) \in \mathbb{R}^{p_0} \times \{-1, 1\}^{p_2}$ is the imputation for observation $\mathbf{x}_i$. Note that if all features are continuous, Problem (14) reduces to a linear optimization problem. Because we are using the dummy encoding in this formulation, it is possible to obtain an imputation in which multiple classes are selected for a single categorical entry. In this case, when `opt.svm` terminates, we select the imputation among the set of potential candidates which minimizes the objective function of Problem (14).

In step (2b), we update the imputed values one at a time. To update $w_{id}, (i,d) \in \mathcal{M}_0$, we solve the one-dimensional linear optimization problem:

$$\begin{aligned}
\min_{w_{id}} \quad & \sum_{d=1}^{p_0}(\gamma_{id} + \gamma_{id}^*) + \sum_{d=p_0+1}^{p_0+p_1}\xi_{id} \\
\text{s.t.} \quad & \gamma_{id} \geq w_{id} - (\boldsymbol{\beta}_d^T \begin{bmatrix} \mathbf{w}_i \\ \widetilde{\mathbf{v}}_i \end{bmatrix} + \beta_{d0}) - \epsilon \quad d = 1, \ldots, p_0, \\
& \gamma_{id}^* \geq (\boldsymbol{\beta}_d^T \begin{bmatrix} \mathbf{w}_i \\ \widetilde{\mathbf{v}}_i \end{bmatrix} + \beta_{d0}) - w_{id} - \epsilon \quad d = 1, \ldots, p_0, \\
& \xi_{id} \geq 1 - \widetilde{v}_{id}(\boldsymbol{\theta}_d^T \begin{bmatrix} \mathbf{w}_i \\ \widetilde{\mathbf{v}}_i \end{bmatrix} + \theta_{d0}) \qquad d = 1, \ldots, p_2, \\
& \gamma_{id} \geq 0 \qquad\qquad\qquad\qquad d = 1, \ldots, p_0, \\
& \gamma_{id}^* \geq 0 \qquad\qquad\qquad\qquad d = 1, \ldots, p_0, \\
& \xi_{id} \geq 0 \qquad\qquad\qquad\qquad d = 1, \ldots, p_2.
\end{aligned}$$

We update $\widetilde{v}_{id}, (i,d) \notin \mathcal{N}_2$ by solving the binary optimization problem:

$$\begin{aligned}
\min_{\widetilde{v}_{id} \in \{-1,1\}} \quad & \sum_{i=1}^{n}\sum_{d=1}^{p_0}\big(\max\{w_{id} - (\boldsymbol{\beta}_d^T \begin{bmatrix} \mathbf{w}_i \\ \widetilde{\mathbf{v}}_i \end{bmatrix} + \beta_{d0}) - \epsilon, 0\} + \max\{(\boldsymbol{\beta}_d^T \begin{bmatrix} \mathbf{w}_i \\ \widetilde{\mathbf{v}}_i \end{bmatrix} + \beta_{d0}) - w_{id} - \epsilon, 0\}\big) + \\
& \sum_{i=1}^{n}\sum_{d=1}^{p_2}\big(1 - \widetilde{v}_{id}(\boldsymbol{\theta}_d^T \begin{bmatrix} \mathbf{w}_i \\ \widetilde{\mathbf{v}}_i \end{bmatrix} + \theta_{d0})\big).
\end{aligned}$$

2.5 Tree Based Imputation

Finally, we consider an imputation model based on classification and regression trees. For each dimension we train a decision tree to predict the missing values, using the other features as covariates. We train regression trees to predict each of the continuous variables and classification trees to predict each of the categorical variables. Given a regression tree for continuous dimension d, we will impute $x_{id}, (i,d) \in \mathcal{M}_0$ to be the mean in dimension d of all points in the same leaf node as $\mathbf{x}_i$. Similarly, given a classification tree for dimension d, we will impute $x_{id}, (i,d) \in \mathcal{M}_1$ to be the mode in dimension d of all points in the same leaf node as $\mathbf{x}_i$.

For general prediction tasks, we can use greedy (Breiman et al., 1984) or globally optimal (Bertsimas and Dunn, 2017) solution methods to train the decision trees. In this case, we consider the latter approach because it admits a clear optimization model with mixed integer decision variables which fits into our framework for imputation. For each dimension d, let $\mathbf{T}^d \in \{0,1\}^{n \times n}$ denote the set of indicator variables

$$t_{ij}^d = \begin{cases} 1, & \text{if } (\mathbf{w}_i, \mathbf{v}_i), (\mathbf{w}_j, \mathbf{v}_j) \text{ are in the same leaf node} \\ & \text{of the decision tree for dimension } d, \\ 0, & \text{otherwise.} \end{cases}$$

Let $(\mathbf{T}^d, \mathbf{W}, \mathbf{V}) \in \mathcal{T}^d$ denote the set of optimal decision tree constraints for dimension d as described in (Bertsimas and Dunn, 2017). We consider the following optimization problem:

$$\begin{aligned} \min \quad & c(\mathbf{T}, \mathbf{W}, \mathbf{V}; \mathbf{X}) := \sum_{i=1}^n \sum_{j=1}^n \Big[\sum_{d=1}^{p_0} t_{ij}^d (w_{id} - w_{jd})^2 + \sum_{d=p_0+1}^{p_0+p_1} t_{ij}^d \mathbb{1}_{\{v_{id} \neq v_{jd}\}} \Big] \\ \text{s.t.} \quad & w_{id} = x_{id} && (i,d) \in \mathcal{N}_0, \\ & v_{id} = x_{id} && (i,d) \in \mathcal{N}_1, \\ & (\mathbf{T}^d, \mathbf{W}, \mathbf{V}) \in \mathcal{T}^d && d = 1, \ldots, p, \end{aligned} \tag{15}$$

Next, we describe the updates in Algorithm 1 for decision tree based imputation, which we refer to as `opt.tree`.

2.5.1 `opt.tree`

In step ①, we fix the imputed values $\mathbf{W}, \mathbf{V}$ and update the decision tree variables $\mathbf{T}$. For each continuous feature, we fit a regression tree to predict $\mathbf{w}^d$ based upon the other features. Similarly, for each categorical feature, we fit a classification tree to predict $\mathbf{v}^d$ based upon the other features. In practice, we may use greedy or optimal methods to find these trees; however, if we use greedy trees then the objective function value $c(\mathbf{T}, \mathbf{W}, \mathbf{V}; \mathbf{X})$ is not guaranteed to be monotonically decreasing over the course of the algorithm.

Next, we fix $\mathbf{T}$ and update the imputed values $\mathbf{W}, \mathbf{V}$ using BCD or CD. In step (2a), Problem (3) decomposes by dimension into p_0 quadratic optimization problems and p_1

integer optimization problems. For each continuous dimension $d = 1, \ldots, p_0$, we solve:

$$\begin{aligned} \min_{\mathbf{w}^d} \quad & \sum_{i=1}^n \sum_{j=1}^n t_{ij}^d (w_{id} - w_{jd})^2 \\ \text{s.t.} \quad & w_{id} = x_{id} && (i,d) \in \mathcal{N}_0, \end{aligned}$$

where $\mathbf{w}^d \in \mathbb{R}^n$ are the imputed values in the dth dimension. This is a quadratic optimization problem with an explicit optimum. For each $w_{id}, (i,d) \in \mathcal{M}_0$, an optimal solution is

$$w_{id} = \begin{cases} \dfrac{\sum_{(j,d) \in \mathcal{N}_0^d} t_{ij}^d x_{jd}}{\sum_{(j,d) \in \mathcal{N}_0^d} t_{ij}^d}, & \text{if} \quad \sum_{(j,d) \in \mathcal{N}_0^d} t_{ij}^d \geq 1, \\ \dfrac{1}{|\mathcal{N}_0^d|} \displaystyle\sum_{(j,d) \in \mathcal{N}_0^d} x_{jd}, & \text{otherwise,} \end{cases}$$

where $\mathcal{N}_0^d := \{(i,r) \in \mathcal{N}_0 : r = d\}$. This solution corresponds to setting each missing entry equal to the mean of all observed values in the same leaf node. If the number of non-missing values in the same leaf node as w_{id} is zero, i.e., $\sum_{(j,d) \in \mathcal{N}_0^d} t_{ij}^d = 0$, then we set all of the values in that leaf node to the mean impute solution.

For each categorical dimension $d = p_0 + 1, \ldots, p_0 + p_1$, we solve the following integer optimization problem:

$$\begin{aligned} \min_{\mathbf{v}^d} \quad & \sum_{i=1}^n \sum_{j=1}^n t_{ij}^d \mathbb{1}_{\{v_{id} \neq v_{jd}\}} \\ \text{s.t.} \quad & v_{id} = x_{id}, && (i,d) \in \mathcal{N}_1, \end{aligned}$$

where $\mathbf{v}^d \in \{1, \ldots, k_d\}^n$ are the imputed values for the dth dimension. An optimal solution is

$$v_{id} = \begin{cases} \text{mode}\big(\{x_{jd} : t_{ij}^d = 1, (j,d) \in \mathcal{N}_1\}\big) & \text{if} \quad |\{x_{jd} : t_{ij}^d = 1, (j,d) \in \mathcal{N}_1\}| \geq 1, \\ \text{mode}\big(\{x_{jd} : (j,d) \in \mathcal{N}_1\}\big) & \text{otherwise.} \end{cases}$$

In step (2b), we update the missing imputed values one at a time, which results in slightly different closed form solutions for $w_{id}, (i,d) \in \mathcal{M}_0$ and $v_{id}, (i,d) \in \mathcal{M}_1$. First, we update the continuous variables $w_{id}, (i,d) \in \mathcal{M}_0$ by solving:

$$\min_{w_{id}} \quad 2 \sum_{j=1}^n t_{ij}^d (w_{id} - w_{jd})^2. \tag{16}$$

An optimal solution to Problem (16) is

$$w_{id} = \begin{cases} \dfrac{\sum_{j \neq i} t_{ij}^d w_{jd}}{\sum_{j \neq i} t_{ij}^d}, & \text{if} \quad \sum_{j \neq i} t_{ij}^d \geq 1, \\ \dfrac{1}{|\mathcal{N}_0^d|} \displaystyle\sum_{(j,d) \in \mathcal{N}_0^d} x_{jd}, & \text{otherwise.} \end{cases}$$

Next, we update the categorical variables $v_{id}, (i,d) \in \mathcal{M}_1$ one at a time by solving:

$$\min_{v_{id}} \quad 2\sum_{j=1}^{n} t_{ij}^{d} \mathbb{1}_{\{v_{id} \neq v_{jd}\}}. \tag{17}$$

An optimal solution to Problem (17) is

$$v_{id} = \begin{cases} \text{mode}\big(\{v_{jd} : t_{ij}^{d} = 1\}\big), & \text{if} \quad |\{v_{jd} : t_{ij}^{d} = 1\}| \geq 1, \\ \text{mode}\big(\{x_{jd} : (j,d) \in \mathcal{N}_1\}\big), & \text{otherwise.} \end{cases}$$

Both of these updates coincide with the predicted values from the decision trees constructed.

2.6 Model Selection Procedure

Each of the above methods and choice of hyperparameters generates some imputed values. For single imputation, a single set of imputed values should be generated in the end. We propose the following procedure for model selection.

Given $\mathbf{X}$ with existing missing data $\mathcal{M}_0, \mathcal{M}_1$, we generate an additional fixed percentage of data missing $\mathcal{M}_0^{valid}, \mathcal{M}_1^{valid}$, with the known values as the hold-out set, and perform each of the imputation methods under the combined missing pattern. We evaluate the imputation quality on the hold-out validation set by measuring how closely the imputed values resemble the ground truth values. In particular, the mean absolute error (MAE) between true and imputed values for each imputation method is calculated. The validation MAE is defined to be

$$\frac{1}{|\mathcal{M}_0^{valid}|} \sum_{(i,d) \in \mathcal{M}_0^{valid}} |w_{id} - x_{id}| + \frac{1}{|\mathcal{M}_1^{valid}|} \sum_{(i,d) \in \mathcal{M}_1^{valid}} \mathbb{1}_{\{v_{id} \neq x_{id}\}}.$$

Lower values indicate closer imputation, and perfect imputation corresponds to an MAE of zero. Another metric of imputation quality is root mean squared error (RSME), which is given by

$$\sqrt{\frac{1}{|\mathcal{M}_0^{valid}|} \sum_{(i,d) \in \mathcal{M}_0^{valid}} (w_{id} - x_{id})^2 + \frac{1}{|\mathcal{M}_1^{valid}|} \sum_{(i,d) \in \mathcal{M}_1^{valid}} \mathbb{1}_{\{v_{id} \neq x_{id}\}}}.$$

For each imputation method, the combination of hyperparameters that achieves the lowest MAE in validation (or RMSE) is selected, and the $\mathbf{X}$ is again imputed but under the original missing patterns $\mathcal{M}_0$, $\mathcal{M}_1$. This set of imputed values is now ready to be evaluated or used for downstream tasks.

The hyperparameters that we tune via this method are summarized in Table 3. In addition, we also use this cross-validation procedure to select the best method out of `opt.knn`, `opt.svm`, and `opt.tree`. We refer to this composite method as `opt.cv`. Similarly, we may use the cross-validation procedure for model selection for any set of imputations. We define `benchmark.cv` to be the procedure that selects the best method out of: `mean`, `pmm`, `bpca`, `knn`, and `iknn` that will be later used in computational comparisons (see Section 3.1 for descriptions of these individual methods).

Method	Hyperparameters
K-NN	K
SVM	C, σ^2
Trees	cp

Table 3: Hyperparameters tuned via the model selection procedure outlined in Section 2.6. σ^2 is a parameter in the radial basis function kernel, $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{\sigma^2})$. cp is a complexity parameter related to the depth of the decision tree.

2.7 Extensions to Multiple Imputation

Thus far, we have described `opt.impute` methods for single imputation which output a single completed data set. On the other hand, multiple imputation methods output $m \geq 2$ different completed data sets for a single missing data problem. Afterwards, analysis is performed on each of the m data sets separately, and the results are pooled (Little and Rubin, 1987). For some applications, multiple imputation is preferred because it captures the variation in missing data imputation, which enables us to compute confidence intervals for downstream models trained on the imputed data sets. In addition, the pooled results from models fit on multiple imputed data sets may provide better point estimates than models fit on a single imputed data set in some cases.

To extend `opt.impute` to produce multiple imputations, we generate m warm starts using a probabilistic procedure, run `opt.knn`, `opt.svm`, or `opt.tree` from these starting points, and output the full set of m completed data sets. These warm starts can be generated from sample draws under a previously estimated posterior distribution; an example would be using outputs from the `mice` procedure. This provides us with a representative set of imputations found by the `opt.impute` algorithm, which converges to local optima. We refer to the multiple imputation method as `opt.mi`. In the computational experiments, we use the benchmark multiple imputation method `mice` to generate the warm starts.

Note that there are other possible ways of adapting `opt.impute` to the multiple imputation schema. We may introduce m instances of artificial noise in the observed values, and solve the resulting optimization problems. Alternatively, we may run `opt.impute` on m bootstrapped samples of the original data set. Afterwards, we can analyze each of the m imputed data sets separately and pool the results as before.

3. Real-World Data Experiments

In this section, we evaluate the performance of `opt.impute` on many real-world data sets. Our comparisons include 1) the effect on imputation accuracy, and 2) the effect on the performance of downstream machine learning tasks. We compare to the most commonly used state-of-the-art methods on a large sample of data sets from the UCI Machine Learning Repository. For data sets that include categorical variables, we impute the discrete values directly using our specialized imputation methods for categorical variables and benchmark methods.

3.1 Experimental Setup

To test the accuracy of the proposed missing data imputation method, we run a series of computational experiments on data sets taken from the UCI Machine Learning Repository for both regression and classification tasks. The data sets cover a range of number of observations n and number of features p, potentially mixed with both continuous and categorical variables. The numbers of continuous (p_0) and categorical (p_1) variables in each of these data sets are given in Table 10.

In these experiments, we use full data sets in which all entries are known, and we generate patterns of missing data for various percentages ranging from 10% to 50%. We take the full data sets $\mathbf{X}$ that have no missing entries to be the ground truth. We run some of the most commonly-used and state-of-the-art methods for data imputation on these data sets to predict the missing values and compare against our optimization based imputation methods. The individual methods in this comparison are:

1. **Mean Impute** (`mean`): The simplest imputation method. For each missing value x_{id}, imputes the mean of all known values in dimension d.

2. **Predictive-Mean Matching** (`pmm`): An iterative method which imputes missing values from known values in a given dimension using linear regressions. It is commonly used for multiple imputation and can be generalized to multiple missing dimensions using the chained equations process (Buuren and Groothuis-Oudshoorn, 2011). Implemented using the `MICE` package in R.

3. **Bayesian PCA** (`bpca`): A missing data estimation method based on Bayesian principal component analysis (Oba et al., 2003). Implemented using the `pcaMethods` package in R.

4. K-**Nearest Neighbors** (`knn`): A single-step, greedy method which imputes missing values using the K-nearest neighbors of an observation based upon Euclidean distance. The candidate neighbors must have non-missing values in the imputed feature. Averaged distance is used if some other coordinates are missing. Implemented using the `impute` package in R.

5. **Iterative** K-**Nearest Neighbors** (`iknn`): Implemented in R and Julia, based on the description in the original papers (Brás and Menezes, 2007; Caruana, 2001) .

6. **Optimal Impute** (`opt.impute`): All sub-methods below use warm starts including: `mean`, `knn`, `bpca` and five random starts where the values are imputed by a random sampling of the non-missing observations of that feature. The imputation which results in the lowest objective value is selected for each method.

   (a) K-NN based (`opt.knn`): This method solves the optimal K-nearest neighbors problem (7). Convergence time depends upon the quality of the initial warm start. We run both block coordinate descent and coordinate descent for small data sets of size $n \leq 10{,}000$, and only coordinate descent for large data sets with higher n. The implementation was in the programming language Julia with fast algorithms for K-nearest neighbor calculations.

   (b) SVM Regression and Classification based (`opt.svm`): This method solves the maximum margin support vector machine problem (11) using a radial basis function kernel. For continuous variables, we use ϵ-support vector regression; for categorical variables, we use classical support vector machines. These problems were solved using coordinate descent methods. The implementation was in Julia using the `scikit-learn` package in Python.

   (c) Decision Tree based (`opt.tree`): This method solves the optimal decision-tree problem (15). For continuous variables, a single-leaf regularized regression tree is used; for categorical variables, a fast coordinate descent-based algorithm for solving Optimal Classification Trees is used (Bertsimas and Dunn, 2017). We run coordinate descent for the imputation problems. The implementation was in Julia using the packages `glmnet` and `OptimalTrees`.

In addition, we consider two composite methods: `opt.cv`, which selects the best method from `opt.knn`, `opt.svm`, and `opt.tree`; and `benchmark.cv`, which selects the best method from `mean`, `pmm`, `bpca`, `knn`, and `iknn`. These composite methods use the cross-validation procedure described in Section 2.6. To generate the validation set for each missing data problem, we randomly sample an additional 10% of the entries to be hidden under the MCAR assumption. After running each individual method, we select the one that gives the lowest MAE on the validation set. We re-run this method on the original missing data set to obtain the final imputation.

Each imputation method was run for a maximum time limit of 12 hours on each data set. The quality of the imputations is evaluated using the same MAE and RMSE metrics defined in Section 2.6. For each of the `opt.impute` methods, we also record and present the convergence in objective value and MAE to show the progress over the iterations.

3.1.1 Missing Pattern

Because the mechanism which generates the pattern of missing data can affect imputation quality, we run experiments under two different missing data mechanisms: missing completely at random (MCAR) and not missing at random (NMAR). These statistical assumptions are summarized in Table 4. The MCAR assumption implies that the missing pattern is completely independent from both the missing and observed values. The NMAR assumption implies that the missing pattern depends upon the missing values. There is an intermediate type of assumption, missing at random (MAR), which implies that the missing pattern depends only upon the observed values, but not upon the missing values. Because this assumption is less general than NMAR, we do not consider this mechanism for our experiments.

To generate MCAR patterns of missing data, we randomly sample a subset of the entries in $\mathbf{X}$ to be missing, assuming that each entry is equally likely to be chosen. The NMAR patterns are generated by sampling missingness indicators as independent Bernoulli random variables where each probability p_{id} equals the probability that a normal random variable $N(x_{id}, \epsilon)$ is greater than a particular threshold for dimension d. The threshold for each dimension d is the quantile of $\mathbf{X}^d$ which corresponds to the desired missing percentage level.

Mechanism of Missing Data	Assumption
Missing Completely at Random (MCAR)	$f(\mathcal{M}\|\mathbf{X}^{obs},\mathbf{X}^{miss}) = f(\mathcal{M})$
Missing at Random (MAR)	$f(\mathcal{M}\|\mathbf{X}^{obs},\mathbf{X}^{miss}) = f(\mathcal{M}\|\mathbf{X}^{obs})$
Not Missing at Random (NMAR)	$f(\mathcal{M}\|\mathbf{X}^{obs},\mathbf{X}^{miss})$ is a function of $\mathbf{X}^{miss}$

Table 4: Statistical assumptions of mechanisms used to generate patterns of missing data $\mathcal{M}$ for data set $\mathbf{X}$. Here, we suppose that f is the underlying density of the missing pattern, and $\mathbf{X}^{obs}, \mathbf{X}^{miss}$ are the observed and missing components of the data set, respectively.

Note that regardless of the missing data scenarios generated for the experiments, in order to make fair comparisons, we always use MCAR as the generating mechanism for cross-validation.

3.1.2 Downstream Tasks

For 10 data sets from the UCI Machine Learning Repository, we run further experiments to evaluate the impact of these imputations on the intended downstream machine learning tasks. This selection includes a representative sample of 5 data sets for regression and 5 data sets for classification, with dependent variable observations $\mathbf{Y} \in \mathbb{R}^n$ and $\mathbf{Y} \in \{0,1\}^n$ respectively. We evaluate both single and multiple imputation methods in these experiments.

For single imputation, we consider `opt.cv` and `benchmark.cv`. First, we divide each downstream data set using a 50% training/testing split. Next, we randomly sample a fixed percentage of the entries in $\mathbf{X}$ to be missing completely at random, ranging from 10% to 50%. For each missing percentage, we impute the missing values in the training set and then fit standard machine learning algorithms to obtain a classification or regression model. We impute the missing values in the testing set by running the imputation methods on the full data set. For the regression tasks, we fit cross-validated LASSO and SVR models and compute the out-of-sample accuracy on the imputed testing set. For the classification tasks, we fit cross-validated SVM and Optimal Trees models and compute the out-of-sample R^2 on the imputed testing set.

We also evaluate the performance of multiple imputation methods on the downstream tasks. In these experiments, we consider the following methods:

1. **Multivariate Imputation by Chained Equations** (`mice`): An iterative method which imputes each dimension with missing values one at a time drawing from distributions fully conditional on the other variables. We use predictive mean matching for continuous variables and logistic regression for categorical variables. This process is repeated to generate m fully imputed data sets. Implemented via the `MICE` package in R.

2. **Optimal Impute for Multiple Imputation** (`opt.mi`): Starting from m warm starts, we run `opt.knn`, `opt.svm`, or `opt.tree` to generate a new set of m fully imputed data sets. We use warm starts produced by `mice`, and the best model among K-NN, SVM, and trees is selected initially via cross-validation.

For both `mice` and `opt.mi`, we generate $m = 5$ multiple imputations for the training set and fit an ensemble of predictive models on these completed training sets. We make predictions on the test set by averaging the predictions from the model ensemble. For the classification tasks, we use a threshold value of 0.5. We run this experiment 100 times with different training/testing splits and distributions of missing values for each data set and report the averaged out-of-sample of the predictive models.

3.2 Results

We run the methods on 84 data sets from the UCI Machine Learning Repository. These data sets range in size from $n = 23$ to 5,875 observations and dimension $p = 2$ to 124. In the following sections, we first show the convergence for each of the `opt.impute` methods is fast and generally leads to a decrease in MAE. Next, we demonstrate that the quality of the imputations is significantly higher for `opt.impute` compared to the reference methods, and that this leads to improved performance on downstream classification and regression tasks. We further discuss the sensitivity of imputation quality to the model parameters (K, cp, C), warm starts, descent method (BCD or CD), and data characteristics including the missing pattern. Finally, we compare the computational burden of each method.

3.2.1 Convergence

Figure 1 represents the change in objective value and MAE over the iterations for each of the `opt.impute` methods based on mean warm start, using `iris` data set as an example. We present results for `opt.knn` (CD and BCD), `opt.svm` (CD), and `opt.tree` (CD). The convergence is relatively fast for all methods; in particular, the BCD algorithm for K-NN converges significantly faster than the CD algorithm. When comparing the change in MAE, the value generally monotonically decreases with each iteration in concordance with the change in objective, especially during the first few iterations. In some paths, MAE increases slightly after a certain point. RMSE exhibits the same behavior and is therefore not plotted. This suggests a potential issue of overfitting to the known observations, which may be remedied by regularization or early stopping. In summary, the solution paths illustrate: 1) convergence is often fast, and 2) the objective functions are decent proxies for out-of-sample MAEs, and 3) imputation quality for each first-order method generally improves until convergence.

In general, we found that the BCD algorithm for `opt.knn` did not significantly improve upon imputation accuracy compared to the CD algorithm, but only improved upon speed. Because the BCD algorithms do not scale as well, we restricted our analysis to the CD algorithms for `opt.svm` and `opt.tree`.

3.2.2 Imputation Accuracy

The imputation accuracy for each data set is presented in Table 10 for the scenario in which 30% of the entries are missing, assuming MCAR. We compare the benchmark ones and each individual `opt.impute` method (not cross-validated); the method with the lowest

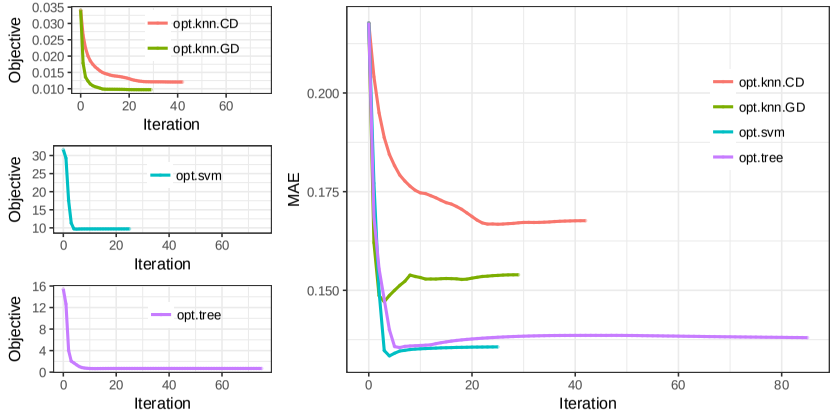


Figure 1: Solution paths of `opt.impute` methods on the `iris` data set. These plots show the objective value and mean absolute error (MAE) of the imputation over the course of the algorithm. Each path represents a different algorithm: `opt.knn` (BCD and CD), `opt.svm` (CD), and `opt.tree` (CD). Mean imputation warm start is used.

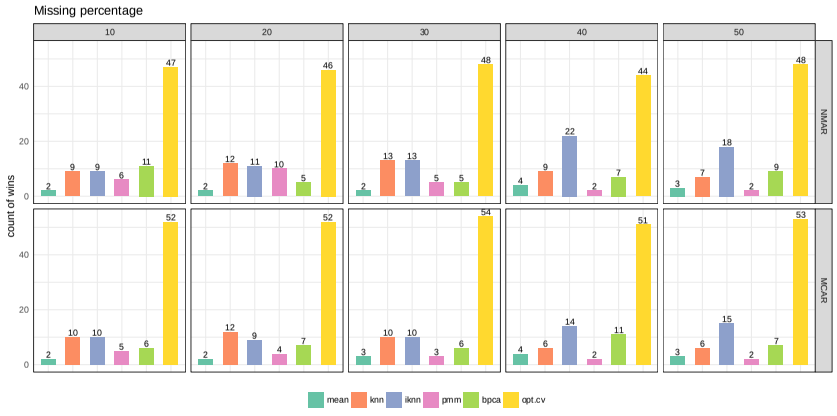
MAE (i.e., best imputation accuracy) is bolded. Among all data sets, at least one of the `opt.impute` methods obtains the lowest MAE in 76.2% of the data sets, followed by `iknn` and `bpca` imputation methods with 9 and 4 wins each. Comparatively, `mean`, `knn`, and `pmm` impute have the weakest performances. Among the `opt.impute` methods, the tree based model achieves the lowest MAE in most data sets.

We repeat this experiment for other percentages of missing data with the winning counts summarized in Figure 2, using `opt.cv` as our proposed method. We show the number of times that each method achieves the best overall imputation with lowest MAE and RMSE under five different missing data percentages, as well MCAR and NMAR scenarios. In all missing data scenarios, our proposed method produces the best imputations in more than half of the data sets according to both performance metrics. Among the comparator methods, `mean` and `pmm` are generally among the weaker ones. When MAE is the metric, the heuristic method `iknn` performs the best among the benchmark methods, suggesting that the idea of iteratively updating the imputed values have merits. At higher percentages of missing values (the right-most subfigures), `bpca` improves in its performance when RMSE is the metric of evaluation, but still not as strong as `opt.cv`.

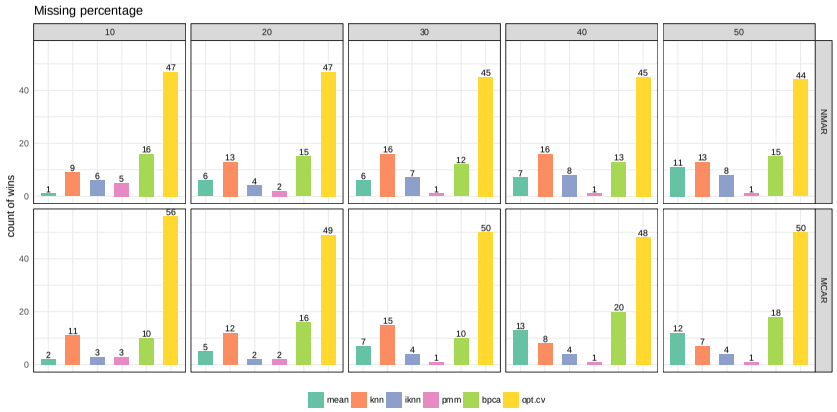
In Figure 3, we present summary results of the MAE and RMSE values as geometric means across all data sets for each missing percentage and missing data mechanism, with the confidence bands representing one geometric standard deviation multiplied above and divided below by the mean. Comparatively, `opt.cv` achieves the lowest average MAE and RMSE values for all missing percentages. At the 10% missing data percentage, the average MAE of the `opt.cv` imputations is 0.100, a reduction of 14.9% from the average MAE of 0.118 obtained by the best benchmark method `knn`. As missing percentages increase, `opt.cv` remains the most accurate imputation method, with the average MAE of 0.142 at 50% missing, a reduction of 12.1% from the average MAE of 0.172 obtained by the next best method `knn`. The performance of `opt.cv` relative to benchmark ones does not appear to differ drastically between the MCAR and NMAR scenarios, with overall higher MAE for NMAR across most methods, as expected.

To isolate the effect of each individual method from the cross-validation procedure, we further summarize the results by comparing one method at a time against the benchmark ones. Table 5 presents the statistical comparisons between each `opt.impute` method and each benchmark method. We conduct pairwise Wilcoxon signed rank tests and paired t-tests between each pair of methods. When comparing `opt.cv` against the benchmark methods, our proposed cross-validated method achieves statistically significant lower rank and lower MAE compared to each benchmark. For each individual `opt.impute` method, with the exception of `opt.svm` against heuristic `iknn`, the `opt.impute` one has statistically significant lower rank than every benchmark. The decrease in MAE is still statistically significant when `mean`, `bpca`, and `pmm` are comparators, but no longer statistically significant when compared to `knn` or `iknn`. This suggests that each of the proposed methods holds its own against most benchmark ones, especially under rank comparisons, but the cross-validation procedure adds another layer of improvement in imputation quality.

Finally, we compare against the same cross-validated procedure introduced in Section 2.6 applied on all the benchmark methods (`benchmark.cv`) with results in Figure 2b. At 30% missing data, we observe 10.1% average improvement in MAE down to 0.118 from 0.131.



(a) Counts based on lowest MAE



(b) Counts based on lowest RMSE

Figure 2: Number of data sets in which each missing data imputation method achieves lowest mean absolute error (MAE) or root mean squared error (RMSE) from true value, with ties included. Each panel represents a different missing percentage ranging from 10% to 50%. Panels in the top row are for not missing at random scenarios, whereas the ones in the bottom row are for missing completely at random scenarios.

Table 5: Pairwise Wilcoxon signed-rank tests and t-tests between `opt.impute` and benchmark methods, with the p-values adjusted for multiple comparisons.

`opt.impute`	**Benchmark**	Δ rank (adjusted p-value)	Δ MAE (adjusted p-value)
opt.cv	mean	-0.7855 (<0.001***)	-0.0502 (<0.001***)
opt.cv	pmm	-0.8355 (<0.001***)	-0.0399 (<0.001***)
opt.cv	bpca	-0.6329 (<0.001***)	-0.0214 (0.0019**)
opt.cv	knn	-0.6281 (<0.001***)	-0.0134 (0.0499*)
opt.cv	iknn	-0.5352 (<0.001***)	-0.0199 (0.0046**)
opt.knn	mean	-0.6424 (<0.001***)	-0.0419 (<0.001***)
opt.knn	pmm	-0.6091 (<0.001***)	-0.0316 (<0.001***)
opt.knn	bpca	-0.4875 (<0.001***)	-0.0131 (0.0601)
opt.knn	knn	-0.3850 (<0.001***)	-0.0051 (0.4574)
opt.knn	iknn	-0.3611 (<0.001***)	-0.0116 (0.1011)
opt.svm	mean	-0.5852 (<0.001***)	-0.0355 (<0.001***)
opt.svm	pmm	-0.4875 (<0.001***)	-0.0252 (<0.001***)
opt.svm	bpca	-0.2515 (<0.001***)	-0.0067 (0.3335)
opt.svm	knn	-0.1371 (0.0033**)	+0.0013 (0.8485)
opt.svm	iknn	-0.0322 (0.0884)	-0.0052 (0.4589)
opt.tree	mean	-0.7139 (<0.001***)	-0.0454 (<0.001***)
opt.tree	pmm	-0.7712 (<0.001***)	-0.0351 (<0.001***)
opt.tree	bpca	-0.5137 (<0.001***)	-0.0165 (0.0176*)
opt.tree	knn	-0.4136 (<0.001***)	-0.0086 (0.2152)
opt.tree	iknn	-0.3135 (<0.001***)	-0.0151 (0.0337*)

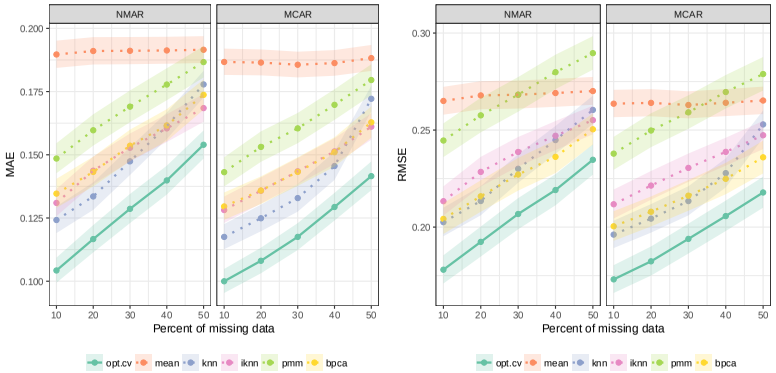
Further, `opt.cv` achieves highest imputation accuracy in more than 78.6% of the data sets compared to `benchmark.cv`.

3.2.3 Performance on Downstream Tasks

Next, we evaluate the performance of standard machine learning algorithms for classification and regression trained on the imputed data. We consider the data sets in Table 6, which were selected as a representative subsample from the UCI Machine Learning Repository data sets. These data sets range in size, having $n = 150$ to 5,875 observations and $p = 4$ to 16 features. The difficulty of the regression or classification task on the completely known data set also varies widely. The baseline out-of-sample accuracy of an SVM model for the binary classification problems ranges from 77% to 100%, and the baseline out-of-sample R^2 of a LASSO model for the regression problems ranges from 0.09 to 0.82. For each of these data sets, the downstream tasks become more difficult as the missing data percentage increases.

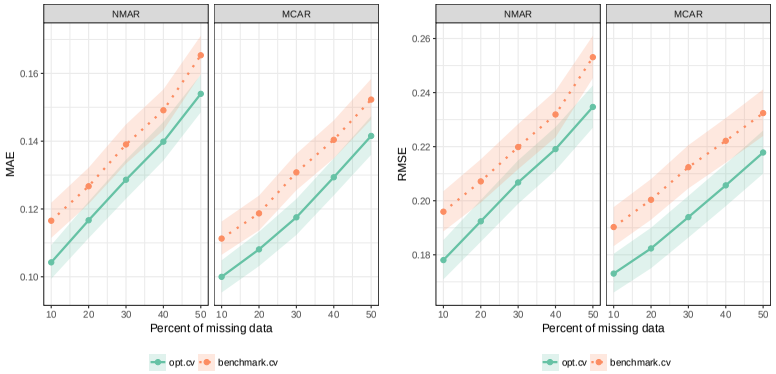
In Figure 4, we show how the imputation method chosen impacts the performance for downstream tasks, across different data sets and different missing data percentages. In Tables 7 and 8, we show pairwise t-test results, aggregating out-of-sample performance results by downstream task and missing percentage. These results include comparisons for both single and multiple imputation methods.

For the single imputation methods, we observe that the improvement of `opt.cv` over the best cross-validated benchmark method is statistically significant for all missing percentages in both classification and regression tasks. Moreover, this improvement in out-of-sample



(a) Average MAE

(b) Average RMSE



(c) Average RMSE

(d) Average RMSE

Figure 3: Mean absolute error (MAE) and root mean squared error (RMSE) across 84 data sets for each imputation method, comparing `opt.cv` against all benchmark methods and against the cross-validated best benchmark method, `benchmark.cv`. The center lines are geometric mean with one geometric standard deviation multiplied above and divided below. The x-axis corresponds to the percentage of missing entries.

Downstream Task	Name	(n,p)	Baseline Accuracy or R^2
Classification	climate-model-crashes	(540, 18)	0.95
	connectionist-bench	(990, 10)	0.93
	ecoli	(336, 8)	0.96
	iris	(150, 4)	1.00
	pima-indians-diabetes	(768, 8)	0.77
Regression	abalone	(4177, 7)	0.51
	auto-mpg	(392, 8)	0.82
	housing	(506, 13)	0.71
	parkinsons-telemonitoring-total	(5875, 16)	0.09
	wine-quality-white	(4898, 11)	0.27

Table 6: Data sets considered for downstream regression and classification tasks. For classification tasks, we list the average baseline out-of-sample accuracy of an SVM model fit on the full data set, and for regression tasks, we list the average baseline out-of-sample R^2 of a LASSO model fit on the full data set.

accuracy and R^2 is monotonically increasing with the missing percentage. At 50% missing data, the average improvement in out-of-sample accuracy is 1.7% for classification tasks, and the average improvement in out-of-sample R^2 is 0.024 for regression tasks.

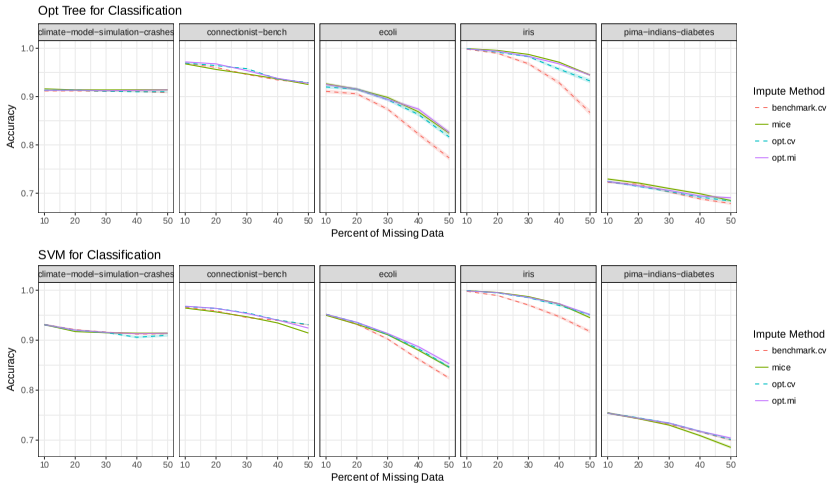
For the multiple imputation methods, we observe that the improvement of `opt.mi` over `mice` is statistically significant for all missing percentages in the regression tasks, and 3/5 missing percentages in the classification tasks. At the 50% missing percentage, the average improvement is 0.5% in out-of-sample accuracy for classification tasks and 0.010 in out-of-sample R^2 for regression tasks. While these improvements are smaller than those for single imputation, they are significant at the $p = 0.001$ level.

Overall, these results suggest that `opt.impute` leads to gains in out-of-sample performance in both single and multiple imputation settings. The relative improvements are consistently greatest at the highest missing percentages, where the imputation method selected has the largest impact on the downstream performance.

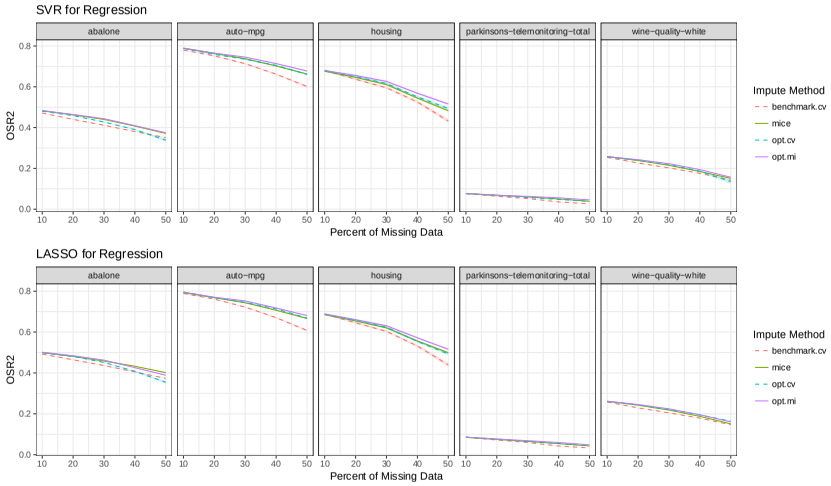
Finally, we compare the performance of single vs multiple imputation for `opt.impute`. We observe that the improvement of `opt.mi` over `opt.cv` is statistically significant in 8/10 scenarios, with the largest improvements occurring at the highest missing percentages. At the 50% missing percentage, the average improvement is 0.4% in out-of-sample accuracy for classification tasks and 0.017 in out-of-sample R^2 for regression tasks. These improvements are similar to the gains in performance over `mice`.

3.2.4 Sensitivity to Parameters

Model performance can be impacted by various parameters. For a specific data set and model, the performance can be sensitive to hyperparameters such as the number of neighbors K in K-NN and the trade-off parameter C for SVM. It is also affected by the number of random starts and choice of algorithm between block coordinate descent and coordinate descent. Data characteristics such as sample size n, feature dimension p, and missing data



(a) Average out-of-sample accuracy values with standard errors of Optimal Trees and SVM models



(b) Average out-of-sample R^2 values with standard errors of SVR and LASSO models

Figure 4: Average out-of-sample performance of downstream models trained on data imputed via `opt.impute` and benchmark methods across a sample of classification and regression problems and a range of missing data percentages. Multiple and single imputation methods are solid and dotted lines respectively.

	Δ Out-of-Sample Accuracy (adjusted p-value)		
Missing %	`opt.mi - mice`	`opt.cv - benchmark.cv`	`opt.mi - opt.cv`
10	-0.0001 (1.0000)	0.0016 (0.0059**)	0.0006 (0.2076)
20	0.0018 (0.0059**)	0.0026 (<0.001***)	0.0008 (0.2076)
30	0.0005 (0.9858)	0.0082 (<0.001***)	0.0002 (1.0000)
40	0.0018 (0.0491*)	0.0113 (<0.001***)	0.0043 (<0.001***)
50	0.0052 (<0.001***)	0.0171 (<0.001***)	0.0038 (<0.001***)

Table 7: Pairwise t-tests between `opt.impute` and benchmark methods for downstream classification tasks, with the p-values adjusted for multiple comparisons.

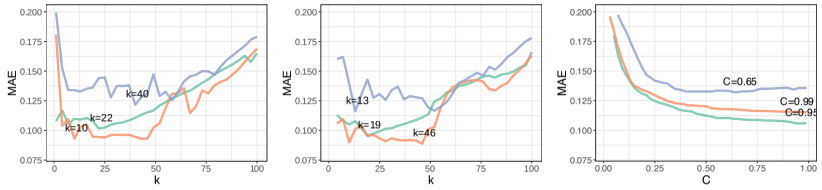
	Δ Out-of-Sample R^2 (adjusted p-value)		
Missing %	`opt.mi - mice`	`opt.cv - benchmark.cv`	`opt.mi - opt.cv`
10	0.0014 (<0.001***)	0.0034 (<0.001***)	0.0013 (<0.001***)
20	0.0029 (<0.001***)	0.0113 (<0.001***)	0.0027 (<0.001***)
30	0.0071 (<0.001***)	0.0161 (<0.001***)	0.0077 (<0.001***)
40	0.0085 (<0.001***)	0.0195 (<0.001***)	0.0108 (<0.001***)
50	0.0097 (<0.001***)	0.0237 (<0.001***)	0.0174 (<0.001***)

Table 8: Pairwise t-tests between `opt.impute` and benchmark methods for downstream regression tasks, with the p-values adjusted for multiple comparisons.

percentage may affect the imputation quality as well. This section explores how these parameters impact the imputation quality.

We found that all of the imputation model hyperparameters that we investigated affect imputation accuracy. Figure 5 shows the relationship between the hyperparameters and MAE for various data sets and missing patterns. For `opt.knn` (CD and BCD), the out-of-sample MAE first decreases and then increases as the hyperparameter increases. When K reaches the sample size, the imputation is equivalent to mean imputation. For `opt.svm`, the imputation accuracy remains relatively constant with respect to changes in parameter C after a certain threshold. There were no external parameters for trees, as the trees in each step were pruned during the training process. Overall, these plots suggest that the `opt.impute` methods are relatively robust even if their hyperparameters are not known exactly.

For `opt.knn`, the performances of block coordinate descent and coordinate descent are comparable. Under most missing data scenarios, block coordinate descent achieves the lower MAE in a few more data sets. As the missing data percentage increases, in many problems both block coordinate descent and coordinate descent methods find the same solutions, thus resulting in a tie. Comparing between the two, there is no clear dominant strategy; in practice we recommend running both methods and then selecting the imputation which yields the lowest objective value.



(a) K in `opt.knn.CD` (b) K in `opt.knn.BCD` (c) C for `opt.svm`

Figure 5: Sensitivity of MAE to the choice of K for the number of neighbors for K-NN coordinate descent, K-NN block coordinate descent, and the trade-off parameter C for SVM in data set `iris`. The colors represent different missing data percentages. The parameter value that achieves lowest MAE is labeled for each missing data percentage.

3.2.5 Computational Speed

Next, we compare the computational time required for all imputation methods across a selection of six UCI data sets and missing data patterns. Each method was run on a single thread of a machine with an Intel Xeon CPU E5-2650 (2.00 GHz) Processor and limited to 8 GB RAM with a time limit of 4 hours. For various `opt.impute` methods, we report the running times for `mean` warm starts, as multiple warm starts can be trivially parallelized. The results are shown below in Table 9.

Mean imputation is almost instantaneous and is therefore not presented in the table. For small-scale problems on the `iris` data set, all imputation methods finish quickly. As the data dimension p increases (for example, in the `libras-movement` data set), most `opt.impute` methods scale better than the `pmm` method. As the sample size n increases, `opt.knn.CD` also scales better than `pmm`, as seen in `banknote-authentication` and `skin-segmentation`. Among the `opt.impute` methods, tree based imputation scales very well with respect to sample size n but not dimension p. Despite its high imputation quality, SVM based imputation scales relatively poorly with respect to both n and p. Among the proposed methods, `opt.knn.CD` has the best scalability in both n and p.

In particular, when comparing coordinate descent and block coordinate descent methods, the former performs best when the data size is large. When n is in the 100,000s, the coordinate descent method still converges within one hour (see `skin-segmentation`). For the block coordinate descent method, each iteration requires solving a separate system of linear equations for each continuous dimension, or an integer optimization problem for each of the categorical dimensions. On the other hand, the main bottleneck of `opt.knn.CD` is computing the K-NN assignment on $\mathbf{X}$ to update $\mathbf{Z}$ each iteration, which requires only $O(n \log n)$ time. When the problem size is small, the running times of the two methods are comparable, and the block coordinate descent method is slightly faster because it converges in fewer iterations. However, when the number of data entries to be imputed exceeds a certain threshold, the block coordinate descent method slows down and takes much longer.

			Time (in seconds)						
			Benchmark			`opt.impute`			
Name	(n, p)	Missing %	`bpca`	`knn`	`pmm`	`knn.CD`	`knn.BCD`	`svm.CD`	`tree.CD`
iris	(150, 4)	10	0.802	0.088	0.353	0.006	0.023	0.131	0.049
		30	1.717	0.446	0.474	0.036	0.041	0.498	0.091
		50	1.875	0.736	0.334	0.085	0.097	0.762	0.062
banknote-authen.	(1372, 4)	10	2.262	2.552	1.717	0.261	1.285	3.269	0.046
		30	14.058	14.914	1.911	0.772	4.981	15.625	0.116
		50	17.820	16.889	2.141	1.578	17.573	15.280	0.159
libras-movement	(360, 90)	10	2.624	0.088	0.353	0.006	0.023	0.131	0.049
		30	3.423	0.446	0.474	0.036	0.041	0.498	0.091
		50	1.892	0.736	0.334	0.085	0.097	0.762	0.062
mushroom	(5644, 76)	10	26.432	387.386	4782.855	8.037	72.169	1442.942	-
		30	46.726	8.134	1068.476	12.818	17.572	-	-
		50	63.556	10.155	893.243	10.511	12.948	-	-
skin-segmentation	(245057, 3)	10	392.310	1144.120	12193.105	1144.144	144.679	-	9.574
		30	450.584	1380.138	-	1420.641	-	-	15.616
		50	615.037	2503.464	-	2582.102	-	-	17.818
cnae-9	(1080, 856)	10	30.310	13.038	-	12.701	12.727	-	-
		30	58.205	13.970	-	13.931	13.972	-	-
		50	126.059	14.361	-	14.284	14.343	-	-

Table 9: Computational time comparison of benchmark and `opt.impute` imputation methods. Blank entries indicate that the method failed to converge with the 4 hour time limit.

In practice, we recommend running both when $n \leq 10{,}000$ and performing model selection between the two, and running only coordinate descent when n is larger.

4. Discussion

One of the primary contributions of this paper is the formulation of the missing data problem as a family of optimization problems. This framework accommodates almost any predictive model that describes the conditional relationship within the data, ranging from parametric to fully non-parametric models. By design, these formulations admit arbitrary missing pattern and mixed data types and do not require specific joint distributional assumptions on the data. In addition, we show how these methods can be used to generate multiple imputations.

The first-order methods that we developed to solve these optimization problem are highly scalable and produce high quality solutions. These methods are computationally fast; for example, the coordinate descent method for SVM solves problems with 100,000s of data points and 1,000s of features in seconds on a standard desktop computer. With more random starts, we obtain solutions which continue to improve upon the objective. Since random warm starts can be trivially parallelized, increasing the number of warm starts does not change the computational times materially if implemented efficiently.

For single imputation, we propose `opt.cv`, a combination method which uses cross-validation to select the best imputation objective function from K-NN, SVM, and decision tree models. We provide evidence on `opt.cv`'s strong empirical performance against benchmark single imputation methods in large scale computational experiments on 84 real-world data sets. For all of the missing data scenarios considered, `opt.cv` produces the best overall imputation for the largest number of data sets. In addition, `opt.cv` produces the lowest average MAE and RMSE for the majority of missing data scenarios. Our proposed cross-validation procedure generates additional missing pattern under MCAR, which may be further improved by adapting the generative procedure for more accurate reflection of imputation quality in the original data missing.

Further, we demonstrate that using the imputations produced by `opt.cv` with values closer to the ground truth leads to gains in out-of-sample performance on downstream regression and classification tasks. This suggests that at medium-to-high missing percentage scenarios, machine learning practitioners will benefit significantly by adopting this framework for single imputation.

For multiple imputation, we propose `opt.mi`, a method which runs `opt.impute` on a set of probabilistically generated warm starts. We show that this method offers a statistically significant improvement over both `mice` and `opt.cv` in the downstream tasks. However, the multiple imputation methods have drawbacks because they are computationally slower, require pooling after analyzing multiple data sets, and produce an ensemble of models which is less interpretable than a single model. Therefore, unless statistical inference is required, `opt.cv` may be preferable for many applications.

Given the optimization formulations introduced in this paper, there are multiple open questions for future research. We may consider alternate cost functions for missing data imputation that reflect out-of-sample performance better. For example, in the K-NN based model, we could add a regularizer term or use the L_1 distance or Mahalanobis distance

metric instead of the squared Euclidean distance metric. The tree based imputation method invites future development in fast optimal trees for convergence and better performance. Finally, solving the global optimization problem (1) fast and accurately for any of the three examples of non-convex, non-linear cost functions $c(\mathbf{U}, \mathbf{W}, \mathbf{V}; \mathbf{X})$ proposed in this paper remains an open question.

5. Conclusions

In summary, we frame the classical missing data problem as a non-convex optimization problem based upon a variety of predictive models. We propose a family of new imputation methods, `opt.impute`, which finds high quality solutions to this problem using fast first-order methods. Through extensive computational experiments on 84 data sets from the UCI Machine Learning Repository, we show that `opt.impute` yields statistically significant gains in imputation quality over state-of-the-art imputation methods, which leads to improved out-of-sample performance on downstream tasks. This approach scales to large problem sizes, generalizes to multiple imputation, and improves over state-of-the-art methods across a broad range of missing data scenarios.

Acknowledgments

The authors thank the reviewers who provided many insightful comments which improved the final manuscript. The research of the second author was supported by a National Science Foundation Predoctoral Fellowship.

				Benchmark					opt.impute		
Name	n	p_0	p_1	mean	pmm	bpca	knn	iknn	knn	svm	tree
acute-inflammations-1	120	1	5	0.3701	0.3626	0.2307	0.2694	0.3598	0.2285	0.2267	**0.2185**
acute-inflammations-2	120	1	5	0.3701	0.3626	0.2307	0.2694	0.3598	0.2285	0.2267	**0.2185**
airfoil-self-noise	1503	5	0	0.2332	0.2270	0.2332	0.2018	0.2054	**0.1944**	0.1949	0.2002
airline-costs	31	9	0	0.1799	0.1566	0.1054	0.1113	0.1071	**0.0970**	0.1084	0.1037
auto-mpg	392	5	2	0.2404	0.1793	0.1547	0.1623	0.1690	0.1396	0.1362	**0.1291**
balance-scale	625	4	0	**0.3011**	0.4112	**0.3011**	0.3503	0.3113	0.3701	0.3206	0.3049
banknote-authentication	1372	4	0	0.1608	0.1596	0.1608	0.1321	0.1361	**0.1117**	0.1182	0.1243
beer-aroma	23	8	0	0.2036	0.2004	0.1772	0.1773	0.1838	0.1728	0.1638	**0.1628**
blood-transfusion	748	4	0	0.1123	0.1215	0.1123	0.0945	0.0880	0.0799	0.0824	**0.0664**
breast-cancer-diagnostic	569	30	0	0.1066	0.0431	0.0558	0.0520	0.0565	0.0486	0.0512	**0.0351**
breast-cancer-prognostic	194	31	1	0.1304	0.0727	0.0850	0.0846	0.0911	0.0794	0.0682	**0.0576**
breast-cancer	683	8	1	0.2458	0.1531	**0.1318**	0.1541	0.1788	0.1367	0.1355	0.1333
climate-model-crashes	540	18	0	**0.2505**	0.3404	**0.2505**	0.2651	0.2570	0.2750	0.2921	0.2519
communities-and-crime-2	111	101	23	0.1374	0.2191	0.1137	0.0864	0.1053	0.0845	0.0875	**0.0577**
communities-and-crime	123	99	23	0.1613	0.2901	0.1327	0.0987	0.1252	0.0973	0.0936	**0.0711**
computer-hardware	209	7	1	0.1989	0.1888	0.1989	0.1824	**0.1703**	0.1917	0.1780	0.1832
concrete-compressive	103	7	0	0.2338	0.2005	0.2057	0.2053	0.1982	0.1854	0.1868	**0.1750**
concrete-flow	103	7	0	0.2338	0.2005	0.2057	0.2053	0.1982	0.1854	0.1868	**0.1750**
concrete-slump	103	7	0	0.2338	0.2005	0.2057	0.2053	0.1982	0.1854	0.1868	**0.1750**
congressional-voting-records	232	0	16	0.4357	0.4351	0.2150	0.2504	0.4357	**0.2107**	0.2449	0.3509
connectionist-bench-sonar	208	60	0	0.1629	0.1208	0.1440	0.1088	0.1219	0.1071	0.0918	**0.0905**
connectionist-bench	990	10	0	0.1506	0.1632	0.1294	0.1049	0.1001	**0.0829**	0.1143	0.1224
construction-maintenance	33	4	0	0.3614	**0.2461**	0.3638	0.3299	0.2836	0.3283	0.3250	0.3979
contraceptive-method-choice	1473	8	1	0.2767	0.2768	0.2519	0.2634	0.2336	**0.2229**	0.2263	0.2452
dermatology	358	33	1	0.2254	0.1447	0.1484	0.1212	0.1421	**0.1082**	0.1364	0.1957
diabetes	43	2	0	0.1868	0.2768	0.1868	**0.1844**	0.2095	0.2404	0.1847	0.1950
ecoli	336	7	0	0.1215	0.1224	0.0938	0.1071	0.0908	0.0990	0.1109	**0.0904**
fertility	100	7	2	0.3526	0.3854	0.3433	0.3432	0.3476	**0.3369**	0.3450	0.3665
flags	194	22	6	0.3246	0.3146	0.3246	0.2542	0.3039	**0.2475**	0.3290	0.2603
geographic-origin	1059	68	0	0.0827	0.0764	0.0599	0.0510	0.0557	0.0477	0.0584	**0.0438**
glass-identification	214	9	0	0.1140	**0.0825**	0.0956	0.0862	0.0865	0.0851	0.0923	0.0862
haberman-survival	306	3	0	0.1701	0.2258	0.1701	0.1754	**0.1663**	0.1734	0.1727	0.1696
hayes-roth	132	4	0	**0.2768**	0.3719	0.2778	0.2873	0.2779	0.2965	0.2948	0.2770
heart-disease-cleveland	297	8	5	0.3261	0.3386	0.2878	0.2945	0.3023	0.2763	**0.2738**	0.3041
hepatitis	80	4	15	0.3094	0.3019	0.3094	0.2753	0.2626	**0.2573**	0.2657	0.3480
hill-valley-noise	606	100	0	0.0998	0.0105	0.0066	0.0052	0.0283	**0.0051**	0.0781	0.0114
hill-valley	606	100	0	0.0971	0.0974	0.0055	0.0042	0.0273	0.0042	0.0783	**0.0031**
housing	506	13	0	0.1821	0.1211	0.1154	0.0985	0.1042	**0.0798**	0.1049	0.1261
hybrid-price	153	3	0	0.1538	0.1605	0.1538	0.1289	**0.1069**	0.1370	0.1202	0.1231
image-segmentation	210	19	0	0.1450	0.0806	0.0856	0.0637	0.0672	**0.0627**	0.0846	0.0628
immigrant-salaries	35	3	0	0.2247	0.2134	0.2247	0.1869	0.1700	0.1901	**0.1673**	0.1808
indian-liver-patient	579	8	2	0.1039	0.0953	0.0954	0.0981	0.0873	0.0910	0.1167	**0.0789**
ionosphere	351	34	0	0.2016	0.1739	0.1552	**0.1107**	0.1187	0.1172	0.1206	0.1475
iris	150	4	0	0.2200	0.1292	0.1571	0.1274	0.1370	0.1132	**0.1048**	0.1130
japan-emmigration	45	5	0	0.2096	0.2625	0.2098	0.2064	**0.1737**	0.2097	0.1866	0.2131
lenses	24	0	4	0.6607	0.6667	0.6696	**0.6339**	0.6607	0.6786	0.6786	0.6667
libras-movement	360	90	0	0.1823	0.0304	0.1022	0.0670	0.1014	0.0688	0.0522	**0.0139**
lpga-2008	157	6	0	0.1459	0.1769	0.1424	0.1448	0.1414	0.1496	**0.1294**	0.1299
lpga-2009	146	11	0	0.1750	0.1048	0.1074	0.1169	0.1131	0.1047	0.0889	**0.0881**
lung-cancer	27	0	56	0.3677	0.3475	0.3644	0.3426	0.3677	0.3586	**0.3348**	0.3438
mammographic-mass	830	0	5	0.2803	0.3307	0.2691	0.2386	0.2762	0.3390	0.2439	**0.2243**
monks-problems-1	124	0	6	0.6441	0.6396	0.6441	0.6059	0.6441	0.6411	**0.5991**	0.6502
monks-problems-2	169	0	6	0.6405	0.6373	0.6454	**0.6340**	0.6405	0.6481	0.6438	0.6383
monks-problems-3	122	0	6	0.6554	**0.5976**	0.6554	0.6813	0.6554	0.6577	0.6877	0.6622
parkinsons-telemonitoring-motor	5875	16	0	0.0623	0.0395	0.0372	0.0389	0.0342	0.0301	0.0458	**0.0265**
parkinsons-telemonitoring-total	5875	16	0	0.0623	0.0395	0.0372	0.0389	0.0342	0.0301	0.0458	**0.0265**
parkinsons	195	21	0	0.1348	0.0888	0.0849	0.0754	0.0814	**0.0690**	0.0824	0.0691
pima-indians-diabetes	768	8	0	0.1217	0.1453	0.1109	0.1164	0.1098	0.1089	**0.1049**	0.1069
planning-relax	182	12	0	0.1441	0.0823	0.1143	0.1188	0.1195	0.1019	0.0809	**0.0680**
post-operative-patient	87	0	8	0.3891	0.4428	0.3891	0.4143	**0.3861**	0.3937	0.4348	0.3955
pyrim	74	27	0	0.1798	0.1235	0.1758	0.1172	0.1193	**0.1145**	0.1219	0.1282
qsar-biodegradation	1055	41	0	0.0749	0.0379	0.0656	0.0385	0.0410	**0.0324**	0.0566	0.0452

Table 10: Mean absolute errors of imputation methods on 84 data sets from the UCI Machine Learning repository with 30% missing values. The lowest MAE for each data set is indicated in bold.

				Benchmark					opt.impute		
Name	n	p_0	p_1	mean	pmm	bpca	knn	iknn	knn	svm	tree
seeds	210	7	0	0.2082	0.0795	0.0651	0.1099	0.0862	0.0715	0.0730	**0.0644**
soybean-large	266	0	35	0.2880	0.2583	0.2467	0.1874	0.2880	**0.1858**	0.1865	0.2103
soybean-small	47	0	35	0.2689	0.2816	0.2673	0.1577	0.2689	**0.1571**	**0.1571**	0.1837
spect-heart	80	0	22	0.2173	0.2134	0.2083	0.1899	0.2173	0.1951	**0.1869**	0.1913
spectf-heart	80	44	0	0.1307	0.1631	0.1307	0.1226	0.1195	0.1141	**0.1058**	0.1138
statlog-project-landsat-satellite	4435	36	0	0.1556	0.0405	0.0472	0.0390	0.0480	0.0329	0.0345	**0.0293**
teaching-assistant-evaluation	151	1	4	0.4017	0.4074	0.4094	0.3711	0.3992	0.4086	0.5131	**0.3370**
thoracic-surgery	470	3	13	0.1469	0.1704	**0.1388**	0.1433	0.1463	0.1415	0.2205	0.1397
thyroid-disease-ann-thyroid	3772	21	0	0.0773	0.0774	0.0869	0.0723	**0.0603**	0.0838	0.1162	0.0729
thyroid-disease-new-thyroid	215	5	0	0.0935	0.1083	0.0887	0.0849	**0.0754**	0.0774	0.0893	0.0851
triazines	186	60	0	0.1574	0.0667	0.1184	0.0503	0.0708	**0.0454**	0.0892	0.0495
tv-sales	31	8	0	0.2073	0.1949	0.1808	0.1934	**0.1729**	0.1952	0.1731	0.1964
vote-for-clinton	2704	9	0	0.0644	0.0715	0.0538	0.0676	0.0552	**0.0523**	0.0633	0.0537
wall-following-robot-2	5456	2	0	0.0721	0.0955	0.0721	0.0754	0.0720	0.0792	0.0847	**0.0717**
wall-following-robot-24	5456	4	0	0.0917	0.1172	0.0917	0.0886	0.0872	**0.0862**	0.0946	0.0895
wiki4he	176	0	44	0.2200	0.2234	0.1857	0.1872	0.1968	0.1777	**0.1731**	0.2085
wine-quality-red	1599	11	0	0.0976	0.0945	0.0761	0.0796	0.0744	**0.0683**	0.0757	0.0742
wine-quality-white	4898	11	0	0.0756	0.0782	0.0668	0.0771	0.0645	0.0598	0.0676	**0.0597**
wine	178	13	0	0.1680	0.1537	0.1203	0.1184	0.1144	**0.1091**	0.1105	0.1296
yacht-hydrodynamics	308	6	0	0.2102	0.1991	0.2088	0.1858	0.1861	0.1866	0.1867	**0.1799**
yeast	1484	8	0	0.0721	0.0917	0.0689	0.0740	**0.0671**	0.0683	0.0928	0.0680
zoo	101	1	15	0.2892	0.2832	0.1835	0.1518	0.2860	0.1502	0.3637	**0.1478**

Table 10: Mean absolute errors of imputation methods on 84 data sets from the UCI Machine Learning repository with 30% missing values. The lowest MAE for each data set is indicated in bold.

References

Dimitri P Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1999.

Dimitris Bertsimas and Jack Dunn. Optimal classification trees. *Machine Learning*, pages 1–44, 2017.

Dimitris Bertsimas and Rahul Mazumder. Least quantile regression via modern optimization. *Annals of Statistics*, 42(6):2494–2525, 2014.

Dimitris Bertsimas and Bart Van Parys. Sparse high-dimensional regression: Exact scalable algorithms and phase transitions. *arXiv preprint arXiv:1709.10029*, 2017.

Dimitris Bertsimas, Jack Dunn, Colin Pawlowski, and Daisy Zhuo. Robust classification. *Submitted for publication*, 2017.

Trond Hellem Bø, Bjarte Dysvik, and Inge Jonassen. LSimpute: accurate estimation of missing values in microarray data with least squares methods. *Nucleic Acids Research*, 32(3):e34–e34, 2004.

Lígia P Brás and José C Menezes. Improving cluster-based missing value estimation of DNA microarray data. *Biomolecular Engineering*, 24(2):273–282, 2007.

Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

Lane F Burgette and Jerome P Reiter. Multiple imputation for missing data via sequential regression trees. *American Journal of Epidemiology*, 172:1070–1076, 2010.

Stef Buuren and Karin Groothuis-Oudshoorn. MICE: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, 45(3), 2011.

Zhipeng Cai, Maysam Heydari, and Guohui Lin. Iterated local least squares microarray missing value imputation. *Journal of Bioinformatics and Computational Biology*, 4(05): 935–957, 2006.

Rich Caruana. A non-parametric EM-style algorithm for imputing missing values. In *AISTATS*, 2001.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3): 273–297, 1995.

Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2(Dec):265–292, 2001.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (methodological)*, pages 1–38, 1977.

Zoubin Ghahramani and Michael I Jordan. Supervised learning from incomplete data via an EM approach. In *Advances in Neural Information Processing Systems*, pages 120–127, 1994.

James Honaker, Gary King, Matthew Blackwell, et al. Amelia II: A program for missing data. *Journal of Statistical Software*, 45(7):1–47, 2011.

Mohammad E Khan, Guillaume Bouchard, Kevin P Murphy, and Benjamin M Marlin. Variational bounds for mixed-data factor analysis. In *Advances in Neural Information Processing Systems*, pages 1108–1116, 2010.

Hyunsoo Kim, Gene H Golub, and Haesun Park. Missing value estimation for DNA microarray gene expression data: local least squares imputation. *Bioinformatics*, 21(2): 187–198, 2005.

Ki-Yeol Kim, Byoung-Jin Kim, and Gwan-Su Yi. Reuse of imputed data in microarray analysis increases imputation efficiency. *BMC Bioinformatics*, 5(1):1, 2004.

Alan Wee-Chung Liew, Ngai-Fong Law, and Hong Yan. Missing value imputation for gene expression data: computational techniques to recover missing data from available information. *Briefings in Bioinformatics*, 12(5):498–513, 2011.

Roderick JA Little and Donald B Rubin. *Statistical Analysis with Missing Data*. John Wiley & Sons, 1987.

Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11(Aug): 2287–2322, 2010.

Shakir Mohamed, Zoubin Ghahramani, and Katherine A Heller. Bayesian exponential family PCA. In *Advances in Neural Information Processing Systems*, pages 1089–1096, 2009.

Shigeyuki Oba, Masa-aki Sato, Ichiro Takemasa, Morito Monden, Ken-ichi Matsubara, and Shin Ishii. A Bayesian missing value estimation method for gene expression profile data. *Bioinformatics*, 19(16):2088–2096, 2003.

Trivellore E Raghunathan, James M Lepkowski, John Van Hoewyk, and Peter Solenberger. A multivariate technique for multiply imputing missing values using a sequence of regression models. *Survey Methodology*, 27(1):85–96, 2001.

Daniel J Stekhoven and Peter Bühlmann. Missforest: non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 2012.

Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525, 2001.

Stef Van Buuren. Multiple imputation of discrete and continuous data by fully conditional specification. *Statistical Methods in Medical Research*, 16(3):219–242, 2007.

Kiri Wagstaff. Clustering with missing values: No imputation required. In *Classification, Clustering, and Data Mining Applications*, pages 649–658. Springer, 2004.

Xian Wang, Ao Li, Zhaohui Jiang, and Huanqing Feng. Missing value estimation for DNA microarray gene expression data by support vector regression imputation and orthogonal coding scheme. *BMC Bioinformatics*, 7(1):1, 2006.

Stephen J Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1): 3–34, 2015.

Xiaobai Zhang, Xiaofeng Song, Huinan Wang, and Huanping Zhang. Sequential local least squares imputation estimating missing value of microarray data. *Computers in Biology and Medicine*, 38(10):1112–1120, 2008.

Journal of Machine Learning Research 18 (2018) 1-32 Submitted 3/17; Revised 11/17; Published 4/18

Saturating Splines and Feature Selection

Nicholas Boyd NICKBOYD@BERKELEY.EDU
Department of Statistics
University of California
Berkeley, CA 94720-1776, USA

Trevor Hastie HASTIE@STANFORD.EDU
Department of Statistics
Stanford University
Stanford, CA 94305, USA

Stephen Boyd BOYD@STANFORD.EDU
Department of Electrical Engineering
Stanford University
Stanford, CA 94305, USA

Benjamin Recht BRECHT@BERKELEY.EDU
Department of Electrical Engineering and Computer Science
University of California
Berkeley, CA 94720-1776, USA

Michael I. Jordan JORDAN@CS.BERKELEY.EDU
Division of Computer Science and Department of Statistics
University of California
Berkeley, CA 94720-1776, USA

Editor: Jie Peng

Abstract

We extend the adaptive regression spline model by incorporating *saturation*, the natural requirement that a function extend as a constant outside a certain range. We fit saturating splines to data via a convex optimization problem over a space of measures, which we solve using an efficient algorithm based on the conditional gradient method. Unlike many existing approaches, our algorithm solves the original infinite-dimensional (for splines of degree at least two) optimization problem without pre-specified knot locations. We then adapt our algorithm to fit generalized additive models with saturating splines as coordinate functions and show that the saturation requirement allows our model to simultaneously perform feature selection and nonlinear function fitting. Finally, we briefly sketch how the method can be extended to higher order splines and to different requirements on the extension outside the data range.

Keywords: Convex optimization, feature selection, splines, lasso, regression

1. Introduction

Splines—piecewise polynomials with continuity constraints—are widely used to fit data (Hastie et al., 2001, §5.1). One issue with piecewise polynomials is that they behave erratically beyond their boundary knot points, and (typically) grow without bound outside of that range (Hastie et al., 2001, §5.2). This instability makes extrapolation dangerous; practitioners must take care to avoid querying spline models near or outside of the range of the training data.

Smoothing spline algorithms (De Boor, 2001; Wahba, 1990; Green and Silverman, 1994) ameliorate this problem by fitting *natural splines*, which reduce to a lower-degree polynomial beyond the boundary knots. The most commonly used varieties of smoothing splines are cubic smoothing splines (degree-three splines that reduce to linear outside the boundary knots) and linear smoothing splines, which extend as constant. The saturating splines we propose are closely related to linear smoothing splines.

Smoothing splines use an ℓ_2 or quadratic notion of complexity, and hence fit models with a predetermined and dense set of knot points (Hastie et al., 2001, §5.4). *Adaptive regression splines* (Mammen and van de Geer, 1997), on the other hand, use an ℓ_1-type penalty, which can result in a sparse set of adaptively chosen knots. However, adaptive regression splines do not reduce to lower degree outside of the range of their largest knots, and hence may suffer from instability.

We propose fitting adaptive regression splines with explicit constraints on the degree of the spline outside of a certain interval. We call such splines *saturating splines*. While the approach we take can be extended to fitting splines of arbitrary degree with constraints on arbitrary derivatives, in this paper we focus on fitting linear splines that are flat (constant) outside the data range; we mention the extension to higher degree splines in §8. We show that saturating splines inherit the knot-selection property of adaptive regression splines, while at the same time behave like natural splines near the boundaries of the data.

We also show a very important benefit of our approach in the context of fitting generalized additive models (Hastie and Tibshirani, 1990) with saturating spline coordinate functions: the saturation constraint naturally results in variable selection. Not only do we control the complexity of each coordinate function through knot selection, but with the saturation condition, no knots on a variable means the variable is out of the model. This is not true for adaptive splines, since the linear term is unpenalized and hence each variable would always be in the model. The lack of feature selection can hurt interpretability and, in certain cases, generalization. The saturation constraint we propose precludes linear functions, and in concert with the adaptive spline ℓ_1 penalty encourages coordinate functions to be identically zero. As a result, generalized additive models fit with saturating spline component functions often depend on only a few input features.

Like smoothing splines and adaptive regression splines, saturating splines arise as solutions to certain natural functional regression problems. We solve the saturating spline fitting problem by reformulating it as a convex optimization problem over a space of measures, roughly speaking, the second derivative of the fitted function. To the best of our knowledge, this approach is novel. We then apply a variant of the classical conditional gradient method (Jaggi, 2013.; Boyd et al., 2017) to this problem. At each iteration of our algorithm, an atomic measure is produced; moreover, we can uniformly bound the number of atoms, which corresponds to the number of knot points in the spline function. (While we manipulate atomic measures, we solve the problem over the space of all measures with finite total variation.) In contrast to standard coordinate descent methods, in each iteration of the conditional gradient method the weights of *two* knot points are adjusted. In the fully corrective step, we solve a finite-dimensional convex optimization problem with ℓ_1

and simple linear constraints. Numerical experiments show that the method is extremely effective in practice.

Our optimization method can exploit warm starts, i.e., it can use an initial guess for the fitted function. This allows us to compute an entire regularization path efficiently, at a cost typically just a small multiple of the effort to solve the problem for one value of the regularization parameter. Because our algorithm is based on the conditional gradient method, we can use the framework of Giesen et al. (2012) to compute a provably ϵ-suboptimal approximate regularization path. When fitting generalized additive models, the regularization path has attractive features: at critical values of the regularization parameter, new regressors are brought into (or, occasionally, out of) the model, or new knot points are added to (or deleted from) one of the existing coordinate functions. Thus our approach combines feature selection and knot point selection.

1.1 Outline

In §2 we introduce a univariate function fitting problem, inspired by the adaptive spline estimation problem of Mammen and van de Geer (1997), that includes the additional requirement that the fitted function saturate. In §3 we make the connection between our function estimation problem and standard adaptive splines, and pose the saturating spline fitting problem as a convex optimization problem over measures. In §4 we modify the classical conditional gradient method to solve this optimization problem. In §5 we extend the optimization problem and algorithm to fit generalized additive models to multivariate data. We illustrate the effectiveness of the method with several examples in §7. We discuss generalizations to higher-degree splines in §8. Finally, we discuss potential extensions and variations in §9. The appendix includes implementation details and proofs.

2. Univariate Function Fitting

We wish to fit a continuous bounded function $f : \mathbf{R} \to \mathbf{R}$ from data $(x_i, y_i) \in \mathbf{R} \times \mathcal{Y}$, $i = 1, \ldots, n$, $x_i \in [0, 1]$. To do this we will choose f to minimize a data mismatch or loss function subject to a constraint that encourages regularity in f, and an additional constraint, saturation, that we describe below.

The loss is given by

$$L(f) = \sum_{i=1}^{n} \ell(f(x_i), y_i),$$

where $\ell : \mathbf{R} \times \mathcal{Y} \to \mathbf{R}$ is nonnegative, twice differentiable, and strictly convex in its first argument. Typical loss functions include $\ell(z, w) = (z-w)^2/2$ (standard regression, $\mathcal{Y} = \mathbf{R}$), or $\ell(z, w) = \log(1 + \exp -(zw))$ (logistic regression, with $\mathcal{Y} = \{-1, 1\}$). The loss L is a convex functional of the function f that only depends on the values of f at the data points x_i. The smaller the loss, the better f fits the given data.

We constrain the function f to be simple by limiting the value of a nonnegative regularization functional R. In this paper, we take R to be the total variation of the derivative of f,

$$R(f) = \mathrm{TV}(f'),$$

a convex functional of f. For a twice-differentiable function f, recall that

$$\mathrm{TV}(f') = \int |f''(x)| \, dx, \tag{1}$$

i.e., the regularization is the ℓ_1 norm of the second derivative. (As we review in the following section, the modern definition of total variation extends this equality to nondifferentiable functions.) The total variation limit we impose on f is $R(f) \leq \tau$, where τ is a parameter that we use to trade off model fit and model regularity. This regularization constraint implicitly constrains f to be differentiable almost everywhere, with its derivative having finite total variation.

Our model f will be subject to one more constraint, that it saturates (outside the interval $[0, 1]$), which means that it is a (possibly different) constant on the two intervals outside $[0, 1]$: $f(x) = f(0)$ for $x \leq 0$, and $f(x) = f(1)$ for $x \geq 1$. In other words, f extends as a constant outside the nominal data range of $[0, 1]$. In terms of the derivative, this is equivalent to the requirement that f' exists and is zero outside $[0, 1]$.

The fitting problem is then

$$\begin{array}{ll} \text{minimize} & L(f) \\ \text{subject to} & R(f) \leq \tau, \\ & f'(x) = 0 \text{ for } x \notin [0, 1], \end{array} \tag{2}$$

where $\tau \geq 0$ is the regularization parameter. The variable to be determined is the function f, which is in the vector space of continuous functions with derivatives of finite total variation. This fitting problem is an infinite-dimensional convex optimization problem.

In applications the problem (2) is solved for a range of values of τ, which yields the regularization path. The final model is selected using a hold-out set or cross-validation. For $\tau = 0$, f must be constant and the problem (2) reduces to fitting the best constant to the data. As τ increases, f is less constrained, and our fitted model becomes more complex; eventually we expect overfitting. For example, in the case of regression, with a loss function that satisfies $\ell(u, u) = 0$ and data with distinct x_i, the fitting function is the piecewise-linear function that interpolates the data, for large enough τ.

3. Splines and Functions of Bounded Variation

In this section we explore the connection between our fitting problem and degree-one splines, i.e., piecewise-linear continuous functions, which have the form

$$f(x) = c + \sum_{i=1}^{K} w_i (x - t_i)_+, \tag{3}$$

where $(z)_+ = \max\{z, 0\}$. We assume that the t_i are distinct, and refer to them as knot points or simply knots. The scalars w_i are the weights, and c is the offset. We refer to the function $x \mapsto (x - t_i)_+$ as a hinge function, so a degree-one spline is a finite linear combination of hinge functions, plus a constant.

3.1 Functions of Bounded Variation

A right-continuous function $h : [0,1] \to \mathbf{R}$ is of bounded variation if and only if there exists a signed measure μ on $[0,1]$ with

$$h(z) = \int 1(y \le z)\, d\mu(y), \tag{4}$$

where $1(y \le z) = 1$ for $y \le z$ and 0 otherwise. The measure μ is unique; we can think of it as the derivative of h. That is, (4) is essentially the second fundamental theorem of calculus with h' replaced by μ.

We also have $\mathrm{TV}(h) = \int d|\mu|$. (This is called the total variation of the measure μ.) We will denote this using the notation $\|\mu\|_1$, to emphasize the similarity with the finite-dimensional case, or the case when h is differentiable: $\mathrm{TV}(h) = \|h'\|_1$. When the measure μ is atomic, the function h is piecewise constant with jumps at the points in the support of μ.

3.2 Splines and Derivatives with Bounded Variation

Now suppose that $f : [0,1] \to \mathbf{R}$ has a right-continuous *derivative* of bounded variation. From (4), with $h = f'$, and the fundamental theorem of calculus, we have

$$f(x) = f(0) + \int_0^x f'(z)\, dz = f(0) + \int_0^x \int 1(y \le z)\, d\mu(y)\, dz \tag{5}$$

$$= f(0) + \int \int_0^x 1(y \le z)\, dz\, d\mu(y) \tag{6}$$

$$= f(0) + \int (x-y)_+\, d\mu(y). \tag{7}$$

This shows that any such function is a (possibly infinite) linear combination of hinge functions, plus a constant (i.e., $f(0)$). In this case, the measure μ can be thought of as the *second* derivative of f.

When μ is atomic and supported on a finite set, that is,

$$\mu = \sum_{i=1}^{K} w_i \delta_{t_i},$$

f is a degree-one spline of the form (3), with $c = f(0)$. So degree-one splines correspond exactly to the case where the measure μ (roughly, the second derivative) has finite support.

We introduce the notation

$$f_\mu(x) = \int_0^x \int 1(t \le z)\, d\mu(t)\, dz = \int (x-t)_+\, d\mu(t) \tag{8}$$

to denote the function derived from the measure μ. It is, roughly speaking, the double integral of the measure μ, or the (potentially infinite) linear combination of hinge functions associated with the measure μ. The mapping from μ to f_μ is linear, and we have $\mathrm{TV}(f'_\mu) = \|\mu\|_1$. A simple example of f_μ, its first derivative f'_μ, and its (atomic measure) second derivative μ is shown in Figure 1.

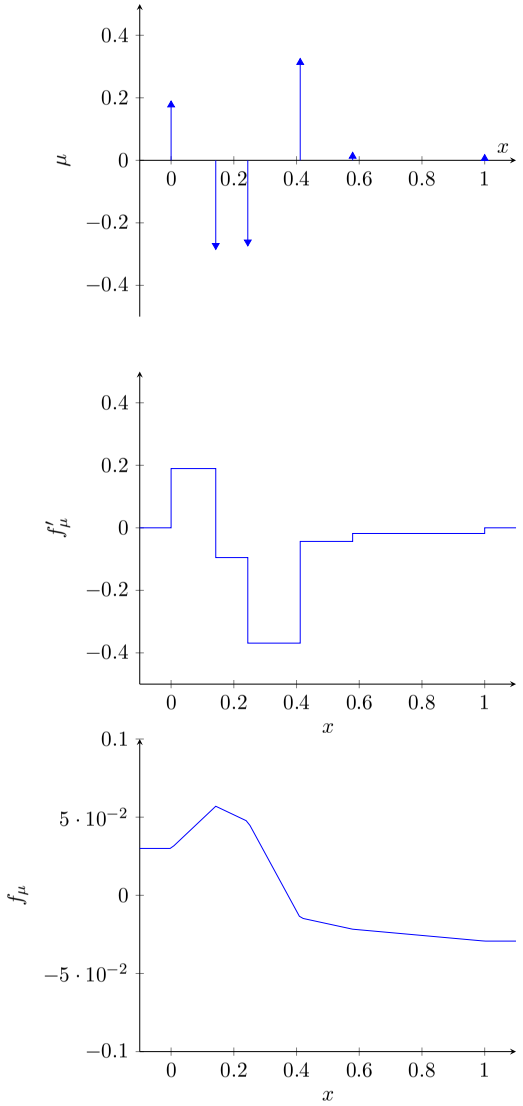


Figure 1: f_μ and f'_μ generated by the atomic measure μ (f''_μ). The regularization functional, $\mathrm{TV}(f'_\mu)$, is the sum of the absolute values of the spikes in μ. Note that the (signed) sum of the spikes in μ is zero: that is, $\int d\mu = 0$, which implies that f_μ saturates.

3.3 Fitting Splines by Optimizing Over Measures

Identifying $f = c + f_\mu$, we can solve the fitting problem (2) by minimizing over the bounded measure μ on $[0,1]$, and the constant c. The measure μ is the second derivative of f, and the constant c corresponds to $f(0)$. The total variation regularization constraint $\mathrm{TV}(f') \leq \tau$ corresponds to $\|\mu\|_1 \leq \tau$. The saturation condition holds by construction for $x < 0$; to ensure that $f'(x) = 0$ for $x > 1$, we need

$$f'(1) = f'(0) + \int_0^1 d\mu = 0.$$

In other words, saturation of f corresponds to μ having total (net) mass zero. Thus (2) can be rephrased as

$$\begin{array}{ll} \text{minimize} & L(E_x\mu + c) \\ \text{subject to} & \|\mu\|_1 \leq \tau, \\ & \int d\mu = 0 \end{array} \tag{9}$$

over the bounded measure μ on $[0,1]$, and $c \in \mathbf{R}$. Note the slight abuse of notation here: we now (and for the remainder of the paper) consider L as a functional on $\mathbf{R}^n$. In the above, E_x is the linear operator that maps μ to the vector $(f_\mu(x_1), \ldots, f_\mu(x_n))$, given by (8). E_x is clearly linear, as it is the integral of the function $\psi : \mathbf{R} \to \mathbf{R}^n$:

$$\psi(t) = ((x_1 - t)_+, \ldots, (x_n - t)_+)$$

against μ. We will apply the conditional gradient method directly to this problem.

To gain intuition about the optimization problem (9), we can consider it as a infinite-dimensional analogue of the standard lasso (Tibshirani, 1994). The lasso is the solution to the optimization problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|Aw - y\|_2^2 \\ \text{subject to} & \|w\|_1 \leq \tau. \end{array} \tag{10}$$

Here w is a vector in $\mathbf{R}^d$, and $A \in \mathbf{R}^{(n,d)}$ is a matrix. Ignoring the constant term c, we see that (9) looks very similar to (10), where E_x plays the role of A; indeed, E_x is essentially a matrix with n rows and infinitely many columns. Our intuition from the lasso suggests that there should be solutions of (9) that are sparse, which here means that μ is atomic. In terms of f_μ, sparsity means there are solutions of the original functional fitting problem (2) that are degree-one splines. This is indeed the case. Theorem 1 shows that there is a solution of (9) with μ atomic, supported on no more than $n+2$ points; in other words, f_μ is a degree-one spline with $K \leq n+2$. Moreover, in practice the solution of (9) will exhibit selection, that is, it will be supported on far fewer than $n+2$ points.

Theorem 1 *Fix $x_1, \ldots, x_n \in [0,1]$ and $f : \mathbf{R} \to \mathbf{R}$ with f' (right-continuous) of bounded total variation, and f constant outside of $[0,1]$. Then there exists a degree-one saturating spline $\hat{f}$ (with an most $n+2$ knots) that matches f on x_i with $\mathrm{TV}(\hat{f}') \leq \mathrm{TV}(f')$.*

For the remainder of the paper we will ignore the constant term c. It is not difficult to adapt the algorithms we present to handle the constant term, but doing so does add some notational complexity. It's also possible to minimize out c, as it does not affect the regularization term; the resulting problem is still convex in w.

4. The Conditional Gradient Method for Fitting Splines

In this section we outline our algorithm for solving (9) (and therefore also (2)). To that end, we briefly review the classical conditional gradient method (Jaggi, 2013.) and the measure-theoretic version proposed in (Boyd et al., 2017).

The optimization problem we need to solve, (9), (without the constant term c) is

$$\begin{array}{ll} \text{minimize} & L(E_x\mu) \\ \text{subject to} & \int d\mu = 0, \\ & \|\mu\|_1 \leq \tau. \end{array} \tag{11}$$

As noted in the last section, (11) is a convex optimization problem over a space of measures. We closely follow the approach taken in Boyd et al. (2017) and apply the conditional gradient method to this problem directly.

The main benefit of this approach is that we can restrict our attention to *atomic* measures, i.e., μ of the form

$$\mu = \sum_{j=1}^K w_j \delta_{t_j}.$$

Measures of this form are easily representable in a computer, by simply storing a list of (w_j, t_j) pairs. Theorem 1 ensures that the number of knots we need to store is absolutely bounded, i.e., that our algorithm runs in bounded memory. While we manipulate atomic measures, we solve the problem (11) over all bounded measures.

One thing to note about finitely-supported atomic measures is that we can easily optimize over the weights w_j with the knot locations t_j fixed, since this corresponds to a finite-dimensional convex optimization problem amenable to any standard algorithm. Our algorithm makes use of this fact, and alternates between adding pairs of knots and optimizing over the weights w at each iteration. In this latter step knots can be (and indeed eventually must be) removed. In an additional and optional step the knot points can be moved continuously within $[0,1]$, or to neighboring data points. This step is not needed for theoretical convergence but can improve convergence and the sparsity of the final solution in practice.

4.1 The Conditional Gradient Method

The conditional gradient method (CGM) solves constrained convex optimization problems of the form

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathcal{C}, \end{array} \tag{12}$$

with variable $x \in \mathbf{R}^d$. In the above, it is always assumed that the (convex) function f is differentiable. At each iteration of the CGM we form the standard linear approximation to the function f at the current iterate x_m:

$$\hat{f}(x; x_m) = f(x_m) + f'(x - x_m; x_m).$$

Here $f'(d; x)$ is the directional derivative of the function f at x in the direction d, defined by

$$f'(d; x) = \lim_{t \searrow 0} \frac{f(x + td) - f(x)}{t}.$$

Our use of the directional derivative here may seem surprising: for differentiable functions on $\mathbf{R}^d$, $f'(d;x)$ is always equal to $\langle \nabla f(x), d\rangle$. The direct applicability of directional derivatives to convex functionals of measures motivates us to prefer the directional derivative.

Convexity of f implies that $\hat{f}$ is a *lower* bound on f, that is:

$$\hat{f}(x; x_m) \leq f(x). \tag{13}$$

In the next step of the CGM, we minimize this first-order approximation over the feasible set $\mathcal{C}$:

$$s_m \in \arg\min_{s\in\mathcal{C}} \hat{f}(s; x_m) = \arg\min_{s\in\mathcal{C}} f'(s; x_m).$$

The point s_m is called the conditional gradient of f. Note that s_m provides a lower bound on $f(x_\star)$:

$$\hat{f}(s_m; x_m) \leq f(x_\star).$$

In particular, we can bound the sub-optimality of the point x_m:

$$f(x_m) - f(x_\star) \leq -f'(s_m - x_m; x_m). \tag{14}$$

One can show (Jaggi, 2013.) that this bound decreases to zero, which means that it can be used as a (non-heuristic) termination criterion. After determining s_m, there are several options for updating x_m. In this paper, we will use the fully-corrective variant of the CGM, which chooses x_{m+1} to minimize f over the convex hull of $\{s_1, s_2, \ldots, s_m\}$. Note that this last step may become computationally intensive as k grows, and indeed limits the applicability of the conditional gradient method to problems where this step is computationally feasible. One option is to remove previous conditional gradients as soon as they are not selected in the minimization step. Caratheodory's theorem ensures us that the set of previous conditional gradients we need to track is then bounded by $d+1$. In practice, however, the algorithm is usually terminated well before $d+1$ iterations.

Algorithm 1 Fully-corrective conditional gradient method

For $m = 1, \ldots$

1. Linearize: $\hat{f}(s; x_m) \leftarrow f(x_m) + f'(s - x_m; x_m)$.
2. Minimize: $s_m \in \arg\min_{s\in\mathcal{C}} \hat{f}(s; x_m)$.
3. Update: $x_m \in \arg\min_{x\in\mathrm{conv}(s_1,\ldots,s_m)} f(s)$.

4.2 Conditional Gradient for Measures

In this subsection, we apply the conditional gradient method to the infinite-dimensional problem (11), which we repeat here:

$$\begin{array}{ll} \text{minimize} & L(E_x\mu) \\ \text{subject to} & \int d\mu = 0, \\ & \|\mu\|_1 \leq \tau. \end{array} \tag{15}$$

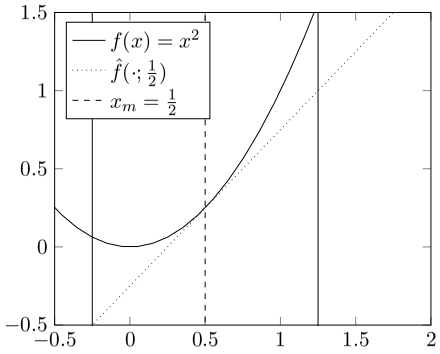


Figure 2: An illustration of a single iteration of the conditional gradient method on the function $f(x) = x^2$ at the point $\frac{1}{2}$. The set $\mathcal{C}$ is the interval $[-0.25, 1.25]$, indicated by the solid vertical lines. The first order approximation $\hat{f}(\cdot; \frac{1}{2})$ is plotted as the dotted line tangential to $f(x)$ at $\frac{1}{2}$. The conditional gradient s_m is the point -0.25.

First we'll show that the conditional gradient, i.e., the measure s_m, can be chosen to be supported on exactly two points, and is computable in time linear in n. The directional derivative of the objective function in the direction of the measure s at the point μ is given by

$$\begin{aligned} &\lim_{t\searrow 0} \frac{L(E_x(\mu + ts)) - L(E_x\mu)}{t} \\ &= \lim_{t\searrow 0} \frac{L(E_x\mu + tE_x s) - L(E_x\mu)}{t} \\ &= L'(E_x s; E_x\mu) \\ &= \langle \nabla L(E_x\mu), E_x s\rangle_{\mathbf{R}^n}. \end{aligned}$$

We can then interchange the inner-product in $\langle \nabla L(E_x\mu), E_x s\rangle$ with the integral in $E_x s = \int \psi(t)\, ds(t)$:

$$\langle \nabla L(E_x\mu), E_x s\rangle = \int \langle \nabla L(E_x\mu), \psi(t)\rangle\, ds(t). \tag{16}$$

Let $g = \nabla L(E_x\mu) \in \mathbf{R}^n$. Note that in the case $\ell(x,y) = \frac{(x-y)^2}{2}$, g is simply the residual $E_x\mu - y$ and $\langle g, \psi(t)\rangle$ is the correlation between the residual and a single hinge function located at t. A conditional gradient is any solution to the following optimization problem

$$\begin{array}{ll} \text{minimize} & \int \langle g, \psi(t)\rangle\, ds(t) \\ \text{subject to} & \int ds = 0, \\ & \|s\|_1 \leq \tau. \end{array} \tag{17}$$

Without the integral constraint, we would expect there to be a solution to (17) that is a single point-mass: the objective function is the integral of a scalar-valued function against a bounded measure. We'll show that there is always a solution to (17) that is supported on exactly two points. Furthermore, we'll show that those two points can be computed in time linear in n.

First we'll construct a particular feasible point for (17) and then we'll show that it achieves the optimal value. Let

$$t_+ \in \arg\min_t \langle g, \psi(t)\rangle, \quad t_- \in \arg\min_t -\langle g, \psi(t)\rangle.$$

Define

$$s_\star = \frac{\tau}{2}\delta_{t_+} - \frac{\tau}{2}\delta_{t_-}.$$

The objective value achieved by $s_\star$ is

$$o_\star = \frac{\tau}{2}\left(\langle g, \psi(t_+)\rangle - \langle g, \psi(t_-)\rangle\right).$$

We'll show that either any measure s that is feasible for (17) has objective value bounded below by $o_\star$ or μ is optimal for (11). Let s be any feasible measure for (17). Decompose s into the difference of two mutually singular non-negative measures: $s = s_+ - s_-$. Then as s is feasible we have $\|s_+\|_1 = \|s_-\|_1 \leq \frac{\tau}{2}$. The objective value achieved by s can be bounded below as follows

$$\begin{aligned}\int \langle g, \psi(t)\rangle ds(t) &= \int \langle g, \psi(t)\rangle ds_+(t) + \int -\langle g, \psi(t)\rangle ds_-(t) \\ &\geq \|s_+\|_1 \left(\min_t \langle g, \psi(t)\rangle\right) + \|s_-\|_1 \left(\min_t -\langle g, \psi(t)\rangle\right) \\ &\geq \|s_+\|_1 \left(\min_t \langle g, \psi(t)\rangle + \min_t -\langle g, \psi(t)\rangle\right).\end{aligned}$$

Suppose $(\min_t \langle g, \psi(t)\rangle + \min_t -\langle g, \psi(t)\rangle) \geq 0$. Then the argument above implies $s_\star = 0$ is a conditional gradient for (11), and thus (14) implies μ is optimal. Otherwise we have

$$\left(\min_t \langle g, \psi(t)\rangle + \min_t -\langle g, \psi(t)\rangle\right) < 0,$$

which implies

$$\|s_+\|_1 \left(\min_t \langle g, \psi(t)\rangle + \min_t -\langle g, \psi(t)\rangle\right) \geq \frac{\tau}{2}\left(\min_t \langle g, \psi(t)\rangle + \min_t -\langle g, \psi(t)\rangle\right) = o_\star.$$

This proves the assertion.

Note that finding t_- and t_+ involves two *separate* optimization problems over $[0, 1]$ instead of one over $[0, 1] \times [0, 1]$. These problems are readily solved by gridding, though in this case they can be solved exactly in time linear in n if we have access to a sorted vector of the data points x_i. To see this, we expand the objective function for t_+ above,

$$t_+ = \arg\min_{0\leq t\leq 1} \sum_{i=1}^n g_i(x_i - t)_+ = \arg\min_t \sum_{i:x_i\geq t} g_i(x_i - t).$$

If x_i are sorted, we can compute the minimizer between each pair of consecutive data points exactly, since this is simply computing the minimizer of a linear functional over an interval. Thus in a single pass over the data we can compute the global minimizer exactly.

Immediately after computing t_- and t_+ we can use (14) to bound the suboptimality of μ by

$$L(E_x\mu) - L(E_x\mu_\star) \leq -\int \langle g, \psi(t)\rangle d(s_\star - \mu)(t).$$

With this choice of conditional gradient, the fully-corrective step is a finite-dimensional convex problem. Fixing the knot locations encountered as conditional gradients so far, $t_1, \ldots, t_{2k}$, we can do at least as well as the fully-corrective algorithm by solving the following optimization problem:

$$\begin{array}{ll}\text{minimize} & L(E_x\mu) \\ \text{subject to} & \int d\mu = 0, \\ & \|\mu\|_1 \leq \tau, \\ & \text{supp}(\mu) \subset \{t_1, \ldots, t_{2k}\}.\end{array} \tag{18}$$

This is equivalent to the following optimization problem in $\mathbf{R}^{2k}$:

$$\begin{array}{ll}\text{minimize} & L(\sum_j w_j E_x \delta_{t_j}) \\ \text{subject to} & 1^T w = 0, \\ & \|w\|_1 \leq \tau.\end{array} \tag{19}$$

We can solve this using any of a number of existing algorithms (Boyd et al., 2011; den Berg and Friedlander, 2011). In our implementation we use the conditional gradient method with line-search for simplicity.

By warm starting with an increasing sequence of τ's, we can efficiently compute an approximate regularization path. Indeed we can even provide a provably ϵ-suboptimal path using the approach of Giesen et al. (2012).

4.3 Convergence

As in the case of ADCG (Boyd et al., 2017) convergence follows immediately from the conditional gradient method proof in general Banach spaces (Dunn and Harshbarger, 1978; Demyanov and Rubinov, 1973; Jaggi, 2013.). The convergence of the conditional gradient method depends on a curvature parameter C_f. C_f is a constant such that the following inequality is satisfied for all $x, s \in \mathcal{S}$ and $\eta \in (0, 1)$:

$$f(x + \eta(s - x)) \leq f(x) + \eta f'(s - x; x) + \frac{C_f}{2}\eta^2.$$

For our purposes $f : \mathbf{R}^n \to \mathbf{R}$ is simply L and $\mathcal{S} = \{E_x\mu : \|\mu\|_1 \leq \tau, \int d\mu = 0\}$. A simple sufficient condition for C_f to be finite is that ℓ is differentiable with Lipschitz gradient. If C_f is finite, the conditional gradient method converges (in terms of function value) at a rate of at least $1/m$ where m is the iteration counter.

5. Generalized Additive Models

One natural application of univariate splines is fitting generalized additive models (Hastie and Tibshirani, 1990) to multivariate data: $(x_i, y_i) \in \mathbf{R}^D \times \mathcal{Y}$, $i = 1, \ldots, n$. That is, fitting a function of the form

$$f(x) = \sum_{d=1}^{D} f_d(x[d])$$

where each f_d is a simple function from $\mathbf{R}$ to $\mathbf{R}$ (here $x[d]$ is the d-th coordinate of the vector x). We can mimic our approach in the scalar case with the following optimization problem:

$$\begin{array}{ll} \text{minimize} & L(f) \\ \text{subject to} & \sum_d R(f_d) \leq \tau, \\ & f_d'(x) = 0 \ \forall x \notin [0,1], d. \end{array} \tag{20}$$

Here R is the same regularizer used in the scalar case, namely

$$R(g) = \mathrm{TV}(g') \simeq \|g''\|_1.$$

As in the scalar case, one can show that there is always an optimal f with each coordinate function f_d a degree-one saturating spline.

This allows us to rephrase (20) as an optimization problem over measures. The only change from the scalar case is that the measure is over the set $\{1, \ldots, D\} \times [0, 1]$—each knot is now attached to a particular coordinate. In other words, we search for a function of the following form:

$$f_\mu(x) = \int (x[d] - t)_+ d\mu(d, t).$$

We again have equality between the ℓ_1 norm of μ and the regularization term:

$$\sum_d R((f_\mu)_d) = \|\mu\|_1.$$

The analogue of (11) is then

$$\begin{array}{ll} \text{minimize} & L(E_x \mu) \\ \text{subject to} & \int 1(d = \hat{d})\ d\mu(d, t) = 0, \ \forall \hat{d} \\ & \|\mu\|_1 \leq \tau. \end{array} \tag{21}$$

The conditional gradient algorithm from the scalar case generalizes immediately to fitting generalized additive models—the only difference is that we now need to find a pair of knots for the same coordinate. This involves solving d pairs of nonconvex optimization problems over $[0, 1]$—again this can be done by gridding or by sorting the training data.

Saturating splines gain an additional advantage over standard adaptive splines when fitting generalized additive models. The addition of the saturation constraint (that f_d be constant outside of $[0, 1]$) naturally leads to variable selection when fitting generalized additive models. What we mean by variable selection is that the functions f_d are often exactly 0. This is because the saturation constraint means that linear coordinate functions no longer escape the regularization (indeed, they are impossible). This is very different

from the standard adaptive spline setup without the saturation constraint. In that case, linear functions, i.e., $f_d(x[d]) = wx[d]$ completely escape the regularization, and as a result are essentially always included in the model. Linear functions are *not* free with saturation constraints (in fact, outside of the function 0, they are not feasible). When we solve (21) we simultaneously fit nonlinear coordinate functions while doing variable selection.

6. Prior and Related Work

Smoothing splines also have an interpretation as the solution of an infinite-dimensional optimization problem (Hastie et al., 2001, §5.4). In fact, (degree-one) smoothing splines solve

$$\begin{array}{ll} \text{minimize} & L(f) \\ \text{subject to} & \hat{R}(f) \leq \tau, \end{array} \tag{22}$$

where

$$\hat{R}(f) = \int f'(x)^2\ dx.$$

The solution to (22) is *also* a degree-one natural spline that saturates outside of $[0, 1]$. However, the solutions to (22) and (2) are *very* different. Roughly, (22) is analogous to ridge regression, while (2) is analogous to the lasso. That is, (22) fits functions with as many knots as datapoints, while (2) often fits splines with very few knots.

Another type of spline, that is adaptive but does not saturate, are adaptive regression splines (Mammen and van de Geer, 1997). These splines also arise as solutions to a functional regression problem:

$$\begin{array}{ll} \text{minimize} & L(f) \\ \text{subject to} & \hat{R}(f) \leq \tau, \end{array} \tag{23}$$

where

$$\hat{R}(f) = TV(f'(x)).$$

Note that this is (2) without the saturation constraint. Algorithms for solving (23) (for degree-one splines) are based on an extension of Theorem 1, that shows there is a solution to (23) which is actually supported on the data points x_i. Hence a lasso algorithm can be used to find the solution. This also suggests a very simple method to solve our problem (9): we fix the n knot points to be the values of the data x_i, and solve the finite-dimensional convex optimization problem to find the weights. While simple coordinate-descent methods like GLMNet (Friedman et al., 2010) will not immediately work because of the saturation constraint, they could be modified to handle the constraint.

This method does work, but can be much slower than ours since in practice the number of knots is typically much smaller than n for useful values of the regularization parameter τ, and the finite-dimensional problem with n basis functions is very poorly conditioned. With that said, the algorithm we propose—for the piecewise linear case—can be interpreted as a forward active set method for the finite dimensional problem, where we avoid explicitly evaluating all basis functions. One advantage of our measure-theoretic approach is that it immediately generalizes to higher-degree splines, where the support of μ need not be on data points, as we will see in §9. In this case (9) is truly infinite-dimensional, yet our algorithm can still be directly applied.

Trend filtering is a nonparametric function estimation technique, first introduced by Kim et al. (2009), that is very similar to adaptive splines. Indeed, as discussed by Tibshirani (2014), the trend filtering estimate in the constant or piecewise-linear case is exactly the same as the adaptive spline estimate. Trend filtering is increasingly popular as it admits extremely efficient, robust algorithms (Tibshirani, 2014; Ramdas and Tibshirani, 2015). Indeed, some of these algorithms (especially those adapted to fit GAMs, Petersen, Witten, and Simon (2015)) may be adapted to efficiently fit saturating trend filter estimates, which would benefit from the feature selection properties of saturating splines and the computational efficiency of trend filtering.

There are a number of methods for fitting generalized additive models with spline component functions. One approach (Lin and Zhang, 2006) is to use the group-lasso version of (6):

$$R(f) = \sum_d \sqrt{\int f_d'(x)^2 \, dx}.$$

Extending this idea, Chouldechova and Hastie (2015) use an overlap group-lasso that facilitates selection between zero, linear and nonlinear terms. The differences between these approaches and ours are analogous to the differences between the standard group-lasso and the lasso. While both do feature selection, the penalty functional (6) does not do knot-selection within each coordinate function.

One very similar approach to fitting splines that does not require knot selection (but does not incorporate saturation) is discussed by Rosset et al. (2007).

7. Examples

In all examples we affinely preprocess the data so that all training features lie in $[0, 1]$, and apply the same transformation to the test features (which thus may have values outside of $[0, 1]$). All plots are in terms of the standardized features. For the bone density and abalone data sets we select τ to minimize error on the validation sets. For the Spam and ALS data sets we use cross-validation to estimate τ. We hold out a random subset of size 100 from the training set and train on the remaining data. For each random validation/train split we estimate τ to minimize hold-out error and take our final estimate of τ as the mean over 50 trials.

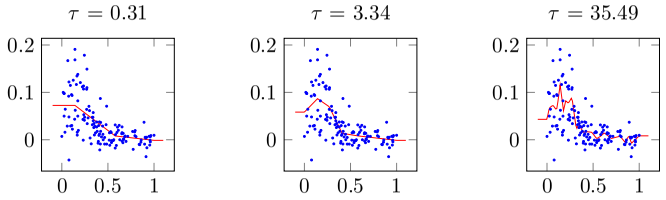


Figure 3: Saturating splines fit to bone density data (shown as scattered points) for 3 values of the regularization parameter τ. *Top:* $\tau = 0.31$; *Middle:* $\tau = 3.34$; *Bottom:* $\tau = 35.45$.

7.1 Bone Density

We start with a simple univariate data set from (Hastie et al., 2001, §5.4). The response variable for this data set is the change in spinal bone density between two doctor visits for female adolescents as a function of age. There are 259 data points, of which we hold out 120 for validation, leaving 139 data points to which we fit a saturating spine. We start with the square loss.

The results are shown in figure 3, for three values of the regularization parameter τ. The scattered points are the training data, the solid line is the saturating spline fit by our algorithm. The figure demonstrates the clear link between τ and the complexity of the optimized spline. Out-of-sample validation suggests setting $\tau \simeq 3.34$, which achieves a validation RMSE of 0.036.

To demonstrate that our proposed method works with more general loss functions, we add 30 simulated outliers to the training set and fit with the pseudo-Huber loss (Charbonnier et al., 1997), a smooth approximation to the Huber loss function given by

$$l_\delta(u) = \delta \left(\sqrt{1 + \frac{u^2}{\delta}} - 1 \right),$$

where $\delta > 0$ is a parameter that interpolates between the absolute value loss and the squared loss. For our experiment we take $\delta = 0.0015$; roughly speaking, the transition between square and linear loss occurs around $\sqrt{\delta} = 0.039$. The results are shown in figure 4. These plots demonstrate that our algorithm can fit losses other than the square loss, and confirms that the pseudo-Huber loss is far more robust to outliers than the basic square loss function. Indeed, on the validation set the least-squares fit achieves a minimum RMSE of 0.096, while the pseudo-Huber fit achieves 0.038, only slightly worse than the fit obtained before the outliers were added to the training data. While this one-dimensional problem is very easy, it shows one advantage of the adaptive spline penalty over smoothing splines: the optimal model has only 5 knot points.

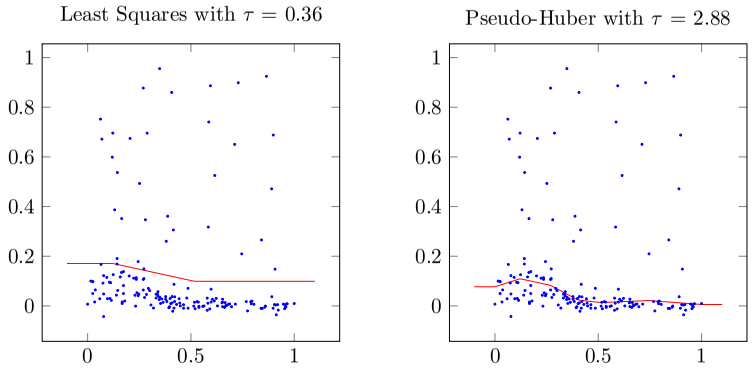


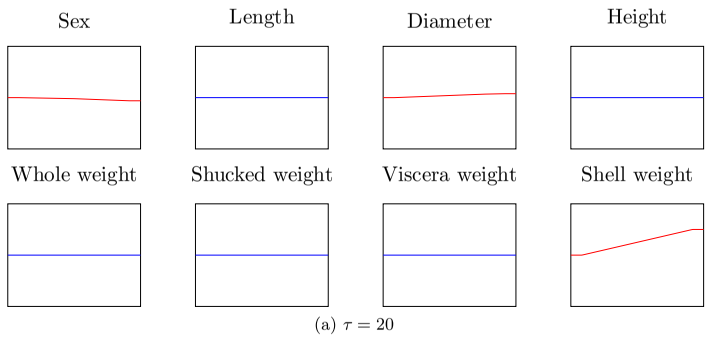
Figure 4: Saturating splines fit to bone density data (shown as scattered points) with simulated outliers for square loss function (left) and pseudo-Huber loss function (right), each for the value of τ that minimizes RMSE on the test set.

7.2 Abalone

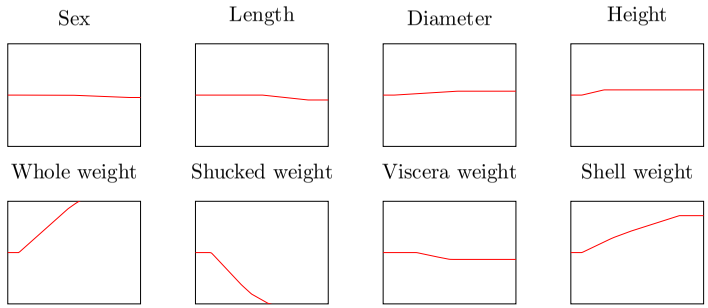
We fit a generalized additive model with saturating spline coordinate functions to the Abalone data set from the UCI Machine Learning Repository (Lichman, 2013). The data consists of 4177 observations of 8 features of abalone along with the target variable, the age of the abalone. We hold out 400 data points as a validation set, leaving 3777 data points to fit the model. The first feature (labeled sex) has three values: Male, Female, and Juvenile, which are coded with values $0, 1, 2$; the other 7 are (directly) real numbers. The task is to estimate the age of the abalone from the features.

Cross-validation suggests we choose $\tau \simeq 200$, which achieves a validation set RMSE of 2.131. Because the number of features is low, we can plot the entire generalized additive model. Each plot shows one coordinate function f_d for $d = 1, \ldots, 8$ as a function of the standardized feature in $[0, 1]$. The coordinate functions are shown for three values of τ, with the middle one corresponding to the value that minimizes cross-validation RMSE. When a coordinate function is zero, which means that the feature is not used in the model, it is shown in blue. We can see that in the case of strong regularization ($\tau = 20$), several coordinates are not used; for the best model ($\tau = 200$), all features are used, with a few having only a small effect. It is interesting to see how the sex factors into the optimal model. It is neutral on Male or Female, but subtracts a small fixed amount from its age prediction for a Juvenile abalone.

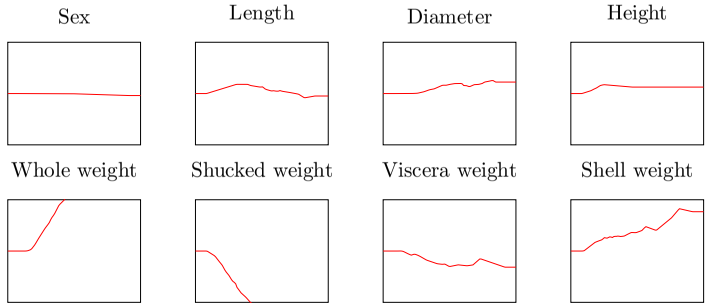
This data set is small enough that we can compare against standard adaptive splines fit using a coarse grid of $[0, 1]$. For this experiment, we fit a GAM with standard adaptive spline component functions using GLMNET (Friedman et al., 2010). The standard adaptive GAM fit, which does no variable selection, achieves a validation set RMSE of 2.137, not significantly worse than the saturating spline model. Our algorithm, however, selects many fewer knot points. The increased number of knots when fitting with GLMNET is perhaps due to the poor conditioning of the gridded problem.



(a) $\tau = 20$



(b) $\tau = 200$



(c) $\tau = 2000$

Figure 5: Coordinate functions for saturating spline generalized additive models fit to Abalone data for three values of the regularization parameter τ.

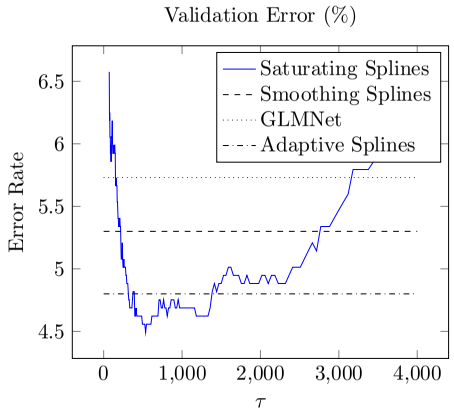


Figure 6: Validation error for saturating spline generalized additive model fit to Spam data set versus regularization parameter τ.

7.3 Spam

We consider the problem of classifying email into spam/not spam, with a data set taken from ESL (Hastie et al., 2001). The data set consists of 57 word-frequency features from 4601 email messages, along with their labels as spam or not spam. Following the approach in ESL (Hastie et al., 2001) we log-transform the features and use the standard train/validation split, with a training set of size 3065, and test set with 1536 samples. We fit a saturating spline generalized additive model with standard logistic loss.

Figure 6 shows the validation error versus the regularization parameter τ. Cross-validation suggests the choice $\tau \simeq 1100$. To show the benefit of nonlinear coordinate functions, we also include the best validation error achieved using a linear model (fit using GLMNet Friedman, Hastie, and Tibshirani (2010)).

With regularization parameter $\tau = 500$, the model selects 55 of the 57 features. We note that our saturating spline generalized additive model modestly outperforms many methods from ESL (Hastie et al., 2001); for example, smoothing splines yield 5.3% error, while our model has an error rate well below 5%. Figure 7 shows (some of) the coordinate functions for the model with $\tau = 500$. The coordinate functions use very few knots, making them readily interpretable.

For comparison, we fit a GAM with standard adaptive spline coordinate functions. To do so, we grid each dimension with 20 knots and solve the resulting finite-dimensional problem with GLMNET (Friedman et al., 2010). Note that adaptive splines do not penalize linear functions, so there is no feature selection. Adaptive splines achieve a minimum error of 4.8%, significantly worse than saturating splines.

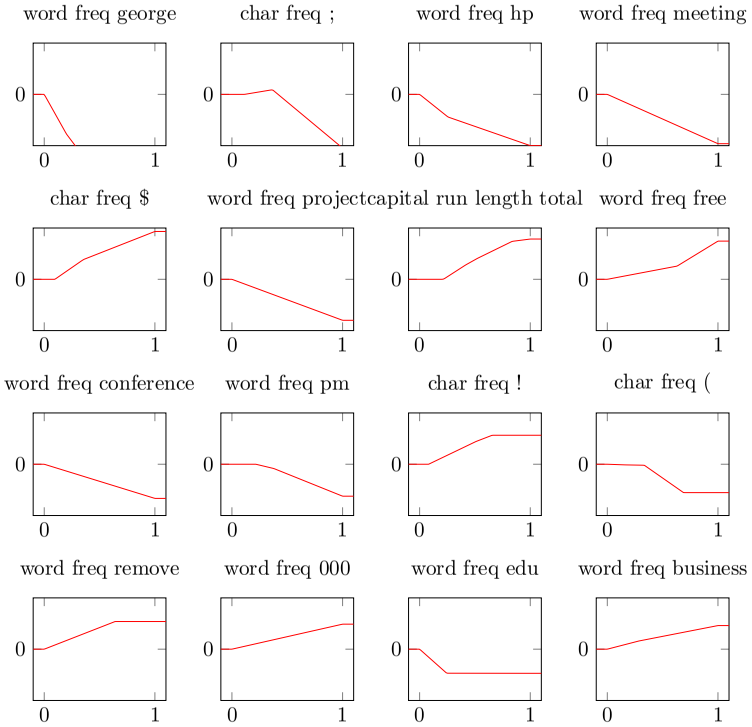


Figure 7: 16 coordinate functions for $\tau = 500$, labeled with the corresponding feature name.

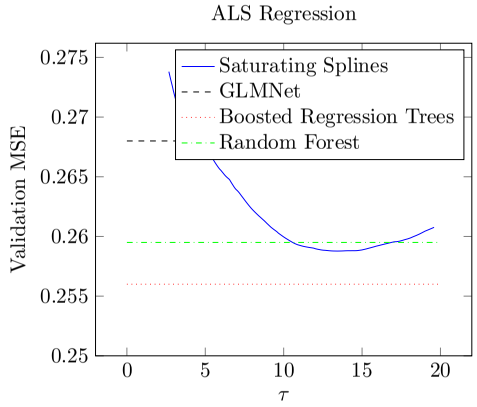


Figure 8: Validation MSE on ALS data set versus regularization parameter τ.

7.4 ALS

Using this data set we try to predict the rate of progression of ALS (amyotrophic lateral sclerosis) in medical patients, as measured by the rate of change in their functional rating score, a measurement of functional impairment. The data set is split into a training set of 1197 examples and a validation set of 625 additional patients. Each datapoint has dimension 369. We fit a generalized additive model with saturating spline component functions to the data using a least-squares objective function. Following (Efron and Hastie, 2016, §17.2), we measure performance using mean-squared error.

We estimate the optimal value of τ using cross validation with a hold-out size of 100 examples and 50 samples; this procedure suggests $\tau = 13$. Figure 8 shows the validation error versus the regularization parameter τ; the value of τ selected by cross validation achives low error. On the same plot, we also show the results from Efron and Hastie (2016) using boosted regression trees and random forests. The optimal saturating spline GAM model selects only 50 out of the 369 features, in contrast to boosted regression trees, which use 267. The saturating spline GAM model performs comparably to boosted regression trees and random forests. This is surprising as the saturating spline GAM has no interaction terms. It also uses substantially fewer features, further improving interpretability.

Again we fit a GAM with standard adaptive spline coordinate functions (using GLMNET) to show the advantage of saturation. The standard adaptive spline fit achieves an MSE of 0.547, substantially worse than any other model. We speculate that this is because the unpenalized linear functions lead to immediate overfitting. Indeed, removing the unpenalized linear functions and fitting a model with only hinges gives very similar performance to the saturating spline fit, suggesting that the main advantage of saturation for this application is the removal of the unpenalized linear functions.

Practical advantages of saturating splines These experiments show that saturating splines achieve competitive performance on small classification and regression data sets. In addition, the experiments demonstrate that saturating splines exhibit both knot selection and feature selection—in the context of fitting GAMs. While it is no surprise that saturating splines select fewer knots than smoothing splines (which choose a fully-dense set of knots), it is somewhat surprising that our algorithm selects fewer knots than even *adaptive* splines fit with GLMNET. Finally, the Spam and ALS data sets demonstrate a major advantage of saturating splines over adaptive splines: they simultaneously perform non-linear coordinate function fitting and feature selection. This aids in generalization performance and interpretability. In particular, for the ALS data set saturating spline GAMs achieve *half* the test MSE of adaptive spline GAMs by selecting only 50 of 369 available features.

8. Higher-degree Splines

In the majority of this paper we focused on the functional regression problem (2), with a total variation constraint on the first derivative and a saturation constraint on the zeroth derivative (the function itself). In this section, we consider constraints on higher order derivatives, which lead to solutions that are splines of higher degrees.

$$\begin{array}{ll} \text{minimize} & L(f) \\ \text{subject to} & \mathrm{TV}(f^{(k)}) \leq \tau, \\ & f^{(k-j)}(x) = 0, \ \forall x \notin [0,1]. \end{array} \tag{24}$$

We consider the family of nonparametric function estimation problems indexed by $0 \leq j \leq k$. This is the analogue of the functional regression problem (2) with a total variation constraint on the k-th derivative and a saturation constraint on the $(k-j)$-th derivative. The saturating spline case from the rest the paper is the special case of (24) with $k = 1$, $j = 0$. Widely used cubic natural splines correspond to $k = 3$, $j = 1$. Note that unlike natural splines, which are only defined for some values of j and k, there are no constraints on j and k.

We now show that higher-degree saturating splines solve (24) in general. As $f^{(k)}$ is of bounded TV, there exists a measure μ s.t. $f^{(k)}(x) = \int 1(t \leq x)\, d\mu(t)$. Then we have

$$\begin{aligned} f^{(k-j)}(x) &= \int \dots \int f^{(k)}(x) dx \dots dx \\ &= \int \dots \int \int 1(t \leq x)\, d\mu(t) dx \dots dx \\ &= j! \int (x-t)_+^j \, d\mu(t) + \sum_{l=0}^{j-1} w_l x^l \end{aligned}$$

for some w_l. In the above, all iterated integrals take place j times.

Note that the constraint that $f^{(k-j)}(x) = 0$ for all $x < 0$ implies that the polynomial term, $\sum_{l=0}^{j-1} w_l x^l$ is identically zero. So, we have

$$f^{(k-j)}(x) = j! \int (x-t)_+^j \, d\mu(t).$$

For $x > 1$, we can remove the nonlinearity, that is, for $x > 1$, $f^{(k-j)}(x)$ is simply the integral of a polynomial in x. We can pull terms involving x out of the integral to get a polynomial in x whose coefficients are nonzero multiples of the first j moments of μ:

$$f^{(k-j)}(x) = j! \sum_{l=0}^{j} \binom{j}{l} x^{j-k} \int (-t)^k \, d\mu(t).$$

Again, we note that as this polynomial is identically zero for infinitely many points, all of the coefficients must be zero. In terms of the measure μ, this means:

$$\int t^l \, d\mu(t) = 0 \quad \text{for } l = 0, \dots, j.$$

This shows that the constraint that the $(k-j)$-th derivative of f saturate translates to constraints on all moments of μ up to the j-th moment.

While the conditional gradient step becomes more complex with the addition of more moment constraints, the approach taken in this paper can still be applied to (24) as long as j is fairly small—the conditional gradient step for (24) involves a nonconvex optimization problem over $[0,1]^{j+2}$. This is because we need at least $(j+2)$ point-masses to satisfy the moment constraints. So, fitting quadratic splines that saturate to linear is very easy—in fact the code to do so is essentially identical to that for fitting piecewise linear saturating splines splines—but fitting quadratic splines that saturate to constant is slightly more difficult due to the additional linear constraint on the measure μ. Unfortunately for larger values of j and k, we can no longer hope to find the conditional gradient analytically and must resort to recursive gridding or other global optimization algorithms to find the locations of the new knots.

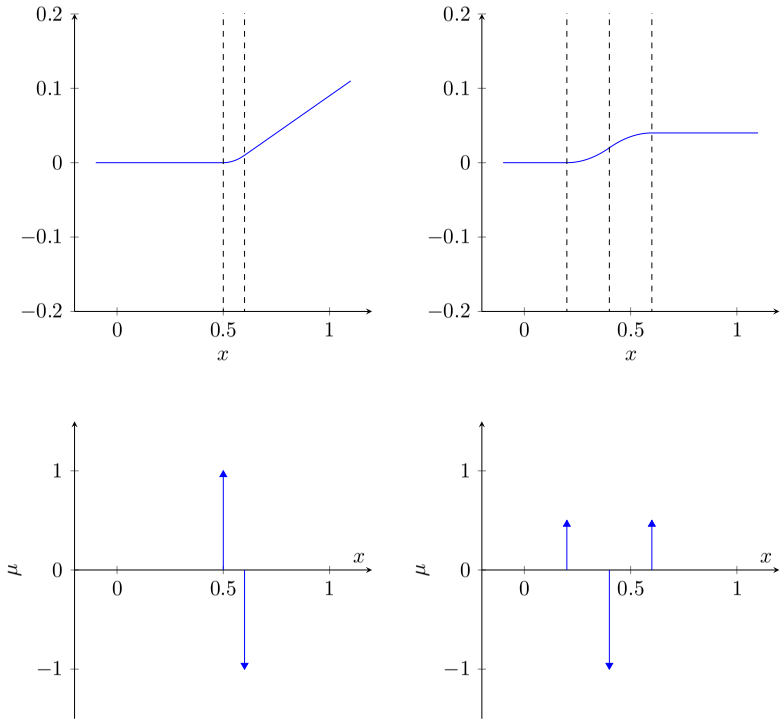


Figure 9: The top two plots show conditional gradients for $k = 2$ with $j = 0$ and $j = 1$ respectively. The dashed lines denote the locations of the point masses: when $j = 1$, the conditional gradient consists of three point masses. The bottom plots show the corresponding measures.

9. Variations and Extensions

While saturation is often a natural prior, the approach we take in this paper can also be applied to other (convex) variations on (9). For example, we could add the constraint that the fitted function is monotone nondecreasing, or takes values in a given interval.

A simple algorithmic extension would be to incorporate nonconvex optimization in the spirit of Boyd et al. (2017). At each iteration we adjust the weights of the atomic measure (w), but we could also adjust the knot locations (t). The objective in (19) is nonconvex in t_i, but we can still attempt to find a local minimum. As long as we do not increase the objective function the algorithm is still guaranteed to converge (Boyd et al., 2017). In the case of degree one splines, we can use the fact that the knot points can, without loss of generality, be chosen to be on the data points to make discrete adjustments to the knot locations.

To fit vector-valued functions, for example in multiclass classification, we would need to extend (9) to use *vector-valued* measures. This is the natural measure-theoretic analogue to the group-lasso.

In multivariate fitting problems with significant interactions between features generalized additive models may underfit. One possible solution is to use single-layer neural networks: i.e., learn functions of the form

$$x \mapsto \sum_{i=1}^{K} w_i (v_i^T x - t_i)_+.$$

In the above, v_i are constrained to lie in the unit ball. Unfortunately, the conditional gradient step for networks of this form is NP-hard (Bach, 2017). In many practical applications, however, we might expect that the degree of the interaction is bounded. That is, each v_i has bounded cardinality. If we assume $\|v_i\|_0 \leq 2$, i.e., we only fit pairwise interactions, we can still apply the conditional gradient method. In this case, the fitting function is a sum of functions of pairs of the variables, formed from the basis elements

$$((\cos\theta)x_p + (\sin\theta)x_q - t)_+,$$

with (continuous) parameters θ and t and (index) parameters p and q (i.e., $v = (\cos\theta)e_p + (\sin\theta)e_q$). (This is practical only if d is small enough.) Such functions capture nonlinear relationships between (pairs of) variables.

10. Conclusion

In this paper we propose a modification of the adaptive spline regression model—namely saturation constraints. We show that saturating splines inherit knot-selection from adaptive splines, and have a very important quality in the context of generalized additive models: feature selection. This allows saturating spline generalized additive models to remain interpretable and (crucially) avoid overfitting when applied to multivariate data. We also propose a simple, effective algorithm based on the standard conditional gradient method for solving the saturating spline estimation problem with arbitrary convex losses. Finally, we apply our algorithm to several data sets, demonstrating the simplicity of the resulting models.

Acknowledgments

We would like to thank Aaditya Ramdas for many helpful discussions about trend filtering. NB was generously supported by a Google Fellowship from the Hertz Foundation.

Appendix A. Implementation Details

We provide a simple, unoptimized implementation in the Rust language. The runtime of our algorithm is dominated by the fully-corrective step, that is, solving the finite-dimensional convex optimization problem (19). We solve (19) using a proximal Newton method and the standard conditional gradient method with exact linesearch. To be precise, at each iteration, we form the second-order approximation to the objective function

$$f(w) \simeq C + (w - \hat{w})^T \nabla_w f(\hat{w}) + \frac{1}{2}(w - \hat{w})^T \nabla^2 f(\hat{w})(w - \hat{w})$$

which we then minimize (over the constraint set) using the standard conditional gradient method with (exact) linesearch. Note that this is a Newton step with fixed step-length of 1: as in GLMNET (Friedman et al., 2010), we omit a line search in the interest of speed.

We chose to use a proximal Newton method because of its relative simplicity; other standard convex optimization algorithms may give much better practical performance, especially when the number of data points, n, is extremely large.

Appendix B. Saturating Hinges

In this section we introduce a heuristic for solving an approximation to (9) using existing algorithms for the lasso. Here we consider the case where $\hat{R}(f) = TV(f'(x))$, and hence as pointed out in Section 6 the solution is an expansion in piecewise linear splines with knots at the unique data points. Let h_j be a hinge function at knot t_j: $h_j(x) = (x - t_j)_+$, and suppose we have knots $t_1 < t_t < \cdots < t_k$. Define $f(x) = w_0 + \sum_{j=1}^k w_j h_j(x)$. Given a sample $T = \{(x_i, y_i)\}_1^N$, solving (9) amounts to solving

$$\underset{w_0, w}{\text{minimize}} \left\{ \sum_{i=1}^N \ell(y_i, f(x_i)) + \lambda \|w\|_1 \right\} \quad \text{s.t. } f'(t_k) = 0. \tag{25}$$

Here we've exchanged a constraint on the total variation of $TV(f'(x))$ with a penalty. The condition $f'(t_k) = 0$ is equivalent to $\sum_{j=1}^k w_j = 0$. If the points x_i are unique, then $k = n$; irrespective $t_1 = \min(x_i)$, $t_k = \max(x_i)$ and by construction the estimate $\hat{f}$ is constant beyond the data.

Without the gradient condition $f'(t_k) = 0$, solving (25) amounts to a large lasso problem, for which efficient software is available. Here our goal is to transform the problem to get rid of this constraint. For more generality we do this for an arbitrary set of ordered knots.

Suppose $k < n$ and the right-most knot is inside the range of the data. Consider the following formulation. Let $s_j(x) = h_j(x) - h_k(x)$, a "saturating" hinge function. It looks like a piecewise linear sigmoid, and goes horizontal at t_k (see figure 10).

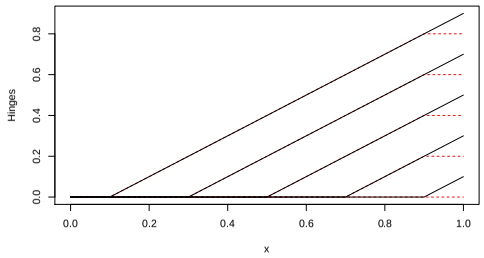


Figure 10: Hinges and saturating hinges.

Without the ℓ_1 constraint (or when $\lambda = 0$) the solution to (25) is equivalent to the solution to the problem with the reduced basis $g(x) = \theta_0 + \sum_{j=1}^{k-1} \theta_j s_j(x)$:

$$\underset{\theta_0, \theta}{\text{minimize}} \sum_{i=1}^N \ell(y_i, g(x_i)). \tag{26}$$

This is easy to see. f is an affine expansion in the h_j, and hence any nonsingular $k \times k$ transformation C of the vector of functions $h(x) = (h_1(x), h_2(x), \ldots, h_k(x))$ spans the same space. It is easy to see that with $s(x) = C^T h(x)$, and

$$C = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ -1 & -1 & \cdots & -1 & -1 \end{bmatrix}$$

that $s_j(x)$ are as described (and $s_k(x) = -h_k(x)$). Now $h(x)^T w = h(x)^T C C^{-1} w = s(x)^T \theta$, with $\theta = C^{-1} w$. However, in this case $C^{-1} = C$, and hence $\theta_j = w_j$, $j = 1, \ldots, k-1$ and $\theta_k = \sum_{j=1}^k w_j$. In this new basis, imposing the constraint amounts to setting $\theta_k = 0$, or simply deleting the last basis function. So fitting the linear model f subject to $f'(t_k) = 0$ is equivalent to fitting the model g without constraints, and in fact the $\theta_j = w_j$, $j = 1, \ldots, k-1$.

So in summary, fitting a constrained optimization with the hinge functions is equivalent to fitting an unconstrained optimization with the reduced set of saturating hinges. The remaining question is does this also work with the penalty $\lambda\|w\|$ as in (25). Not quite, but close. It turns out we are still missing a penalty term $\lambda|\sum_{j=1}^{k-1} \theta_j|$. Hence the transformed problem is

$$\underset{\theta_0, \theta \in \mathbf{R}^{k-1}}{\text{minimize}} \left\{ \sum_{i=1}^N \ell(y_i, g(x_i)) + \lambda\|\theta\| + \lambda|\textstyle\sum_{j=1}^{k-1} \theta_j| . \right\} \tag{27}$$

If we are willing to ignore this last penalty, we can fit the saturated spline model using a fast lasso solver, such as `glmnet`. By generating such a basis for each variable in a GAM, this same approach can be used to fit a saturated GAM regularization path.

The impact is that one could fit a saturated gam model by running say `glmnet` on the $\{s\}$ bases. An example is given on the spam data in figure 11, where the regularization path was computed at a 100 values of λ in seconds. There are of course some caveats.

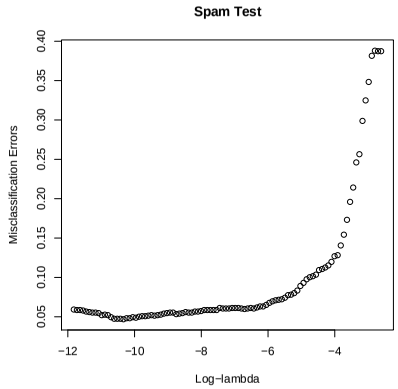


Figure 11: Performance on the spam test data, with 20 knots per variable. Increasing to 50 did not make much difference. Minimum error is 0.047.

- If you use all the knots for each of p variables, you end up with a data matrix of dimension $N \times Np$, which does not scale too well. So instead one might use a smaller grid of knots; e.g., map the variables onto $[0, 1]$, and then use a grid of say 50 evenly spaced knots on this grid, including the end knots.

- With a large number of knots, the "variables" are highly correlated, and this can cause numerical issues. The main issue we see is that active sets tend to be larger than they should be.

Nevertheless, this is an alternative algorithm, which is closer in spirit to the adaptive splines algorithm.

Appendix C. Proof of Theorem 1

Theorem 1 *Fix $x_1, \ldots, x_n \in [0, 1]$ and $f : \mathbf{R} \to \mathbf{R}$ with f' (right-continuous) of bounded total variation, and f constant outside of $[0, 1]$. Then there exists a degree-one saturating spline $\hat{f}$ that matches f on x_i with $\mathrm{TV}(\hat{f}') \leq \mathrm{TV}(f')$.*

Proof Without loss of generality, we will assume $f(0) = 0$. Let $\tau = \mathrm{TV}(f')$. As f' has bounded total variation, there exists a measure μ on $[0, 1]$ such that $f(x) = f_\mu$:

$$f(x) = \int (x - t)_+ \, d\mu(t).$$

That is, f is a spline with infinitely many knots. The idea is to use Caratheodory's theorem for convex hulls to see that, as we only care about μ in terms of its action on a finite number of functions (basically, we only care about the values of f at x_i), we can replace μ with a measure supported on finitely many points.

To make this idea rigorous, note that the vector

$$v = (f(x_1), \ldots, f(x_n), 0) = \int ((x_1 - t)_+, \ldots, (x_n - t)_+, 1) \, d\mu(t)$$

must lie in convex hull of the (convex) set

$$C = \{\pm(\tau(x_1 - t)_+, \ldots, \tau(x_n - t)_+, \tau) : t \in [0, 1]\} \subset \mathbf{R}^{n+1}$$

as $\|\mu\|_1 = \tau$. Caratheodory's theorem for convex hulls ensures us that v can be represented as a convex combination of at most $n + 2$ points from C. Letting these $n + 2$ points be represented by their indicies, $t_1, \ldots, t_{n+2}$, and their weights $\alpha_1, \ldots, \alpha_{n+2}$ we define $w_j = \alpha_j \tau$ to obtain:

$$f(x_i) = \sum_j w_j (x_i - t_j)_+ = f_\mu(x_i)$$
$$\sum_j w_j = 0.$$

Here $\mu = \sum_j w_j \delta_{t_j}$. As $\sum_j |w_j| = \tau$, we have $\mathrm{TV}(f'_\mu) = \|\mu\|_1 = \tau$. ■

References

F. Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(19):1–53, 2017.

N. Boyd, G. Schiebinger, and B. Recht. The alternating descent conditional gradient method for sparse inverse problems. *SIAM Journal on Optimization*, 27(2):616–639, 2017.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud. Deterministic edge-preserving regularization in computed imaging. *Trans. Img. Proc.*, 6(2):298–311, February 1997. ISSN 1057-7149.

A. Chouldechova and T. Hastie. Generalized additive model selection. *ArXiv e-prints*, June 2015.

C. De Boor. *A Practical Guide to Splines*. Applied mathematical sciences. Springer, Berlin, 2001.

V. F. Demyanov and A. M. Rubinov. Approximate methods in optimization problems. *Journal of Applied Mathematics and Mechanics*, 53(7):499–499, 1973. ISSN 1521-4001.

E. Van den Berg and M. P Friedlander. Sparse optimization with least-squares constraints. *SIAM Journal on Optimization*, 21(4):1201–1229, 2011.

J.C. Dunn and S. Harshbarger. Conditional gradient algorithms with open loop step size rules. *Journal of Mathematical Analysis and Applications*, 62(2):432 – 444, 1978. ISSN 0022-247X.

B. Efron and T. Hastie. *Computer Age Statistical Inference*. Institute of Mathematical Statistics Monographs. Cambridge University Press, 2016. ISBN 9781107149892.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010. ISSN 1548-7660.

J. Giesen, M. Jaggi, and S. Laue. Approximating parameterized convex optimization problems. *ACM Trans. Algorithms*, 9(1):10:1–10:17, December 2012. ISSN 1549-6325.

P. J. Green and B. W. Silverman. *Nonparametric Regression and Generalized Linear Models: a Roughness Penalty Approach*. Monographs on statistics and applied probability. Chapman & Hall, Boca Raton, London, New York, 1994. ISBN 0-412-30040-0.

T. Hastie and R. Tibshirani. *Generalized Additive Models*, volume 43. CRC Press, 1990.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. *ICML*, 2013.

S. Kim, K. Koh, S. Boyd, and D. Gorinevsky. ℓ_1 trend filtering. *SIAM review*, 51(2): 339–360, 2009.

M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Y. Lin and H. Zhang. Component selection and smoothing in multivariate nonparametric regression. *Ann. Statist.*, 34(5):2272–2297, 10 2006.

E. Mammen and S. van de Geer. Locally adaptive regression splines. *Ann. Statist.*, 25(1): 387–413, 02 1997.

A. Petersen, D. Witten, and N. Simon. Fused lasso additive model. *Journal of Computational and Graphical Statistics*, pages 1–37, 2015.

A. Ramdas and R.J. Tibshirani. Fast and flexible admm algorithms for trend filtering. *Journal of Computational and Graphical Statistics*, 2015.

S. Rosset, G. Swirszcz, N. Srebro, and J. Zhu. ℓ_1 regularization in infinite dimensional feature spaces. pages 544–558, 2007.

R.J. Tibshirani. Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics*, 42(1):285–323, 2014.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

G. Wahba. *Spline Models for Observational Data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1990.

Journal of Machine Learning Research 18 (2018) 1-42 Submitted 6/17; Revised 11/17; Published 4/18

Nonasymptotic convergence of stochastic proximal point methods for constrained convex optimization

Andrei Patrascu ANDREI.PATRASCU@FMI.UNIBUC.RO
Department of Computer Science
University of Bucharest
Str. Academiei 14, 010014 Bucharest

Ion Necoara ION.NECOARA@ACSE.PUB.RO
Automatic Control and Systems Engineering Department
University Politehnica of Bucharest
Spl. Independentei 313, 060042 Bucharest

Editor: Mark Schmidt

Abstract

A popular approach for solving stochastic optimization problems is the stochastic gradient descent (SGD) method. Although the SGD iteration is computationally cheap and its practical performance may be satisfactory under certain circumstances, there is recent evidence of its convergence difficulties and instability for unappropriate choice of parameters. To avoid some of the drawbacks of SGD, stochastic proximal point (SPP) algorithms have been recently considered. We introduce a new variant of the SPP method for solving stochastic convex problems subject to (in)finite intersection of constraints satisfying a linear regularity condition. For the newly introduced SPP scheme we prove new nonasymptotic convergence results. In particular, for convex Lipschitz continuous objective functions, we prove nonasymptotic convergence rates in terms of the expected value function gap of order $\mathcal{O}\left(\frac{1}{k^{1/2}}\right)$, where k is the iteration counter. We also derive better nonasymptotic convergence rates in terms of expected quadratic distance from the iterates to the optimal solution for smooth strongly convex objective functions, which in the best case is of order $\mathcal{O}\left(\frac{1}{k}\right)$. Since these convergence rates can be attained by our SPP algorithm only under some natural restrictions on the stepsize, we also introduce a restarting variant of SPP that overcomes these difficulties and derive the corresponding nonasymptotic convergence rates. Numerical evidence supports the effectiveness of our methods in real problems.

Keywords: Stochastic convex optimization, intersection of convex constraints, stochastic proximal point, nonasymptotic convergence analysis, rates of convergence.

1. Introduction

The randomness in most of the practical optimization applications led the stochastic optimization field to become an essential tool for many applied mathematics areas, such as machine learning (Polyak and Juditsky, 1992), distributed optimization (Necoara et al., 2011), control (Karimi and Kammer, 2017), and sensor networks problems (Blatt and Hero, 2006). Since the randomness usually enters the problem through the cost function and/or the constraints set, in this paper we approach both randomness sources and consider stochastic objective functions subject to stochastic constraints. Usually, in the literature, the following unconstrained stochastic model has been considered:

$$\min_{x\in\mathbb{R}^n} F(x) = (\mathbb{E}[f(x;S)]), \tag{1}$$

where the expectation is taken w.r.t. the random variable S. In the following subsections, we recall some popular numerical optimization algorithms for solving the previous unconstrained stochastic optimization problem and set the context for our contributions.

1.1 Previous work

A very popular approach for solving the unconstrained stochastic problem (1) is the stochastic gradient method (SGD) (Nemirovski et al., 2009; Moulines and Bach, 2011; Rosasco et al., 2014; Polyak and Juditsky, 1992). At each iteration k, the SGD algorithm randomly samples S and takes a step along the gradient of the chosen individual function:

$$x^{k+1} = x^k - \mu_k \nabla f(x^k; S_k),$$

where μ_k is a positive stepsize. Convergence behavior of SGD for the last iterate sequence has been analyzed in (Nemirovski et al., 2009) and for the average of the iterates sequence has been given in (Polyak and Juditsky, 1992). However, there is a recent nonasymptotic convergence analysis of SGD provided in (Moulines and Bach, 2011), under various differentiability assumptions on the objective function. While the SGD scheme is the method of choice in practice for many machine learning applications due to its superior empirical performance, the theoretical estimates obtained in (Moulines and Bach, 2011) highlights several difficulties regarding its practical limitations and robustness. For example, the stepsize is highly constrained to small values by an exponential term from the convergence rate which could be catastrophically increased by uncontrolled variations of the stepsize. More precisely, the convergence rates of SGD with decreasing stepsize $\mu_k = \frac{\mu_0}{k}$, given for the quadratic mean $\{\mathbb{E}[\|x^k - x^*\|^2]\}_{k\geq 0}$, where x^* is the optimal solution of (1), contains certain exponential terms (depending on the initial stepsize) of the following form (Moulines and Bach, 2011):

$$\mathbb{E}[\|x^k - x^*\|^2] \leq \frac{C_1 e^{C_2\mu_0^2}}{k^{\alpha\mu_0}} + \mathcal{O}\left(\frac{1}{k}\right), \tag{2}$$

for $\mu_0 > 2/\alpha$ and for appropriate positive constants C_1, C_2 and α. Note that this convergence rate holds under strong convexity and gradient Lipschitz assumptions on the objective function F. From (2) we observe that $\{\mathbb{E}[\|x^k - x^*\|^2]\}_{k\geq 0}$ can grow exponentially until the stepsizes becomes sufficiently small, a behavior which can be also observed in practical simulations.

Since these drawbacks are naturally introduced by the SGD iteration, other essential modifications of this scheme have been applied for avoiding the issues. One resulted method is the stochastic proximal point (SPP) algorithm for solving the unconstrained stochastic problem (1) having the following iteration (Ryu and Boyd, 2016; Toulis et al., 2016; Bianchi, 2016):

$$x^{k+1} = \arg\min_{z\in\mathbb{R}^n} \left[f(z;S_k) + \frac{1}{2\mu_k}\|z - x^k\|^2 \right].$$

Note that SGD represents a particular SPP iteration applied to the linearization of $f(z;S_k)$ in x^k, that is to the linear function $l_f(z;x^k,S_k)=f(x^k;S_k)+\langle\nabla f(x^k;S_k),z-x^k\rangle$. Of course, when f has an easily computable proximal operator, it is natural to use f instead of its linearization l_f. In (Ryu and Boyd, 2016), the SPP algorithm has been applied to problems with the objective function having Lipschitz continuous gradient and the following *restricted strong convexity* property:

$$f(x;S)\geq f(y;S)+\langle\nabla f(y;S),x-y\rangle+\frac{1}{2}\langle M_S(x-y),x-y\rangle\quad\forall x,y\in\mathbb{R}^n, \tag{3}$$

for some matrix $M_S\succeq 0$, satisfying $\lambda=\lambda_{\min}(\mathbb{E}[M_S])>0$. In (Ryu and Boyd, 2016) the asymptotic global convergence of SPP with decreasing stepsize $\mu_k=\frac{\mu_0}{k}$ is derived, followed by a nonasymptotic analysis for the SPP with constant stepsize. In particular, it has been proven that SPP converges linearly to a noise-dominated region around the optimal solution. Moreover, the following *asymptotic* (i.e. for a *sufficiently large* k) convergence rate in the quadratic mean have been given:

$$\mathbb{E}[\|x^k-x^*\|^2]\leq\left(\frac{1}{e}\right)^{\mu_0\lambda\ln(k+1)}C_1+\begin{cases}\frac{C_2}{(\mu_0\lambda-1)k} & \text{if}\quad \mu_0\lambda>1\\ \frac{C_2\ln(k)}{k} & \text{if}\quad \mu_0\lambda=1\\ \frac{C_2}{(1-\mu_0\lambda)k^{\mu_0\lambda}} & \text{if}\quad \mu_0\lambda<1,\end{cases}$$

where C_1 and C_2 are some positive constants. With the essential difference that no exponential terms depending on μ_0 are encountered, these rates of convergence have similar orders with those for the variable stepsize SGD method. Although in this paper we make similar assumptions on the objective function, we additionally assume the presence of *convex constraints* and provide a *nonasymptotic* convergence analysis of the SPP for a more general stepsize $\mu_k=\frac{\mu_0}{k^\gamma}$, with $\gamma>0$. Moreover, the Moreau smoothing framework used in our paper leads to more elegant and intuitive proofs. Another paper related to the SPP algorithm is (Toulis et al., 2016), where the considered stochastic model involves minimization of the expectation of random particular components $f(x;S)$ defined by the composition of a smooth function and a linear operator, i.e.:

$$f(x;S)=f(a_S^Tx),$$

where $a_S\in\mathbb{R}^n$. Moreover, the objective function $F(x)=\mathbb{E}[f(a_S^Tx)]$ needs to satisfy $\lambda_{\min}\left(\nabla^2F(x)\right)\geq\lambda>0$ for all $x\in\mathbb{R}^n$. The nonasymptotic convergence of the SPP with decreasing stepsize $\mu_k=\frac{\mu_0}{k^\gamma}$, with $\gamma\in(1/2,1]$, has been analyzed in the quadratic mean and the following convergence rate has been derived in (Toulis et al., 2016):

$$\mathbb{E}[\|x^k-x^*\|^2]\leq C\left(\frac{1}{1+\lambda\mu_0\alpha}\right)^{k^{1-\gamma}}+\mathcal{O}\left(\frac{1}{k^\gamma}\right),$$

where C and α are some positive constants. However, the analysis used in (Toulis et al., 2016) cannot be extended to general convex objective functions and complicated constraints, since it is essential in the proofs that each component of the objective function has the form $f(a_S^Tx)$, where $a_S\in\mathbb{R}^n$. In our paper we consider general convex objective functions, which lack the previously discussed structure, with (in)finite number of convex constraints.

Further, in (Bianchi, 2016) a general asymptotic convergence analysis of several variants of SPP scheme within operator theory settings has been provided, under mild convexity assumptions. A particular optimization model instance analyzed in (Bianchi, 2016), related to our paper, is:

$$\min_x\ f(x)\quad\text{s.t.}\ \ x\in\cap_{i=1}^m X_i,$$

for which has been derived the following SPP type algorithm:

$$x^{k+1}=\begin{cases}\arg\min_{z\in\mathbb{R}^n}\left[f(z)+\frac{1}{2\mu_k}\|z-x^k\|^2\right] & \text{if } S_k=0\\ \Pi_{X_{S_k}}(x^k) & \text{otherwise,}\end{cases}$$

where S_k is randomly chosen in $\Omega=\{0,1,\cdots,m\}$ according to a probability distribution $\mathbb{P}$. Although this scheme is very similar to the SPP algorithm, only the almost sure asymptotic convergence has been provided in (Bianchi, 2016). Convergence results of order $\mathcal{O}\left(\frac{1}{k}\right)$ in the strongly convex case, as well as almost sure convergence results under weaker assumptions, are also provided in (Rosasco et al., 2017) for the stochastic proximal gradient algorithm on convex composite optimization problems. In (Combettes and Pesquet, 2016) the asymptotic behavior of a stochastic forward-backward splitting algorithm for finding a zero of the sum of a maximally monotone set-valued operator and a co-coercive operator in Hilbert spaces is investigated. Weak and strong almost sure convergence properties of the iterates are established under mild conditions on the underlying stochastic processes.

A particular case of the stochastic optimization problem (1) is the discrete stochastic model, where the random variable S is discrete and thus, usually the objective function is given as a finite sum of functional components. There exists a large amount of work in the literature on deterministic and randomized algorithms for the finite sum optimization problem. Linear convergence of SGD for solving convex feasibility problems is proven recently in (Necoara, 2017). Convergence analysis of SGD for minimizing an objective function subject to a finite number of convex constraints is provided e.g. in (Necoara, 2017; Nedic, 2011). Linear convergence results on a restarted variant of SGD for finite-sum problems is given in (Yang and Lin, 2016). On the deterministic side, the cyclic incremental gradient methods were extensively analyzed e.g. in (Bertsekas, 2011). Recently, highly efficient algorithms with improved convergence estimates (compared to SGD) for finite sums have been developed using aggregated (averaged) or variance reduction techniques. The first category is based on the common idea of updating the current iterate along the aggregated (averaged) gradient step: e.g. incremental aggregated gradient (IAG) (Vanli et al., 2017), stochastic averaged gradient (SAG) (Roux et al., 2012) and its generalization SAGA (Defazio et al., 2014). Regarding the second category, there are simpler schemes, but memory intensive, such as stochastic variance reduced gradient (SVRG) method introduced in (Johnson and Zhang, 2013). It has been proved that all these schemes can achieve linear convergence under strong convexity and gradient Lipschitz assumptions on the finite sum objective function. Similar optimal performances on finite sum minimization, as for the previous two classes of algorithms, are obtained also for the stochastic dual coordinate ascent (SDCA) method, which has been analyzed in (Shalev-Shwartz and Zhang, 2013).

Other stochastic proximal (gradient) schemes together with their theoretical guarantees are studied in several recent papers as we further exemplify. In (Atchade et al., 2014) a perturbed proximal gradient method is considered for solving composite optimization

problems, where the gradient is intractable and approximated by Monte Carlo methods. Conditions on the stepsize and the Monte Carlo batch size are derived under which the convergence is guaranteed. Two classes of stochastic approximation strategies (stochastic iterative Tikhonov regularization and the stochastic iterative proximal point) are analyzed in (Koshal et al., 2013) for monotone stochastic variational inequalities and almost sure convergence results are presented. A new stochastic optimization method is analyzed in (Yurtsever et al., 2016) for the minimization of the sum of three convex functions, one of which has Lipschitz continuous gradient and satisfies a restricted strong convexity condition. In (Xu, 2011) a finite sample analysis for the averaged SGD is provided, which shows that it usually takes a huge number of samples for averaged SGD to reach its asymptotic region, for improperly chosen learning rate (stepsize). Moreover, simple strategies to properly set the learning rate are derived in the same paper so that it takes a reasonable amount of data for averaged SGD to reach its asymptotic region. In (Niu et al., 2011) it is shown through a novel theoretical analysis that SGD can be implemented in a parallel fashion without any locking. Moreover, for sparse optimization problems (meaning that the most gradient updates only modify small parts of the decision variable) the developed scheme achieves a nearly optimal rate of convergence. A regularized stochastic version of the BFGS method is proposed in (Mokhtari and Ribeiro, 2014) to solve convex optimization problems. Convergence analysis shows that lower and upper bounds on the Hessian eigenvalues of the sample functions are sufficient to guarantee convergence of order $\mathcal{O}\left(\frac{1}{k}\right)$. A comprehensive survey on modern optimization algorithms for machine learning problems is given recently in (Bottou et al., 2016). Based on experience, theoretical results are presented on a straightforward, yet versatile SGD algorithm, its practical behavior is discussed, and opportunities are highlighted for designing new algorithms with improved performance.

1.2 Contributions

In this paper we consider both randomness sources (i.e. objective function and constraints) and thus our problem of interest involves stochastic objective functions subject to (in)finite intersection of constraints. Given the clear superior features of SPP algorithm over the classical SGD scheme, we consider the SPP scheme for solving our problem of interest. The main contributions of this paper are:

(i) More general stochastic optimization model and a new stochastic proximal point algorithm: While most of the existing papers from the stochastic optimization literature consider convex models without constraints or simple (easy projection onto) constraints, in this paper we consider stochastic convex optimization problems subject to (in)finite intersection of constraints satisfying a linear regularity type condition. It turns out that many practical applications, including those from machine learning, fits into this framework: e.g. classification, regression, finite sum minimization, portfolio optimization, convex feasibility, optimal control problems. For this general stochastic optimization model we introduce a new stochastic proximal point (SPP) algorithm. It is worth to mention that although the analysis of an SPP method for stochastic models with complicated constraints is nontrivial and does not follow from the analysis corresponding to the unconstrained setting, our framework allows us to deal with even an infinite number of constraints. To the best of our knowledge, our SPP method is the first stochastic proximal point algorithm that can tackle optimization problems with complicated constraints.

(ii) New nonasymptotic convergence results for the SPP method: For the newly introduced SPP scheme we prove new nonasymptotic convergence results. In particular, for convex and Lipschitz continuous objective functions, we prove nonasymptotic estimates for the rate of convergence of the SPP scheme in terms of the expected value function gap and feasibility violation of order $\mathcal{O}\left(\frac{1}{k^{1/2}}\right)$, where k is the iteration counter. We also derive better nonasymptotic bounds for the rate of convergence of SPP scheme with decreasing stepsize $\mu_k = \frac{\mu_0}{k^\gamma}$, with $\gamma \in (0,1]$, for smooth strongly convex objective functions. For this case the convergence rates are given in terms of expected quadratic distance from the iterates to the optimal solution and are of order:

$$\mathbb{E}[\|x^k - x^*\|^2] \leq C\left(\mathbb{E}\left[\frac{1}{1+\bar{\alpha}_S\mu_0}\right]\right)^{k^{1-\gamma}} + \mathcal{O}\left(\frac{1}{k^\gamma}\right),$$

where C and $\bar{\alpha}_S$ are appropriate nonnegative constants. Note that the derived rates of convergence do not contain any exponential term in μ_0, as it is the case for the SGD scheme, which makes SPP more robust than SGD even in the constrained case. This can be also observed in numerical simulations, see Section 7 below.

(iii) Restarted variant of SPP algorithm and the corresponding convergence analysis: Since the best complexity of our basic SPP scheme can be attained only under some natural restrictions on the initial stepsize μ_0, we also introduce a restarting stochastic proximal point algorithm that overcomes these difficulties. The main advantage of this restarted variant of SPP algorithm is that it is parameter-free and thus it is easily implementable in practice. Under strong convexity and smoothness assumptions on the objective function, for $\gamma > 0$ and epoch counter t, the restarting SPP scheme with the constant stepsize (per epoch) $\frac{1}{t^\gamma}$ provides a nonasymptotic complexity of order $\mathcal{O}\left(\frac{1}{\epsilon^{1+\frac{1}{\gamma}}}\right)$.

Paper outline. The paper is organized as follows. In Section 2 the problem of interest is formulated and analyzed. Further in Section 3, a new stochastic proximal point algorithm is introduced and its relations with the previous work are highlighted. We provide in Section 4 the first main result of this paper regarding the nonasymptotic convergence of SPP in the convex case. Further, stronger convergence results are presented in Section 5 for smooth strongly convex objective functions. In order to improve the convergence of the simple SPP scheme, in Section 6 we introduce a restarted variant of SPP algorithm. In Section 7 we provide some preliminary numerical simulations to highlight the empirical performance of our schemes. Some long proofs are moved in the Appendix.

Notations. We consider the space $\mathbb{R}^n$ composed by column vectors. For $x, y \in \mathbb{R}^n$ denote the scalar product $\langle x, y\rangle = x^T y$ and Euclidean norm by $\|x\| = \sqrt{x^T x}$. The projection operator onto the nonempty closed convex set X is denoted by $\Pi_X(\cdot)$ and the distance from a given x to the set X is denoted by $\text{dist}_X(x) = \min_{z\in X}\|x - z\|$. Given any convex set X, the function $\text{dist}_X(\cdot)$ is convex and the squared distance function $\text{dist}_X^2(\cdot)$ has Lipschitz gradient with constant 1. For some function f, we denote by $\partial f(x)$ the subdifferential set

at x. We also use the following definition of the indicator function of a set X:

$$\mathbb{I}_X(x) = \begin{cases} 0, & \text{if } x \in X \\ \infty, & \text{otherwise.} \end{cases}$$

Finally, we define the function $\varphi_\alpha : (0, \infty) \to \mathbb{R}$ as:

$$\varphi_\alpha(x) = \begin{cases} (x^\alpha - 1)/\alpha, & \text{if } \alpha \neq 0 \\ \log(x), & \text{if } \alpha = 0. \end{cases}$$

2. Problem formulation

In many machine learning applications randomness usually enters the problem through the cost function and/or the constraint set. Minimization of problems having complicating constraints can be very challenging. This is usually alleviated by approximating the feasible set by an (in)finite intersection of simple sets (Necoara, 2017; Necoara et al., 2017; Nedic, 2011). Therefore, in this paper we tackle the following stochastic convex constrained optimization problem:

$$\begin{aligned} F^* = \min_{x \in \mathbb{R}^n} F(x) \quad (:= \mathbb{E}[f(x;S)]) \\ \text{s.t.} \quad x \in X \quad (:= \cap_{S \in \Omega} X_S), \end{aligned} \tag{4}$$

where $f(\cdot; S) : \mathbb{R}^n \to \mathbb{R}$ are convex functions with full domain $\text{dom} f = \mathbb{R}^n$, X_S are nonempty closed convex sets, and S is a random variable with its associated probability space $(\Omega, \mathbb{P})$. Notice that this formulation allows us to include (in)finite number of constraints. We denote the set of optimal solutions with X^* and x^* any optimal point for (4). For the optimization problem (4) we make the following assumptions.

Assumption 1 *For any $S \in \Omega$, the function $f(\cdot; S)$ is proper, closed, convex and Lipschitz continuous, that is there exists $L_{f,S} > 0$ such that*

$$|f(x;S) - f(y;S)| \leq L_{f,S}\|x - y\| \qquad \forall x, y \in \mathbb{R}^n.$$

Notice that Assumption 1 implies that any subgradient $g_f(x;S) \in \partial f(x;S)$ is bounded, that is $\|g_f(x;S)\| \leq L_{f,S}$ for all $x \in \mathbb{R}^n$ and $S \in \Omega$. For the sets we assume:

Assumption 2 *Given $S \in \Omega$, the following two properties hold:*
(i) X_S are simple convex sets (i.e. projections onto these sets are easy).
(ii) There exists $\zeta > 0$ such that the feasible set X satisfies linear regularity:

$$dist_X^2(x) \ \leq \zeta \, \mathbb{E}[dist_{X_S}^2(x)] \qquad \forall x \in \mathbb{R}^n.$$

Assumption 2 (ii) is known in the literature as the *linear regularity property* and it is essential for proving linear convergence for (alternating) projection algorithms, see (Necoara, 2017; Necoara et al., 2017; Nedic, 2011). For example, when X_S are hyperplanes, halfspaces or when X has nonempty interior, then the linear regularity property holds. In particular, if the set X contains a ball of radius $\bar{r}$ and X is contained in a ball of radius $\bar{R}$, then the

ratio $\bar{R}/\bar{r}$ can be taken as the linear regularity constant ζ (Necoara et al., 2017). The linear regularity property is related to the relaxation of strong convexity, the so-called quadratic functional growth condition for an objective function, for smooth convex optimization introduced in (Necoara et al., 2017). In (Necoara et al., 2017) it has been proved that several first order methods converge linearly under functional growth condition and smoothness of the objective function.
Notice that this general optimization model (4) covers a long range of applications from various fields, such as optimization, machine learning, statistics, control, which we discuss in more details below.

2.1 Convex feasibility problem

Let us consider the following objective function and constraints (Necoara, 2017):

$$f(x;S) := \frac{\lambda}{2}\|x\|^2 \quad \forall S \in \Omega \quad \text{and} \quad X = \cap_{S \in \Omega} X_S,$$

where $\lambda > 0$. Then, we obtain the least norm convex feasibility problem:

$$\min_{x \in \mathbb{R}^n} \frac{\lambda}{2}\|x\|^2 \quad \text{s.t.} \ \ x \in \cap_{S \in \Omega} X_S.$$

We can also consider another reformulation of the least norm convex feasibility problem:

$$f(x;S) := \frac{\lambda_S}{2}\|x\|^2 + \mathbb{I}_{X_S}(x) \quad \forall S \in \Omega,$$

where $\lambda_S \geq 0$ and $\mathbb{E}[\lambda_S] = \lambda$. Then, this leads to the stochastic optimization model:

$$\min_{x \in \mathbb{R}^n} \ \mathbb{E}\left[\frac{\lambda_S}{2}\|x\|^2 + \mathbb{I}_{X_S}(x)\right].$$

Finding a point in the intersection of a collection of closed convex sets represents a modeling paradigm for solving important applications such as data compression, neural networks and adaptive filtering, see (Censor et al., 2012) for a complete list.

2.2 Regression problem

Let us consider the matrix $A \in \mathbb{R}^{m \times n}$. For any $S \in \Omega \subseteq \mathbb{R}$, let us define:

$$f(x;S) := \ell(A_S^T x),$$

where ℓ is some loss function and $A_S \in \mathbb{R}^n$. This results in the following constrained optimization model:

$$\min_{x \in \mathbb{R}^n} \quad \mathbb{E}[\ell(A_S^T x)] \quad \text{s.t.} \ \ x \in \cap_{S \in \Omega} X_S.$$

Many learning problems can be modeled into this form, see e.g. (Toulis et al., 2016; Shalev-Shwartz and Zhang, 2013). This type of optimization model has been also considered in (Bianchi, 2016; Rosasco et al., 2014).

2.3 Finite sum problem

Let $\Omega = \{1, \cdots, m\}$ and $\mathbb{P}$ be the uniform discrete probability distribution on Ω. Further, we consider convex functions $f(x; i) = \ell_i(x)$. Then, the following constrained finite sum problem is recovered:

$$\min_{x \in \mathbb{R}^n} \quad \frac{1}{m} \sum_{i=1}^{m} \ell_i(x) \quad \text{s.t. } x \in \cap_{i=1}^{m} X_i.$$

This constrained optimization model appears often in statistics and machine learning applications, where the functions $\ell_i(\cdot)$ typically represent loss functions associated to a given estimator and the feasible set comes from physical constraints, see e.g. (Defazio et al., 2014; Roux et al., 2012; Vanli et al., 2017; Yurtsever et al., 2016). It is also a particular problem of a more general optimization model considered in (Bianchi, 2016).

2.4 Multiple kernel learning problem

In many classification problems we want to learn a convex combination of kernels $\kappa(x, x') = \sum_{j=1}^{M} \beta_j \kappa_j(x, x')$ (Bach et al., 2004). This approach is useful in complex classification problems, where we use polynomial kernels of different degrees or kernels on different domains. The goal is to learn the weights β_j and they are usually found through SVM optimization:

$$\begin{aligned}
&\min_{(w,\beta,\xi,b)} \ \frac{1}{2} \left(\sum_{j=1}^{M} \beta_j \|w_j\| \right)^2 + C \sum_{i=1}^{N} \xi_i \\
&w = (w_1, \cdots, w_M), \ w_j \in \mathbb{R}^{n_j}, \ \beta = (\beta_1, \cdots, \beta_M), \ \xi = (\xi_1, \cdots, \xi_N) \\
&y_i \left(\sum_{j=1}^{M} \beta_j w_j^T x_{ij} + b \right) \geq 1 - \xi_i \ \forall i = 1 : N, \ \xi \geq 0, \ \beta \geq 0, \ \sum_{j=1}^{M} \beta_j = 1.
\end{aligned}$$

Note that this formulation is equivalent to linear SVM for $M = 1$. We usually obtain a sparse solution in β, where each component β_j corresponds to one kernel κ_j. The dual of this optimization problem takes the form:

$$\begin{aligned}
&\min_{(\gamma,\alpha)} \ \frac{1}{2}\gamma^2 - \sum_{i=1}^{N} \alpha_i \\
&0 \leq \alpha \leq C, \ \sum_{i=1}^{N} \alpha_i y_i = 0, \ \sum_{p=1}^{N} \sum_{q=1}^{N} \alpha_p \alpha_q y_p y_q \kappa_j(x_p, x_q) \leq \gamma^2 \ \ \forall j = 1 : M.
\end{aligned}$$

This convex Quadratic Optimization problem with Quadratic Constraints can be easily reformulated as a Linear Program with infinite number of simple constraints by introducing the notation $Q_j(\alpha) = \sum_{p=1}^{N} \sum_{q=1}^{N} \alpha_p \alpha_q y_p y_q \kappa_j(x_p, x_q)$ (Sonnenburg et al., 2006):

$$\begin{aligned}
&\max_{(\theta,\beta)} \ \theta \\
&\theta \in \mathbb{R}, \ \ \beta \geq 0, \ \ \sum_{j=1}^{M} \beta_j = 1, \ \ \sum_{j=1}^{M} \beta_j \left(\frac{1}{2} Q_j(\alpha) - \sum_{i=1}^{N} \alpha_i \right) \geq \theta \quad \forall \alpha \in \Omega(y),
\end{aligned}$$

where we use the notation

$$\Omega(y) = \left\{ \alpha : \ 0 \leq \alpha \leq C, \ \sum_{i=1}^{N} \alpha_i y_i = 0 \right\}.$$

There are many methods for solving Linear Programs with infinite number of constraints, in particular algorithms related to boosting (Sonnenburg et al., 2006). Note that in this Linear Program formulation the sets X_S are simple hyperplanes.

2.5 Optimal control problem

In this section we briefly present the H_2 optimal control problem for linear systems (see (Karimi and Kammer, 2017) for a detailed exposition). In this application one aims at finding a stabilizing controller K for a linear system which minimize an H_2 performance indicator. This problem can be formulated as:

$$\begin{aligned}
&\min_{K(\omega),\Gamma(\omega)} \int_{-\frac{\pi}{T}}^{\frac{\pi}{T}} \text{trace}[\Gamma(\omega)] d\omega \\
&\text{s.t. : } W(\omega)[(I_n + G(\omega)K(\omega))^*(I_n + G(\omega)K(\omega))]^{-1} W^*(\omega) \preceq \Gamma(\omega) \quad \forall \omega \in \Omega,
\end{aligned}$$

where the frequencies ω are taken in the interval $\Omega = \left[-\frac{\pi}{T}, \frac{\pi}{T}\right]$, $G(\omega), W(\omega)$ are the parameters associated with the linear dynamical system under consideration, $\Gamma(\omega)$ is a positive semidefinite matrix and $K(\omega)$ is the controller that needs to be identified. Note that the previous H_2 optimal control problem requires that the constraints, expressed through matrix inequalities, to hold for all frequencies ω in the interval Ω. Moreover, the objective function can be expressed as an expectation over the same interval Ω. In control theory, $\Gamma(\omega)$ and $K(\omega)$ are taken as polynomial matrices in the frequencies ω. Moreover, the previous matrix inequalities are usually convexified using Schur complement and linearization techniques and then the interval Ω is discretized to get a finite number of constraints (linear matrix inequalities) (Karimi and Kammer, 2017).

3. Stochastic Proximal Point algorithm

In this section we propose solving the optimization problem (4) through stochastic proximal point type algorithms. It has been proven in (Necoara et al., 2017) that the optimization problem (4) can be equivalently reformulated under Assumption 2 into the following stochastic optimization problem:

$$\min_{x \in \mathbb{R}^n} \ \mathbb{E}\left[f(x; S) + \mathbb{I}_{X_S}(x) \right]. \tag{5}$$

Since each component of the stochastic objective is nonsmooth, a first possible approach is to apply stochastic subgradient methods (Duchi and Singer, 2009; Moulines and Bach, 2011), which would yield simple algorithms, but having usually a relatively slow sublinear convergence rate. Therefore, for more robustness, one can deal with the nonsmoothness through the Moreau smoothing framework. However, there are multiple potential approaches in

this direction. For a given smoothing parameter $\mu > 0$, we can smooth each functional component and the associated indicator function together to obtain the following smooth approximation for the nonsmooth convex function $f(\cdot;S) + \mathbb{I}_{X_S}$:

$$\bar{f}_\mu(x;S) := \min_{z\in\mathbb{R}^n} \; f(z;S) + \mathbb{I}_{X_S}(z) + \frac{1}{2\mu}\|z-x\|^2.$$

Let us denote the corresponding prox operator by $\bar{z}_\mu(x;S) = \arg\min_{z\in\mathbb{R}^n} f(z;S) + \mathbb{I}_{X_S}(z) + \frac{1}{2\mu}\|z-x\|^2$. It is known that any Moreau approximation $\bar{f}_\mu(\cdot;S)$ is differentiable having the gradient $\nabla \bar{f}_\mu(x;S) = \frac{1}{\mu}(x - \bar{z}_\mu(x;S))$ (Rockafellar and Wets, 1998). Moreover, the gradient is Lipschitz continuous with constants bounded by $\frac{1}{\mu}$. Then, instead of solving the nonsmooth problem (5) we can consider solving the smooth approximation:

$$\min_{x\in\mathbb{R}^n} \; \bar{F}_\mu(x) \quad \left(:= \mathbb{E}[\bar{f}_\mu(x;S)]\right).$$

Notice that we can easily apply the classical SGD strategy to the newly created smooth objective function, which results in the following iteration:

$$\begin{aligned} x^{k+1} &= x^k - \mu_k \nabla \bar{f}_{\mu_k}(x^k;S_k) = \bar{z}_{\mu_k}(x^k;S_k) \\ &= \arg\min_{z\in\mathbb{R}^n} \; f(z;S_k) + \mathbb{I}_{X_{S_k}}(z) + \frac{1}{2\mu_k}\|z-x^k\|^2. \end{aligned}$$

However, the nonasymptotic analysis technique considered in our paper encounters difficulties with this variant of the algorithm. The main difficulty consists in proving the bound $\|\nabla \bar{f}_\mu(x;S)\| \leq \|g_{f(\cdot;S)+\mathbb{I}_{X_S}}(x)\|$ for all $x \in \mathbb{R}^n$, where $g_{f(\cdot;S)+\mathbb{I}_{X_S}}(x) \in \partial(f(\cdot;S) + \mathbb{I}_{X_S})(x)$. We believe that such a bound is essential in our convergence analysis and we leave for future work the analysis of this iterative scheme. Therefore, we considered a second approach based on a smooth Moreau approximation only for the functional component $f(\cdot;S)$ and keeping the indicator function $\mathbb{I}_{X_S}$ in its original form, that is:

$$f_\mu(x;S) := \min_{z\in\mathbb{R}^n} \; f(z;S) + \frac{1}{2\mu}\|z-x\|^2$$

for some smoothing parameter $\mu > 0$. Then, instead of solving nonsmooth problem (5), we solve the following composite approximation:

$$\min_{x\in\mathbb{R}^n} \; F_\mu(x) \quad (:= \mathbb{E}[f_\mu(x;S) + \mathbb{I}_{X_S}(x)]). \tag{6}$$

Let us denote the corresponding prox operator by:

$$z_\mu(x;S) = \arg\min_{z\in\mathbb{R}^n} f(z;S) + \frac{1}{2\mu}\|z-x\|^2.$$

Further, on the stochastic composite approximation (6) we can apply the stochastic projected gradient method, which leads to a stochastic proximal point like scheme for solving the original problem (4):

Algorithm SPP $(x_0, \{\mu_k\}_{k\geq 0})$

For $k \geq 1$ compute:
1. Choose randomly $S_k \in \Omega$ w.r.t. probability distribution $\mathbb{P}$
2. Update: $y^k = z_{\mu_k}(x^k;S_k)$ and $x^{k+1} = \Pi_{X_{S_k}}(y^k)$

where $x^0 \in \mathbb{R}^n$ is some initial starting point and $\{\mu_k\}_{k\geq 0}$ is a nonincreasing positive sequence of stepsizes. We assume that the algorithm SPP returns either the last point x^k or the average point $\hat{x}^k = \frac{1}{\sum_{i=0}^{k-1}\mu_i}\sum_{i=0}^{k-1}\mu_i x^i$ when it is called as a subroutine. Since the update rule of the positive smoothing (stepsize) sequence $\{\mu_k\}_{k\geq 0}$ strongly contributes to the convergence of the scheme, we discuss in the following sections the most advantageous choices. We first prove the following useful auxiliary result:

Lemma 3 *Let $\mu > 0$, $S \in \Omega$. Then, for any $g_f(x;S) \in \partial f(x;S)$, the following holds:*

$$\|\nabla f_\mu(x;S)\| \leq \|g_f(x;S)\| \qquad \forall x \in \mathbb{R}^n.$$

Proof The optimality condition of problem $\min_{z\in\mathbb{R}^n} f(z;S) + \frac{1}{2\mu}\|x-z\|^2$ is given by:

$$\frac{1}{\mu}(x - z_\mu(x;S)) \in \partial f(z_\mu(x;S);S).$$

The above inclusion easily implies that there is $g_f(z_\mu(x;S);S) \in \partial f(z_\mu(x;S);S)$ such that:

$$\begin{aligned} \frac{1}{\mu}\|z_\mu(x;S) - x\|^2 &= \langle g_f(z_\mu(x;S);S), x - z_\mu(x;S)\rangle \\ &= \langle g_f(x;S), x - z_\mu(x;S)\rangle + \langle g_f(z_\mu(x;S);S) - g_f(x;S), x - z_\mu(x;S)\rangle \\ &\leq \langle g_f(x;S), x - z_\mu(x;S)\rangle, \end{aligned}$$

where in the last inequality we used the convexity of f. Lastly, by applying the Cauchy-Schwarz inequality in the right hand side we get the above statement. ■

The following two well-known inequalities, which can be found in (Bullen, 2003), will be also useful in the sequel:

(*i*) **[Bernoulli]** Let $t \in [0,1]$ and $x \in [-1,\infty)$, then the following holds:

$$(1+x)^t \leq 1 + tx. \tag{7}$$

(*ii*) **[Minkowski]** Let x and y be two random variables. Then, for any $1 \leq p < \infty$, the following inequality holds:

$$(\mathbb{E}[|x+y|^p])^{1/p} \leq (\mathbb{E}[|x|^p])^{1/p} + (\mathbb{E}[|y|^p])^{1/p}. \tag{8}$$

4. Nonasymptotic complexity of SPP: convex objective function

In this section we analyze, under Assumptions 1 and 2, the iteration complexity of SPP scheme with nonincreasing stepsize rule to approximately solve the optimization problem (4). In order to prove this nonasymptotic result, we define $\hat{\mu}_{1,k}=\sum\limits_{i=0}^{k-1}\mu_i$, $\hat{\mu}_{2,k}=\sum\limits_{i=0}^{k-1}\mu_i^2$ and the averaged sequences $\hat{x}^k=\frac{1}{\hat{\mu}_{1,k}}\sum_{i=0}^{k-1}\mu_i x^i$ and $\hat{y}^k=\frac{1}{\hat{\mu}_{1,k}}\sum_{i=0}^{k-1}\mu_i y^i$. Moreover, denote by $\mathcal{F}_k$ the history of random choices $\{S_k\}_{k\geq 0}$, i.e. $\mathcal{F}_k=\{S_0,\cdots,S_k\}$.

Lemma 4 *Let Assumptions 1 and 2 hold and the sequences $\{x^k,y^k\}_{k\geq 0}$ be generated by SPP scheme with positive stepsize $\{\mu_k\}_{k\geq 0}$. Then the following relation holds:*

$$\mathbb{E}\left[dist^2_{X_{S_k}}(\hat{y}^k)\right]\geq\frac{1}{\zeta}\mathbb{E}\left[dist^2_X(\hat{x}^k)\right]-\frac{\hat{\mu}_{2,k}}{\hat{\mu}_{1,k}}\sqrt{\mathbb{E}[dist^2_X(\hat{x}^k)]}\sqrt{\mathbb{E}[L^2_{f,S}]}.$$

Proof See Appendix for the proof. ■

Now, we are ready to derive the convergence rate of SPP in the average sequence $\hat{x}^k$:

Theorem 5 *Under Assumptions 1 and 2, let the sequence $\{x^k\}_{k\geq 0}$ be generated by the algorithm SPP with nonincreasing positive stepsize $\{\mu_k\}_{k\geq 0}$. Define $\mathcal{R}_\mu=\mu_0\zeta(\|x^0-x^*\|^2+\mathbb{E}[L^2_{f,S}]\hat{\mu}_{2,k})$, then the following estimates for suboptimality and feasibility violation hold:*

$$\begin{aligned}-\zeta\mathbb{E}[L^2_{f,S}]\left(\frac{\hat{\mu}_{2,k}}{\hat{\mu}_{1,k}}+2\mu_0\right)-\sqrt{\mathbb{E}[L^2_{f,S}]\,\frac{\mathcal{R}_\mu}{\hat{\mu}_{1,k}}}\leq\mathbb{E}[F(\hat{x}^k)]-F^*\leq\frac{\mathcal{R}_\mu}{2\mu_0\zeta\hat{\mu}_{1,k}}\\ \mathbb{E}[dist^2_X(\hat{x}^k)]\leq 2\zeta^2\mathbb{E}[L^2_{f,S}]\left(\frac{\hat{\mu}_{2,k}}{\hat{\mu}_{1,k}}+2\mu_0\right)^2+\frac{2\mathcal{R}_\mu}{\hat{\mu}_{1,k}}.\end{aligned}\tag{9}$$

Proof See Appendix for the proof. ■

Note that the right suboptimality bound (9), obtained for the SPP algorithm, is similar with the one given for the standard subgradient method (Nesterov, 2004). Below we provide the convergence estimates for the algorithm SPP with constant stepsize for a desired accuracy $\epsilon>0$. For simplicity, assume that $\|x^0-x^*\|\geq 1$ and $\mathbb{E}[L^2_{f,S}]\geq 2$.

Corollary 6 *Under the assumptions of Theorem 5, let $\{x^k\}_{k\geq 0}$ be the sequence generated by algorithm SPP with constant stepsize $\mu_k=\mu>0$. Also let $\epsilon>0$ be the desired accuracy, K be an integer satisfying:*

$$K\geq\frac{\mathbb{E}[L^2_{f,S}]\|x^0-x^*\|^2}{\epsilon^2}\max\left\{1,(3\zeta+\sqrt{2\zeta})^2\right\},$$

and the stepsize be chosen as:

$$\mu=\frac{\epsilon}{\mathbb{E}[L^2_{f,S}](3\zeta+\sqrt{2\zeta})}.$$

Then, after K iterations, the average point $\hat{x}^K=\frac{1}{K}\sum\limits_{i=0}^{K-1}x^i$ satisfies:

$$\left|\mathbb{E}[F(\hat{x}^K)]-F^*\right|\leq\epsilon\quad\text{and}\quad\sqrt{\mathbb{E}[dist^2_X(\hat{x}^K)]}\leq\epsilon.$$

Proof We consider $k=K$ in Theorem 5 and, by taking into account that $\mu_k=\mu$ for all $k\geq 0$, we aim to obtain the lowest value of the right hand side of (9) by minimizing over $\mu>0$. Thus, by denoting that $r_0=\|x^0-x^*\|$, we obtain for the optimal smoothing parameter:

$$\mu=\sqrt{\frac{r_0^2}{K\mathbb{E}[L^2_{f,S}]}}$$

the optimal rate

$$\mathbb{E}[F(\hat{x}^K)]-F^*\leq\sqrt{\frac{\mathbb{E}[L^2_{f,S}]r_0^2}{K}}.\tag{10}$$

Also using the optimal parameter $\tilde{\mu}$ into the other relations of Theorem 5 result in:

$$\mathbb{E}[\text{dist}^2_X(\hat{x}^K)]\leq\frac{r_0^2}{K}\left(18\zeta^2+4\zeta\right)\tag{11}$$

and

$$\mathbb{E}[F(\hat{x}^K)]-F^*\geq-(3\zeta+\sqrt{2\zeta})\sqrt{\frac{\mathbb{E}[L^2_{f,S}]r_0^2}{K}}.\tag{12}$$

From the upper and lower suboptimality bounds (10), (12) and feasibility bound (11), we deduce the following bound:

$$K\geq\frac{\mathbb{E}[L^2_{f,S}]r_0^2}{\epsilon^2}\max\left\{1,(3\zeta+\sqrt{2\zeta})^2\right\}$$

which confirms our result. ■

In conclusion, Corollary 6 states that for a desired accuracy ϵ, if we choose a constant stepsize $\mu=\mathcal{O}(\epsilon)$ and perform a number of SPP iterations $\mathcal{O}\left(\frac{1}{\epsilon^2}\right)$ we obtain an ϵ-optimal solution for our original stochastic constrained convex problem (4). Note that for convex problems with objective function having bounded subgradients the previous convergence estimates derived for the SPP algorithm are similar to those corresponding to the classical deterministic proximal point method (Guler, 1991) and subgradient method (Nesterov, 2004).

5. Nonasymptotic complexity of SPP: strongly convex objective function

In this section we analyze the convergence behavior of the SPP scheme under smoothness and strong convexity assumptions on the objective function of constrained problem (4). Therefore, in this section the Assumption 1 is replaced by the following assumptions:

Assumption 7 *Each function $f(\cdot;S)$ is differentiable and $\sigma_{f,S}$-strongly convex, that is there exists strong convexity constant $\sigma_{f,S}\geq 0$ such that:*

$$f(x;S)\geq f(y;S)+\langle\nabla f(y;S),x-y\rangle+\frac{\sigma_{f,S}}{2}\|x-y\|^2 \qquad \forall x,y\in\mathbb{R}^n.$$

Moreover, the strong convexity constants $\sigma_{f,S}$ satisfy $\sigma_F=\mathbb{E}[\sigma_{f,S}]>0$.

Notice that if for some function $f(\cdot;S)$ the corresponding constant $\sigma_{f,S}=0$, then $f(\cdot;S)$ is only convex. However, relation $\mathbb{E}[\sigma_{f,S}]=\sigma_F>0$ implies that the whole objective function F of problem (4) is strongly convex with constant $\sigma_F>0$. In the sequel we will analyze the SPP scheme under the following additional smoothness assumption:

Assumption 8 *Each function $f(\cdot;S)$ has Lipschitz gradient, that is there exists Lipschitz constant $L_{f,S}>0$ such that:*

$$\|\nabla f(x;S)-\nabla f(y;S)\|\leq L_{f,S}\|x-y\| \qquad \forall x,y\in\mathbb{R}^n.$$

Note that Assumptions 7 and 8 are standard for the convergence analysis of SPP like schemes, see e.g. (Moulines and Bach, 2011; Ryu and Boyd, 2016). We first present an auxiliary result on the behavior of the proximal mapping $z_\mu(\cdot;S)$.

Lemma 9 *Let $f(\cdot;S)$ satisfy Assumption 7. Further, for any $S\in\Omega$ and $\mu>0$, we define $\theta_S(\mu)=\frac{1}{1+\mu\sigma_{f,S}}$. Then, the following contraction inequality holds for the prox operator:*

$$\|z_\mu(x;S)-z_\mu(y;S)\|\leq\theta_S(\mu)\|x-y\| \qquad \forall x,y\in\mathbb{R}^n.$$

Proof See Appendix for the proof. ■

Notice that if all the functions $f(\cdot;S)$ are just convex, that is they satisfy Assumption 7 with $\sigma_{f,S}=0$, then Lemma 9 highlights the nonexpansiveness property of the proximal operator $z_\mu(\cdot;S)$. We will further keep using the notation $\theta_S(\mu)$ for the contraction factor of the operator $z_\mu(\cdot;S)$. Moreover, in all our proofs below, regarding the results in expectation, we use the standard technique of applying first expectation with respect to S_k conditioned on $\mathcal{F}_{k-1}$ and then apply the expectation over the entire history $\mathcal{F}_{k-1}$ (see the proof of Theorem 5). For simplicity of the exposition and for saving space, we omit these details below.

5.1 Linear convergence to noise dominated region for constant stepsize SPP

Next we analyze the sequence generated by the SPP scheme with constant stepsize $\mu>0$ and provide a nonasymptotic bound on the quadratic mean $\{\mathbb{E}[\|x^k-x^*\|^2]\}_{k\geq 0}$.

Theorem 10 *Under Assumption 7, let the sequence $\{x^k\}_{k\geq 0}$ be generated by the algorithm SPP with constant stepsize $\mu>0$. Further, assume $\sigma_f^{\max}=\sup_{S\in\Omega}\sigma_{f,S}<\infty$. Then, $\mathbb{E}[\theta_S^2(\mu)]\leq\mathbb{E}[\theta_S(\mu)]<1$ and the following linear convergence to some region around the optimal point in the quadratic mean holds:*

$$\mathbb{E}[\|x^k-x^*\|^2]\leq 2\left(\mathbb{E}\left[\theta_S^2(\mu)\right]\right)^k\|x^0-x^*\|^2+\frac{2\mu^2\mathbb{E}[\|\nabla f(x^*;S)\|^2]}{\left(1-\sqrt{\mathbb{E}[\theta_S^2(\mu)]}\right)^2}.$$

Proof First, it can be easily seen that for any $\mu>0$ and $S\in\Omega$ we have $\theta_S^2(\mu)\leq\theta_S(\mu)\leq 1$ and assuming that $\sigma_f^{\max}<\infty$ we obtain:

$$0\leq\mathbb{E}[\theta_S^2(\mu)]\leq\mathbb{E}[\theta_S(\mu)]=\mathbb{E}\left[\frac{1}{1+\mu\sigma_{f,S}}\right]=1-\mathbb{E}\left[\frac{\mu\sigma_{f,S}}{1+\mu\sigma_{f,S}}\right]\leq 1-\frac{\mu\sigma_F}{1+\mu\sigma_f^{\max}}<1.$$

Then, by applying Lemma 9 with $S=S_k, x=x^k$ and $z=x^*$, results in:

$$\left\|z_\mu(x^k;S_k)-z_\mu(x^*;S_k)\right\|\leq\theta_{S_k}(\mu)\|x^k-x^*\|,$$

which, by the triangle inequality, further implies:

$$\left\|z_\mu(x^k;S_k)-x^*\right\|\leq\theta_{S_k}(\mu)\|x^k-x^*\|+\|z_\mu(x^*;S_k)-x^*\|.$$

By using the nonexpansiveness property of the projection operator we get that $\|x^{k+1}-x^*\|\leq\|y^k-x^*\|$, then the last inequality leads to the reccurent relation:

$$\left\|x^{k+1}-x^*\right\|\leq\left\|z_\mu(x^k;S_k)-x^*\right\|\leq\theta_{S_k}(\mu)\|x^k-x^*\|+\|z_\mu(x^*;S_k)-x^*\|. \tag{13}$$

The relation (13), Minkowski inequality and Lemma 3 lead to the following recurrence:

$$\begin{aligned}\sqrt{\mathbb{E}[\|x^{k+1}-x^*\|^2]}&\overset{(13)}{\leq}\sqrt{\mathbb{E}\left[\left(\theta_{S_k}(\mu)\|x^k-x^*\|+\|z_\mu(x^*;S_k)-x^*\|\right)^2\right]}\\&\overset{(8)}{\leq}\sqrt{\mathbb{E}\left[\theta_{S_k}^2(\mu)\|x^k-x^*\|^2\right]}+\sqrt{\mathbb{E}\left[\|z_\mu(x^*;S_k)-x^*\|^2\right]}\\&=\sqrt{\mathbb{E}\left[\theta_S^2(\mu)\right]}\sqrt{\mathbb{E}\left[\|x^k-x^*\|^2\right]}+\mu\sqrt{\mathbb{E}\left[\|\nabla f_\mu(x^*;S)\|^2\right]}\\&\overset{\text{Lemma 3}}{\leq}\sqrt{\mathbb{E}\left[\theta_S^2(\mu)\right]}\sqrt{\mathbb{E}[\|x^k-x^*\|^2]}+\mu\sqrt{\mathbb{E}\left[\|\nabla f(x^*;S)\|^2\right]}.\end{aligned}$$

This yields the following relation valid for all $\mu>0$ and $k\geq 0$:

$$\sqrt{\mathbb{E}[\|x^{k+1}-x^*\|^2]}\leq\sqrt{\mathbb{E}\left[\theta_S^2(\mu)\right]}\sqrt{\mathbb{E}[\|x^k-x^*\|^2]}+\mu\sqrt{\mathbb{E}\left[\|\nabla f(x^*;S)\|^2\right]}, \tag{14}$$

Denote $r_k=\sqrt{\mathbb{E}[\|x^k-x^*\|^2]}$, $\eta=\sqrt{\mathbb{E}\left[\|\nabla f(x^*;S)\|^2\right]}$ and $\theta(\mu)=\sqrt{\mathbb{E}\left[\theta_S^2(\mu)\right]}$. Then, we get:

$$r_{k+1}\leq\theta(\mu)r_k+\mu\eta.$$

Finally, a simple inductive argument leads to:

$$\begin{aligned}r_k&\leq r_0\theta(\mu)^k+\mu\eta\left[1+\theta(\mu)+\cdots+\theta(\mu)^{k-1}\right]\\&=r_0\theta(\mu)^k+\mu\eta\frac{1-\theta(\mu)^k}{1-\theta(\mu)}\\&\leq r_0\theta(\mu)^k+\frac{\mu\eta}{1-\theta(\mu)}.\end{aligned}$$

By squaring and returning to our basic notations, we recover our statement. ■

Theorem 10 proves a linear convergence rate in expectation, without assuming any kind of smoothness on the objective function, for the sequence $\{x^k\}_{k\geq 0}$ generated by SPP with constant stepsize $\mu > 0$ when the iterates are outside of a *noise dominated* neighborhood of the optimal set of radius $\frac{\mu\sqrt{\mathbb{E}[\|\nabla f(x^*;S)\|^2]}}{1-\sqrt{\mathbb{E}[\theta_S^2(\mu)]}}$. It also establishes the boundedness of the sequence $\{x^k\}_{k\geq 0}$ when the stepsize is constant. Notice that in (Ryu and Boyd, 2016) a similar result has been given for an unconstrained optimization model with the difference that the convergence rate was provided for $\mathbb{E}[\|x^k - x^*\|]$. However, our proof is simpler and more elegant, based on the properties of Moreau approximation, despite the fact that we consider the constrained case.

5.2 Nonasymptotic sublinear convergence rate of variable stepsize SPP

In this section we derive sublinear convergence rate of order $\mathcal{O}(1/k^\gamma)$ for the variable stepsize SPP scheme, in a nonasymptotic fashion. We first prove the boundedness of $\{x^k\}_{k\geq 0}$ when the stepsize is nonincreasing, which will be useful for the subsequent convergence results.

Lemma 11 *Under Assumption 7, let the sequence $\{x^k\}_{k\geq 0}$ be generated by the algorithm SPP with nonincreasing positive stepsize $\{\mu_k\}_{k\geq 0}$. Then, the following relation holds:*

$$\mathbb{E}\left[\|x^k - x^*\|\right] \leq \sqrt{\mathbb{E}[\|x^k - x^*\|^2]} \leq \max\left\{\|x^0 - x^*\|, \frac{\mu_0\sqrt{\mathbb{E}\left[\|\nabla f(x^*;S)\|^2\right]}}{1-\sqrt{\mathbb{E}\left[\theta_S^2(\mu_0)\right]}}\right\}.$$

Proof See Appendix for the proof. ■

Furthermore, we need an upper bound on the sequence $\{\mathbb{E}[\|\nabla f(x^k;S)\|]\}_{k\geq 0}$:

Lemma 12 *Under Assumptions 7 and 8, let the sequence $\{x^k\}_{k\geq 0}$ be generated by the algorithm SPP with nonincreasing positive stepsizes $\{\mu_k\}_{k\geq 0}$. Then, the following holds:*

$$\mathbb{E}[\|\nabla f(x^k;S)\|^2] \leq 2\mathbb{E}[\|\nabla f(x^*;S)\|^2] + 2\mathbb{E}[L_{f,S}^2]\mathcal{A}^2,$$

where $\mathcal{A} = \max\left\{\|x^0 - x^*\|, \frac{\mu_0\sqrt{\mathbb{E}[\|\nabla f(x^*;S)\|^2]}}{1-\sqrt{\mathbb{E}\left[\theta_S^2(\mu_0)\right]}}\right\}$.

Proof From the Lipschitz continuity of $\nabla f(\cdot;S)$ we have that $\|\nabla f(x;S) - \nabla f(x^*;S)\| \leq L_{f,S}\|x - x^*\|$ for all $x \in \mathbb{R}^n$, which implies:

$$\|\nabla f(x^k;S)\|^2 \leq (\|\nabla f(x^*;S)\| + L_{f,S}\|x^k - x^*\|)^2 \leq 2\|\nabla f(x^*;S)\|^2 + 2L_{f,S}^2\|x^k - x^*\|^2.$$

By taking expectation in both sides we get:

$$\mathbb{E}[\|\nabla f(x^k;S)\|^2] \leq 2\mathbb{E}[\|\nabla f(x^*;S)\|^2] + 2\mathbb{E}[L_{f,S}^2]\mathbb{E}[\|x^k - x^*\|^2].$$

Lastly, by using Lemma 11 we obtain our statement. ■

Finally, we provide a non-trivial upper bound on the feasibility gap, which automatically leads to a iterative descent in the distance to the feasible set of the sequence $\{x^k\}_{k\geq 0}$, generated by the SPP scheme with nonincreasing stepsizes.

Lemma 13 *Under Assumptions 2, 7 and 8, let the sequence $\{x^k\}_{k\geq 0}$ be generated by SPP scheme with nonincreasing stepsizes $\{\mu_k\}_{k\geq 0}$. Then, the following relation holds:*

$$\sqrt{\mathbb{E}[dist_X^2(x^k)]} \leq \left(1 - \frac{1}{\zeta}\right)^{k/2}\left[dist_X(x^0) + 2\mu_0\zeta\mathcal{B}\right] + 2\mu_{k-\lceil\frac{k}{2}\rceil}\zeta\mathcal{B},$$

where $\mathcal{B} = \sqrt{2\mathbb{E}[\|\nabla f(x^*;S)\|^2]} + \mathcal{A}\sqrt{2\mathbb{E}[L_{f,S}^2]}$.

Proof See Appendix for the proof. ■

Now, we are ready to derive the nonasymptotic convergence rate of the Algorithm SPP with nonincreasing stepsizes. For simplicity, we denote $\eta = \sqrt{\mathbb{E}[\|\nabla f(x^*;S)\|^2]}$ and keep the notations for $\mathcal{A}$ from Lemma 12 and for $\mathcal{B}$ from Lemma 13.

Theorem 14 *Under Assumptions 2, 7 and 8, let the sequence $\{x^k\}_{k\geq 0}$ be generated by the algorithm SPP with the stepsize $\mu_k = \frac{\mu_0}{k^\gamma}$ for all $k \geq 1$, with $\mu_0 > 0$ and $\gamma \in (0,1]$, and denote $\theta_0 = \mathbb{E}\left[\theta_S^2(\mu_0)\right] = \mathbb{E}\left[\frac{1}{(1+\mu_0\sigma_{f,S})^2}\right]$. Then, the following relations hold:*
(i) If $\gamma \in (0,1)$, then we have the following nonasymptotic convergence rates:

$$\mathbb{E}[\|x^k - x^*\|^2] \leq \theta_0^{\varphi_{1-\gamma}(k)} r_0^2 + \mathcal{D}\theta_0^{\varphi_{1-\gamma}(k) - \varphi_{1-\gamma}(\frac{k+1}{2})}\mu_0^2\left[\varphi_{1-2\gamma}\left(\frac{k+1}{2}\right) + 2\right] + \frac{\mathcal{D}\mu_0^2 4^\gamma}{(1-\theta_0)k^\gamma}.$$

(ii) If $\gamma = 1$, then we have the following nonasymptotic convergence rate:

$$\mathbb{E}[\|x^k - x^*\|^2] \leq \begin{cases} \theta_0^{\varphi_0(k)} r_0^2 + \frac{2\mu_0^2}{k\left(\ln\left(\frac{1}{\theta_0}\right)-1\right)} & \text{if} \quad \theta_0 < \frac{1}{e} \\ \theta_0^{\varphi_0(k)} r_0^2 + \frac{2\mu_0^2 \ln k}{k} & \text{if} \quad \theta_0 = \frac{1}{e} \\ \theta_0^{\varphi_0(k)} r_0^2 + \left(\frac{2}{k}\right)^{\ln\left(\frac{1}{\theta_0}\right)} \frac{\mu_0^2}{1-\ln\left(\frac{1}{\theta_0}\right)} & \text{if} \quad \theta_0 > \frac{1}{e}, \end{cases}$$

where $\mathcal{D} = 4\|\nabla F(x^*)\|\left[\frac{dist_X(x^0)+2\mu_0\zeta\mathcal{B}}{\mu_0\ln(\zeta/(\zeta-1))} + 3^\gamma\mathcal{B}\zeta\right] + 2\eta\sqrt{2\eta^2 + 2\mathbb{E}[L_{f,S}^2]\mathcal{A}^2} + 2\eta\mathcal{A}\sqrt{\mathbb{E}[L_{f,S}^2]}$.

Proof See Appendix for the proof. ■

For more clear estimates of the convergence rates obtained in Theorem 14, we provide in the next corollary a summary given in terms of the dominant terms:

Corollary 15 *Under the assumptions of Theorem 14 the following convergence rates hold:*
(i) If $\gamma \in (0,1)$, then we have convergence rate of order:

$$\mathbb{E}[\|x^k - x^*\|^2] \leq \mathcal{O}\left(\frac{1}{k^\gamma}\right)$$

(ii) If $\gamma = 1$, then we have convergence rate of order:

$$\mathbb{E}[\|x^k - x^*\|^2] \leq \begin{cases} \mathcal{O}\left(\frac{1}{k}\right) & \text{if } \theta_0 < \frac{1}{e} \\ \mathcal{O}\left(\frac{\ln k}{k}\right) & \text{if } \theta_0 = \frac{1}{e} \\ \mathcal{O}\left(\frac{1}{k}\right)^{2\ln\left(\frac{1}{\theta_0}\right)} & \text{if } \theta_0 > \frac{1}{e}. \end{cases}$$

Proof First assume that $\gamma \in (0, \frac{1}{2})$. This assumption implies that $1 - 2\gamma > 0$ and that:

$$\varphi_{1-2\gamma}\left(\frac{k}{2}+2\right) = \frac{\left(\frac{k}{2}+2\right)^{1-2\gamma} - 1}{1-2\gamma} \leq \frac{\left(\frac{k}{2}+2\right)^{1-2\gamma}}{1-2\gamma}. \tag{15}$$

On the other hand, by using the inequality $e^{-x} \leq \frac{1}{1+x}$ for all $x \geq 0$, we obtain:

$$\begin{aligned}
&\theta_0^{\varphi_{1-\gamma}(k+1) - \varphi_{1-\gamma}\left(\frac{k+1}{2}\right)} \varphi_{1-2\gamma}\left(\frac{k}{2}+2\right) = e^{\left(\varphi_{1-\gamma}(k+1) - \varphi_{1-\gamma}\left(\frac{k+1}{2}\right)\right)\ln\theta_0} \varphi_{1-2\gamma}\left(\frac{k}{2}+2\right) \\
&\leq \frac{\varphi_{1-2\gamma}\left(\frac{k}{2}+2\right)}{1 + [\varphi_{1-\gamma}(k+1) - \varphi_{1-\gamma}(\frac{k}{2}+1)]\ln\frac{1}{\theta_0}} \overset{(15)}{\leq} \frac{\frac{(k+4)^{1-2\gamma}}{2^{1-2\gamma}(1-2\gamma)}}{\frac{1}{1-\gamma}[(k+1)^{1-\gamma} - (\frac{k}{2}+1)^{1-\gamma}]\ln\frac{1}{\theta_0}} \\
&= \frac{\frac{(k+4)^{1-2\gamma}}{2^{1-2\gamma}(1-2\gamma)}}{\frac{(k+2)^{1-\gamma}}{1-\gamma}[(\frac{2}{3})^{1-\gamma} - (\frac{1}{2})^{1-\gamma}]\ln\frac{1}{\theta_0}} = \frac{1-\gamma}{1-2\gamma}\frac{2^\gamma(k+4)^{-\gamma}}{[(\frac{2}{3})^{1-\gamma} - (\frac{1}{2})^{1-\gamma}]\ln\frac{1}{\theta_0}} \approx \mathcal{O}\left(\frac{1}{k^\gamma}\right).
\end{aligned}$$

Therefore, in this case, the overall rate will be given by:

$$r_{k+1}^2 \leq \theta_0^{\mathcal{O}(k^{1-\gamma})} r_0^2 + \mathcal{O}\left(\frac{1}{k^\gamma}\right) \approx \mathcal{O}\left(\frac{1}{k^\gamma}\right).$$

If $\gamma = \frac{1}{2}$, then the definition of $\varphi_{1-2\gamma}(\frac{k}{2}+2)$ provides that:

$$r_{k+1}^2 \leq \theta_0^{\mathcal{O}(\sqrt{k})} r_0^2 + \theta_0^{\mathcal{O}(\sqrt{k})}\mathcal{O}(\ln k) + \mathcal{O}\left(\frac{1}{\sqrt{k}}\right) \approx \mathcal{O}\left(\frac{1}{\sqrt{k}}\right).$$

When $\gamma \in (\frac{1}{2}, 1)$, it is obvious that $\varphi_{1-2\gamma}\left(\frac{k}{2}+2\right) \leq \frac{1}{2\gamma-1}$ and therefore the order of the convergence rate changes into:

$$r_{k+1}^2 \leq \theta_0^{\mathcal{O}(k^{1-\gamma})}[r_0^2 + \mathcal{O}(1)] + \mathcal{O}\left(\frac{1}{k^\gamma}\right) \approx \mathcal{O}\left(\frac{1}{k^\gamma}\right).$$

Lastly, if $\gamma = 1$, by using $\theta_0^{\ln k + 1} \leq \left(\frac{1}{k}\right)^{\ln\frac{1}{\theta_0}}$ we obtain the second part of our result. ■

Notice that the above results state that our SPP algorithm with variable stepsize $\frac{\mu_0}{k^\gamma}$ converges with $\mathcal{O}\left(\frac{1}{k^\gamma}\right)$ rate. Similar results have been obtained in (Toulis et al., 2016) for a particular objective function of the form $f(a_S^T x)$ without any constraints and for $\gamma \in (1/2, 1]$. Moreover, for $\gamma = 1$ similar convergence rate, but in asymptotic fashion and for unconstrained problems, has been derived in (Ryu and Boyd, 2016). As we have already mentioned in the introduction section, the convergence rate for the SGD scheme contains an exponential term of the form $\frac{e^{C_2\mu_0^2}}{k^{\alpha\mu_0}}$, which for a given iteration counter k grows exponentially in the initial stepsize μ_0, see (Moulines and Bach, 2011). Thus, although the SGD method achieves a rate $O(\frac{1}{k})$ for a variable stepsize $\frac{\mu_0}{k}$, if μ_0 is chosen too large, then it can induce catastrophic effects in the convergence rate. However, one should notice that for our SPP method, Theorem 14 does not contain this kind of exponential term, therefore SPP is more robust than SGD scheme even in the constrained case. This can be also observed in numerical simulations, see Section 7 below. Clearly, Corollary 15 directly implies the following complexity estimates for attaining a suboptimal point x^k satisfying $\mathbb{E}[\|x^k - x^*\|^2] \leq \epsilon$.

Corollary 16 *Under the assumptions of Theorem 14 and $\epsilon > 0$ the following estimates hold. For $\gamma \in (0,1)$, if we perform:*

$$\left\lceil \mathcal{O}\left(\frac{1}{\epsilon^{1/\gamma}}\right) \right\rceil$$

iterations of SPP scheme with variable stepsize, then the sequence $\{x^k\}_{k\geq 0}$ satisfies $\mathbb{E}[\|x^k - x^*\|^2] \leq \epsilon$. Moreover, for $\gamma = 1$ and $\theta_0 < \frac{1}{e}$, if we perform:

$$\left\lceil \mathcal{O}\left(\frac{1}{\epsilon}\right) \right\rceil$$

iterations of SPP scheme with variable stepsize, then we have $\mathbb{E}[\|x^k - x^*\|^2] \leq \epsilon$.

Proof The proof follows immediately from Corollary 15. ■

6. A restarted variant of Stochastic Proximal Point algorithm

From previous section we easily notice that an $\mathcal{O}\left(\frac{1}{\epsilon}\right)$ convergence rate is obtained for the SPP algorithm with variable stepsize $\mu_k = \frac{\mu_0}{k}$ only when the initial stepsize μ_0 is chosen sufficiently large such that $\theta_0 = \mathbb{E}\left[\frac{1}{(1+\mu_0\sigma_{f,S})^2}\right] < \frac{1}{\sqrt{e}}$. However, this condition is not easy to check. Therefore, if μ_0 is not chosen adequately, we can encounter the case $\theta_0 > \frac{1}{\sqrt{e}}$, which leads to a worse convergence rate for the SPP scheme of order $\mathcal{O}\left(\epsilon^{-\frac{1}{2\ln(1/\theta_0)}}\right)$, that is implicitly dependent on the choice of the initial stepsize μ_0. In conclusion, in order to remove this dependence on the initial stepsize of the simple SPP scheme, we develop a restarting variant of it. This variant consists of running the SPP algorithm (as a routine) for multiple times (epochs) and restarting it each time after a certain number of iterations. In each epoch t, the SPP scheme runs for an estimated number of iterations K_t, which may vary over the epochs, depending on the assumptions made on the objective function.

More explicitly, the Restarted Stochastic Proximal Point (RSPP) scheme has the following iteration:

Algorithm RSPP

Let $\mu_0 > 0$ and $x^{0,0} \in \mathbb{R}^n$. For $t \geq 1$ do:

1. Compute stepsize μ_t and number of inner iterations K_t
2. Set $x^{K_t,t}$ the average output of SPP$(x^{K_{t-1},t-1}, \mu_t)$ runned for K_t iterations with constant stepsize μ_t
3. If an outer stopping criterion is satisfied, then **STOP**, otherwise $t := t+1$ and go to step 1.

We analyze below the nonasymptotic convergence rate of the RSPP algorithm under Assumptions 7 and 8.

6.1 Nonasymptotic sublinear convergence of algorithm RSPP

In this section we analyze the convergence rate of the sequence generated by the RSPP scheme, which repeatedly calls the subroutine SPP with a constant stepsize, in multiple epochs. We consider that SPP runs in epoch $t \geq 1$ with the constant stepsize μ_t for K_t iterations. As in previous sections, we first provide a descent lemma for the feasibility gap. For simplicity, we keep the notations of $\mathcal{A}$ from Lemma 12 and $\mathcal{B}$ from Lemma 13.

Lemma 17 *Let Assumptions 2, 7 and 8 hold. Also let the sequence $\{x^{K_t,t}\}_{t\geq 0}$ be generated by RSPP scheme with nonincreasing stepsizes $\{\mu_t\}_{t\geq 0}$ and nondecreasing epoch lengths $\{K_t\}_{t\geq 1}$ such that $K_t \geq 1$ for all $t \geq 1$. Then, the following relation holds:*

$$\sqrt{\mathbb{E}[dist_X^2(x^{K_t,t})]} \leq \left(1-\frac{1}{\zeta}\right)^{\sum_{i=1}^{t}\frac{K_i}{2}} dist_X(x^{0,0}) + 2\left(1-\frac{1}{\zeta}\right)^{\sum_{i=t-\lceil \frac{t}{2}\rceil}^{t}\frac{K_i}{2}} \mu_0\zeta^2\mathcal{B} + 2\mu_{t-\lceil\frac{t}{2}\rceil}\zeta^2\mathcal{B}.$$

Proof See Appendix for the proof. ■

Next, we provide the non-asymptotic bounds on the iteration complexity of RSPP scheme.

Theorem 18 *Let Assumptions 2, 7 and 8 hold and $\epsilon, \mu_0 > 0$. Also let $\gamma > 0$ and $\{x^{K_t,t}\}_{t\geq 0}$ be generated by RSPP scheme with $\mu_t = \frac{\mu_0}{t^\gamma}$ and $K_t = \lceil t^\gamma \rceil$. If we perform the following number of epochs:*

$$T = \left\lceil \max\left\{ \ln\left(\frac{2r_{0,0}^2}{\epsilon}\right)\frac{1}{\ln(1/\theta_0)}, \left(\frac{2^{\gamma+1}\mathcal{D}_r}{\epsilon}\mathcal{C}\right)^{1/\gamma}\right\}\right\rceil,$$

then after a total number of SPP iterations of $\frac{T^{1+\gamma}}{1+\gamma}$, which is bounded by

$$\left\lceil \frac{1}{1+\gamma}\max\left\{ \ln\left(\frac{2r_{0,0}^2}{\epsilon}\right)^{1+\gamma}\frac{1}{\ln(1/\theta_0)^{1+\gamma}}, \left(\frac{2^{\gamma+1}\mathcal{D}_r}{\epsilon}\mathcal{C}\right)^{1+\frac{1}{\gamma}}\right\}\right\rceil,$$

where $\mathcal{D}_r = 4\|\nabla F(x^*)\|\left[\frac{dist_X(x^{0,0})+2\mu_0\zeta^2\mathcal{B}}{\mu_0\ln(\zeta/(\zeta-1))} + 3^\gamma\mathcal{B}\zeta^2\right] + 2\eta\sqrt{2\eta^2+2\mathbb{E}[L_{f,S}^2]\mathcal{A}^2} + 2\eta\mathcal{A}\sqrt{\mathbb{E}[L_{f,S}^2]}$ and $\mathcal{C} = \frac{1}{2(1-\gamma)\ln 1/\sqrt{\theta_0}} + \frac{\mu_1^2}{(1-\theta_0)^2}$, we have $\mathbb{E}[\|x^{K_T,T}-x^*\|^2] \leq \epsilon$.

Proof See Appendix for the proof. ■

In conclusion Theorem 18 states that the RSPP algorithm with the choices $(\mu_t, K_t) = \left(\frac{\mu_0}{t^\gamma}, \frac{t^\gamma}{2}\right)$ requires $\mathcal{O}\left(\epsilon^{-\left(1+\frac{1}{\gamma}\right)}\right)$ simple SPP iterations to reach an ϵ optimal point. It is important to observe that this convergence rate is achieved when the stepsize and the epoch length are not dependent on any inaccessible constant, making our restarting scheme easily implementable. Moreover, the parameter γ can be chosen in $(0,\infty)$, i.e. our RSPP scheme allows also stepsizes $\frac{\mu_0}{t^\gamma}$, with $\gamma > 1$. By comparison, an $\mathcal{O}\left(\epsilon^{-1}\right)$ complexity is obtained for SPP with stepsize $\mu_k = \frac{\mu_0}{k}$ only when μ_0 is chosen sufficiently large such that $\theta_0 < \frac{1}{e}$. However, this condition is not easy to check. Moreover, we may fall in the case when $\theta_0 > \frac{1}{e}$, which leads to a complexity of $\mathcal{O}\left(\epsilon^{-\frac{1}{2\ln(1/\theta_0)}}\right)$ of the variable stepsize SPP scheme. Observe that the last convergence rate is implicitly dependent on the constant μ_0 and can be arbitrarily bad, while for $\gamma > 1$ sufficiently large the RSPP scheme achieves the optimal convergence rate $\mathcal{O}\left(\epsilon^{-1}\right)$.

Remark 19 *Notice that there exists a connection between the quadratic mean residual $\mathbb{E}[\|x^k - x^*\|^2]$ and the function value residual in a certain point. To obtain this relation, denote $v^k = [x^k - \frac{1}{L_F}\nabla F(x^k)]_X$ and observe that for some constant $L_F \geq \mathbb{E}[L_{f,S}]$ we have:*

$$\begin{aligned} F(v^k) &\leq F(x^k) + \langle \nabla F(x^k), v^k - x^k\rangle + \frac{L_F}{2}\|v^k - x^k\|^2 \\ &= \min_{y\in X} F(x^k) + \langle \nabla F(x^k), y - x^k\rangle + \frac{L_F}{2}\|y - x^k\|^2 \\ &\leq \min_{y\in X} F(y) + \frac{L_F}{2}\|y - x^k\|^2 \\ &\leq F(x^*) + \frac{L_F}{2}\|x^k - x^*\|^2, \end{aligned}$$

where in the second inequality we used the convexity relation. The last relation leads to $F(v^k) - F(x^*) \leq \frac{L_F}{2}\|x^k - x^*\|^2$.

7. Numerical experiments

We present numerical evidence to assess the theoretical convergence guarantees of the SPP algorithm. We provide three numerical examples: constrained stochastic least-square with

random generated data (Moulines and Bach, 2011; Toulis et al., 2016), Markowitz portfolio optimization using real data (Brodie et al., 2009; Yurtsever et al., 2016) and logistic regression using real data (Platt, 1998). In all our figures the results are averaged over 20 Monte-Carlo simulations for an algorithm.

7.1 Stochastic least-square problems using random data

In this section we evaluate the practical performance of the SPP schemes on finite large scale least-squares models. To do so, we follow a simple normal (constrained) linear regression example from (Moulines and Bach, 2011; Toulis et al., 2016). Let $m = 10^5$ be the number of observations, and $n = 20$ be the number of features. Let x^* be a randomly a priori chosen ground truth. The feature vectors $a_1, \cdots, a_m \approx \mathcal{N}_n(0, H)$ are i.i.d. normal random variables, and H is a randomly generated symmetric matrix with eigenvalues $1/k$, for $k = 1, \cdots, n$. The outcome b_S is sampled from a normal distribution as $b_S|a_S \approx \mathcal{N}(a_S^T x^*, 1)$, for $S = 1, \cdots, m$. Since the typical loss function is defined as the elementary squared residual $(a_S^T x - b_S)^2$, which is not strongly convex, we consider batches of residuals to form our loss functions, i.e we consider $\ell(x, S)$ of two forms:

$$\ell(x, S) = \|A_{j(S):j(S)+n}x - b_{j(S):j(S)+n}\|^2 \quad \text{or} \quad \ell(x, S) = (a_S^T x - b_S)^2,$$

where a_S is the Sth row of A and $A_{j(S):j(S)+n} \in \mathbb{R}^{n\times n}$ is a submatrix containing n rows of A so that the function $x \mapsto \|A_{j(S):j(S)+n}x - b_{j(S):j(S)+n}\|^2$ is strongly convex. In our tests we used $\texttt{round}(m/2n)$ batches of dimension n and we let the rest as elementary residuals, thus having in total $p = m/2 + m/n$ loss functions. Additionally, we impose on the estimator x also p linear inequality constraints $\{x \mid Cx \leq d\}$. This constraints can be found in many applications and they come from physical constraints, see e.g. (Censor et al., 2012; Rosasco et al., 2014). We choose randomly the matrix C for the constraints and $d = C \cdot x^* + [0\ 0\ 0\ v^T]^T$, where $v \geq 0$ is a random vector of appropriate dimension, i.e. three inequalities are active at the solution x^*. Besides the SPP and RSPP algorithms analyzed in the previous sections of our paper, we also implemented SGD and the averaged variant of SPP algorithm (A-SPP), which has the same SPP iteration, but outputs the average of iterates: $\hat{x}^k = (1/\sum_{i=1}^k \mu_i)\sum_{i=1}^k \mu_i x_i$. Convergence behavior of the averaged iterates of stochastic gradient has been initially proposed in the seminal paper (Polyak and Juditsky, 1992).

In Figure 1 we run algorithms SPP, RSPP, A-SPP and SGD for two values of the initial stepsize: $\mu_0 = 0.5$ and $\mu_0 = 1$. Each scheme runs for two stepsize exponents: $\gamma_1 = 1$ (left) and $\gamma_2 = 1/2$ (right). From Figure 1 we can asses one conclusion of Theorem 15: the best performance for SPP is achieved for stepsize exponent $\gamma = 1$. Moreover, we can observe that algorithm RSPP has the fastest behavior, while the averaged variant A-SPP is more robust to changes in the initial stepsize μ_0. The performance of SGD is much worse as exponent γ decreases and it is also sensitive to the learning rate μ_0. Notice that both tests are performed over m iterations (i.e. one pass through data).

In the second set of experiments, we generate random least-square problems of the form $\min_{x:Cx\leq d}\ 1/2\|Ax - b\|^2$, where both matrices A and C have $m = 10^3$ rows and generated randomly. Now, we do not impose the solution x^* to have the form given in the first test. We let SPP and RSPP algorithms to do one pass through data for various stepsize exponents

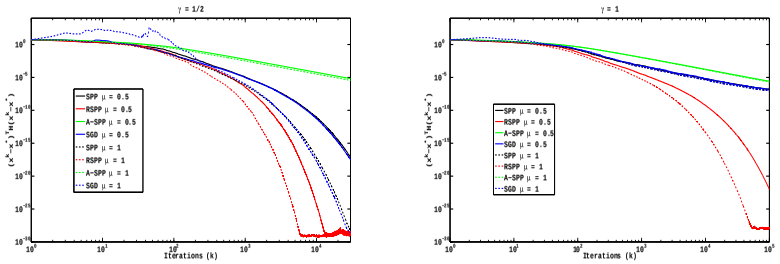


Figure 1: Performance comparison of SPP, A-SPP, RSPP and SGD for two values of initial stepsize $\mu_0 = 0.5$ and $\mu_0 = 1$ and for two values of exponent $\gamma = 1/2$ (left) and $\gamma = 1$ (right).

γ. From Figure 2 we can assess the empirical evidence of the $\mathcal{O}(1/\epsilon^{1/\gamma})$ convergence rate of Theorem 15 for SPP and $\mathcal{O}(1/\epsilon^{1+1/\gamma})$ convergence rate of Theorem 19 for RSPP, by presenting squared relative distance to the optimum solution. Moreover, the simulation results match other conclusions of Theorems 15 and 19 regarding the stepsize exponent γ: (i) the performance of SPP deteriorates with the decrease in the value of the stepsize exponent γ; (ii) from our preliminary numerical experiments we observed that RSPP scheme runs faster for higher values of γ and it has a more robust performance with respect to the variation of γ than SPP algorithm.

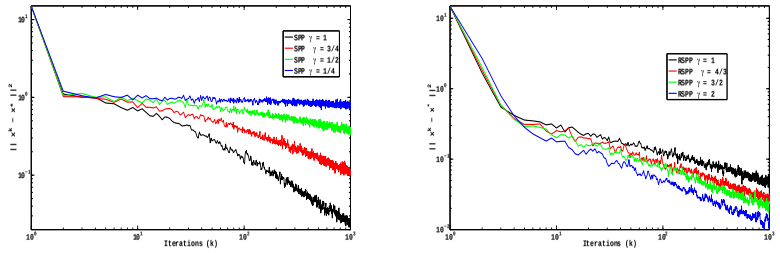


Figure 2: Performance of: SPP for four values of the stepsize exponent $\gamma = 1,\ 3/4,\ 1/2$ and $1/4$ (left); RSPP for four values of the stepsize exponent $\gamma = 1,\ 4/3,\ 3/2$ and 2 (right).

7.2 Markowitz portfolio optimization using real data

Markowitz portfolio optimization aims to reduce the risk by minimizing the variance for a given expected return. This can be mathematically formulated as a convex optimization

problem (Brodie et al., 2009; Yurtsever et al., 2016):

$$\min_{x\in\mathbb{R}^n} \mathbb{E}[(a_S^T x - b)^2] \quad \text{s.t.} \quad x \in X = \{x:\ x \geq 0,\ e^T x \leq 1,\ a_{av}^T x \geq b\},$$

where $a_{av} = \mathbb{E}[a_S]$ is the average returns for each asset that is assumed to be known (or estimated), and b represents a minimum desired return. Since new data points are arriving on-line, one cannot access the entire dataset at any moment of time, which makes the stochastic setting more favorable. For simulations, we approximate the expectation with the empirical mean as follows:

$$\min_{x\in\mathbb{R}^n} \frac{1}{m}\sum_{S=1}^{m}(a_S^T x - b)^2 \quad \text{s.t.} \quad x \in X = X_1 \cap X_2 \cap X_3,$$

where $X_1 = \{x:\ x \geq 0\}$, $X_2 = \{x:\ e^T x \leq 1\}$ and $X_3 = \{x:\ a_{av}^T x \geq b\}$. In this application we have the number of samples m larger than the number of constraints. However, by taking a certain partition of $[m] = \Omega_1 \cup \Omega_2 \cup \Omega_3$, then one can consider: $X_S = X_i$ for all $S \in \Omega_i$, with $i \in \{1, 2, 3\}$. We use 2 different real portfolio datasets: Standard & Poor's 500 (SP500, with 25 stocks for 1276 days) and one dataset by Fama and French (FF100, with 100 portfolios for 23.647 days) that is commonly used in financial literature, see e.g. (Brodie et al., 2009). We split all the datasets into test (10%) and train (90%) partitions randomly. We set the desired return a_{av} as the average return over all assets in the training set and $b = \text{mean}(a_{av})$. The results of this experiment are presented in Figure 3. We plot the value of the objective function over the datapoints in the test partition F_{test} along the iterations. We observe that SGD is very sensitive to both parameters, initial stepsize (μ_0) and stepsize exponent (γ), while SPP is more robust to changes in both parameters and also performs better over one pass through data in the train partition.

7.3 Logistic regression using real data

Finally, we consider the logistic regression problem. In this task we train an estimator over a given dataset (A, b), where $A \in \mathbb{R}^{m\times n}$ is the observations matrix and $b \in \mathbb{R}^m$ is the labels vector. For any $S \in \{1, \cdots, m\}$ we define the logistic loss function:

$$\ell(a_S^T x) = \log\left(1 + e^{-b_S(a_S^T x)}\right),$$

where $a_S \in \mathbb{R}^n$ is the Sth row of matrix A. Notice that the logistic loss function $\ell(a_S^T x)$ is only convex and smooth. However, in logistic regression we also consider a quadratic regularization term (Toulis et al., 2016; Bach, 2010):

$$\min_{x\in\mathbb{R}^n} \frac{1}{m}\sum_{S=1}^{m}\log\left(1 + e^{-b_S(a_S^T x)}\right) + \frac{\lambda}{2}\|x\|^2,$$

where $\lambda > 0$ is taken small, which makes the objective function λ-strongly convex. We have tested the four schemes (SGD, SPP, ASPP and RSPP), on the Adult datasets (a2a with $m = 2265, n = 123$ and a5a with $m = 6414, n = 123$) from LIBSVM/UCI database (Platt, 1998). We set the initial stepsize at value $\mu_0 = 0.6$ and the regularization parameter

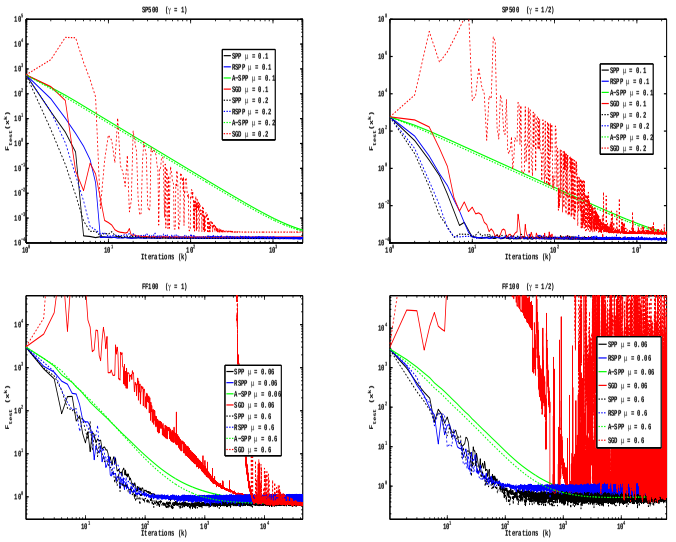


Figure 3: Performance on Markowitz portfolio using real datasets (SP500 - top and FF100 - bottom) for the SPP, A-SPP, RSPP and SGD schemes for several values of the initial stepsize μ_0 and for two values of the exponent ($\gamma = 1/2$ - left and $\gamma = 1$ - right).

$\lambda = 10^{-3}$. Once an approximate solution $\tilde{x}^*$ of the logistic regression problem is obtained, we evaluate the resulted estimator on the test dataset, i.e. $\frac{1}{2p}\sum_{S=1}^{p}|sgn(\tilde{a}_S^T\tilde{x}^*) - \tilde{b}_S|$, where $\tilde{A} \in \mathbb{R}^{p\times n}$ and $\tilde{b}$ are the testing dataset. The results are displayed in Figure 4. We observe that for large stepsize ($\gamma = 1/2$) the performances of all four methods (SGD, SPP, A-SPP and RSPP) are similar. However, when we use a smaller stepsizes ($\gamma = 1$), the RSPP algorithm outperforms the other methods. We also observe that the variation of stepsize exponent γ does not influence too much the performance of RSPP algorithm, showing once more the robustness of this scheme against variations in the stepsize choices μ_0/k^γ.

Since we used different parameter values in our experiments, we want to provide some details on the parameter choices. From the theoretical viewpoint, Theorems 15 and 19 show that the stepsize exponent γ has to be chosen as large as possible to obtain the best convergence rate. Let us consider for simplicity that $\sigma_{f,S} = \sigma > 0$ for all $S \in \Omega$. Then, for the initial stepsize μ_0 Corollary 16 indicates that the best convergence rate is obtained for $\mu_0 > \frac{\sqrt{e}-1}{\sigma}$. Therefore, in the case when σ is known (e.g. regularized logistic regression) we can choose μ_0 appropriately so that we obtain the best convergence. However, when this parameter σ is not known, then there is an inherent need for parameter tuning. From practical point of view, our plots show that the performance of SPP/RSPP deteriorates with the decrease

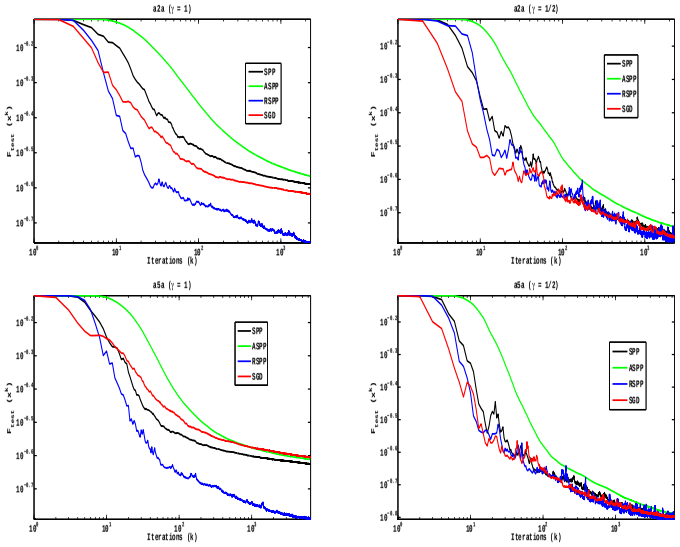


Figure 4: Performance on logistic regression using real datasets (a2a - top and a5a - bottom) for the SPP, A-SPP, RSPP and SGD schemes for two values of the exponent ($\gamma = 1/2$ - left and $\gamma = 1$ - right).

in value of the exponent γ. That is, they indicate that the higher values of this parameter the better performance. However, there is empirical evidence in the literature regarding the SGD performance which shows that values $\gamma < 1$ provide a better performance than the choice $\gamma = 1$. In these cases, the choice of initial stepsize μ_0 requires a detailed tuning procedure. As a general conclusion of our experiments, we can state that both parameters μ_0 and γ are strongly linked to the problem conditioning and, up to some extent, they have to be tuned accordingly to the problem datasets.

8. Appendix

To make the paper more readable, in this Appendix we provide the proofs of some lemmas and theorems.

Proof of Lemma 4:

Proof By using the convexity of the function $\mathbb{I}_{\mu,S}(x) = \frac{1}{2\mu}\text{dist}^2_{X_S}(x)$ and taking the conditional expectation w.r.t. S_k over the history $\mathcal{F}_{k-1} = \{S_0, \cdots, S_{k-1}\}$, we get:

$$\mathbb{E}[\mathbb{I}_{1,S_k}(\hat{y}^k)|\mathcal{F}_{k-1}] \geq \mathbb{E}\left[\mathbb{I}_{1,S_k}(\hat{x}^k) + \langle \nabla \mathbb{I}_{1,S_k}(\hat{x}^k), \hat{y}^k - \hat{x}^k\rangle |\mathcal{F}_{k-1}\right].$$

Taking further the expectation over $\mathcal{F}_{k-1}$ we obtain:

$$\begin{aligned}
\mathbb{E}[\mathbb{I}_{1,S_k}(\hat{y}^k)] &\geq \mathbb{E}\left[\mathbb{I}_{1,S_k}(\hat{x}^k)\right] + \mathbb{E}\left[\langle \nabla \mathbb{I}_{1,S_k}(\hat{x}^k), \hat{y}^k - \hat{x}^k\rangle\right] \\
&= \mathbb{E}\left[\mathbb{I}_{1,S_k}(\hat{x}^k)\right] + \frac{\mathbb{E}\left[\langle \nabla \mathbb{I}_{1,S_k}(\hat{x}^k), \sum_{i=0}^{k-1} \mu_i^2 \nabla f_{\mu_i}(x^i;S_i)\rangle\right]}{\hat{\mu}_{1,k}} \\
&\geq \mathbb{E}\left[\mathbb{I}_{1,S_k}(\hat{x}^k)\right] - \mathbb{E}\left[\frac{\hat{\mu}_{2,k}}{\hat{\mu}_{1,k}}\|\nabla \mathbb{I}_{1,S_k}(\hat{x}^k)\|\left\|\sum_{i=0}^{k-1}\frac{\mu_i^2}{\hat{\mu}_{2,k}}\nabla f_{\mu_i}(x^i;S_i)\right\|\right] \\
&\geq \mathbb{E}\left[\mathbb{I}_{1,S_k}(\hat{x}^k)\right] - \frac{\hat{\mu}_{2,k}}{\hat{\mu}_{1,k}}\mathbb{E}\left[\|\nabla \mathbb{I}_{1,S_k}(\hat{x}^k)\|\sum_{i=0}^{k-1}\frac{\mu_i^2}{\hat{\mu}_{2,k}}\|\nabla f_{\mu_i}(x^i;S_i)\|\right],
\end{aligned}$$

where in the second inequality we used the Cauchy-Schwarz inequality and in the third the convexity relation regarding $\|\cdot\|$. Further, using as well Lemma 3, Assumption 2 and Cauchy-Schwarz inequality, we have:

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{2}\text{dist}^2_{X_{S_k}}(\hat{y}^k)\right] &\overset{\text{Lemma 3}}{\geq} \mathbb{E}\left[\frac{1}{2}\text{dist}^2_{X_{S_k}}(\hat{x}^k)\right] - \frac{\hat{\mu}_{2,k}}{2\hat{\mu}_{1,k}}\mathbb{E}\left[\text{dist}_{X_{S_k}}(\hat{x}^k)L_{f,S_k}\right] \\
&\overset{\text{Assump. 2}}{\geq} \frac{1}{2\zeta}\mathbb{E}\left[\text{dist}^2_X(\hat{x}^k)\right] - \frac{\hat{\mu}_{2,k}}{2\hat{\mu}_{1,k}}\sqrt{\mathbb{E}[\text{dist}^2_X(\hat{x}^k)]}\sqrt{\mathbb{E}[L^2_{f,S}]},
\end{aligned}$$

which proves the statement of the lemma. ■

Proof of Theorem 5:

Proof Since the function $z \to f(z;S) + \frac{1}{2\mu}\|z - x\|^2$ is strongly convex, we have:

$$\begin{aligned}
f(z;S) + \frac{1}{2\mu}\|z - x\|^2 &\geq f(z_\mu(x;S);S) + \frac{1}{2\mu}\|z_\mu(x;S) - x\|^2 + \frac{1}{2\mu}\|z_\mu(x;S) - z\|^2 \\
&= f_\mu(x;S) + \frac{1}{2\mu}\|z_\mu(x;S) - z\|^2 \quad \forall z \in \mathbb{R}^n. \qquad (16)
\end{aligned}$$

By taking $x = x^k, S = S_k, z = x^*, \mu = \mu_k$ in (16) and using the strictly nonexpansive property of the projection operator, see e.g. (Nedic, 2011):

$$\|x - \Pi_{X_{S_k}}(x)\|^2 \leq \|x - z\|^2 - \|z - \Pi_{X_{S_k}}(x)\|^2 \qquad \forall z \in X_{S_k},\ x \in \mathbb{R}^n, \qquad (17)$$

then these lead to:

$$\begin{aligned}
f(x^*;S_k) + \frac{1}{2\mu_k}\|x^k - x^*\|^2 &\geq f_{\mu_k}(x^k;S_k) + \frac{1}{2\mu_k}\|y^k - x^*\|^2 \\
&\overset{(17)}{\geq} f_{\mu_k}(x^k;S_k) + \frac{1}{2\mu_k}\|\Pi_{X_{S_k}}(y^k) - x^*\|^2 + \frac{1}{2\mu_k}\|y^k - \Pi_{X_{S_k}}(y^k)\|^2 \\
&= f_{\mu_k}(x^k;S_k) + \frac{1}{2\mu_k}\|x^{k+1} - x^*\|^2 + \frac{1}{2\mu_k}\|y^k - x^{k+1}\|^2, \qquad (18)
\end{aligned}$$

where in the second inequality we used (17) with $x = y^k$ and $z = x^*$. For simplicity we denote $\mathbb{I}_{\mu,S}(x) = \frac{1}{2\mu}\|x - \Pi_{X_S}(x)\|^2$. From relation (18), it can be easily seen that:

$$\begin{aligned}
&\mu_k(f(x^k;S_k) - f(x^*;S_k)) + \mathbb{I}_{1,S_k}(y^k) - \frac{\mu_k^2}{2}L_{f,S_k}^2 \\
&\leq \mu_k(f(x^k;S_k) - f(x^*;S_k)) + \mathbb{I}_{1,S_k}(y^k) - \frac{\mu_k^2}{2}\|\nabla f(x^k;S_k)\|^2 \\
&= \mu_k(f(x^k;S_k) - f(x^*;S_k)) + \mathbb{I}_{1,S_k}(y^k) + \min_{z\in\mathbb{R}^n}\left[\mu_k\langle\nabla f(x^k;S_k), z - x^k\rangle + \frac{1}{2}\|z - x^k\|^2\right] \\
&\leq \mu_k(f(x^k;S_k) - f(x^*;S_k)) + \mathbb{I}_{1,S_k}(y^k) + \mu_k\langle\nabla f(x^k;S_k), y^k - x^k\rangle + \frac{1}{2}\|y^k - x^k\|^2 \\
&= \mu_k(f(x^k;S_k) + \langle\nabla f(x^k;S_k), y^k - x^k\rangle + \frac{1}{2\mu_k}\|y^k - x^k\|^2 - f(x^*;S_k)) + \mathbb{I}_{1,S_k}(y^k) \\
&\overset{\text{conv. f}}{\leq} \mu_k(f_{\mu_k}(x^k;S_k) - f(x^*;S_k)) + \mathbb{I}_{1,S_k}(y^k) \\
&\overset{(18)}{\leq} \frac{1}{2}\|x^k - x^*\|^2 - \frac{1}{2}\|x^{k+1} - x^*\|^2.
\end{aligned}$$

Taking now the conditional expectation in S_k w.r.t. the history $\mathcal{F}_{k-1} = \{S_0, \cdots, S_{k-1}\}$ in the last inequality we have:

$$\begin{aligned}
\mu_k(F(x^k) - F(x^*)) + \mathbb{E}[\mathbb{I}_{1,S_k}(y^k)|\mathcal{F}_{k-1}] - \frac{\mu_k^2}{2}\mathbb{E}[L_{f,S_k}^2] \\
\leq \frac{1}{2}\|x^k - x^*\|^2 - \frac{1}{2}\mathbb{E}[\|x^{k+1} - x^*\|^2|\mathcal{F}_{k-1}].
\end{aligned}$$

Taking further the expectation over $\mathcal{F}_{k-1}$ and summing over $i = 0, \cdots, k-1$, results in:

$$\begin{aligned}
\frac{\|x^0 - x^*\|^2}{2\sum_{i=0}^{k-1}\mu_i} &\geq \frac{1}{\sum_{i=0}^{k-1}\mu_i}\sum_{i=0}^{k-1}\mathbb{E}[\mu_i(F(x^i) - F(x^*))] + \mathbb{E}[\mathbb{I}_{1,S}(y^i)] - \frac{\mu_i^2}{2}\mathbb{E}[L_{f,S}^2] \\
&= \frac{1}{\sum_{i=0}^{k-1}\mu_i}\sum_{i=0}^{k-1}\mathbb{E}[\mu_i(F(x^i) - F(x^*))] + \mu_i\mathbb{E}[\mathbb{I}_{\mu_i,S}(y^i)] - \frac{\mu_i^2}{2}\mathbb{E}[L_{f,S}^2] \\
&\geq \frac{1}{\sum_{i=0}^{k-1}\mu_i}\sum_{i=0}^{k-1}\mathbb{E}[\mu_i(F(x^i) - F(x^*))] + \mu_i\mathbb{E}[\mathbb{I}_{\mu_0,S}(y^i)] - \frac{\mu_i^2}{2}\mathbb{E}[L_{f,S}^2] \\
&\overset{\text{Jensen}}{\geq} \mathbb{E}[F(\hat{x}^k) - F(x^*)] + \mathbb{E}[\mathbb{I}_{\mu_0,S}(\hat{y}^k)] - \frac{\mathbb{E}[L_{f,S}^2]\hat{\mu}_{2,k}}{2\hat{\mu}_{1,k}},
\end{aligned} \tag{19}$$

where in the second inequality we used that $\mathbb{I}_{\mu_i,S}(y) \geq \mathbb{I}_{\mu_0,S}(y)$ for all $S \in \Omega, i \geq 0$. The relation (19) implies the following upper bound on the suboptimality gap:

$$\mathbb{E}[F(\hat{x}^k) - F(x^*)] \leq \frac{\|x^0 - x^*\|^2 + \mathbb{E}[L_{f,S}^2]\hat{\mu}_{2,k}}{2\hat{\mu}_{1,k}}. \tag{20}$$

On the other hand, recalling $\nabla F(x^*) = \mathbb{E}[\nabla f(x^*;S)]$, we use the following fact:

$$\begin{aligned}
\mathbb{E}[F(\hat{x}^k)] - F(x^*) &\geq \mathbb{E}[\langle\nabla F(x^*), \hat{x}^k - x^*\rangle] \\
&= \mathbb{E}[\langle\nabla F(x^*), \Pi_X(\hat{x}^k) - x^*\rangle] + \mathbb{E}[\langle\nabla F(x^*), \hat{x}^k - \Pi_X(\hat{x}^k)\rangle] \\
&\geq -\mathbb{E}[L_{f,S}]\mathbb{E}[\text{dist}_X(\hat{x}^k)] \\
&\overset{\text{Jensen}}{\geq} -\sqrt{\mathbb{E}[L_{f,S}^2]\ \mathbb{E}[\text{dist}_X^2(\hat{x}^k)]} \qquad \forall k \geq 0,
\end{aligned} \tag{21}$$

which is derived from the optimality conditions $\langle\nabla F(x^*), z - x^*\rangle \geq 0$ for all $z \in X$, the Cauchy-Schwarz and Jensen inequalities. By denoting $r_0 = \|x^0 - x^*\|$ and combining (19) with Lemma 4 and the last inequality (21), we obtain:

$$\begin{aligned}
\mathbb{E}[\text{dist}_X^2(\hat{x}^k)] - \zeta\sqrt{\mathbb{E}[L_{f,S}^2]}\left(\frac{\hat{\mu}_{2,k}}{\hat{\mu}_{1,k}} + 2\mu_0\right)\sqrt{\mathbb{E}[\text{dist}_X^2(\hat{x}^k)]} \\
\overset{\text{Lemma 4+(21)}}{\leq} 2\mu_0\zeta\mathbb{E}[F(\hat{x}^k) - F(x^*)] + 2\mu_0\zeta\mathbb{E}[\mathbb{I}_{\mu_0,S}(\hat{y}^k)] \\
\overset{(20)}{\leq} \frac{\mu_0\zeta r_0^2 + \mu_0\zeta\mathbb{E}[L_{f,S}^2]\hat{\mu}_{2,k}}{\hat{\mu}_{1,k}}.
\end{aligned}$$

This last relation clearly implies an upper bound on the feasibility residual:

$$\sqrt{\mathbb{E}[\text{dist}_X^2(\hat{x}^k)]} \leq \zeta\sqrt{\mathbb{E}[L_{f,S}^2]}\left(\frac{\hat{\mu}_{2,k}}{\hat{\mu}_{1,k}} + 2\mu_0\right) + \sqrt{\frac{\mu_0\zeta r_0^2 + \mu_0\zeta\mathbb{E}[L_{f,S}^2]\hat{\mu}_{2,k}}{\hat{\mu}_{1,k}}}. \tag{22}$$

Also, combining (21) and (22) we obtain the lower bound on the suboptimality gap:

$$\mathbb{E}[F(\hat{x}^k)] - F^* \geq -\zeta\mathbb{E}[L_{f,S}^2]\left(\frac{\hat{\mu}_{2,k}}{\hat{\mu}_{1,k}} + 2\mu_0\right) - \sqrt{\mathbb{E}[L_{f,S}^2]}\sqrt{\frac{\mu_0\zeta r_0^2 + \mu_0\zeta\mathbb{E}[L_{f,S}^2]\hat{\mu}_{2,k}}{\hat{\mu}_{1,k}}}. \tag{23}$$

From the upper and lower suboptimality bounds (20), (23) and feasibility bound (22), we deduce our convergence rate results. ■

Proof of Lemma 9:

Proof Let $\sigma_{f,S} \geq 0$ be the strong convexity constant of the function $f(\cdot;S)$. Notice that we allow the convex case, that is $\sigma_{f,S} = 0$ for some S. Then, it is known that the Moreau approximation $f_\mu(\cdot;S)$ is also a $\hat{\sigma}_{f,S}$-strongly convex function with strong convexity constant, see e.g. (Rockafellar and Wets, 1998):

$$\hat{\sigma}_{f,S} = \frac{\sigma_{f,S}}{1 + \mu\sigma_{f,S}}.$$

Clearly, in the simple convex case, that is $\sigma_{f,S} = 0$, we also have $\hat{\sigma}_{f,S} = 0$. By denoting $\hat{L}_{f,S} = \frac{1}{\mu}$ the Lipschitz constant of the gradient of $f_\mu(\cdot;S)$, the following well-known relation holds for the smooth and (strongly) convex function $f_\mu(\cdot;S)$, see e.g. (Nesterov, 2004):

$$\begin{aligned}
\langle\nabla f_\mu(x;S) - \nabla f_\mu(y;S), x - y\rangle \geq \frac{1}{\hat{\sigma}_{f,S} + \hat{L}_{f,S}}\|\nabla f_\mu(x;S) - \nabla f_\mu(y;S)\|^2 \\
+ \frac{\hat{\sigma}_{f,S}\hat{L}_{f,S}}{\hat{L}_{f,S} + \hat{\sigma}_{f,S}}\|x - y\|^2 \quad \forall x, y \in \mathbb{R}^n.
\end{aligned} \tag{24}$$

By using Assumption 7, then it can be also obtained that:

$$\|\nabla f_\mu(x;S)-\nabla f_\mu(y;S)\|\geq \hat{\sigma}_{f,S}\|x-y\| \qquad \forall x,y\in\mathbb{R}^n. \tag{25}$$

Using this relation, we further derive that:

$$\begin{aligned}
&\|z_\mu(x;S)-z_\mu(y;S)\|^2=\|x-y+\mu(\nabla f_\mu(y;S)-\nabla f_\mu(x;S))\|^2\\
&=\|x-y\|^2+2\mu\langle\nabla f_\mu(y;S)-\nabla f_\mu(x;S),x-y\rangle+\mu^2\|\nabla f_\mu(x;S)-\nabla f_\mu(y;S)\|^2\\
&\overset{(24)}{\leq}\left(1-\frac{2\mu\hat{\sigma}_{f,S}\hat{L}_{f,S}}{\hat{L}_{f,S}+\hat{\sigma}_{f,S}}\right)\|x-y\|^2+\mu\left(\mu-\frac{2}{\hat{L}_{f,S}+\hat{\sigma}_{f,S}}\right)\|\nabla f_\mu(x;S)-\nabla f_\mu(y;S)\|^2\\
&\overset{(25)}{\leq}\left[1+\hat{\sigma}_{f,S}^2\left(\mu^2-\frac{2\mu}{\hat{\sigma}_{f,S}+\hat{L}_{f,S}}\right)-\frac{2\mu\hat{\sigma}_{f,S}\hat{L}_{f,S}}{\hat{L}_{f,S}+\hat{\sigma}_{f,S}}\right]\|x-y\|^2\\
&=(1-\hat{\sigma}_{f,S}\mu)^2\|x-y\|^2 \qquad \forall x,y\in\mathbb{R}^n,
\end{aligned}$$

which implies our result. ■

Proof of Lemma 11:
Proof By taking $\mu=\mu_k$ in relation (14), we obtain:

$$\sqrt{\mathbb{E}[\|x^{k+1}-x^*\|^2]}\leq\sqrt{\mathbb{E}\left[\theta_S^2(\mu_k)\right]}\sqrt{\mathbb{E}[\|x^k-x^*\|^2]}+\mu_k\sqrt{\mathbb{E}\left[\|\nabla f(x^*;S)\|^2\right]}.$$

By using the notations $r_k=\sqrt{\mathbb{E}[\|x^k-x^*\|^2]}$, $\theta_k=\sqrt{\mathbb{E}\left[\theta_S^2(\mu_k)\right]}$ and $\eta=\sqrt{\mathbb{E}\left[\|\nabla f(x^*;S)\|^2\right]}$, the last inequality leads to:

$$\begin{aligned}
r_{k+1}&\leq\theta_k r_k+(1-\theta_k)\frac{\mu_k}{1-\theta_k}\eta\\
&\leq\max\left\{r_k,\frac{\mu_k}{1-\theta_k}\eta\right\}\leq\max\left\{r_0,\frac{\mu_0}{1-\theta_0}\eta,\cdots,\frac{\mu_k}{1-\theta_k}\eta\right\}.
\end{aligned}\tag{26}$$

By observing the fact that $t\mapsto\mathbb{E}\left[\frac{\sigma_{f,S}}{(1+t\sigma_{f,S})^2}+\frac{\sigma_{f,S}}{1+t\sigma_{f,S}}\right]$ is nonincreasing in t, and implicitly:

$$\begin{aligned}
\frac{\mu_{k-1}}{1-\theta_{k-1}}&=\frac{1}{\mathbb{E}\left[\frac{\sigma_{f,S}}{(1+\mu_{k-1}\sigma_{f,S})^2}+\frac{\sigma_{f,S}}{1+\mu_{k-1}\sigma_{f,S}}\right]}\\
&\geq\frac{1}{\mathbb{E}\left[\frac{\sigma_{f,S}}{(1+\mu_k\sigma_{f,S})^2}+\frac{\sigma_{f,S}}{1+\mu_k\sigma_{f,S}}\right]}=\frac{\mu_k}{1-\theta_k},
\end{aligned}$$

then we have $\max\limits_{0\leq i\leq k}\frac{\mu_i}{1-\theta_i}=\frac{\mu_0}{1-\theta_0}$ and the relation (26) becomes:

$$r_k\leq\max\left\{r_0,\frac{\mu_0}{1-\theta_0}\eta\right\}\quad\forall k\geq 0, \tag{27}$$

which implies our result. ■

We also present the following useful auxiliary result:

Lemma 20 *Let $\gamma\in(0,1]$ and the integers $p,q\in\mathbb{N}$ with $q\geq p\geq 1$. Given the sequence of stepsizes $\mu_k=\frac{\mu_0}{k^\gamma}$ for all $k\geq 1$, where $\mu_0>0$, then the following relation holds:*

$$\prod_{i=p}^{q}\mathbb{E}[\theta_S^2(\mu_i)]\leq\left(\mathbb{E}\left[\theta_S^2(\mu_0)\right]\right)^{\varphi_{1-\gamma}(q+1)-\varphi_{1-\gamma}(p)}$$

Proof From definition of $\theta_S(\mu)$ for any $k\geq 1$ we have:

$$\begin{aligned}
\mathbb{E}[\theta_S^2(\mu_k)]&=\mathbb{E}\left[\left(\frac{1}{1+\mu_k\sigma_{f,S}}\right)^2\right]=\mathbb{E}\left[\frac{1}{\left(1+\frac{\mu_0}{k^\gamma}\sigma_{f,S}\right)^2}\right]\\
&\overset{(7)}{\leq}\mathbb{E}\left[\left(\frac{1}{1+\mu_0\sigma_{f,S}}\right)^{\frac{2}{k^\gamma}}\right]\leq\left(\mathbb{E}\left[\frac{1}{(1+\mu_0\sigma_{f,S})^2}\right]\right)^{\frac{1}{k^\gamma}}=\left(\mathbb{E}[\theta_S^2(\mu_0)]\right)^{\frac{1}{k^\gamma}}.
\end{aligned}\tag{28}$$

By taking into account that $\mathbb{E}[\theta_S^2(\mu_0)]=\mathbb{E}\left[\frac{1}{(1+\mu_0\sigma_{f,S})^2}\right]\leq 1$ and that

$$\sum_{i=p}^{q}\frac{1}{i^\gamma}\geq\varphi_{1-\gamma}(q+1)-\varphi_{1-\gamma}(p)=\int_p^{q+1}\frac{1}{t^\gamma}dt=\begin{cases}\ln\frac{q+1}{p} & \text{if}\quad\gamma=1\\ \frac{(q+1)^{1-\gamma}-p^{1-\gamma}}{1-\gamma} & \text{if}\quad\gamma<1,\end{cases}$$

then the relation (28) implies:

$$\begin{aligned}
\prod_{i=p}^{q}\mathbb{E}[\theta_S^2(\mu_i)]&\leq\left(\mathbb{E}\left[\theta_S^2(\mu_0)\right]\right)^{\sum\limits_{i=p}^{q}\frac{1}{i^\gamma}}\leq\left(\mathbb{E}\left[\theta_S^2(\mu_0)\right]\right)^{\varphi_{1-\gamma}(q+1)-\varphi_{1-\gamma}(p)}\\
&=\begin{cases}\left(\mathbb{E}\left[\theta_S^2(\mu_0)\right]\right)^{\ln\frac{q+1}{p}} & \text{if}\quad\gamma=1\\ \left(\mathbb{E}\left[\theta_S^2(\mu_0)\right]\right)^{\frac{(q+1)^{1-\gamma}-p^{1-\gamma}}{1-\gamma}} & \text{if}\quad\gamma<1,\end{cases}
\end{aligned}\tag{29}$$

which immediately implies the above statement. ■

Proof of Lemma 13:
Proof By using the strictly nonexpansive property of the projection operator (17), with $z=\Pi_X(y^k), x=y^k$, and the linear regularity assumption, we obtain:

$$\begin{aligned}
\mathbb{E}[\text{dist}_X^2(x^{k+1})]&\leq\mathbb{E}[\|x^{k+1}-\Pi_X(y^k)\|^2]\overset{(17)}{\leq}\mathbb{E}[\|y^k-\Pi_X(y^k)\|^2]-\mathbb{E}[\|y^k-x^{k+1}\|^2]\\
&\overset{\text{As. 2}}{\leq}\mathbb{E}[\|y^k-\Pi_X(y^k)\|^2]-\frac{1}{\zeta}\mathbb{E}[\|y^k-\Pi_X(y^k)\|^2]\\
&=\left(1-\frac{1}{\zeta}\right)\mathbb{E}[\text{dist}_X^2(y^k)].
\end{aligned}\tag{30}$$

On the other hand, from triangle inequality and Minkowski inequality, we obtain:

$$\begin{aligned}\sqrt{\mathbb{E}[\mathrm{dist}_X^2(y^k)]} &\leq \sqrt{\mathbb{E}[\|y^k-\Pi_X(x^k)\|^2]} \leq \sqrt{\mathbb{E}[(\|y^k-x^k\|+\mathrm{dist}_X(x^k))^2]}\\ &\overset{(8)}{\leq} \sqrt{\mathbb{E}[\|z_{\mu_k}(x^k;S_k)-x^k\|^2]}+\sqrt{\mathbb{E}[\mathrm{dist}_X^2(x^k)]}\\ &= \sqrt{\mathbb{E}[\mathrm{dist}_X^2(x^k)]}+\mu_k\sqrt{\mathbb{E}[\|\nabla f_{\mu_k}(x^k;S_k)\|^2]}\\ &\overset{\text{Lemma 3}}{\leq} \sqrt{\mathbb{E}[\mathrm{dist}_X^2(x^k)]}+\mu_k\sqrt{\mathbb{E}[\|\nabla f(x^k;S_k)\|^2]}\\ &\overset{\text{Lemma 12}}{\leq} \sqrt{\mathbb{E}[\mathrm{dist}_X^2(x^k)]}+\mu_k\left(\sqrt{2\mathbb{E}[\|\nabla f(x^*;S)\|^2]}+\mathcal{A}\sqrt{2\mathbb{E}[L_{f,S}^2]}\right). \quad (31)\end{aligned}$$

For simplicity we use notations: $\alpha=\sqrt{1-\frac{1}{\zeta}}$, $d_k=\sqrt{\mathbb{E}[\mathrm{dist}_X^2(x^k)]}$ and $\mathcal{B}=\sqrt{2\mathbb{E}[\|\nabla f(x^*;S)\|^2]}+\mathcal{A}\sqrt{2\mathbb{E}[L_{f,S}^2]}$. Combining (30) and (31) yields:

$$d_{k+1}\leq \alpha d_k+\alpha\mu_k\mathcal{B}\leq \alpha^{k+1}d_0+\mathcal{B}\sum_{i=1}^{k+1}\alpha^i\mu_{k-i+1}. \quad (32)$$

Define $m=\lceil\frac{k+1}{2}\rceil$. By dividing the sum from the right side of (32) in two parts and by taking into account that $\{\mu_k\}_{k\geq 0}$ is nonincreasing, then results in:

$$\begin{aligned}\sum_{i=1}^{k+1}\alpha^i\mu_{k-i+1} &= \sum_{i=1}^{m}\alpha^i\mu_{k-i+1}+\sum_{i=m+1}^{k+1}\alpha^i\mu_{k-i+1}\\ &\leq \mu_{k-m+1}\sum_{i=1}^{m}\alpha^i+\alpha^{m+1}\sum_{i=0}^{k-m}\alpha^i\mu_{k-i-m}\\ &\leq \mu_{k-m+1}\frac{\alpha(1-\alpha^m)}{1-\alpha}+\mu_0\alpha^{m+1}\frac{1-\alpha^{k-m+1}}{1-\alpha}\\ &\leq \mu_{k-m+1}\frac{\alpha}{1-\alpha}+\alpha^{m+1}\frac{\mu_0}{1-\alpha}.\end{aligned}$$

By using the last inequality into (32) and using the bound $\frac{\alpha}{1-\alpha}\leq 2\zeta$, then these facts imply the statement of the lemma. ■

Proof of Theorem 14:
Proof Let $\mu>0, x\in\mathbb{R}^n$ and $S\in\Omega$, then we have:

$$\begin{aligned}&\frac{1}{2}\|z_\mu(x;S)-x^*\|^2\\ &=\frac{1}{2}\|z_\mu(x;S)-z_\mu(x^*;S)\|^2+\langle z_\mu(x;S)-z_\mu(x^*;S),z_\mu(x^*;S)-x^*\rangle+\frac{1}{2}\|z_\mu(x^*;S)-x^*\|^2\\ &\leq \frac{\theta_S^2(\mu)}{2}\|x-x^*\|^2-\mu\langle\nabla f(x^*;S),x-x^*\rangle+\langle z_\mu(x^*;S)-x^*+\mu\nabla f(x^*;S),x-x^*\rangle\\ &\qquad+\langle z_\mu(x;S)-x,z_\mu(x^*;S)-x^*\rangle-\frac{\mu^2}{2}\|\nabla f_\mu(x^*;S)\|^2. \quad (33)\end{aligned}$$

Now we take expectation in both sides and consider $x=x^k$ and $\mu=\mu_k$. We thus seek a bound for each term from the right hand side in (33). For the second term, by using the optimality conditions $\langle\nabla F(x^*),z-x^*\rangle\geq 0$ for all $z\in X$, we have:

$$\begin{aligned}\mathbb{E}[\langle\nabla f(x^*;S),x^*-x^k\rangle] &= \mathbb{E}[\langle\nabla F(x^*),x^*-\Pi_X(x^k)\rangle]+\mathbb{E}[\langle\nabla F(x^*),\Pi_X(x^k)-x^k\rangle]\\ &\leq \mathbb{E}[\langle\nabla F(x^*),\Pi_X(x^k)-x^k\rangle]\\ &\overset{\text{C.-S.}}{\leq} \|\nabla F(x^*)\|\,\mathbb{E}[\mathrm{dist}_X(x^k)]\leq\|\nabla F(x^*)\|\sqrt{\mathbb{E}[\mathrm{dist}_X^2(x^k)]}\\ &\overset{\text{Lemma 13}}{\leq} \|\nabla F(x^*)\|\left[\left(1-\frac{1}{\zeta}\right)^{\frac{k}{2}}\left(\mathrm{dist}_X(x^0)+2\mu_0\zeta\mathcal{B}\right)+2\mu_{k-\lceil\frac{k}{2}\rceil}\zeta\mathcal{B}\right],\end{aligned}$$

where in the second inequality we used the Cauchy-Schwarz inequality. By using that $e^x\geq 1+x$, for all $x\geq 0$, and the fact that $\frac{1}{k}\leq\frac{1}{k^\gamma}$ when $k\geq 1$ and $\gamma\in(0,1]$, then the last inequality implies:

$$\begin{aligned}\mathbb{E}[\langle\nabla f(x^*;S),x^*-x^k\rangle] &\leq \|\nabla F(x^*)\|\left[\frac{2\mathrm{dist}_X(x^0)+4\mu_0\zeta\mathcal{B}}{k\ln(\zeta/(\zeta-1))}+2\mu_{k-\lceil\frac{k}{2}\rceil}\mathcal{B}\zeta\right]\\ &\leq \mu_k\|\nabla F(x^*)\|\left[\frac{2\mathrm{dist}_X(x^0)+4\mu_0\zeta\mathcal{B}}{\mu_0\ln(\zeta/(\zeta-1))}+\frac{2\mu_{k-\lceil\frac{k}{2}\rceil}\mathcal{B}\zeta}{\mu_k}\right]. \quad (34)\end{aligned}$$

For the third term in (33) we observe from the optimality conditions for $z_{\mu_k}(x^*;S)$ that:

$$\begin{aligned}\left\|\frac{1}{\mu_k}\left(z_{\mu_k}(x^*;S)-x^*\right)+\nabla f(x^*;S)\right\| &= \|\nabla f(z_{\mu_k}(x^*;S);S)-\nabla f(x^*;S)\|\\ &\overset{\text{As.1}}{\leq} L_{f,S}\|z_{\mu_k}(x^*;S)-x^*\|=\mu_k L_{f,S}\|\nabla f_{\mu_k}(x^*;S)\|\\ &\overset{\text{Lemma 3}}{\leq} \mu_k L_{f,S}\|\nabla f(x^*;S)\|,\end{aligned}$$

which yields the following bound:

$$\begin{aligned}&\langle z_{\mu_k}(x^*;S)-x^*+\mu_k\nabla f(x^*;S),x^k-x^*\rangle\leq\|z_{\mu_k}(x^*;S)-x^*+\mu_k\nabla f(x^*;S)\|\cdot\|x^k-x^*\|\\ &\leq\|\mu_k\nabla f(x^*;S)-\mu_k\nabla f(z_{\mu_k}(x^*;S);S)\|\cdot\|x^k-x^*\|\overset{\text{As.1}}{\leq}\mu_k L_{f,S}\|x^*-z_{\mu_k}(x^*;S)\|\cdot\|x^k-x^*\|\\ &\leq\mu_k^2L_{f,S}\|\nabla f_{\mu_k}(x^*;S)\|\cdot\|x^k-x^*\|\overset{\text{Lemma 3}}{\leq}\mu_k^2L_{f,S}\|\nabla f(x^*;S)\|\cdot\|x^k-x^*\|,\end{aligned}$$

where in the first inequality we used the Cauchy-Schwarz. By taking expectation in both sides and using Lemma 11, we obtain the refinement:

$$\begin{aligned}\mathbb{E}[\langle z_{\mu_k}(x^*;S)-x^*+\mu_k\nabla f(x^*;S),x^k-x^*\rangle] &= \mu_k\mathbb{E}[\langle\nabla f(x^*;S)-\nabla f(z_{\mu_k}(x^*;S);S),x^k-x^*\rangle]\\ &\leq \mu_k\mathbb{E}[\|\nabla f(x^*;S)-\nabla f(z_{\mu_k}(x^*;S);S)\|\|x^k-x^*\|] \quad (35)\end{aligned}$$

$$\begin{aligned}
&\overset{\text{As. 1}}{\leq} \mu_k\mathbb{E}[L_{f,S}\|x^*-z_{\mu_k}(x^*;S)\|\|x^k-x^*\|]\\
&=\mu_k\mathbb{E}[L_{f,S}\|\nabla f_{\mu_k}(x^*;S)\|\|x^k-x^*\|]\\
&\overset{\text{Lemma 3}}{\leq} \mu_k\mathbb{E}[L_{f,S}\|\nabla f(x^*;S)\|\|x^k-x^*\|]\\
&\leq \mu_k^2\sqrt{\mathbb{E}[L_{f,S}^2]}\sqrt{\mathbb{E}[\|\nabla f(x^*;S)\|^2]}\mathbb{E}[\|x^k-x^*\|]\\
&\leq \mu_k^2\sqrt{\mathbb{E}[L_{f,S}^2]}\sqrt{\mathbb{E}[\|\nabla f(x^*;S)\|^2]}\mathbb{E}[\|x^k-x^*\|]\\
&\overset{\text{Lemma 11}}{\leq} \mu_k^2\sqrt{\mathbb{E}[L_{f,S}^2]}\eta\mathcal{A}, \qquad (36)
\end{aligned}$$

where in the first inequality we again used Cauchy-Schwarz relation and in the second we used Assumption 1. Finally, for the fourth term in (33) we use Lemma 12:

$$\begin{aligned}
&\mathbb{E}[\langle z_{\mu_k}(x^k;S)-x^k, z_{\mu_k}(x^*;S)-x^*\rangle] = \mu_k^2\mathbb{E}[\langle\nabla f_{\mu_k}(x^k;S),\nabla f_{\mu_k}(x^*;S)\rangle]\\
&\leq \mu_k^2\mathbb{E}[\|\nabla f_{\mu_k}(x^k;S)\|\|\nabla f_{\mu_k}(x^*;S)\|]\\
&\overset{\text{Lemma 3}}{\leq} \mu_k^2\mathbb{E}[\|\nabla f(x^k;S)\|\|\nabla f(x^*;S)\|] \leq \mu_k^2\sqrt{\mathbb{E}[\|\nabla f(x^k;S)\|^2]\mathbb{E}[\|\nabla f(x^*;S)\|^2]}\\
&\overset{\text{Lemma 12}}{\leq} \mu_k^2\eta\sqrt{2\eta^2+2\mathbb{E}[L_{f,S}^2]\mathcal{A}^2}, \qquad (37)
\end{aligned}$$

where in the first inequality we used Cauchy-Schwarz. By taking expectation in (33), using the relations (34)-(37) and taking into account that $\frac{\mu_k}{\mu_{k-\lceil\frac{k}{2}\rceil}}\leq 3^\gamma$ for all $k\geq 1$, we obtain:

$$\begin{aligned}
&\mathbb{E}[\|z_{\mu_k}(x^k;S)-x^*\|^2]\\
&\leq \mathbb{E}\left[\theta_S^2(\mu_k)\|x^k-x^*\|^2\right]+4\mu_k^2\|F(x^*)\|\left[\frac{\operatorname{dist}_X(x^0)+2\mu_0\zeta\mathcal{B}}{\mu_0\ln\left(\zeta/(\zeta-1)\right)}+3^\gamma\mathcal{B}\zeta\right]\\
&\qquad +2\mu_k^2\eta\sqrt{2\eta^2+2\mathbb{E}[L_{f,S}^2]\mathcal{A}^2}+2\mu_k^2\eta\mathcal{A}\sqrt{\mathbb{E}[L_{f,S}^2]}\\
&=\mathbb{E}\left[\theta_S^2(\mu_k)\right]\mathbb{E}[\|x^k-x^*\|^2]+\mu_k^2\mathcal{D}.
\end{aligned}$$

For simplicity, we use further in the proof the following notations: $r_k=\sqrt{\mathbb{E}[\|x^k-x^*\|^2]}$ and $\theta_k=\mathbb{E}[\theta_S^2(\mu_k)]$. Then, through the nonexpansiveness property of the projection operator, the previous inequality turns into:

$$\begin{aligned}
r_{k+1}^2 &\leq \mathbb{E}[\|z_{\mu_k}(x^k;S)-x^*\|^2]\leq \theta_k r_k^2+\mu_k^2\mathcal{D}\\
&\leq r_0^2\prod_{i=0}^k\theta_i+\mathcal{D}\sum_{i=0}^k\left(\prod_{j=i+1}^k\theta_j\right)\mu_i^2. \qquad (38)
\end{aligned}$$

To further refine the right hand side in (38), we first notice from Lemma 20 that we have $\prod_{i=0}^k\theta_i\leq\theta_0^{\varphi_{1-\gamma}(k+1)}$. Then, from (38) we can derive different upper bounds for the two cases of the parameter γ: $\gamma<1$ and $\gamma=1$.

Case (i) $\gamma<1$. From Lemma 20, we derive an upper approximation for the second term in the right hand side of (38). Therefore, if we let $m=\lceil\frac{k}{2}\rceil$ we obtain:

$$\begin{aligned}
\sum_{i=0}^k\mu_i^2\left(\prod_{j=i+1}^k\theta_j\right) &= \sum_{i=0}^m\mu_i^2\left(\prod_{j=i+1}^k\theta_j\right)+\sum_{i=m+1}^k\mu_i^2\left(\prod_{j=i+1}^k\theta_j\right)\\
&\overset{\text{Lemma 20}}{\leq} \sum_{i=0}^m\mu_i^2\theta_0^{\varphi_{1-\gamma}(k+1)-\varphi_{1-\gamma}(i+1)}+\mu_{m+1}\sum_{i=m+1}^k\mu_i\left(\prod_{j=i+1}^k\theta_j\right)\\
&\leq \theta_0^{\varphi_{1-\gamma}(k+1)-\varphi_{1-\gamma}(m+1)}\sum_{i=0}^m\mu_i^2+\mu_{m+1}\sum_{i=m+1}^k\mu_i\left(\prod_{j=i+1}^k\theta_j\right)\\
&= \theta_0^{\varphi_{1-\gamma}(k+1)-\varphi_{1-\gamma}(m+1)}\sum_{i=0}^m\mu_i^2+\mu_{m+1}\sum_{i=m+1}^k\frac{\mu_i}{1-\theta_i}(1-\theta_i)\left(\prod_{j=i+1}^k\theta_j\right). \qquad (39)
\end{aligned}$$

We will further refine the right hand side of (39) by noticing the following two facts. First, the constant $\frac{\mu_i}{1-\theta_i}$ can be upper bounded by:

$$\frac{\mu_i}{1-\theta_i}=\frac{1}{\mathbb{E}\left[\frac{\sigma_S}{(1+\mu_i\sigma_S)^2}+\frac{\sigma_S}{1+\mu_i\sigma_S}\right]}\leq\frac{\mu_{i-1}}{1-\theta_{i-1}}\leq\cdots\leq\frac{\mu_0}{1-\theta_0}.$$

Second, the sum of products is upper bounded as:

$$\sum_{i=m+1}^k(1-\theta_i)\left(\prod_{j=i+1}^k\theta_j\right)=\sum_{i=m+1}^k\left(\prod_{j=i+1}^k\theta_j-\prod_{j=i}^k\theta_j\right)=1-\prod_{j=m+1}^k\theta_j\leq 1.$$

By using the last two inequalities into (39), we have:

$$\sum_{i=0}^k\mu_i^2\left(\prod_{j=i+1}^k\theta_j\right)\leq\theta_0^{\varphi_{1-\gamma}(k+1)-\varphi_{1-\gamma}(m+1)}\sum_{i=0}^m\mu_i^2+\mu_{m+1}\frac{\mu_0}{1-\theta_0}. \qquad (40)$$

Since $\sum_{i=0}^m\mu_i^2\leq\mu_0^2(\varphi_{1-2\gamma}(m)+2)\leq\mu_0^2(\varphi_{1-2\gamma}(m)+2)\leq\mu_0^2[\varphi_{1-2\gamma}\left(\frac{k}{2}+1\right)+2]$ and using (40) into (38), we obtain the above result.

Case (ii) $\gamma=1$. In this case we have:

$$\begin{aligned}
&\sum_{i=1}^k\mu_i^2\left(\prod_{j=i+1}^k\theta_j\right)\overset{\text{Lemma 20}}{\leq}\sum_{i=1}^k\mu_i^2\theta_0^{\varphi_0(k+1)-\varphi_0(i+1)}\\
&=\sum_{i=1}^k\frac{\mu_1^2}{i^2}\theta_0^{\ln\frac{k+1}{i+1}}=\sum_{i=1}^k\frac{\mu_1^2}{i^2}\left(\frac{k+1}{i+1}\right)^{\ln\theta_0}\leq\left(\frac{1}{k}\right)^{\ln\left(\frac{1}{\theta_0}\right)}\sum_{i=1}^k\frac{\mu_1^2}{i^{2-\ln\frac{1}{\theta_0}}}\\
&\leq\left(\frac{1}{k}\right)^{\ln\left(\frac{1}{\theta_0}\right)}\mu_0^2\varphi_{\ln\frac{1}{\theta_0}-1}(k).
\end{aligned}$$

Therefore, the variation of θ_0 leads to the following cases:

$$\sum_{i=1}^{k}\mu_i^2\left(\prod_{j=i+1}^{k}\theta_j\right)\leq\begin{cases}\frac{\mu_0^2}{k\left(\ln\left(\frac{1}{\theta_0}\right)-1\right)} & \text{if } \theta_0<\frac{1}{e}\\ \frac{\mu_0^2\ln k}{k} & \text{if } \theta_0=\frac{1}{e}\\ \left(\frac{1}{k}\right)^{\ln\left(\frac{1}{\theta_0}\right)}\frac{\mu_0^2}{1-\ln\left(\frac{1}{\theta_0}\right)} & \text{if } \theta_0>\frac{1}{e},\end{cases}$$

which leads to the second part of the result. ■

Proof of Lemma 17:

Proof The proof follows similar lines with the one of Lemma 30. Therefore, by using notations: $\alpha=\sqrt{1-\frac{1}{\zeta}}$ and $d_{k,t}=\sqrt{\mathbb{E}[\mathrm{dist}_X^2(x^{k,t})]}$ results in:

$$\begin{aligned}d_{k+1,t}&\leq\alpha d_{k,t}+\alpha\mu_t\mathcal{B}\leq\alpha^{k+1}d_{0,t}+\mu_t\mathcal{B}\sum_{i=1}^{k+1}\alpha^i\\&\leq\alpha^{k+1}d_{0,t}+\mu_t\mathcal{B}\frac{\alpha}{1-\alpha}.\end{aligned}$$

By setting $k=K_t-1$, then the last inequality implies:

$$\begin{aligned}d_{K_t,t}&\leq\alpha^{K_t}d_{K_{t-1},t-1}+\mu_t\mathcal{B}\frac{\alpha}{1-\alpha}\\&\leq\alpha^{\sum_{i=1}^{t}K_i}d_{0,0}+\mathcal{B}\frac{\alpha}{1-\alpha}\sum_{j=0}^{t-1}\alpha^{\sum_{i=t-j+1}^{t}K_i}\mu_{t-j}.\end{aligned}$$

Now set $m=\lceil\frac{t}{2}\rceil$. By dividing the sum from the right side of (32) in two parts, by taking into account that $\{\mu_t\}_{t\geq0}$ is nonincreasing and $\{K_t\}_{t\geq0}$ is nondecreasing, then results in:

$$\begin{aligned}\sum_{j=0}^{t-1}\alpha^{\sum_{i=t-j+1}^{t}K_i}\mu_{t-j}&=\sum_{j=0}^{m}\alpha^{\sum_{i=t-j+1}^{t}K_i}\mu_{t-j}+\sum_{j=m+1}^{t-1}\alpha^{\sum_{i=t-j+1}^{t}K_i}\mu_{t-j}\\&\leq\mu_{t-m}\sum_{j=0}^{m}\alpha^{\sum_{i=t-j+1}^{t}K_i}+\mu_0\alpha^{K_t}\sum_{j=m+1}^{t-1}\alpha^{\sum_{i=t-j+1}^{t-1}K_i}\\&\leq\mu_{t-m}\frac{1-\alpha^{m+1}}{1-\alpha}+\mu_0\alpha^{\sum_{i=t-m}^{t}K_i}\frac{1-\alpha^{t-m+2}}{1-\alpha}\\&\leq\frac{\mu_{t-m}}{1-\alpha}+\frac{\mu_0\alpha^{\sum_{i=t-m}^{t}K_i}}{1-\alpha}.\end{aligned}$$

By using the last inequality into (32) and using the bound $\frac{\alpha}{1-\alpha}\leq2\zeta$, then these facts imply the statement of the lemma. ■

Proof of Theorem 18:

Proof First notice that from $e^x\geq1+x$ for all $x\geq0$, we have $\left(1-\frac{1}{\zeta}\right)^{\sum_{i=1}^{t}\frac{K_i}{2}}\leq\left(1-\frac{1}{\zeta}\right)^{\frac{K_t}{2}}\leq\frac{2}{K_t\ln(\zeta/\zeta-1)}$ and $\left(1-\frac{1}{\zeta}\right)^{\sum_{i=t-\lceil\frac{t}{2}\rceil}^{t}\frac{K_i}{2}}\leq\left(1-\frac{1}{\zeta}\right)^{\frac{K_t}{2}}\leq\frac{2}{K_t\ln(\zeta/\zeta-1)}$, which imply that Lemma 2 becomes

$$\sqrt{\mathbb{E}[\mathrm{dist}_X^2(x^{K_t,t})]}\leq\mu_t\frac{2\mathrm{dist}_X(x^{0,0})}{\mu_0\ln(\zeta/\zeta-1)}+\mu_t\frac{4\zeta^2\mathcal{B}}{\ln(\zeta/\zeta-1)}+2\mu_{t-\lceil\frac{t}{2}\rceil}\zeta^2\mathcal{B}.\tag{41}$$

It can be seen that by combining (41) with a similar argument as in Theorem 14 we obtain a similar descent as (38). Therefore, let $k\geq0$ and $x^{k,t}$ be the kth iterate from the tth epoch. Then, by denoting $r_{k,t}^2=\mathbb{E}[\|x^{k,t}-x^*\|^2]$, results in:

$$r_{k+1,t}^2\leq\mathbb{E}[\theta_S(\mu_t)^2]r_{k,t}^2+\mu_t^2\mathcal{D}_r.$$

Now taking $k=K_t$ results in:

$$r_{0,t+1}^2=r_{K_t,t}^2\leq r_{0,t}^2\theta_t^{K_t}+\mathcal{D}_r\mu_t^2\sum_{i=0}^{k}\theta_t^i\leq r_{0,t}^2\theta_t^{K_t}+\frac{\mathcal{D}_r\mu_t^2}{1-\theta_t}.\tag{42}$$

Recalling that we chose $\mu_t=\frac{\mu_0}{t^\gamma}$ and $K_t=\lceil t^\gamma\rceil$, then (7) leads to:

$$\theta_t^{K_t}\leq\left(\mathbb{E}\left[\frac{1}{(1+\mu_0\sigma_{f,S})^2}\right]\right)^{\frac{K_t}{t^\gamma}}\leq\theta_0.$$

Therefore, (42) leads to:

$$r_{0,t+1}\overset{(42)}{\leq}\theta_0r_{0,t}^2+\frac{\mathcal{D}_r\mu_t^2}{1-\theta_t}\leq\theta_0^tr_{0,1}^2+\mathcal{D}_r\sum_{i=1}^{t}\frac{\mu_i^2\theta_0^{t-i}}{1-\theta_i}.\tag{43}$$

Note that $\frac{\mu_i^2}{1-\theta_i}$ is nonincreasing in i. Then, if we fix $m=\lceil\frac{t}{2}\rceil$, then the sum $\sum_{i=1}^{t}\frac{\mu_i^2\theta_0^{t-i}}{1-\theta_i}$ can be bounded as follows:

$$\begin{aligned}\sum_{i=1}^{t}\frac{\mu_i^2\theta_0^{t-i}}{1-\theta_i}&\leq\theta_0^m\sum_{i=1}^{m}\frac{\mu_i^2}{1-\theta_i}+\sum_{i=m}^{t}\frac{\mu_i^2\theta_0^{t-i}}{1-\theta_i}\\&\leq\theta_0^m\sum_{i=1}^{m}\frac{\mu_i^2}{1-\theta_i}+\frac{\mu_m^2}{1-\theta_m}\sum_{i=1}^{t-m}\theta_0^i\\&\leq\frac{\theta_0^m\mu_1}{1-\theta_0}\left(\sum_{i=1}^{m}\mu_i\right)+\frac{\mu_m^2}{(1-\theta_m)(1-\theta_0)}\\&\leq\frac{\theta_0^m\mu_1}{1-\theta_0}\left(\sum_{i=1}^{m}\mu_i\right)+\mu_m\frac{\mu_1}{(1-\theta_0)^2}.\end{aligned}\tag{44}$$

Taking into account that $\sum_{i=1}^{m}\mu_i \leq \int_1^m \frac{1}{s^\gamma}ds \leq \frac{2^{\gamma-1}}{(1-\gamma)t^{\gamma-1}}$ and that $\theta_0^m \leq \frac{1}{1+\frac{t}{2}\ln\frac{1}{\theta_0}}$, the previous relation (44) implies:

$$\sum_{i=1}^{t}\frac{\mu_i^2\theta_0^{t-i}}{1-\theta_i} \leq \left(\frac{2}{t}\right)^{\gamma}\left[\frac{1}{2(1-\gamma)\ln 1/\sqrt{\theta_0}}+\frac{\mu_1^2}{(1-\theta_0)^2}\right]. \tag{45}$$

By using this bound in relation (44), then in order to obtain $r_{0,t+1}^2 \leq \epsilon$ it is sufficient that the number of epochs t to satisfy:

$$t \geq \max\left\{\ln\left(\frac{2r_{0,0}^2}{\epsilon}\right)\frac{1}{\ln(1/\theta_0)}, \left(\frac{2^{\gamma+1}\mathcal{D}_r}{\epsilon}\mathcal{C}\right)^{1/\gamma}\right\}. \tag{46}$$

Finally, the total number of SPP iterations performed by RSPP algorithm satisfies:

$$\begin{aligned}\sum_{i=1}^{t}K_t &\geq \sum_{i=1}^{t}i^\gamma \geq \int_0^t s^\gamma ds = \frac{t^{1+\gamma}}{1+\gamma}\\ &\geq \frac{1}{1+\gamma}\max\left\{\ln\left(\frac{2r_{0,0}^2}{\epsilon}\right)^{1+\gamma}\frac{1}{\ln(1/\theta_0)^{1+\gamma}}, \left(\frac{2^{\gamma+1}\mathcal{D}_r}{\epsilon}\mathcal{C}\right)^{1+\frac{1}{\gamma}}\right\},\end{aligned}$$

which proves the statement of the theorem. ■

9. Acknowledgments

The research leading to these results has received funding from the Executive Agency for Higher Education, Research and Innovation Funding (UEFISCDI), Romania: PNIII-P4-PCE-2016-0731, project ScaleFreeNet, no. 39/2017.

References

Y.F. Atchade, G. Fort and E. Moulines, *On perturbed proximal gradient algorithms*, Journal of Machine Learning Research, 18(1):310–342, 2014.

F. Bach, *Self-concordant analysis for logistic regression*, Electronic Journal of Statistics, 4:384–414, 2010.

F. Bach, G. Lanckriet and M. Jordan, *Multiple kernel learning, conic duality, and the SMO algorithm*, International Conference on Machine Learning (ICML), 2004.

D.P. Bertsekas, *Incremental proximal methods for large scale convex optimization*, Mathematical Programming, 129(2):163–195, 2011.

P. Bianchi, *Ergodic convergence of a stochastic proximal point algorithm*, SIAM Journal on Optimization, 26(4):2235–2260, 2016.

D. Blatt and A.O. Hero, *III: Energy based sensor network source localization via projection onto convex sets (POCS)*, IEEE Transactions on Signal Processing, 54(9):3614–3619, 2006.

L. Bottou, F.E. Curtis and J. Nocedal, *Optimization methods for large-scale machine learning*, arXiv:1606.04838, 2016.

J. Brodie, I. Daubechies, C. de Mol, D. Giannone and I. Loris, *Sparse and stable Markowitz portfolios*, Proc. Natl. Acad. Sci, 106:12267–12272, 2009.

P.S. Bullen, *Handbook of means and their inequalities*, Kluwer Academic Publisher, Dordercht, 2003.

Y. Censor, W. Chen, P. L. Combettes, R. Davidi and G. T. Herman, *On the effectiveness of projection methods for convex feasibility problems with linear inequality constraints*, Computational Optimization and Applications, 51(3):1065–1088, 2012.

P.L. Combettes and J.C. Pesquet, *Stochastic approximations and perturbations in forward-backward splitting for monotone operators*, Pure and Applied Functional Analysis, 1(1):13–37, 2016.

A. Defazio, F. Bach and S. Lacoste-Julien, *SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives*, Advances in Neural Information Processing Systems (NIPS), 1646–1654, 2014.

J. Duchi and Y. Singer, *Efficient online and batch learning using forward backward splitting*, Journal of Machine Learning Research, 10:2899–2934, 2009.

O. Guler, *On the convergence of the proximal point algorithm for convex minimization*, SIAM Journal on Control and Optimization, 29(2):403–419, 1991.

R. Johnson and T. Zhang, *Accelerating stochastic gradient descent using predictive variance reduction*, Advances in Neural Information Processing Systems (NIPS), 315–323, 2013.

A. Karimi and C. Kammer, *A data-driven approach to robust control of multivariable systems by convex optimization*, Automatica, 85:227–233, 2017.

J. Koshal, A. Nedic and U.V. Shanbhag, *Regularized iterative stochastic approximation methods for stochastic variational inequality problems*, IEEE Transactions on Automatic Control, 58(3):594–609, 2013.

A. Mokhtari and A. Ribeiro, *RES: Regularized stochastic BFGS algorithm*, IEEE Transactions on Signal Processing, 62(23):6089–6104, 2014.

E. Moulines and F.R. Bach, *Non-asymptotic analysis of stochastic approximation algorithms for machine learning*, Advances in Neural Information Processing Systems (NIPS), 451–459, 2011.

I. Necoara, *Random algorithms for convex minimization over intersection of simple sets*, submitted to European Control Conference (ECC18), 2017.

I. Necoara, V. Nedelcu and I. Dumitrache, *Parallel and distributed optimization methods for estimation and control in networks*, Journal of Process Control, 21(5):756–766, 2011.

I. Necoara, Yu. Nesterov and F. Glineur, *Linear convergence of first order methods for non-strongly convex optimization*, Mathematical Programming, in press:135, 2017a.

I. Necoara, P. Richtarik, and A. Patrascu, *Randomized projection methods for convex feasibility problems*, Technical Report, UPB:1–30, 2017b.

A. Nedic, *Random algorithms for convex minimization problems*. Mathematical Programming, 129(2):225253, 2011.

A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, *Robust stochastic approximation approach to stochastic programming*, SIAM Journal on Optimization, 19(4):15741609, 2009.

Yu. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Kluwer Academic Publisher, Boston, 2004.

F. Niu, B. Recht, C. Re, and S. J. Wright, *HOGWILD!: A lock-free approach to parallelizing stochastic gradient descent*, In Advances in Neural Information Processing Systems (NIPS), pages 693–701, 2011.

J. C. Platt, *Fast training of support vector machines using sequential minimal optimization*, In Advances in Kernel Methods - Support Vector Learning, Cambridge, MA, 1998.

B. T. Polyak and A. B. Juditsky, *Acceleration of stochastic approximation by averaging*, SIAM Journal on Control and Optimization, 30(4):838–855, 1992.

R.T. Rockafellar and R.J.-B. Wets, *Variational Analysis*, Springer-Verlag, Berlin Heidelberg, 1998.

L. Rosasco, S. Villa, and B. C. Vu, *Convergence of stochastic proximal gradient algorithm*, arXiv:1403.5074, 2014.

L. Rosasco, S. Villa, and B. C. Vu, *A first-order stochastic primal-dual algorithm with correction step*, Numerical Functional Analysis and Optimization, 38(5):602–626, 2017.

N. L. Roux, M. Schmidt, and F. Bach, *A stochastic gradient method with an exponential convergence rate for finite training sets*, In Advances in Neural Information Processing Systems (NIPS), 2672–2680, 2012.

E. Ryu and S. Boyd, *Stochastic proximal iteration: A non-asymptotic improvement upon stochastic gradient descent*, www.math.ucla.edu/eryu/papers/spi.pdf, 2016.

S. Shalev-Shwartz and T. Zhang, *Stochastic dual coordinate ascent methods for regularized loss*, Journal of Machine Learning Research, 14(1):567–599, 2013.

S. Sonnenburg, G. Ratscha, C. Schafer, and B. Scholkopf, *Large scale multiple kernel learning*, Journal of Machine Learning Research, 7:15311565, 2006.

P. Toulis, D. Tran, and E. M. Airoldi, *Towards stability and optimality in stochastic gradient descent*, In International Conference on Artificial Intelligence and Statistics (AISTATS), 1290–1298, 2016.

N. Denizcan Vanli, M. Gurbuzbalaban, and A. Ozdaglar, *Global convergence rate of proximal incremental aggregated gradient methods*, arXiv:1608.01713, 2016.

W. Xu, *Towards optimal one pass large scale learning with averaged stochastic gradient descent*, CoRR, abs/1107.2490, 2011.

T. Yang and Q. Lin, *Stochastic subgradient methods with linear convergence for polyhedral convex optimization*, arXiv:1510.01444, 2016.

A. Yurtsever, B. C. Vu, and V. Cevher, *Stochastic three-composite convex minimization*, In Advances in Neural Information Processing Systems (NIPS), 4322–4330, 2016.

Journal of Machine Learning Research 18 (2018) 1-38 Submitted 4/16; Revised 11/17; Published 4/18

Simple, Robust and Optimal Ranking from Pairwise Comparisons

Nihar B. Shah nihars@cs.cmu.edu
Machine Learning Department and Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Martin J. Wainwright wainwrig@berkeley.edu
Department of Electrical Engineering and Computer Sciences and Department of Statistics
University of California
Berkeley, CA 94720, USA

Editor: Sujay Sanghavi

Abstract

We consider data in the form of pairwise comparisons of n items, with the goal of identifying the top k items for some value of $k < n$, or alternatively, recovering a ranking of all the items. We analyze the Borda counting algorithm that ranks the items in order of the number of pairwise comparisons won, and show it has three attractive features: (a) it is an optimal method achieving the information-theoretic limits up to constant factors; (b) it is robust in that its optimality holds without imposing conditions on the underlying matrix of pairwise-comparison probabilities, in contrast to some prior work that applies only to the BTL parametric model; and (c) its computational efficiency leads to speed-ups of several orders of magnitude. We address the problem of exact recovery, and for the top-k recovery problem we also extend our results to obtain sharp guarantees for approximate recovery under the Hamming distortion metric, and more generally, to any arbitrary error requirement that satisfies a simple and natural monotonicity condition. In doing so, we introduce a general framework that allows us to treat a variety of problems in the literature in an unified manner.

Keywords: Pairwise comparisons, Ranking, Set recovery, Approximate recovery, Borda count, Permutation-based models, Occam's razor

1. Introduction

Ranking problems involve a collection of n items, and some unknown underlying total ordering of these items. In many applications, one may observe noisy comparisons between various pairs of items. Examples include matches between football teams in tournament play; consumer's preference ratings in marketing; and certain types of voting systems in politics. Given a set of such noisy comparisons between items, it is often of interest to find the true underlying ordering of all n items, or more generally, given some given positive integer $k \leq n$, to find the subset of k most highly rated items. These two problems are the focus of this paper.

There is a substantial literature on the problem of finding approximate rankings based on noisy pairwise comparisons. A number of papers (e.g., Kenyon-Mathieu and Schudy, 2007; Braverman and Mossel, 2008; Eriksson, 2013) consider models in which the probability of a pairwise comparison agreeing with the underlying order is identical across all pairs. These results break down when, for one or more pairs, the probability of agreeing with the underlying ranking is close to or exactly equal to $\frac{1}{2}$. Another set of papers (Hunter, 2004; Negahban et al., 2012; Hajek et al., 2014; Soufiani et al., 2014; Shah et al., 2016a) work using parametric models of pairwise comparisons, and address the problem of recovering the parameters associated to every individual item. A more recent line of work (Chatterjee, 2014; Shah et al., 2017a, 2016d) studies a more general class of models based on the notion of strong stochastic transitivity (SST), and derives conditions on recovering the pairwise comparison probabilities themselves. However, it remains unclear whether or not these results can directly extend to tight bounds for the problem of recovery of the top k items. Another line of work (Jagabathula and Shah, 2008; Mitliagkas et al., 2011; Ammar and Shah, 2012; Ding et al., 2015) focuses on mixture models, in which every pairwise comparison is associated to a certain individual making the comparison, and it is assumed that the preferences across individuals can be described by a low-dimensional model.

Most related to our work are the papers by Wauthier et al. (2013); Rajkumar and Agarwal (2014); Rajkumar et al. (2015), and Chen and Suh (2015), which we briefly discuss here and in a more detailed manner in the sequel. Wauthier et al. (2013) analyze a weighted counting algorithm to recover approximate rankings; their analysis applies to a specific model in which the pairwise comparison between any pair of items remains faithful to their relative positions in the true ranking with a probability common across all pairs. They consider recovery of an approximate ranking under Kendall's tau and maximum displacement metrics, but do not provide results on exact recovery. As the analysis of this paper shows, their bounds are quite loose: more precisely, their results are tight only when there are a total of at least $\Theta(n^2)$ comparisons. Two other papers (Rajkumar and Agarwal, 2014; Rajkumar et al., 2015) consider ranking under several models and several metrics. In the part that is common with our setting, they show that the counting algorithm is consistent in terms of recovering the full ranking, which automatically implies consistency in exactly recovering the top k items. They obtain upper bounds on the sample complexity in terms of a separation threshold that is identical to a parameter Δ_k defined subsequently in this paper (see Section 3). However, as our analysis shows, their bounds are loose by at least an order of magnitude. They also assume a certain high-SNR condition on the probabilities, an assumption that is not imposed in our analysis.

Finally, in very recent work on this problem, Chen and Suh (2015) proposed an algorithm called the Spectral MLE for exact recovery of the top k items. They showed that, if the pairwise observations are assumed to drawn according to the Bradley-Terry-Luce (BTL) parametric model (Bradley and Terry, 1952; Luce, 1959), the Spectral MLE algorithm recovers the k items correctly with high probability under certain regularity conditions. In addition, they also show, via matching lower bounds, that their regularity conditions are tight up to constant factors. While these guarantees are attractive, it is natural to ask how such an algorithm behaves when the data is *not* drawn from the BTL model. In real-world instances of pairwise ranking data, it is often found that parametric models, such as the BTL model and its variants, fail to provide accurate fits (for instance, see the papers Davidson and Marschak, 1959; McLaughlin and Luce, 1965; Tversky, 1972; Ballinger and Wilcox, 1997 and references therein).

With this context, the main contribution of this paper is to analyze a classical counting-based method for ranking, often called the Borda count method (de Borda, 1781), and to show that it is optimal and robust. Our analysis does not require that the data-generating mechanism follow either the BTL or other parametric assumptions, nor other regularity conditions such as stochastic transitivity. We show that the Borda counting algorithm has the following properties:

- Simplicity: The algorithm is simple, as it just orders the items by the number of pairwise comparisons won. As we will subsequently see, the execution time of this counting algorithm is several orders of magnitude lower as compared to prior work on ranking from noisy pairwise comparisons.
- Optimality: We derive conditions under which the counting algorithm achieves the stated goals, and by means of matching information-theoretic lower bounds, show that these conditions are tight.
- Robustness: The guarantees that we prove do not require any assumptions on the pairwise-comparison probabilities, and the counting algorithm performs well for various classes of data sets. In contrast, we find that the spectral MLE algorithm performs poorly when the data is not drawn from the BTL model.

In doing so, we consider three different instantiations of the problem of set-based recovery:

(i) Recovering the top k items perfectly;

(ii) Recovering the top k items allowing for a certain Hamming error tolerance; and

(iii) A more general recovery problem for set families that satisfy a natural "set-monotonicity" condition. In order to tackle this general problem, we introduce a general framework that allows us to treat a variety of problems in the literature in an unified manner.

The remainder of this paper is organized as follows. We begin in Section 2 with background and a more precise formulation of the problem. Section 3 presents our main theoretical results on top-k recovery under various requirements. Section 4 provides the results of experiments on both simulated and real-world data sets. We provide all proofs in Section 5. The paper concludes with a discussion in Section 6.

2. Background and problem formulation

In this section, we provide a more formal statement of the problem along with background on various types of ranking models.

2.1 Problem statement

Given an integer $n \geq 2$, we consider a collection of n items, indexed by the set $[n] := \{1, \ldots, n\}$. For each pair $i \neq j$, we let M_{ij} denote the probability that item i wins the comparison with item j. We assume that each comparison necessarily results in one winner, meaning that

$$M_{ij} + M_{ji} = 1, \qquad \text{and} \quad M_{ii} = \frac{1}{2}, \tag{1}$$

where we set the diagonal as $\frac{1}{2}$ for concreteness.

For any item $i \in [n]$, we define an associated score τ_i as

$$\tau_i(M) := \frac{1}{n} \sum_{j=1}^{n} M_{ij}. \tag{2}$$

In words, the score $\tau_i(M)$ of any item $i \in [n]$ corresponds to the probability that item i beats an item chosen uniformly at random from all n items. In the sequel, we will use the shorthand τ_i for the score of any item i, and drop the dependence on M from the notation wherever the value of M is clear from context.

Given a set of noisy pairwise comparisons, our goals are (a) to recover the k items with the maximum values of their scores; and (b) to recover the full ordering of all the items as defined by the score vector. The notion of ranking items via their scores (2) generalizes the explicit rankings under popular models in the literature. Indeed, as we discuss shortly, most models of pairwise comparisons considered in the literature either implicitly or explicitly assume that the items are ranked according to their scores. Note that neither the scores $\{\tau_i\}_{i \in [n]}$ nor the matrix $M := \{M_{ij}\}_{i,j \in [n]}$ of probabilities are assumed to be known.

More concretely, we consider a random-design observation model defined as follows. Each pair is associated with a random number of noisy comparisons, following a binomial distribution with parameters (r, p), where $r \geq 1$ is the number of trials and $p \in (0, 1]$ is the probability of making a comparison on any given trial. Thus, each pair (i, j) is associated with a binomial random variable with parameters (r, p) that governs the number of comparisons between the pair of items. We assume that the observation sequences for different pairs are independent. Note that in the special case $p = 1$, this random binomial model reduces to the case in which we observe exactly r observations of each pair; in the special case $r = 1$, the set of pairs compared form an (n, p) Erdős-Rényi random graph.

In this paper, we begin in Section 3.1 by analyzing the problem of exact recovery. More precisely, for a given matrix M of pairwise probabilities, suppose that we let $\mathcal{S}_k^*$ denote the (unknown) set of k items with the largest values of their respective scores, assumed to be unique for concreteness.

Given noisy observations specified by the pairwise probabilities M, our goal is to establish conditions under which there exists some algorithm $\widehat{\mathcal{S}}_k$ that identifies k items based on the outcomes of various comparisons such that the probability $\mathbb{P}_M(\widehat{\mathcal{S}}_k = \mathcal{S}_k^*)$ is very close to one. In the case of recovering the full ranking, our goal is to identify conditions which ensure that the probability $\mathbb{P}_M\big(\bigcap_{k \in [n]} (\widehat{\mathcal{S}}_k = \mathcal{S}_k^*)\big)$ is close to one.

In Section 3.2, we consider the problem of recovering a set of k items that approximates $\mathcal{S}_k^*$ with a minimal Hamming error. For any two subsets of the set $[n]$, we define their Hamming distance D_{H}, also referred to as their Hamming error, to be the number of items that belong to exactly one of the two sets—that is

$$D_{\mathrm{H}}(A, B) = \mathrm{card}\Big(\{A \cup B\} \backslash \{A \cap B\}\Big). \tag{3}$$

For a given user-defined tolerance parameter $h \geq 0$, we derive conditions that ensure that $D_{\mathrm{H}}(\widehat{\mathcal{S}}_k, \mathcal{S}_k^*) \leq 2h$ with high probability.

Finally, we generalize our results to the problem of satisfying any a general class of requirements on set families. These requirement are specified in terms of which k-sized

subsets of the items are allowed, and is required to satisfy only one natural condition, that of set-monotonicity, meaning that replacing an item in an allowed set with a higher rank item should also be allowed. See Section 3.3 for more details on this general framework.

2.2 A range of pairwise comparison models

Our work makes minimal assumptions on the the pairwise comparison probabilities. Our model is based on a "permutation-based" approach, and is described below (see Shah et al., 2017a, 2016d,c, 2017b for other uses of permutation-based models and Shah, 2017, Chapter 1 and Part 1 for a general treatment). In order to put our work in context of the literature, we also briefly review some standard models used for pairwise comparison data – all of these models form special cases of our general model.

Our model: We assume that any requirements or metrics for recovery of a partial or total order of the items are governed by the scores of the items defined in equation (2). In other words, any item i is considered as ranked higher than any item j when their scores satisfy $\tau_i > \tau_j$. We make no other assumptions on the probabilities $\{M_{ij}\}_{i,j\in[n]}$. In what follows, we show that several other popular classes of models arise as special cases of our model.

Parametric models: A broad class of parametric models, including the Bradley-Terry-Luce (BTL) model (Bradley and Terry, 1952; Luce, 1959) as a special case, are based on assuming the existence of "quality" parameter $w_i \in \mathbb{R}$ for each item $i \in [n]$, and requiring that the probability of an item beating another is a specific function of the difference between their values. In the BTL model, the probability M_{ij} that i beats j is given by the logistic model

$$M_{ij} = \frac{1}{1 + e^{-(w_i - w_j)}}. \tag{4a}$$

More generally, parametric models assume that the pairwise comparison probabilities take the form

$$M_{ij} = F(w_i - w_j), \tag{4b}$$

where $F : \mathbb{R} \to [0, 1]$ is some strictly increasing cumulative distribution function. The function F is typically assumed to be known. By construction, any parametric model has the following property: if $w_i > w_j$ for some pair of items (i, j), then we are also guaranteed that $M_{i\ell} > M_{j\ell}$ for every item ℓ. As a consequence, we are guaranteed that $\tau_i > \tau_j$, which implies that ordering of the items in terms of their quality vector $w \in \mathbb{R}^n$ is identical to their ordering in terms of the score vector $\tau \in \mathbb{R}^n$. Consequently, if the data is actually drawn from a parametric model, then recovering the top k items according to their scores is the same as recovering the top k items according their respective quality parameters.

Strong Stochastic Transitivity (SST) class: The class of strong stochastic transitivity (SST) models is a superset of parametric models (Shah et al., 2017a). It does not assume the existence of a quality vector, nor does it assume any specific form of the probabilities as in equation (4a). Instead, the SST class is defined by assuming the existence of a total ordering of the n items, and imposing the inequality constraints $M_{i\ell} \geq M_{j\ell}$ for every pair of items (i, j) in which i is ranked above j in the ordering, and every item ℓ. One can verify

that an ordering by the scores $\{\tau_i\}_{i\in[n]}$ of the items lead to an ordering of the items that is consistent with that defined by the SST class.

Thus, we see that in a broad class of models for pairwise ranking, the total ordering defined by the score vector (2) coincides with the underlying ordering used to define the models. In this paper, we analyze the performance of a counting algorithm, essentially without imposing any modeling conditions on the family of pairwise probabilities. The next three sections establish theoretical guarantees on the recovery of the top k items under various requirements.

2.3 Borda counting algorithm

The analysis of this paper focuses on a simple counting-based algorithm, often called the Borda count method (de Borda, 1781). We employ this method here for the setting of pairwise comparisons, noting that the Borda count method more generally also supports comparisons between more than two items.

More precisely, for each distinct $i, j \in [n]$ and every integer $\ell \in [r]$, let $Y_{ij}^{\ell} \in \{-1, 0, +1\}$ represent the outcome of the ℓ^{th} comparison between the pair i and j, defined as

$$Y_{ij}^{\ell} = \begin{cases} 0 & \text{no comparison between } (i,j) \text{ in trial } \ell \\ +1 & \text{if comparison is made and item } i \text{ beats } j \\ -1 & \text{if comparison is made and item } j \text{ beats } i. \end{cases} \tag{5}$$

Note that this definition ensures that $Y_{ij}^{\ell} = -Y_{ji}^{\ell}$. For each $i \in [n]$, the quantity

$$N_i := \sum_{j\in[n]} \sum_{\ell\in[r]} \mathbf{1}\{Y_{ij}^{\ell} = 1\} \tag{6}$$

corresponds to the number of pairwise comparisons won by item i. Here we use $\mathbf{1}\{\cdot\}$ to denote the indicator function that takes the value 1 if its argument is true, and the value 0 otherwise. For each integer k, the vector $\{N_i\}_{i=1}^n$ of number of pairwise wins defines a k-sized subset

$$\widetilde{\mathcal{S}}_k = \Big\{ i \in [n] \mid N_i \text{ is among the } k \text{ highest number of pairwise wins} \Big\}, \tag{7}$$

corresponding to the set of k items with the largest values of N_i. In other words, the set $\widetilde{\mathcal{S}}_k$ corresponds to the rank statistics of the top k-items in the pairwise win ordering. (If there are any ties, we resolve them by choosing the indices with the smallest value of i.)

3. Main results

In this section, we present our main theoretical results on top-k recovery under the three settings described earlier. Note that the three settings are ordered in terms of increasing generality, with the advantage that the least general setting leads to the simplest form of theoretical claim. We also discuss optimal exact recovery of the full ranking.

3.1 Thresholds for exact recovery of the top k items

We begin with the goal of exactly recovering the k top-ranked items. As one might expect, the difficulty of this problem turns out to depend on the degree of separation between the top k items and the remaining items. More precisely, let us use (k) and $(k+1)$ to denote the indices of the items that are ranked k^{th} and $(k+1)^{th}$ respectively. With this notation, the *k-separation threshold* Δ_k is given by

$$\Delta_k(M) := \tau_{(k)}(M) - \tau_{(k+1)}(M) = \underbrace{\frac{1}{n}\sum_{i=1}^{n} M_{(k)i}}_{\text{Term (i)}} - \underbrace{\frac{1}{n}\sum_{i=1}^{n} M_{(k+1)i}}_{\text{Term (ii)}}. \tag{8}$$

In words, the quantity $\Delta_k(M)$ is the difference between (i) the probability of item (k) beating another item chosen uniformly at random, and (ii) the same probability for item $(k+1)$.

As shown by the following theorem, success or failure in recovering the top k entries is determined by the size of $\Delta_k(M)$ relative to the number of items n, observation probability p and number of repetitions r. In particular, consider the family of matrices

$$\mathcal{F}_k(\alpha; n, p, r) := \left\{ M \in [0,1]^{n \times n} \mid M + M^T = 11^T, \text{ and } \Delta_k(M) \geq \alpha \sqrt{\frac{\log n}{npr}} \right\}. \tag{9}$$

To simplify notation, we often adopt $\mathcal{F}_k(\alpha)$ as a convenient shorthand for this set, where its dependence on (n, p, r) should be understood implicitly.

With this notation, the achievable result in part (a) of the following theorem is based on the estimator that returns the set $\widetilde{\mathcal{S}}_k$ of the the k items defined by the number of pairwise comparisons won, as defined in equation (7). On the other hand, the lower bound in part (b) applies to *any estimator*, meaning any measurable function of the observations.

Theorem 1 *(a) Consider any $n \geq 2$, $r \geq 1$ and $p \in (0, 1]$. Then if $\alpha \geq 8$, the set $\widetilde{\mathcal{S}}_k$ of top k items (7) given by the Borda counting algorithm satisfies*

$$\sup_{M \in \mathcal{F}_k(\alpha)} \mathbb{P}_M\big[\widetilde{\mathcal{S}}_k \neq \mathcal{S}_k^*\big] \leq \frac{1}{n^{14}}. \tag{10a}$$

(b) Conversely, suppose that $n \geq 7$ and $p \geq \frac{\log n}{2nr}$. Then for any $\alpha \leq \frac{1}{7}$, the error probability of any *estimator $\widehat{\mathcal{S}}_k$ is lower bounded as*

$$\sup_{M \in \mathcal{F}_k(\alpha)} \mathbb{P}_M\big[\widehat{\mathcal{S}}_k \neq \mathcal{S}_k^*\big] \geq \frac{1}{7}. \tag{10b}$$

Remarks: First, it is important to note that the negative result in part (b) holds even if the supremum is further restricted to a particular parametric sub-class of $\mathcal{F}_k(\alpha)$, such as the pairwise comparison matrices generated by the BTL model, or by the SST model. The proof for the lower bound constructs a packing set of possible pairwise comparison probabilities

and then applies Fano's inequality. The construction ensures that every element of the packing set also lies in the parametric and SST models. The packing set is based on a generalization of a construction introduced by Chen and Suh (2015) for the BTL model, which we adapt to the general definition (8) of the separation threshold Δ_k.

Second, we note that in the regime $p < \frac{\log n}{2nr}$, standard results from random graph theory (Erdős and Rényi, 1960) can be used to show that there are at least $\sqrt{n}$ items (in expectation) that are never compared to any other item. Of course, estimating the rank is impossible in this pathological case, so we omit it from consideration.

Third, the two parts of the theorem in conjunction show that the counting algorithm is essentially optimal. The only room for improvement is in the difference between inequality $\alpha \geq 8$ in the achievable result, and $\alpha \leq \frac{1}{7}$ in the lower bound.

Theorem 1 can also be used to derive guarantees for recovery of other functions of the underlying ranking. Here we consider the problem of identifying the ranking of all n items, which we denote by the permutation π^*. In this case, we require that each of the separations $\{\Delta_j\}_{j=1}^{n-1}$ are suitably lower bounded: more precisely, we study models M that belong to the intersection $\cap_{j=1}^{n-1}\mathcal{F}_j(\alpha)$.

Theorem 2 *Consider any $n \geq 2$, $r \geq 1$ and $p \in (0, 1]$. Let $\widetilde{\pi}$ be the permutation of the items specified by the Borda counting algorithm in order of the number of pairwise comparisons won. Then for any $\alpha \geq 8$, we have*

$$\sup_{M \in \cap_{j=1}^{n-1}\mathcal{F}_j(\alpha)} \mathbb{P}_M\big[\widetilde{\pi} \neq \pi^*\big] \leq \frac{1}{n^{13}}.$$

Conversely, if $n \geq 9$, then the separation condition on $\{\Delta_j\}_{j=1}^{n-1}$ that defines the set $\cap_{j=1}^{n-1}\mathcal{F}_j(\alpha)$ is unimprovable beyond constant factors.

The upper bound of Theorem 2 follows from the equivalence of the correct recovery of the ranking with the recovery of the top k items for every value of $k \in [n]$. The proof of the lower bound requires a markedly different set of arguments; the proof does not follow from Theorem 1(b) since for any given value of k a condition of the form $\min_{j \in [n-1]} \Delta_j \leq \alpha$ in general does not imply $\Delta_k \leq \alpha$ which would otherwise be required to use Theorem 1(b).

Detailed comparison to related work: In the remainder of this subsection, we make a detailed comparison to the related works (Wauthier et al., 2013; Rajkumar and Agarwal, 2014; Rajkumar et al., 2015; Chen and Suh, 2015) that were briefly discussed in Section 1.

Wauthier et al. (2013) analyze a weighted counting algorithm for approximate recovery of rankings; they work under a model in which $M_{ij} = \frac{1}{2} + \gamma$ whenever item i is ranked above item j in an assumed underlying ordering. Here the parameter $\gamma \in (0, \frac{1}{2}]$ is independent of (i, j), and as a consequence, the best ranked item is assumed to be as likely to beat the worst item as it is to beat the second ranked item, for instance. They analyze approximate ranking under Kendall tau and maximum displacement metrics. In order to have a displacement upper bounded by by some $\delta > 0$, their bounds require the order of $\frac{n^5}{\delta^2\gamma^2}$ pairwise comparisons. In comparison, our model is more general in that we do not impose the γ-condition on the pairwise probabilities When specialized to the γ-model, the

quantities $\{\Delta_j\}_{j=1}^n$ in our analysis takes the form $\Delta_j = \frac{2\gamma}{n}$, and Theorem 2 shows that $\frac{n \log n}{\min_{j \in [n]} \Delta_j^2} = \frac{n^3 \log n}{\gamma^2}$ observations are sufficient to recover the exact total ordering. Thus, for any constant δ, Theorem 2 guarantees exact recovery with a sample complexity that is a multiplicative factor of order $\frac{n^2}{\log n}$ smaller than that established by Wauthier et al. (2013).

The two papers by Rajkumar and Agarwal (2014) and Rajkumar et al. (2015) consider ranking under several models and several metrics. For the subset of their models common with our setting—namely, Bradley-Terry-Luce and the so-called low noise models—they show that the counting algorithm is consistent in terms of recovering the full ranking or the top subset of items. The guarantees are obtained under a low-noise assumption: namely, that the probability of any item i beating j is at least $\frac{1}{2} + \gamma$ whenever item i is ranked higher than item j in an assumed underlying ordering. Their guarantees are based on a sample size of at least $\frac{\log n}{\gamma^2 \mu^2}$, where μ is a parameter lower bounded as $\mu \geq \frac{1}{n^2}$. Once again, our setting allows for the parameter γ to be arbitrarily close to zero, and furthermore as one can see from the discussion above, our bounds are much stronger. Moreover, while Rajkumar et al. focus on upper bounds alone, we also prove matching lower bounds on sample complexity showing that our results are unimprovable beyond constant factors. It should be noted that Rajkumar et al. also provide results for other types of ranking problems that lie outside the problem class treated in the current paper.

Most recently, Chen and Suh (2015) consider a random-design setting and show that if the pairwise observations are assumed to drawn according to the Bradley-Terry-Luce (BTL) parametric model (4a), then their proposed Spectral MLE algorithm recovers the k items correctly with high probability when a certain separation condition on the parameters $\{w_i\}_{i=1}^n$ of the BTL model is satisfied. Their random-design setting is similar to ours except that they first choose a set of pairs of items with each pair chosen with probability p, and then make r comparisons between the two items in every chosen pair. We believe our random-design setting is more natural; the two are identical when $r = 1$. Chen and Suh also show, via matching lower bounds, that the separation condition they derive for the BTL model is tight up to constant factors. In real-world instances of pairwise ranking data, it is often found that parametric models, such as the BTL model and its variants, fail to provide accurate fits (Davidson and Marschak, 1959; McLaughlin and Luce, 1965; Tversky, 1972; Ballinger and Wilcox, 1997). Our results make no such assumptions on the noise, and furthermore, our notion of the ordering of the items in terms of their scores (2) strictly generalizes the notion of the ordering with respect to the BTL parameters. In empirical evaluations presented subsequently, we see that the counting algorithm is significantly more robust to various kinds of noise, and takes several orders of magnitude lesser time to compute.

Finally, in addition to the notion of exact recovery considered so far, in the next two subsections we also derive tight guarantees for the Hamming error metric and more general metrics inspired by the requirements of many relevant applications (Ilyas et al., 2008; Michel et al., 2005; Babcock and Olston, 2003; Metwally et al., 2005; Kimelfeld and Sagiv, 2006; Fagin et al., 2003).

3.2 Approximate recovery under Hamming error

In the previous section, we analyzed performance in terms of exactly recovering the top-k subset. Although exact recovery is suitable for some applications (e.g., a setting with high stakes, in which any single error has a large price), there are other settings in which it may be acceptable to return a subset that is "close" to the correct k-ranked subset. In this section, we analyze this problem of approximate recovery when closeness is measured under the Hamming error. More precisely, for a given threshold $h \in [0, k)$, suppose that our goal is to output a set k-sized set $\widehat{\mathcal{S}}_k$ such that its Hamming distance to the set $\mathcal{S}_k^*$ of the true top k items, as defined in equation (3), is bounded as

$$D_{\mathrm{H}}(\widehat{\mathcal{S}}_k, \mathcal{S}_k^*) \leq 2h. \tag{11}$$

Our goal is to establish conditions under which it is possible (or impossible) to return an estimate $\widehat{\mathcal{S}}_k$ satisfying the bound (11) with high probability.[1]

As before, we use $(1), \ldots, (n)$ to denote the permutation of the n items in decreasing order of their scores. With this notation, the following quantity plays a central role in our analysis:

$$\Delta_{k,\,h}(M) := \tau_{(k-h)}(M) - \tau_{(k+h+1)}(M). \tag{12a}$$

The quantity $\Delta_{k,\,h}$ measures the difference between the scores associated to the items which are h positions on either side of our desired boundary between the k^{th} and $(k+1)^{\text{th}}$ items. Observe that $\Delta_{k,\,h}$ is a generalization of the quantity Δ_k defined previously in equation (8); and the quantity Δ_k corresponds to $\Delta_{k,\,h}$ with $h = 0$. We then define a generalization of the family $\mathcal{F}_k(\alpha; n, p, r)$, namely

$$\mathcal{F}_{k,h}(\alpha; n, p, r) := \left\{ M \in [0,1]^{n \times n} \;\middle|\; M + M^T = 11^T, \text{ and } \Delta_{k,\,h} \geq \alpha \sqrt{\frac{\log n}{npr}} \right\}. \tag{12b}$$

As before, we adopt the shorthand $\mathcal{F}_{k,h}(\alpha)$, with the dependence on (n, p, r) being understood implicitly.

Theorem 3 *(a) Consider any $n \geq 2$, $r \geq 1$ and $p \in (0, 1]$. Then if $\alpha \geq 8$, the set $\widetilde{\mathcal{S}}_k$ of top k items (7) given by the Borda counting algorithm satisfies*

$$\sup_{M \in \mathcal{F}_{k,h}(\alpha)} \mathbb{P}_M\big[D_{\mathrm{H}}(\widetilde{\mathcal{S}}_k, \mathcal{S}_k^*) > 2h\big] \leq \frac{1}{n^{14}}. \tag{13a}$$

(b) Conversely, in the regime $p \geq \frac{\log n}{2nr}$ and for given constants $\nu_1, \nu_2 \in (0, 1)$, suppose that $2h \leq \frac{1}{1+\nu_2} \min\{n^{1-\nu_1}, k, n-k\}$. Then for any $\alpha \leq \frac{\sqrt{\nu_1 \nu_2}}{14}$, any estimator $\widehat{\mathcal{S}}_k$ has error at least

$$\sup_{M \in \mathcal{F}_{k,h}(\alpha)} \mathbb{P}_M\big[D_{\mathrm{H}}(\widehat{\mathcal{S}}_k, \mathcal{S}_k^*) > 2h\big] \geq \frac{1}{7}, \tag{13b}$$

for all n larger than a constant $c(\nu_1, \nu_2)$.

1. The requirement $h < k$ is sensible because if $h \geq k$, the problem is trivial: any two k-sized sets $\widehat{\mathcal{S}}_k$ and $\mathcal{S}_k^*$ satisfy the bound $D_{\mathrm{H}}(\widehat{\mathcal{S}}_k, \mathcal{S}_k^*) \leq 2k \leq 2h$.

This result is similar to that of Theorem 1, except that the relaxation of the exact recovery condition allows for a less constrained definition of the separation threshold $\Delta_{k,h}$. As with Theorem 1, the lower bound in part (b) applies even if probability matrix M is restricted to lie in a parametric model (such as the BTL model), or the more general SST class. The counting algorithm is thus optimal for estimation under the relaxed Hamming metric as well.

The proof of the upper bound involves a transformation of the Hamming error requirement into one of exact recovery requirement, and then transforming the result of Theorem 1(a) to that required here via certain algebraic arguments. The lower bound is significantly more intricate: we carefully design a packing set using a coding-theoretic result due to Levenshtein (1971), which we then employ in Fano's inequality.

Finally, it is worth making a few comments about the constants appearing in these claims. We can weaken the lower bound on Δ_k required in Theorem 3(a) at the expense of a lower probability of success; for instance, if we instead require that $\alpha \geq 4$, then the probability of error is guaranteed to be at most n^{-2}. Subsequently in the paper, we provide the results of simulations with $n = 500$ items and $\alpha = 4$. On the other hand, in Theorem 3(b), if we impose the stronger upper bound $\alpha = \mathcal{O}(1/\sqrt{h \log n})$, then we can remove the condition $h \leq n^{1-\nu_1}$.

3.3 An abstract form of k-set recovery

In earlier sections, we investigated recovery of the top k items either exactly or under a Hamming error. Exact recovery may be quite strict for certain applications, whereas the property of Hamming error allowing for a few of the top k items to be replaced by *arbitrary* items may be undesirable. Indeed, many applications have requirements that go beyond these metrics; for instance, see the papers Ilyas et al. (2008); Michel et al. (2005); Babcock and Olston (2003); Metwally et al. (2005); Kimelfeld and Sagiv (2006); Fagin et al. (2003) and references therein for some examples. In this section, we generalize the notion of exact or Hamming-error recovery in order to accommodate a fairly general class of requirements.

Both the exact and approximate Hamming recovery settings require the estimator to output a set of k items that are either exactly or approximately equal to the true set of top k items. When is the estimate deemed successful? One way to think about the problem is as follows. The specified requirement of exact or approximate Hamming recovery is associated to a set of k-sized subsets of the n possible ranks. The estimator is deemed successful if the true ranks of the chosen k items equals one of these subsets. In our notion of generalized recovery, we refer to such sets as *allowed sets*. For example, in the case $k = 3$, we might say that the set $\{1, 4, 10\}$ is allowed, meaning that an output consisting of the items that are ranked "first", "fourth" and "tenth" in the ground truth is considered correct.

In more generality, let $\mathfrak{S}$ denote a family of k-sized subsets of $[n]$, which we refer to as *family of allowed sets*. Notice that any allowed set is defined by the *positions* of the items in the true ordering and not the items themselves.[2] Once some true underlying ordering of the n items is fixed, each element of the family $\mathfrak{S}$ then specifies a set of the items themselves. We use these two interpretations depending on the context — the definition in terms of

2. In case of two or more items with identical scores, the choice of any of these items is considered valid.

positions to specify the requirements, and the definition in terms of the items to evaluate an estimator for a given underlying probability matrix M.

We let $\mathcal{S}_k^\dagger$ denote a k-set estimate, meaning a function that given a set of observations as input, returns a k-sized subset of $[n]$ as output.

Definition 4 ($\mathfrak{S}$-respecting estimators) *For any family $\mathfrak{S}$ of allowed sets, a k-set estimate $\mathcal{S}_k^\dagger$* respects *its structure if the set of k positions of the items in $\mathcal{S}_k^\dagger$ belongs to the set family $\mathfrak{S}$.*

Our goal is to determine conditions on the set family $\mathfrak{S}$ under which there exist estimators $\mathcal{S}_k^\dagger$ that respect its structure. In order to illustrate this definition, let us return to the examples treated thus far.

Example 1 (Exact and approximate Hamming recovery) *The requirement of exact recovery of the top k items has $\mathfrak{S}$ consisting of exactly one set, the set of the top k positions $\mathfrak{S} = \{[k]\}$. In the case of recovery with a Hamming error at most $2h$, the set $\mathfrak{S}$ of all allowed sets consists all k-sized subsets of $[n]$ that contain at least $(k - h)$ positions in the top k positions. For instance, in the case $h = 1$, $k = 2$ and $n = 4$, we have*

$$\mathfrak{S} = \Big\{\{1,2\},\{1,3\},\{1,4\},\{2,3\},\{2,4\}\Big\}.$$

Apart from these two requirements, there are several other requirements for top-k recovery popular in the literature (Carmel et al., 2001; Fagin et al., 2003; Babcock and Olston, 2003; Michel et al., 2005; Metwally et al., 2005; Kimelfeld and Sagiv, 2006; Ilyas et al., 2008). Let us illustrate them with another example:

Example 2 *Let $\pi^* : [n] \to [n]$ denote the true underlying ordering of the n items. The following are four popular requirements on the set $\mathcal{S}_k^\dagger$ for top-k identification, with respect to the true permutation π^*, for a pre-specified parameter $\epsilon \geq 0$.*

(i) All items in the set $\mathcal{S}_k^\dagger$ must be contained contained within the top $(1+\epsilon)k$ entries:

$$\max_{i \in \mathcal{S}_k^\dagger} \pi^*(i) \leq (1+\epsilon)k. \tag{14a}$$

(ii) The rank of any item in the set $\mathcal{S}_k^\dagger$ must lie within a multiplicative factor $(1+\epsilon)$ of the rank of any item not in the set $\mathcal{S}_k^\dagger$:

$$\max_{i \in \mathcal{S}_k^\dagger} \pi^*(i) \leq (1+\epsilon) \min_{j \in [n] \setminus \mathcal{S}_k^\dagger} \pi(j). \tag{14b}$$

(iii) The rank of any item in the set $\mathcal{S}_k^\dagger$ must lie within an additive factor ϵ of the rank of any item not in the set $\mathcal{S}_k^\dagger$:

$$\max_{i \in \mathcal{S}_k^\dagger} \pi^*(i) \leq \min_{j \in [n] \setminus \mathcal{S}_k^\dagger} \pi^*(j) + \epsilon. \tag{14c}$$

(iv) The sum of the ranks of the items in the set $\mathcal{S}_k^\dagger$ *must be contained within a factor* $(1+\epsilon)$ *of the sums of ranks of the top* k *entries:*

$$\sum_{i \in \mathcal{S}_k^\dagger} \pi^*(i) \le (1+\epsilon)\frac{1}{2}k(k+1). \tag{14d}$$

Note that each of these requirements reduces to the exact recovery requirement when $\epsilon = 0$. *Moreover, each of these requirements can be rephrased in terms of families of allowed sets. For instance, if we focus on requirement (i), then any* k*-sized subset of the top* $(1+\epsilon)k$ *positions is an allowed set.*

In this paper, we derive conditions that govern k-set recovery for allowed set systems that satisfy a natural "monotonicity" condition. Informally, the monotonicity condition requires that the set of k items resulting from replacing an item in an allowed set with a higher ranked item must also be an allowed set. More precisely, for any set $\{t_1, \ldots, t_k\} \subseteq [n]$, let $\Lambda(\{t_1, \ldots, t_k\}) \subseteq 2^{[n]}$ be the set defined by all of its monotone transformations—that is

$$\Lambda(\{t_1, \ldots, t_k\}) := \Big\{\{t'_1, \ldots, t'_k\} \subseteq [n] \mid t'_j \le t_j \text{ for every } j \in [k]\Big\}.$$

Using this notation, we have the following:

Definition 5 (Monotonic set systems) *The set* $\mathfrak{S}$ *of allowed sets is a* monotonic set system *if*

$$\Lambda(T) \subseteq \mathfrak{S} \quad \textit{for every } T \in \mathfrak{S}. \tag{15}$$

One can verify that condition (15) is satisfied by the settings of exact and Hamming-error recovery, as discussed in Example 1. The condition is also satisfied by all four requirements discussed in Example 2.

Our next result establishes conditions under which one can (or cannot) produce an estimator that respects an allowed set requirement. In order to state the result, we recall the score $\tau_i(M) := \frac{1}{n}\sum_{j=1}^n M_{ij}$, as previously defined in equation (2) for each $i \in [n]$. For notational convenience, we also define $\tau_i(M) := -\infty$ for every $i > n$ and every M. Consider any monotonic family of allowed sets $\mathfrak{S}$, and for some integer $\beta \ge 1$, let $T^1, \ldots, T^\beta \in \mathfrak{S}$ such that $\mathfrak{S} = \bigcup_{b \in [\beta]} \Lambda(T^b)$. For every $b \in [\beta]$, let $t_1^b < \cdots < t_k^b$ denote the entries of T^b. We then define the critical threshold based on the scores:

$$\Delta_{\mathfrak{S}}(M) := \max_{b \in [\beta]} \min_{j \in [k]} \big(\tau_{(j)}(M) - \tau_{(k + t_j^b - j + 1)}(M)\big). \tag{16}$$

The term $\Delta_{\mathfrak{S}}$ is a further generalization of the quantities Δ_k and $\Delta_{k,h}$ defined in earlier sections. For example, for the exact recovery setting we have $\beta = 1$ and $T^1 = \{1, \ldots, k\}$, and after some algebraic simplifications of (16), we obtain that the critical threshold $\Delta_{\mathfrak{S}}$ reduces exactly to the threshold Δ_k defined earlier in (8). As a second example, the setting allowing a Hamming error at most $2h$ can be described with the choice $\beta = 1$ and $T^1 = \{h+1, \ldots, k,\ n-h+1, \ldots, n\}$. Some algebraic simplifications of (16) reduce $\Delta_{\mathfrak{S}}$ to the threshold $\Delta_{k,h}$ defined in (12a). As an example with $\beta > 1$, consider requirement (ii) in Example 2. For simplicity of exposition, assume that $\epsilon > \frac{1}{k-1}$ and $n \ge 2k(1+\epsilon)$. For this requirement, we have $\beta = (k - \lceil \frac{k}{1+\epsilon} \rceil)$, and for every $b \in \{\lceil \frac{k}{1+\epsilon} \rceil + 1, \ldots, k\}$, we have $T^{b - \lceil \frac{k}{1+\epsilon} \rceil} = \{1, \ldots, b-1, \lceil (1+\epsilon)b - (k-b) \rceil, \ldots, \lceil (1+\epsilon)b \rceil\}$. Then some algebraic simplifications of equation (16) yield that the critical threshold for this requirement is given by $\Delta_{\mathfrak{S}}(M) = \max_{b \in \{\lceil \frac{k}{1+\epsilon} \rceil + 1, \ldots, k\}} \min\{\tau_{(b-1)}(M) - \tau_{(k+1)}(M), \tau_{(k)}(M) - \tau_{(\lceil (1+\epsilon)b \rceil + 1)}(M)\}$.

With the definition of the critical threshold $\Delta_{\mathfrak{S}}$ in place, we now define a generalization $\mathcal{F}_{\mathfrak{S}}(\cdot)$ of the families $\mathcal{F}_k(\cdot)$ and $\mathcal{F}_{k,h}(\cdot)$ as

$$\mathcal{F}_{\mathfrak{S}}(\alpha; n, p, r) := \left\{ M \in [0,1]^{n \times n} \mid M + M^T = 11^T \text{ and } \Delta_{\mathfrak{S}}(M) \ge \alpha \sqrt{\frac{\log n}{npr}} \right\}. \tag{17}$$

As before, we use the shorthand $\mathcal{F}_{\mathfrak{S}}(\alpha)$, with the dependence on (n, p, r) being understood implicitly.

Theorem 6 *Consider any allowed set requirement specified by a monotonic set class* $\mathfrak{S}$.

(a) For any $\alpha \ge 8$, *the set* $\widetilde{\mathcal{S}}_k$ *of top* k *items (7) given by the Borda counting algorithm satisfies*

$$\sup_{M \in \mathcal{F}_{\mathfrak{S}}(\alpha)} \mathbb{P}_M\big[\widetilde{\mathcal{S}}_k \notin \mathfrak{S}\big] \le \frac{1}{n^{13}}.$$

(b) Conversely, in the regime $p \ge \frac{\log n}{2nr}$, *and for given constants* $\mu_1 \in (0,1), \mu_2 \in (\frac{3}{4}, 1]$, *suppose that* $\max_{b \in [\beta]} t^b_{\lceil \mu_2 k \rceil} \le \frac{n}{2}$ *and* $8(1-\mu_2)k \le n^{1-\mu_1}$. *Then for any* α *smaller than a constant* $c_u(\mu_1, \mu_2) > 0$, *any estimator* $\widehat{\mathcal{S}}_k$ *has error at least*

$$\sup_{M \in \mathcal{F}_{\mathfrak{S}}(\alpha)} \mathbb{P}_M\big[\widetilde{\mathcal{S}}_k \notin \mathfrak{S}\big] \ge \frac{1}{15}, \tag{18}$$

for all n *larger than a constant* $c_0(\mu_1, \mu_2)$.

A few remarks on the bounds are in order. For the lower bound, first, it continues to hold even if the probability matrix M is restricted to follow a parametric model such as BTL or restricted to lie in the SST class. Second, in terms of the threshold for α, the lower bound holds with $c_u(\mu_1, \mu_2) = \frac{1}{15}\sqrt{\mu_1 \min\left\{\frac{1}{4(1-\mu_2)-1}, \frac{1}{2}\right\}}$. Third, it is worth noting that one must necessarily impose some conditions for the lower bound, along the lines of those required in Theorem 6(b) for the allowed sets to be "interesting" enough. As a concrete illustration of the necessity of this condition, consider the requirement defined by the parameters $b = 1$, $k = 1$ and $\mathfrak{S} = \Lambda(\{n - \sqrt{n}\})$. For $\mu_1 = \mu_2 = \frac{9}{10}$, this requirement satisfies the condition $8(1-\mu_2)k \le n^{1-\mu_1}$ but violates the condition $t_{\lceil \mu_2 k \rceil} \le \frac{n}{2}$. Now, a selection of $k = 1$ item made uniformly at random (independent of the data) satisfies this allowed set requirement with probability $1 - \frac{1}{\sqrt{n}}$. Given the success of such a random selection algorithm in this parameter regime, we see that the lower bounds therefore cannot be universal, but must require some conditions on the allowed sets.

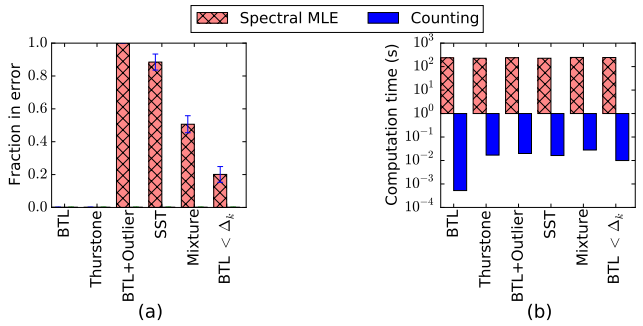


Figure 1. Simulation results comparing Spectral MLE and the counting algorithm in terms of error rates for exact recovery of the top k items, and computation time. (a) Histogram of fraction of instances where the algorithm failed to recover the k items correctly, with each bar being the average value across 50 trials. The counting algorithm has 0% error across all problems, while the spectral MLE is accurate for parametric models (BTL, Thurstone), but not very accurate for other models. (b) Histogram plots of the maximum computation time taken by the counting algorithm and the minimum computation time taken by Spectral MLE across all trials. Even though this maximum-to-minimum comparison is unfair to the counting algorithm, it involves five or more orders of magnitude less computation.

4. Simulations and experiments

In this section, we empirically evaluate the performance of the counting algorithm and compare it with the Spectral MLE algorithm via simulations on synthetic data, as well as experiments using datasets from the Amazon Mechanical Turk crowdsourcing platform.

4.1 Simulated data

We begin with simulations using synthetically generated data with $n = 500$ items and observation probability $p = 1$, and with pairwise comparison models ranging over six possible types. Panel (a) in Figure 1 provides a histogram plot of the associated error rates (with a bar for each one of these six models) in recovering the $k = n/4 = 125$ items for the counting algorithm versus the Spectral MLE algorithm. Each bar corresponds to the average over 50 trials. Panel (b) compares the CPU times of the two algorithms. The value of α (and in turn, the value of r) in the first five models is as derived in Section 3.1. In more detail, the six model types are given by:

(I) *Bradley-Terry-Luce (BTL) model:* Recall that the theoretical guarantees for the Spectral MLE algorithm (Chen and Suh, 2015) are applicable to data that is generated from the BTL model (4a), and as guaranteed, the Spectral MLE algorithm gives a 100% accuracy under this model. The counting algorithm also obtains a 100% accuracy, but importantly, the counting algorithm requires a computational time that is five orders of magnitude lower than that of Spectral MLE.

(II) *Thurstone model:* The Thurstone model (Thurstone, 1927) is another parametric model, with the function F in equation (4b) set as the cumulative distribution function of the standard Gaussian distribution. Both Spectral MLE and the counting algorithm gave 100% accuracy under this model.

(III) *BTL model with one (non-transitive) outlier:* This model is identical to BTL, with one modification. Comparisons among $(n-1)$ of the items follow the BTL model as before, but the remaining item always beats the first $\frac{n}{4}$ items and always loses to each of the other items. We see that the counting algorithm continues to achieve an accuracy of 100% as guaranteed by Theorem 1. The departure from the BTL model however prevents the Spectral MLE algorithm from identifying the top k items.

(IV) *Strong stochastic transitivity (SST) model:* We simulate the "independent diagonals" construction of Shah et al. (2017a) in the SST class. Spectral MLE is often unsuccessful in recovering the top k items, while the counting algorithm always succeeds.

(V) *Mixture of BTL models:* Consider two sets of people with opposing preferences. The first set of people have a certain ordering of the items in their mind and their preferences follow a BTL model under this ordering. The second set of people have the opposite ordering, and their preferences also follow a BTL model under this opposite ordering. The overall preference probabilities is a mixture between these two sets of people. In the simulations, we observe that the counting algorithm is always successful while the Spectral MLE method often fails.

(VI) *BTL with violation of separation condition:* We simulate the BTL model, but with a choice of parameter r small enough that the value of α is about one-tenth of its recommended value in Section 3.1. We observe that the counting algorithm continues to incur a lower error than the Spectral MLE algorithm, thereby demonstrating its robustness.

To summarize, the performance of the two algorithms can be contrasted in the following way. When our stated lower bounds on α are satisfied, then consistent with our theoretical claims, the Borda counting algorithm succeeds irrespective of the form of the pairwise probability distributions. The Spectral MLE algorithm performs well when the pairwise comparison probabilities are faithful to parametric models, but is often unsuccessful otherwise. Even when the condition on α is violated, the performance of the counting algorithm remains superior to that of the Spectral MLE.[3] In terms of computational complexity, for every instance we simulated, the counting algorithm took several orders of magnitude less time as compared to Spectral MLE.

Simulations with adversarial, imbalanced choice of pairs: The theoretical results in the earlier sections addressed a random design setting where the pairs to be compared are chosen at random in a homogeneous manner. While such a random design setting is widespread in various applications such as crowdsourcing and others, and is also the focus of a bulk of past literature on related topics, it is also of interest to understand situations where the

3. Note that part (b) of Theorem 1 is a minimax converse meaning that it appeals to the worst case scenario.

comparisons may be imbalanced. With this goal, we new present simulations that contrast the behavior of the counting algorithm in a random-design setting with an adversarial-design setting.

In this set of simulations, we consider the problem of recovering the top item (that is, $k = 1$). Moreover, we adopt a parametric model in which every item $i \in [n]$ is assumed to be governed by a parameter $w_i^* \in [0, 1]$, and the probability of any item i beating item j is set as $M_{ij} = \frac{1+w_i^*-w_j^*}{2}$. As before, the number of times any pair of items (i, j) is compared is drawn as a binomial distribution with parameters r and p. In the standard random-design setting studied throughout the remainder of this paper, the choice of the number of comparisons is unrelated to the choice of the parameters w^*. However in the adversarial setting, we make w^* adversarially misaligned to the number of comparisons between various pairs. In particular, the parameters associated to the items are chosen as follows for the random and the adversarial settings:

1. Random: $w_1^* = 1$ and $w_2^* = 0.9$ are the top two items. For every $i \in \{3, \ldots, n\}$, we draw w_i^* uniformly at random from the set $\{0.1, 0.7\}$.

2. Adversarial: $w_1^* = 1$ and $w_2^* = 0.9$ are the top two items. For every $i \in \{3, \ldots, n\}$, we set $w_i^* = 0.7$ if item i is compared more often to item 1 than to item 2, set $w_i^* = 0.1$ if item i is compared more often to item 2 than to item 1, and draw w_i^* uniformly at random from the set $\{0.1, 0.7\}$ otherwise.

The results of applying the counting algorithm are shown in Figure 2. From these simulations we observe that the simple counting algorithm is indeed sensitive to the imbalanced choice of pairs to be compared (that is, when the top item is compared more often to higher ranked items and the second item is compared more often to lower ranked items). Designing algorithms for ranking from pairwise comparisons that can optimally handle such imbalanced, adversarial-design settings is left as a problem for future work.

4.2 Experiments on data from Amazon Mechanical Turk

In this section, we describe experiments on real world datasets collected from the Amazon Mechanical Turk (`mturk.com`) commercial crowdsourcing platform.

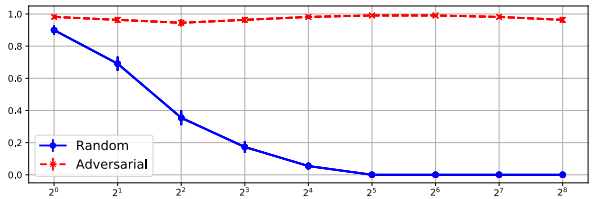


Figure 2. Performance of the counting algorithm for top $k = 1$ recovery when the pairs to be compared are chosen randomly and when they are chosen adversarially to create an imbalanced setting.

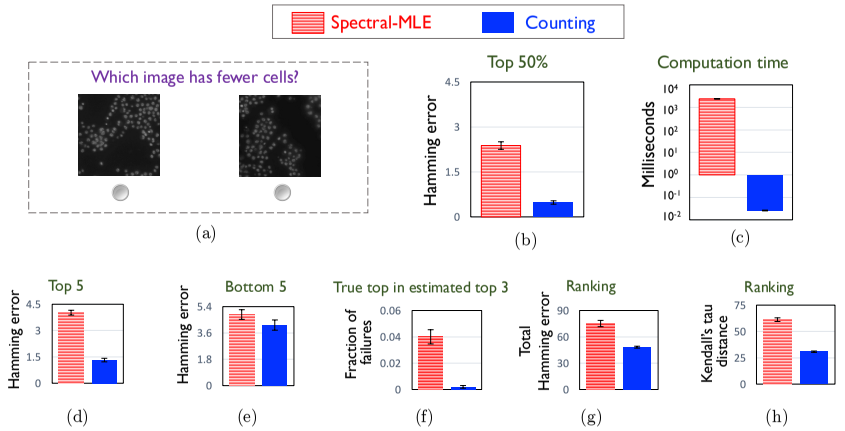


Figure 3. An illustration of the cell counting experiment (panel a) and results comparing Spectral MLE and the counting algorithm in terms of accuracy and computation time (panels b–h).

4.2.1 Experiment on counting cells

We begin with an experiment on counting (biological) cells in images.

Data. We employed a dataset of 23 images, each comprising several (biological) cells. The images of the cells and the ground truth counts of the numbers of cells in each image were obtained from the dataset collected by Carpenter et al. (2006).

On the Amazon Mechanical Turk crowdsourcing platform, we recruited a total of 64 workers and showed multiple pairs of such images to each worker – see Figure 3(a) for an illustrative example. For each pair of images, the worker was asked to select the image that the worker considered to have fewer cells. For each worker, the pairs were chosen by permuting the 23 images uniformly at random and asking for a comparison between the first and second images, between the third and fourth images and so on, for a total of 11 pairs of images per worker. In the raw data, 9.8% of the responses provided by the workers were erroneous. In the raw data we also observed that unsurprisingly, the number of errors increase as the actual cell counts in the pair of images come closer. The interface seen by the workers as well as the raw data obtained from Amazon Mechanical Turk is available on the website of the first author.

The goal of any algorithm is to take this set of noisy pairwise comparisons from the workers and estimate the images with the fewest cells. Such estimates are useful to detect various conditions, for instance, a low count of red blood cells in images of human cells indicates anemia. To this end, we executed the Spectral MLE algorithm (Chen and Suh, 2015) and the Borda counting algorithm on the set of pairwise comparisons obtained from the workers.

Results. We compared the performance of the two algorithms on a variety of metrics. In what follows, we subsample the responses with $p = 0.5$, that is, for each response for each question, we keep the response independently with probability 0.5 and discard it otherwise. We execute the two algorithms on this subsampled data. We repeat this process for 100 trials and plot the mean of the metric under consideration along with error bars representing the standard error of the mean.

We first consider recovering the set of top $k = \frac{n}{2}$ items. The natural metric of error here is the Hamming error, that is, the number of images that are misclassified. For this objective, while Spectral MLE does quite well, counting incurs a significantly lower Hamming error – see Figure 3(b). As one may expect, the count estimator also requires a much lower computation time – see Figure 3(c) for a comparison. In Figures 3(d) to (h), we see that counting also performs quite well for other exact or approximate requirements of top k or ranking recovery.

4.2.2 Data from earlier experiments on Amazon Mechanical Turk

We now describe three additional experiments using data collected from Amazon Mechanical Turk in our past work Shah et al. (2016a).

Data. In order to evaluate the accuracy of the algorithms under consideration, we require datasets consisting of pairwise comparisons in which the questions can be associated with an objective and verifiable ground truth. To this end, we used the "cardinal versus ordinal" dataset from our past work Shah et al. (2016a); three of the experiments performed in that paper are suitable for the evaluations here—namely, ones in which each question has a ground truth, and the pairs of items are chosen uniformly at random. The three experiments tested the workers' general knowledge, audio, and visual understanding, and the respective tasks involved: (i) identifying the pair of cities with a greater geographical distance, (ii) identifying the higher frequency key of a piano, and (iii) identifying spelling mistakes in a paragraph of text. The number of items n in the three experiments were 16, 10 and 8 respectively. The total number of pairwise comparisons were 408, 265 and 184 respectively. The fraction of pairwise comparisons whose outcomes were incorrect (as compared to the ground truth) in the raw data are 17%, 20% and 40% respectively.

Results. We compared the performance of the counting algorithm with that of the Spectral MLE algorithm. For each value of a "subsampling probability" $q \in \{0.1, 0.2, \ldots, 1.0\}$, we subsampled a fraction q of the data and executed both algorithms on this subsampled data. We evaluated the performance of the algorithms on their ability to recover the top $k = \lceil \frac{n}{4} \rceil$ items under the Hamming error metric.

Figure 4 shows the results of the experiments. Each point in the plots is an average across 100 trials. Observe that the counting algorithm consistently outperforms Spectral MLE. (We think that the erratic fluctuations in the spelling mistakes data are a consequence of a high noise and a relatively small problem size.) Moreover, the Spectral MLE algorithm required about 5 orders of magnitude more computation time (not shown in the figure) as compared to counting. Thus the counting algorithm performs well on simulated as well as real data. It outperforms Spectral MLE not only when the number of items is large (as in the simulations) but also when the problem sizes are small as seen in these experiments.

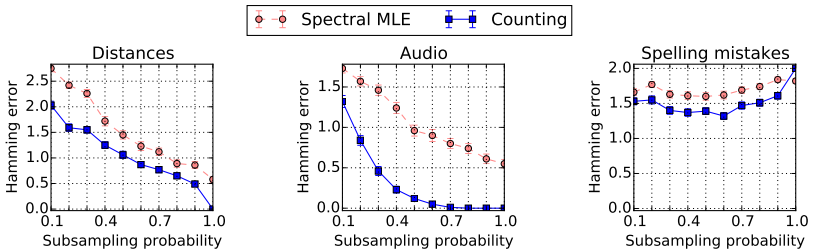


Figure 4. Evaluation of Spectral MLE and the counting algorithm on three datasets (from left to right: Distances, Audio, Spelling mistakes) from Amazon Mechanical Turk in terms of the error rates for top k-subset recovery. The three panels plot the Hamming error when recovering the top k items in the three datasets when a q^{th} fraction of the total data is used, for various values of subsampling probability $q \in (0, 1]$.

5. Proofs

We now turn to the proofs of our main results. We continue to use the notation $[i]$ to denote the set $\{1, \ldots, i\}$ for any integer $i \geq 1$. We ignore floor and ceiling conditions unless critical to the proof. All logarithms are taken to the base e.

Our lower bounds are based on a standard form of Fano's inequality (Cover and Thomas, 2012; Tsybakov, 2008) for lower bounding the probability of error in an L-ary hypothesis testing problem. We state a version here for future reference. For some integer $L \geq 2$, fix some collection of distributions $\{\mathbb{P}^1, \ldots, \mathbb{P}^L\}$. Suppose that we observe a random variable Y that is obtained by first sampling an index A uniformly at random from $[L] = \{1, \ldots, L\}$, and then drawing $Y \sim \mathbb{P}^A$. (As a result, the variable Y is marginally distributed according to the mixture distribution $\bar{\mathbb{P}} = \frac{1}{L} \sum_{a=1}^{L} \mathbb{P}^a$.) Given the observation Y, our goal is to "decode" the value of A, corresponding to the index of the underlying mixture component. Using $\mathcal{Y}$ to denote the sample space associated with the observation Y, Fano's inequality asserts that any test function $\phi : \mathcal{Y} \to [L]$ for this problem has error probability lower bounded as

$$\mathbb{P}[\phi(Y) \neq A] \geq 1 - \frac{I(Y; A) + \log 2}{\log L},$$

where $I(Y; A)$ denotes the mutual information between Y and A. A standard convexity argument for the mutual information yields the weaker bound

$$\mathbb{P}[\phi(Y) \neq A] \geq 1 - \frac{\max\limits_{a,b \in [L]} D_{\mathrm{KL}}(\mathbb{P}^a \| \mathbb{P}^b) + \log 2}{\log L}, \tag{19}$$

We make use of this weakened form of Fano's inequality in several proofs.

5.1 Proof of Theorem 1

We begin with the proof of Theorem 1, dividing our argument into two parts.

5.1.1 Proof of part (a)

For any pair of items (i, j), let us encode the outcomes of the r trials by an i.i.d. sequence $V_{ij}^{(\ell)} = [X_{ij}^{(\ell)} \quad X_{ji}^{(\ell)}]^T$ of random vectors, indexed by $\ell \in [r]$. Each random vector follows the distribution

$$\mathbb{P}\big[x_{ij}^{(\ell)},\, x_{ji}^{(\ell)}\big] = \begin{cases} 1-p & \text{if } (x_{ij}^{(\ell)},\, x_{ji}^{(\ell)}) = (0,0) \\ pM_{ij} & \text{if } (x_{ij}^{(\ell)},\, x_{ji}^{(\ell)}) = (1,0) \\ p(1-M_{ij}) & \text{if } (x_{ij}^{(\ell)},\, x_{ji}^{(\ell)}) = (0,1) \\ 0 & \text{otherwise.} \end{cases}$$

With this encoding, the variable $W_a := \sum_{\ell\in[r]} \sum_{z\in[n]\setminus\{a\}} X_{aj}^{(r)}$ encodes the number of wins for item a.

Consider any item $a \in \mathcal{S}_k^*$ which ranks among the top k in the true underlying ordering, and any item $b \in [n]\setminus\mathcal{S}_k^*$ which ranks outside the top k. We claim that with high probability, item a will win more pairwise comparisons than item b. More precisely, let $\mathcal{E}_{ba}$ denote the event that item b wins at least as many pairwise comparisons than a. We claim that

$$\mathbb{P}(\mathcal{E}_{ba}) \overset{(i)}{\leq} \exp\left(-\frac{\frac{1}{2}(rpn\Delta_k)^2}{rpn(2-\Delta_k)+\frac{2}{3}rpn\Delta_k}\right) \overset{(ii)}{\leq} \frac{1}{n^{16}}. \tag{20}$$

Given this bound, the probability that the counting algorithm will rank item b above a is no more than n^{-16}. Applying the union bound over all pairs of items $a \in \mathcal{S}_k^*$ and $b \in [n]\setminus\mathcal{S}_k^*$ yields $\mathbb{P}\big[\widehat{\mathcal{S}}_k \neq \mathcal{S}_k^*\big] \leq n^{-14}$ as claimed.

We note that inequality (ii) in equation (20) follows from inequality (i) combined with the condition on Δ_k that arises by setting $\alpha \geq 8$ as assumed in the hypothesis of the theorem. Thus, it remains to prove inequality (i) in equation (20). By definition of $\mathcal{E}_{ba}$, we have

$$\mathbb{P}(\mathcal{E}_{ba}) = \mathbb{P}\Big(\underbrace{\sum_{\ell\in[r]}\sum_{z\in[n]\setminus\{b\}} X_{bz}^{(\ell)}}_{W_b} - \underbrace{\sum_{\ell\in[r]}\sum_{z\in[n]\setminus\{a\}} X_{az}^{(\ell)}}_{W_a} \geq 0\Big). \tag{21}$$

It is convenient to recenter the random variables. For every $\ell \in [r]$ and $z \in [n]\setminus\{a,b\}$, define the zero-mean random variables

$$\overline{X}_{az}^{(\ell)} = X_{az}^{(\ell)} - \mathbb{E}[X_{az}^{(\ell)}] = X_{az}^{(\ell)} - pM_{az} \;\text{ and }\; \overline{X}_{bz}^{(\ell)} = X_{bz}^{(\ell)} - \mathbb{E}[X_{bz}^{(\ell)}] = X_{bz}^{(\ell)} - pM_{bz}.$$

Also, let

$$\overline{X}_{ab}^{(\ell)} = (X_{ab}^{(\ell)} - X_{ba}^{(\ell)}) - \mathbb{E}[X_{ab}^{(\ell)} - X_{ba}^{(\ell)}] \;=\; (X_{ab}^{(\ell)} - X_{ba}^{(\ell)}) - (pM_{ab} - pM_{ba}).$$

We then have

$$\mathbb{P}(\mathcal{E}_{ba}) = \mathbb{P}\Bigg(\sum_{\ell\in[r]}\Big(\sum_{z\in[n]\setminus\{a,b\}} \overline{X}_{bz}^{(\ell)} - \sum_{z\in[n]\setminus\{a,b\}} \overline{X}_{az}^{(\ell)} - \overline{X}_{ab}^{(\ell)}\Big) \geq rp\sum_{z\in[n]}\Big(M_{az} - M_{bz}\Big)\Bigg).$$

Since $a \in \mathcal{S}_k^*$ and $b \in [n]\setminus\mathcal{S}_k^*$, from the definition of Δ_k, we have $n\Delta_k \leq \sum_{z\in[n]}(M_{az} - M_{bz})$, and consequently

$$\mathbb{P}\left(\mathcal{E}_{ba}\right) \leq \mathbb{P}\left(\sum_{\ell\in[r]}\Big(\sum_{z\in[n]\setminus\{a,b\}} \overline{X}_{bz}^{(\ell)} - \sum_{z\in[n]\setminus\{a,b\}} \overline{X}_{az}^{(\ell)} - \overline{X}_{ab}^{(\ell)}\Big) \geq rpn\Delta_k\right). \tag{22}$$

By construction, all the random variables in the above inequality are zero-mean, mutually independent, and bounded in absolute value by 2. These properties alone would allow us to obtain a tail bound by Hoeffding's inequality; however, in order to obtain the stated result (20), we need the more refined result afforded by Bernstein's inequality (e.g., Boucheron et al., 2013). In order to derive a bound of Bernstein type, the only remaining step is to bound the second moments of the random variables at hand. Some straightforward calculations yield

$$\mathbb{E}[(-\overline{X}_{az}^{(\ell)})^2] \leq pM_{az}, \qquad \mathbb{E}[(\overline{X}_{bz}^{(\ell)})^2] \leq pM_{bz}, \quad \text{and} \quad \mathbb{E}[(\overline{X}_{ab}^{(\ell)})^2] \leq pM_{ab} + pM_{ba}.$$

It follows that

$$\begin{aligned} \sum_{z\in[n]\setminus\{a,b\}} \mathbb{E}[(-\overline{X}_{az}^{(\ell)})^2] + \sum_{z\in[n]\setminus\{a,b\}} \mathbb{E}[(\overline{X}_{bz}^{(\ell)})^2] + \mathbb{E}[(\overline{X}_{ab}^{(\ell)})^2] & \\ \leq p\left(\sum_{z\in[n]\setminus\{a,b\}}(M_{az}+M_{bz}) + M_{ab} + M_{ba}\right) & \\ \overset{(iii)}{\leq} p\left(2\sum_{z\in[n]} M_{az} - n\Delta_k\right) & \\ \overset{(iv)}{<} pn(2-\Delta_k), & \end{aligned}$$

where the inequality (iii) follows from the definition of Δ_k, and step (iv) follows because $M_{az} \leq 1$ for every z and $M_{aa} = \frac{1}{2}$. Applying the Bernstein inequality now yields the stated bound (20)(i).

5.1.2 Proof of part (b)

We prove the claim by constructing a packing set that satisfies our general requirements as well as also lies within the SST model and all parametric models, and subsequently using the packing set in an application of Fano's inequality. We then carefully bound the Kullback-Leibler divergence between the probability distributions on the outcomes induced by any pair of elements in the packing set in order to obtain a tractable bound.

In more detail, the symmetry of the problem allows us to assume, without loss of generality, that $k \leq \frac{n}{2}$. We first construct an ensemble of $n-k+1$ different problems, and considering the problem of distinguishing between them. For each $a \in \{k, \ldots, n\}$, let us define the k-sized subset $\mathcal{S}^*[a] := \{1, \ldots, k-1\} \cup \{a\}$, and the associated matrix of pairwise

probabilities

$$M^a_{ij} := \begin{cases} \frac{1}{2} & \text{if } i,j \in \mathcal{S}^*[a], \text{ or } i,j \notin \mathcal{S}^*[a] \\ \frac{1}{2}+\delta & \text{if } i \in \mathcal{S}^*[a] \text{ and } j \notin \mathcal{S}^*[a] \\ \frac{1}{2}-\delta & \text{if } i \notin \mathcal{S}^*[a] \text{ and } j \in \mathcal{S}^*[a], \end{cases}$$

where $\delta \in (0, \frac{1}{2})$ is a parameter to be chosen. We use $\mathbb{P}^a$ to denote probabilities taken under pairwise comparisons drawn according to the model M^a.

One can verify that the construction above falls in the intersection of parametric models and the SST model. In the parametric case, this construction amounts to having the parameters associated to every item in $\mathcal{S}^*_k$ to have the same value, and those associated to every item in $[n]\backslash\mathcal{S}^*_k$ to have the same value. Also observe that for every such distribution $\mathbb{P}^a$, the associated k-separation threshold is $\Delta_k = \delta$.

Any given set of observations can be described by the collection of random variables $Y = \{Y^{(\ell)}_{ij}, j > i \in [n], \ell \in [r]\}$. When the true underlying model is $\mathbb{P}^a$, the random variable $Y^{(\ell)}_{ij}$ follows the distribution

$$Y^{(\ell)}_{ij} = \begin{cases} 0 & \text{with probability } 1-p \\ i & \text{with probability } pM^a_{ij} \\ j & \text{with probability } p(1-M^a_{ij}). \end{cases}$$

The random variables $\{Y^{(\ell)}_{ij}\}_{i,j\in[n],i<j,\ell\in[r]}$ are mutually independent, and the distribution $\mathbb{P}^a$ is a product distribution across pairs $\{i > j\}$ and repetitions $\ell \in [r]$.

Let $A \in \{k, \ldots, n\}$ follow a uniform distribution over the index set, and suppose that given $A = a$, our observations Y has components drawn according to the model $\mathbb{P}^a$. Consequently, the marginal distribution of Y is the mixture distribution $\frac{1}{n-k+1}\sum_{a=k}^n \mathbb{P}^a$ over all $(n-k+1)$ models. Based on observing Y, our goal is to recover the correct index $A = a$ of the underlying model, which is equivalent to recovering the planted subset $\mathcal{S}^*[a]$. We use the Fano bound (19) to lower bound the error bound associated with any test for this problem. In order to apply Fano's inequality, the following result provides control over the Kullback-Leibler divergence between any pair of probabilities involved.

Lemma 7 *For any distinct pair $a, b \in \{k, \ldots, n\}$, we have*

$$D_{\mathrm{KL}}(\mathbb{P}^a \| \mathbb{P}^b) \leq \frac{2npr}{\frac{1}{4\delta^2}-1}. \tag{23}$$

See the end of this section for the proof of this claim.

Given this bound on the Kullback-Leibler divergence, Fano's inequality (19) implies that any estimator ϕ of A has error probability lower bounded as

$$\mathbb{P}[\phi(Y) \neq A] \geq 1 - \frac{\frac{2npr}{\frac{1}{4\delta^2}-1} + \log 2}{\log(n-k+1)} \;\geq\; \frac{1}{7}.$$

Here the final inequality holds whenever $\delta \leq \frac{1}{7}\sqrt{\frac{\log n}{npr}}$, $p \geq \frac{\log n}{2nr}$, $n \geq 7$ and $k \leq \frac{n}{2}$. The condition $p \geq \frac{\log n}{2nr}$ also ensures that $\delta < \frac{1}{2}$ thereby ensuring that our construction is valid. It only remains to prove Lemma 7.

5.1.3 Proof of Lemma 7

Since the distributions $\mathbb{P}^a$ and $\mathbb{P}^b$ are formed by components that are independent across edges $i > j$ and repetitions $\ell \in [r]$, we have

$$D_{\mathrm{KL}}(\mathbb{P}^a \| \mathbb{P}^b) = \sum_{\ell\in[r]} \sum_{1\leq i<j\leq n} D_{\mathrm{KL}}(\mathbb{P}^a(X^{(\ell)}_{ij}) \| \mathbb{P}^b(X^{(\ell)}_{ij})) \;=\; r \sum_{1\leq i<j\leq n} D_{\mathrm{KL}}(\mathbb{P}^a(X^{(1)}_{ij}) \| \mathbb{P}^b(X^{(1)}_{ij})),$$

where the second equality follows since the r trials are all independent and identically distributed.

We now evaluate each individual term in right hand side of the above equation. Consider any $i, j \in [n]$. We divide our analysis into three disjoint cases:

Case I: Suppose that $i, j \in [n]\backslash\{a, b\}$. The distribution of $X^{(1)}_{ij}$ is identical across the distributions $\mathbb{P}^a$ and $\mathbb{P}^b$. As a result, we find that

$$D_{\mathrm{KL}}(\mathbb{P}^a(X^{(1)}_{ij}) \| \mathbb{P}^b(X^{(1)}_{ij})) = 0.$$

Case II: Suppose that $i = a,\ j \in [n]\backslash\{a, b\}$ or $i = b,\ j \in [n]\backslash\{a, b\}$. We then have

$$D_{\mathrm{KL}}(\mathbb{P}^a(X^{(1)}_{ij}) \| \mathbb{P}^b(X^{(1)}_{ij})) \leq p\frac{\delta^2}{(\frac{1}{2}-\delta)(\frac{1}{2}+\delta)}.$$

Case III: Suppose that $i = a,\ j = b$. We then have

$$D_{\mathrm{KL}}(\mathbb{P}^a(X^{(1)}_{ij}) \| \mathbb{P}^b(X^{(1)}_{ij})) \leq p\frac{(2\delta)^2}{(\frac{1}{2}-\delta)(\frac{1}{2}+\delta)}.$$

Combining the bounds from all three cases, we find that the KL divergence is upper bounded as

$$\frac{1}{r}D_{\mathrm{KL}}(\mathbb{P}^a \| \mathbb{P}^b) \leq 2(n-2)p\frac{\delta^2}{(\frac{1}{2}-\delta)(\frac{1}{2}+\delta)} + p\frac{(2\delta)^2}{(\frac{1}{2}-\delta)(\frac{1}{2}+\delta)}.$$

Some simple algebraic manipulations yield the claimed result.

5.2 Proof of Theorem 2

We now turn to the proof of Theorem 2. Beginning with the claim of sufficiency, it is easy to see that the ranking is correctly recovered whenever the top k items are correctly recovered for every value of $k \in [n]$. Consequently, one can apply the union bound to (10a) over all values of $k \in [n]$ and this gives the desired upper bound.

Now turning to the claim of necessity, we first introduce some notation to aid in subsequent discussion. Defining the parameter $\Delta_0 := \min_{j\in[n-1]}(\tau_{(j)} - \tau_{(j+1)})$, we have shown that the lower bound

$$\Delta_0 \geq 8\sqrt{\frac{\log n}{npr}}$$

is sufficient to guarantee exact recovery of the full ranking. Further, one must also have

$$\Delta_0 \leq \frac{1}{n-1}\sum_{j=1}^{n-1}(\tau_{(j)} - \tau_{(j+1)}) = \frac{1}{n-1}(\tau_{(1)} - \tau_{(n)}) \leq \frac{1}{n-1}.$$

Here we show that this pair of requirements is jointly tight up to constant factors, meaning that for any value of Δ_0 satisfying $\Delta_0 \leq \frac{1}{9}\sqrt{\frac{\log n}{npr}}$ and $\Delta_0 \leq \frac{1}{9}\frac{1}{n-1}$, there are instances where recovery of the underlying ranking fails with probability at least $\frac{1}{70}$ for any estimator.

Consider the following ensemble of $(n-1)$ different problems, indexed by $a \in [n-1]$. For every value of $a \in [n-1]$, define a permutation π^a of the n items as

$$\pi^a(i) = \begin{cases} i+1 & \text{if } i = a \\ i-1 & \text{if } i = a+1 \\ i & \text{otherwise.} \end{cases}$$

In words, the permutation π^a equals the identity permutation except for the swapping of items a and $(a+1)$. Define an associated matrix of pairwise-comparison probabilities M^a as

$$M^a_{ij} = \frac{1}{2} - (\pi^a(i) - \pi^a(j))\Delta_0,$$

and $M^a_{ji} = 1 - M^a_{ij}$. Let $\mathbb{P}^a$ denote the probabilities taken under pairwise comparisons drawn according to the model M^a. The condition $\Delta_0 \leq \frac{1}{9}\frac{1}{n-1}$ ensures that this construction is a valid probability distribution. One can then compute that under distribution $\mathbb{P}^a$, the score τ^a_i of any item i equals

$$\tau^a_i = \frac{1}{2} - \big(\pi^a(i) - \frac{n+1}{2}\big)\Delta_0.$$

One can also verify that for any $a \in [n-1]$, and any $i \in [n-1]$, we have

$$\tau^a_{\pi^a(i)} - \tau^a_{\pi^a(i+1)} = \Delta_0,$$

where we have used the fact that $\pi^a(\pi^a(i)) = i$. The requirement imposed by the hypothesis is thus satisfied.

We now use Fano's inequality (19) obtain the claimed lower bound. In order to apply this result, we first obtain an upper bound on the Kullback-Leibler divergence between the probability distributions of the observed data under any pair of problems constructed above.

Lemma 8 *For any distinct pair $a, b \in [n-1]$, we have*

$$D_{\mathrm{KL}}(\mathbb{P}^a \| \mathbb{P}^b) \leq 50npr\Delta_0^2.$$

See the end of this section for the proof of this claim.

Given this bound on the Kullback-Leibler divergence, the Fano bound (19) implies that any method ϕ for identifying the true ranking has error probability

$$\mathbb{P}[\phi(Y) \neq A] \geq 1 - \frac{50npr\Delta_0^2 + \log 2}{\log(n-1)} \geq \frac{1}{70},$$

where the final inequality holds whenever $\Delta_0 \leq \frac{1}{9}\sqrt{\frac{\log n}{npr}}$ and $n \geq 9$.

The only remaining detail is the proof of Lemma 8.

5.2.1 Proof of Lemma 8

Since the distributions $\mathbb{P}^a$ and $\mathbb{P}^b$ are formed by components that are independent across edges $i > j$ and repetitions $\ell \in [r]$, we have

$$D_{\mathrm{KL}}(\mathbb{P}^a \| \mathbb{P}^b) = \sum_{\ell \in [r]} \sum_{1 \leq i < j \leq n} D_{\mathrm{KL}}(\mathbb{P}^a(X^{(\ell)}_{ij}) \| \mathbb{P}^b(X^{(\ell)}_{ij})) \;=\; r \sum_{1 \leq i < j \leq n} D_{\mathrm{KL}}(\mathbb{P}^a(X^{(1)}_{ij}) \| \mathbb{P}^b(X^{(1)}_{ij})),$$

where the second equality follows since the r trials are all independent and identically distributed.

We now evaluate each individual term in right hand side of the above equation. Consider any $i, j \in [n]$. We divide our analysis into three disjoint cases:

Case I: Suppose that $i, j \in [n] \backslash \{a, a+1, b, b+1\}$. The distribution of $X^{(1)}_{ij}$ is identical across the distributions $\mathbb{P}^a$ and $\mathbb{P}^b$. As a result, we find that

$$D_{\mathrm{KL}}(\mathbb{P}^a(X^{(1)}_{ij}) \| \mathbb{P}^b(X^{(1)}_{ij})) = 0.$$

Case II: Alternatively, suppose $i \in \{a, a+1, b, b+1\}$ and $j \in [n] \backslash \{a, a+1, b, b+1\}$ or if $j \in \{a, a+1, b, b+1\}$ and $i \in [n] \backslash \{a, a+1, b, b+1\}$. Then we have

$$D_{\mathrm{KL}}(\mathbb{P}^a(X^{(1)}_{ij}) \| \mathbb{P}^b(X^{(1)}_{ij})) \leq 5p\Delta_0^2,$$

where we have used the fact that $\mathbb{P}^a(X^{(1)}_{ij})$ and $\mathbb{P}^b(X^{(1)}_{ij})$ both take values in $[\frac{7}{18}, \frac{11}{18}]$ since $\Delta_0 \leq \frac{1}{9}\frac{1}{n-1}$.

Case III: Otherwise, suppose that both $i, j \in \{a, a+1, b, b+1\}$. Then we have

$$D_{\mathrm{KL}}(\mathbb{P}^a(X^{(1)}_{ij}) \| \mathbb{P}^b(X^{(1)}_{ij})) \leq 20p\Delta_0^2.$$

Combining the bounds from the three cases, we find that the KL divergence is upper bounded as

$$\frac{1}{r} D_{\mathrm{KL}}(\mathbb{P}^a \| \mathbb{P}^b) \leq 40(n-4)p\Delta_0^2 + 240p\Delta_0^2 \leq 50np\Delta_0^2,$$

where we have used the assumption $n \geq 9$ to obtain the final inequality.

5.3 Proof of Theorem 3

We now turn to the proof of Theorem 3, beginning with part (a).

5.3.1 Proof of part (a)

Without loss of generality, we can assume that the true underlying ranking is the identity ranking, that is, item i is ranked at position i for every $i \in [n]$. Given the the lower bound $\alpha \geq 8$ is satisfied, Theorem 1 ensures that with probability at least $1 - n^{-16}$, the counting estimator $\widetilde{\mathcal{S}}_k$ ranks every item in $\{1, \ldots, k-h\}$ higher than every item in the set $\{k+h+1, \ldots, n\}$. Thus, we are guaranteed that either $\widetilde{\mathcal{S}}_k \subseteq [k+h]$ and/or $[k-h] \subseteq \widetilde{\mathcal{S}}_k$. One can verify either case leads to $|\widetilde{\mathcal{S}}_k \cap [k]| \geq k-h$, thereby proving the claimed result.

5.3.2 Proof of part (b)

At a higher level, the crux of this proof is the construction of a packing set of pairwise comparison probability matrices, where every element of the set is also guaranteed to lie in the parametric classes and the SST class. The packing set is constructed via a careful application of a coding theoretic result due to Levenshtein (1971), such that the pairwise Kullback-Leibler divergence is small but the pairwise Hamming error is large enough, and that the packing set is also large enough. An application of Fano's inequality and some algebra yields the claimed result.

In more detail, we assume without loss of generality that $k \leq \frac{n}{2}$. (Otherwise, one can equivalently study the problem of recovering the last k items.) Since the case $h = 0$ is already covered by Theorem 1(b), we may also assume that $h \geq 1$.

The proof involves construction of $L \geq 1$ sets of probability matrices $\{M^a\}_{a \in [L]}$ of the pairwise comparisons with the following two properties:

(i) For every $a \in [L]$, let $S_k^a \subseteq [n]$ denote the set of the top k items under the a^{th} set of distributions. Then for every k-sized set $S \in [n]$,

$$\sum_{a=1}^{L} \mathbf{1}\{D_{\mathrm{H}}(S, S_k^a) \leq 2h\} \leq 1.$$

(ii) If the underlying distribution a is chosen uniformly at random from this set of L distributions, then any estimator that attempts to identify the underlying distribution $a \in [L]$ errs with probability at least $\frac{1}{7}$.

Now consider any estimator $\widehat{\mathcal{S}}_k$ for identifying the top k items $\mathcal{S}_k^*$. Given property (i), whenever the estimator is successful under the Hamming error requirement $D_{\mathrm{H}}(\widehat{\mathcal{S}}_k, \mathcal{S}_k^*) \leq 2h$, it must be able to uniquely identify the index $a \in [L]$ of the underlying distribution of pairwise comparison probabilities. However, property (ii) mandates that any estimator for identifying the underlying distribution errs with a probability at least $\frac{1}{7}$. Assuming that such sets of probability distributions satisfying these two properties exist, putting these results together yields the claimed result.

We now proceed to construct probability distributions satisfying the two aforementioned properties. Consider any positive number Δ_0 satisfying the upper bound

$$\Delta_0 \leq \frac{1}{14}\sqrt{\frac{\nu_1 \nu_2 \log n}{npr}}. \tag{24}$$

The L matrices $\{M^a\}_{a \in [L]}$ of probability distributions we construct differ only in a permutation of their rows and columns, and modulo this permutation, have identical values. In other words, these L distributions differ only in the identities of the n items and the values of the pairwise-comparison probabilities $M^a_{(i)(j)}$ among the ordered sequence of the n items are identical across all distributions $a \in [L]$.

For any ordering $(1), \ldots, (n)$ of the n items, for every $a \in [L]$, set

$$M^a_{(i)(j)} = \begin{cases} \frac{1}{2} + \Delta_0 & \text{if } i \in [k] \text{ and } j \notin [k] \\ \frac{1}{2} - \Delta_0 & \text{if } i \notin [k] \text{ and } j \in [k] \\ \frac{1}{2} & \text{otherwise.} \end{cases} \tag{25}$$

Note that the upper bound (24) on Δ_0, coupled with the assumption $p \geq \sqrt{\frac{\log n}{2nr}}$, ensures that $\Delta_0 < \frac{1}{3}$ and hence that our definition (25) leads to a valid set of probabilities. Given this construction, the scores of the n items are $\tau_{(1)} = \cdots = \tau_{(k)} = \tau_{(k+1)} + \Delta_0 = \cdots = \tau_{(n)} + \Delta_0$. The bound (24) ensures that the condition $\alpha \leq \frac{\sqrt{\nu_1 \nu_2}}{14}$ required by the hypothesis of the theorem is satisfied.

It remains to specify the ordering of the n items in each set of probability distributions. This specification relies on the following lemma, that in turn uses a coding-theoretic result due to Levenshtein (1971). It applies in the regime $2h \leq \frac{1}{1+\nu_2} \min\{n^{1-\nu_1}, k, n-k\}$ for some constants $\nu_1 \in (0,1)$ and $\nu_2 \in (0,1)$, and when n is larger than a (ν_1, ν_2)-dependent constant. For any pair of binary vectors b, b' of the same length, we define the Hamming error as $D_{\mathrm{H}}(b, b') = \sum_i \mathbf{1}\{b_i \neq b'_i\}$. We also let $\mathbf{0}$ denote the all-zero vector.

Lemma 9 *Under the previously given conditions, there exists a subset $\{b^1, \ldots, b^L\} \subseteq \{0,1\}^{n/2}$ with cardinality $L \geq e^{\frac{9}{10}\nu_1\nu_2 h \log n}$, such that*

$$D_{\mathrm{H}}(b^j, \mathbf{0}) = 2(1+\nu_2)h, \quad \textit{and} \quad D_{\mathrm{H}}(b^j, b^\ell) > 4h \quad \textit{for all } j \neq \ell \in [L].$$

We prove this lemma at the end of this section. Given this lemma, we now complete the proof of the theorem. Map the $\frac{n}{2}$ items $\{\frac{n}{2}+1, \ldots, n\}$ to the $\frac{n}{2}$ bits in each of the strings given by Lemma 9. For each $\ell \in [e^{\frac{9}{10}\nu_1\nu_2 h \log n}]$, let B_ℓ denote the $2(1+\nu_2)h$-sized subset of $\{\frac{n}{2}+1, \ldots, n\}$ corresponding to the $2(1+\nu_2)h$ positions equaling 1 in the ℓ^{th} string. Also define sets $A_\ell = \{1, \ldots, k - 2(1+\nu_2)h\}$ and $C_\ell = [n] \backslash (A_\ell \cup B_\ell)$. We note that this construction is valid since $2h \leq \frac{1}{1+\nu_2} k$.

We now construct $L = e^{\frac{9}{10}\nu_1\nu_2 h \log n}$ sets of pairwise comparison probability distributions $M^1, \ldots, M^L$ and show that these sets satisfy the two required properties. As mentioned earlier, each matrix of comparison-probabilities M^ℓ takes values as given in (25), but differs in the underlying ordering of the n items. In particular, associate the set $\ell \in [L]$ of distributions to any ordering of the n items that ranks every item in A_ℓ higher than every

item in B_ℓ, and every item in B_ℓ in turn higher than every item in C_ℓ. Then for any ℓ, the set of top k items is given by $A_\ell \cup B_\ell$. From the guarantees provided by Lemma 9, for any distinct $\ell, m \in [L]$, we have $D_{\mathrm{H}}(A_\ell \cup B_\ell, A_m \cup B_m) \geq 4h+1$. This construction consequently satisfies the first required property.

We now show that the construction also satisfies the second property: namely, it is difficult to identify the true index. We do so using Fano's inequality (19), for which we denote the probability distribution of the observations due to any matrix M^ℓ, $\ell \in [L]$, as $\mathbb{P}^\ell$.

We first derive an upper bound on the Kullback-Leibler divergence between any two distributions $\mathbb{P}^\ell$ and $\mathbb{P}^m$ of the observations. Observe that $[M^\ell]_{ij} \neq [M^m]_{ij}$ only if $i \in B_\ell \cup B_m$ or $j \in B_\ell \cup B_m$. In this case, we have $D_{\mathrm{KL}}([M^\ell]_{ij} \| [M^m]_{ij}) \leq \frac{4\Delta_0^2}{\frac{1}{4}-\Delta_0^2}$. Since both sets B_ℓ and B_m have a cardinality of $2(1+\nu_2)h$, aggregating over all possible observations across all pairs, we obtain that

$$D_{\mathrm{KL}}(\mathbb{P}^\ell \| \mathbb{P}^m) \leq 4(1+\nu_2)hnpr \frac{4\Delta_0^2}{\frac{1}{4}-\Delta_0^2}. \tag{26}$$

In the regime $p \geq \frac{\log n}{2nr}$ and $\Delta_0 \leq \frac{1}{14}\sqrt{\frac{\nu_1\nu_2 \log n}{npr}}$, we have $\Delta_0 \leq \frac{1}{14\sqrt{2}}$. Substituting the inequality $\Delta_0 \leq \frac{1}{14}\sqrt{\frac{\nu_1 \log n}{npr}}$ in the numerator and $\frac{1}{4} - \Delta_0^2 \geq \frac{1}{4} - \big(\frac{1}{14\sqrt{2}}\big)^2$ in the denominator of the right hand side of the bound (26), we find that

$$D_{\mathrm{KL}}(\mathbb{P}^\ell \| \mathbb{P}^m) \leq \frac{3}{4}\nu_1\nu_2 h \log n.$$

Now suppose that we draw Y from some distribution chosen uniformly at random from $\{\mathbb{P}^1, \ldots, \mathbb{P}^L\}$. Applying Fano's inequality (19) ensures that any test ϕ for estimating the index A of the chosen distribution must have error probability lower bounded as

$$\mathbb{P}\big[\phi(Y) \neq A\big] \geq \left(1 - \frac{\frac{3}{4}\nu_1\nu_2 h \log n + \log 2}{\frac{9}{10}\,\nu_1\nu_2 h \log n}\right) \geq \frac{1}{7}.$$

Here the final inequality holds as long as n is larger than some universal constant.

5.3.3 Proof of Lemma 9

We divide the proof into two cases depending on the value of h.

Case I: $h \geq \frac{1}{2\nu_1\nu_2}$: Let L denote the number of binary strings of length m_0 such that each has a Hamming weight w_0 and each pair has a Hamming distance at least d_0. It is known (Levenshtein, 1971; Jiang and Vardy, 2004) that L can be lower bounded as:

$$L \geq \frac{\binom{m_0}{w_0}}{\sum_{i=0}^{\lfloor \frac{d_0-1}{2} \rfloor} \binom{w_0}{j}\binom{m_0-w_0}{j}} \geq \frac{\big(\frac{m_0}{w_0}\big)^{w_0}}{\frac{d_0+1}{2}\big(\frac{ew_0}{\min\{d_0,w_0\}/2}\big)^{\min\{d_0,w_0\}/2}\big(\frac{em_0}{\min\{d_0,m_0\}/2}\big)^{\min\{d_0,m_0\}/2}}.$$

Note that for the setting at hand, we have $m_0 = \frac{n}{2}$, $w_0 = 2(1+\nu_2)h$ and $d_0 = 4h+1$. Since $\nu_1 \in (0,1)$ and $\nu_2 \in (0,1)$, we have the chain of inequalities

$$w_0 < d_0 \leq 4n^{1-\nu_1} \overset{(i)}{<} \frac{n}{2} = m_0,$$

where the inequality (i) holds when n is large enough. These relations allow for the simplification:

$$\begin{aligned}\log L &\geq \log\left\{\frac{\big(\frac{m_0}{w_0}\big)^{w_0}}{\frac{d_0+1}{2}\big(\frac{ew_0}{w_0/2}\big)^{w_0/2}\big(\frac{em_0}{d_0/2}\big)^{d_0/2}}\right\} \\ &= (w_0 - d_0/2)\log m_0 - w_0 \log w_0 + \frac{d_0}{2}\log d_0 - \frac{d_0+w_0}{2}\log(2e) - \log((d_0+1)/2).\end{aligned}$$

Substituting the values of w_0, d_0 and m_0 and then simplifying yields

$$\begin{aligned}\log L &\geq (2\nu_2 h - \frac{1}{2})\log\frac{n}{2} - 2(1+\nu_2)h\log(2(1+\nu_2)h) + (2h+\frac{1}{2})\log(4h+1) \\ &\quad - (((3+\nu_2)h) + \frac{1}{2})\log(2e) - \log(2h+1) \\ &\geq (2\nu_2 h - \frac{1}{2})\log\frac{n}{2} - 2\nu_2 h\log(2(1+\nu_2)h) - c_1' h,\end{aligned}$$

where c_1' is a constant whose value depends only on (ν_1, ν_2). In the regime $\frac{1}{\nu_1\nu_2} \leq 2h \leq \frac{n^{1-\nu_1}}{1+\nu_2}$, some algebraic manipulations then yield

$$\log L \geq (2\nu_1\nu_2 h - \frac{1}{2})\log\frac{n}{2} - c_2' h \geq \nu_1\nu_2 h(\log n - \log 2 - c_3') \geq \frac{9}{10}\nu_1\nu_2 h \log n,$$

where the final inequality holds when n is large enough, and where c_2' and c_3' are (ν_1, ν_2)-dependent positive constants.

Case II: $h < \frac{1}{2\nu_1\nu_2}$ Consider a partition of the $\frac{n}{2}$ bits into $\frac{n}{4(1+\nu_2)h}$ sets of size $2(1+\nu_2)h$ each. Define an associated set of $\frac{n}{4(1+\nu_2)h}$ binary strings, each of length $\frac{n}{2}$, with the i^{th} string having ones in the positions corresponding to the i^{th} set in the partition and zeros elsewhere. Then each of these strings have a Hamming weight of $2(1+\nu_2)h$, and every pair has a Hamming distance at least $4(1+\nu_2)h > 4h$. The total number of such strings equals

$$\exp\big(\log\frac{n}{4(1+\nu_2)h}\big) \overset{(i)}{\geq} \exp\big(\log n - \log(\frac{2(1+\nu_2)}{\nu_1\nu_2})\big) \overset{(ii)}{\geq} \exp\big(\frac{9}{10}\log n\big) \overset{(iii)}{>} \exp\big(1.8\nu_1\nu_2 h \log n\big),$$

where the inequalities (i) and (iii) are a result of operating in the regime $h < \frac{1}{2\nu_1\nu_2}$ and the inequality (ii) assumes that n is greater than a (ν_1, ν_2)-dependent constant.

5.4 Proof of Theorem 6

We now turn to the proof of Theorem 6.

5.4.1 Proof of part (a)

For every $i \in [n]$, let (i) denote the item ranked i according to their latent scores, as defined in equation (2). Recall from the proof of Theorem 1 that for any $u < v \in [n]$, the condition

$$\tau_{(u)} - \tau_{(v)} \geq 8\sqrt{\frac{\log n}{npr}}$$

ensures that with probability at least $1 - n^{-14}$, every item in the set $\{(1), \ldots, (u)\}$ wins more comparisons than every item in the set $\{(v), \ldots, (n)\}$. Consequently, if the set $\widetilde{\mathcal{S}}_k$ contains any item in $\{(v), \ldots, (n)\}$, then it must contain the entire set $\{(1), \ldots, (u)\}$. In other words, at least one of the following must be true: either $\{(1), \ldots, (u)\} \subseteq \widetilde{\mathcal{S}}_k$ or $\widetilde{\mathcal{S}}_k \subseteq \{(1), \ldots, (v-1)\}$. Consequently, in the regime $v = k + t - u + 1$ for any $1 \leq u \leq k$ and $u \leq t \leq n$, we have that

$$|\widetilde{\mathcal{S}}_k \cap \{(1), \ldots, (t)\}| \geq u. \tag{27}$$

Now consider any $b \in [\beta]$ that satisfies the condition

$$\min_{j \in [k]} (\tau_{(j)} - \tau_{(k + t_j^b - j + 1)}) \geq 8\sqrt{\frac{\log n}{npr}}.$$

For any $j \in [k]$, setting $u = j$ and $v = (k + t_j^b - j + 1)$ in (27), and applying the union bound over all values of $j \in [k]$ yields that

$$|\widetilde{\mathcal{S}}_k \cap \{(1), \ldots, (t_j^b)\}| \geq j \quad \text{for every } j \in [k],$$

with probability at least $1 - n^{-13}$. Consequently, we have that

$$\mathbb{P}\big(\widetilde{\mathcal{S}}_k \in \Lambda(T_b)\big) \geq 1 - n^{-13},$$

completing the proof of the claim.

5.4.2 Proof of part (b)

In the regime $t^b_{\mu_2 k} \leq \frac{n}{2}$ for every $b \in [\beta]$, it suffices to show that any estimator $\widehat{\mathcal{S}}_k$ will incur an error lower bounded as

$$\mathbb{P}\big(|\widehat{\mathcal{S}}_k \cap \{(1), \ldots, (n/2)\}| < \mu_2 k\big) \geq \frac{1}{15},$$

where (i) denotes the item ranked i according to their latent scores according to equation (2).

Our proof relies on the result and proof of the Hamming error case analyzed in Theorem 3(b). To this end, let us set the parameter h of Theorem 3(b) as $h = 2(1 - \mu_2)k$. We claim that this value of h lies in the regime $h \leq \frac{1}{2(1+\nu_2)} \min\{k, n-k, n^{1-\nu_1}\}$ for some values $\nu_1 \in (0,1)$ and $\nu_2 \in (0,1)$, as required by Theorem 3(b). This claim follows from the fact that

$$h = 2(1 - \mu_2)k \leq \frac{1}{2(1+\nu_2)} k,$$

for $\nu_2 = \min\{\frac{1}{4(1-\mu_2)} - 1, \frac{1}{2}\} \in (0,1)$. Furthermore,

$$h = 2(1-\mu_2)k \overset{(i)}{\leq} \frac{n^{1-\mu_1}}{4} \overset{(ii)}{\leq} \frac{1}{2(1+\nu_2)} n^{1-\nu_1}$$

for $\nu_1 = \frac{9}{10}\mu_1 \in (0,1)$, where (i) is a result of our assumption $8(1-\mu_2)k \leq n^{1-\mu_1}$ and (ii) holds when n is large enough. This assumption also implies that $k \leq n - k$ for a large enough value of n. We have now verified operation in the regime required by Theorem 3(b).

The construction in the proof of Theorem 3 is based on setting

$$\tau_{(1)} = \cdots \tau_{(k)} = \tau_{(k+1)} + \Delta_0 = \cdots = \tau_{(n)} + \Delta_0,$$

for any real number Δ_0 in the interval $\Big(0, \frac{1}{14}\sqrt{\frac{\nu_1\nu_2 \log n}{npr}}\Big]$. This condition is also satisfied in our construction due to the assumed upper bound $\alpha \leq \frac{1}{15}\sqrt{\mu_1 \min\big\{\frac{1}{4(1-\mu_2)-1}, \frac{1}{2}\big\}}$. Consequently, the result of Theorem 3(b) implies that in this setting, any estimator $\widehat{\mathcal{S}}_k$ will incur a Hamming error greater than $h = 2(1-\mu_2)k$ with probability at least $\frac{1}{7}$, or equivalently,

$$\mathbb{P}\big(|\widehat{\mathcal{S}}_k \cap \{(1), \ldots, (k)\}| < (2\mu_2 - 1)k\big) \geq \frac{1}{7}.$$

Under this event, the estimator $\widehat{\mathcal{S}}_k$ contains at most $(2\mu_2 - 1)k - 1$ items from the set of top k items. In order to ensure it gets at least $\mu_2 k$ items from $\{(1), \ldots, (n/2)\}$, the remaining $2(1-\mu_2)k + 1$ chosen items must have at least $(1-\mu_2)k + 1$ items from $\{(k+1), \ldots, (n/2)\}$. However, in the construction, items $(k+1), \ldots, (n)$ are indistinguishable from each other, and hence by symmetry these $2(1-\mu_2)k+1$ chosen items must contain at least $(1-\mu_2)k+1$ items from the set $\{(n/2+1), \ldots, (n)\}$ with probability at least $\frac{1}{2}$. Putting these arguments together, we obtain that under this construction, any estimator $\widehat{\mathcal{S}}_k$ has error probability lower bounded as

$$\mathbb{P}\big(|\widehat{\mathcal{S}}_k \cap \{(1), \ldots, (n/2)\}| < \mu_2 k\big) \geq \frac{1}{14}. \tag{28}$$

It remains to deal with a subtle technicality. The construction above involves items $(k+1), \ldots, (n)$ with identical scores. Recall that in the definition of the user-defined requirement, in case of multiple items with identical scores, we considered the choice of either of such items as valid. The following lemma helps overcome this issue.

Lemma 10 *Consider any two $(n \times n)$ matrices M^a and M^b of pairwise probabilities such that*

$$\max_{(i,j) \in [n]^2} |[M^a]_{ij} - [M^b]_{ij}| \leq \epsilon, \tag{29a}$$

for some $\epsilon \in [0,1]$. Then for any k-sized sets of items $T_1, \ldots, T_\beta \subseteq [n]$*, and any estimator $\widehat{\mathcal{S}}_k$, we have*

$$|\, \mathbb{P}_{M^a}(\widehat{\mathcal{S}}_k \in \{T_1, \ldots, T_\beta\}) - \mathbb{P}_{M^b}(\widehat{\mathcal{S}}_k \in \{T_1, \ldots, T_\beta\}) \,| \leq 6^{n^2 r}\epsilon. \tag{29b}$$

See Section 5.4.3 for the proof of this claim.

Now consider an $(n \times n)$ pairwise probability matrix M' whose entries take values

$$M'_{(i)(j)} = \begin{cases} \frac{1}{2} + \Delta_0 + \epsilon & \text{if } i \in [k] \text{ and } j \in [n] \backslash [n/2] \\ \frac{1}{2} + \Delta_0 & \text{if } i \in [k] \text{ and } j \in [n/2] \backslash [k] \\ \frac{1}{2} + \epsilon & \text{if } i \in [n/2] \backslash [k] \text{ and } j \in [n] \backslash [n/2] \\ \frac{1}{2} & \text{otherwise,} \end{cases}$$

and $M'_{ji} = 1 - M'_{ij}$, whenever $i \leq j$. Set $\epsilon = 7^{-n^2 r}$.

One can verify that under the probability matrix M', the scores of the n items satisfy the relations

$$\tau_{(1)} = \cdots = \tau_{(k)} = \tau_{(k+1)} + \Delta_0 = \cdots = \tau_{(n/2)} + \Delta_0 = \tau_{(n/2+1)} + \Delta_0 + \epsilon = \cdots = \tau_{(n)} + \Delta_0 + \epsilon.$$

The set of items $\{(1), \ldots, (n/2)\}$ are thus explicitly distinguished from the items $\{(n/2+1), \ldots, (n)\}$. We now call upon Lemma 10 with $M^a = M'$, and M^b as the matrix of probabilities constructed in the proof of Theorem 3, where both sets have the same ordering of the items. This assignment is valid given that $\Delta_0 < \frac{1}{3}$ and $\epsilon = 7^{-n^2 r}$. Lemma 10 then implies that any estimator that is $\mathfrak{S}$-respecting with probability at least $1 - \frac{1}{15}$ under M^b must also be $\mathfrak{S}$-respecting with probability at least $1 - \frac{1}{14.5}$ under M^a. But by equation (28), the latter condition is impossible, which implies our claimed lower bound.

5.4.3 Proof of Lemma 10

Let $\mathbb{P}^a$ and $\mathbb{P}^b$ denote the probabilities induced by the matrices M^a and M^b respectively. Consider any fixed observation $Y_1 \subseteq \{0, 1, \phi\}^{n \times n \times r}$, where ϕ denotes the absence of an observation. Let $\mathbb{P}^a(Y = Y_1)$ and $\mathbb{P}^b(Y = Y_1)$ denote the probabilities of observing Y_1 under $\mathbb{P}^a$ and $\mathbb{P}^b$, respectively. Given the bounds (29a), some algebra leads to

$$\begin{aligned}
| \, \mathbb{P}^a(Y = Y_1) - \mathbb{P}^b(Y = Y_1) \, | &\leq \max_{u \in [0, 1-\epsilon]^{n^2 r}} \Big(\prod_{i=1}^{n^2 r} (u_i + \epsilon) - \prod_{i=1}^{n^2 r} u_i \Big) \\
&\leq \max_{u \in [0, 1-\epsilon]^{n^2 r}} \Big(u_{n^2 r} \big(\prod_{i=1}^{n^2 r - 1} (u_i + \epsilon) - \prod_{i=1}^{n^2 r - 1} u_i \big) + \epsilon \Big) \\
&\vdots \\
&\leq n^2 r \epsilon. \qquad (30)
\end{aligned}$$

Now consider any estimator $\widehat{\mathcal{S}}_k$, which is permitted to be randomized. Let $L \leq 3^{n^2 r}$ denote the total number of possible values of the observation Y, and let $\{Y_1, \ldots, Y_L\} = \{0, 1, \phi\}^{n \times n \times r}$ denote the set of all possible valid values of the observation. For each $i \in [L]$, let $q_i \in [0, 1]$ denote the probability that the estimator $\widehat{\mathcal{S}}_k$ succeeds in satisfying the given requirement when the data observed equals Y_i. (Recall that the given requirement is in terms of the actual items and not their positions.) Then we have

$$\begin{aligned}
\big| \mathbb{P}^1(\widehat{\mathcal{S}}_k \in \{T_1, \ldots, T_\beta\}) - \mathbb{P}^2(\widehat{\mathcal{S}}_k \in \{T_1, \ldots, T_\beta\}) \big| &= \big| \sum_{i=1}^{L} \mathbb{P}^1(Y = Y_i) q_i - \sum_{i=1}^{L} \mathbb{P}^2(Y = Y_i) q_i \big| \\
&\leq \sum_{i=1}^{L} | \, \mathbb{P}^1(Y = Y_i) - \mathbb{P}^2(Y = Y_i) \, | \, q_i \\
&\overset{(i)}{\leq} \sum_{i=1}^{L} n^2 r \epsilon q_i \overset{(ii)}{\leq} 6^{n^2 r} \epsilon,
\end{aligned}$$

as claimed, where step (i) follows from our earlier bound (30) and step (ii) uses the bounds $L \leq 3^{n^2 r}$ and $n^2 r \leq 2^{n^2 r}$.

6. Discussion

In this paper, we analyzed the problem of recovering the k most highly ranked items based on observing noisy comparisons. We proved that an algorithm that simply selects the items that win the maximum number of comparisons is, up to constant factors, an information-theoretically optimal procedure. Our results also extend to recovering the entire ranking of the items. The results of this paper thus underscore the philosophy of *Occam's razor* that the simplest answer is often correct.

Empirical evaluations reveal the superior performance of the counting algorithm that we analyzed through our "permutation-based" approach as compared to the Spectral MLE algorithm. The intuition is that Spectral MLE is too tied to the restrictive parameter-based model and the model mismatch in this crowdsourcing data causes the high amount of error. On the other hand, the robustness and accuracy guarantees of the count estimator due to our permutation-based approach carry over to practice. More generally, parameter-based models are popular in many applications in part because they are quite intuitive to write down, and in part because they are sometimes analytically more tractable. However, instead if one were to consider rich enough models like permutation-based models then they may yield a broader perspective and richer insights into the problem that can lead to superior results (Shah, 2017, Chapter 1, Part 1).

There are number of open questions suggested by our work. The notion of allowed sets introduced in this paper apply to recovery of k-sized subsets of the items; such a formulation and associated results may apply to recovery of partial or total orderings of the items. The observation model considered here is based on a random number of observations for all pairs of comparisons. It would be interesting to extend our results to cases in which only specific subsets of pairs are observed, and particularly when these pairs are chosen adversarially. A parallel line of literature (e.g., Kaufmann and Kalyanakrishnan, 2013; Busa-Fekete et al., 2013; Jamieson et al., 2015; Heckel et al., 2016) studies settings in which the pairs to be compared can be chosen sequentially in a data-dependent manner, but to the best of our knowledge, this line of literature considers only the metric of exact recovery of the top k items. It is of interest to investigate the Hamming and allowed set recovery problems in such an active setting. Finally, it will also be useful to obtain analogous results for ranking problems where the identity of the person making the comparison is known and influences the outcomes of the comparison, for instance, in applications of peer-grading (Shah et al., 2013; Song et al., 2017) and peer-reviews (Shah et al., 2017c).

Acknowledgments

We would like to thank the AE Sujay Sanghavi for the efficient handling of the paper and the anonymous reviewers for their valuable comments. This work was partially supported by National Science Foundation Grant NSF-DMS-1612948; DOD Advanced Research Projects Agency W911NF-16-1-0552 and Office of Naval Research grant DOD ONR-N00014. In addition, NBS was supported in part by a Microsoft Research PhD fellowship.

References

Ammar, A. and Shah, D. Efficient rank aggregation using partial data. In *ACM SIGMETRICS Performance Evaluation Review*, 2012.

Babcock, B. and Olston, C. Distributed top-k monitoring. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, 2003.

Ballinger, T. P. and Wilcox, N. Decisions, error and heterogeneity. *The Economic Journal*, 107(443):1090–1105, 1997.

Boucheron, S., Lugosi, G., and Massart, P. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford University Press, 2013.

Bradley, R. and Terry, M. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 1952.

Braverman, M. and Mossel, E. Noisy sorting without resampling. In *Proc. ACM-SIAM symposium on Discrete algorithms*, 2008.

Busa-Fekete, R., Szorenyi, B., Cheng, W., Weng, P., and Hüllermeier, E. Top-k selection based on adaptive sampling of noisy preferences. In *International Conference on Machine Learning*, 2013.

Carmel, D., Cohen, D., Fagin, R., Farchi, E., Herscovici, M., Maarek, Y. S., and Soffer, A. Static index pruning for information retrieval systems. In *ACM SIGIR conference on Research and development in information retrieval*, 2001.

Carpenter, A., Jones, T., Lamprecht, M., Clarke, C., Kang, I., Friman, O., Guertin, D., Chang, J., Lindquist, R., Moffat, J., and Golland, P. Cellprofiler: image analysis software for identifying and quantifying cell phenotypes. *Genome biology*, 7(10):R100, 2006.

Chatterjee, S. Matrix estimation by universal singular value thresholding. *The Annals of Statistics*, 43(1):177–214, 2014.

Chen, Y. and Suh, C. Spectral MLE: Top-k rank aggregation from pairwise comparisons. In *International Conference on Machine Learning*, 2015.

Cover, T. M. and Thomas, J. A. *Elements of information theory*. John Wiley & Sons, 2012.

Davidson, D. and Marschak, J. Experimental tests of a stochastic decision theory. *Measurement: Definitions and theories*, 1959.

de Borda, J. C. Mémoire sur les élections au scrutin. *Histoire de l\'Academie Royale des Sciences*, 1781.

Ding, W., Ishwar, P., and Saligrama, V. A topic modeling approach to ranking. In *Conference on Artificial Intelligence and Statistics*, 2015.

Erdős, P. and Rényi, A. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5:17–61, 1960.

Eriksson, B. Learning to top-k search using pairwise comparisons. In *Conference on Artificial Intelligence and Statistics*, 2013.

Fagin, R., Lotem, A., and Naor, M. Optimal aggregation algorithms for middleware. *Journal of computer and system sciences*, 66(4):614–656, 2003.

Hajek, B., Oh, S., and Xu, J. Minimax-optimal inference from partial rankings. In *Advances in Neural Information Processing Systems*, 2014.

Heckel, R., Shah, N. B., Ramchandran, K., and Wainwright, M. J. Active ranking from pairwise comparisons and when parametric assumptions don't help. *arxiv:1606.08842*, 2016.

Hunter, D. MM algorithms for generalized Bradley-Terry models. *Annals of Statistics*, 2004.

Ilyas, I. F., Beskales, G., and Soliman, M. A. A survey of top-k query processing techniques in relational database systems. *ACM Computing Surveys*, 2008.

Jagabathula, S. and Shah, D. Inferring rankings under constrained sensing. In *Advances in Neural Information Processing Systems*, 2008.

Jamieson, K., Katariya, S., Deshpande, A., and Nowak, R. Sparse dueling bandits. *arXiv preprint arXiv:1502.00133*, 2015.

Jiang, T. and Vardy, A. Asymptotic improvement of the Gilbert-Varshamov bound on the size of binary codes. *IEEE Transactions on Information Theory*, 2004.

Kaufmann, E. and Kalyanakrishnan, S. Information complexity in bandit subset selection. In *Conference on Learning Theory*, 2013.

Kenyon-Mathieu, C. and Schudy, W. How to rank with few errors. In *Symposium on Theory of computing (STOC)*. ACM, 2007.

Kimelfeld, B. and Sagiv, Y. Finding and approximating top-k answers in keyword proximity search. In *Symposium on Principles of database systems*, 2006.

Levenshtein, V. I. Upper-bound estimates for fixed-weight codes. *Problemy Peredachi Informatsii*, 7(4):3–12, 1971.

Luce, R. D. *Individual choice behavior: A theoretical analysis*. New York: Wiley, 1959.

McLaughlin, D. H. and Luce, R. D. Stochastic transitivity and cancellation of preferences between bitter-sweet solutions. *Psychonomic Science*, 1965.

Metwally, A., Agrawal, D., and El Abbadi, A. Efficient computation of frequent and top-k elements in data streams. In *Database Theory-ICDT*. 2005.

Michel, S., Triantafillou, P., and Weikum, G. Klee: A framework for distributed top-k query algorithms. In *International conference on Very large data bases*, 2005.

Mitliagkas, I., Gopalan, A., Caramanis, C., and Vishwanath, S. User rankings from comparisons: Learning permutations in high dimensions. In *Allerton Conference on Communication, Control, and Computing*, 2011.

Negahban, S., Oh, S., and Shah, D. Iterative ranking from pair-wise comparisons. In *Advances in Neural Information Processing Systems*, 2012.

Rajkumar, A. and Agarwal, S. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In *International Conference on Machine Learning*, 2014.

Rajkumar, A., Ghoshal, S., Lim, L.-H., and Agarwal, S. Ranking from stochastic pairwise preferences: Recovering Condorcet winners and tournament solution sets at the top. In *International Conference on Machine Learning*, 2015.

Shah, N. B. *Learning From People*. PhD thesis, EECS Department, University of California, Berkeley, 2017.

Shah, N. B., Bradley, J. K., Parekh, A., Wainwright, M., and Ramchandran, K. A case for ordinal peer-evaluation in MOOCs. In *NIPS Workshop on Data Driven Education*, December 2013.

Shah, N. B., Balakrishnan, S., Bradley, J., Parekh, A., Ramchandran, K., and Wainwright, M. J. Estimation from pairwise comparisons: Sharp minimax bounds with topology dependence. *Journal on Machine Learning Research*, 2016a.

Shah, N. B., Balakrishnan, S., and Wainwright, M. J. A permutation-based model for crowd labeling: optimal estimation and robustness. *arXiv:1606.09632*, 2016c.

Shah, N. B., Balakrishnan, S., and Wainwright, M. J. Feeling the Bern: Adaptive estimators for Bernoulli probabilities of pairwise comparisons. In *International Symposium on Information Theory*, 2016d.

Shah, N. B., Balakrishnan, S., Guntuboyina, A., and Wainwright, M. J. Stochastically transitive models for pairwise comparisons: Statistical and computational issues. *IEEE Transactions on Information Theory*, 63(2):934–959, 2017a.

Shah, N. B., Balakrishnan, S., and Wainwright, M. J. Low permutation-rank matrices: Structural properties and noisy completion. *arXiv preprint arXiv:1709.00127*, 2017b.

Shah, N. B., Tabibian, B., Muandet, K., Guyon, I., and von Luxburg, U. Design and analysis of the nips 2016 review process. *arXiv preprint arXiv:1708.09794*, 2017c.

Song, Y., Yifan, G., and Edward, G. An exploratory study of reliability of ranking vs. rating in peer assessment. In *ICALT*, volume 2017, 2017.

Soufiani, H., Parkes, D., and Xia, L. Computing parametric ranking models via rank-breaking. In *International Conference on Machine Learning*, 2014.

Thurstone, L. L. A law of comparative judgment. *Psychological Review*, 34(4):273, 1927.

Tsybakov, A. *Introduction to Nonparametric Estimation*. Springer Series in Statistics. 2008.

Tversky, A. Elimination by aspects: A theory of choice. *Psychological review*, 79(4):281, 1972.

Wauthier, F., Jordan, M., and Jojic, N. Efficient ranking from pairwise comparisons. In *International Conference on Machine Learning*, 2013.

Journal of Machine Learning Research 18 (2018) 1-28 Submitted 11/16; Revised 8/17; Published 4/18

Surprising properties of dropout in deep networks

David P. Helmbold dph@soe.ucsc.edu
Department of Computer Science
University of California, Santa Cruz
Santa Cruz, CA 95064, USA

Philip M. Long plong@google.com
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043, USA

Editor: Sanjoy Dasgupta

Abstract

We analyze dropout in deep networks with rectified linear units and the quadratic loss. Our results expose surprising differences between the behavior of dropout and more traditional regularizers like weight decay. For example, on some simple data sets dropout training produces negative weights even though the output is the sum of the inputs. This provides a counterpoint to the suggestion that dropout discourages co-adaptation of weights. We also show that the dropout penalty can grow exponentially in the depth of the network while the weight-decay penalty remains essentially linear, and that dropout is insensitive to various re-scalings of the input features, outputs, and network weights. This last insensitivity implies that there are no isolated local minima of the dropout training criterion. Our work uncovers new properties of dropout, extends our understanding of why dropout succeeds, and lays the foundation for further progress.

Keywords: Dropout, deep neural networks, regularization, learning theory.

1. Introduction

The 2012 ImageNet Large Scale Visual Recognition challenge was won by the University of Toronto team by a surprisingly large margin. In an invited talk at NIPS, Hinton (2012) credited the dropout training technique for much of their success. Dropout training is a variant of stochastic gradient descent (SGD) where, as each example is processed, the network is temporarily perturbed by randomly "dropping out" nodes of the network. The gradient calculation and weight updates are performed on the reduced network, and the dropped out nodes are then restored before the next SGD iteration. Since the ImageNet competition, dropout has been successfully applied to a variety of domains (Dahl, 2012; Deng et al., 2013; Dahl et al., 2013; Kalchbrenner et al., 2014; Chen and Manning, 2014), and is widely used (Schmidhuber, 2015; He et al., 2015; Szegedy et al., 2015; Yang et al., 2016; Havaei et al., 2017); for example, it is incorporated into popular packages such as TensorFlow, Torch, and Caffe. It is intriguing that crippling the network during training often leads to such dramatically improved results, and dropout has also sparked substantial research on related methods (for example, (Goodfellow et al., 2013; Wan et al., 2013; Gal and Ghahramani, 2016)).

In this work, we examine the effect of dropout on the inductive bias of the learning algorithm. A match between dropout's inductive bias and some important applications could explain the success of dropout, and its popularity also motivates the study of its properties.

Weight decay training optimizes the empirical error plus an L_2 regularization term, $\frac{\lambda}{2}||\mathbf{w}||_2^2$, so we call $\frac{\lambda}{2}||\mathbf{w}||_2^2$ the *L_2 penalty* of $\mathbf{w}$ since it is the difference between training criterion evaluated at $\mathbf{w}$ and the empirical loss of $\mathbf{w}$. By analogy, we define the *dropout penalty* of $\mathbf{w}$ to be the difference between the dropout training criterion and the empirical loss of $\mathbf{w}$ (see Section 2). Dropout penalties measure how much dropout discriminates against weight vectors, so they are key to understanding dropout's inductive bias.

Even in one-layer networks, conclusions drawn from (typically quadratic) approximations of the dropout penalty can be misleading (Helmbold and Long, 2015). Therefore we focus on exact formal analysis of dropout in multi-layer networks. Theoretical analysis of deep networks is notoriously difficult, so we might expect that a thorough understanding of dropout in deep networks must be achieved in stages. In this paper we further the process by exposing some of the surprising ways that the inductive bias of dropout differs from L_2 and other standard regularizers. These include the following:

- We show that dropout training can lead to negative weights *even when the output is a positive multiple of the the inputs.* Arguably, such use of negative weights constitutes co-adaptation – this adds a counterpoint to previous analyses showing that dropout discourages co-adaptation (Srivastava et al., 2014; Helmbold and Long, 2015).

- Unlike weight decay and other p-norm regularizers, dropout training is insensitive to the rescaling of input features, and largely insensitive to rescaling of the outputs; this may play a role in dropout's practical success. Dropout is also unaffected if the weights in one layer are scaled up by a constant c, and the weights of another layer are scaled down by c; this implies that dropout training does not have isolated local minima.

- The dropout penalty grows exponentially in the depth of the network in cases where the L_2 regularizer grows linearly. This may enable dropout to penalize the complexity of the network in a way that more meaningfully reflects the richness of the network's behaviors. (The exponential growth with d of the dropout penalty is reminiscent of some regularizers for deep networks studied by Neyshabur et al. (2015).)

- Dropout in deep networks has a variety of other behaviors different from standard regularizers. In particular: the dropout penalty for a set of weights can be negative; the dropout penalty of a set of weights depends on both the training instances and the labels; and although the dropout probability intuitively measures the strength of dropout regularization, the dropout penalties are often non-monotonic in the dropout probability. In contrast, Wager et al. (2013) show that when dropout is applied to generalized linear models, the dropout penalty is always non-negative and does not depend on the labels.

Our analysis is for multilayer neural networks with the square loss at the output node. The hidden layers use the popular rectified linear units (Nair and Hinton, 2010) outputting $\sigma(a) = \max(0, a)$ where a is the node's activation (the weighted sum of its inputs). We study the minimizers of a criterion that may be viewed as the objective function when using dropout. This abstracts away sampling and optimization issues to focus on the inductive bias, as in some previous work (Breiman, 2004; Zhang, 2004; Bartlett et al., 2006; Long and Servedio, 2010; Helmbold and Long, 2015). See Section 2 for a complete explanation.

Related work

A number of possible explanations have been suggested for dropout's success. Hinton et al. (2012) suggest that dropout controls network complexity by restricting the ability to co-adapt weights and illustrate how it appears to learn simpler functions at the second layer. Others (Baldi and Sadowski, 2013; Bachman et al., 2014) view dropout as an ensemble method combining the different network topologies resulting from the random deletion of nodes. Wager et al. (2014) observe that in 1-layer networks dropout essentially forces learning on a more challenging distribution akin to 'altitude training' of athletes.

Most formal analysis of the inductive bias of dropout has concentrated on the single-layer setting, where a single neuron combines the (potentially dropped-out) inputs. Wager et al. (2013) considered the case that the distribution of label y given feature vector $\mathbf{x}$ is a member of the exponential family, and the log-loss is used to evaluate models. They pointed out that, in this situation, the criterion optimized by dropout can be decomposed into the original loss and a term that does not depend on the labels. They then gave approximations to this dropout regularizer and discussed its relationship with other regularizers. As we have seen, many aspects of the behavior of dropout and its relationship to other regularizers are qualitatively different when there are hidden units.

Wager et al. (2014) considered dropout for learning topics modeled by a Poisson generative process. They exploited the conditional independence assumptions of the generative process to show that the excess risk of dropout training due to training set variation has a term that decays more rapidly than the straightforward empirical risk minimization, but also has a second additive term related to document length. They also discussed situations where the model learned by dropout has small bias.

Baldi and Sadowski (2014) analyzed dropout in linear networks, and showed how dropout can be approximated by normalized geometric means of subnetworks in the nonlinear case. Gal and Ghahramani (2015) described an interpretation of dropout as an approximation to a deep Gaussian process.

The impact of dropout (and its relative dropconnect) on generalization (roughly, how much dropout restricts the search space of the learner) was studied in (Wan et al., 2013).

In the on-line learning with experts setting, Van Erven et al. (2014) showed that applying dropout in on-line trials leads to algorithms that automatically adapt to the input sequence without requiring doubling or other parameter-tuning techniques.

The rest of the paper is organized as follows. Section 2 introduces our notation and formally defines the dropout model. We prove that dropout enjoys several scaling invariances that weight-decay doesn't in Section 3, and that dropout requires negative weights even in very simple situations in Section 4. Section 5 uncovers various properties of the

dropout penalty function. Section 6 describes some simulation experiments. We provide some concluding remarks in Section 7.

2. Preliminaries

Throughout, we will analyze fully connected layered networks with K inputs, one output, d layers (counting the output, but not the inputs), and n nodes in each hidden layer. We assume that n is a positive multiple of K and that K is a perfect square and a power of two to avoid unilluminating floor/ceiling clutter in the analysis. We will call this the *standard architecture*. We use $\mathcal{W}$ to denote a particular setting of the weights and biases in the network and $\mathcal{W}(\mathbf{x})$ to denote the network's output on input $\mathbf{x}$ using $\mathcal{W}$. The hidden nodes are ReLUs, and the output node is linear. $\mathcal{W}$ can be decomposed as $(W_1, \mathbf{b}_1, ..., W_{d-1}, \mathbf{b}_{d-1}, \mathbf{w}, b)$, where each W_j is the matrix of weights on connections from the $j-1$st into the jth hidden layer, each $\mathbf{b}_j$ is the vector of bias inputs into the jth hidden layer, $\mathbf{w}$ are the weights into the output node, and b is the bias into the output node.

An *example distribution* is a joint probability distribution over $(\mathbf{x}, y)$ pairs We focus on square loss, so the loss of $\mathcal{W}$ on example $(\mathbf{x}, y)$ is $(\mathcal{W}(\mathbf{x}) - y)^2$. The *risk* is the expected loss with respect to an example distribution P, we denote the risk of $\mathcal{W}$ as $R_P(\mathcal{W}) \stackrel{\text{def}}{=} \mathbf{E}_{(\mathbf{x},y)\sim P}\left((\mathcal{W}(\mathbf{x}) - y)^2\right)$. The subscript will often be omitted when P is clear from the context.

The goal of L_2 training is to find weights and biases minimizing the *L_2 criterion* with regularization strength λ: $J_2(\mathcal{W}) \stackrel{\text{def}}{=} R(\mathcal{W}) + \frac{\lambda}{2}||\mathcal{W}||^2$. Here and throughout, we use $||\mathcal{W}||^2$ to denote the sum of the squares of the weights of $\mathcal{W}$. (As usual, the biases are not penalized.) We use $\mathcal{W}_{L_2}$ to denote a minimizer of this criterion. The L_2 penalty, $\frac{\lambda}{2}||\mathcal{W}||^2$, is non-negative. This is useful, for example, to bound the risk of a minimizer $\mathcal{W}_{L_2}$ of J_2, since $R(\mathcal{W}) \le J_2(\mathcal{W})$.

Dropout training independently removes nodes in the network. In our analysis each non-output node is dropped out with the same probability q, so $p = 1 - q$ is the probability that a node is kept. (The output node is always kept; dropping it out has the effect of cancelling the training iteration.) When a node is dropped out, the node's output is set to 0. To compensate for this reduction, the values of the kept nodes are multiplied by $1/p$. With this compensation, the dropout can be viewed as injecting zero-mean additive noise at each non-output node (Wager et al., 2013). [1]

A *dropout pattern* is a boolean vector indicating the choices, for each node in the network, of whether the node is kept (1) or dropped out (0). For a network $\mathcal{W}$, an input $\mathbf{x}$, and dropout pattern $\mathcal{R}$, let $\mathcal{D}(\mathcal{W}, \mathbf{x}, \mathcal{R})$ be the output of $\mathcal{W}$ when nodes are dropped out or not following $\mathcal{R}$ (including the $1/p$ rescaling of kept nodes' outputs). The goal of dropout training on an example distribution P is to find weights and biases minimizing the *dropout criterion* for a given dropout probability:

$$J_D(\mathcal{W}) \stackrel{\text{def}}{=} \mathbf{E}_{\mathcal{R}}\mathbf{E}_{(\mathbf{x},y)\sim P}\left((\mathcal{D}(\mathcal{W}, \mathbf{x}, \mathcal{R}) - y)^2\right).$$

1. Some authors use a similar adjustment where the weights are scaled down at prediction time instead of inflating the kept nodes' outputs at training time.

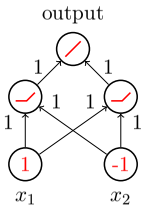


Figure 1: A network where the dropout penalty is negative.

This criterion is equivalent to the expected risk of the dropout-modified network, and we use $\mathcal{W}_D$ to denote a minimizer of it. Since the selection of dropout pattern and example from P are independent, the order of the two expectations can be swapped, yielding

$$J_D(\mathcal{W}) = \mathbf{E}_{(\mathbf{x},y)\sim P}\mathbf{E}_{\mathcal{R}}\left((\mathcal{D}(\mathcal{W},\mathbf{x},\mathcal{R})-y)^2\right). \quad (1)$$

Equation (1) is a key property of the dropout criterion. It indicates when something is true about the dropout criterion for a family of distributions concentrated on single examples, then (usually) the same thing will be true for any mixture of these single-example distributions.

Consider now the example network in Figure 1. The weight parameters W_1 and $\mathbf{w}$ are all-1's, and all of the biases are 0. $\mathcal{W}(1,-1)=0$ as each hidden node computes 0. Each dropout pattern indicates the subset of the four lower nodes to be kept, and when $q=p=1/2$ each subset is equally likely to be kept. If $\mathcal{R}$ is the dropout pattern where input x_2 is dropped and the other nodes are kept, then the network computes $\mathcal{D}(\mathcal{W},(1,-1),\mathcal{R})=8$ (recall that when $p=1/2$ the values of non-dropped out nodes are doubled). Only three dropout patterns produce a non-zero output, so if P is concentrated on the example $\mathbf{x}=(1,-1), y=8$ the dropout criterion is:

$$J_D(\mathcal{W}) = \frac{1}{16}(8-8)^2 + \frac{2}{16}(4-8)^2 + \frac{13}{16}(0-8)^2 = 54.$$

As mentioned in the introduction, the *dropout penalty* of a weight vector for a given example distribution and dropout probability is the amount that the dropout criterion exceeds the risk, $J_D(\mathcal{W}) - R(\mathcal{W})$. Wager et al. (2013) show that for 1-layer generalized linear models, the dropout penalty is non-negative.

Since $\mathcal{W}(1,-1)=0$, we have $R(\mathcal{W})=64$, and the dropout penalty is negative in our example. This is because the variance in the output due to dropout causes the network to better fit the data (on average) than the network's non-dropout evaluation. In Section 5.2, we give a necessary condition for this variance to be beneficial. As with the dropout criterion, the dropout penalty decomposes into an expectation of penalties over single examples:

$$J_D(\mathcal{W}) - R(\mathcal{W}) = \mathbf{E}_{(\mathbf{x},y)\sim P}\left(\mathbf{E}_{\mathcal{R}}\left((\mathcal{D}(\mathcal{W},\mathbf{x},\mathcal{R})-y)^2\right) - (\mathcal{W}(\mathbf{x})-y)^2\right).$$

Definition 1 *Define $P_{(\mathbf{x},y)}$ as the distribution with half of its weight on example $(\mathbf{x},y)$ and half of its weight on example $(\mathbf{0},0)$.*

Unless indicated otherwise, we assume $p=q=1/2$ for simplicity, although this is not crucial for our results.

3. Scaling inputs, weights and outputs

This section compares the sensitivity of dropout and weight decay on the scale of the training data.

3.1 Dropout is scale-free

Here we prove that dropout regularizes deep networks in a manner that is independent of the scale of the input features. In other words, training under dropout regularization does not penalize the use of large weights when needed to compensate for small input values.

Definition 2 *For any example distribution P, define the dropout aversion of P to be the maximum, over minimizers $\mathcal{W}_\mathcal{D}$ of the dropout criterion $J_\mathcal{D}(\mathcal{W}_\mathcal{D})$, of $R_P(\mathcal{W}_\mathcal{D})-\inf_\mathcal{W} R_P(\mathcal{W})$.*

The dropout aversion of P measures the extent to which P is incompatible with the inductive bias of dropout, measured by the risk gap between the true risk minimizer and the optimizers of the dropout criterion.

Definition 3 *For example distribution P and square matrix A, denote by $A\circ P$ the distribution obtained by sampling $(\mathbf{x},y)$ from P, and outputting $(A\mathbf{x},y)$.*

When A is diagonal and has full rank, then $A\circ P$ is a rescaling of the inputs, like changing one input from minutes to seconds and another from feet to meters.

Theorem 4 *For any example distribution P, and any diagonal full-rank $K\times K$ matrix A, the dropout aversion of P equals the dropout aversion of $A\circ P$.*

Proof: Choose a network

$$\mathcal{W} = (W_1, \mathbf{b}_1, ..., W_{d-1}, \mathbf{b}_{d-1}, \mathbf{w}, b).$$

Let

$$\mathcal{W}' = (W_1A^{-1}, \mathbf{b}_1..., W_{d-1}, \mathbf{b}_{d-1}, \mathbf{w}, b).$$

For any $\mathbf{x}$, $\mathcal{W}(\mathbf{x}) = \mathcal{W}'(A\mathbf{x})$, as A^{-1} undoes the effect of A before it gets to the rest of the network, which is unchanged. Furthermore, for any dropout pattern $\mathcal{R}$, we have $\mathcal{D}(\mathcal{W},\mathbf{x},\mathcal{R}) = \mathcal{D}(\mathcal{W}',A\mathbf{x},\mathcal{R})$. Once again A^{-1} undoes the effects of A on kept nodes (since A is diagonal), and the rest of the network $\mathcal{W}'$ is modified by $\mathcal{R}$ an a manner paralleling $\mathcal{W}$. Thus, there is bijection between networks $\mathcal{W}$ and networks $\mathcal{W}'$ with $J_\mathcal{D}(\mathcal{W}') = J_\mathcal{D}(\mathcal{W})$ and $R(\mathcal{W}') = R(\mathcal{W})$, yielding the theorem. ■

Theorem 4 indicates that some common normalizations of the input features (for example, to have unit variance) do not affect the quality of the dropout criterion minimizers, but normalization might change the speed of convergence and which minimizer is reached. Centering the features has slightly different properties. Although it is easy to use the biases to define a $\mathcal{W}'$ that "undoes" the centering in the non-dropout computation, different $\mathcal{W}'$ appear to be required for different dropout patterns, breaking the bijection exploited in Theorem 4.

As we will see in Section 3.4, weight decay does not enjoy such scale-free status.

3.2 Dropout's invariance to parameter scaling

Next, we describe an equivalence relation among parameterizations for dropout networks of depth $d \geq 2$. Basically, scaling the parameters at a level creates a corresponding scaling of the output. (A similar observation was made in a somewhat different context by Neyshabur et al. (2015).)

Theorem 5 *For any input* $\mathbf{x}$*, dropout pattern* $\mathcal{R}$*, any network*

$$\mathcal{W} = (W_1, \mathbf{b}_1, ..., W_{d-1}, \mathbf{b}_{d-1}, \mathbf{w}, b),$$

and any positive $c_1, ..., c_d$*, if*

$$\mathcal{W}' = \left(c_1 W_1, c_1 \mathbf{b}_1, c_2 W_2, c_1 c_2 \mathbf{b}_2, ..., c_{d-1} W_{d-1}, \left(\prod_{j=1}^{d-1} c_j \right) \mathbf{b}_{d-1}, c_d \mathbf{w}, \left(\prod_{j=1}^{d} c_j \right) b \right), \quad (2)$$

then $\mathcal{D}(\mathcal{W}', \mathbf{x}, \mathcal{R}) = \left(\prod_{j=1}^{d} c_j \right) \mathcal{D}(\mathcal{W}, \mathbf{x}, \mathcal{R})$*. In particular, if* $\prod_{j=1}^{d} c_j = 1$*, then for any example distribution* P*, networks* $\mathcal{W}$ *and* $\mathcal{W}'$ *have the same dropout criterion, dropout penalty, and expected loss.*

Note that the re-scaling of the biases at layer j depends not only on the rescaling of the connection weights at layer j, but also the re-scalings at lower layers.

Proof: Choose an input $\mathbf{x}$ and a dropout pattern $\mathcal{R}$. Define $\mathcal{W}'$ as in (2). For each hidden layer j, let $(h_{j1}, ..., h_{jn})$ be the jth hidden layer when applying $\mathcal{W}$ to $\mathbf{x}$ with $\mathcal{R}$, and let $(\tilde{h}_{j1}, ..., \tilde{h}_{jn})$ be the jth hidden layer when applying $\mathcal{W}'$ instead. By induction, for all i, $\tilde{h}_{ji} = \left(\prod_{\ell \leq j} c_\ell \right) h_{ji}$; the key step is that the pre-rectified value used to compute $\tilde{h}_{ji}$ has the same sign as for h_{ji}, since rescaling by c_j preserves the sign. Thus the same units are zeroed out in $\mathcal{W}$ and $\mathcal{W}'$ and $\mathcal{D}(\mathcal{W}, \mathbf{x}, \mathcal{R}) = (\prod_j c_j) \mathcal{D}(\mathcal{W}', \mathbf{x}, \mathcal{R})$. When $\prod_j c_j = 1$, this implies $\mathcal{D}(\mathcal{W}, \mathbf{x}, \mathcal{R}) = \mathcal{D}(\mathcal{W}', \mathbf{x}, \mathcal{R})$. Since this is true for all $\mathbf{x}$ and $\mathcal{R}$, we have $J_{\mathcal{D}}(\mathcal{W}) = J_{\mathcal{D}}(\mathcal{W}')$. Since, similarly, $\mathcal{W}(\mathbf{x}) = \mathcal{W}'(\mathbf{x})$ for all $\mathbf{x}$, so $R(\mathcal{W}) = R(\mathcal{W}')$, the dropout penalities for $\mathcal{W}$ and $\mathcal{W}'$ are also the same. ■

Theorem 5 implies that the dropout criterion never has isolated minimizers, since one can continuously up-scale the weights on one layer with a compensating down-scaling at another layer to get a contiguous family of networks computing the same function and having the same dropout criterion. It may be possible to exploit the parameterization equivalence of Theorem 5 in training by using canonical forms for the equivalent networks or switching to an equivalent networks whose gradients have better properties. We leave this question for future work.

3.3 Output scaling with dropout

Scaling the output values of an example distribution P does affect the aversion, but in a very simple and natural way.

Theorem 6 *For any example distribution* P*, if* P' *is obtained from* P *by scaling the outputs of* P *by a positive constant* c*, the dropout aversion of* P' *is* c^2 *times the dropout aversion of* P*.*

Proof: If a network $\mathcal{W}$ minimizes the dropout criterion for P, then the network $\mathcal{W}'$ obtained by scaling up the weights and bias for the output unit by c minimizes the dropout criterion for P', and for any $\mathbf{x}$, y, and dropout pattern $\mathcal{R}$, $(\mathcal{D}(\mathcal{W}, \mathbf{x}, \mathcal{R}) - y)^2 = (\mathcal{D}(\mathcal{W}', \mathbf{x}, \mathcal{R}) - cy)^2 / c^2$. ■

3.4 Scaling properties of weight decay

Weight decay is not scale-free. Define the *weight-decay aversion* analogously to the dropout aversion: the weight-decay aversion with P is the maximum, over minimizers $\mathcal{W}^*$ of $J_2(\mathcal{W}^*)$, of $R_P(\mathcal{W}^*) - \inf_{\mathcal{W}} R_P(\mathcal{W})$.

We analyze the L_2 criterion for depth-2 networks in Appendix B, resulting in the following theorem. Our proof shows that, despite the non-convexity of the L_2 criterion, it is still possible to identify a closed form for one of its optimizers.

Theorem 7 *Choose an arbitrary number* n *of hidden nodes,* $\lambda > 0$ *and* $\mathbf{x} \in \mathbf{R}^K$ *other than* $(0, 0, ..., 0)$*.*

The weight-decay aversion of $P_{(\mathbf{x},1)}$ *is* $\min(\frac{1}{4}, \frac{\lambda^2}{\mathbf{x} \cdot \mathbf{x}})$*.*

Theorem 7 shows that, unlike dropout, the weight decay aversion *does* depend on the scaling of the input features.

Furthermore, when $2\lambda > \sqrt{\mathbf{x} \cdot \mathbf{x}}$, the weight-decay criterion for $P_{(\mathbf{x},1)}$ has only a single isolated optimum weight setting[2] – all weights set to zero and bias $1/2$ at the output node. This means that weight-decay in 2-layer networks can completely regularize away significant signal in the sample *even when* λ *is finite*, contrasting starkly with weight-decay's behavior in 1-layer networks.

The "vertical" flexibility to rescale weights between layers enjoyed by dropout (Theorem 5) does not hold for L_2: one can always drive the L_2 penalty to infinity by scaling one layer up by a large enough positive c, even while scaling another down by c. On the other hand, the proof of Theorem 7 shows that the L_2 criterion has an alternative "horizontal" flexibility involving the rescaling of weights across nodes on the hidden layer (under the theorem's assumptions). Lemma 31 shows that at the optimizers each hidden node's contributions to the output are a constant (depending on the input) times their contribution to the the L_2 penalty. Shifting the magnitudes of these contributions between hidden nodes leads to alternative weights that compute the same value and have the same weight decay penalty. This is a more general observation than the permutation symmetry between hidden nodes because any portion of a hidden node's contribution can be shifted to another hidden node.

2. Since the output node puts weight 0 on each hidden node and the biases are unregularized, this optimum actually represents a class of networks differing only in the irrelevant biases at the hidden nodes. One can easily construct other cases when weight-decay has isolated minima in this sense, for example when $n = 2$ and there is equal probability on $\mathbf{x}$ and $-\mathbf{x}$, both with label 1.

4. Negative weights for monotone functions

If the weights of a unit are non-negative, then the unit computes a monotone function, in the sense that increasing any input while keeping the others fixed increases the output. The bias does not affect a node's monotonicity. A network of monotone units is also monotone.

We first present our theoretical results for many features (Section 4.1) and few features (Section 4.2), and then discuss the implication of these results in Section 4.3.

4.1 The basic case – many features

In this section, we analyze the simple distribution $P_{(\mathbf{1},1)}$ that assigns probability 1/2 to the example $(0,...,0),0$, and probability 1/2 to the example $(1,...,1),1$. The support of $P_{(\mathbf{1},1)}$ is arguably the simplest monotone function. Nevertheless, we prove that dropout uses negative weights to fit this data.

The key intuition is that optimizing the dropout criterion requires controlling the variance. Negative weights at the hidden nodes can be used to control the variance due to dropout at the input layer. When there are enough hidden nodes this becomes so beneficial that every minimizer of the dropout criterion uses such negative weights.

Theorem 8 *For the standard architecture, if $K > 18$ and n is large enough relative to K and d, every optimizer of the dropout criterion for $P_{(\mathbf{1},1)}$ uses at least one negative weight.*

To prove Theorem 8, we first calculate $J_{\mathcal{D}}(\mathcal{W}_{\text{neg}})$ for a network $\mathcal{W}_{\text{neg}}$ that uses negative weights, and then prove a lower bound greater than this value that holds for all networks using only non-negative weights.

All of the biases in $\mathcal{W}_{\text{neg}}$ are 0.

A key building block in the definition of $\mathcal{W}_{\text{neg}}$ is a block of hidden units that we call the *first-one gadget*. Each such block has K hidden nodes, and takes its input from the K input nodes. The ith hidden node in the block takes the value 1 if the ith input node is 1, and all inputs $x_{i'}$ for $i' < i$ are 0; otherwise it takes the value 0. This can be accomplished with a weight vector $\mathbf{w}$ with $w_{i'} = -1$ for $i' < i$, with $w_i = 1$, and with $w_{i'} = 0$ for $i' > i$. The first hidden layer of $\mathcal{W}_{\text{neg}}$ comprises n/K copies of the first-one gadget.

Informally, this construction removes most of the variance in the number of $1's$ in the input, as recorded in the following lemma.

Lemma 9 *On any input $\mathbf{x} \in \{0,1\}^n$ except $(0,0,...,0)$, the sum of the values on the first hidden layer of $\mathcal{W}_{\text{neg}}$ is exactly n/K.*

The weights into the remaining hidden layers of $\mathcal{W}_{\text{neg}}$ are all 1, and all the weights into the output layer take a value $c \stackrel{\text{def}}{=} \frac{K^2}{2n^{d-1}\left(1+\frac{K}{n}\right)\left(1+\frac{1}{n}\right)^{d-2}}$, chosen to minimize the dropout criterion for the network. The following lemma analyzes $\mathcal{W}_{\text{neg}}$.

Lemma 10 $J_{\mathcal{D}}(\mathcal{W}_{\text{neg}}) = \frac{1}{2}\left(1 - \frac{(1-2^{-K})}{\left(1+\frac{K}{n}\right)\left(1+\frac{1}{n}\right)^{d-2}}\right)$.

When n is large relative to K and d, the $\left(1+\frac{K}{n}\right)\left(1+\frac{1}{n}\right)^{d-2}$ denominator in Lemma 10 approaches 1, so $J_{\mathcal{D}}(\mathcal{W}_{\text{neg}})$ approaches $2^{-K}/2$ in this case. Lemma 16 below gives a larger

lower bound for any network with all positive weights. In the concrete case when $d = 2$ and $n = K^3$, then Lemma 10 implies $J_{\mathcal{D}}(\mathcal{W}_{\text{neg}}) < 1/K^2$.

Proof (of Lemma 10): Consider a computation of $\mathcal{W}_{\text{neg}}(1,1,...,1)$ under dropout and let $\hat{y}$ be the (random) output. Let k_0 be the number of input nodes kept, and, for each $j \geq 2$, let k_j be the number of nodes in the jth hidden layer kept. Call the node in each first-one gadget that computes 1 a *key* node, and if no node in the gadget computes 1 because the input is all dropped, arbitrarily make the gadget's first hidden node the key node. This ensures there is exactly one key node per gadget, and every non-key node computes 0. Let k_1 be the number of kept key nodes on the first hidden layer. If $k_0 = 0$, the output $\hat{y}$ of the network is 0. Otherwise, $\hat{y} = c2^d \prod_{j=1}^{d-1} k_j$.

Note that k_0 is zero with probability 2^{-K}. Whenever $k_0 \geq 1$, k_1 is distributed as $B(n/K, 1/2)$. Each other k_j is distributed as $B(n, 1/2)$, and $k_1, k_2, ..., k_{d-1}$ are independent of one another.

$$\begin{aligned}\mathbf{E}(\hat{y}) &= \mathbf{Pr}(k_0 \geq 1)c2^d \mathbf{E}[k_1 | k_0 \geq 1] \prod_{j=2}^{d-1} \mathbf{E}[k_j] \\ &= (1-2^{-K})c2^d \left(\frac{n}{2K}\right)\left(\frac{n}{2}\right)^{d-2} = \frac{2c}{K}(1-2^{-K})n^{d-1}.\end{aligned}$$

Using the value of the second moment of the binomial, we get

$$\begin{aligned}\mathbf{E}(\hat{y}^2) &= \mathbf{E}\left[\left(\mathbb{1}_{k_0 \geq 1} c2^d \prod_{j=1}^{d-1} k_j\right)^2\right] = 4c^2(1-2^{-K})\left(\frac{n}{K}\right)\left(\frac{n}{K}+1\right)n^{d-2}(n+1)^{d-2} \\ &= \frac{4c^2(1-2^{-K})}{K^2} n^{2(d-1)} \left(1+\frac{K}{n}\right)\left(1+\frac{1}{n}\right)^{d-2}.\end{aligned}$$

Thus,

$$\begin{aligned}J_{\mathcal{D}}(\mathcal{W}_{\text{neg}}) &= \frac{1}{2}\left(1 - 2\mathbf{E}(\hat{y}) + \mathbf{E}(\hat{y}^2)\right) \\ &= \frac{1}{2}\left(1 - \frac{4c}{K}(1-2^{-K})n^{d-1} + \frac{4c^2(1-2^{-K})}{K^2} n^{2(d-1)} \left(1+\frac{K}{n}\right)\left(1+\frac{1}{n}\right)^{d-2}\right) \\ &= \frac{1}{2}\left(1 - \frac{(1-2^{-K})}{\left(1+\frac{K}{n}\right)\left(1+\frac{1}{n}\right)^{d-2}}\right),\end{aligned}$$

since $c = \frac{K}{2n^{d-1}\left(1+\frac{K}{n}\right)\left(1+\frac{1}{n}\right)^{d-2}}$, completing the proof. ■

Next we prove a lower bound on $J_{\mathcal{D}}$ for networks with nonnegative weights. Let $\mathcal{W}$ be an arbitrary such network.

Our lower bound will use a property of the function computed by $\mathcal{W}$ that we now define.

Definition 11 *A function $\phi : \mathbf{R}^K \to \mathbf{R}$ is* supermodular *if for all $\mathbf{x}, \boldsymbol{\delta}_1, \boldsymbol{\delta}_2 \in \mathbf{R}^K$ where $\boldsymbol{\delta}_1, \boldsymbol{\delta}_2 \geq \mathbf{0}$*

$$\phi(\mathbf{x}) + \phi(\mathbf{x} + \boldsymbol{\delta}_1 + \boldsymbol{\delta}_2) \geq \phi(\mathbf{x} + \boldsymbol{\delta}_1) + \phi(\mathbf{x} + \boldsymbol{\delta}_2),$$

or equivalently:

$$\phi(\mathbf{x}+\boldsymbol{\delta}_1+\boldsymbol{\delta}_2)-\phi(\mathbf{x}+\boldsymbol{\delta}_2)\geq\phi(\mathbf{x}+\boldsymbol{\delta}_1)-\phi(\mathbf{x})$$

The latter indicates that adding $\boldsymbol{\delta}_1$ to the bigger input $\mathbf{x}+\boldsymbol{\delta}_2$ has at least as large an effect as adding it to the smaller input $\mathbf{x}$.

Since $\mathcal{W}$ has all non-negative weights, it computes a supermodular function of its inputs. (This fact may be of independent interest.)

Lemma 12 *If a network has non-negative weights and its activation functions $\sigma(\cdot)$ are convex, continuous, non-decreasing, and differentiable except on a finite set, then the network computes a supermodular function of its input $\mathbf{x}$.*

Proof: We will prove by induction over the layers that, for any unit h in the network, if $h(\mathbf{x})$ is the output of unit h when $\mathbf{x}$ is the input to $\mathcal{W}$, then $h(\cdot)$ is a supermodular function of its input.

The base case holds since each input node h outputs the corresponding component of the input, and $(\mathbf{x}+\boldsymbol{\delta}_1)-\mathbf{x}=(\mathbf{x}+\boldsymbol{\delta}_1+\boldsymbol{\delta}_2)-(\mathbf{x}+\boldsymbol{\delta}_2)$.

Now, for the inductive step, let $\mathbf{w}$ be the weight vector for node h, let b be its bias, and $\sigma(\cdot)$ be its activation function. Let $I(\mathbf{x})$, $I(\mathbf{x}+\boldsymbol{\delta}_1)$, $I(\mathbf{x}+\boldsymbol{\delta}_2)$, and $I(\mathbf{x}+\boldsymbol{\delta}_1+\boldsymbol{\delta}_2)$ be the inputs to node h when the inputs to the network are $\mathbf{x}$, $\mathbf{x}+\boldsymbol{\delta}_1$, $\mathbf{x}+\boldsymbol{\delta}_2$ and $\mathbf{x}+\boldsymbol{\delta}_1+\boldsymbol{\delta}_2$ respectively.

By induction, these inputs to node h (componentwise) satisfy

$$I(\mathbf{x}+\boldsymbol{\delta}_1)-I(\mathbf{x})\leq I(\mathbf{x}+\boldsymbol{\delta}_1+\boldsymbol{\delta}_2)-I(\mathbf{x}+\boldsymbol{\delta}_2).$$

Therefore, since $\mathbf{w}$, $\boldsymbol{\delta}_1$, and $\boldsymbol{\delta}_2$ are non-negative, the interval $[\mathbf{w}\cdot I(\mathbf{x}+\boldsymbol{\delta}_2)+b, \mathbf{w}\cdot I(\mathbf{x}+\boldsymbol{\delta}_1+\boldsymbol{\delta}_2)+b]$ is at least as long and starts at least as high as the interval $[\mathbf{w}\cdot I(\mathbf{x})+b, \mathbf{w}\cdot I(\mathbf{x}+\boldsymbol{\delta}_1)+b]$. Since σ is continuous and differentiable except on a finite set, we have

$$\begin{aligned}h(I(\mathbf{x}+\boldsymbol{\delta}_1))-h(I(\mathbf{x}))&=\int_{\mathbf{w}\cdot I(\mathbf{x})+b}^{\mathbf{w}\cdot I(\mathbf{x}+\boldsymbol{\delta}_1)+b}\sigma'(z)dz\\&\leq\int_{\mathbf{w}\cdot I(\mathbf{x}+\boldsymbol{\delta}_2)+b}^{\mathbf{w}\cdot I(\mathbf{x}+\boldsymbol{\delta}_1+\boldsymbol{\delta}_2)+b}\sigma'(z)dz\quad\text{(since }\sigma'\text{ is non-decreasing)}\\&=h(I(\mathbf{x}+\boldsymbol{\delta}_1+\boldsymbol{\delta}_2))-h(I(\mathbf{x}+\boldsymbol{\delta}_2)).\end{aligned}$$

■

Definition 13 *Let $\mathbf{r}_0\in\{0,1\}^K$ be the dropout pattern concerning the input layer, and let $\mathcal{R}'$ be the dropout pattern concerning the rest of the network, so that the dropout pattern $\mathcal{R}=(\mathbf{r}_0,\mathcal{R}')$.*

For each $\ell\in\{0,...,K\}$, let $\psi_{\mathcal{W}}(\ell)$ be the average output of $\mathcal{W}$ under dropout when ℓ of the inputs are kept: that is,

$$\psi_{\mathcal{W}}(\ell)=\mathbf{E}\left(\mathcal{D}(\mathcal{W},1^K,(\mathbf{r}_0,\mathcal{R}'))\middle|\sum_j r_{0j}=\ell\right).$$

Lemma 14 *For any $\ell\in\{1,...,K-1\}$,*

$$\psi_{\mathcal{W}}(\ell+1)-\psi_{\mathcal{W}}(\ell)\geq\psi_{\mathcal{W}}(\ell)-\psi_{\mathcal{W}}(\ell-1).$$

Proof: Generate $\mathbf{u}$, i and j randomly by, first choosing $\mathbf{u}$ uniformly at random from among bit vectors with ℓ ones, then choosing i uniformly from the 0-components of $\mathbf{u}$, and j uniformly from the 1-components of $\mathbf{u}$. By Lemma 12,

$$\mathcal{W}(\mathbf{u}+\mathbf{e}_i)-\mathcal{W}(\mathbf{u})\geq\mathcal{W}(\mathbf{u}-\mathbf{e}_j+\mathbf{e}_i)-\mathcal{W}(\mathbf{u}-\mathbf{e}_j)\tag{3}$$

always holds. Furthermore, $\mathbf{u}+\mathbf{e}_i$ is uniformly distributed among bit vectors with $\ell+1$ ones, $\mathbf{u}-\mathbf{e}_j$ is uniformly distributed among bit vectors with $\ell-1$ ones, and $\mathbf{u}+\mathbf{e}_i-\mathbf{e}_j$ is uniformly distributed among bit vectors with ℓ ones. This is true for $\mathcal{W}$, but it is also true for any network obtained by dropping out some of the hidden nodes of $\mathcal{W}$. Thus

$$\begin{aligned}&\psi_{\mathcal{W}}(\ell+1)-\psi_{\mathcal{W}}(\ell)\\&=\mathbf{E}(\mathcal{D}(\mathcal{W},1^K,(\mathbf{r}_0,\mathcal{R}')|\sum_j r_{0j}=\ell+1))-\mathbf{E}(\mathcal{D}(\mathcal{W},1^K,(\mathbf{r}_0,\mathcal{R}')|\sum_j r_{0j}=\ell))\\&=\mathbf{E}(\mathcal{D}(\mathcal{W},\mathbf{r}_0,(1^K,\mathcal{R}')|\sum_j r_{0j}=\ell+1))-\mathbf{E}(\mathcal{D}(\mathcal{W},\mathbf{r}_0,(1^K,\mathcal{R}')|\sum_j r_{0j}=\ell))\\&\geq\mathbf{E}(\mathcal{D}(\mathcal{W},\mathbf{r}_0,(1^K,\mathcal{R}')|\sum_j r_{0j}=\ell))-\mathbf{E}(\mathcal{D}(\mathcal{W},\mathbf{r}_0,(1^K,\mathcal{R}')|\sum_j r_{0j}=\ell-1)),\end{aligned}$$

by (3)) and the distributions of $\mathbf{u}+\mathbf{e}_i$, $\mathbf{u}-\mathbf{e}_j+\mathbf{e}_i$ and $\mathbf{u}-\mathbf{e}_j$. Since

$$\begin{aligned}&\mathbf{E}(\mathcal{D}(\mathcal{W},\mathbf{r}_0,(1^K,\mathcal{R}')|\sum_j r_{0j}=\ell))-\mathbf{E}(\mathcal{D}(\mathcal{W},\mathbf{r}_0,(1^K,\mathcal{R}')|\sum_j r_{0j}=\ell-1))\\&=\psi_{\mathcal{W}}(\ell)-\psi_{\mathcal{W}}(\ell-1),\end{aligned}$$

this completes the proof. ■

We will use the following lower bound on the tail of the binomial. (Many similar lower bounds are known.)

Lemma 15 *If X is distributed according to* $\mathrm{Bin}(n,1/2)$*, then*

$$\mathbf{Pr}(X<n/2-\sqrt{n}/4)=\mathbf{Pr}(X>n/2+\sqrt{n}/4)\geq 1/4.$$

Proof: Using the fact that, for any i, $\mathbf{Pr}(X=i)\leq 1/\sqrt{n}$, we get $\mathbf{Pr}(|X-n/2|<\sqrt{n}/4)\leq 1/2$, so $\mathbf{Pr}(X<n/2-\sqrt{n}/4)\geq 1/4$. ■

Now we are ready for the lower bound on $J_{\mathcal{D}}(\mathcal{W})$.

Lemma 16 *If $K>18$ and the weights in $\mathcal{W}$ are non-negative, then $J_{\mathcal{D}}(\mathcal{W})\geq\frac{1}{36K}$.*

Proof: Assume to the contrary that $J_{\mathcal{D}}(\mathcal{W})<\frac{1}{36K}$. First, note that $\psi_{\mathcal{W}}(0)\leq\sqrt{\frac{1}{18K}}$, or else the contribution to $J_{\mathcal{D}}(\mathcal{W})$ due to the $(0,0),0$ example is at least $\frac{1}{36K}$. Applying Lemma 15, we have

$$\psi_{\mathcal{W}}(K/2-\sqrt{K}/4)>1-\sqrt{\frac{2}{9K}}\text{ and }\psi_{\mathcal{W}}(K/2+\sqrt{K}/4)<1+\sqrt{\frac{2}{9K}}\tag{4}$$

as otherwise the contribution of one of the tails to $J_{\mathcal{D}}(\mathcal{W})$ will be at least $\frac{1}{36K}$ for the $(1,\ldots,1),1$ example. We will contradict this small variation of $\psi_{\mathcal{W}}(\ell)$ around $K/2$. The bounds on $\psi_{\mathcal{W}}(0)$ and $\psi(K/2-\sqrt{K}/4)$ and Lemma 14 imply that $\psi_{\mathcal{W}}(\ell)$ grows rapidly when ℓ is around $K/2$, in particular:

$$\psi(K/2-\sqrt{K}/4+1)-\psi(K/2-\sqrt{K}/4) \geq \frac{1-\sqrt{\frac{2}{9K}}-\sqrt{\frac{1}{18K}}}{K/2-\sqrt{K}/4} > \frac{1}{\sqrt{9/32}K},$$

since $K>18$. Now using Lemma 14 repeatedly shows that

$$\psi(K/2+\sqrt{K}/4)-\psi(K/2-\sqrt{K}/4) > \frac{\sqrt{K}}{2}\times\frac{1}{\sqrt{9/32}K} = 2\sqrt{\frac{2}{9K}},$$

which contradicts (4), completing the proof. ■

Putting together Lemmas 10 and 16 immediately proves Theorem 8, since for $K>18$ and large enough n, the criterion for $\mathcal{W}_{\text{neg}}$ must be less than the criterion for any network with all non-negative weights.

4.2 The case when $K=2$

Theorem 8 uses the assumption that $K>18$ and n is large enough; is the lower bound on K really necessary? Here we show that it is not, by treating the case that $K=2$.

Theorem 17 *For the standard architecture, if $K=2$, for any fixed d and large enough n, every optimizer of the dropout criterion for $P_{(1,1),1}$ uses negative weights.*

Proof: Define $\mathcal{W}_{\text{neg}}$ as in the proof of Lemma 10, except that the output layer has a bias of $1/5$.

We claim that

$$\lim_{n\to\infty} J_{\mathcal{D}}(\mathcal{W}_{\text{neg}}) = 1/10. \tag{5}$$

To see this, consider the joint input/label distribution under dropout:

$$\begin{aligned}
\mathbf{Pr}((0,0),0) &= 1/2\\
\mathbf{Pr}((0,0),1) &= 1/8\\
\mathbf{Pr}((2,0),1) &= 1/8\\
\mathbf{Pr}((0,2),1) &= 1/8\\
\mathbf{Pr}((2,2),1) &= 1/8.
\end{aligned}$$

Due to the bias of $1/5$ on the output, $\mathcal{W}_{\text{neg}}(0,0)=1/5$. Thus, contribution to $J_{\mathcal{D}}$ from examples with $\mathbf{x}=(0,0)$ in this joint distribution is $1/2\times(1/5)^2+1/8\times(4/5)^2=1/10$.

Now, choose $\mathbf{x}\neq(0,0)$. If, after dropout, the input is $\mathbf{x}$, each node in the hidden layer closest to the input computes 1. Arguing exactly as in the proof of Lemma 10, in such cases,

$$\mathbf{E}((\hat{y}-1)^2) = 1-\frac{1}{\left(1+\frac{2}{n}\right)\left(1+\frac{1}{n}\right)^{d-2}}.$$

This proves (5).

Now, let $\mathcal{W}$ be an arbitrary network with non-negative weights.

For our distribution,

$$J_{\mathcal{D}}(\mathcal{W}) = \frac{\mathbf{E}_{\mathcal{R}}\left((\mathcal{D}(\mathcal{W},(0,0),\mathcal{R})-0)^2\right)+\mathbf{E}_{\mathcal{R}}\left((\mathcal{D}(\mathcal{W},(1,1),\mathcal{R})-1)^2\right)}{2}.$$

Let $V_{00}=\mathbf{E}(\mathcal{W}(0,0)), V_{22}=\mathbf{E}(\mathcal{W}(2,2)), V_{20}=\mathbf{E}(\mathcal{W}(2,0)), V_{02}=\mathbf{E}(\mathcal{W}(0,2))$ where the expectations are taken with respect to the dropout patterns at the hidden nodes (with no dropout at the inputs). Since each dropout pattern over the hidden nodes defines a particular network, and Lemma 12 holds for all of them, the relationships also hold for the expectations, so

$$V_{22} \geq V_{20}+V_{02}-V_{00}.$$

Using this V notation, handling the dropout at the input explicitly, and the bias-variance decomposition keeping just the bias terms we get:

$$\begin{aligned}
J_{\mathcal{D}}(\mathcal{W}) &\geq \frac{(V_{00}-0)^2+\left((V_{00}-1)^2+(V_{22}-1)^2+(V_{20}-1)^2+(V_{02}-1)^2\right)/4}{2}\\
8J_{\mathcal{D}}(\mathcal{W}) &\geq 4(V_{00}-0)^2+(V_{00}-1)^2+(V_{22}-1)^2+(V_{20}-1)^2+(V_{02}-1)^2.
\end{aligned}$$

We will continue lower bounding the RHS. We can re-write V_{22} as $V_{20}+V_{02}-V_{00}+\epsilon$ where $\epsilon\geq 0$. This is convex and symmetric in V_{02} and V_{20} so they both take the same value at the minimizer of the RHS, so we proceed using V_{20} for this common minimizing value.

$$8J_{\mathcal{D}}(\mathcal{W}) \geq (2V_{20}-V_{00}+\epsilon-1)^2+2(V_{20}-1)^2+(V_{00}-1)^2+4V_{00}^2.$$

Differentiating with respect to V_{20}, we see that the RHS is minimized when $V_{20}=(2+V_{00}-\epsilon)/3$, giving

$$8J_{\mathcal{D}}(\mathcal{W}) \geq \frac{(V_{00}-\epsilon-1)^2}{3}+(V_{00}-1)^2+4V_{00}^2.$$

If $V_{00}\geq 1$, then $J_{\mathcal{D}}\geq 1/2$ (just from the (0,0),0 example) and when $V_{00}<1$ the RHS of the above is minimized for non-negative ϵ when $\epsilon=0$. Using this substitution, the minimizing value of V_{00} is $1/4$ giving

$$\begin{aligned}
8J_{\mathcal{D}}(\mathcal{W}) &\geq 1\\
J_{\mathcal{D}}(\mathcal{W}) &\geq 1/8.
\end{aligned}$$

Combining this with (5) completes the proof. ■

4.3 More general distributions and implications

In the previous sub-section we analyzed the distribution P over the 2-feature examples $(1,1),1$ and $(0,0),0$. However, these two examples can be embedded in a larger feature space by using any fixed vector of additional feature values, creating, for instance, a distribution over $(\underline{0,1},0,0,\underline{0,1/2,1}),0$ and $(\underline{0,1},1,1,\underline{0,1/2,1}),1$ (with the additional features underlined) The results of Section 4.2 still apply to distribution over these extended examples after defining $\mathcal{W}_{\text{neg}}$ network to have zero weight on the additional features, and noticing that any weight on the additional features in the positive-weight network $\mathcal{W}$ can be simulated using the biases at the hidden nodes.

It is particularly interesting when the additional features all take the value 0, we call these *zero-embeddings*. Every network $\mathcal{W}$ with non-negative weights has $J_{\mathcal{D}}(\mathcal{W}) \geq 1/8$ on each of these *zero-embeddings* of P. On the other hand, a single $\mathcal{W}_{\text{neg}}$ network with n/K copies of the K-input first-one gadget has $J_{\mathcal{D}}(\mathcal{W}_{\text{neg}}) \approx 1/10$ simultaneously for all of these zero-embeddings of P (when $n >> K$).

Any source distribution over $\{0,1\}^K \times \{0,1\}$ that puts probability $1/2$ on the the $\mathbf{0},0$ example and distributes the other $1/2$ probability over examples where exactly two inputs are one is a mixture of zero-embeddings of P, and thus $J_{\mathcal{D}}(\mathcal{W}) \geq 1/8$ while $J_{\mathcal{D}}(\mathcal{W}_{\text{neg}}) \approx 1/10$ for this mixture and optimizing the dropout criterion requires negative weights.

In our analysis the negative weights used by dropout are counterintuitive for fitting monotone behavior, but are needed to control the variance due to dropout. This suggests that dropout may be less effective when layers with sparse activation patterns are fed into wider layers, as dropout training can hijack part of the expressiveness of the wide layer to control the artificial variance due to dropout rather than fitting the underlying patterns in the data.

5. Properties of the dropout penalty

This section examines some properties of the dropout penalty applied to deep networks with ReLUs and the quadratic loss, which are often different than the case of networks without hidden layers.

5.1 Growth of the dropout penalty as a function of d

Weight-decay penalizes large weights, while Theorem 5 shows that compensating rescaling of the weights does not affect the dropout penalty or criterion. On the other hand, dropout can be more sensitive to the calculation of large outputs than weight decay, and large outputs can be produced in deep networks using only small weights. We make this observation concrete by exhibiting a family of networks where the depth and desired output are linked while the size of individual weights remains constant. For this family, the dropout penalty grows exponentially in the depth d (as opposed to linearly for weight-decay), suggesting that dropout training is less willing to fit the data in this kind of situation.

Theorem 18 *If $\mathbf{x} = (1,1,...,1)$ and $0 \leq y \leq Kn^{d-1}$, for $P_{\mathbf{x},y}$ there are weights $\mathcal{W}$ for the standard architecture with $R(\mathcal{W}) = 0$ such that (a) every weight has magnitude at most one, but (b) $J_{\mathcal{D}}(\mathcal{W}) \geq \frac{y^2}{K}$, whereas (c) $J_2(\mathcal{W}) \leq \frac{\lambda y^{2/d}}{2}(Kn + n^2(d-2) + n)$.*

Proof: Let $\mathcal{W}$ be the network whose weights are all $c = \frac{y^{1/d}}{K^{1/d}n^{(d-1)/d}}$ and biases are all 0, so that the L_2 penalty is the number of weights times $\lambda c^2/2$. It is a simple induction to show that, for these weights and input $(1,1,...,1)$, the value computed at each hidden node on level j is $c^j Kn^{j-1}$, so the the network outputs $c^d Kn^{d-1}$, and has zero square loss (since $\mathcal{W}(\mathbf{x}) = c^d Kn^{d-1} = y$).

Consider now dropout on this network. This is equivalent to changing all of the weights from c to $2c$ and, independently with probability $1/2$, replacing the value of each node with 0. For a fixed dropout pattern, each node on a given layer has the same weights, and receives the same (kept) inputs. Thus, the value computed at every node on the same layer is the same. For each j, let H_j be the value computed by the units in the jth hidden layer.

If k_0 is the number of input nodes kept under dropout, and, for each $j \in \{1,...,d-1\}$, k_j is the number of hidden nodes kept in layer j, a straightforward induction shows that, for all ℓ, we have $H_\ell = (2c)^\ell \prod_{j=0}^{\ell-1} k_j$, so that the output $\hat{y}$ of the network is $(2c)^d \prod_{j=0}^{d-1} k_j$.

Using a bias-variance decomposition,

$$\mathbf{E}((\hat{y}-y)^2) = (\mathbf{E}[\hat{y}]-y)^2 + \mathbf{Var}(\hat{y}).$$

Since each k_j is binomially distributed, and $k_0,...,k_{d-1}$ are independent, we have

$$\mathbf{E}(\hat{y}) = (2c)^d(K/2)(n/2)^{d-1} = c^d Kn^{d-1} = y,$$

so

$$\mathbf{E}((\hat{y}-y)^2) = \mathbf{Var}(\hat{y}).$$

Since

$$\mathbf{E}(\hat{y}^2) = (2c)^{2d}(K(K+1)/4)(n(n+1)/4)^{d-1} = y^2(1+1/K)(1+1/n)^{d-1},$$

we have

$$\mathbf{Var}(\hat{y}) = \mathbf{E}(\hat{y}^2) - \mathbf{E}(\hat{y})^2 = y^2((1+1/K)(1+1/n)^{d-1} - 1) \geq y^2/K,$$

completing the proof. ■

Theorem 18 shows that if $y = \exp(\Theta(d))$, the dropout penalty grows exponentially in d, whereas the L_2 penalty grows polynomially. Since the dropout penalty of a distribution decomposes into the expectation over single examples, the dropout penalty for any distribution P that puts positive probability on this $((1,1,\ldots,1),y)$ example has a term that grows exponentially in the depth.

5.2 A necessary condition for negative dropout penalty

Section 2 contains an example where the dropout penalty is negative. The following theorem provides a necessary condition.

Theorem 19 *The dropout penalty can be negative. For all example distributions, a necessary condition for this in rectified linear networks is that either a weight, input, or bias is negative.*

Proof: Baldi and Sadowski (2014) show that for networks of linear units (as opposed to the non-linear rectified linear units we focus on) the network's output without dropout equals the expected output over dropout patterns, so in our notation: $\mathcal{W}(\mathbf{x})$ equals $\mathbf{E}_{\mathcal{R}}(\mathcal{D}(\mathcal{W},\mathbf{x},\mathcal{R}))$. Assume for the moment that the network consists of linear units and the example distribution is concentrated on the single example $(\mathbf{x},y)$. Using the bias-variance decomposition for square loss and this property of linear networks,

$$J_D(\mathcal{W}) = \mathbf{E}_{\mathcal{R}}\left((\mathcal{D}(\mathcal{W},\mathbf{x},\mathcal{R})-y)^2\right) = (\mathbf{E}_{\mathcal{R}}(\mathcal{D}(\mathcal{W},\mathbf{x},\mathcal{R})-y)^2 + \mathbf{Var}_{\mathcal{R}}(\mathcal{D}(\mathcal{W},\mathbf{x},\mathcal{R})) \geq (\mathcal{W}(\mathbf{x})-y)^2$$

and the dropout penalty is again non-negative. Since the same calculations go through when averaging over multiple examples, we see that the dropout penalty is always non-negative for networks of linear nodes. When all the weights, biases and inputs in a network of rectified linear units are positive, then the rectified linear units behave as linear units, so the dropout penalty will again be non-negative. ■

5.3 Multi-layer dropout penalty does depend on labels

In contrast with its behavior on a variety of linear models including logistic regression (Wager et al., 2013), the dropout penalty can depend on the value of the response variable in deep networks with ReLUs and the quadratic loss. Thus in a fundamental and important respect, dropout differs from traditional regularizers like weight-decay or an L_1 penalty.

Theorem 20 *There are joint distributions P and Q, and weights $\mathcal{W}$ such that, for all dropout probabilities $q \in (0,1)$, (a) the marginals of P and Q on the input variables are equal, but (b) the dropout penalties of $\mathcal{W}$ with respect to P and Q are different.*

We will prove Theorem 20 by describing a general, somewhat technical, condition that implies that P and Q are witnesses to Theorem 20.

For each input $\mathbf{x}$ and dropout pattern $\mathcal{R}$, let $\mathcal{H}(\mathcal{W},\mathbf{x},\mathcal{R})$ be the values presented to the output node with dropout. As before, let $\underline{\mathbf{w}} \in \overline{\mathbf{R}^n}$ be those weights of $\mathcal{W}$ on connections directly into the output node and let b be the bias at the output node. Let $\underline{\mathbf{r}} \in \{0,1\}^n$ be the indicator variables for whether the various nodes connecting to the output node are kept.

Proof (of Theorem 20): Suppose that P is concentrated on a single $(\mathbf{x},y)$ pair. We will then get Q by modifying y.

Let $\mathbf{h}$ be the values coming into the output node in the non-dropped out network. Therefore the output of the non-dropout network is $\mathbf{w}\cdot\mathbf{h}+b$ while the output of the network with dropout is $\mathbf{w}\cdot\mathcal{H}(\mathcal{W},\mathbf{x},\mathcal{R})+b$. We now examine the dropout penalty, which is the expected dropout loss minus the non-dropout loss. We will use δ as a shorthand for $\mathbf{w}\cdot(\mathcal{H}(\mathcal{W},\mathbf{x},\mathcal{R})-\mathbf{h})$.

$$\begin{aligned}\text{dropout penalty} &= \mathbf{E}\left((\mathbf{w}\cdot\mathcal{H}(\mathcal{W},\mathbf{x},\mathcal{R})+b-y)^2\right) - (\mathbf{w}\cdot\mathbf{h}+b-y)^2 \\ &= \mathbf{E}\left((\mathbf{w}\cdot\mathcal{H}(\mathcal{W},\mathbf{x},\mathcal{R})+b-\mathbf{w}\cdot\mathbf{h}+\mathbf{w}\cdot\mathbf{h}-y)^2\right) - (\mathbf{w}\cdot\mathbf{h}+b-y)^2 \\ &= \mathbf{E}\left(\delta^2\right) + 2(\mathbf{w}\cdot\mathbf{h}+b-y)\mathbf{E}(\delta)\end{aligned}$$

which depends on the label y unless $\mathbf{E}(\delta)=0$.

Typically $\mathbf{E}(\delta)\neq 0$. To prove the theorem, consider the case where

- there are two inputs and one hidden layer,
- $\mathbf{x} = (1,-2)$,
- all weights in the network are 1, and
- all biases are 0.

Without dropout $\mathbf{h}=(0,0)$, but with dropout the hidden nodes are never negative and compute positive values when only the negative input is dropped, so that the expectation of δ is positive. ■

6. Experiments

To complement our theoretical results we performed two sets of experiments. The first set tests the scale dependence of dropout and weight decay, while the the second set examines its promotion of negative weights even when learning monotone functions. The code is accessible at

`https://www.dropbox.com/sh/6s2lcfrq17zshmp/AAAQO6uDa4gOAuAnw2MAghEMa?dl=0`

6.1 Scale (in)sensitivity

The scale dependence simulations were implemented using Torch. Our experiment used the standard architecture with $K=5$ inputs, depth $d=2$, $n=5$ hidden nodes. We used stochastic gradient using the `optim` package for Torch, with learning rate $\frac{0.01}{1+0.00001t}$ and momentum of 0.5, and a maximum of 100000 iterations.

We performed 10 sets of training runs. In each run:

- Ten training examples were generated uniformly at random from $[-1,1]^K$.
- Target outputs were assigned using $y=\prod_i \text{sign}(x_i)$.
- Five training sets $S_1,...,S_5$ with ten examples each were obtained by rescaling the inputs by $\{0.5, 0.75, 1, 1.25, 1.5\}$ and leaving the outputs unchanged.
- The weights of a network $\mathcal{W}_{\text{init}}$ were initialized using the default initialization from Torch.
- For each S_i:
  - $\mathcal{W}_{\text{init}}$ was cloned three times to produce $\mathcal{W}_D$, $\mathcal{W}_2$ and $\mathcal{W}_{\text{none}}$ with identical starting parameters.
  - $\mathcal{W}_D$ was trained with dropout probability 1/2 and no weight decay.
  - $\mathcal{W}_2$ was trained with weight decay with $\lambda = 1/2$ and no dropout.
  - $\mathcal{W}_{\text{none}}$ was trained without any regularization.

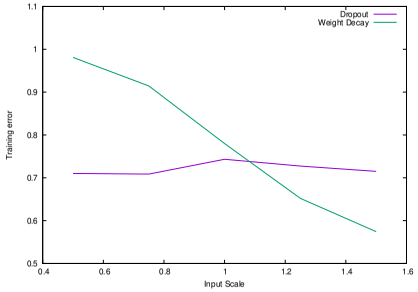


Figure 2: Training error as a function of the scale of the inputs for Dropout and Weight Decay in the experiment of Section 3.

The average training losses of $\mathcal{W}_D$ and $\mathcal{W}_2$, over the 10 runs, are shown in Figure 2. (The average training loss of $\mathcal{W}_{\text{none}}$ was less than 0.05 at all scales.)

The theoretical insensitivity of dropout to the scale of the inputs described in Theorem 4 is also seen here, along with the contrast with weight decay analyzed in Theorem 7.

The scale of the inputs also affects the dynamics of stochastic gradient descent. With very small inputs, convergence is very slow, and with very large inputs, SGD is unstable. The effects of the scale of the inputs on inductive bias analyzed in this paper are visible at the scales where optimization can be done effectively.

6.2 Negative Weights

Here we demonstrate dropout's propensity to use negative weights even when the function to be learned is monotonic. These experiments were implemented with Keras on top of TensorFlow and using SGD optimization with a learning rate of 0.005. Weight decay learning used a parameter of 0.05, and dropout training used a dropout rate of 0.5. We used the standard architecture with six inputs, 12 hidden nodes, and one output. The training set consists of six inputs as follows:

inputs	label	multiplicity
(0,0,0,0,0,0)	0	$\times 3$
(1,1,0,0,0,0)	1	$\times 1$
(0,0,1,1,0,0)	1	$\times 1$
(0,0,0,0,1,1)	1	$\times 1$

A natural network for this data uses all weights set to the same value $w = \sqrt{1/24}$ (with the biases set to 0). This network computes the correct labels for all of the training inputs. We ran each of dropout and weight-decay for 10,000 epochs from 10 different initial weight settings created by by sampling the weights independently from the uniform distribution

Table 1: Summary of weights resulting from noisy "all same" initialization

weights from	num neg	below -1.0	-1.0 to -0.1	-0.1 to -0.001	-0.001 to 0	0 to 0.001	0.001 to 0.1	0.1 to 1.0	above 1.0
initial wts	76	0	0	75	1	1	166	597	0
weight-decay	7	0	0	0	7	1	276	556	0
dropout	82	0	70	12	0	0	146	612	0

Table 2: Summary of weights resulting from noisy "first-one" initialization

weights from	num neg	below -1.0	-1.0 to -0.1	-0.1 to -0.001	-0.001 to 0	0 to 0.001	0.001 to 0.1	0.1 to 1.0	above 1.0
initial wts	204	90	80	34	180	1	47	232	176
weight-decay	111	0	0	57	68	68	298	349	0
dropout	303	91	93	119	5	0	168	189	175

over $[-0.2w, 2.2w]$. This randomization gives a one-in-twelve chance of changing the weight's sign.

Table 1 summarizes the initial weights, weights learned by weight-decay, and weights learned by dropout aggregated over the 10 initializations. This table shows that dropout not only (slightly) increases the number of negative weights, but also tends to greatly increase their magnitude. As expected, weight-decay eliminates almost all negative weights, and the few remaining negative weights have very small magnitudes.

We also ran a similar experiment based on the "first-one" gadget construction of Section 4.1. The hidden nodes are organized into six groups of two, with each group implementing three first-one gadgets – one for each of the input "pairs". Now the uncorrupted weights into the hidden layer are 0, 1, or -1, and those at the output node are $1/6$ so that the network computes the labels in the training set. As before, 10 noisy initial weight settings are chosen by selecting each weight uniformly from $[-0.2w, 2.2w]$, where w is the corresponding uncorrupted weight.

Table 2 summarizes the initial weights, weights learned by weight-decay, and weights learned by dropout aggregated over the 10 initializations. From this initialization, dropout tends to preserve the magnitudes of the negative weights. Although dropout increases the number of small-magnitude negative weights, much of this increase can be explained by half of the 180 weights initialized to exactly zero becoming negative. Again, weight-decay dramatically decreases both the number of the negative weights and their magnitudes.

7. Conclusions

The reasons behind dropout's surprisingly good performance in training deep networks across a variety of applications are somewhat mysterious and there is relatively little existing formal analysis. A variety of explanations have been offered (Bachman et al., 2014; Baldi and Sadowski, 2014, 2013; Gal and Ghahramani, 2015; Wager et al., 2014), including the

possibility that dropout reduces the amount of coadaptation in a network's weights (Hinton et al., 2012).

The dropout criterion is an expected loss over dropout patterns, and the variance in the output values over dropout patterns contributes to this expected loss. Therefore dropout may co-adapt weights in order to reduce this (artificial) variance. We prove that this happens even in very simple situations where nothing in the training data justifies negative weights (Theorem 8). This indicates that the relationship between dropout and co-adaption is not a simple one.

The effects of dropout in deep neural networks are rather complicated, and approximations can be misleading since the dropout penalty is very non-convex even in 1-layer networks (Helmbold and Long, 2015). In Section 3 we show that dropout does enjoy several scale-invariance properties that are not shared by weight-decay. A perhaps surprising consequence of these invariances is that there are never isolated local minima when learning a deep network with dropout. Further exploration of these scale invariance properties is warranted to see if they are a contributor to dropout's empirical success or can be exploited to facilitate training. While contrasting dropout to weight-decay in simple situations, we found that a degenerate all-zero network results (Theorem 7) when the L_2 regularization parameter is above a threshold. This is in dramatic contrast to our previous intuition from the 1-layer case.

In (Wager et al., 2013), dropout was viewed as a regularization method, adding a data dependent penalty to the empirical loss of (presumably) undesirable solutions. Section 5 shows that, unlike the generalized linear models case analyzed there, the dropout penalty in deeper networks can be negative and depends on the labels in the training data, and thus behaves unlike most regularizers. On the other hand, the dropout penalty can grow exponentially in the depth of the network, and thus may better reflect the complexity of the underlying model space than L_2 regularization.

This paper uncovers a number of dropout's interesting fundamental properties using formal analysis of simple cases. However, the effects of using dropout training in deep networks are subtle and complex, and we hope that this paper lays a foundation to promote further formal analysis of dropout's properties and behavior.

Acknowledgments

We are very grateful to Peter Bartlett, Seshadhri Comandur, and anonymous reviewers for valuable communications. We would like to acknowledge support for this project from the National Science Foundation (NSF grant IIS-9988642) and the Multidisciplinary Research Program of the Department of Defense (MURI N00014-00-1-0637).

Appendix A. Table of Notation

Notation	Meaning
$\mathbb{1}_{\text{set}}$	indicator function for "set"
$(\mathbf{x}, y)$	an example with feature vector $\mathbf{x}$ and label y
$\sigma(\cdot)$	the rectified linear unit computing $\max(0, \cdot)$
$\mathcal{W}$	an arbitrary weight setting for the network
w, v	specific weights, often subscripted
$\mathcal{W}(\mathbf{x})$	the output value produced by weight setting $\mathcal{W}$ on input $\mathbf{x}$
P	an arbitrary source distribution over $(\mathbf{x}, y)$ pairs
$P_{\mathbf{x},y}$	the source distribution concentrated on the single example $(\mathbf{x}, y)$
$R_P(\mathcal{W})$	the risk (expected square loss) of $\mathcal{W}$ under source P
q, p	probabilities that a node is dropped out (q) or kept (p)
$\mathcal{R}$	a dropout pattern, indicates the kept nodes
$\mathbf{r}, \mathbf{s}$	dropout patterns on subsets of the nodes
$\mathcal{D}(\mathcal{W}, \mathbf{x}, \mathcal{R})$	Output with weights $\mathcal{W}$, input $\mathbf{x}$, and dropout pattern $\mathcal{R}$
$J_D(\mathcal{W})$	the dropout criterion
$J_2(\mathcal{W})$	the L_2 criterion
λ	the L_2 regularization strength parameter
$\mathcal{W}_D$	an optimizer of the dropout criterion
$\mathcal{W}_{L_2}$	an optimizer of the L_2 criterion
n, d	the network width and depth
K	the number of input nodes

Appendix B. Proof of Theorem 7

Here we prove Theorem 7, showing that the weight-decay aversion depends on the values of the inputs and the number of input nodes K. Furthermore, unlike the single-layer case, the L_2 regularization strength has a threshold where the minimizer of the L_2 criterion degenerates to the all-zero network.

We will focus on the standard architecture with depth $d = 2$. Recall that we are analyzing the distribution $P_{(\mathbf{x},1)}$ that assigns probability $1/2$ to $(\mathbf{x}, 1)$ and probability $1/2$ to $(\mathbf{0}, 0)$.

Also, recall that, for $P_{(\mathbf{x},1)}$, the weight-decay aversion is the maximum risk incurred by a minimizer of J_2.

The proof of Theorem 7 involves a series of lemmas, in which we show that there is a minimizer of J_2 of a special form, and then relate any hidden node's effect on the output to the regularization penalty on the weights in and out of that node. This will allow us to treat optimizing the L_2 criterion as a one-dimensional problem, whose solution yields the theorem.

For some minimizer $\mathcal{W}_{L_2}$ of J_2, let $\underline{\mathbf{v}_j^*}$ denote the vector of weights into hidden node j and $\underline{w_j^*}$ denote the weight from j to the output node. Let $\underline{a_j^*}$ be the bias for hidden node j and let $\underline{b^*}$ be the bias for the output node. Let $\underline{h_j}$ be the function computed by hidden node j.

Lemma 21 *We may assume without loss of generality that for each hidden node j, there is an input $\tilde{\mathbf{x}} \in \{\mathbf{0}, \mathbf{x}\}$ such that $h_j(\tilde{\mathbf{x}}) = 0$.*

Proof: Suppose neither of $h_j(\mathbf{0})$ or $h_j(\mathbf{x})$ was 0. If $w_j^* = 0$, then replacing both with 0 does not affect the output of $\mathcal{W}_{L_2}$, and does not increase the penalty. If $w_j^* \neq 0$, then subtracting $\min\{h_j(\mathbf{0}), h_j(\mathbf{x})\}$ from a_j^* and adding $w_j^* \cdot \min\{h_j(\mathbf{0}), h_j(\mathbf{x})\}$ to b does not affect the output of $\mathcal{W}_{L_2}$ or the penalty, but, after this transformation, $\min\{h_j(\mathbf{0}), h_j(\mathbf{x})\} = 0$. ■

Lemma 22 *We may assume without loss of generality that for each hidden node j, we have $|\mathbf{v}_j^* \cdot \mathbf{x}| = \max\{h_j(\mathbf{0}), h_j(\mathbf{x})\}$.*

Proof: If $h_j(\mathbf{x}) \geq h_j(\mathbf{0}) = 0$, then bias $a_j^* = 0$, and $h_j(\mathbf{x}) = \mathbf{v}_j^* \cdot \mathbf{x}$.

If $h_j(\mathbf{0}) > h_j(\mathbf{x}) = 0$. Then, $a_j^* > 0$, and $\mathbf{v}_j^* \cdot \mathbf{x} \leq -a_j^*$. If needed, the magnitude of $\mathbf{v}_j^*$ can be decreased to make $\mathbf{v}_j^* \cdot \mathbf{x} = -a_j^*$. This decrease does not affect $\mathcal{W}_{L_2}(\mathbf{x})$ or $\mathcal{W}_{L_2}(\mathbf{0})$, and can only reduce the L_2 penalty. ■

Lemma 23 *We may assume without loss of generality that for each hidden node j, we have $h_j(\mathbf{0}) = 0$.*

Proof: Suppose $h_j(\mathbf{0}) > 0$. Let z be this old value of $h_j(\mathbf{0})$. Then $h_j(\mathbf{x}) = 0$ and $z = h_j(\mathbf{0}) = -\mathbf{v}_j^* \cdot \mathbf{x}$. If we negate $\mathbf{v}^*$ and set $a_j = 0$, then Lemma 22 implies that we swap the values of $h_j(\mathbf{x})$ and $h_j(\mathbf{0})$.

Then, by adding zw_j^* to b^* and negating w_j^*, we correct for this swap at the output node and do not affect the function computed by $\mathcal{W}_{L_2}$ or the penalty. ■

Note that Lemma 23 implies that, without loss of generality, $a_1^* = ... = a_K^* = 0$. We continue with that assumption.

Lemma 24 *For all j, $\mathbf{v}_j^* \cdot \mathbf{x} \geq 0$.*

Proof: Since $a_j^* = 0$, if $\mathbf{v}_j^* \cdot \mathbf{x} < 0$, we could make $\mathcal{W}_{L_2}$ compute the same function with a smaller penalty by replacing $\mathbf{v}_j^*$ with $\mathbf{0}$. ■

Lemma 24 implies that the optimal $\mathcal{W}_{L_2}$ computes the linear function:

$$\mathcal{W}_{L_2}(\tilde{\mathbf{x}}) = (\mathbf{w}^*)^T V^* \tilde{\mathbf{x}} + b^*,$$

when $\tilde{\mathbf{x}}$ is a non-negative multiple of $\mathbf{x}$. Later we will call $(\mathbf{w}^*)^T V^* \tilde{\mathbf{x}}$ the <u>activation</u> at the output node.

Lemma 25 $b^* = \frac{1-(\mathbf{w}^*)^T V^* \mathbf{x}}{2}$.

Proof: Minimize J_2 (wrt distribution $P_{(\mathbf{x},1)}$) as a function of b using Calculus. ■

Now, for $\tilde{\mathbf{x}}$ a non-negative multiple of $\mathbf{x}$,

$$\mathcal{W}_{L_2}(\tilde{\mathbf{x}}) = \frac{1}{2} + (\mathbf{w}^*)^T V^* (\tilde{\mathbf{x}} - \mathbf{x}/2) \tag{6}$$

which immediately implies

$$\mathcal{W}_{L_2}(\mathbf{0}) = 1 - \mathcal{W}_{L_2}(\mathbf{x}). \tag{7}$$

Lemma 26 *Each $\mathbf{v}_j^*$ is a (positive) rescaling of $\mathbf{x}$.*

Proof: Projecting $\mathbf{v}_j^*$ onto the span of $\mathbf{x}$ does not affect h_j, and cannot increase the penalty. ■

Lemma 27 $\mathcal{W}_{L_2}(\mathbf{x}) \leq 1$.

Proof: By (7), if $\mathcal{W}_{L_2}(\mathbf{x}) > 1$ then $\mathcal{W}_{L_2}(\mathbf{0}) < 0$ and the loss and the penalty would both be reduced by scaling down $\mathbf{w}^*$. ■

Lemma 28 *$\mathcal{W}_{L_2}$ maximizes $\mathcal{W}(\mathbf{x})$ over those weight vectors $\mathcal{W}$ that have the same penalty as $\mathcal{W}_{L_2}$ and compute a function of the form $\mathcal{W}(\tilde{x}) = \frac{1}{2} + (\mathbf{w})^T V(\tilde{\mathbf{x}} - \mathbf{x}/2)$ (that is, Equation 6).*

Proof: Let $\mathcal{W}$ maximize $\mathcal{W}(\mathbf{x})$ over the networks considered. If $\mathcal{W}_{L_2}(\mathbf{x}) < \mathcal{W}(\mathbf{x}) \leq 1$, then $\mathcal{W}$ would have the same penalty as $\mathcal{W}_{L_2}$ but smaller error, contradicting the optimality of $\mathcal{W}_{L_2}$.

If $\mathcal{W}(\mathbf{x}) > 1$, then the network $\widetilde{\mathcal{W}}$ obtained by scaling down the weights in the output layer so that $\widetilde{\mathcal{W}}(\mathbf{x}) = 1$ has a smaller penalty than $\mathcal{W}_{L_2}$ and smaller error, again contradicting $\mathcal{W}_{L_2}$'s optimality. ■

Informally, Lemmas 27 and 28 engender a view of the learner straining against the yolk of the L_2 penalty to produce a large enough output on $\mathbf{x}$. This motivates us to ask how large $\mathcal{W}(\mathbf{x})$ can be, for a given value of $||\mathcal{W}||_2^2$ (recall that Lemma 24 allows us to assume that the biases at the hidden nodes are all 0).

Definition 29 *For each hidden node j, let $\underline{\alpha_j}$ be the constant such $\mathbf{v}_j^* = \alpha_j \mathbf{x}$, so that $h_j(\mathbf{x}) = \alpha_j \mathbf{x} \cdot \mathbf{x}$.*

Recall that the *activation* at the output node on input $\mathbf{x}$ is the weighted sum of the hidden-node outputs, $(\mathbf{w}^*)^T V^* \mathbf{x}$.

Definition 30 *The <u>contribution to the activation</u> at the output due to hidden node j is*

$$w_j^* h_j(\mathbf{x}) = w_j^* \alpha_j \mathbf{x} \cdot \mathbf{x}$$

and the <u>contribution to the L_2 penalty</u> from these weights is

$$\frac{\lambda}{2}\left((w_j^*)^2 + \alpha_j^2 \mathbf{x} \cdot \mathbf{x}\right).$$

We now bound the contribution to the activation in terms of the contribution to the L_2 penalty. Note that as the L_2 "budget" increases, so does the the maximum possible contribution to the output node's activation.

Lemma 31 *If B is hidden node j's weight-decay contribution, $(w_j^*)^2+\alpha_j^2\mathbf{x}\cdot\mathbf{x}$, then hidden node j's contribution to the output node's activation is maximized when $w_j^* = \sqrt{\frac{B}{2}}$ and $\alpha_j = \sqrt{\frac{B}{2\mathbf{x}\cdot\mathbf{x}}}$, where it achieves the value $B\sqrt{\mathbf{x}\cdot\mathbf{x}}/2$*

Proof: Since $\alpha_j^2\mathbf{x}\cdot\mathbf{x}+(w_j^*)^2 = B$, we have $w_j^* = \sqrt{B-\alpha_j^2\mathbf{x}\cdot\mathbf{x}}$, so the contribution to the activation can be re-written as $\alpha_j\mathbf{x}\cdot\mathbf{x}\sqrt{B-\alpha_j^2\mathbf{x}\cdot\mathbf{x}}$. Taking the derivative with respect to α_j, and solving, we get $\alpha_j = \pm\sqrt{\frac{B}{2\mathbf{x}\cdot\mathbf{x}}}$ and we want the positive solution (otherwise the node outputs 0). When $\alpha_j = \sqrt{\frac{B}{2\mathbf{x}\cdot\mathbf{x}}}$ we have $w_j^* = \sqrt{\frac{B}{2}}$ and thus the node's maximum contribution to the activation is

$$\sqrt{\frac{B}{2}}\sqrt{\frac{B}{2\mathbf{x}\cdot\mathbf{x}}}\,(\mathbf{x}\cdot\mathbf{x}) = \frac{B\sqrt{\mathbf{x}\cdot\mathbf{x}}}{2}.$$

■

Lemma 32 *The minimum sum-squared weights for a network $\mathcal{W}$ (without biases at the hidden nodes) that has an activation A at the output node on input $\mathbf{x}$ is $\frac{2A}{\sqrt{\mathbf{x}\cdot\mathbf{x}}}$.*

Proof: When maximized, the contribution of each hidden node to the activation at the output is $\sqrt{\mathbf{x}\cdot\mathbf{x}}/2$ times the hidden node's contribution to the sum of squared-weights. Since each weight in $\mathcal{W}$ is used in exactly one hidden node's contribution to the output node's activation, this completes the proof. ■

Note that this bound is independent of n, the number of hidden units, but does depend on the input $\mathbf{x}$.

Proof (of Theorem 7): Let $A \geq 0$ be the activation at the output node for $\mathcal{W}_{L_2}$ on input $\mathbf{x}$. From Lemma 25 we get that $b^* = \frac{1-A}{2}$. Combining Lemmas 27 , 28 and 32, we can re-write the J_2 criterion for $\mathcal{W}_{L_2}$ and distribution $P_{(\mathbf{x},1)}$ in terms of A as follows.

$$\begin{aligned} J_2(\mathcal{W}_{L_2}) &= \frac{1}{2}\left(b^*-0\right)^2 + \frac{1}{2}\left(\mathcal{W}_{L_2}(\mathbf{x})-1\right)^2 + \frac{\lambda}{2}||\mathcal{W}||_2^2 \\ &= \frac{1}{2}\left(\left(\frac{1-A}{2}\right)^2 + \left(\frac{1+A}{2}-1\right)^2 + \lambda\frac{2A}{\sqrt{\mathbf{x}\cdot\mathbf{x}}}\right). \end{aligned} \tag{8}$$

Differentiating with respect to A, we see that the criterion is minimized when

$$\begin{aligned} A &= 1-\frac{2\lambda}{\sqrt{\mathbf{x}\cdot\mathbf{x}}} && \text{when } \frac{2\lambda}{\sqrt{\mathbf{x}\cdot\mathbf{x}}} \leq 1 \\ A &= 0 && \text{when } \frac{2\lambda}{\sqrt{\mathbf{x}\cdot\mathbf{x}}} > 1 \end{aligned}$$

since we assumed $A \geq 0$; when $A = 0$ then $\mathcal{W}_{L_2}$ has all zero weights with a bias of $1/2$ at the output.

The risk part of (8) simplifies to

$$\frac{(1-A)^2}{4} = \frac{\lambda^2}{\mathbf{x}\cdot\mathbf{x}},$$

so the overall the risk of $\mathcal{W}_{L_2}$, which minimizes J_2, is $\min\left\{\frac{1}{4}, \frac{\lambda^2}{\mathbf{x}\cdot\mathbf{x}}\right\}$. ■

References

P. Bachman, O. Alsharif, and D. Precup. Learning with pseudo-ensembles. *NIPS*, 2014.

P. Baldi and P. Sadowski. The dropout learning algorithm. *Artificial intelligence*, 210: 78–122, 2014.

Pierre Baldi and Peter J Sadowski. Understanding dropout. In *Advances in Neural Information Processing Systems*, pages 2814–2822, 2013.

P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

L. Breiman. Some infinity theory for predictor ensembles. *Annals of Statistics*, 32(1):1–11, 2004.

Caffe. Caffe, 2016. http://caffe.berkeleyvision.edu.

Danqi Chen and Christopher D Manning. A fast and accurate dependency parser using neural networks. In *EMNLP*, pages 740–750, 2014.

G. E. Dahl. Deep learning how I did it: Merck 1st place interview, 2012. http://blog.kaggle.com.

G. E. Dahl, T. N. Sainath, and G. E. Hinton. Improving deep neural networks for LVCSR using rectified linear units and dropout. *ICASSP*, 2013.

L. Deng, J. Li, J. Huang, K. Yao, D. Yu, F. Seide, M. L. Seltzer, G. Zweig, X. He, J. Williams, Y. Gong, and A. Acero. Recent advances in deep learning for speech research at microsoft. *ICASSP*, 2013.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *arXiv:1506.02142*, 2015.

Yarin Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1019–1027, 2016.

Ian J. Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron C. Courville, and Yoshua Bengio. Maxout networks. In *ICML*, pages 1319–1327, 2013.

Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, Chris Pal, Pierre-Marc Jodoin, and Hugo Larochelle. Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35:18–31, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, pages 1026–1034, 2015.

D. P. Helmbold and P. M. Long. On the inductive bias of dropout. *JMLR*, 16:3403–3454, 2015.

G. E. Hinton. Dropout: a simple and effective way to improve neural networks, 2012. videolectures.net.

G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors, 2012. Arxiv, arXiv:1207.0580v1.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In *ACL*, pages 655–665, 2014.

P. M. Long and R. A. Servedio. Random classification noise defeats all convex potential boosters. *Machine Learning*, 78(3):287–304, 2010.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *COLT*, pages 1376–1401, 2015.

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

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 15:1929–1958, 2014.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CVPR*, 2015.

TensorFlow. Tensorflow, 2016. https://www.tensorflow.org.

Torch. Torch, 2016. http://torch.ch.

T. Van Erven, W. Kotowski, and M. K. Warmuth. Follow the leader with dropout perturbations. *COLT*, pages 949–974, 2014.

S. Wager, S. Wang, and P. Liang. Dropout training as adaptive regularization. *NIPS*, 2013.

S. Wager, W. Fithian, S. Wang, and P. S. Liang. Altitude training: Strong bounds for single-layer dropout. *NIPS*, 2014.

L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus. Regularization of neural networks using dropconnect. In *ICML*, pages 1058–1066, 2013.

Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 21–29, 2016.

T. Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, 32(1):56–85, 2004.

Journal of Machine Learning Research 18 (2018) 1-63 Submitted 5/16; Revised 2/18; Published 4/18

Exact Learning of Lightweight Description Logic Ontologies

Boris Konev KONEV@LIVERPOOL.AC.UK
Department of Computer Science
University of Liverpool, United Kingdom

Carsten Lutz CLU@INFORMATIK.UNI-BREMEN.DE
Department of Computer Science
University of Bremen, Germany

Ana Ozaki ANA.OZAKI@TU-DRESDEN.DE
Department of Computer Science
Dresden University of Technology, Germany

Frank Wolter WOLTER@LIVERPOOL.AC.UK
Department of Computer Science
University of Liverpool, United Kingdom

Editor: Luc De Raedt

Abstract

We study the problem of learning description logic (DL) ontologies in Angluin et al.'s framework of exact learning via queries. We admit membership queries ("is a given subsumption entailed by the target ontology?") and equivalence queries ("is a given ontology equivalent to the target ontology?"). We present three main results: (1) ontologies formulated in (two relevant versions of) the description logic DL-Lite can be learned with polynomially many queries of polynomial size; (2) this is not the case for ontologies formulated in the description logic $\mathcal{EL}$, even when only acyclic ontologies are admitted; and (3) ontologies formulated in a fragment of $\mathcal{EL}$ related to the web ontology language OWL 2 RL can be learned in polynomial time. We also show that neither membership nor equivalence queries alone are sufficient in cases (1) and (3).

Keywords: Exact Learning, Description Logic, Complexity

1. Introduction

In many subfields of artificial intelligence, ontologies are used to provide a common vocabulary for the application domain of interest and to give a meaning to the terms in the vocabulary, and to describe the relations between them. Description logics (DLs) are a prominent family of ontology languages with a long history that goes back to Brachman's famous knowledge representation system KL-ONE in the early 1980s (Brachman and Schmolze, 1985). Today, there are several widely used families of DLs that differ in expressive power, computational complexity, and intended application. The most important ones are the $\mathcal{ALC}$ family which aims at high expressive power, the $\mathcal{EL}$ family (Baader et al., 2005) which aims to provide scalable reasoning, and the DL-Lite family (Calvanese et al., 2007; Artale et al., 2009) which is tailored specifically towards applications in data access. In 2004, the World Wide Web Committee (W3C) has standardised a DL of the $\mathcal{ALC}$ family as an ontology language for the web, called OWL. The standard was updated to OWL 2 in 2009, and since then comprises a family of five languages including the OWL 2 profiles OWL 2 EL, OWL 2 QL, and OWL 2 RL. While OWL 2 EL is based on $\mathcal{EL}$ and OWL 2 QL on DL-Lite, OWL 2 RL is closely related to the fragment of $\mathcal{EL}$ that is obtained by allowing only concept names on the right-hand side of concept inclusions. In this paper we study DLs from the $\mathcal{EL}$ and DL-Lite families. Designing an ontology for an application domain is a subtle, error-prone, and time consuming task. From its beginnings, DL research was driven by the aim to provide various forms of support for ontology engineers, assisting them in the design of high-quality ontologies; examples include the ubiquitous task of ontology classification (Baader et al., 2017), reasoning support for debugging ontologies (Wang et al., 2005; Schlobach et al., 2007), support for modular ontology design (Stuckenschmidt et al., 2009), and checking the completeness of the modelling in a systematic way (Baader et al., 2007). The same aim is pursued by the field of ontology learning, where the goal is to use machine learning techniques for various ontology engineering tasks such as to identify the relevant vocabulary of the application domain (Cimiano et al., 2010; Wong et al., 2012), to learn an initial version of the ontology that is then refined manually (Borchmann and Distel, 2011; Ma and Distel, 2013; Jiménez-Ruiz et al., 2015), and to learn concept expressions as building blocks of an ontology (Lehmann and Hitzler, 2010). For details we refer the reader to a collection of articles in ontology learning edited by Lehmann and Völker (2014) and Section 7.

In this paper we concentrate on learning the full logical structure of a description logic ontology. Our starting point is the observation that building a high-quality ontology relies on the successful communication between an ontology engineer and a domain expert because the former is typically not sufficiently familiar with the domain and the latter is rarely an expert in ontology engineering. We study the foundations of this communication process in terms of a simple communication model and analyse, within this model, the complexity of constructing a correct and complete domain ontology. Our model rests on the following assumptions:

1. The domain expert has perfect knowledge of the domain, but is not able to formalise or communicate the target ontology $\mathcal{O}$ to be constructed.

2. The domain expert is able to communicate the vocabulary (predicate symbols, which in the case of DLs take the form of concept and role names) of $\mathcal{O}$ and shares it with the ontology engineer. The ontology engineer knows nothing else about the domain.

3. The ontology engineer can pose queries to the domain expert which the domain expert answers truthfully. The main queries posed by the ontology engineer are of the form

   "Is the concept inclusion $C \sqsubseteq D$ entailed by $\mathcal{O}$?"

4. In addition, the ontology engineer needs a way to find out whether the ontology $\mathcal{H}$ that has been constructed so far, called the hypothesis ontology, is complete. If not, he requests an example illustrating the incompleteness. The engineer can thus ask:

   "Is the ontology $\mathcal{H}$ complete? If not, then return a concept inclusion $C \sqsubseteq D$ entailed by $\mathcal{O}$ but not by $\mathcal{H}$."

JMLR 18(201):1-63, 2018

We are then interested in whether the target ontology $\mathcal{O}$ can be constructed with only polynomially many queries of polynomial size (polynomial query learnability) or, even better, with overall polynomial time (polynomial time learnability). In both cases, the polynomial is in the size of the ontology to be constructed plus the size of the counterexamples returned by the domain expert. Without taking into account the latter, one can never expect to achieve polynomial time learnability because the domain expert could provide unnecessarily large counterexamples. Note that polynomial time learnability implies polynomial query learnability, but that the converse is false because polynomial query learnability allows the ontology engineer to run computationally costly procedures between posing queries.

The above model is an instance of Angluin et al.'s framework of exact learning via queries (Angluin, 1987b). In this context, the queries mentioned in Point 3 above are called *membership queries*. The queries in Point 4 are a form of *equivalence queries*. In Angluin's framework, however, such queries are slightly more general:

> "Is the hypothesis ontology $\mathcal{H}$ equivalent to the target ontology $\mathcal{O}$? If not, then return a concept inclusion $C \sqsubseteq D$ entailed by $\mathcal{O}$ but not by $\mathcal{H}$ (a *positive counterexample*) or vice versa (a *negative counterexample*)."

In our upper bounds (that is, polynomial learnability results), we admit only queries of the more restricted form in Point 4 above: the learning algorithm is designed in a way so that the hypothesis ontology $\mathcal{H}$ is a consequence of the target ontology $\mathcal{O}$ at all times, and thus the only meaningful equivalence query is a query of the form "Is $\mathcal{H}$ already complete?". Our lower bounds (results saying that polynomial learnability is impossible), in contrast, apply to unrestricted equivalence queries, that is, they do not assume that the hypothesis is implied by the target. In this way, we achieve maximum generality.

Within the setup outlined above, we study the following description logics:

(a) DL-Lite$^{\exists}_{\mathcal{R}}$, which is a member of the DL-Lite family that admits role inclusions and allows nested existential quantification on the right-hand side of concept inclusions;

(b) the extension DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ of DL-Lite$^{\exists}_{\mathcal{R}}$ with conjunction on the left-hand side of concept inclusions;

(c) the basic member $\mathcal{EL}$ of the $\mathcal{EL}$ family;

(d) the fragment $\mathcal{EL}_{\mathsf{lhs}}$ of $\mathcal{EL}$ where only concept names (but no compound concept expressions) are admitted on the right-hand side of concept inclusions.

We remark that DL-Lite$^{\exists}_{\mathcal{R}}$ is closely related to OWL 2 QL, which is based on the fragment of DL-Lite$^{\exists}_{\mathcal{R}}$ that does not allow nested existential quantification on the right-hand side of concept inclusions. In this more restricted case, though, polynomial learnability is uninteresting. In fact, the number of concept inclusions formulated in a fixed finite vocabulary Σ is bounded polynomially in the size of Σ instead of being infinite as in the description logics studied in this paper; consequently, TBoxes are trivially learnable in polynomial time, even when only membership queries (but no equivalence queries) are available or vice versa. The extension DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ of DL-Lite$^{\exists}_{\mathcal{R}}$ is not part of the OWL 2 QL standard, but admitting conjunctions on the left-hand side of concept inclusions is a useful and widely considered extension of basic DL-Lite dialects (Artale et al., 2009). $\mathcal{EL}_{\mathsf{lhs}}$ is a significant part

Not polynomial query learnable	⋯›	$\mathcal{EL}$
Polynomial query learnable	⋯›	DL-Lite$^{\exists}_{\mathcal{R}}$ DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$
Polynomial time learnable	⋯›	$\mathcal{EL}_{\mathsf{rhs}}$ $\mathcal{EL}_{\mathsf{lhs}}$

Figure 1: Summary of main results

of the OWL 2 RL language and can be viewed as a natural fragment of Datalog. An even better approximation of OWL 2 RL would be the extension of $\mathcal{EL}_{\mathsf{lhs}}$ with inverse roles, but polynomial learnability in that language remains an open problem. And finally, unrestricted $\mathcal{EL}$ can be viewed as a logical core of the OWL 2 EL language.

After introducing preliminaries in Section 2, we study exact learning of DL-Lite$^{\exists}_{\mathcal{R}}$ ontologies in Section 3, establishing polynomial query learnability. We strengthen this result to DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ in Section 4, using a significantly more subtle algorithm. It remains open whether DL-Lite$^{\exists}_{\mathcal{R}}$ and DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ admit polynomial time learnability. Our algorithms do not yield such a stronger result since they use subsumption checks to analyse counterexamples provided by the oracle and to integrate them into the current hypothesis ontology, and subsumption is NP-complete in these DLs (Kikot et al., 2011). In Section 5, we show that $\mathcal{EL}_{\mathsf{lhs}}$ ontologies are learnable in polynomial time, a result that extends the known polynomial time learnability of propositional Horn formulas (Angluin et al., 1992), which correspond to $\mathcal{EL}$ ontologies without existential restrictions. In fact, our algorithms take inspiration from learning algorithms for propositional Horn formulas and combine the underlying ideas with modern concepts from DL such as canonical models, simulations, and products. The algorithm for $\mathcal{EL}_{\mathsf{lhs}}$ also uses subsumption checks, which in this case does not get in the way of polynomial time learnability since subsumption in $\mathcal{EL}_{\mathsf{lhs}}$ can be decided in polynomial time.

In Section 6, we then establish that $\mathcal{EL}$ ontologies are not polynomial query learnable. Note that the fragment $\mathcal{EL}_{\mathsf{rhs}}$ of $\mathcal{EL}$, which is symmetric to $\mathcal{EL}_{\mathsf{lhs}}$ and only admits concept names on the *left*-hand side of concept inclusions is a fragment of DL-Lite$^{\exists}_{\mathcal{R}}$. Together, our upper bounds for DL-Lite$^{\exists}_{\mathcal{R}}$ and $\mathcal{EL}_{\mathsf{lhs}}$ thus establish that failure of polynomial query learnability of $\mathcal{EL}$ ontologies is caused by the interaction between existential restrictions on the left- and right-hand sides of concept inclusions. Interestingly, our result already applies to acyclic $\mathcal{EL}$ TBoxes, which disallow recursive definitions of concepts and are of a rather restricted syntactic form. However, the result does rely on concept inclusions as counterexamples that are of a form not allowed in acyclic TBoxes. We also show that ontologies formulated in DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ and in $\mathcal{EL}_{\mathsf{lhs}}$ are neither polynomial query learnable with membership queries alone nor with equivalence queries alone; corresponding results for propositional Horn formulas are well known (Frazier and Pitt, 1993; Angluin et al., 1992; Angluin, 1987a; Arias and Balcázar, 2011). Figure 1 summarises the main results obtained in this paper.

In Section 7 we provide an extensive discussion of related work on the exact learning of logical formulas and theories, and we close the paper with a discussion of open problems. A small number of proofs are deferred to an appendix.

2. Preliminaries

We introduce the description logics studied in this paper, then consider a representation of concept expressions in terms of labelled trees and show how important semantic notions such as subsumption between concept expressions can be characterised by homomorphisms between the corresponding trees. This also involves introducing canonical models, which are an important tool throughout the paper. Finally, we formally introduce the framework of exact learning.

2.1 Description Logics

Let $\mathsf{N_C}$ be a countably infinite set of *concept names* (denoted by upper case letters A, B, etc) and let $\mathsf{N_R}$ be a countably infinite set of *role names* disjoint from $\mathsf{N_C}$ (denoted by lower case letters r, s, etc). Concept and role names can be regarded as unary and binary predicates, respectively. In description logic, constructors are used to define compound concept and role expressions from concept and role names. In this paper, the only role constructor is the inverse role constructor: for $r \in \mathsf{N_R}$, the expression r^- is the *inverse role* of r. Semantically, r^- represents the converse of the binary relation r. A *role expression* is a role name or an inverse role. We set $r^- := s$ if $r = s^-$ for a role name s. For brevity, we will typically speak of *roles* rather than of role expressions. The concept constructors used in this paper are $\top$ (*everything*), $\sqcap$ (*conjunction*), and $\exists r.C$ (*qualified existential restriction*). Formally, *concept expressions* C are defined according to the following syntactic rule:

$$C, D \quad := \quad \top \quad | \quad A \quad | \quad C \sqcap D \quad | \quad \exists r.C$$

where A is a concept name and r is a role. For example, $\exists \mathsf{hasChild}.\top \sqcap \exists \mathsf{gender}.\mathsf{Male}$ denotes the class of individuals who have a child and whose gender is male.

Terminological knowledge is captured by finite sets of *inclusions* between concept expressions or roles. Specifically,

- a *concept inclusion (CI)* takes the form $C \sqsubseteq D$, where C and D are concept expressions, and
- a *role inclusion (RI)* takes the form $r \sqsubseteq s$, where r and s are roles.

An *ontology* or *TBox* is a finite set of CIs and RIs.[1] We use $C \equiv D$ as an abbreviations for the two CIs $C \sqsubseteq D$ and $D \sqsubseteq C$ and likewise for $r \equiv s$; we speak of *concept equivalences (CEs)* and *role equivalences (REs)*, respectively.

1. In the description logic literature, CIs of the form introduced here are often called $\mathcal{ELI}$ CIs to distinguish them from CIs that use concept expressions formulated in other description logics. The TBoxes are called $\mathcal{ELIH}$ TBoxes (TBoxes that consist of $\mathcal{ELI}$ CIs and RIs).

Example 1 *Consider the following TBox:*

$$\begin{aligned}
\mathsf{Prof} &\sqsubseteq \exists \mathsf{supervisor_of}.\mathsf{Student} \sqcap \exists \mathsf{conduct_research}.\top && (1)\\
\mathsf{Graduate} &\equiv \exists \mathsf{has_degree}.\top && (2)\\
\mathsf{GraduateStudent} &\equiv \mathsf{Student} \sqcap \mathsf{Graduate} && (3)\\
\mathsf{GraduateStudent} &\sqsubseteq \exists \mathsf{supervisor_of}^-.\mathsf{Prof} && (4)\\
\mathsf{supervisor_of} &\sqsubseteq \mathsf{advisor_of} && (5)\\
\mathsf{CS_Graduate} &\equiv \exists \mathsf{has_degree}.\mathsf{CS_Degree} && (6)
\end{aligned}$$

The CI in Line 1 states that every professor supervises students and conducts research. Notice that we do not specify the specific area of research, hence we use an unqualified existential restriction of the form $\exists r.\top$. The CE in Line 2 defines a graduate as anyone who has a degree. The CE in Line 3 defines a graduate student as a student who is a graduate. The CI in Line 4 states that graduate students are supervised by professors. Notice that we use the inverse role of supervisor_of *here. Line 5 shows an RI which states that every supervisor is an advisor. The CE in the last line defines a computer science graduate as someone with a degree in computer science.*

A *signature* is a set of concept and role names and we use $\Sigma_{\mathcal{T}}$ to denote the signature of the TBox $\mathcal{T}$, that is, the set of concept and role names that occur in it. The *size* $|C|$ of a concept expression C is the length of the string that represents C, where concept names and role names are considered to be of length one. The *size* $|\mathcal{T}|$ of a TBox $\mathcal{T}$ is defined as $\sum_{C \sqsubseteq D \in \mathcal{T}} |C| + |D|$.

The semantics of concept expressions and TBoxes is defined as follows (Baader et al., 2017). An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is given by a non-empty set $\Delta^{\mathcal{I}}$ (the *domain* of $\mathcal{I}$) and a mapping $\cdot^{\mathcal{I}}$ that maps every concept name A to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$ and every role name r to a subset $r^{\mathcal{I}}$ of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The *interpretation* $r^{\mathcal{I}}$ of an inverse role $r = s^-$ is given by $r^{\mathcal{I}} = \{(d, d') \mid (d', d) \in s^{\mathcal{I}}\}$ and the *interpretation* $C^{\mathcal{I}}$ of a concept expression C is defined inductively by

$$\begin{aligned}
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}}\\
(C_1 \sqcap C_2)^{\mathcal{I}} &= C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}\\
(\exists r.C)^{\mathcal{I}} &= \{d \in \Delta^{\mathcal{I}} \mid \text{there exists } d' \in C^{\mathcal{I}} \text{ with } (d, d') \in r^{\mathcal{I}}\}.
\end{aligned}$$

An interpretation $\mathcal{I}$ *satisfies* a concept expression C if $C^{\mathcal{I}}$ is not empty. It satisfies the CI $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, written as $\mathcal{I} \models C \sqsubseteq D$. Similarly, $\mathcal{I}$ satisfies RI $r \sqsubseteq s$ if $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$, written as $\mathcal{I} \models r \sqsubseteq s$. $\mathcal{I}$ is a *model of a TBox* $\mathcal{T}$ if it satisfies all CIs and RIs in $\mathcal{T}$. A TBox $\mathcal{T}$ *entails* a CI or RI α, in symbols $\mathcal{T} \models \alpha$, if α is satisfied in every model of $\mathcal{T}$. Concept expressions C and D are *equivalent w.r.t.* $\mathcal{T}$, written $\mathcal{T} \models C \equiv D$, if $\mathcal{T} \models C \sqsubseteq D$ and $\mathcal{T} \models D \sqsubseteq C$; equivalence of roles r and s is defined accordingly, written $\mathcal{T} \models s \equiv r$. TBoxes $\mathcal{T}$ and $\mathcal{T}'$ are *logically equivalent*, in symbols $\mathcal{T} \equiv \mathcal{T}'$, if $\mathcal{T} \models \alpha$ for all $\alpha \in \mathcal{T}'$ and vice versa.

JMLR 18(201):1-63, 2018

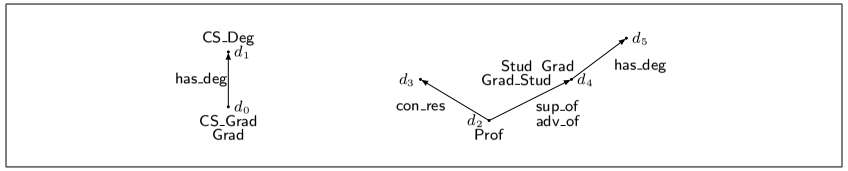


Figure 2: Illustration to Example 2.

Example 2 *Consider the TBox $\mathcal{T}$ from Example 1 and the interpretation $\mathcal{I}$ that is illustrated in Figure 2 and defined by setting $\Delta^{\mathcal{I}} = \{d_0, \ldots, d_5\}$ and*

$$\begin{aligned}
\mathsf{Prof}^{\mathcal{I}} &= \{d_2\}, & \mathsf{conduct_research}^{\mathcal{I}} &= \{(d_2, d_3)\}\\
\mathsf{Student}^{\mathcal{I}} &= \{d_4\}, & \mathsf{supervisor_of}^{\mathcal{I}} &= \{(d_2, d_4)\},\\
\mathsf{Graduate_Student}^{\mathcal{I}} &= \{d_4\}, & \mathsf{advisor_of}^{\mathcal{I}} &= \{(d_2, d_4)\},\\
\mathsf{Graduate}^{\mathcal{I}} &= \{d_4, d_0\}, & \mathsf{has_degree}^{\mathcal{I}} &= \{(d_4, d_5), (d_0, d_1)\},\\
\mathsf{CS_Graduate}^{\mathcal{I}} &= \{d_0\}, & \mathsf{CS_Degree}^{\mathcal{I}} &= \{d_1\}.
\end{aligned}$$

It is easy to see that $\mathcal{I}$ is a model of $\mathcal{T}$. Moreover, $\mathcal{I} \not\models \mathsf{Graduate} \sqsubseteq \mathsf{CS\_Graduate}$ as $\mathsf{Graduate}^{\mathcal{I}} = \{d_4, d_0\}$ but $\mathsf{CS\_Graduate}^{\mathcal{I}} = \{d_0\}$, thus $\mathcal{T} \not\models \mathsf{Graduate} \sqsubseteq \mathsf{CS\_Graduate}$. It can be shown that $\mathcal{T} \models \mathsf{CS\_Graduate} \sqsubseteq \mathsf{Graduate}$.

It is ExpTime-complete to decide, given a TBox $\mathcal{T}$ and a concept inclusion $C \sqsubseteq D$, whether $\mathcal{T} \models C \sqsubseteq D$ (Baader et al., 2008); this reasoning problem is known as *subsumption*. Because of this high complexity, the profiles of OWL 2 are based on syntactically more restricted description logics in which subsumption is less complex. We next introduce a few relevant such logics. A *basic concept* is a concept name or a concept expression of the form $\exists r.\top$, where r is a role. For example, $\exists \mathsf{hasChild}^-.\top$ is a basic concept, but $\exists \mathsf{hasChild}^-.\mathsf{Graduate}$ is not.

DL-Lite$^{\exists}_{\mathcal{R}}$. A *DL-Lite$^{\exists}_{\mathcal{R}}$ CI* takes the form

$$B \sqsubseteq C$$

where B is a basic concept and C is a concept expression. A *DL-Lite$^{\exists}_{\mathcal{R}}$ inclusion* is a DL-Lite$^{\exists}_{\mathcal{R}}$ CI or an RI. A *DL-Lite$^{\exists}_{\mathcal{R}}$ TBox* is a finite set of DL-Lite$^{\exists}_{\mathcal{R}}$ inclusions.

Example 3 *Lines* (1), (4), *and* (5) *of Example 1 are DL-Lite$^{\exists}_{\mathcal{R}}$ inclusions and Line* (2) *abbreviates the two DL-Lite$^{\exists}_{\mathcal{R}}$ CIs* $\mathsf{Graduate} \sqsubseteq \mathsf{has\_degree}.\top$ *and* $\mathsf{has\_degree}.\top \sqsubseteq \mathsf{Graduate}$. *Lines* (3) *and* (6) *do not fall within DL-Lite$^{\exists}_{\mathcal{R}}$.*

DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$. In the extension DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ of DL-Lite$^{\exists}_{\mathcal{R}}$, CIs take the form

$$B_1 \sqcap \cdots \sqcap B_n \sqsubseteq C$$

where $B_1, \ldots, B_n$ are basic concepts and C is a concept expression. A *DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ TBox* $\mathcal{T}$ is a finite set of DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ CIs and RIs. Both DL-Lite$^{\exists}_{\mathcal{R}}$ and DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ have been investigated in detail (Artale et al., 2009).

Example 4 *As Lines* (1), (2), (4) *and* (5) *from Example 1 fall within DL-Lite$^{\exists}_{\mathcal{R}}$, they also fall within DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$. Line* (3) *falls within DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$. Line* (6) *is not in DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$.*

$\mathcal{EL}$. An $\mathcal{EL}$ *concept expression* is a concept expression that does not use inverse roles. An $\mathcal{EL}$ *concept inclusion* is a CI of the form

$$C \sqsubseteq D$$

where C and D are $\mathcal{EL}$ concept expressions. An $\mathcal{EL}$ *TBox* is a finite set of $\mathcal{EL}$ CIs. Thus, $\mathcal{EL}$ does neither admit role inclusions nor inverse roles. In contrast to DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$, however, it allows existential restrictions $\exists r.C$ with $C \neq \top$ on the left-hand side of CIs.

Example 5 *Inclusions* (1), (2), (3) *and* (6) *from Example 1 are $\mathcal{EL}$ inclusions. Inclusion* (5) *is not an $\mathcal{EL}$ inclusion.*

Subsumption is NP-complete in DL-Lite$^{\exists}_{\mathcal{R}}$ and in DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ (Kikot et al., 2011; Calvanese et al., 2007; Artale et al., 2009). Subsumption in $\mathcal{EL}$ is in PTime (Baader et al., 2005) and this is still true if RIs that do not use inverse roles are admitted in the TBox. Given a TBox $\mathcal{T}$ and an RI $r \sqsubseteq s$, deciding whether $\mathcal{T} \models r \sqsubseteq s$ is possible in PTime in all description logics considered in this paper. In fact, $\mathcal{T} \models r \sqsubseteq s$ if, and only if, there exists a sequence $r_0, \ldots, r_n$ of roles such that $r = r_0$, $s = r_n$, and for every $i < n$ either $r_i \sqsubseteq r_{i+1} \in \mathcal{T}$ or $r_i^- \sqsubseteq r_{i+1}^- \in \mathcal{T}$. Our learning algorithms will carry out various subsumption checks as a subprocedure, as detailed later on.

2.2 Tree Representation of Concept Expressions

To achieve an elegant and succinct exposition of our learning algorithms, it will be convenient to represent concept expressions C as a finite directed tree T_C whose nodes are labelled with sets of concept names and whose edges are labelled with roles, and to describe manipulations of concept expressions in terms of manipulations of the corresponding tree such as merging nodes, replacing subgraphs, modifying node and edge labels, etc. We generally use ρ_C to denote the root node of the tree T_C. In detail, T_C is defined as follows. For $C = \top$, the tree T_C has a single node d with label $l(d) = \emptyset$; if $C = A$, where A is a concept name, then T_C has a single node d with $l(d) = \{A\}$; if $C = \exists r.D$, then T_C is obtained from T_D by adding a new root d_0 and an edge from d_0 to the root d of T_D with label $l(d_0, d) = r$ (we then call d an *r-successor* of d_0); if $C = D_1 \sqcap D_2$, then T_C is obtained by identifying the roots of T_{D_1} and T_{D_2}.

Example 6 *For $C = \mathsf{Student} \sqcap \exists \mathsf{has\_degree}.\exists \mathsf{has\_degree}^-.\mathsf{Graduate\_Student}$, T_C has three nodes, e_0, e_1, e_2, where e_0 is the root ρ_C of T_C, e_1 is a successor of e_0 and e_2 is a successor of e_1, the labelling of the nodes is given by $l(e_0) = \{\mathsf{Student}\}$, $l(e_1) = \emptyset$, and $l(e_2) = \{\mathsf{Graduate\_Student}\}$, and the labelling of the edges is given by $l(e_0, e_1) = \mathsf{has\_degree}$ and $l(e_1, e_2) = \mathsf{has\_degree}^-$, see Figure 3 (left).*

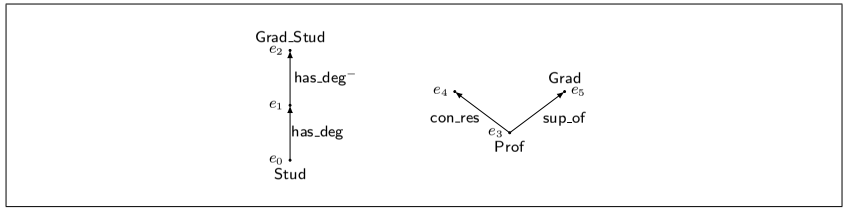


Figure 3: Illustration to Examples 6 (left) and 7 (right).

Conversely, every labelled finite directed tree T of the described form gives rise to a concept expression C_T in the following way: if T has a single node d labelled by $\{A_1, \dots, A_n\}$, then $C_T = A_1 \sqcap \dots \sqcap A_n$ (we treat $\top$ as the empty conjunction here, so if $l(d) = \emptyset$ then $C_T = \top$). Inductively, let d be the root of T labelled with $l(d) = \{A_1, \dots, A_n\}$, let $d_1, \dots, d_m$ be the successors of d, and let $l(d, d_1) = r_1, \dots, l(d, d_m) = r_m$. Assume $C_{d_1}, \dots, C_{d_m}$ are the concept expressions corresponding to the subtrees of T with roots $d_1, \dots, d_m$, respectively. Then $C_T = A_1 \sqcap \dots \sqcap A_n \sqcap \exists r_1.C_{d_1} \sqcap \dots \sqcap \exists r_m.C_{d_m}$.

Example 7 *Let T be the tree with root e_3 labelled by $\{\mathsf{Prof}\}$ and successors e_4, e_5 labelled by $\emptyset$ and $\{\mathsf{Graduate}\}$, respectively, and with edge labelling given by $l(e_3, e_4) = \mathsf{conduct\_research}$ and $l(e_3, e_5) = \mathsf{supervisor\_of}$. Then*

$$C_T = \mathsf{Prof} \sqcap \exists \mathsf{conduct\_research}.\top \sqcap \exists \mathsf{supervisor\_of}.\mathsf{Graduate};$$

see Figure 3 (right).

In what follows, we will not always distinguish explicitly between C and its tree representation T_C which allows us to speak, for example, about the nodes and subtrees of a concept expression.

One important use of the tree representation of concept expressions is that both the truth relation '$d \in C^{\mathcal{I}}$' and the entailment '$\mathcal{T} \models C \sqsubseteq D$' can be characterised in terms of homomorphisms between labelled trees and interpretations. A mapping h from a tree T_C corresponding to a concept expression C to an interpretation $\mathcal{I}$ is a *homomorphism* if $A \in l(d)$ implies $h(d) \in A^{\mathcal{I}}$ for every concept name A and $r = l(d, d')$ implies $(h(d), h(d')) \in r^{\mathcal{I}}$ for all role names r. The following characterisation of the truth relation $d \in C^{\mathcal{I}}$ by means of homomorphisms is well-known.

Lemma 8 *Let $\mathcal{I}$ be an interpretation, $d \in \Delta^{\mathcal{I}}$, and C a concept expression. Then $d \in C^{\mathcal{I}}$ if, and only if, there is a homomorphism from T_C to $\mathcal{I}$ mapping ρ_C to d.*

The proof is by a straightforward induction on the structure of C (Baader et al., 1999).

Example 9 *Consider the interpretation $\mathcal{I}$ from Example 2 and the tree representations of the concept expressions given in Figure 3. It can be seen that functions g and h defined as $g(e_0) = d_4$, $g(e_1) = d_5$, $g(e_2) = d_4$ and $h(e_3) = d_2$, $h(e_4) = d_3$, $h(e_5) = d_4$ are homomorphisms and so, by Lemma 8, $d_4 \in (\mathsf{Student} \sqcap \exists \mathsf{has\_degree}.\exists \mathsf{has\_degree}^-.\mathsf{Graduate\_Student})^{\mathcal{I}}$ and $d_2 \in (\mathsf{Prof} \sqcap \exists \mathsf{conduct\_research}.\top \sqcap \exists \mathsf{supervisor\_of}.\mathsf{Graduate})^{\mathcal{I}}$.*

It is also standard to characterise the subsumption relation $\emptyset \models C \sqsubseteq D$ (that is, subsumption relative to the empty TBox) by means of homomorphisms between the tree representations T_D and T_C. A *homomorphism* h from labelled tree T_1 to labelled tree T_2 is a mapping from the nodes of T_1 to the nodes of T_2 such that $A \in l(d)$ implies $A \in l(h(d))$ for every concept name A and $r = l(d, d')$ implies $r = l(h(d), h(d'))$ for every role r.

Lemma 10 *Let C and D be concept expressions. Then $\emptyset \models C \sqsubseteq D$ if, and only if, there is a homomorphism from T_D to T_C that maps ρ_D to ρ_C.*

The 'if' direction is essentially a consequence of Lemma 8 and the fact that the composition of two homomorphisms is again a homomorphism. For the 'only if' direction, one can consider T_C as an interpretation $\mathcal{I}$ and apply Lemma 8 (Baader et al., 1999).

Next, we characterise subsumption in the presence of TBoxes in terms of homomorphisms. To achieve this, we make use of the *canonical model* $\mathcal{I}_{C_0,\mathcal{T}}$ of a concept expression C_0 and a TBox $\mathcal{T}$. If $\mathcal{T} = \emptyset$, then we want $\mathcal{I}_{C_0,\mathcal{T}}$ to be T_{C_0} viewed as a tree-shaped interpretation which we denote by $\mathcal{I}_{C_0}$ rather than by $\mathcal{I}_{C_0,\mathcal{T}}$. More precisely, the domain of $\mathcal{I}_{C_0}$ is the set of nodes of T_{C_0} and

$$\begin{aligned} d \in A^{\mathcal{I}_{C_0}} \quad &\text{iff} \quad A \in l(d), \text{ for all } d \in \Delta^{\mathcal{I}_{C_0}} \text{ and concept names } A \\ (d, d') \in r^{\mathcal{I}_{C_0}} \quad &\text{iff} \quad r = l(d, d'), \text{ for all } d, d' \in \Delta^{\mathcal{I}_{C_0}} \text{ and roles names } r \end{aligned}$$

We call the root ρ_{C_0} of T_{C_0} the *root of* $\mathcal{I}_{C_0}$. If $\mathcal{T} \neq \emptyset$, then $\mathcal{I}_{C_0,\mathcal{T}}$ is obtained by extending $\mathcal{I}_{C_0}$ so that the CIs in $\mathcal{T}$ are satisfied. For example, if $\mathcal{T} = \{A \sqsubseteq \exists r.B\}$ and $C_0 = A$, then $\mathcal{I}_{C_0}$ is a single node ρ_{C_0} with $A^{\mathcal{I}_{C_0}} = \{\rho_{C_0}\}$ and $X^{\mathcal{I}_{C_0}} = \emptyset$ for all concept and role names X distinct from A. To define $\mathcal{I}_{C_0,\mathcal{T}}$ we add a node d to $\Delta^{\mathcal{I}_{C_0}}$ and set $B^{\mathcal{I}_{C_0,\mathcal{T}}} = \{d\}$ and $r^{\mathcal{I}_{C_0,\mathcal{T}}} = \{(\rho_{C_0}, d)\}$. In general, $\mathcal{I}_{C_0,\mathcal{T}}$ is defined as the limit of a sequence $\mathcal{I}_0, \mathcal{I}_1, \dots$ of interpretations, where $\mathcal{I}_0 = \mathcal{I}_{C_0}$. For the inductive definition of the sequence, assume that $\mathcal{I}_n$ has been defined. Then obtain $\mathcal{I}_{n+1}$ by applying one of the following rules once:

1. if $C \sqsubseteq D \in \mathcal{T}$ and $d \in C^{\mathcal{I}_n}$ but $d \notin D^{\mathcal{I}_n}$, then take the interpretation $\mathcal{I}_D$ and add it to $\mathcal{I}_n$ by identifying its root ρ_C with d. In more detail, assume that $\Delta^{\mathcal{I}_n} \cap \Delta^{\mathcal{I}_C} = \{d\}$ and $d = \rho_C$ and define $\mathcal{I}_{n+1}$ by setting, for all concept names A and role names r:

$$\Delta^{\mathcal{I}_{n+1}} = \Delta^{\mathcal{I}_n} \cup \Delta^{\mathcal{I}_C}, \quad A^{\mathcal{I}_{n+1}} = A^{\mathcal{I}_n} \cup A^{\mathcal{I}_C}, \quad r^{\mathcal{I}_{n+1}} = r^{\mathcal{I}_n} \cup r^{\mathcal{I}_C};$$

2. if $r \sqsubseteq s \in \mathcal{T}$ and $(d, d') \in r^{\mathcal{I}_n}$ but $(d, d') \notin s^{\mathcal{I}_n}$, then define $\mathcal{I}_{n+1}$ as $\mathcal{I}_n$ except that $s^{\mathcal{I}_{n+1}} := s^{\mathcal{I}_n} \cup \{(d, d')\}$ if s is a role name; otherwise there is a role name s_0 with $s = s_0^-$ and we define $\mathcal{I}_{n+1}$ as $\mathcal{I}_n$ except that $s_0^{\mathcal{I}_{n+1}} = s_0^{\mathcal{I}_n} \cup \{(d', d)\}$.

We assume that rule application is fair, that is, if a rule is applicable in a certain place, then it will indeed eventually be applied there. If for some $n > 0$ no rule is applicable then we set $\mathcal{I}_{n+1} = \mathcal{I}_n$. We obtain $\mathcal{I}_{C_0,\mathcal{T}}$ by setting for all concept names A and role names r:

$$\Delta^{\mathcal{I}_{C_0,\mathcal{T}}} = \bigcup_{n \geq 0} \Delta^{\mathcal{I}_n}, \quad A^{\mathcal{I}_{C_0,\mathcal{T}}} = \bigcup_{n \geq 0} A^{\mathcal{I}_n}, \quad r^{\mathcal{I}_{C_0,\mathcal{T}}} = \bigcup_{n \geq 0} r^{\mathcal{I}_n}.$$

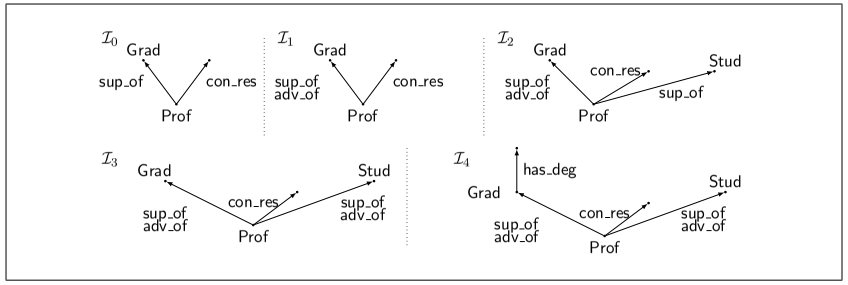


Figure 4: Canonical model construction for Example 11.

Note that the interpretation $\mathcal{I}_{C_0,\mathcal{T}}$ obtained in the limit is tree-shaped and might be infinite.[2] The following example illustrates the definition of $\mathcal{I}_{C_0,\mathcal{T}}$.

Example 11 *Consider the following TBox $\mathcal{T}$:*

$$\mathsf{Prof} \sqsubseteq \exists \mathsf{supervisor\_of}.\mathsf{Student} \quad (7)$$

$$\mathsf{Prof} \sqsubseteq \exists \mathsf{conduct\_research}.\top \quad (8)$$

$$\mathsf{Graduate} \sqsubseteq \exists \mathsf{has\_degree}.\top \quad (9)$$

$$\exists \mathsf{has\_degree}.\top \sqsubseteq \mathsf{Graduate} \quad (10)$$

$$\mathsf{supervisor\_of} \sqsubseteq \mathsf{advisor\_of} \quad (11)$$

and the concept expression

$$C_0 = \mathsf{Prof} \sqcap \exists \mathsf{conduct\_research}.\top \sqcap \exists \mathsf{supervisor\_of}.\mathsf{Graduate}.$$

Figure 4 illustrates the steps of the canonical model construction with $\mathcal{I}_0$ being $\mathcal{I}_{C_0}$ and $\mathcal{I}_4$ being the canonical model $\mathcal{I}_{C_0,\mathcal{T}}$.

The following lemma provides the announced characterisation of subsumption in the presence of TBoxes.

Lemma 12 *Let $\mathcal{T}$ be a TBox and C a concept expression. Then $\mathcal{I}_{C,\mathcal{T}}$ is a model of $\mathcal{T}$ and the following conditions are equivalent, for every concept expression D:*

1. *$\mathcal{T} \models C \sqsubseteq D$;*
2. *$\rho_C \in D^{\mathcal{I}_{C,\mathcal{T}}}$;*
3. *there is a homomorphism from T_D to $\mathcal{I}_{C,\mathcal{T}}$ that maps ρ_D to ρ_C.*

2. The exact shape of $\mathcal{I}_{C_0,\mathcal{T}}$ depends on the order of rule applications. However, all possible resulting interpretations $\mathcal{I}_{C_0,\mathcal{T}}$ are homomorphically equivalent and, as a consequence, the order of rule application is not important for our purposes.

The proof is completely standard, see, for example, the introduction by Krötzsch (2012). We only give a high-level overview. Using the construction of $\mathcal{I}_{C,\mathcal{T}}$, it is not hard to show that $\mathcal{I}_{C,\mathcal{T}}$ is a model of $\mathcal{T}$ and that $\rho_C \in C^{\mathcal{I}_{C,\mathcal{T}}}$. This implies '1 $\Rightarrow$ 2' and '2 $\Rightarrow$ 3' follows from Lemma 8. For '3 $\Rightarrow$ 1', one can show that for any model $\mathcal{I}$ of $\mathcal{T}$ and any $d \in C^{\mathcal{I}}$, there is a homomorphism from $\mathcal{I}_{C,\mathcal{T}}$ to $\mathcal{I}$ that maps ρ_C to d. In fact, one constructs a homomorphism to $\mathcal{I}$ from each of the interpretations $\mathcal{I}_0, \mathcal{I}_1, \ldots$ built during the construction of $\mathcal{I}_{C,\mathcal{T}}$, which is not hard by analysing the rules applied during that construction. The homomorphism built for each $\mathcal{I}_{n+1}$ extends that for $\mathcal{I}_n$ and thus we can take the unions of all those homomorphisms to obtain a homomorphism from $\mathcal{I}_{C,\mathcal{T}}$ to $\mathcal{I}$. It remains to compose homomorphisms and apply Lemma 8.

We are only going to use canonical models and Lemma 12 in the context of DL-Lite$^{\exists}_{\mathcal{R}}$ and DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$. Next, we identify a more subtle property of canonical models of DL-Lite$^{\exists}_{\mathcal{R}}$ and DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ TBoxes that we need later on. Very roughly speaking, it states a form of locality which is due to the fact that existential restrictions on the left-hand side of CIs in DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ are unqualified. Assume that $d \in (\exists r.D)^{\mathcal{I}_{C,\mathcal{T}}}$ for the canonical model $\mathcal{I}_{C,\mathcal{T}}$ of a concept expression C and TBox $\mathcal{T}$. Assume $d \in \Delta^{\mathcal{I}_C}$. We know that there exists a homomorphism h from $T_{\exists r.D}$ to $\mathcal{I}_{C,\mathcal{T}}$ mapping $\rho_{\exists r.D}$ to d. Then either h maps some elements of T_D into $\Delta^{\mathcal{I}_C}$ or it maps the whole tree T_D into $\Delta^{\mathcal{I}_{C,\mathcal{T}}} \setminus \Delta^{\mathcal{I}_C}$. We are interested in the latter case. The following lemma states that if $\mathcal{T}$ is a DL-Lite$^{\exists}_{\mathcal{R}}$ TBox, then there is a basic concept B with $d \in B^{\mathcal{I}_C}$ such that $\mathcal{T} \models B \sqsubseteq \exists r.D$. Thus, the question whether $d \in (\exists r.D)^{\mathcal{I}_{C,\mathcal{T}}}$ only depends on the concept names A with $d \in A^{\mathcal{I}_C}$ and the roles r with $d \in (\exists r.\top)^{\mathcal{I}_C}$. If $\mathcal{T}$ is a DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ TBox, it might not be sufficient to take a single basic concept but at least a set of basic concepts suffices (corresponding to the fact that DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ admits conjunctions on the left-hand side of CIs). This observation does not hold for $\mathcal{EL}$ TBoxes and is ultimately the reason for the fact that one cannot polynomially learn $\mathcal{EL}$ TBoxes. The following example illustrates this observation.

Example 13 *Consider the $\mathcal{EL}$ TBox $\mathcal{T} = \{\exists r.A \sqsubseteq A, A \sqsubseteq \exists r.B\}$ and let $C = \exists r.\exists r.A$. Then $\rho_C \in (\exists r.B)^{\mathcal{I}_{C,\mathcal{T}}}$ since $\mathcal{T} \models C \sqsubseteq \exists r.B$. We therefore find a homomorphism h from $T_{\exists r.B}$ to $\mathcal{I}_{C,\mathcal{T}}$ mapping $\rho_{\exists r.B}$ to ρ_C. This homomorphism maps T_B (which has a single node only) into $\Delta^{\mathcal{I}_C} \setminus \Delta^{\mathcal{I}_{C,\mathcal{T}}}$. The only basic concept B with $\rho_C \in B^{\mathcal{I}_C}$ is $\exists r.\top$ but clearly $\mathcal{T} \not\models \exists r.\top \sqsubseteq \exists r.B$ and so the observation we sketched above does not hold for $\mathcal{EL}$.*

We present this result in a more formal way. Let T_1 and T_2 be trees with labelling functions l_1 and l_2, respectively. We call T_1 a *subtree of* T_2 if the following conditions hold: $T_1 \subseteq T_2$, l_1 is the restriction of l_2 to T_1, and if $d \in T_1$ and d' is a successor of d in T_2, then d' is a successor of d in T_1 as well. The *one-neigbourhood* $N_{\mathcal{I}_C}(d)$ *of* $d \in \Delta^{\mathcal{I}_C}$ is the set of concept names A with $d \in A^{\mathcal{I}_C}$ and basic concepts $\exists r.\top$ such that there exists $d' \in \Delta^{\mathcal{I}_C}$ with $(d, d') \in r^{\mathcal{I}_C}$.

Lemma 14 *Let $\mathcal{T}$ be a DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ TBox, $D = \exists r.D'$ and assume $h : T_D \to \mathcal{I}_{C,\mathcal{T}}$ is such that $h(\rho_D) = d \in \Delta^{\mathcal{I}_C}$ and the image of the subtree $T_{D'}$ of T_D under h is included in $\Delta^{\mathcal{I}_{C,\mathcal{T}}} \setminus \Delta^{\mathcal{I}_C}$. Then there exists $I \subseteq N_{\mathcal{I}_C}(d)$ such that $\mathcal{T} \models \bigsqcap_{E \in I} E \sqsubseteq D$. Moreover, if $\mathcal{T}$ is a DL-Lite$^{\exists}_{\mathcal{R}}$ TBox, then there exists such a set $I \subseteq N_{\mathcal{I}_C}(d)$ with a single concept.*

Proof (sketch) This property of canonical models for DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ has been proved implicitly in many papers, for example the work of Artale et al. (2009). We give a sketch. Let N be the conjunction of all $E \in N_{\mathcal{I}_C}(d)$ and assume $\mathcal{T} \not\models N \sqsubseteq D$. Consider the canonical model $\mathcal{I}_{N,\mathcal{T}}$. By definition, the one-neighbourhoods of ρ_N in $\mathcal{I}_{N,\mathcal{T}}$ and d in $\mathcal{I}_{C,\mathcal{T}}$ coincide. Now observe that the canonical model of any concept expression C_0 and TBox $\mathcal{T}$ is obtained from $\mathcal{I}_{C_0}$ by hooking tree-shaped interpretations $\mathcal{I}_d$ with root d to every d in $\mathcal{I}_{C_0}$. As in DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ the concept expressions on the left-hand side of CIs are basic concepts, the interpretations $\mathcal{I}_d$ only depend on the one-neighbourhood $N_{\mathcal{I}_{C_0}}(d)$ of d in $\mathcal{I}_{C_0}$. Thus, the tree-shaped interpretations hooked to ρ_N in $\mathcal{I}_{N,\mathcal{T}}$ and to d in $\mathcal{I}_{C,\mathcal{T}}$ coincide and the homomorphism h given in Lemma 14 provides a homomorphism $h : \mathcal{I}_D \to \mathcal{I}_N$ such that $h(\rho_D) = \rho_N$. By Lemma 12, $\mathcal{T} \models N \sqsubseteq D$. We have derived a contradiction. For DL-Lite$^{\exists}_{\mathcal{R}}$ one only requires a single member of $N_{\mathcal{I}_C}(d)$ since the left-hand side of CIs in DL-Lite$^{\exists}_{\mathcal{R}}$ consists of a single basic concept only. ❑

We close the introduction of description logics with some comments about the choice of our languages. In the DL literature it is not uncommon to consider the weaker variant DL-Lite$_{\mathcal{R}}$ of DL-Lite$^{\exists}_{\mathcal{R}}$ in which only basic concepts are admitted on the right-hand side of CIs, but compound concepts are not. This is often without loss of generality since every DL-Lite$^{\exists}_{\mathcal{R}}$ TBox can be expressed in DL-Lite$_{\mathcal{R}}$ by using additional role names; in this way, standard DL-Lite$^{\exists}_{\mathcal{R}}$ reasoning tasks such as subsumption and conjunctive query answering can be reduced in polynomial time to the corresponding tasks for DL-Lite$_{\mathcal{R}}$. Such a reduction is not possible in the framework of exact learning that we are concerned with in this paper. In fact, in contrast to DL-Lite$^{\exists}_{\mathcal{R}}$ TBoxes, TBoxes in DL-Lite$_{\mathcal{R}}$ are trivially polynomial time learnable using either membership queries only or equivalence queries only as there are only polynomially many CIs and RIs over a given signature.

2.3 Exact Learning

We introduce the relevant notation for exact learning. A *learning framework* $\mathfrak{F}$ is a triple $(X, \mathcal{L}, \mu)$, where X is a set of *examples* (also called *domain* or *instance space*), $\mathcal{L}$ is a set of *concepts*,[3] and μ is a mapping from $\mathcal{L}$ to 2^X. We say that $x \in X$ is a *positive example* for $l \in \mathcal{L}$ if $x \in \mu(l)$ and a *negative example* for l if $x \notin \mu(l)$.

We give a formal definition of polynomial query and time learnability within a learning framework. Let $\mathfrak{F} = (X, \mathcal{L}, \mu)$ be a learning framework. We are interested in the exact identification of a *target* concept representation $l \in \mathcal{L}$ by posing queries to oracles. Let $\mathsf{MEM}_{\mathfrak{F},l}$ be the oracle that takes as input some $x \in X$ and returns 'yes' if $x \in \mu(l)$ and 'no' otherwise. A *membership query* is a call to the oracle $\mathsf{MEM}_{\mathfrak{F},l}$. Similarly, for every $l \in \mathcal{L}$, we denote by $\mathsf{EQ}_{\mathfrak{F},l}$ the oracle that takes as input a *hypothesis* concept representation $h \in \mathcal{L}$ and returns 'yes' if $\mu(h) = \mu(l)$ and a *counterexample* $x \in \mu(h) \oplus \mu(l)$ otherwise, where $\oplus$ denotes the symmetric set difference. There is no assumption regarding which counterexample in $\mu(h) \oplus \mu(l)$ is chosen by the oracle. An *equivalence query* is a call to the oracle $\mathsf{EQ}_{\mathfrak{F},l}$.

A *learning algorithm* for $\mathfrak{F}$ is a deterministic algorithm that takes no input, is allowed to make queries to $\mathsf{MEM}_{\mathfrak{F},l}$ and $\mathsf{EQ}_{\mathfrak{F},l}$ (without knowing what the target l to be learned is), and that eventually halts and outputs some $h \in \mathcal{L}$ with $\mu(h) = \mu(l)$. We say that $\mathfrak{F}$ is *exact learnable* if there is a learning algorithm for $\mathfrak{F}$ and that $\mathfrak{F}$ is *polynomial query learnable* if it is exact learnable by an algorithm A such that at every step the sum of the sizes of the inputs to membership and equivalence queries made by A up to that step is bounded by a polynomial $p(|l|, |x|)$, where l is the target and $x \in X$ is the largest counterexample seen so far (Arias, 2004). Finally, $\mathfrak{F}$ is *polynomial time learnable* if it is exact learnable by an algorithm A such that at every step (we count each call to an oracle as one step of computation) of computation the time used by A up to that step is bounded by a polynomial $p(|l|, |x|)$, where $l \in \mathcal{L}$ is the target and $x \in X$ is the largest counterexample seen so far. Clearly, a learning framework $\mathfrak{F}$ that is polynomial time learnable is also polynomial query learnable.

The aim of this paper is to study learnability of description logic TBoxes. In this context, each DL L gives rise to a learning framework $(X, \mathcal{L}, \mu)$, as follows: $\mathcal{L}$ is the set of all TBoxes formulated in L, X is the set of all CIs and RIs formulated in L, and $\mu(\mathcal{T}) = \{\alpha \in X \mid \mathcal{T} \models \alpha\}$ for every $\mathcal{T} \in \mathcal{L}$. Observe that $\mu(\mathcal{T}) = \mu(\mathcal{T}')$ iff $\mathcal{T} \equiv \mathcal{T}'$, for all TBoxes $\mathcal{T}$ and $\mathcal{T}'$. We say that L *TBoxes are polynomial query learnable* if the learning framework defined by L is polynomial query learnable, and likewise for polynomial time learnability. What does not show up directly in this representation is our assumption that the signature of the target TBox is known to the learner. Note that this is a standard assumption. For example, when learning propositional Horn formulas, it is common to assume that the variables in the target formula are known to the learner.

3. Learning DL-Lite$^{\exists}_{\mathcal{R}}$ TBoxes

We prove that DL-Lite$^{\exists}_{\mathcal{R}}$ TBoxes are polynomial query learnable. If inverse roles are disallowed in CIs and RIs of the target TBox then our algorithm runs in polynomial time and thus shows that TBoxes in this restricted language are polynomial time learnable. Without this restriction, polynomial time learnability remains open.

To simplify the presentation, we make two minor assumptions about the target TBox $\mathcal{T}$. We will show later how these assumptions can be overcomed. First, we assume that $\mathcal{T}$ does not entail non-trivial role equivalences, that is, there do not exist distinct roles r and s such that $\mathcal{T} \models r \equiv s$. This allows us to avoid dealing with classes of equivalent roles, simplifying notation. The second requirement is a bit more subtle. A concept inclusion is in *reduced form* if it is between basic concepts or its left-hand side is a concept name. A TBox $\mathcal{T}$ is in *named form* if all CIs in it are in reduced form and it contains a concept name A_r such that $A_r \equiv \exists r.\top \in \mathcal{T}$, for each role r. We assume that the target TBox is in named form and that all CIs considered by the learner are in reduced form. In particular, counterexamples returned by the oracle are immediately converted into this form.

Example 15 *Although the TBox $\mathcal{T}$ from Example 11 does not entail role equivalences and all its CIs are in reduced form, it is not in named form. To fix this, we introduce concept names $A_{\mathsf{supervisor\_of}}$, $A_{\mathsf{conduct\_research}}$ and $A_{\mathsf{advisor\_of}}$ and extend $\mathcal{T}$ with the following equivalences:*

$$A_{\mathsf{supervisor\_of}} \equiv \exists \mathsf{supervisor\_of}.\top \quad (12)$$

$$A_{\mathsf{conduct\_research}} \equiv \exists \mathsf{conduct\_research}.\top \quad (13)$$

3. The similarity of this name to 'concept expression' is accidental and should not be taken to mean that these two notions are closely related. Both is standard terminology in the respective area.

JMLR 18(201):1-63, 2018

Algorithm 1 Naïve learning algorithm for DL-Lite$^{\exists}_{\mathcal{R}}$

Input: A DL-Lite$^{\exists}_{\mathcal{R}}$ TBox $\mathcal{T}$ in named form given to the oracle; $\Sigma_{\mathcal{T}}$ given to the learner.
Output: TBox $\mathcal{H}$, computed by the learner, such that $\mathcal{T} \equiv \mathcal{H}$.

1: Compute $\mathcal{H}_{basic} = \{r \sqsubseteq s \mid \mathcal{T} \models r \sqsubseteq s\} \cup \{B_1 \sqsubseteq B_2 \mid \mathcal{T} \models B_1 \sqsubseteq B_2,\ B_1, B_2 \text{ basic}\}$
2: Set $\mathcal{H}_{add} = \emptyset$
3: **while** $\mathcal{H}_{basic} \cup \mathcal{H}_{add} \not\equiv \mathcal{T}$ **do**
4: Let $A \sqsubseteq C$ be the returned positive counterexample for $\mathcal{T}$ relative to $\mathcal{H}_{basic} \cup \mathcal{H}_{add}$
5: **if** there is $A \sqsubseteq C' \in \mathcal{H}_{add}$ **then**
6: Replace $A \sqsubseteq C'$ by $A \sqsubseteq C \sqcap C'$ in $\mathcal{H}_{add}$
7: **else**
8: Add $A \sqsubseteq C$ to $\mathcal{H}_{add}$
9: **end if**
10: **end while**
11: **return** $\mathcal{H} = \mathcal{H}_{basic} \cup \mathcal{H}_{add}$

$$A_{\mathsf{advisor\_of}} \quad \equiv \quad \exists \mathsf{advisor\_of}.\top. \tag{14}$$

Notice that Graduate *acts as a name for* $\exists$has_degree.$\top$ *so no new definition is needed for the role* has_degree. *The TBox* $\mathcal{T}' = \mathcal{T} \cup \{(12), (13), (14)\}$ *is in named form.*

To develop the learning algorithm it is instructive to start with a naïve version that does not always terminate but which can be refined to obtain the desired algorithm. This version is presented as Algorithm 1. Given the signature $\Sigma_{\mathcal{T}}$ of the target TBox $\mathcal{T}$, the learner starts with computing the set $\mathcal{H}_{basic}$ by posing to the oracle the membership query '$\mathcal{T} \models r \sqsubseteq s$?' for all $r, s \in \Sigma_{\mathcal{T}}$ and '$\mathcal{T} \models B_1 \sqsubseteq B_2$?' for all basic concept B_1, B_2 over $\Sigma_{\mathcal{T}}$. Observe that $\mathcal{T} \models \mathcal{H}_{basic}$. Then it enters the main **while** loop. Note that the condition '$\mathcal{H}_{basic} \cup \mathcal{H}_{add} \not\equiv \mathcal{T}$?' in Line 3 is implemented using an equivalence query to the oracle, and that $A \sqsubseteq C$ in Line 4 refers to the counterexample returned by the oracle in the case that equivalence does not hold. The counterexample must be positive since we maintain the invariant $\mathcal{T} \models \mathcal{H}_{basic} \cup \mathcal{H}_{add}$ throughout the run of the algorithm. If there is no CI of the form $A \sqsubseteq C'$ in $\mathcal{H}_{add}$ then $A \sqsubseteq C$ is added to $\mathcal{H}_{add}$, otherwise $A \sqsubseteq C \sqcap C'$ is (Lines 6 and 8). The algorithm terminates when $\mathcal{H}_{basic} \cup \mathcal{H}_{add} \equiv \mathcal{T}$, implying that the target TBox has been learned.

Example 16 For the TBox $\mathcal{T}'$ from Example 15, Algorithm 1 first computes $\mathcal{H}_{basic}$ which coincides with $\mathcal{T}'$ except that Prof $\sqsubseteq \exists$supervisor_of.Student is not included since the concept $\exists$supervisor_of.Student is not basic. In the main loop the only counterexamples to $\mathcal{H}_{basic} \cup \mathcal{H}_{add} \equiv \mathcal{T}'$ are (up to logical equivalence modulo $\mathcal{H}_{basic}$) the CIs

$$\mathsf{Prof} \sqsubseteq \exists \mathsf{supervisor\_of.Student}, \quad \mathsf{Prof} \sqsubseteq \exists \mathsf{advisor\_of.Student}.$$

If the oracle returns the first CI in the first iteration, the algorithm terminates immediately having learned $\mathcal{T}'$. Otherwise the oracle first returns the second CI and then returns the first CI in the second iteration. The algorithm terminates with

$$\mathcal{H}_{add} = \{\mathsf{Prof} \sqsubseteq \exists \mathsf{supervisor\_of.Student} \sqcap \exists \mathsf{advisor\_of.Student}\}$$

which is equivalent to $\mathcal{T}'$.

We now consider five examples on which this naïve algorithm fails to terminate after polynomially many steps (or at all), each example motivating a different *modification step* that is added to Algorithm 1 after Lines 4 and 5. The final, corrected algorithm is given as Algorithm 2 below. Each modification step takes as input a counterexample $A \sqsubseteq C$ against the equivalence $\mathcal{H}_{basic} \cup \mathcal{H}_{add} \equiv \mathcal{T}$ and modifies it by posing membership queries to the oracle to obtain a CI $A' \sqsubseteq C'$ which is still a counterexample and has additional desired properties. CIs satisfying all five additional properties will be called $\mathcal{T}$*-essential*. The five modification steps are of three different types:

1. two *saturations steps*: the underlying tree of T_C is left unchanged but the labelling is modified by adding concept names to node labels or replacing roles in edge labels;

2. two *merging steps*: nodes in the tree T_C are merged, resulting in a tree with fewer nodes;

3. a *decomposition step*: T_C is replaced by a subtree or a subtree is removed from T_C, and the concept name A on the left-hand side might be replaced.

The saturation and merging steps do not change the left-hand side A of the CI $A \sqsubseteq C$ and result in a logically stronger CI $A \sqsubseteq C'$ in the sense that $\emptyset \models C' \sqsubseteq C$. In contrast, the decomposition step can be regarded as a reset operation in which also the left-hand side can change and which is logically not related to $A \sqsubseteq C$. We start with an example which motivates the first saturation step.

Example 17 *Let*

$$\mathcal{T} = \{A \sqsubseteq \exists r.A\} \cup \mathcal{T}_{\text{NF}},$$

where $\mathcal{T}_{\text{NF}} = \{A_r \equiv \exists r.\top\}$ *ensures that* $\mathcal{T}$ *is in named form. First, Algorithm 1 computes* $\mathcal{H}_{basic}$*. Afterwards the oracle can provide for the n-th equivalence query in the while loop the positive counterexample* $A \sqsubseteq \exists r^{n+1}.\top$*, for any* $n \geq 1$ *(here we set inductively* $\exists r^{m+1}.\top = \exists r.\exists r^m.\top$ *and* $\exists r^1.\top = \exists r.\top$*). Thus, the algorithm does not terminate.*

Informally, the problem for the learner in Example 17 is that the concepts $\exists r^n.\top$ used in the counterexamples $A \sqsubseteq \exists r^n.\top$ get larger and larger, but still none of the counterexamples implies $A \sqsubseteq \exists r.A$. We address this problem by saturating T_C with implied concept names. For the following discussion, recall that we do not distinguish between the concept expression C and its tree representation T_C. For example, if we say that C' is obtained from C by adding a concept name B to the label of node d in C, then this stands for: C' is the concept expression corresponding to the tree obtained from T_C by adding B to the label of d in T_C.

Definition 18 (Concept saturation for $\mathcal{T}$) *A CI* $A \sqsubseteq C$ *is* concept saturated for $\mathcal{T}$ *if* $\mathcal{T} \models A \sqsubseteq C$ *and* $\mathcal{T} \not\models A \sqsubseteq C'$ *for any* C' *obtained from* C *by adding a concept name to the label of some node of* C*. A CI* $A \sqsubseteq C'$ *is a* concept saturation for $\mathcal{T}$ *of a CI* $A \sqsubseteq C$ *if it is concept saturated for* $\mathcal{T}$ *and* C' *is obtained from* C *by adding concept names to the labels of some nodes in* C*.*

Observe that the learner can compute a concept saturation $A \sqsubseteq C'$ for $\mathcal{T}$ from a counterexample $A \sqsubseteq C$ by posing polynomially many membership queries to the oracle: it simply asks for any node d in T_C and concept name $E \in \Sigma_{\mathcal{T}}$ whether $\mathcal{T} \models A \sqsubseteq C^{E,d}$, where $C^{E,d}$ is obtained from C by adding E to the label of d. If the answer is positive, it replaces C by $C^{E,d}$ and proceeds. Note that there can be several concept saturations of a given CI $A \sqsubseteq C$ for a TBox $\mathcal{T}$. Consider, for example, $\mathcal{T} = \{A \sqsubseteq \exists r.(B \sqcap \exists r.\top), A \sqsubseteq \exists r.\exists r.B\}$ and the CI $A \sqsubseteq \exists r.\exists r.\top$. Then both $A \sqsubseteq A \sqcap \exists r.(B \sqcap \exists r.\top)$ and $A \sqsubseteq A \sqcap \exists r.\exists r.B$ are concept saturations of $A \sqsubseteq \exists r.\exists r.\top$ for $\mathcal{T}$ since $\mathcal{T} \not\models A \sqsubseteq \exists r.(B \sqcap \exists r.B)$.

Example 19 (Example 17 continued) *The CIs $A \sqsubseteq \exists r^n.\top$ are not concept saturated for $\mathcal{T}$. For example, for $n = 2$, a concept saturation of $A \sqsubseteq \exists r.\exists r.\top$ for $\mathcal{T}$ is given by $A \sqsubseteq A \sqcap A_r \sqcap \exists r.(A \sqcap A_r \sqcap \exists r.(A_r \sqcap A))$ (in fact, this is the only concept saturation for $\mathcal{T}$ of $A \sqsubseteq \exists r.\exists r.\top$). Now observe that if the CI $A \sqsubseteq C$ returned by the oracle to the first equivalence query is transformed by the learner into a concept saturated CI (after Line 4), then the TBox $\mathcal{T} = \{A \sqsubseteq \exists r.A\} \cup \mathcal{T}_{\text{NF}}$ is learned in one step: the only possible counterexamples returned by the oracle to the equivalence query $\mathcal{H}_{basic} \equiv \mathcal{T}$ are of the form $A \sqsubseteq C_1 \sqcap \exists r.C_2$ for some concepts C_1 and C_2. Concept saturation results in a concept of the form $C_1' \sqcap \exists r.(A \sqcap C_2')$ and $\{A \sqsubseteq C_1' \sqcap \exists r.(A \sqcap C_2')\} \models A \sqsubseteq \exists r.A$.*

The following example motivates the second saturation step. Here and in the subsequent examples we do not transform the TBoxes into named form as this does not effect the argument and simplifies presentation.

Example 20 *Consider for $n \geq 1$ the TBoxes*

$$\mathcal{T}_n \quad = \quad \{A \sqsubseteq \exists e_1.\exists e_2. \ldots \exists e_n.\top\} \cup \{e_i \sqsubseteq r_i, e_i \sqsubseteq s_i \mid 1 \leq i \leq n\}.$$

For $M \subseteq \{1, \ldots, n\}$, set $C_M = \exists t_1.\exists t_2. \ldots \exists t_n.\top$, where $t_i = r_i$ if $i \in M$ and $t_i = s_i$ if $i \notin M$. Then for the first 2^n equivalence queries in the while loop the oracle can provide a positive counterexample $A \sqsubseteq C_M$ by always choosing a fresh set $M \subseteq \{1, \ldots, n\}$.

Intuitively, the problem for the learner in Example 20 is that there are exponentially many logically incomparable CIs that are entailed by $\mathcal{T}_n$ but do not entail $A \sqsubseteq \exists e_1.\exists e_2. \ldots \exists e_n.\top$. A step towards resolving this problem is to replace the roles r_i and s_i by the roles e_i in the counterexamples $A \sqsubseteq C_M$.

Definition 21 (Role saturation for $\mathcal{T}$) *A CI $A \sqsubseteq C$ is* role saturated for $\mathcal{T}$ *if $\mathcal{T} \models A \sqsubseteq C$ and $\mathcal{T} \not\models A \sqsubseteq C'$ for any C' obtained from C by replacing in some edge label a role r by a role $s \neq r$ with $\mathcal{T} \models s \sqsubseteq r$. A CI $A \sqsubseteq C'$ is a* role saturation for $\mathcal{T}$ *of a CI $A \sqsubseteq C$ if it is role saturated for $\mathcal{T}$ and C' is obtained from C by replacing in some edge labels a role r by a role s with $\mathcal{T} \models s \sqsubseteq r$.*

Similarly to concept saturation, the learner can compute a role saturated $A \sqsubseteq C'$ from a counterexample $A \sqsubseteq C$ by posing polynomially many membership queries. Again, there can be several role saturations of a given $A \sqsubseteq C$ for a TBox $\mathcal{T}$. For example, if

$$\mathcal{T} = \{s_1 \sqsubseteq r, s_2 \sqsubseteq r, A \sqsubseteq \exists s_1.B, A \sqsubseteq \exists s_2.B\},$$

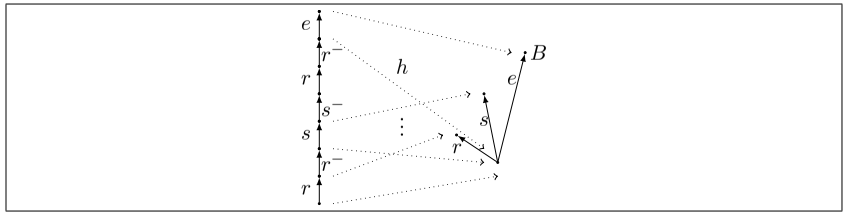


Figure 5: Tree representation of $C_{\{1,3\}}$ and homomorphism to $\exists r.\top \sqcap \exists s.\top \sqcap \exists e.B$.

then $A \sqsubseteq \exists s_1.B$ and $A \sqsubseteq \exists s_2.B$ are role saturations of $A \sqsubseteq \exists r.B$ for $\mathcal{T}$. Observe that in Example 20 the single role saturation of any $A \sqsubseteq C_M$ is $A \sqsubseteq \exists e_1.\exists e_2. \ldots \exists e_n.\top$. Thus, if the counterexample $A \sqsubseteq C_M$ returned by the first equivalence query is transformed into a role saturated CI, then the algorithm terminates after one step. We now introduce and motivate our two merging rules.

Example 22 Consider the TBox

$$\mathcal{T} = \{A \sqsubseteq \exists r.\top \sqcap \exists s.\top \sqcap \exists e.B\}.$$

and fix an $n \geq 1$. For $M \subseteq \{1, \ldots, n\}$, set $C_M = \exists t_1.\exists t_1^-.\exists t_2.\exists t_2^-. \ldots \exists t_n.\exists t_n^-.\exists e.\top$, where $t_i = r$ if $i \in M$ and $t_i = s$ if $i \notin M$. Figure 5 (left) illustrates the concept expression $C_{\{1,3\}}$, assuming $n = 3$. By Lemma 10, $\mathcal{T} \models A \sqsubseteq C_M$ since there is a homomorphism h from T_{C_M} to the labelled tree that corresponds to $\exists r.\top \sqcap \exists s.\top \sqcap \exists e.B$, as shown in Figure 5. Thus, the oracle can provide for the first 2^n equivalence queries a positive counterexample $A \sqsubseteq C_M$ by always choosing a fresh set $M \subseteq \{1, \ldots, n\}$.

The problem for the learner in Example 22 is similar to that in Example 20: there are exponentially many logically incomparable CIs that are entailed by $\mathcal{T}$ but do not entail $A \sqsubseteq \exists r.\top \sqcap \exists s.\top \sqcap \exists e.B$. A step towards solving this problem is to merge the predecessor and successor nodes of a node if the edge labels are inverse to each other and the resulting CI is still implied by the TBox.

Definition 23 (Parent/Child Merging for $\mathcal{T}$) *A concept C' is* obtained from a concept C by parent/child merging *if C' is obtained from C by choosing nodes d, d', d'' such that d is an r-successor of d', and d'' is an r^--successor of d, for some role r, and then removing d'', setting $l(d') = l(d') \cup l(d'')$, and making every s-successor e of d'' in C an s-successor of d', for any role s.*

Let $A \sqsubseteq C$ be a CI with $\mathcal{T} \models A \sqsubseteq C$. A CI $A \sqsubseteq C'$ is obtained from $A \sqsubseteq C$ by parent/child merging *if $\mathcal{T} \models A \sqsubseteq C'$ and C' is obtained from C by parent/child merging. We say that $A \sqsubseteq C$ is* parent/child merged for $\mathcal{T}$ *if there is no $A \sqsubseteq C'$ with $C \neq C'$ that can be obtained from $A \sqsubseteq C$ by parent/child merging.*

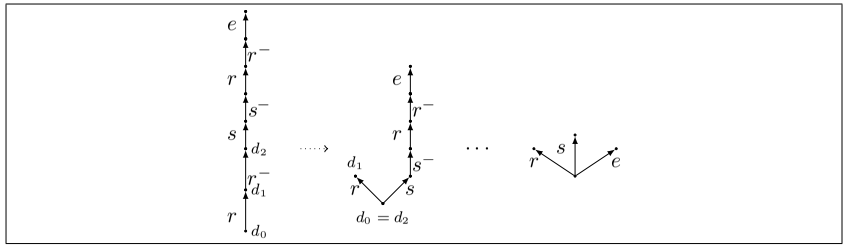


Figure 6: Parent/Child Merging of $C_{\{1,3\}}$ for $\mathcal{T}$

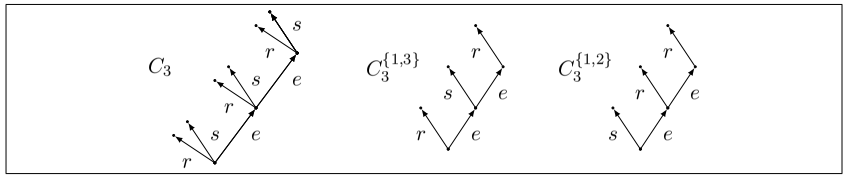


Figure 7: Tree representation of the concept expressions C_3, $C_3^{\{1,3\}}$ and $C_3^{\{1,2\}}$.

Note that when C' is obtained from C by parent/child merging with d' and d'' as in Definition 23, then $\emptyset \models C' \sqsubseteq C$. To show this, one can use Lemma 10 and the natural homomorphism h from T_C to $T_{C'}$, that is, the identity except that $h(d'') = d'$.

Similarly to the saturation operations, the learner can compute a parent/child merged $A \sqsubseteq C'$ by posing polynomially many membership queries. In Example 22 the parent/child merging of any $A \sqsubseteq C_M$ with $\emptyset \neq M \neq \{1, \dots, n\}$ is $A \sqsubseteq \exists r.\top \sqcap \exists s.\top \sqcap \exists e.\top$, as illustrated in Figure 6: in the first step the nodes d_0 and d_2 are merged, two additional merging steps give $\exists r.\top \sqcap \exists s.\top \sqcap \exists e.\top$. The following example motivates the second merging operation.

Example 24 *Define concept expressions C_i by induction as follows:*

$$C_1 = \exists r.\top \sqcap \exists s.\top, \quad C_{i+1} = C_1 \sqcap \exists e.C_i$$

and let

$$\mathcal{T}_n = \{A \sqsubseteq \exists e.C_n\}.$$

For $M \subseteq \{1, \dots, n\}$, set $C_1^M = \exists r.\top$ if $1 \in M$ and $C_1^M = \exists s.\top$ if $1 \notin M$. Also, let $C_{i+1}^M = \exists r.\top \sqcap \exists e.C_i^M$ if $i+1 \in M$ and $C_{i+1}^M = \exists s.\top \sqcap \exists e.C_i^M$ if $i+1 \notin M$, $1 \leq i < n$. Figure 7 illustrates concept expressions of the form C_n and C_n^M.

As an answer to the first 2^n equivalence queries the oracle can compute a positive counterexample $A \sqsubseteq \exists e.C_n^M$ by always choosing a fresh set $M \subseteq \{1, \dots, n\}$.

To deal with this example we introduce a modification step that identifies siblings in C rather than a parent and a child.

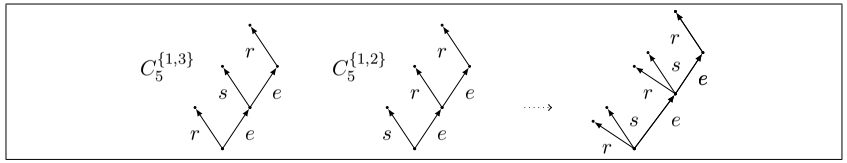


Figure 8: Sibling Merging $C_3^{\{1,3\}}$ and $C_3^{\{1,2\}}$ for $\mathcal{T}$.

Definition 25 (Sibling Merging for $\mathcal{T}$) *A concept C' is* obtained from a concept C by sibling merging *if C' is obtained from C by choosing nodes d, d', d'' such that d' and d'' are r-successors of d, for some role r, and then removing d'', setting $l(d') = l(d') \cup l(d'')$, and making every s-successor e of d'' in T_C an s-successor of d', for any role s.*

Let $A \sqsubseteq C$ be a CI with $\mathcal{T} \models A \sqsubseteq C$. A CI $A \sqsubseteq C'$ is obtained from $A \sqsubseteq C$ by sibling merging *if $\mathcal{T} \models A \sqsubseteq C'$ and C' is obtained from C by sibling merging. We say that $A \sqsubseteq C$ is* sibling merged for $\mathcal{T}$ *if there is no $A \sqsubseteq C'$ with $C \neq C'$ that can be obtained from $A \sqsubseteq C$ by sibling merging.*

It can be verified that when C' is obtained from C by sibling merging, then $\emptyset \models C' \sqsubseteq C$.

In Example 24 the counterexamples $A \sqsubseteq C_n^M$ are actually sibling merged for $\mathcal{T}_n$. Thus, producing a sibling merged $A \sqsubseteq C'$ directly from the counterexamples returned by the oracle does not overcome the problem illustrated by the example. Instead, we apply sibling merging after Line 5 of the algorithm: instead of adding $A \sqsubseteq C \sqcap C'$ to $\mathcal{H}_{add}$, the learner computes a sibling merged $A \sqsubseteq D$ from this CI and adds it to $\mathcal{H}_{add}$. For Example 24, this is illustrated in Figure 8. Clearly, after at most $n+1$ counterexamples, the learner has added $A \sqsubseteq \exists e.C_n$, as required.

Finally, we need a decomposition rule. The following variant of Example 17 illustrates that the four modification steps introduced so far do not yet lead to a polynomial learning algorithm even if they are applied both after Line 4 and after Line 5 in Algorithm 1.

Example 26 *Let*

$$\mathcal{T} = \{A \sqsubseteq B, B \sqsubseteq \exists r.B\}.$$

The oracle can provide for the n-th equivalence query the positive counterexample $A \sqsubseteq C_{B,n}$, where $C_{B,n} = A \sqcap D_{B,n}$ and, inductively, $D_{B,0} = B$ and $D_{B,n+1} = B \sqcap \exists r.D_{B,n}$, for any $n \geq 0$. The algorithm does not terminate even with the four modification steps introduced above applied after Lines 4 and 5: the CIs $A \sqsubseteq C_{B,n}$ are concept and role saturated and they are parent/child and sibling merged.

The problem illustrated in Example 26 is that so far the learning algorithm attempts to learn $\mathcal{T}$ without ever considering to add to $\mathcal{H}_{add}$ a CI whose left-hand side is B (rather than A). To deal with this problem we introduce a 'reset step' that, in contrast to the previous modification steps, can lead to a different left-hand side and also to a CI that does not imply the original CI given $\mathcal{T}$, as in all previous modification steps.

Definition 27 (Decomposed CI for $\mathcal{T}$) *Let $A \sqsubseteq C$ be a CI with $\mathcal{T} \models A \sqsubseteq C$. We say that $A \sqsubseteq C$ is* decomposed for $\mathcal{T}$ *if for every non-root node d in C, every concept name*

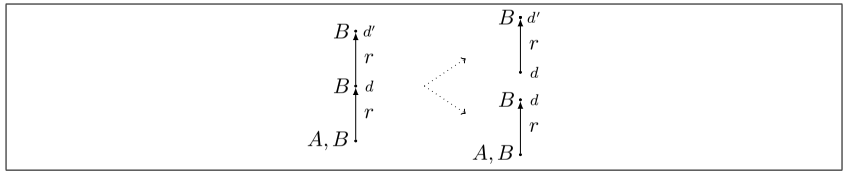


Figure 9: Illustration of decomposition of CIs. Here $C|^{-}_{d'\downarrow} = A \sqcap B \sqcap \exists r.B$ and $C' = B$.

$A' \in l(d)$, and every r-successor d' of d in C, we have $\mathcal{T} \not\models A' \sqsubseteq \exists r.C'$ where C' corresponds to the subtree of C rooted at d'.

In contrast to the previous four modification steps, the membership queries used by the learner to obtain a decomposed CI do not only depend on $\mathcal{T}$ but also on the hypothesis $\mathcal{H}_{add} \cup \mathcal{H}_{basic}$ computed up to that point: starting from CI $A \sqsubseteq C$, the learner takes a non-root node d in C, a concept name $A' \in l(d)$, and an r-successor d' of d in C, and then checks using a membership query whether $\mathcal{T} \models A' \sqsubseteq \exists r.C'$, where C' is the subtree rooted at d' in C. If the check succeeds, $A \sqsubseteq C$ is replaced by

(a) $A' \sqsubseteq \exists r.C'$ if $\mathcal{H}_{basic} \cup \mathcal{H}_{add} \not\models A' \sqsubseteq \exists r.C'$; and otherwise by

(b) $A \sqsubseteq C|^{-}_{d'\downarrow}$, where $C|^{-}_{d'\downarrow}$ is obtained from C by removing the subtree rooted in d' from C.

Note that $\{A \sqsubseteq C|^{-}_{d'\downarrow}, A' \sqsubseteq \exists r.C'\} \models A \sqsubseteq C$. Thus, one of the CIs $A \sqsubseteq C|^{-}_{d'\downarrow}$ and $A' \sqsubseteq \exists r.C'$ is not entailed by $\mathcal{H}_{basic} \cup \mathcal{H}_{add}$, and this is the CI that replaces the original CI.

In Example 26, assume that the oracle returns $A \sqsubseteq C$ with $C = A \sqcap B \sqcap \exists r.(B \sqcap \exists r.B)$ as the first counterexample. The tree T_C corresponding to C is shown on the left-hand side of Figure 9. This CI is not decomposed for $\mathcal{T}$: the label of node d contains B, the concept C' rooted in d' in C is B and $\mathcal{T} \models B \sqsubseteq \exists r.B$. Since $\mathcal{H} \cup \mathcal{H}_{add} \not\models B \sqsubseteq \exists r.B$, Case (a) applies and $A \sqsubseteq B$ is replaced by $B \sqsubseteq \exists r.B$.

This finishes the description of the modification steps. It turns out that they cure all problems with the initial version of the algorithm and enable polynomial query learnability.

Definition 28 *A DL-Lite$^{\exists}_{\mathcal{R}}$ CI is* $\mathcal{T}$-essential *if it is concept saturated, role saturated, parent/child merged, sibling merged, and decomposed for $\mathcal{T}$.*

After Lines 4 and 5 of Algorithm 1, we need to make the CI currently considered $\mathcal{T}$-essential, by exhaustively applying the modification steps described above in all possible orders. The resulting refined version of the learning algorithm is shown as Algorithm 2. We next analyse the properties of this algorithm.

Polynomial Query Bound on the Algorithm

If Algorithm 2 terminates, then it obviously has found a TBox $\mathcal{H}_{basic} \cup \mathcal{H}_{add}$ that is logically equivalent to $\mathcal{T}$. It thus remains to show that the algorithm terminates after polynomially many polynomial size queries. Observe that $\mathcal{H}_{add}$ contains at most one CI $A \sqsubseteq C$ for each concept name A. At each step in the while loop, either some $A' \sqsubseteq C'$ is added to $\mathcal{H}_{add}$ such

Algorithm 2 The learning algorithm for DL-Lite$^{\exists}_{\mathcal{R}}$

Input: A DL-Lite$^{\exists}_{\mathcal{R}}$ TBox $\mathcal{T}$ in named form given to the oracle; $\Sigma_{\mathcal{T}}$ given to the learner.
Output: TBox $\mathcal{H}$, computed by the learner, such that $\mathcal{T} \equiv \mathcal{H}$.

1: Compute $\mathcal{H}_{basic} = \{r \sqsubseteq s \mid \mathcal{T} \models r \sqsubseteq s\} \cup \{B_1 \sqsubseteq B_2 \mid \mathcal{T} \models B_1 \sqsubseteq B_2,\ B_1, B_2 \text{ basic}\}$
2: Set $\mathcal{H}_{add} = \emptyset$
3: **while** $\mathcal{H}_{basic} \cup \mathcal{H}_{add} \not\equiv \mathcal{T}$ **do**
4: Let $A \sqsubseteq C$ be the returned positive counterexample for $\mathcal{T}$ relative to $\mathcal{H}_{basic} \cup \mathcal{H}_{add}$
5: Find a $\mathcal{T}$-essential CI $A' \sqsubseteq C'$ such that $\mathcal{H}_{basic} \cup \mathcal{H}_{add} \not\models A' \sqsubseteq C'$
6: **if** there is $A' \sqsubseteq C'' \in \mathcal{H}_{add}$ **then**
7: Find $\mathcal{T}$-essential CI $A' \sqsubseteq C^*$ such that $\emptyset \models C^* \sqsubseteq C'' \sqcap C'$
8: Replace $A' \sqsubseteq C''$ by $A' \sqsubseteq C^*$ in $\mathcal{H}_{add}$
9: **else**
10: Add $A' \sqsubseteq C'$ to $\mathcal{H}_{add}$
11: **end if**
12: **end while**
13: **return** $\mathcal{H} = \mathcal{H}_{basic} \cup \mathcal{H}_{add}$

that no CI with A' on the left-hand side existed in $\mathcal{H}_{add}$ before (Line 10) or an existing CI $A' \sqsubseteq C''$ in $\mathcal{H}_{add}$ is replaced by a fresh CI $A' \sqsubseteq C^*$ with $\emptyset \models C^* \sqsubseteq C''$.

We start with showing that Lines 5 and 7 can be implemented with polynomially many membership queries. The next lemma addresses Line 5.

Lemma 29 *Given a positive counterexample $A \sqsubseteq C$ for $\mathcal{T}$ relative to $\mathcal{H}_{basic} \cup \mathcal{H}_{add}$, one can construct a $\mathcal{T}$-essential counterexample $A' \sqsubseteq C'$ using only polynomially many polynomial size membership queries in $|C| + |\mathcal{T}|$.*

Proof Let $A \sqsubseteq C$ be a positive counterexample for $\mathcal{T}$ relative to $\mathcal{H}_{basic} \cup \mathcal{H}_{add}$ and assume the five modification steps introduced above are applied exhaustively by posing membership queries to the oracle. Observe that the number of applications of modifications steps is bounded polynomially in $|C| \times |\mathcal{T}|$. To show this, let n_C be the number of nodes in T_C. Then $n_{C''} = n_{C'}$ if $A'' \sqsubseteq C''$ is obtained from $A' \sqsubseteq C'$ by a concept or role saturation step and $n_{C''} < n_{C'}$ if $A'' \sqsubseteq C''$ is obtained from $A' \sqsubseteq C'$ by a merging or decomposition step. Thus, the number of applications of merging and decomposition steps is bounded by n_C and the number of applications of concept and role saturated steps is bounded by $|\Sigma_{\mathcal{T}}| \times n_C$ and $|\Sigma_{\mathcal{T}}| \times n_C^2$, respectively. Thus, after at most $n_C + |\Sigma_{\mathcal{T}}| \times n_C + |\Sigma_{\mathcal{T}}| \times n_C^2$ steps no modification step is applicable and the final CI is $\mathcal{T}$-essential. We verify that it is also a positive counterexample for $\mathcal{T}$ relative to $\mathcal{H}_{basic} \cup \mathcal{H}_{add}$. It suffices to show that the CI resulting from each single modification step is entailed by $\mathcal{T}$, but not by $\mathcal{H}_{basic} \cup \mathcal{H}_{add}$. The former has been shown when we introduced the modification steps. Regarding the latter, in the first four modification steps we have $\mathcal{H}_{basic} \models C' \sqsubseteq C$ if $A \sqsubseteq C$ is replaced by $A \sqsubseteq C'$. Hence $\mathcal{H}_{basic} \cup \mathcal{H}_{add} \not\models A \sqsubseteq C'$. For the decomposition step, we have already argued, after Definition 27, that the added CI is not entailed by $\mathcal{H}_{basic} \cup \mathcal{H}_{add}$. ❑

The following lemma addresses Line 7.

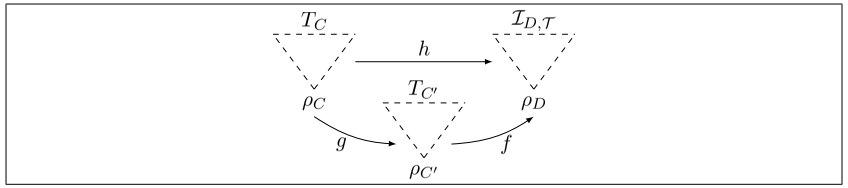


Figure 10: Homomorphisms in Lemma 31.

Lemma 30 *Assume that $A \sqsubseteq C_1$ and $A \sqsubseteq C_2$ are $\mathcal{T}$-essential. Then one can construct a $\mathcal{T}$-essential $A \sqsubseteq C$ such that $\emptyset \models C \sqsubseteq C_1 \sqcap C_2$ using polynomially many polynomial size membership queries in $|C_1| + |C_2|$.*

Proof We start with $A \sqsubseteq C_1 \sqcap C_2$. Using the fact that C_1 and C_2 are both $\mathcal{T}$-essential, one can show that this CI is (i) concept saturated for $\mathcal{T}$, (ii) role saturated for $\mathcal{T}$, (iii) parent/child merged for $\mathcal{T}$, and (iv) decomposed for $\mathcal{T}$. Assume, for example, that $A \sqsubseteq C_1 \sqcap C_2$ is not concept saturated. Then one can add a new concept name A' to the label $l(d)$ for some node d in $T_{C_1 \sqcap C_2}$ and $\mathcal{T} \models A \sqsubseteq C'$ for the resulting concept C'. Clearly d is a node in T_{C_1} or in T_{C_2}. Assume without loss of generality that d is in T_{C_1} and let C_1' be the concept obtained from C_1 by adding A' to $l(d)$. Then $\mathcal{T} \models A \sqsubseteq C_1'$ since $\mathcal{T} \models A \sqsubseteq C'$ which contradicts the assumption that $A \sqsubseteq C_1$ is concept saturated. The remaining three modification steps are considered similarly. We now exhaustively apply the modification step 'Sibling merging for $\mathcal{T}$' and use the resulting CI as the desired $A \sqsubseteq C$. Similarly to the argument above one can show that a CI with properties (i)–(iv) still has those properties after applying sibling merging. Thus, $A \sqsubseteq C$ is $\mathcal{T}$-essential. We have argued in the proof of Lemma 29 already that the number of applications of a sibling merging step to a CI of the form $A \sqsubseteq D$ is bounded by the number of nodes in T_D. Thus, the number of modification steps is bounded polynomially in $|C_1| + |C_2|$. ❑

To analyse the algorithm further we first prove a polynomial upper bound on the size of $\mathcal{T}$-essential CIs. To this end, we require the notion of an isomorphic embedding and an auxiliary lemma. A homomorphism $h : T_C \to \mathcal{I}$ is an *isomorphic embedding for* $\mathcal{T}$ if it is injective, $A \in l(d)$ if $h(d) \in A^{\mathcal{I}}$ for all concept names A, and for $r = l(d, d')$ it holds that $\mathcal{T} \models r \sqsubseteq s$ for all $(h(d), h(d')) \in s^{\mathcal{I}}$. The following lemma shows that for $\mathcal{T}$-essential CIs $A \sqsubseteq C$ and any D that interpolates between A and C (meaning that $\mathcal{T} \models A \sqsubseteq D$ and $\mathcal{T} \models D \sqsubseteq C$) the homomorphism h from T_C to $\mathcal{I}_{D,\mathcal{T}}$ that witnesses $\rho_D \in C^{\mathcal{I}_{D,\mathcal{T}}}$ (see Lemma 12) is an isomorphic embedding.

Lemma 31 *Assume the $A \sqsubseteq C$ is $\mathcal{T}$-essential, $\mathcal{T} \models A \sqsubseteq D$ and $\mathcal{T} \models D \sqsubseteq C$. Then any homomorphism $h : T_C \to \mathcal{I}_{D,\mathcal{T}}$ that maps ρ_C to ρ_D is an isomorphic embedding.*

Proof Assume first that h is not injective. Then there is a parent/child or sibling merging C' of C and a homomorphism $f : T_{C'} \to \mathcal{I}_{D,\mathcal{T}}$ such that $h = f \circ g$ for the natural homomorphism $g : T_C \to T_{C'}$ (Figure 10). By Lemma 12, $\mathcal{T} \models D \sqsubseteq C'$. Thus, $\mathcal{T} \models A \sqsubseteq C'$ and we have derived a contradiction to the assumption that C is parent/child and sibling merged.

Now let T' be the following labelled tree: the nodes in T' are the same as in T, $A \in l'(d)$ iff $h(d) \in A^{\mathcal{I}_{D,\mathcal{T}}}$, and for any two nodes d, d' with d' a successor of d: $l'(d, d') = r$ for the unique role r with $\mathcal{T} \models r \sqsubseteq s$ for all s with $(d, d') \in s^{\mathcal{I}_{D,\mathcal{T}}}$. Let C' be the concept expression that corresponds to T'. Then $\rho_D \in C'^{\mathcal{I}_{D,\mathcal{T}}}$ and so, by Lemma 12, $\mathcal{T} \models D \sqsubseteq C'$. Thus $\mathcal{T} \models A \sqsubseteq C'$. But then $A \sqsubseteq C'$ can be obtained from $A \sqsubseteq C$ by concept and role saturation steps. As $A \sqsubseteq C$ is concept and role saturated already, $C = C'$ and so h is an isomorphic embedding. ❑

We are now able to prove that $\mathcal{T}$-essential CIs are of polynomial size in $\mathcal{T}$. For a concept C, let n_C denote the number of nodes in the tree representation T_C of C and let

$$A^{\mathcal{T}} = \{A\} \cup \{B \mid A \sqsubseteq B \in \mathcal{H}_{basic}\} \cup \{D \mid \mathcal{T} \models A \sqsubseteq B, B \sqsubseteq D \in \mathcal{T}\}.$$

Lemma 32 *If $A \sqsubseteq C$ is $\mathcal{T}$-essential, then $n_C \leq \sum_{D \in A^{\mathcal{T}}} n_D$.*

Proof Assume $A \sqsubseteq C$ is $\mathcal{T}$-essential. Let $D_0 := \bigsqcap_{D \in A^{\mathcal{T}}} D$ and $\mathcal{I}_{D_0,\mathcal{T}}$ be the canonical model of D_0 and $\mathcal{T}$. By Lemma 12, there is a homomorphism $h : T_C \to \mathcal{I}_{D_0,\mathcal{T}}$ mapping ρ_C to ρ_{D_0}. By Lemma 31, h is an isomorphic embedding. Using that $A \sqsubseteq C$ is decomposed for $\mathcal{T}$, we now show that h maps T_C into the restriction of $\mathcal{I}_{D_0,\mathcal{T}}$ to $\Delta^{\mathcal{I}_{D_0}}$ from which the lemma follows since h is injective.

For a proof by contradiction, assume that there exists d' in T_C with $h(d') \notin \Delta^{\mathcal{I}_{D_0}}$. As $h(\rho_C) \in \Delta^{\mathcal{I}_{D_0}}$, we may assume that all d' on the path from ρ_C to d' are mapped to $\Delta^{\mathcal{I}_{D_0}}$ (if this is not the case, we can replace d' by the first element on the path from ρ_C to d' not mapped into $\Delta^{\mathcal{I}_{D_0}}$). In particular, the parent d of d' in T_C is mapped into $\Delta^{\mathcal{I}_{D_0}}$. Let $l(d, d') = r$. Observe that the whole subtree rooted in d' must be mapped into $\Delta^{\mathcal{I}_{D_0,\mathcal{T}}} \setminus \Delta^{\mathcal{I}_{D_0}}$ since otherwise h would not be injective.

Let $C' = \exists r.C''$, where C'' corresponds to the subtree rooted in d' in C. By Lemma 14, there exists a basic concept B such that $h(d) \in B^{\mathcal{I}_{D_0,\mathcal{T}}}$ and $\mathcal{T} \models B \sqsubseteq C'$. As $\mathcal{T}$ is in named form there exists a concept name E with $\mathcal{T} \models E \equiv B$. Thus, $h(d) \in E^{\mathcal{I}_{D_0,\mathcal{T}}}$ and $\mathcal{T} \models E \sqsubseteq C'$. As h is an isomorphic embedding, $E \in l(d)$. We make a case distinction:

- $h(d) \neq \rho_{D_0}$. Then $A \sqsubseteq C$ is not decomposed for $\mathcal{T}$ since C contains an edge (d, d') such that E is in the node label of d, $l(d, d') = r$, and $\mathcal{T} \models E \sqsubseteq \exists r.C''$. We have derived a contradiction.

- $h(d) = \rho_{D_0}$. As $h(d') \in \Delta^{\mathcal{I}_{D_0,\mathcal{T}}} \setminus \Delta^{\mathcal{I}_{D_0}}$, by construction of $\mathcal{I}_{D_0,\mathcal{T}}$, there exists a CI $B_0 \sqsubseteq D \in \mathcal{T}$ with B_0 a basic concept such that $\rho_{D_0} \in B_0^{\mathcal{I}_{D_0,\mathcal{T}}}$ and $h(d')$ is in the copy of the tree-shaped interpretation $\mathcal{I}_D$ which was attached to ρ_{D_0} in the construction of $\mathcal{I}_{D_0,\mathcal{T}}$. But since $\mathcal{T} \models A \sqsubseteq B_0$ we have $D \in A^{\mathcal{T}}$ and so $\rho_{D_0} \in D^{\mathcal{I}_{D_0}}$. But then, by the construction of $\mathcal{I}_{D_0,\mathcal{T}}$, no fresh $\mathcal{I}_D$ was attached to ρ_{D_0} because D is already satisfied in $\mathcal{I}_{D_0}$ and we have derived a contradiction.

❑

We are now in the position to prove that the learning algorithm terminates after posing a polynomial number of queries.

Lemma 33 *For every concept name A, the number of replacements of a CI $A \sqsubseteq C$ in $\mathcal{H}_{add}$ by a CI of the form $A \sqsubseteq C'$ is bounded polynomially in $|\mathcal{T}|$.*

Proof We aim to show that when $A \sqsubseteq C$ is replaced with $A \sqsubseteq C'$, then the number of nodes in the tree representation of C' is strictly larger than the number of nodes in the tree representation of C. Since any CI $A \sqsubseteq C$ ever added to $\mathcal{H}_{add}$ is $\mathcal{T}$-essential, Lemma 32 then yields that the number of replacements is bounded by $\sum_{D \in A^{\mathcal{T}}} n_D$, which is polynomial in $|\mathcal{T}|$.

A straightforward analysis of Algorithm 2 reveals that when $A \sqsubseteq C$ is replaced with $A \sqsubseteq C'$, then $\emptyset \models C' \sqsubseteq C$ and $\emptyset \not\models C \sqsubseteq C'$ (otherwise the positive counterexample returned by the oracle would be a consequence of $\mathcal{H}_{basic} \cup \mathcal{H}_{add}$). Moreover, both $A \sqsubseteq C$ and $A \sqsubseteq C'$ are consequences of $\mathcal{T}$. It thus suffices to establish the following.

Claim. If $A \sqsubseteq C$ is $\mathcal{T}$-essential, $\mathcal{T} \models A \sqsubseteq C'$, $\emptyset \models C' \sqsubseteq C$, and $\emptyset \not\models C \sqsubseteq C'$, then T_C is obtained from $T_{C'}$ by removing at least one subtree.

We prove the claim. Since $\emptyset \models C' \sqsubseteq C$, by Lemma 12 there is a homomorphism h from T_C to the canonical model $\mathcal{I}_{C'}$ that maps ρ_C to $\rho_{C'}$. Then h is also a homomorphism into the canonical model $\mathcal{I}_{C',\mathcal{T}}$. Since $A \sqsubseteq C$ is $\mathcal{T}$-essential, $\mathcal{T} \models A \sqsubseteq C'$, and $\emptyset \models C' \sqsubseteq C$, Lemma 31 yields that h is an isomorphic embedding into $\mathcal{I}_{C',\mathcal{T}}$. Then, trivially, h is also an isomorphic embedding into $\mathcal{I}_{C'}$ which means that T_C is obtained from $T_{C'}$ by removing subtrees. Since $\emptyset \not\models C \sqsubseteq C'$, at least one subtree must in fact have been removed. ❑

We have obtained the following main result of this section.

Theorem 34 *DL-Lite$^{\exists}_{\mathcal{R}}$ TBoxes are polynomial query learnable using membership and equivalence queries. Moreover, DL-Lite$^{\exists}_{\mathcal{R}}$ TBoxes without inverse roles can be learned in polynomial time using membership and equivalence queries.*

Proof Recall that our algorithm requires the target TBox to be in named form. We first show Theorem 34 under that assumption and then argue that the assumption can be dropped.

In each iteration of Algorithm 2, either a CI is added to $\mathcal{H}_{add}$ or a CI is replaced in $\mathcal{H}_{add}$. Since the number of times the former happens is bounded by $|\Sigma_{\mathcal{T}}|$ and (by Lemma 33) the number of times the latter happens is polynomial in $|\mathcal{T}|$, the number of iterations of Algorithm 2 is polynomial in $|\mathcal{T}|$. For polynomial query learnability of DL-Lite$^{\exists}_{\mathcal{R}}$ TBoxes, it remains to show that in each iteration Algorithm 2 makes only polynomially many polynomial size queries in $|\mathcal{T}|$ and the size of the largest counterexample seen so far. We start with equivalence queries, made only in Line 3. We have already argued that the number of iterations is polynomial in $|\mathcal{T}|$ and thus so is the number of equivalence queries made. Regarding their size, we observe that there are at most $|\Sigma_{\mathcal{T}}|^2$ CIs in $\mathcal{H}_{basic}$ and at most $|\Sigma_{\mathcal{T}}|$ CIs in $\mathcal{H}_{add}$, that the size of CIs in $\mathcal{H}_{basic}$ is constant and by Lemma 32 the size of CIs in $\mathcal{H}_{add}$ is polynomial in $|\mathcal{T}|$. Membership queries are made only in Lines 5 and 7 for which it suffices to invoke Lemmas 29 and 30.

Now for the "moreover" part of Theorem 34. Observe that since each (membership or equivalence) query counts as one step of computation, the only potentially costly step of Algorithm 2 is the implementation of the decomposition step in Line 5, which relies on making subsumption checks of the form $\mathcal{H}_{basic} \cup \mathcal{H}_{add} \models A \sqsubseteq C$. As discussed in Section 2, deciding subsumption in DL-Lite$^{\exists}_{\mathcal{R}}$ is NP-complete while in $\mathcal{EL}$ with role inclusions subsumption is in PTime. As DL-Lite$^{\exists}_{\mathcal{R}}$ without inverse roles is a fragment of $\mathcal{EL}$ with role inclusions, we obtain polynomial time learnability for TBoxes in this case.

To drop the requirement that the target TBox is in named form, we show that any polynomial (query or time) learning algorithm for TBoxes in named form can be transformed into the same kind of algorithm for unrestricted target TBoxes. In fact, the learner can use at most $\mathcal{O}(|\Sigma_{\mathcal{T}}|^2)$ membership queries "Does $\mathcal{T}$ entail $r \sqsubseteq s$?" to compute for every role r the class $[r]_{\mathcal{T}}$ of roles s with $\mathcal{T} \models s \equiv r$ and choose a representative $r_{\mathcal{T}}$ for this class. Then whenever some $s \in [r]_{\mathcal{T}}$ is used in any counterexample returned by the oracle, it gets replaced with $r_{\mathcal{T}}$. Likewise, whenever $\mathcal{T}$ does not have a name for some $\exists r.\top$, the algorithm still uses the concept name A_r in its internal representations (although they are no longer included in the signature $\Sigma_{\mathcal{T}}$ of the target TBox) and replaces $\exists r.\top$ with A_r in the counterexamples returned by the oracle. It also replaces each A_r with $\exists r.\top$ in membership queries to the oracle and in the hypothesis used for posing equivalence queries. ❑

4. Learning DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ TBoxes

We study exact learnability of TBoxes in DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$, the extension of DL-Lite$^{\exists}_{\mathcal{R}}$ that admits conjunctions of basic concepts on the left-hand side of CIs. This language is a generalisation of both DL-Lite$^{\exists}_{\mathcal{R}}$ and propositional Horn logic. In fact, the algorithm we present combines the classical algorithms for propositional Horn logic (Angluin et al., 1992; Frazier and Pitt, 1993) with the algorithm for DL-Lite$^{\exists}_{\mathcal{R}}$ presented in Section 3. The resulting algorithm is quite subtle and indeed this is the reason why we treated the DL-Lite$^{\exists}_{\mathcal{R}}$ case separately in Section 3.

To simplify the presentation, we make the same assumptions as in Section 3 about the target TBox $\mathcal{T}$ with signature $\Sigma_{\mathcal{T}}$. In particular, we assume that $\mathcal{T}$ is in named form, suitably generalised to DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$: there are no distinct roles r and s such that $\mathcal{T} \models r \equiv s$, for each role r the TBox $\mathcal{T}$ contains an equivalence $A_r \equiv \exists r.\top$ and all CIs of $\mathcal{T}$ are either CIs between basic concepts or contain no concept expressions of the form $\exists r.\top$ on the left-hand side, for any role r. Denote by $\mathsf{lhs}(\alpha)$ the set of concept names that occur as conjuncts on the left-hand side of a CI α and denote by $\mathsf{rhs}(\alpha)$ the set of concept expressions that occur as top-level conjuncts on the right-hand side of α (that is, they are not nested inside restrictions). We often do not distinguish between the set $\mathsf{lhs}(\alpha)$ and the conjunction over all its concept expressions, and similarly for $\mathsf{rhs}(\alpha)$. For example, if $\alpha_1 = C_1 \sqsubseteq D_1$ and $\alpha_2 = C_2 \sqsubseteq D_2$ then $\mathsf{lhs}(\alpha_1) \sqsubseteq \mathsf{rhs}(\alpha_2)$ stands for $C_1 \sqsubseteq D_2$. Also, if $\mathsf{lhs}(\alpha_1) = \{A_1, A_2, A_3\}$ and $\mathsf{lhs}(\alpha_2) = \{A_2, A_3, A_4\}$ then $\mathsf{lhs}(\alpha_1) \cap \mathsf{lhs}(\alpha_2) \sqsubseteq D$ stands for $A_2 \sqcap A_3 \sqsubseteq D$.

The algorithm for learning DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ TBoxes is shown as Algorithm 3. Like Algorithm 2, Algorithm 3 first determines the set $\mathcal{H}_{basic}$ that contains all CIs $B_1 \sqsubseteq B_2$ with B_1, B_2 basic concepts such that $\mathcal{T} \models B_1 \sqsubseteq B_2$ and all RIs $r \sqsubseteq s$ such that $\mathcal{T} \models r \sqsubseteq s$. The hypothesis $\mathcal{H}$ is the union of $\mathcal{H}_{basic}$ and $\mathcal{H}_{add}$. In contrast to Algorithm 2, $\mathcal{H}_{add}$ is an *ordered list* of CIs rather than a set. We write α_i to denote the CI α at position i in the list $\mathcal{H}_{add}$. In

Algorithm 3 The learning algorithm for DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ TBoxes

Input: A DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ TBox $\mathcal{T}$ in named form given to the oracle; $\Sigma_{\mathcal{T}}$ given to the learner.
Output: TBox $\mathcal{H}$, computed by the learner, such that $\mathcal{T} \equiv \mathcal{H}$.

1: Compute $\mathcal{H}_{basic} = \{r \sqsubseteq s \mid \mathcal{T} \models r \sqsubseteq s\} \cup \{B_1 \sqsubseteq B_2 \mid \mathcal{T} \models B_1 \sqsubseteq B_2,\ B_1, B_2 \text{ basic}\}$
2: Set $\mathcal{H}_{add}$ to be the empty list and $\mathcal{H} = \mathcal{H}_{basic} \cup \mathcal{H}_{add}$
3: **while** $\mathcal{H} \not\equiv \mathcal{T}$ **do**
4: Let γ be the returned positive counterexample for $\mathcal{T}$ and $\mathcal{H}$
5: Find a $\mathcal{T}$-essential γ' with $\mathcal{H} \not\models \gamma'$ and $|\{\exists r.F \mid \exists r.F \in \mathsf{rhs}(\gamma')\}| \leq 1$
6: Left saturate γ' for $\mathcal{H}$
7: **if** there is $A \in \mathsf{N_C}$ such that $\mathcal{T} \models \mathsf{lhs}(\gamma') \sqsubseteq A$ and $\mathcal{H} \not\models \mathsf{lhs}(\gamma') \sqsubseteq A$ **then**
8: $\mathcal{H}$:=CN-Refine$(\mathcal{H}, \mathsf{lhs}(\gamma') \sqsubseteq A)$
9: **else**
10: $\mathcal{H}$:=$\exists$-Refine$(\mathcal{H}, \gamma')$
11: **end if**
12: Set $\mathcal{H} = \mathcal{H}_{basic} \cup \mathcal{H}_{add}$
13: **end while**
14: **return** $\mathcal{H}$

the learning algorithm, working with an ordered list of CIs allows the learner to pick the first α_i in $\mathcal{H}_{add}$ with a certain property and merge it with a new CI, a technique we adopt from the work of Angluin et al. (1992) and Frazier and Pitt (1993). As in Algorithm 2, $\mathcal{T} \models \mathcal{H}$ is a loop invariant, thus, γ is necessarily positive. The algorithm terminates when $\mathcal{H} \equiv \mathcal{T}$.

Algorithm 4 Function CN-Refine$(\mathcal{H}, \gamma)$

1: **if** there is $A \in \mathsf{N_C}$ and $\alpha_i \in \mathcal{H}_{add}$ such that $\mathcal{T} \models \mathsf{lhs}(\alpha_i) \cap \mathsf{lhs}(\gamma) \sqsubseteq A$,
2: and $\mathcal{H} \not\models \mathsf{lhs}(\alpha_i) \cap \mathsf{lhs}(\gamma) \sqsubseteq A$ **then**
3: Concept saturate $\gamma' = \mathsf{lhs}(\alpha_i) \cap \mathsf{lhs}(\gamma) \sqsubseteq A$ for $\mathcal{T}$
4: Replace the first such α_i in $\mathcal{H}_{add}$ by γ'
5: **else**
6: Concept saturate γ for $\mathcal{T}$
7: Append γ to the list $\mathcal{H}_{add}$
8: **end if**
9: **return** $\mathcal{H}$

Algorithm 3 uses membership queries to compute a $\mathcal{T}$-essential counterexample γ such that $\mathsf{rhs}(\gamma)$ contains at most one concept expression of the form $\exists r.F$ (Line 5) and which is 'Left saturated for $\mathcal{H}$' (Line 6); here, a CI is left saturated for $\mathcal{H}$ if its left-hand side contains all subsuming concept names w.r.t. $\mathcal{H}$ (Definition 35 below) and $\mathcal{T}$-essential if it satisfies the conditions for $\mathcal{T}$-essential CIs from Section 3, appropriately modified for CIs with conjunctions of concept names on the left-hand side (Definition 37 below). Then, the algorithm checks whether there is a concept name A such that $\mathsf{lhs}(\gamma) \sqsubseteq A$ is a positive counterexample. If so, then it calls Function CN-Refine (Algorithm 4) and updates the hypothesis either by refining some α_i in $\mathcal{H}_{add}$ or by appending a new CI to $\mathcal{H}_{add}$. The

Algorithm 5 Function $\exists$-Refine$(\mathcal{H}, \gamma)$

1: **if** there is $C \in \mathsf{rhs}(\gamma)$ of the form $\exists r.D$ and $\alpha_i \in \mathcal{H}_{add}$ such that $\mathcal{T} \models \mathsf{lhs}(\alpha_i) \cap \mathsf{lhs}(\gamma) \sqsubseteq C$
2: and $\mathcal{H} \not\models \mathsf{lhs}(\alpha_i) \cap \mathsf{lhs}(\gamma) \sqsubseteq C$ **then**
3: **if** $\mathcal{T} \models \mathsf{lhs}(\alpha_i) \cap \mathsf{lhs}(\gamma) \sqsubseteq C \sqcap \mathsf{rhs}(\alpha_i)$ **then**
4: Find a $\mathcal{T}$-essential $\mathsf{lhs}(\alpha_i) \cap \mathsf{lhs}(\gamma) \sqsubseteq D^*$ with $\emptyset \models D^* \sqsubseteq C \sqcap \mathsf{rhs}(\alpha_i)$
5: Replace the first such α_i in $\mathcal{H}_{add}$ by $\mathsf{lhs}(\alpha_i) \cap \mathsf{lhs}(\gamma) \sqsubseteq D^*$
6: **else**
7: Concept saturate $\gamma' = \mathsf{lhs}(\alpha_i) \cap \mathsf{lhs}(\gamma) \sqsubseteq C$ for $\mathcal{T}$
8: Replace the first such α_i in $\mathcal{H}_{add}$ by γ'
9: **end if**
10: **else**
11: Append γ to the list $\mathcal{H}_{add}$
12: **end if**
13: **return** $\mathcal{H}$

number of replacements of any given α_i in $\mathcal{H}_{add}$ in CN-Refine is bounded by $|\Sigma_{\mathcal{T}}|$ since whenever α_i is replaced in CN-Refine$(\mathcal{H}, \gamma)$, then $\mathsf{lhs}(\alpha_i) \cap \mathsf{lhs}(\gamma) \subsetneq \mathsf{lhs}(\alpha_i)$.[4]

If there is no concept name A such that $\mathsf{lhs}(\gamma) \sqsubseteq A$ is a positive counterexample then Algorithm 3 calls Function $\exists$-Refine (Algorithm 5). In this case one considers the existential restrictions that occur on top-level on the right-hand side of γ. Note that $\exists$-Refine can be viewed as a variation of the body of the while loop in Algorithm 2 in which one considers sets of concept names on the left-hand side of CIs rather than a single concept name. Recall that in Algorithm 2, the new CI γ and a CI α in $\mathcal{H}_{add}$ are merged if they have the same concept name on the left-hand side. In contrast, now they are merged if the intersection of their left-sides is still subsumed by some existential restriction C from $\mathsf{rhs}(\gamma)$ (Lines 1 and 2). There are two cases: if the intersection is also subsumed by $\mathsf{rhs}(\alpha)$ (checked in Line 3), then in the next line a $\mathcal{T}$-essential counterexample is computed and the first such α_i is replaced by the new CI. Otherwise it follows that $\mathsf{lhs}(\alpha_i) \cap \mathsf{lhs}(\gamma) \subsetneq \mathsf{lhs}(\alpha_i)$ and the first such α_i is replaced by the CI computed in Line 7. Note that the latter can happen at most $|\Sigma_{\mathcal{T}}|$ times for each CI in $\mathcal{H}_{add}$ (and the former can happen at most $|\mathcal{T}|$ times for each CI in $\mathcal{H}_{add}$, see Lemma 43 below). If no CI can be refined with γ then $\exists$-Refine appends γ to $\mathcal{H}_{add}$.

We now define the step 'left-saturate γ for $\mathcal{H}$' used in Line 6 of Algorithm 3. Observe that this step is meaningless for DL-Lite$^{\exists}_{\mathcal{R}}$.

Definition 35 (Left saturation for $\mathcal{H}$) *A CI γ' is obtained from a CI γ by* left saturation for $\mathcal{H}$ *if* $\mathsf{rhs}(\gamma') = \mathsf{rhs}(\gamma)$ *and* $\mathsf{lhs}(\gamma') = \{A \in \Sigma_{\mathcal{T}} \mid \mathcal{H} \models \mathsf{lhs}(\gamma) \sqsubseteq A\}$. *A CI γ is* left saturated for $\mathcal{H}$ *if it coincides with its left saturation for $\mathcal{H}$.*

One can clearly left saturate any CI γ for $\mathcal{H}$ by checking whether $\mathcal{H} \models \mathsf{lhs}(\gamma) \models A$ for every $A \in \Sigma_{\mathcal{T}}$. The following example shows that Line 6 is necessary for Algorithm 3 to be polynomial. A similar step is also necessary in Frazier et al.'s algorithm learning propositional Horn logic from entailments (Frazier and Pitt, 1993).

4. This is a consequence of the fact that $\mathcal{H}_{add}$ only contains concept saturated CIs (defined essentially as in the previous section, see Definition 37 below): $\mathsf{lhs}(\alpha_i) \cap \mathsf{lhs}(\gamma) = \mathsf{lhs}(\alpha_i)$ and $\mathcal{T} \models \mathsf{lhs}(\alpha_i) \cap \mathsf{lhs}(\gamma) \sqsubseteq A$ implies $A \in \mathsf{rhs}(\alpha_i)$ by concept saturatedness, thus contradicting $\mathcal{H} \not\models \mathsf{lhs}(\alpha_i) \cap \mathsf{lhs}(\gamma) \sqsubseteq A$.

Example 36 Assume Line 6 of Algorithm 3 is omitted. Let for $n \geq 2$,

$$\mathcal{T}_n \quad = \quad \{E_1 \sqcap \cdots \sqcap E_n \sqsubseteq A\} \cup \{A_i \sqsubseteq E_i, B_i \sqsubseteq E_i \mid 1 \leq i \leq n\}.$$

For $M \subseteq \{1, \ldots, n\}$, set $C_M = \bigsqcap_{i \leq n} C_i$, where $C_i = A_i$ if $i \in M$ and $C_i = B_i$ if $i \notin M$. Then the oracle can provide for the first 2^n equivalence queries in the while loop of Algorithm 3 a positive counterexample $C_M \sqsubseteq A$ by always choosing a fresh set $M \subseteq \{1, \ldots, n\}$.

For the refinements on the right-hand side, we extend to DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ the notion of $\mathcal{T}$-essential CIs introduced in the previous section:

1. (Concept saturation for $\mathcal{T}$) A CI γ is *concept saturated for* $\mathcal{T}$ if $\mathcal{T} \models \gamma$ and $\mathcal{T} \not\models \gamma'$ for any γ' with $\mathsf{lhs}(\gamma) = \mathsf{lhs}(\gamma')$ such that $\mathsf{rhs}(\gamma')$ is obtained from $\mathsf{rhs}(\gamma)$ by adding a concept name to the label of some node of $\mathsf{rhs}(\gamma)$. A CI γ' is a *concept saturation for* $\mathcal{T}$ of a CI γ if it is concept saturated for $\mathcal{T}$, $\mathsf{lhs}(\gamma) = \mathsf{lhs}(\gamma')$, and $\mathsf{rhs}(\gamma')$ is obtained from $\mathsf{rhs}(\gamma)$ by adding concept names to the labels of some nodes of $\mathsf{rhs}(\gamma)$.

2. (Role saturation for $\mathcal{T}$) A CI γ is *role saturated for* $\mathcal{T}$ if $\mathcal{T} \models \gamma$ and $\mathcal{T} \not\models \gamma'$ for any γ' with $\mathsf{lhs}(\gamma) = \mathsf{lhs}(\gamma')$ such that $\mathsf{rhs}(\gamma')$ is obtained from $\mathsf{rhs}(\gamma)$ by replacing in some edge label a role r by a role $s \neq r$ with $\mathcal{T} \models s \sqsubseteq r$. A CI γ' is a *role saturation for* $\mathcal{T}$ of a CI γ if it is role saturated for $\mathcal{T}$, $\mathsf{lhs}(\gamma) = \mathsf{lhs}(\gamma')$, and $\mathsf{rhs}(\gamma')$ is obtained from $\mathsf{rhs}(\gamma)$ by replacing in some edge labels a role r by a role $s \neq r$ with $\mathcal{T} \models s \sqsubseteq r$.

3. (Parent/child merged for $\mathcal{T}$) A CI γ' is *obtained from a CI* γ *by parent/child merging for* $\mathcal{T}$ if $\mathsf{lhs}(\gamma) = \mathsf{lhs}(\gamma')$, $\mathsf{rhs}(\gamma')$ is obtained from $\mathsf{rhs}(\gamma')$ by parent/child merging (as in Definition 23), and $\mathcal{T} \models \gamma'$. A CI γ is *parent/child merged for* $\mathcal{T}$ if $\mathcal{T} \models \gamma$ and there is no γ' with $\gamma \neq \gamma'$ that can be obtained from γ by parent/child merging for $\mathcal{T}$.

4. (Sibling merged for $\mathcal{T}$) A CI γ' is *obtained from a CI* γ *by sibling merging for* $\mathcal{T}$ if $\mathsf{lhs}(\gamma) = \mathsf{lhs}(\gamma')$, $\mathsf{rhs}(\gamma')$ is obtained from $\mathsf{rhs}(\gamma')$ by sibling merging (as in Definition 25), and $\mathcal{T} \models \gamma'$. A CI γ is *sibling merged for* $\mathcal{T}$ if $\mathcal{T} \models \gamma$ and there is no γ' with $\gamma \neq \gamma'$ that can be obtained from γ by sibling merging for $\mathcal{T}$.

5. (Decomposed CI for $\mathcal{T}$) A CI γ is *decomposed for* $\mathcal{T}$ if $\mathcal{T} \models \gamma$ and for every non-root node d in $\mathsf{rhs}(\gamma)$, every role r, and every r-successor d' of d in $\mathsf{rhs}(\gamma)$ we have $\mathcal{T} \not\models l(d) \sqsubseteq \exists r.C'$, where C' corresponds to the subtree of $\mathsf{rhs}(\gamma)$ rooted at d'.

Definition 37 *A DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ CI is* $\mathcal{T}$-essential *if it is concept saturated, role saturated, parent/child merged, sibling merged, and decomposed for* $\mathcal{T}$.

The saturation, merging and decomposition steps defined above are straightforward generalisations of Definitions 18 to 27 to CIs with conjunctions on the left-hand side. One can easily generalise the arguments from DL-Lite$^{\exists}_{\mathcal{R}}$ to show that for any CI γ with $\mathcal{H} \not\models \gamma$ one can compute a $\mathcal{T}$-essential γ' with $\mathcal{H} \not\models \gamma'$ using polynomially many membership queries and entailment checks relative to $\mathcal{H}$. For the analysis of the learning algorithm it is crucial that all CIs in the ordered list $\mathcal{H}_{add}$ are $\mathcal{T}$-essential at all times, which we prove next.

Lemma 38 *At any point in the execution of Algorithm 3, all CIs in* $\mathcal{H}_{add}$ *are* $\mathcal{T}$*-essential.*

Proof If a CI γ is of the form $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq A$ with A a concept name, then the set $\mathsf{rhs}(\gamma')$ of the concept saturation γ' of γ for $\mathcal{T}$ contains concept names only. Thus, γ' is $\mathcal{T}$-essential. It follows that the CIs added to $\mathcal{H}_{add}$ in Lines 4 and 7 of CN-Refine are $\mathcal{T}$-essential. Also, it is easy to see that if γ is $\mathcal{T}$-essential and $C \in \mathsf{rhs}(\gamma)$ then the concept saturation of $\mathsf{lhs}(\gamma) \sqsubseteq C$ for $\mathcal{T}$ is $\mathcal{T}$-essential as well. Thus, the CI γ' in Line 8 of $\exists$-Refine is $\mathcal{T}$-essential. ❑

Polynomial Query Bound on the Algorithm

As in the previous section it is immediate that upon termination the algorithm has found a TBox $\mathcal{H} = \mathcal{H}_{basic} \cup \mathcal{H}_{add}$ that is logically equivalent to the target TBox $\mathcal{T}$. It thus remains to show that it issues only polynomially many queries of polynomial size. We first discuss how Lines 5 and 6 of Algorithm 3 and Line 4 of $\exists$-Refine can be implemented. The next lemma addresses Lines 5 and 6 of Algorithm 3.

Lemma 39 *Given a positive counterexample* γ *for* $\mathcal{T}$ *relative to* $\mathcal{H}$*, one can construct with polynomially many polynomial size membership queries in* $|\gamma|$ *and* $|\mathcal{T}|$*, a counterexample* γ' *that is left saturated for* $\mathcal{H}$*,* $\mathcal{T}$*-essential and such that* $|\{\exists r.F \mid \exists r.F \in \mathsf{rhs}(\gamma')\}| \leq 1$.

The proof of Lemma 39 is a straightforward extension of the proof of Lemma 29 and uses the observation that a left-saturated γ' for $\mathcal{H}$ can be computed from γ by adding all concept names $A \in \Sigma_{\mathcal{T}}$ with $\mathcal{H} \models \mathsf{lhs}(\gamma) \sqsubseteq A$ to $\mathsf{lhs}(\gamma)$. This lemma also requires that $|\{\exists r.F \mid \exists r.F \in \mathsf{rhs}(\gamma')\}| \leq 1$. If there is $A \in \mathsf{rhs}(\gamma')$ such that $\mathcal{H} \not\models \mathsf{lhs}(\gamma') \sqsubseteq A$ then we can simply drop all conjuncts of the form $\exists r.F$ from $\mathsf{rhs}(\gamma')$. Otherwise, we can satisfy the condition by simply choosing a conjunct $\exists r.F \in \mathsf{rhs}(\gamma')$ such that $\mathcal{H} \not\models \mathsf{lhs}(\gamma') \sqsubseteq \exists r.F$ and then apply 'Concept saturation for $\mathcal{T}$' to $\mathsf{lhs}(\gamma') \sqsubseteq \exists r.F$. The resulting γ' is left saturated for $\mathcal{H}$, $\mathcal{T}$-essential and has at most one conjunct of the form $\exists r.F$ in $\mathsf{rhs}(\gamma')$.

The following lemma addresses Line 4 of $\exists$-Refine.

Lemma 40 *Assume that* α *and* γ *are* $\mathcal{T}$*-essential and there is* $C \in \mathsf{rhs}(\gamma)$ *such that* $\mathcal{T} \models \mathsf{lhs}(\alpha) \sqcap \mathsf{lhs}(\gamma) \sqsubseteq \mathsf{rhs}(\alpha) \sqcap C$*. Then one can construct, with polynomially many polynomial size membership queries in* $|\mathsf{rhs}(\alpha)|$ *and* $|C|$*, a* $\mathcal{T}$*-essential* $\mathsf{lhs}(\alpha) \sqcap \mathsf{lhs}(\gamma) \sqsubseteq D^*$ *such that* $\emptyset \models D^* \sqsubseteq \mathsf{rhs}(\alpha) \sqcap C$.

Proof Assume $\mathcal{T} \models \mathsf{lhs}(\alpha) \sqcap \mathsf{lhs}(\gamma) \sqsubseteq \mathsf{rhs}(\alpha) \sqcap C$. Then, similar to Lemma 30, one can show that the only property of $\mathcal{T}$-essential CIs that can fail is being sibling merged for $\mathcal{T}$ and that after applying the step 'Sibling merging for $\mathcal{T}$' to $\mathsf{lhs}(\alpha) \sqcap \mathsf{lhs}(\gamma) \sqsubseteq \mathsf{rhs}(\alpha) \sqcap C$ the resulting CI is $\mathcal{T}$-essential, as required. ❑

We also have to show that the number of CIs in $\mathcal{H}_{add}$ is bounded polynomially in $|\mathcal{T}|$ and for each position of $\mathcal{H}_{add}$ the number of replacements is bounded polynomially in $|\mathcal{T}|$. These properties follow from the following lemma.

Lemma 41 *Let* $\mathcal{H}_{add}$ *be a ordered list of CIs computed at some point of an execution of Algorithm 3. Then*

(i) the length of $\mathcal{H}_{add}$ *is bounded by the number of CIs in* $\mathcal{T}$ *and*

(ii) The number of replacements of an existing CI $\alpha \in \mathcal{H}_{add}$ is bounded polynomially in $|\mathcal{T}|$.

The rest of the section is devoted to proving Lemma 41. We first show Point (ii) of Lemma 41 and start by generalising Lemma 32 on the size of $\mathcal{T}$-essentials CIs. For any conjunction C of concept names we set

$$\begin{aligned} C^{\mathcal{T}} \;=\; & \{D \mid \mathcal{T} \models C \sqsubseteq A_1 \sqcap \cdots \sqcap A_k \text{ and } A_1 \sqcap \cdots \sqcap A_k \sqsubseteq D \in \mathcal{T}\} \cup \\ & \{B \mid \mathcal{T} \models C \sqsubseteq B, B \text{ basic concept over } \Sigma_{\mathcal{T}}\} \end{aligned}$$

Recall that for any concept expression C we denote by n_C the number of nodes in the tree T_C corresponding to C.

Lemma 42 *If α is $\mathcal{T}$-essential, then $n_{\mathsf{rhs}(\alpha)} \leq \sum_{D \in \mathsf{lhs}(\alpha)^{\mathcal{T}}} n_D$.*

Proof The proof is almost the same as the proof of Lemma 32. Assume α is $\mathcal{T}$-essential. Let $D_0 := \bigsqcap_{D \in \mathsf{lhs}(\alpha)^{\mathcal{T}}} D$ and let $\mathcal{I}_{D_0,\mathcal{T}}$ be the canonical model of D_0 and $\mathcal{T}$. Now one can prove in almost the same way as in the proof of Lemma 32 that the homomorphism h from $T_{\mathsf{rhs}(\alpha)}$ into $\mathcal{I}_{D_0,\mathcal{T}}$ mapping $\rho_{\mathsf{rhs}(\alpha)}$ to $\rho_{D_0,\mathcal{T}}$ is an injective mapping into $\mathcal{I}_{D_0}$ (using Lemma 14 for DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ instead of DL-Lite$^{\exists}_{\mathcal{R}}$). ❑

We are now in the position to prove Point (ii) of Lemma 41.

Lemma 43 *The number of replacements of an existing CI $\alpha \in \mathcal{H}_{add}$ is bounded polynomially in $|\mathcal{T}|$.*

Proof A CI $\alpha \in \mathcal{H}_{add}$ can be replaced in Line 4 of CN-Refine or in Lines 5 or 8 of $\exists$-Refine. If α is replaced by α' in Line 4 of CN-Refine or in Line 8 of $\exists$-Refine then $\mathsf{lhs}(\alpha') \subsetneq \mathsf{lhs}(\alpha)$, so the number of replacements is bounded by $|\Sigma_{\mathcal{T}}|$. If α is replaced by α' in Line 5 of $\exists$-Refine, then either $\mathsf{lhs}(\alpha') \subsetneq \mathsf{lhs}(\alpha)$ or $\mathsf{lhs}(\alpha') = \mathsf{lhs}(\alpha)$. For the latter case one can show as in the proof of Lemma 33 for DL-Lite$^{\exists}_{\mathcal{R}}$, the following

Claim. If $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq C$ and $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq C'$ are $\mathcal{T}$-essential, and $\emptyset \models C' \sqsubseteq C$, then T_C is obtained from $T_{C'}$ by removing subtrees.

Thus, each time $\alpha \in \mathcal{H}_{add}$ is replaced in Line 5 of $\exists$-Refine without decreasing the number of concept names in $\mathsf{lhs}(\alpha)$, the number $n_{\mathsf{rhs}(\alpha)}$ of nodes in the tree representation of $\mathsf{rhs}(\alpha)$ strictly increases. By Lemma 42, $n_{\mathsf{rhs}(\alpha)}$ is bounded polynomially in $|\mathcal{T}|$ and the lemma follows. ❑

We now come to the proof of Point (i) of Lemma 41. To formulate an upper bound on the length of $\mathcal{H}_{add}$ in terms of $\mathcal{T}$ it is convenient to assume that the right-hand side of every CI in $\mathcal{T}$ is *primitive*, that is, either a concept name or a concept expression of the form $\exists r.D$. This assumption is w.l.o.g. since one can equivalently transform every CI $C \sqsubseteq D_1 \sqcap D_2$ into two CIs $C \sqsubseteq D_1$ and $C \sqsubseteq D_2$. We call such a TBox *rhs-primitive*. Note that CIs in $\mathcal{H}$ may still have multiple concepts on the right-hand side.

A concept C is called *concept saturated for* $\mathcal{T}$ if $\mathcal{T} \models C \sqsubseteq C'$ whenever C' results from C by adding a new concept name A' to the label of some node in T_C. Denote by C^{sat} the (unique) concept obtained from C by adding concept names to the node labels of T_C until it is concept saturated for $\mathcal{T}$. The following definition enables us to link the CIs in $\mathcal{H}_{add}$ to the CIs in $\mathcal{T}$.

Definition 44 *Let $\mathcal{T}$ be rhs-primitive. We say that a CI α* has target $\beta \in \mathcal{T}$ *if*

1. $\mathsf{lhs}(\beta) \subseteq \mathsf{lhs}(\alpha)$ *and*
2. *there exists $D \in \mathsf{rhs}(\alpha) \setminus \mathsf{lhs}(\alpha)$ such that $\emptyset \models \mathsf{rhs}(\beta)^{\mathsf{sat}} \sqsubseteq D$.*

We aim to show that Algorithm 3 maintains the invariant that

(iii) every $\alpha \in \mathcal{H}_{add}$ has some target $\beta \in \mathcal{T}$ and

(iv) every $\beta \in \mathcal{T}$ is the target of at most one $\alpha \in \mathcal{H}_{add}$.

Then Point (i) of Lemma 41 clearly follows.

Example 45 To illustrate Definition 44, suppose that

$$\mathcal{T} = \{A_1 \sqcap A_4 \sqsubseteq A_2,\ A_2 \sqsubseteq \exists r.A_3,\ A_3 \sqsubseteq A_4,\ A_r \equiv \exists r.\top\}$$

is the target TBox. $\mathcal{T}$ is rhs-primitive. To simplify notation, we use β_i to denote the i-th CI occurring in $\mathcal{T}$ above. Assume $\mathcal{H}_{basic} = \{\beta_3, \beta_4\}$ and $\mathcal{H}_{add} = \emptyset$. Let $\alpha_1 = A_1 \sqcap A_3 \sqsubseteq A_2$. Then there is no $\beta_i \in \mathcal{T}$ such that α_1 has target β_i. However, by applying left saturation for $\mathcal{H}$ to α_1 we obtain $\alpha_1' = A_1 \sqcap A_3 \sqcap A_4 \sqsubseteq A_2$ and since $\mathsf{lhs}(\beta_1) \subseteq \mathsf{lhs}(\alpha_1')$ and $A_2 \notin \mathsf{lhs}(\alpha_1')$, α_1' has target β_1. For $\alpha_2 = A_1 \sqcap A_4 \sqsubseteq \exists r.A_3$, there is no $\beta_i \in \mathcal{T}$ such that α_2 has target β_i. But α_2 is not $\mathcal{T}$-essential and making it $\mathcal{T}$-essential results in $\alpha_2' = A_1 \sqcap A_4 \sqsubseteq A_r \sqcap A_1 \sqcap A_2 \sqcap A_4 \sqcap \exists r.(A_3 \sqcap A_4)$ which again has target β_1. Finally, let $\alpha_3 = A_2 \sqsubseteq \exists r.A_4$. As $\mathsf{lhs}(\beta_2) \subseteq \mathsf{lhs}(\alpha_3)$ and $\emptyset \models A_r \sqcap \exists r.(A_3 \sqcap A_4) \sqsubseteq \exists r.A_4$, α_3 has target β_2. Note that α_3 is not $\mathcal{T}$-essential, but the result of making it $\mathcal{T}$-essential also has target β_2.

Point (iii) is a consequence of the following lemma.

Lemma 46 *Let $\mathcal{T}$ be rhs-primitive and let γ be a $\mathcal{T}$-essential CI such that $\emptyset \not\models \gamma$. Then γ has some target $\beta \in \mathcal{T}$.*

Proof Assume γ is $\mathcal{T}$-essential and $\emptyset \not\models \gamma$. Assume for a proof by contradiction that γ has no target in $\mathcal{T}$. We first show the following

Claim 1. If γ has no target in $\mathcal{T}$ and $\mathcal{T} \models \mathsf{lhs}(\gamma) \sqsubseteq A$ then $A \in \mathsf{lhs}(\gamma)$, for all $A \in \mathsf{N_C}$.

For the proof of Claim 1, consider the canonical model $\mathcal{I}_{\mathsf{lhs}(\gamma),\mathcal{T}}$ of $\mathsf{lhs}(\gamma)$ and $\mathcal{T}$. Recall that $\rho_{\mathsf{lhs}(\gamma),\mathcal{T}}$ denotes the root of $\mathcal{I}_{\mathsf{lhs}(\gamma),\mathcal{T}}$. By Lemma 12, $\rho_{\mathsf{lhs}(\gamma),\mathcal{T}} \in D^{\mathcal{I}_{\mathsf{lhs}(\gamma),\mathcal{T}}}$ iff $\mathcal{T} \models \mathsf{lhs}(\gamma) \sqsubseteq D$, for any concept D. Thus, it suffices to prove that $\rho_{\mathsf{lhs}(\gamma),\mathcal{T}} \in A^{\mathcal{I}_{\mathsf{lhs}(\gamma),\mathcal{T}}}$ implies $A \in \mathsf{lhs}(\gamma)$, for all concept names A. The proof is by induction over the sequence $\mathcal{I}_0, \ldots$ used to construct $\mathcal{I}_{\mathsf{lhs}(\gamma),\mathcal{T}}$, where $\mathcal{I}_0 = \mathcal{I}_{\mathsf{lhs}(\gamma)}$. For $\mathcal{I}_{\mathsf{lhs}(\gamma)}$ this is the case by definition. Now suppose the claim holds for $\mathcal{I}_n$ and $\rho_{\mathsf{lhs}(\gamma),\mathcal{T}} \in A^{\mathcal{I}_{n+1}} \setminus A^{\mathcal{I}_n}$. Then there either exist concept names $A_1, \ldots, A_k$ with $A_1 \sqcap \cdots \sqcap A_k \sqsubseteq A \in \mathcal{T}$ and $\rho_{\mathsf{lhs}(\gamma),\mathcal{T}} \in (A_1 \sqcap \cdots \sqcap A_k)^{\mathcal{I}_n}$ or there exists $\exists r.\top$ with $\exists r.\top \sqsubseteq A \in \mathcal{T}$ and $\rho_{\mathsf{lhs}(\gamma),\mathcal{T}} \in (\exists r.\top)^{\mathcal{I}_n}$. In the first case, we have $\{A_1, \ldots, A_k\} \subseteq \mathsf{lhs}(\gamma)$ by induction hypothesis and so $A \in \mathsf{lhs}(\gamma)$ because otherwise $A_1 \sqcap \cdots \sqcap A_k \sqsubseteq A$ would be a target of γ. In the second case there must be an $\mathcal{I}_m$ with $m < n$ such that there are $E_1 \sqcap \cdots \sqcap E_k \sqsubseteq \exists s.D \in \mathcal{T}$ and $s \sqsubseteq r \in \mathcal{T}$ with $\rho_{\mathsf{lhs}(\gamma),\mathcal{T}} \in (E_1 \sqcap \cdots \sqcap E_k)^{\mathcal{I}_m}$ (the case $s = r$ is similar and omitted). It follows that $A \in \mathsf{lhs}(\gamma)$ because otherwise $E_1 \sqcap \cdots \sqcap E_k \sqsubseteq \exists s.D$ would

be a target of γ since, by induction hypothesis, $\{E_1, \ldots, E_k\} \subseteq \mathsf{lhs}(\gamma)$ and $A \in (\exists s.D)^{\mathsf{sat}}$. This finishes the proof of Claim 1.

By Claim 1, as $\emptyset \not\models \gamma$, there is a conjunct of the form $\exists r.F$ in $\mathsf{rhs}(\gamma)$. Let $(\mathsf{lhs}(\alpha))^{\mathcal{T}}$ be as above and $\mathcal{I}_{D_0,\mathcal{T}}$ be the canonical model of $D_0 = \bigsqcap_{D \in (\mathsf{lhs}(\alpha))^{\mathcal{T}}} D$ and $\mathcal{T}$. As $\exists r.F \in \mathsf{rhs}(\gamma)$ and γ is $\mathcal{T}$-essential one can show in the same way as in the proof of Lemma 32 that there is an injective homomorphism from the labelled tree $T_{\exists r.F}$ corresponding to $\exists r.F$ into the restriction of $\mathcal{I}_{D_0,\mathcal{T}}$ to $\Delta^{\mathcal{I}_{D_0}}$ mapping the root of $T_{\exists r.F}$ to the root $\rho_{D_0,\mathcal{T}}$ of $\mathcal{I}_{D_0,\mathcal{T}}$. Thus, by definition of $(\mathsf{lhs}(\alpha))^{\mathcal{T}}$, there is $\beta \in \mathcal{T}$ such that $\mathcal{T} \models \mathsf{lhs}(\alpha) \sqsubseteq \mathsf{lhs}(\beta)$ and $\emptyset \models \mathsf{rhs}(\beta)^{\mathsf{sat}} \sqsubseteq \exists r.F$. By Lemma 12, $\rho_{\mathsf{lhs}(\gamma),\mathcal{T}} \in A^{\mathcal{I}_{\mathsf{lhs}(\gamma),\mathcal{T}}}$, for all $A \in \mathsf{lhs}(\beta)$. Hence, by Claim 1 and again Lemma 12, $\mathsf{lhs}(\beta) \subseteq \mathsf{lhs}(\gamma)$. We have shown that γ has target β and so derived a contradiction. ❏

Point (iii) is a direct consequence of Lemma 46 and the fact that all CIs in $\mathcal{H}_{add}$ are $\mathcal{T}$-essential (Lemma 38). To prove Point (iv), we first establish the following intermediate Lemmas 47 and 48.

Lemma 47 *Let $\mathcal{T}$ be rhs-primitive and let $\mathcal{H}, \gamma$ be inputs to CN-Refine. Let $\alpha_i \in \mathcal{H}_{add}$, $\beta \in \mathcal{T}$, and concept name $A \notin \mathsf{lhs}(\gamma)$ satisfy the following conditions: (a) $\mathsf{lhs}(\beta) \subseteq \mathsf{lhs}(\gamma)$; (b) $\mathcal{T} \models \mathsf{lhs}(\beta) \sqsubseteq A$; (c) $\mathsf{lhs}(\beta) \subseteq \mathsf{lhs}(\alpha_i)$. Then there is some $j \leq i$ such that α_j is replaced in Line 4 of CN-Refine.*

Proof Assume $\mathcal{H}$, γ, α_i, β, and A satisfy the conditions of the lemma. If CN-Refine replaces some α_j with $j < i$ then we are done. Suppose this does not happen. Then we need to show that α_i is replaced. By Conditions (a), (b), and (c), $\mathcal{T} \models \mathsf{lhs}(\gamma) \sqcap \mathsf{lhs}(\alpha_i) \sqsubseteq A$. As γ is left saturated for $\mathcal{H}$, $A \notin \mathsf{lhs}(\gamma)$ implies that $\mathcal{H} \not\models \mathsf{lhs}(\gamma) \sqsubseteq A$. So $\mathcal{H} \not\models \mathsf{lhs}(\gamma) \sqcap \mathsf{lhs}(\alpha_i) \sqsubseteq A$. Then, the condition in Lines 1 and 2 of CN-Refine is satisfied and α_i is replaced. ❏

Lemma 48 *Let $\mathcal{T}$ be rhs-primitive and let $\mathcal{H}, \gamma$ be inputs to $\exists$-Refine. If γ has target $\beta \in \mathcal{T}$ and $\alpha_i \in \mathcal{H}_{add}$ satisfies $\mathsf{lhs}(\beta) \subseteq \mathsf{lhs}(\alpha_i)$, then there is some $j \leq i$ such that α_j is replaced in Line 5 or 8 of $\exists$-Refine.*

Proof Let $\mathcal{H}$, γ, β, and α_i satisfy the conditions of the lemma. If $\exists$-Refine replaces some α_j with $j < i$ then we are done. Suppose this does not happen. We need to show that α_i is replaced. We first show that there is a concept C of the form $\exists r.F$ in $\mathsf{rhs}(\gamma)$ such that $\mathsf{lhs}(\gamma) \sqsubseteq C$ has target β. Note that if Algorithm 3 calls $\exists$-Refine then there is no concept name A such that $\mathcal{T} \models \mathsf{lhs}(\gamma) \sqsubseteq A$ and $\mathcal{H} \not\models \mathsf{lhs}(\gamma) \sqsubseteq A$ (Line 10). As γ is left saturated for $\mathcal{H}$ and $\mathcal{T}$-essential, this implies $\mathsf{N_C} \cap \mathsf{rhs}(\gamma) \subseteq \mathsf{lhs}(\gamma)$. But then any $C \in \mathsf{rhs}(\gamma) \setminus \mathsf{lhs}(\gamma)$ with $\emptyset \models \mathsf{rhs}(\beta)^{\mathsf{sat}} \sqsubseteq C$ is compound. As γ has target β it follows that $\mathsf{lhs}(\gamma) \sqsubseteq C$ has target β for some C of the form $\exists r.F$ in $\mathsf{rhs}(\gamma)$. By Line 5 of Algorithm 3, there is only one such conjunct C in $\mathsf{rhs}(\gamma)$. From $\emptyset \models \mathsf{rhs}(\beta)^{\mathsf{sat}} \sqsubseteq C$ we obtain $\mathcal{T} \models \mathsf{lhs}(\beta) \sqsubseteq C$. Since $\mathsf{lhs}(\beta) \subseteq \mathsf{lhs}(\gamma) \cap \mathsf{lhs}(\alpha_i)$, we have that $\mathcal{T} \models \mathsf{lhs}(\gamma) \cap \mathsf{lhs}(\alpha_i) \sqsubseteq C$. As γ is a positive counterexample, $\mathcal{H} \not\models \gamma$. From $\mathsf{N_C} \cap \mathsf{rhs}(\gamma) \subseteq \mathsf{lhs}(\gamma)$ we thus obtain $\mathcal{H} \not\models \mathsf{lhs}(\gamma) \sqsubseteq C$, and so, $\mathcal{H} \not\models \mathsf{lhs}(\alpha_i) \cap \mathsf{lhs}(\gamma) \sqsubseteq C$. Hence, the condition in Lines 1 and 2 of $\exists$-Refine is satisfied and α_i is replaced (in Line 5 or 8). ❏

Point (iv) above is now a direct consequence of the following lemma.

Lemma 49 *At any point in the execution of Algorithm 3, if $\alpha_j \in \mathcal{H}_{add}$ has target $\beta \in \mathcal{T}$ then $\mathsf{lhs}(\beta) \not\subseteq \mathsf{lhs}(\alpha_i)$, for all $i < j$.*

Proof The proof is by induction on the number k of iterations. For $k = 1$ the lemma is vacuously true. Assume it holds for $k = n$, $n \geq 1$. Now the algorithm modifies $\mathcal{H}_{add}$ in response to receiving a positive counterexample in iteration $k = n + 1$. We make a case distinction:

Case 1. Algorithm 3 calls CN-Refine: Let $\mathcal{H}, \gamma$ be the inputs to CN-Refine. Assume first that the condition in Lines 1 and 2 is not satisfied. Then CN-Refine appends the result of concept saturating γ for $\mathcal{T}$ to $\mathcal{H}_{add}$. Call this CI γ'. Suppose that the lemma fails to hold. This can only happen if γ' has a target $\beta \in \mathcal{T}$ and there is $\alpha_i \in \mathcal{H}_{add}$ such that $\mathsf{lhs}(\beta) \subseteq \mathsf{lhs}(\alpha_i)$. Then, since $\mathsf{lhs}(\gamma') = \mathsf{lhs}(\gamma)$, we have that $\mathsf{lhs}(\beta) \subseteq \mathsf{lhs}(\gamma)$ and, since $\mathsf{rhs}(\gamma') \subseteq \mathsf{N_C}$, there is a concept name $A \notin \mathsf{lhs}(\gamma)$ such that $\emptyset \models \mathsf{rhs}(\beta)^{\mathsf{sat}} \sqsubseteq A$. So $\mathcal{T} \models \mathsf{lhs}(\beta) \sqsubseteq A$. Then Lemma 47 applies to $\mathcal{H}$, γ, α_i, β and A which contradicts the assumption that CN-Refine did not replace any $\alpha_j \in \mathcal{H}_{add}$, $j \leq i$.

Now assume that the condition in Lines 1 and 2 is satisfied. Suppose that the lemma fails to hold. This can only happen if there are $\alpha_i, \alpha_j \in \mathcal{H}_{add}$ with $i < j$ such that either (a) α_i is replaced by α_i', $\mathsf{lhs}(\beta) \subseteq \mathsf{lhs}(\alpha_i')$ and α_j has target β; or (b) α_j is replaced by α_j', α_j' has target β and $\mathsf{lhs}(\beta) \subseteq \mathsf{lhs}(\alpha_i)$. In case (a), from $\mathsf{lhs}(\gamma) \cap \mathsf{lhs}(\alpha_i) = \mathsf{lhs}(\alpha_i')$, we obtain $\mathsf{lhs}(\beta) \subseteq \mathsf{lhs}(\gamma) \cap \mathsf{lhs}(\alpha_i)$. Thus, $\mathsf{lhs}(\beta) \subseteq \mathsf{lhs}(\alpha_i)$. This contradicts the induction hypothesis. Now assume case (b). Since α_j' has target β, we obtain:

1. $\mathsf{lhs}(\beta) \subseteq \mathsf{lhs}(\alpha_j')$; and
2. as $\mathsf{rhs}(\alpha_j') \subseteq \mathsf{N_C}$, there is $A \in \mathsf{N_C}$ with $A \in \mathsf{rhs}(\alpha_j') \setminus \mathsf{lhs}(\alpha_j')$ and $\emptyset \models \mathsf{rhs}(\beta)^{\mathsf{sat}} \sqsubseteq A$.

Since $\mathsf{lhs}(\gamma) \cap \mathsf{lhs}(\alpha_j) = \mathsf{lhs}(\alpha_j')$, it follows from Point 1 that $\mathsf{lhs}(\beta) \subseteq \mathsf{lhs}(\alpha_j)$ and $\mathsf{lhs}(\beta) \subseteq \mathsf{lhs}(\gamma)$. From $\emptyset \models \mathsf{rhs}(\beta)^{\mathsf{sat}} \sqsubseteq A$ we obtain $\mathcal{T} \models \mathsf{lhs}(\beta) \sqsubseteq A$. If $A \in \mathsf{rhs}(\alpha_j') \setminus \mathsf{lhs}(\alpha_j')$ then either $A \in \mathsf{rhs}(\alpha_j) \setminus \mathsf{lhs}(\alpha_j)$ or $A \notin \mathsf{lhs}(\gamma)$. So either α_j has target β or $A \notin \mathsf{lhs}(\gamma)$. α_j does not have target β as this would contradict the induction hypothesis. Thus, $A \notin \mathsf{lhs}(\gamma)$ and the conditions of Lemma 47 are satisfied by $\mathcal{H}$, γ, α_i, β, and A. Thus, some $\alpha_{i'}$ with $i' \leq i$ is replaced which contradicts the assumption that α_j is replaced.

Case 2. Algorithm 3 calls $\exists$-Refine: Let $\mathcal{H}, \gamma$ be the inputs to $\exists$-Refine. Assume first that the condition in Lines 1 and 2 is not satisfied. Then $\exists$-Refine appends γ to $\mathcal{H}_{add}$. Suppose the lemma fails to hold. This can only happen if γ has a target $\beta \in \mathcal{T}$ and there is $\alpha_i \in \mathcal{H}_{add}$ such that $\mathsf{lhs}(\beta) \subseteq \mathsf{lhs}(\alpha_i)$. By Lemma 48, this contradicts the assumption that $\exists$-Refine did not replace any $\alpha_j \in \mathcal{H}_{add}$, $j \leq i$.

Assume now that the condition in Lines 1 and 2 is satisfied. Suppose that the lemma fails to hold. This can only happen if there are $\alpha_i, \alpha_j \in \mathcal{H}_{add}$ with $i < j$ such that either (a) α_i is replaced by α_i', $\mathsf{lhs}(\beta) \subseteq \mathsf{lhs}(\alpha_i')$ and α_j has target β; or (b) α_j is replaced by α_j', α_j' has target β and $\mathsf{lhs}(\beta) \subseteq \mathsf{lhs}(\alpha_i)$. For case (a) we argue as above: from $\mathsf{lhs}(\gamma) \cap \mathsf{lhs}(\alpha_i) = \mathsf{lhs}(\alpha_i')$, we obtain $\mathsf{lhs}(\beta) \subseteq \mathsf{lhs}(\gamma) \cap \mathsf{lhs}(\alpha_i)$. Thus, $\mathsf{lhs}(\beta) \subseteq \mathsf{lhs}(\alpha_i)$, which contradicts the induction hypothesis. Now assume case (b). As α_j' has target β, we obtain the following:

1. $\mathsf{lhs}(\beta) \subseteq \mathsf{lhs}(\alpha'_j)$; and

2. there is $D \in \mathsf{rhs}(\alpha'_j) \setminus \mathsf{lhs}(\alpha'_j)$ and $\emptyset \models \mathsf{rhs}(\beta)^{\mathsf{sat}} \sqsubseteq D$.

Since $\mathsf{lhs}(\gamma) \cap \mathsf{lhs}(\alpha_j) = \mathsf{lhs}(\alpha'_j)$, it follows from Point 1 that $\mathsf{lhs}(\beta) \subseteq \mathsf{lhs}(\alpha_j)$ and $\mathsf{lhs}(\beta) \subseteq \mathsf{lhs}(\gamma)$. Recall that if Algorithm 3 calls $\exists$-Refine then there is no $A \in \mathsf{N_C}$ such that $\mathcal{T} \models \mathsf{lhs}(\gamma) \sqsubseteq A$ and $\mathcal{H} \not\models \mathsf{lhs}(\gamma) \sqsubseteq A$. So $\mathsf{N_C} \cap \mathsf{rhs}(\gamma) \subseteq \mathsf{lhs}(\gamma)$ (by left saturation of γ for $\mathcal{H}$). Assume $D \in \mathsf{N_C}$. Since $D \in \mathsf{rhs}(\alpha'_j) \setminus \mathsf{lhs}(\alpha'_j)$ (Point 2), it follows that $D \notin \mathsf{lhs}(\alpha_j)$. As $\mathsf{lhs}(\alpha'_j) \subseteq \mathsf{lhs}(\alpha_j)$, we have that $D \in \mathsf{rhs}(\alpha_j)$. So $D \in \mathsf{rhs}(\alpha_j) \setminus \mathsf{lhs}(\alpha_j)$. This means that α_j has target β, which contradicts the induction hypothesis. Otherwise, D is of the form $\exists r.F$. Then, either $D \in \mathsf{rhs}(\gamma)$ or there is $D' \in \mathsf{rhs}(\alpha_j)$ such that $\emptyset \models D \sqsubseteq D'$. In the latter case, $D' \in \mathsf{rhs}(\alpha_j) \setminus \mathsf{lhs}(\alpha_j)$ and $\emptyset \models \mathsf{rhs}(\beta)^{\mathsf{sat}} \sqsubseteq D'$, so α_j has target β, which contradicts the induction hypothesis. In the former case, γ has target β. Then $\mathcal{H}$, γ, and α_i satisfy the conditions of Lemma 48. Thus, some $\alpha_{i'}$ with $i' \leq i$ is replaced which contradicts the assumption that α_j is replaced. ❑

We have proved the main result of this section.

Theorem 50 *DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ TBoxes are polynomial query learnable using membership and equivalence queries. Moreover, DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ TBoxes without inverse roles can be learned in polynomial time using membership and equivalence queries.*

Proof Polynomial query learnability of DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ TBoxes follows from Lemma 41 and the analysis of the number of membership queries in Lemmas 39 and 40, see the proof of Theorem 34. For the second part observe that the only potentially costly steps are entailment checks of the form $\mathcal{H} \models \alpha$, where $\mathcal{H}$ is a DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ TBox and α a DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ CI, both without inverse roles. Then both $\mathcal{H}$ and α are in $\mathcal{EL}$ with role inclusions for which entailment is known to be in PTIME (Baader et al., 2005). ❑

5. Learning $\mathcal{EL}_{\mathsf{lhs}}$ TBoxes

We study polynomial learnability of TBoxes in the restriction $\mathcal{EL}_{\mathsf{lhs}}$ of $\mathcal{EL}$ in which only concept names are allowed on the right-hand side of CIs. We assume that CIs used in membership queries and in equivalence queries and those returned as counterexamples are also of this restricted form and show that under this assumption $\mathcal{EL}_{\mathsf{lhs}}$ TBoxes can be learned in polynomial time. As in the previous section, our learning algorithm is an extension of the polynomial time algorithm for learning propositional Horn theories presented by Angluin et al. (1992) and Arias and Balcázar (2011).

There is a certain similarity between the learning algorithm of this section and the DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ learning algorithm introduced in Section 4. In both cases the left-hand side of inclusions can contain complex concept expressions, which, unless addressed, might lead to several counterexamples with unnecessarily strong left-hand sides targeting the same inclusion in the target TBox. In Algorithm 3 storing multiple such counterexamples in $\mathcal{H}_{add}$ is prevented by taking the intersection of the set of conjuncts of the left-hand sides. To deal with the more complex left-hand sides of inclusions in $\mathcal{EL}_{\mathsf{lhs}}$, a more sophisticated way of 'taking the intersection' of concept expressions is required. To define it, we identify concept

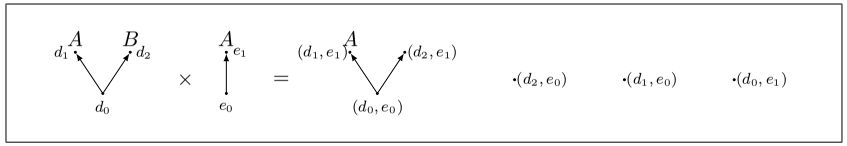


Figure 11: Illustration to Example 52.

expressions with tree-shaped interpretations and then take their product. Products have also been employed in the construction of least common subsumers (Baader et al., 1999).

In detail, we say that an interpretation $\mathcal{I}$ is a *ditree interpretation* if the directed graph $(\Delta^{\mathcal{I}}, E)$ with $E = \bigcup_{r \in \mathsf{N_R}} r^{\mathcal{I}}$ is a directed tree and $r^{\mathcal{I}} \cap s^{\mathcal{I}} = \emptyset$ for all distinct $r, s \in \mathsf{N_R}$. We denote the root of a ditree interpretation $\mathcal{I}$ with $\rho_{\mathcal{I}}$. The interpretation $\mathcal{I}_C$ corresponding to an $\mathcal{EL}$ concept expression C is a ditree interpretation with root ρ_C. Conversely, every ditree interpretation $\mathcal{I}$ can be viewed as an $\mathcal{EL}$ concept expression $C_{\mathcal{I}}$ in the same way as any labelled tree T with edge labels that are role names (rather than arbitrary roles) can be seen as an $\mathcal{EL}$ concept expression.

An interpretation $\mathcal{I}$ is a *$\mathcal{T}$-countermodel* for a given $\mathcal{EL}_{\mathsf{lhs}}$ TBox $\mathcal{T}$ if $\mathcal{I} \not\models \mathcal{T}$. Notice that for any $\mathcal{EL}_{\mathsf{lhs}}$ inclusion $C \sqsubseteq A$ with $\mathcal{T} \models C \sqsubseteq A$ and $\emptyset \not\models C \sqsubseteq A$ the interpretation $\mathcal{I}_C$ is a $\mathcal{T}$-countermodel. Indeed, by construction of $\mathcal{I}_C$, we have $\rho_C \in C^{\mathcal{I}_C}$ and, as $\emptyset \not\models C \sqsubseteq A$, we have $\rho_C \notin A^{\mathcal{I}_C}$. So $\mathcal{I}_C \not\models C \sqsubseteq A$ and, as $\mathcal{T} \models C \sqsubseteq A$, we have $\mathcal{I}_C \not\models \mathcal{T}$. Conversely, given a $\mathcal{T}$-countermodel $\mathcal{I}$, a learning algorithm can construct in polynomial time in $|\Sigma_{\mathcal{T}}|$ all inclusions of the form $C_{\mathcal{I}} \sqsubseteq A$, where A is a concept name, such that $\mathcal{T} \models C_{\mathcal{I}} \sqsubseteq A$ by posing membership queries to the oracle. Thus a learning algorithm can use inclusions and $\mathcal{T}$-countermodels interchangeably. We prefer working with interpretations as we can then use the notion of products to define the 'intersection of concept expressions' and the results of Section 2 linking homomorphisms with entailment in a direct way.

The *product* of two interpretations $\mathcal{I}$ and $\mathcal{J}$ is the interpretation $\mathcal{I} \times \mathcal{J}$ with

$$\begin{aligned} \Delta^{\mathcal{I}\times\mathcal{J}} &= \Delta^{\mathcal{I}} \times \Delta^{\mathcal{J}} \\ A^{\mathcal{I}\times\mathcal{J}} &= \{(d,e) \mid d \in A^{\mathcal{I}}, e \in A^{\mathcal{J}}\} \\ r^{\mathcal{I}\times\mathcal{J}} &= \{((d,e),(d',e')) \mid (d,d') \in r^{\mathcal{I}}, (e,e') \in r^{\mathcal{J}}\} \end{aligned}$$

Products preserve the membership in $\mathcal{EL}$ concept expressions (Lutz et al., 2011):

Lemma 51 *For all interpretations $\mathcal{I}$ and $\mathcal{J}$, all $d \in \Delta^{\mathcal{I}}$ and $e \in \Delta^{\mathcal{J}}$, and for all $\mathcal{EL}$ concept expressions C the following holds: $d \in C^{\mathcal{I}}$ and $e \in C^{\mathcal{J}}$ if, and only if, $(d,e) \in C^{\mathcal{I}\times\mathcal{J}}$.*

One can easily show that the product of ditree interpretations is a disjoint union of ditree interpretations. If $\mathcal{I}$ and $\mathcal{J}$ are ditree interpretations, we denote by $\mathcal{I} \times_\rho \mathcal{J}$ the maximal ditree interpretation that is a subinterpretation of $\mathcal{I} \times \mathcal{J}$ and contains $(\rho_{\mathcal{I}}, \rho_{\mathcal{J}})$.

Example 52 *Figure 11 depicts the product of the ditree interpretations $\mathcal{I}$ with root d_0 and $\mathcal{J}$ with root e_0. The ditree interpretation $\mathcal{I} \times_\rho \mathcal{J}$ has root (d_0, e_0) and does not contain the nodes (d_2, e_0), (d_1, e_0) and (d_0, e_1) from $\mathcal{I} \times \mathcal{J}$.*

Observe that the product $\mathcal{I}_C \times \mathcal{I}_D$ of concept expressions $C = A_1 \sqcap \cdots \sqcap A_n$ and $D = B_1 \sqcap \cdots \sqcap B_m$, where $A_1, \ldots, A_n$ and $B_1, \ldots, B_m$ are concept names, coincides with the interpretation $\mathcal{I}_E$, where E is the conjunction of all concept names in $\{A_1, \ldots, A_n\} \cap \{B_1, \ldots, B_m\}$. Thus, products can be seen as a generalisation of taking the intersection of the concept names from the left-hand side of DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ concept inclusions used in Section 4.

We will now describe a class of $\mathcal{T}$-countermodels that are in a sense minimal and central to our learning algorithm. Let $\mathcal{T}$ be the $\mathcal{EL}_{\mathsf{lhs}}$ TBox to be learned, and assume that its signature $\Sigma_{\mathcal{T}}$ is known to the learner. For a ditree interpretation $\mathcal{I}$, we use $\mathcal{I}|^{-}_{\rho}$ to denote the interpretation obtained from $\mathcal{I}$ by removing the root $\rho_{\mathcal{I}}$ of $\mathcal{I}$. For any $d \in \Delta^{\mathcal{I}} \setminus \{\rho_{\mathcal{I}}\}$, we use $\mathcal{I}|^{-}_{d\downarrow}$ to denote $\mathcal{I}$ with the subtree rooted at d removed. A $\mathcal{T}$-countermodel is *essential* if the following conditions are satisfied:

1. $\mathcal{I}|^{-}_{\rho} \models \mathcal{T}$;
2. $\mathcal{I}|^{-}_{d\downarrow} \models \mathcal{T}$ for all $d \in \Delta^{\mathcal{I}} \setminus \{\rho_{\mathcal{I}}\}$.

Intuitively, Condition 1 states that $\mathcal{I}$ contradicts $\mathcal{T}$ only at the root, that is, the only reason for why $\mathcal{I}$ does not satisfy $\mathcal{T}$ is that for at least one CI $C \sqsubseteq A \in \mathcal{T}$, we have that $\rho_{\mathcal{I}} \in C^{\mathcal{I}}$ and $\rho_{\mathcal{I}} \notin A^{\mathcal{I}}$. Condition 2 is a minimality condition which states that for any such $C \sqsubseteq A \in \mathcal{T}$, $\rho_{\mathcal{I}}$ is no longer in $C^{\mathcal{I}}$ if we remove any node from $\mathcal{I}$. Example 61 at the end of this section shows that working with essential $\mathcal{T}$-countermodels is needed for our learning algorithm to be in polynomial time.

The algorithm for learning $\mathcal{EL}_{\mathsf{lhs}}$ TBoxes is given as Algorithm 6. It maintains an ordered list $\mathfrak{I}$ of ditree interpretations that intuitively represents the TBox $\mathcal{H}$ constructed in Line 13. In Line 6 we write $\mathcal{I} \to_{\rho} \mathcal{J}$ if there is a homomorphism from a ditree interpretation $\mathcal{I}$ to a ditree interpretation $\mathcal{J}$ mapping $\rho_{\mathcal{I}}$ to $\rho_{\mathcal{J}}$. $\mathcal{I} \not\to_{\rho} \mathcal{J}$ denotes that no such homomorphism exists. By Lemma 10, $\mathcal{I} \to_{\rho} \mathcal{J}$ iff $\emptyset \models C_{\mathcal{J}} \sqsubseteq C_{\mathcal{I}}$ which can be checked in polynomial time in the size of $\mathcal{I}$ and $\mathcal{J}$. In Line 8, we write $\mathcal{J}' \subseteq \mathcal{I} \times_{\rho} \mathcal{J}$ as shorthand for the condition that $\mathcal{J}'$ is a subinterpretation of $\mathcal{I} \times_{\rho} \mathcal{J}$ that is obtained from $\mathcal{I} \times_{\rho} \mathcal{J}$ by removing subtrees. Note that the assumption in Line 4 that a *positive* counterexample is returned is justified by the construction of $\mathcal{H}$ in Lines 2 and 13, which ensures that, at all times, $\mathcal{T} \models \mathcal{H}$.

We now provide additional details on how to realise lines 5, 8 and 13. Line 13 is the easiest: we simply use membership queries '$\mathcal{T} \models C_{\mathcal{I}} \sqsubseteq A$?' with $\mathcal{I} \in \mathfrak{I}$ and $A \in \Sigma_{\mathcal{T}}$ to find all CIs $C_{\mathcal{I}} \sqsubseteq A$ entailed by $\mathcal{T}$. We will later show that the length of $\mathfrak{I}$ is bounded polynomially in $|\mathcal{T}|$ and that each interpretation in $\mathfrak{I}$ is replaced only polynomially many times, therefore polynomially many membership queries suffice. Lines 5 and 8 are addressed by Lemmas 53 and 54 below.

Lemma 53 *Given a positive counterexample $C \sqsubseteq A$ for $\mathcal{T}$ relative to $\mathcal{H}$, one can construct an essential $\mathcal{T}$-countermodel $\mathcal{I}$ with $\mathcal{I} \models \mathcal{H}$ using only polynomially many membership queries in $|\mathcal{T}| + |C|$.*

Proof Let $C \sqsubseteq A$ be a positive counterexample for $\mathcal{T}$ relative to $\mathcal{H}$. Let $\mathcal{I}_C$ be the ditree interpretation of C. First observe that $\mathcal{I}_C \not\models \mathcal{T}$: since $\mathcal{H} \not\models C \sqsubseteq A$, we know that A does not occur as a top-level conjunct in C. Consequently, $\rho_C \in C^{\mathcal{I}_C} \setminus A^{\mathcal{I}_C}$ and thus $\mathcal{I}_C \not\models \mathcal{T}$.

We construct an essential $\mathcal{T}$-countermodel $\mathcal{I}$ with $\mathcal{I} \models \mathcal{H}$ by applying the following rules to $\mathcal{I} := \mathcal{I}_C$.

Algorithm 6 The learning algorithm for $\mathcal{EL}_{\mathsf{lhs}}$ TBoxes

Input: $\mathcal{EL}_{\mathsf{lhs}}$ TBox $\mathcal{T}$ given to the oracle; $\Sigma_{\mathcal{T}}$ given to the learner.

Output: TBox $\mathcal{H}$, computed by the learner, such that $\mathcal{T} \equiv \mathcal{H}$.

```
1:  Set 𝔍 to the empty list (of ditree interpretations)
2:  Set ℋ = ∅
3:  while ℋ ⊭≡ 𝒯 do
4:      Let C ⊑ A be the returned positive counterexample for 𝒯 relative to ℋ
5:      Find an essential 𝒯-countermodel ℐ with ℐ ⊨ ℋ
6:      if there is a 𝒥 ∈ 𝔍 such that 𝒥 ↛_ρ (ℐ ×_ρ 𝒥) and ℐ ×_ρ 𝒥 ⊭ 𝒯 then
7:          Let 𝒥 be the first such element of 𝔍
8:          Find an essential 𝒯-countermodel 𝒥' ⊆ ℐ ×_ρ 𝒥
9:          Replace 𝒥 in 𝔍 with 𝒥'
10:     else
11:         Append ℐ to 𝔍
12:     end if
13:     Construct ℋ = {C_ℐ ⊑ A | ℐ ∈ 𝔍, A a concept name in Σ_𝒯, 𝒯 ⊨ C_ℐ ⊑ A}
14: end while
15: return ℋ
```

1. Saturate $\mathcal{I}$ by exhaustively applying the CIs from $\mathcal{H}$ as rules: if $D \sqsubseteq B \in \mathcal{H}$ and $d \in D^{\mathcal{I}}$, then add d to $B^{\mathcal{I}}$.
2. Replace $\mathcal{I}$ by a minimal subtree of $\mathcal{I}$ refuting $\mathcal{T}$ to address Condition 1 of essential $\mathcal{T}$-countermodels. To describe how this can be achieved using membership queries denote for $d \in \Delta^{\mathcal{I}}$ by $\mathcal{I}|_d$ the ditree interpretation obtained from $\mathcal{I}$ by taking the subtree of $\mathcal{I}$ rooted in d. Now check using membership queries for any $d \in \Delta^{\mathcal{I}} \setminus \{\rho_{\mathcal{I}}\}$ and concept name B whether $\mathcal{T} \models C_{\mathcal{I}|_d} \sqsubseteq B$. Then replace $\mathcal{I}$ by any $\mathcal{I}|_d$ such that there exists a B with $\mathcal{T} \models C_{\mathcal{I}|_d} \sqsubseteq B$ and $d \notin B^{\mathcal{I}|_d}$ but there does not exist a d' in $\Delta^{\mathcal{I}|_d}$ and a B' with $\mathcal{T} \models C_{\mathcal{I}|_{d'}} \sqsubseteq B'$ and $d' \notin B^{\mathcal{I}|_{d'}}$. If no such d and B exist, then $\mathcal{I}$ is not replaced.
3. Exhaustively remove subtrees from $\mathcal{I}$ until Condition 2 of essential $\mathcal{T}$-countermodels is also satisfied: if $\mathcal{I}|^{-}_{d\downarrow} \not\models \mathcal{T}$, then replace $\mathcal{I}$ by $\mathcal{I}|^{-}_{d\downarrow}$. This can again be achieved using the membership queries $\mathcal{T} \models C_{\mathcal{I}|^{-}_{d\downarrow}} \sqsubseteq B$ for B a concept name.

Now we show that the interpretation $\mathcal{J}$ constructed above has the required properties. First observe that $\mathcal{J} \models \mathcal{H}$: clearly, the interpretation $\mathcal{I}$ constructed in Step 1 is a model of $\mathcal{H}$. As taking subtrees and removing subtrees from $\mathcal{I}$ preserves being a model of $\mathcal{H}$, we conclude that $\mathcal{J} \models \mathcal{H}$. Next we show that $\mathcal{J} \not\models \mathcal{T}$: the interpretation $\mathcal{I}$ constructed in Step 1 is not a model of $\mathcal{T}$. In fact, we can use $C_{\mathcal{I}} \sqsubseteq A$ as a positive counterexample for $\mathcal{T}$ relative to $\mathcal{H}$ instead of $C \sqsubseteq A$. Observe that $\emptyset \models C_{\mathcal{I}} \sqsubseteq C$, and thus $\mathcal{T} \models C \sqsubseteq A$ implies $\mathcal{T} \models C_{\mathcal{I}} \sqsubseteq A$. On the other hand, $\rho_{\mathcal{I}} \in B^{\mathcal{I}}$ implies $\mathcal{H} \models C \sqsubseteq B$ for all concept names B. Consequently and since $\mathcal{H} \not\models C \sqsubseteq A$, we have $\rho_{\mathcal{I}} \notin A^{\mathcal{I}}$. Thus $\mathcal{I} \not\models \mathcal{T}$. By construction, Steps 2 and 3 preserve the condition that $\mathcal{I}$ is not a model of $\mathcal{T}$ and so $\mathcal{J} \not\models \mathcal{T}$. It remains to argue that $\mathcal{J}$

satisfies Conditions 1 and 2 for essential $\mathcal{T}$-countermodels for $\mathcal{H}$. But Condition 1 is ensured by Step 2 and Condition 2 is ensured by Step 3, respectively. ❑

Lemma 54 *Given essential $\mathcal{T}$-countermodels $\mathcal{I}$ and $\mathcal{J}$ with $\mathcal{I} \times_\rho \mathcal{J} \not\models \mathcal{T}$, one can construct an essential $\mathcal{T}$-countermodel $\mathcal{J}' \subseteq \mathcal{I} \times_\rho \mathcal{J}$ using only polynomially many membership queries in $|\mathcal{T}| + |\mathcal{I}| + |\mathcal{J}|$.*

Proof Let $\mathcal{I}$ and $\mathcal{J}$ be essential $\mathcal{T}$-countermodels with $\mathcal{I} \times_\rho \mathcal{J} \not\models \mathcal{T}$. Obtain the interpretation $\mathcal{J}'$ from $\mathcal{I} \times_\rho \mathcal{J}$ by exhaustively applying Rule 3 from the proof of Lemma 53. As argued above, applying Rule 3 can be implemented using membership queries and $\mathcal{J}'$ is a $\mathcal{T}$-countermodel. Thus, it remains to argue that it satisfies Conditions 1 and 2 for $\mathcal{T}$-essential countermodels. For Condition 1, we have to show that $\mathcal{J}'|_\rho^- \models \mathcal{T}$. We know that $\mathcal{I}|_\rho^- \models \mathcal{T}$ and $\mathcal{J}|_\rho^- \models \mathcal{T}$. Thus, by Lemma 51, $\mathcal{I}|_\rho^- \times \mathcal{J}|_\rho^- \models \mathcal{T}$. Now $\mathcal{J}'|_\rho^-$ is obtained from $\mathcal{I}|_\rho^- \times \mathcal{J}|_\rho^-$ by removing subtrees and removing subtrees preserves being a model of an $\mathcal{EL}_{\mathsf{lhs}}$ TBox. Thus, $\mathcal{J}'|_\rho^- \models \mathcal{T}$. For Condition 2, we have to show that $\mathcal{J}'|_{d\downarrow}^- \models \mathcal{T}$ for all $d \in \Delta^{\mathcal{J}'} \setminus \{\rho_{\mathcal{J}'}\}$. But if this is not the case, then the subtree rooted at d would have been removed during the construction of $\mathcal{J}'$ from $\mathcal{I} \times_\rho \mathcal{J}$ using Rule 3. ❑

If Algorithm 6 terminates, then it obviously returns a TBox $\mathcal{H}$ that is equivalent to the target TBox $\mathcal{T}$. It thus remains to prove that the algorithm terminates after polynomially many steps, which is a consequence of the following lemma.

Lemma 55 *Let $\mathfrak{I}$ be a list computed at some point of an execution of Algorithm 6. Then (i) the length of $\mathfrak{I}$ is bounded by the number of CIs in $\mathcal{T}$ and (ii) each interpretation in each position of $\mathfrak{I}$ is replaced only $|\mathcal{T}| + |\mathcal{T}|^2$ often with a new interpretation.*

The rest of this section is devoted to proving Lemma 55. For easy reference, assume that at each point of the execution of the algorithm, $\mathfrak{I}$ has the form $\mathcal{I}_0, \dots, \mathcal{I}_k$ for some $k \geq 0$. To establish Point (i) of Lemma 55, we closely follow the argument given by Angluin et al. (1992) and show that

(iii) for every $\mathcal{I}_i$, there is a $D_i \sqsubseteq A_i \in \mathcal{T}$ with $\mathcal{I}_i \not\models D_i \sqsubseteq A_i$ and

(iv) if $i \neq j$, then $D_i \sqsubseteq A_i$ and $D_j \sqsubseteq A_j$ are not identical.

In fact, Point (iii) is immediate since whenever a new $\mathcal{I}_i$ is added to $\mathfrak{I}$ in the algorithm, then $\mathcal{I}_i$ is a $\mathcal{T}$-countermodel. To prove Point (iv), we first establish the intermediate Lemma 56 below. For a ditree interpretation $\mathcal{I}$ and a CI $C \sqsubseteq A$, we write $\mathcal{I} \models^\rho C \sqsubseteq A$ if $\rho_\mathcal{I} \notin C^\mathcal{I}$ or $\rho_\mathcal{I} \in A^\mathcal{I}$; that is, the CI $C \sqsubseteq A$ is satisfied at the root of $\mathcal{I}$, but not necessarily at other points in $\mathcal{I}$. It is easy to see that if some interpretation $\mathcal{I}$ is a $\mathcal{T}$-countermodel, then there is $C \sqsubseteq A \in \mathcal{T}$ such that $\mathcal{I} \not\models^\rho C \sqsubseteq A$.

The following lemma shows under which conditions Algorithm 6 replaces an interpretation in the list $\mathfrak{I}$.

Lemma 56 *If the interpretation $\mathcal{I}$ constructed in Line 5 of Algorithm 6 satisfies $\mathcal{I} \not\models^\rho C \sqsubseteq A \in \mathcal{T}$ and $\rho_{\mathcal{I}_j} \in C^{\mathcal{I}_j}$ for some j, then $\mathcal{J} = \mathcal{I}_i$ is replaced with $\mathcal{J}'$ in Line 9 for some $i \leq j$.*

Proof Assume that the interpretation $\mathcal{I}$ constructed in Line 5 of Algorithm 6 satisfies $\mathcal{I} \not\models^\rho C \sqsubseteq A \in \mathcal{T}$ and that there is some j with $\rho_{\mathcal{I}_j} \in C^{\mathcal{I}_j}$. If there is some $i < j$ such that $\mathcal{I}_i \not\to_\rho (\mathcal{I} \times_\rho \mathcal{I}_i)$ and $\mathcal{I} \times_\rho \mathcal{I}_i \not\models \mathcal{T}$, then $\mathcal{J} = \mathcal{I}_{i'}$ will be replaced with $\mathcal{J}'$ in Line 9 for some $i' \leq i$ and we are done. Thus assume that there is no such i. We aim to show that $\mathcal{J} = \mathcal{I}_j$ is replaced with $\mathcal{J}'$ in Line 9. To this end, it suffices to prove that $\mathcal{I}_j \not\to_\rho (\mathcal{I} \times_\rho \mathcal{I}_j)$ and $\mathcal{I} \times_\rho \mathcal{I}_j \not\models \mathcal{T}$. The latter is a consequence of $\mathcal{I} \not\models^\rho C \sqsubseteq A$ and $\rho_{\mathcal{I}_j} \in C^{\mathcal{I}_j}$.

Assume to the contrary of what we have to show that $\mathcal{I}_j \to_\rho (\mathcal{I} \times_\rho \mathcal{I}_j)$. We establish a contradiction against $\mathcal{I} \models \mathcal{H}$ (which holds by construction of $\mathcal{I}$ in the algorithm) by showing that

1. $\mathcal{I} \not\models^\rho C_{\mathcal{I}_j} \sqsubseteq A$ and
2. $C_{\mathcal{I}_j} \sqsubseteq A \in \mathcal{H}$.

For Point 1, $\mathcal{I}_j \to_\rho (\mathcal{I} \times_\rho \mathcal{I}_j)$ and $\rho_{\mathcal{I}_j} \in (C_{\mathcal{I}_j})^{\mathcal{I}_j}$ imply $\rho_{\mathcal{I} \times_\rho \mathcal{I}_j} \in (C_{\mathcal{I}_j})^{\mathcal{I} \times_\rho \mathcal{I}_j}$, which gives $\rho_\mathcal{I} \in (C_{\mathcal{I}_j})^\mathcal{I}$, by Lemma 51. It remains to observe that $\mathcal{I} \not\models^\rho C \sqsubseteq A$ implies $\rho_\mathcal{I} \notin A^\mathcal{I}$.

In view of the construction of $\mathcal{H}$ in the algorithm, Point 2 can be established by showing that $\mathcal{T} \models C_{\mathcal{I}_j} \sqsubseteq A$. Since $C \sqsubseteq A \in \mathcal{T}$, it suffices to prove that $\emptyset \models C_{\mathcal{I}_j} \sqsubseteq C$. This, however, is an immediate consequence of the fact that $\rho_{\mathcal{I}_j} \in C^{\mathcal{I}_j}$ and the definition of $C_{\mathcal{I}_j}$. ❑

Now, Point (iv) above is a consequence of the following.

Lemma 57 *At any time of the algorithm execution, the following condition holds: if $\mathcal{I}_i \not\models^\rho C \sqsubseteq A \in \mathcal{T}$ and $j < i$, then $\rho_{\mathcal{I}_j} \notin C^{\mathcal{I}_j}$.*

Proof We prove the invariant formulated in Lemma 57 by induction on the number of iterations of the while loop. Clearly, the invariant is satisfied before the loop is entered. We now consider the two places where $\mathfrak{I}$ is modified, that is, Line 9 and Line 11, starting with the latter.

In Line 11, $\mathcal{I}$ is appended to $\mathfrak{I}$. Assume that $\mathcal{I} \not\models^\rho C \sqsubseteq A \in \mathcal{T}$. We have to show that, before $\mathcal{I}$ was added to $\mathfrak{I}$, there was no $\mathcal{I}_i \in \mathfrak{I}$ with $\rho_{\mathcal{I}_i} \in C^{\mathcal{I}_i}$. This, however, is immediate by Lemma 56.

Now assume that $\mathcal{J}$ was replaced in Line 9 with $\mathcal{J}'$. We have to show two properties:

1. If $\mathcal{J}' = \mathcal{I}_i \not\models^\rho C \sqsubseteq A \in \mathcal{T}$ and $j < i$, then $\rho_{\mathcal{I}_j} \notin C^{\mathcal{I}_j}$.

   Assume to the contrary that $\rho_{\mathcal{I}_j} \in C^{\mathcal{I}_j}$. Since $\mathcal{J}'$ is obtained from $\mathcal{I} \times \mathcal{J}$ by removing subtrees (see Lemma 54), $\mathcal{J}' \not\models^\rho C \sqsubseteq A$ implies $\mathcal{I} \times \mathcal{J} \not\models^\rho C \sqsubseteq A$. Consequently, $\mathcal{I} \not\models^\rho C \sqsubseteq A$ or $\mathcal{J} \not\models^\rho C \sqsubseteq A$. The former and $\rho_{\mathcal{I}_j} \in C^{\mathcal{I}_j}$ yields $i \leq j$ by Lemma 56, in contradiction to $j < i$. In the latter case, since $\mathcal{I}_i = \mathcal{J}$ before the replacement of $\mathcal{J}$ with $\mathcal{J}'$, we have a contradiction against the induction hypothesis.

2. If $\mathcal{J}' = \mathcal{I}_j$ and $\mathcal{I}_i \not\models^\rho C \sqsubseteq A \in \mathcal{T}$ with $i > j$, then $\rho_{\mathcal{I}_j} \notin C^{\mathcal{I}_j}$.

   Assume to the contrary that $\rho_{\mathcal{I}_j} \in C^{\mathcal{I}_j}$. Since $\mathcal{J}'$ is obtained from $\mathcal{I} \times_\rho \mathcal{J}$ by removing subtrees, we then have $\rho_{\mathcal{I} \times_\rho \mathcal{J}} \in C^{\mathcal{I} \times_\rho \mathcal{J}}$, thus $\rho_\mathcal{J} \in C^\mathcal{J}$. Since $\mathcal{I}_j = \mathcal{J}$ before the replacement of $\mathcal{J}$ with $\mathcal{J}'$, we have a contradiction against the induction hypothesis.

❑

We now turn towards proving Point (ii) of Lemma 55. It is a consequence of Lemma 59 below.

Lemma 58 *If $\mathcal{I}$ is an essential $\mathcal{T}$-countermodel, then $|\Delta^{\mathcal{I}}| \leq |\mathcal{T}|$.*

Proof Let $\mathcal{I}$ be an essential $\mathcal{T}$-countermodel. Then $\mathcal{I} \not\models \mathcal{T}$, but $\mathcal{I}|_\rho^- \models \mathcal{T}$. It follows that there is a $C \sqsubseteq A \in \mathcal{T}$ such that $\rho_{\mathcal{I}} \in C^{\mathcal{I}} \setminus A^{\mathcal{I}}$. By Lemma 8, there is a homomorphism h from $\mathcal{I}_C$ to $\mathcal{I}$ mapping $\rho_{\mathcal{I}_C}$ to $\rho_{\mathcal{I}}$. We show that $|\Delta^{\mathcal{I}}| \leq |C|$, from which $|\Delta^{\mathcal{I}}| \leq |\mathcal{T}|$ follows. It suffices to show that h is surjective. Assume that this is not the case and let $d \in \Delta^{\mathcal{I}}$ be outside the range of h. Then h is a homomorphism from $\mathcal{I}_C$ to $\mathcal{J} := \mathcal{I}|_{d\downarrow}^-$. Therefore, $\rho^{\mathcal{J}} \in C^{\mathcal{J}}$ by Lemma 8, which implies $\mathcal{J} \not\models C \sqsubseteq A$. But $\mathcal{J} \not\models C \sqsubseteq A$ contradicts the assumption that $\mathcal{I}$ is an essential $\mathcal{T}$-countermodel as it violates Condition 2 of being an essential $\mathcal{T}$-countermodel. ❑

Lemma 59 *Let $\mathcal{I}_0, \ldots, \mathcal{I}_n$ be a list of interpretations such that $\mathcal{I}_{i+1}$ replaces $\mathcal{I}_i$ in Line 9 for all $i < n$. Then $n \leq |\mathcal{T}| + |\mathcal{T}|^2$.*

Proof Let $\mathcal{I}$ and $\mathcal{J}$ be ditree interpretations. We set $\mathcal{I} \leq_\rho \mathcal{J}$ if $\rho_{\mathcal{I}} \in A^{\mathcal{I}}$ implies $\rho_{\mathcal{J}} \in A^{\mathcal{J}}$ for all concept names A. We first show that for every $i < n$ either

(a) $\mathcal{I}_i \not\leq_\rho \mathcal{I}_{i+1}$ or

(b) $\mathcal{I}_{i+1} \to_\rho \mathcal{I}_i$ via a surjective homomorphism.

For a proof by contradiction assume that there is $i < n$ such that neither (a) nor (b) holds. Since $\mathcal{I}_{i+1}$ is obtained from some $\mathcal{I} \times_\rho \mathcal{I}_i$ by removing subtrees and $(\mathcal{I} \times_\rho \mathcal{I}_i) \to_\rho \mathcal{I}_i$ we obtain that $\mathcal{I}_{i+1} \to_\rho \mathcal{I}_i$. Since $\mathcal{I}_{i+1}$ is an essential $\mathcal{T}$-countermodel, there is a $C \sqsubseteq A \in \mathcal{T}$ such that $\mathcal{I}_{i+1} \not\models^\rho C \sqsubseteq A$. Let $\mathcal{J}$ be the subinterpretation of $\mathcal{I}_i$ determined by the range of the homomorphism h from $\mathcal{I}_{i+1}$ to $\mathcal{I}_i$ mapping $\rho_{\mathcal{I}_{i+1}}$ to $\rho_{\mathcal{I}_i}$. By Lemma 8, $\rho_{\mathcal{J}} \in C^{\mathcal{J}}$ and so, since $\rho_{\mathcal{J}} \notin A^{\mathcal{J}}$ because (a) does not hold, $\mathcal{J} \not\models^\rho C \sqsubseteq A$. $\mathcal{I}_i$ is an essential $\mathcal{T}$-countermodel and so $\mathcal{J} = \mathcal{I}$. But then h is surjective and we have derived a contradiction.

In addition to the property stated above, we also have for all $i < n$:

(c) $\mathcal{I}_{i+1} \leq_\rho \mathcal{I}_i$ and

(d) $\mathcal{I}_i \not\to_\rho \mathcal{I}_{i+1}$.

It follows that for any $i < n$ with $\mathcal{I}_i \leq_\rho \mathcal{I}_{i+1}$ either $|\Delta^{\mathcal{I}_i}| < |\Delta^{\mathcal{I}_{i+1}}|$ or $|A^{\mathcal{I}_i}| < |A^{\mathcal{I}_{i+1}}|$ for some concept name A. By Lemma 58 we have $|\Delta^{\mathcal{I}_i}| \leq |\mathcal{T}|$ for all $i \leq n$. Hence $k - j \leq |\mathcal{T}|^2$ for any subsequence $\mathcal{I}_j \ldots, \mathcal{I}_k$ of $\mathcal{I}_0, \ldots, \mathcal{I}_n$ with $\mathcal{I}_i \leq_\rho \mathcal{I}_{i+1}$ for all $j \leq i < k$. It follows that $n \leq |\mathcal{T}| + |\mathcal{T}|^2$. ❑

We have thus established the main result of this section. Note that we obtain a polynomial time learning algorithm since checking $\mathcal{T} \models \alpha$ is in polynomial times for $\mathcal{EL}$ TBoxes $\mathcal{T}$ and $\mathcal{EL}$ CIs α (as discussed in Section 2).

Theorem 60 *$\mathcal{EL}_{\mathsf{lhs}}$ TBoxes are polynomial time learnable using membership and equivalence queries.*

The following example shows that Algorithm 6 does not terminate in polynomial time if in Line 5 it does not transform the given counterexample into an essential $\mathcal{T}$-countermodel.

Example 61 Assume that Line 5 of Algorithm 6 does not modify the counterexample $C \sqsubseteq A$ given in Line 4 if the second condition for essential $\mathcal{T}$-countermodels ($\mathcal{I}_C|_{d\downarrow}^- \models \mathcal{T}$ for all $d \in \Delta^{\mathcal{I}_C} \setminus \{\rho_{\mathcal{I}_C}\}$) is satisfied but the first condition ($\mathcal{I}_C|_\rho^- \models \mathcal{T}$) does not hold. Then for the target TBox $\mathcal{T} = \{\exists r.A \sqsubseteq A\}$ the oracle can return the infinite sequence of positive counterexamples $\exists r^n.A \sqsubseteq A$, with n a prime number. In fact, Algorithm 6 would simply construct the list $\mathfrak{I}$ of interpretations $\mathcal{I}_{\exists r^n.A}$, n a prime number, and would not terminate. To show this observe that Algorithm 6 would never replace a CI in the list $\mathfrak{I}$ by another CI since $\mathcal{I}_{\exists r^n.A} \times_\rho \mathcal{I}_{\exists r^{n+m}.A} = \mathcal{I}_{\exists r^n.\top}$ and $\mathcal{I}_{\exists r^n.\top} \models \mathcal{T}$.

Now assume that Line 5 of Algorithm 6 does not modify the counterexample $C \sqsubseteq A$ given in Line 4 if the first condition for essential $\mathcal{T}$-countermodels is satisfied but the second condition does not hold. Let $\mathcal{T}$ be a TBox containing $\exists r.A \sqsubseteq A$ and some CIs containing the concept names B_1 and B_2, say, for simplicity, $B_1 \sqsubseteq B_1$ and $B_2 \sqsubseteq B_2$. Let $\varphi^1 = \exists r.(B_1 \sqcap B_2)$ and $\varphi^{n+1} = \exists r.(\varphi^n \sqcap B_1 \sqcap B_2)$. Then the oracle can return n positive counterexamples $\exists r.A \sqcap C_i \sqsubseteq A$, where the tree T_{C_i} corresponding to C_i is the result of identifying the i-th node of the tree T_{φ^i} corresponding to φ_i with the root of the tree corresponding to $\exists r.(B_1 \sqcap \varphi^n) \sqcap \exists r.(B_2 \sqcap \varphi^n)$. Note that the product of $\mathcal{I}_{C_1}, \ldots, \mathcal{I}_{C_n}$ is an interpretation with $O(2^n)$ elements. Then, at the n-th iteration, Algorithm 6 computes an interpretation of exponential size in n.

6. Limits of Polynomial Learnability

The main result of this section is that $\mathcal{EL}$ TBoxes are not polynomial query learnable using membership and equivalence queries. We also show that DL-Lite$_{\mathcal{R}}^{\exists}$ TBoxes are not polynomial query learnable using membership or equivalence queries alone. The latter result also holds for $\mathcal{EL}_{\mathsf{lhs}}$ TBoxes. In this case, however, it follows already from the fact that propositional Horn logic is not polynomial query learnable from entailments using membership or equivalence queries alone (Frazier and Pitt, 1993; Angluin et al., 1992; Angluin, 1987a).

We start by proving the non-polynomial query learnability result for $\mathcal{EL}$ TBoxes. On our way, we also prove non-polynomial query learnability of DL-Lite$_{\mathcal{R}}^{\exists}$ TBoxes using membership queries only. Our proof shows that even acyclic $\mathcal{EL}$ TBoxes are not polynomial query learnable and, in fact, heavily relies on the additional properties of acyclic TBoxes. Recall that an $\mathcal{EL}$ TBox is called *acyclic* if it satisfies the following conditions (Baader et al., 2017; Konev et al., 2012):

- all CIs and CEs are of the form $A \sqsubseteq C$ or $A \equiv C$, where A is a concept name;
- no concept name occurs more than once on the left-hand side of a CI;
- there are no cyclic definitions: there is no sequence $\alpha_0, \ldots, \alpha_n$ of CIs such that the concept name on the left-hand side of α_0 occurs in α_n and the concept name on the left-hand side of α_{i+1} occurs in the right-hand side of α_i for all $i < n$.

Our non-polynomial query learnability proof is inspired by Angluin's lower bound for the following abstract learning problem (Angluin, 1987b): a learner aims to identify one of N distinct sets $L_1, \dots, L_N$ which have the property that there exists a set $L_\cap$ for which $L_i \cap L_j = L_\cap$, for any $i \neq j$. It is assumed that $L_\cap$ is not a valid argument to an equivalence query. The learner can pose membership queries "$x \in L$?" and equivalence queries "$H = L$?". Then in the worst case it takes at least $N-1$ membership and equivalence queries to exactly identify a hypothesis L_i from $L_1, \dots, L_N$. The proof proceeds as follows. At every stage of computation, the oracle (which here should be viewed as an adversary) maintains a set of hypotheses S, which the learner is not able to distinguish based on the answers given so far. Initially, $S = \{L_1, \dots, L_N\}$. When the learner asks a membership query x, the oracle returns 'Yes' if $x \in L_\cap$ and 'No' otherwise. In the latter case, the (unique) L_i such that $x \in L_i$ is removed from S. When the learner asks an equivalence query H, the oracle returns 'No' and a counterexample $x \in L_\cap \oplus H$ (the symmetric difference of $L_\cap$ and H). This always exists as $L_\cap$ is not a valid query. If the counterexample x is not a member of $L_\cap$, (at most one) $L_i \in S$ such that $x \in L_i$ is eliminated from S. In the worst case, the learner has to reduce the cardinality of S to one to exactly identify a hypothesis, which takes $N-1$ queries.

Similarly to the method outlined above, in our proof we maintain a set of acyclic $\mathcal{EL}$ TBoxes whose members the learning algorithm is not able to distinguish based on the answers obtained so far. For didactic purposes, we first present a set of acyclic TBoxes $S_N = \{\mathcal{T}_1, \dots, \mathcal{T}_N\}$, where N is superpolynomial in the size of every TBox $\mathcal{T}_i$, for which the oracle can respond to membership queries in the way described above but which is polynomial time learnable when equivalence queries are also allowed. We then show how the TBoxes can be modified to obtain a family of acyclic TBoxes that is not polynomial query learnable using membership and equivalence queries.

To present the TBoxes in S_N, fix two role names r and s. We use the following abbreviation. For any sequence $\sigma = \sigma^1\sigma^2 \dots \sigma^n \in \{r, s\}^n$, the expression $\exists \boldsymbol{\sigma}.C$ stands for $\exists \sigma^1.\exists \sigma^2 \dots \exists \sigma^n.C$. Then for every such sequence $\boldsymbol{\sigma}$, of which there are $N = 2^n$ many, consider the acyclic $\mathcal{EL}$ TBox $\mathcal{T}_\sigma$ defined as

$$\begin{array}{rcl} \mathcal{T}_{\boldsymbol{\sigma}} & = & \{A \sqsubseteq \exists \boldsymbol{\sigma}.M \sqcap X_0\} \cup \mathcal{T}_0 \text{ with} \\ \mathcal{T}_0 & = & \{X_i \sqsubseteq \exists r.X_{i+1} \sqcap \exists s.X_{i+1} \mid 0 \leq i < n\}. \end{array}$$

Observe that the canonical model $\mathcal{I}_{X_0,\mathcal{T}_0}$ of X_0 and $\mathcal{T}_0$ consists of a full binary tree whose edges are labelled with the role names r and s and with X_0 at the root ρ_{X_0}, X_1 at level 1, and so on. In the canonical model $\mathcal{I}_{A,\mathcal{T}_{\boldsymbol{\sigma}}}$ of A and $\mathcal{T}_{\boldsymbol{\sigma}}$, the root is labelled by A and X_0 and, *in addition to* the binary tree, there is a path given by the sequence $\boldsymbol{\sigma}$ whose endpoint is marked by the concept name M.

One can use Angluin's strategy to show that TBoxes from the set S_N of all such TBoxes $\mathcal{T}_{\boldsymbol{\sigma}}$ cannot be learned using polynomially many polynomial size membership queries only: notice that for no sequence $\boldsymbol{\sigma}' \neq \boldsymbol{\sigma}$ of length n, we have $\mathcal{T}_{\boldsymbol{\sigma}} \models A \sqsubseteq \exists \boldsymbol{\sigma}'.M$. Thus a membership query of the form $A \sqsubseteq \exists \boldsymbol{\sigma}.M$ eliminates at most one TBox from the set of TBoxes that the learner cannot distinguish. This observation can be generalised to arbitrary membership queries $C \sqsubseteq D$ in $\mathcal{EL}$; however, we instead observe that the TBoxes $\mathcal{T}_{\boldsymbol{\sigma}}$ are formulated in DL-Lite$^\exists_\mathcal{R}$ and prove a stronger result. The proof, given in the appendix, uses the canonical model construction introduced in Section 2.

Lemma 62 *For every DL-Lite$^\exists_\mathcal{R}$ CI $B \sqsubseteq D$ over the signature of $\mathcal{T}_{\boldsymbol{\sigma}}$,*

- *either $\mathcal{T}_{\boldsymbol{\sigma}} \models B \sqsubseteq D$ for every $\mathcal{T}_{\boldsymbol{\sigma}} \in S_N$*
- *or there is at most one $\mathcal{T}_{\boldsymbol{\sigma}} \in S_N$ such that $\mathcal{T}_{\boldsymbol{\sigma}} \models B \sqsubseteq D$.*

The argument outlined above immediately gives us the following side result.

Theorem 63 *DL-Lite$^\exists_\mathcal{R}$ TBoxes (even without inverse roles) are not polynomial query learnable using only membership queries.*

We return now to our proof that $\mathcal{EL}$ TBoxes are not polynomial query learnable using both membership and equivalence queries. Notice that the set of TBoxes S_N is not suitable as a single equivalence query is sufficient to learn any TBox from S_N in two steps: given the equivalence query $\{A \sqsubseteq X_0\} \cup \mathcal{T}_0$, the oracle has no other option but to reveal the target TBox $\mathcal{T}_{\boldsymbol{\sigma}}$ as $A \sqsubseteq \exists \boldsymbol{\sigma}.M$ can be found 'inside' every counterexample.

Our strategy to rule out equivalence queries with the 'intersection TBox' is to modify $\mathcal{T}_1, \dots, \mathcal{T}_N$ in such a way that although a TBox $\mathcal{T}_\cap$ axiomatising the intersection over the set of consequences of each $\mathcal{T}_i$, $i \leq N$, exists, its size is superpolynomial and so it cannot be used as an equivalence query by a polynomial query learning algorithm.

For every $n > 0$ and every n-tuple $L = (\boldsymbol{\sigma}_1, \dots, \boldsymbol{\sigma}_n)$, where every $\boldsymbol{\sigma}_i$ is a role sequence of length n as above, we define an acyclic $\mathcal{EL}$ TBox $\mathcal{T}_L$ as the union of $\mathcal{T}_0$ and the following CIs and CEs:[5]

$$\begin{array}{ccc} A_1 \sqsubseteq \exists \boldsymbol{\sigma}_1.M \sqcap X_0 & & A_n \sqsubseteq \exists \boldsymbol{\sigma}_n.M \sqcap X_0 \\ B_1 \sqsubseteq \exists \boldsymbol{\sigma}_1.M \sqcap X_0 & \cdots & B_n \sqsubseteq \exists \boldsymbol{\sigma}_n.M \sqcap X_0 \end{array}$$

$$A \equiv X_0 \sqcap \exists \boldsymbol{\sigma}_1.M \sqcap \dots \sqcap \exists \boldsymbol{\sigma}_n.M.$$

Observe that every $\mathcal{T}_L$ contains the TBoxes $\mathcal{T}_{\boldsymbol{\sigma}_i}$, $1 \leq i \leq n$, discussed above with A replaced by any of the three concept names A, A_i, B_i. In addition, *every* $\mathcal{T}_L$ entails, among other CIs, $\bigsqcap_{i=1}^n C_i \sqsubseteq A$, where every C_i is either A_i or B_i. There are 2^n different such CIs, which indicates that every representation of the 'intersection TBox' requires superpolynomially many axioms. It follows from Lemma 67 below that this is indeed the case.

Let $\mathfrak{L}_n$ be a set of n-tuples such that for $1 \leq i \leq n$ and every $L, L' \in \mathfrak{L}_n$ with $L = (\boldsymbol{\sigma}_1, \dots, \boldsymbol{\sigma}_n)$, $L' = (\boldsymbol{\sigma}'_1, \dots, \boldsymbol{\sigma}'_n)$, if $\boldsymbol{\sigma}_i = \boldsymbol{\sigma}'_j$ then $L = L'$ and $i = j$. Then for any sequence $\boldsymbol{\sigma}$ of length n there exists at most one $L \in \mathfrak{L}_n$ and at most one $i \leq n$ such that $\mathcal{T}_L \models A_i \sqsubseteq \exists \boldsymbol{\sigma}.M$ and $\mathcal{T}_L \models B_i \sqsubseteq \exists \boldsymbol{\sigma}.M$. We can choose $\mathfrak{L}_n$ such that there are $N = \lfloor 2^n/n \rfloor$ different tuples in $\mathfrak{L}_n$. Notice that the size of each $\mathcal{T}_L$ with $L \in \mathfrak{L}_n$ is polynomial in n and so N is superpolynomial in the size of each $\mathcal{T}_L$ with $L \in \mathfrak{L}_n$. Let the set of TBoxes that the learner cannot distinguish initially be $S_\mathfrak{L} = \{\mathcal{T}_L \mid L \in \mathfrak{L}_n\}$. We use Σ_n to denote the signature of $\mathcal{T}_L$.

For the proof of non-polynomial query learnability, we show that the oracle has a strategy to answer both membership and equivalence queries without eliminating too many TBoxes from $S_\mathfrak{L}$. We start with the former.

5. In fact, to prove non-polynomial query learnability, it suffices to consider $\exists \boldsymbol{\sigma}_1.M \sqcap \dots \sqcap \exists \boldsymbol{\sigma}_n.M \sqsubseteq A$ in place of the concept equivalence; however, CIs of this form are not allowed in acyclic TBoxes. CIs with a complex left-hand side or concept equivalences are essential for non-polynomial query learnability as any acyclic TBox containing expressions of the form $A \sqsubseteq C$ only is a DL-Lite$^\exists_\mathcal{R}$ TBox and thus polynomially learnable with membership and equivalence queries (Section 3).

Unlike the DL-Lite$^{\exists}_{\mathcal{R}}$ case presented above, membership query can eliminate more than one TBox from $S_{\mathfrak{L}}$. Consider, for example, two TBoxes $\mathcal{T}_L$ and $\mathcal{T}_{L'}$, where $\{L, L'\} \subseteq \mathfrak{L}_n$ with $L = (\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_n)$ and $L' = (\boldsymbol{\sigma}'_1, \ldots, \boldsymbol{\sigma}'_n)$. Then the CI

$$X_0 \sqcap \exists \boldsymbol{\sigma}_1.M \sqcap \exists \boldsymbol{\sigma}'_1.M \sqcap A_2 \sqcap \cdots \sqcap A_n \sqsubseteq A$$

is entailed by both $\mathcal{T}_L$ and $\mathcal{T}_{L'}$ but not by any other $\mathcal{T}_{L''}$ with $L'' \in \mathfrak{L}_n$. We prove, however, that the number of TBoxes eliminated from $S_{\mathfrak{L}}$ by a single membership query can be linearly bounded by the size of the query.

Lemma 64 *For all $\mathcal{EL}$ CIs $C \sqsubseteq D$ over Σ_n:*

- *either $\mathcal{T}_L \models C \sqsubseteq D$ for every $L \in \mathfrak{L}_n$*
- *or the number of $L \in \mathfrak{L}_n$ such that $\mathcal{T}_L \models C \sqsubseteq D$ does not exceed $|C|$.*

The proof of Lemma 64 is technical and is deferred to the appendix. To illustrate our proof method here we consider a particular case that deals with membership queries of the form $C \sqsubseteq \exists \boldsymbol{\sigma}.M$ and is used in the proof of the general case. Both proofs rely on the following lemma from the study of the logical difference between ontologies (Konev et al., 2012). It characterises CIs entailed by acyclic $\mathcal{EL}$ TBoxes.

Lemma 65 (Konev et al. (2012)) *Let $\mathcal{T}$ be an acyclic $\mathcal{EL}$ TBox, r a role name and D an $\mathcal{EL}$ concept expression. Suppose that $\mathcal{T} \models \bigsqcap_{1 \leq i \leq n} A_i \sqcap \bigsqcap_{1 \leq j \leq m} \exists r_j.C_j \sqsubseteq D$, where A_i are concept names for $1 \leq i \leq n$, C_j are $\mathcal{EL}$ concept expressions for $1 \leq j \leq m$, and $m, n \geq 0$. Then the following holds:*

- *if D is a concept name such that $\mathcal{T}$ does not contain any CE $D \equiv C$ for any concept expression C, then there exists A_i, $1 \leq i \leq n$, such that $\mathcal{T} \models A_i \sqsubseteq D$;*
- *if D is of the form $\exists r.D'$ then either (i) there exists A_i, $1 \leq i \leq n$, such that $\mathcal{T} \models A_i \sqsubseteq \exists r.D'$ or (ii) there exists r_j, $1 \leq j \leq m$, such that $r_j = r$ and $\mathcal{T} \models C_j \sqsubseteq D'$.*

The following lemma considers membership queries of the form $C \sqsubseteq \exists \boldsymbol{\sigma}.M$.

Lemma 66 *For any $0 \leq m \leq n$, any sequence of role names $\boldsymbol{\sigma} = \sigma^1 \ldots \sigma^m \in \{r, s\}^m$, and any $\mathcal{EL}$ concept expression C over Σ_n:*

- *either $\mathcal{T}_L \models C \sqsubseteq \exists \boldsymbol{\sigma}.M$ for every $\mathcal{T}_L$ with $L \in \mathfrak{L}_n$;*
- *or there is at most one $\mathcal{T}_L$ such that $\mathcal{T}_L \models C \sqsubseteq \exists \boldsymbol{\sigma}.M$.*

Proof The lemma follows from the following claim.

Claim. Let $L = (\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_n) \in \mathfrak{L}_n$ be such that $\mathcal{T}_L \models C \sqsubseteq \exists \boldsymbol{\sigma}.M$. Then *either* (1) there exists $i \leq n$ such that $\boldsymbol{\sigma} = \boldsymbol{\sigma}_i$ and C is of the form $A \sqcap C'$, $A_i \sqcap C'$ or $B_i \sqcap C'$, for some $\mathcal{EL}$ concept expression C'; *or* (2) we have $\emptyset \models C \sqsubseteq \exists \boldsymbol{\sigma}.M$.

Proof of Claim. We prove the claim by induction on m. If $m = 0$, by Lemma 65, the concept expression C is of the form $Z \sqcap C'$, for some concept name Z and concept expression C' such that $\mathcal{T}_L \models Z \sqsubseteq M$. As $\mathcal{T}_L \models Z \sqsubseteq M$ does not hold for any concept name Z distinct from M, we obtain $Z = M$. Thus, $\emptyset \models C \sqsubseteq M$ and Point (2) follows.
Let $m > 0$. By Lemma 65 we have one of the following two cases:

- C is of the form $X \sqcap C'$, for some concept name X and concept expression C' such that $\mathcal{T}_L \models X \sqsubseteq \exists \boldsymbol{\sigma}.M$. But then there exists $i \leq n$ such that $\boldsymbol{\sigma} = \boldsymbol{\sigma}_i$ and $X \in \{A, A_i, B_i\}$ and Point (1) follows.
- C is of the form $\exists \sigma^1.C' \sqcap C''$, for some concept expressions C' and C'', and $\mathcal{T}_L \models C' \sqsubseteq \exists \sigma^2. \cdots \exists \sigma^m.M$. Notice that the length of the sequence $\sigma^2 \ldots \sigma^n$ is strictly less than n. Thus, by induction hypothesis, $\emptyset \models C' \sqsubseteq \exists \sigma^2. \cdots \exists \sigma^m.M$. But then $\emptyset \models C \sqsubseteq \exists \boldsymbol{\sigma}.M$ and Point (2) follows.

This finishes the proof of the claim. To see that the claim entails the lemma observe that at most one $L \in \mathfrak{L}_n$ can satisfy Point (1). Point (2) entails that $\mathcal{T}_L \models C \sqsubseteq \exists \boldsymbol{\sigma}.M$ for every $\mathcal{T}_L$ with $L \in \mathfrak{L}_n$. ❑

We now show how the oracle can answer equivalence queries, aiming to show that for any polynomial size equivalence query $\mathcal{H}$, the oracle can return a counterexample $C \sqsubseteq D$ such that either (i) $\mathcal{H} \models C \sqsubseteq D$ and $\mathcal{T}_L \models C \sqsubseteq D$ for at most one $L \in \mathfrak{L}_n$ or (ii) $\mathcal{H} \not\models C \sqsubseteq D$ and $\mathcal{T}_L \models C \sqsubseteq D$ for every $L \in \mathfrak{L}_n$. Thus, such a counterexample eliminates at most one $\mathcal{T}_L$ from the set $S_{\mathfrak{L}}$ of TBoxes that the learner cannot distinguish. In addition, however, we have to take extra care of the size of counterexamples as the learning algorithm is allowed to formulate queries polynomial not only in the size of the target TBox but also in the size of the counterexamples returned by the oracle. For instance, if the hypothesis TBox $\mathcal{H}$ contains a CI $C \sqsubseteq D$ which is not entailed by any $\mathcal{T}_L$, one cannot simply return $C \sqsubseteq D$ as a counterexample since the learner will be able to 'pump up' its capacity by asking a sequence of equivalence queries $\mathcal{H}_i = \{C_i \sqsubseteq D_i\}$ such that the size of $C_{i+1} \sqsubseteq D_{i+1}$ is twice the size of $C_i \sqsubseteq D_i$. Then at every stage in a run of the learning algorithm, the query size will be polynomial in the size of the input and the size of the largest counterexample received so far, but exponential size queries will become available to the learner. The following lemma addresses this issue.

Lemma 67 *For any $n > 1$ and any $\mathcal{EL}$ TBox $\mathcal{H}$ in Σ_n with $|\mathcal{H}| < 2^n$, there exists an $\mathcal{EL}$ CI $C \sqsubseteq D$ over Σ_n such that the size of $C \sqsubseteq D$ does not exceed $6n$ and*

- *if $\mathcal{H} \models C \sqsubseteq D$, then $\mathcal{T}_L \models C \sqsubseteq D$ for at most one $L \in \mathfrak{L}_n$;*
- *if $\mathcal{H} \not\models C \sqsubseteq D$, then $\mathcal{T}_L \models C \sqsubseteq \ D$ for every $L \in \mathfrak{L}_n$.*

Proof We define an exponentially large TBox $\mathcal{T}_\cap$ and use it to prove that one can select the required $\mathcal{EL}$ CI $C \sqsubseteq D$ in such a way that either $\mathcal{H} \models C \sqsubseteq D$ and $\mathcal{T}_\cap \not\models C \sqsubseteq D$, or vice versa.

To define $\mathcal{T}_\cap$, denote for any sequence $\boldsymbol{b} = b_1 \dots b_n \in \{0,1\}^n$ by $C_{\boldsymbol{b}}$ the conjunction $\bigsqcap_{i \leq n} C_i$, where $C_i = A_i$ if $b_i = 1$ and $C_i = B_i$ if $b_i = 0$. Then we define

$$\mathcal{T}_\cap = \mathcal{T}_0 \cup \{C_{\boldsymbol{b}} \sqsubseteq A \sqcap X_0 \mid \boldsymbol{b} \in \{0,1\}^n\}.$$

Consider the following cases for $\mathcal{H}$ and $\mathcal{T}_\cap$.

1. Suppose $\mathcal{H} \not\models \mathcal{T}_\cap$. Then there exists a CI $C \sqsubseteq D \in \mathcal{T}_\cap$ such that $\mathcal{H} \not\models C \sqsubseteq D$. Clearly, $C \sqsubseteq D$ is entailed by every $\mathcal{T}_L$, for $L \in \mathfrak{L}_n$, and the size of $C \sqsubseteq D$ does not exceed $6n$. Thus $C \sqsubseteq D$ is as required.

2. Suppose there exist $\boldsymbol{b} \in \{0,1\}^n$ and a concept expression of the form $\exists t.D'$ such that $\mathcal{H} \models C_{\boldsymbol{b}} \sqsubseteq \exists t.D'$ and $\mathcal{T}_0 \not\models C_{\boldsymbol{b}} \sqsubseteq \exists t.D'$. It can be seen (Lemma 73 in the appendix), that there exists a sequence of role names $t_1, \dots, t_l \in \{r,s\}^l$ with $0 \leq l \leq n+1$ and $Y \in \{\top\} \cup \mathsf{N_C}$ such that $\emptyset \models \exists t.D' \sqsubseteq \exists t_1. \cdots \exists t_l.Y$. Thus, $\mathcal{H} \models C_{\boldsymbol{b}} \sqsubseteq \exists t_1. \cdots \exists t_l.Y$ and $\mathcal{T}_0 \not\models X_0 \sqsubseteq \exists t_1. \cdots \exists t_l.Y$. We show that the inclusion $C_{\boldsymbol{b}} \sqsubseteq \exists t_1. \cdots \exists t_l.Y$ is as required. Clearly, the size of $C_{\boldsymbol{b}} \sqsubseteq \exists t_1. \cdots \exists t_l.Y$ does not exceed $6n$. It remains to prove that $\mathcal{T}_L \models C_{\boldsymbol{b}} \sqsubseteq \exists t_1 \cdots \exists t_l.Y$ for at most one $L \in \mathfrak{L}_n$.

   Suppose there exists $L \in \mathfrak{L}_n$ such that $\mathcal{T}_L \models C_{\boldsymbol{b}} \sqsubseteq \exists t_1. \cdots \exists t_l.Y$. By Lemma 65, there exists A_j or B_j such that $\mathcal{T}_L \models A_j \sqsubseteq \exists t_1. \cdots \exists t_l.Y$ or $\mathcal{T}_L \models B_j \sqsubseteq \exists t_1. \cdots \exists t_l.Y$, respectively. As $\mathcal{T}_0 \not\models X_0 \sqsubseteq \exists t_1. \cdots \exists t_l.Y$ it is easy to see that $l = n$, $t_1 t_2 \dots t_n = \boldsymbol{\sigma}_j$, and $Y = M$ follow. As $\mathcal{T}_{L'} \not\models C_{\boldsymbol{b}} \sqsubseteq \exists \boldsymbol{\sigma}_j.M$ for any $L' \in \mathfrak{L}_n$ such that $L' \neq L$, it follows that $\mathcal{T}_L \models C_{\boldsymbol{b}} \sqsubseteq \exists t_1 \cdots \exists t_l.Y$ for at most one $L \in \mathfrak{L}_n$

3. Finally, suppose that neither Case 1 nor 2 above apply. Then $\mathcal{H} \models \mathcal{T}_\cap$ and for every $\boldsymbol{b} \in \{0,1\}^n$ and every $\mathcal{EL}$ concept expression over Σ_n of the form $\exists t.D'$: if $\mathcal{H} \models C_{\boldsymbol{b}} \sqsubseteq \exists t.D'$ then $\mathcal{T}_0 \models X_0 \sqsubseteq \exists t.D'$. We show that unless there exists a CI $C \sqsubseteq D$ satisfying the conditions of the lemma, $\mathcal{H}$ contains at least 2^n different CIs (and thus derive a contradiction).

   Fix some $\boldsymbol{b} = b_1 \dots b_n \in \{0,1\}^n$. From $\mathcal{H} \models \mathcal{T}_\cap$ we obtain $\mathcal{H} \models C_{\boldsymbol{b}} \sqsubseteq A$. Then there must exist at least one CI $C \sqsubseteq A \sqcap D \in \mathcal{H}$ such that $\mathcal{H} \models C_{\boldsymbol{b}} \sqsubseteq C$ and $\emptyset \not\models C \sqsubseteq A$. Let $C = Z_1 \sqcap \dots \sqcap Z_m \sqcap \exists t_1.C'_1 \sqcap \dots \sqcap \exists t_l.C'_l$, where $Z_1, \dots, Z_m$ are different concept names. As $\mathcal{H} \models C_{\boldsymbol{b}} \sqsubseteq \exists t_j.C'_j$ we have $\mathcal{T}_0 \models X_0 \sqsubseteq \exists t_j.C'_j$, for $j = 1, \dots l$. As $\mathcal{H} \models \mathcal{T}_\cap$ we have $\mathcal{H} \models X_0 \sqsubseteq \exists t_j.C'_j$, for $j = 1, \dots l$. So $\mathcal{H} \models Z_1 \sqcap \dots \sqcap Z_m \sqcap X_0 \sqsubseteq A$.

   - Suppose there exists i such that there is no $Z_j \in \{A_i, B_i\}$. Then we have $\mathcal{T}_L \not\models Z_1 \sqcap \dots \sqcap Z_m \sqcap X_0 \sqsubseteq A$, for any $L \in \mathfrak{L}_n$. Notice that $Z_1 \sqcap \dots \sqcap Z_m$ contains at most all concepts names in Σ_n, except A_i, B_i. Thus, the size of $Z_1 \sqcap \dots \sqcap Z_m \sqcap X_0 \sqsubseteq A$ does not exceed $6n$, and $Z_1 \sqcap \dots \sqcap Z_m \sqcap X_0 \sqsubseteq A$ is as required.
   - Assume that $Z_0 \sqcap \dots \sqcap Z_m \sqcap X_0$ contains a conjunct B_i such that $b_i \neq 0$. Then $\mathcal{H} \models C_{\boldsymbol{b}} \sqsubseteq B_i$ and there is no $L \in \mathfrak{L}_n$ such that $\mathcal{T}_L \models C_{\boldsymbol{b}} \sqsubseteq B_i$. The size of $C_{\boldsymbol{b}} \sqsubseteq B_i$ does not exceed $6n$, so $C_{\boldsymbol{b}} \sqsubseteq B_i$ is as required.
   - Assume that $Z_0 \sqcap \dots \sqcap Z_m \sqcap X_0$ contains a conjunct A_i such that $b_i \neq 1$. Then $\mathcal{H} \models C_{\boldsymbol{b}} \sqsubseteq A_i$ and there is no $L \in \mathfrak{L}_n$ such that $\mathcal{T}_L \models C_{\boldsymbol{b}} \sqsubseteq A_i$. The size of $C_{\boldsymbol{b}} \sqsubseteq A_i$ does not exceed $6n$, so $C_{\boldsymbol{b}} \sqsubseteq A_i$ is as required.
   - If none of the above applies, then $Z_1 \sqcap \dots \sqcap Z_m \sqcap X_0$ contains exactly the A_i with $b_i = 1$ and exactly the B_i with $b_i = 0$.

   This argument applies to arbitrary $\boldsymbol{b} \in \{0,1\}^n$. Thus, if there exists no CI $C \sqsubseteq D$ satisfying the conditions of the lemma then, by the final case, $\mathcal{H}$ contains at least 2^n CIs.

❑

Now we have all the ingredients to prove that $\mathcal{EL}$ TBoxes are not polynomial query learnable using membership and equivalence queries.

Theorem 68 *$\mathcal{EL}$ TBoxes are not polynomial query learnable using membership and equivalence queries.*

Proof Assume that TBoxes are polynomial query learnable. Then there exists a learning algorithm whose query complexity (the sum of the sizes of the inputs to membership and equivalence queries made by the algorithm up to a computation step) is bounded at any stage by a polynomial $p(n,m)$. Choose n such that $\lfloor 2^n/n \rfloor > (p(n,6n))^2$ and let $S_{\mathfrak{L}} = \{\mathcal{T}_L \mid L \in \mathfrak{L}_n\}$. We follow Angluin's strategy of letting the oracle remove TBoxes from $S_{\mathfrak{L}}$ in such a way that the learner cannot distinguish between any of the remaining TBoxes. Given a membership query $C \sqsubseteq D$, if $\mathcal{T}_L \models C \sqsubseteq D$ for every $L \in \mathfrak{L}_n$, then the answer is 'yes'; otherwise the answer is 'no' and all $\mathcal{T}_L$ with $\mathcal{T}_L \models C \sqsubseteq D$ are removed from $S_{\mathfrak{L}}$ (by Lemma 64, there are at most $|C|$ such TBoxes). Given an equivalence query $\mathcal{H}$, the answer is 'no', a counterexample $C \sqsubseteq D$ guaranteed by Lemma 67 is produced, and (at most one) $\mathcal{T}_L$ such that $\mathcal{T}_L \models C \sqsubseteq D$ is removed from $S_{\mathfrak{L}}$.

As all counterexamples produced are smaller than $6n$, the overall query complexity of the algorithm is bounded by $p(n,6n)$. Hence, the learner asks no more than $p(n,6n)$ queries and the size of every query does not exceed $p(n,6n)$. By Lemmas 64 and 67, at most $(p(n,6n))^2$ TBoxes are removed from $S_{\mathfrak{L}}$ during the run of the algorithm. But then, the algorithm cannot distinguish between any remaining TBoxes and we have derived a contradiction. ❑

We conclude this section by showing that DL-Lite$^{\exists}_{\mathcal{R}}$ TBoxes cannot be learned using polynomially many polynomial size equivalence queries only. We use the following result on non-polynomial query learnability of monotone DNF formulas, that is, DNF formulas that do not use negation, using equivalence queries due to Angluin (1990). Here, equivalence queries take a hypothesis ψ in the form of a monotone DNF formula and return as a counterexample either a truth assignment that satisfies ψ but not the target formula ϕ or vice versa. Let $M(n,t,s)$ denote the set of all monotone DNF formulas whose variables are $x_1, \dots, x_n$, that have exactly t conjunctions, and where each conjunction contains exactly s variables.

Theorem 69 (Angluin (1990)) *For any polynomial $q(\cdot)$ there exist constants t_0 and s_0 and a strategy [6] for the oracle $\mathfrak{O}$ to answer equivalence queries posed by a learning*

6. The existence of this strategy is a direct consequence of Theorem 8 by Angluin (1990), which states that the class of DNF formulae has the *approximate fingerprint* property, and the proof of Theorem 1

JMLR 18(201):1-63, 2018

algorithm in such a way that for sufficiently large n any learning algorithm that asks at most $q(n)$ equivalence queries, each bounded in size by $q(n)$, cannot exactly identify elements of $M(n, t_0, s_0)$.

To employ Theorem 69, we associate with every monotone DNF formula

$$\phi = \bigvee_{i=1}^{t} (x_1^i \wedge \cdots \wedge x_{s^i}^i),$$

where $\{x_1^i, \ldots, x_{s^i}^i\} \subseteq \{x_1, \ldots, x_n\}$, a DL-Lite$_{\mathcal{R}}^{\exists}$ TBox $\mathcal{T}_\phi$ as follows. With each conjunct $x_1^i \wedge \cdots \wedge x_{s^i}^i$ we associate a concept expression

$$C_i := \exists \rho_1^i . \exists \rho_2^i . \ldots \exists \rho_n^i . \top,$$

where $\rho_j^i = r$ if x_j occurs in $x_1^i \wedge \cdots \wedge x_{s^i}^i$ and $\rho_j^i = \bar{r}$ otherwise (r and $\bar{r}$ are role names). Let A be a concept name and set

$$\mathcal{T}_\phi = \{A \sqsubseteq \textstyle\prod_{i=1}^{t} C_i, \quad \bar{r} \sqsubseteq r\}.$$

For example, for $n = 4$ and $\phi = (x_1 \wedge x_4) \vee x_2$ we have

$$\mathcal{T}_\phi = \{A \sqsubseteq \exists r . \exists \bar{r} . \exists \bar{r} . \exists r . \top,\ A \sqsubseteq \exists \bar{r} . \exists r . \exists \bar{r} . \exists \bar{r} . \top,\ \bar{r} \sqsubseteq r\}.$$

We say that a TBox $\mathcal{T}$ *has a DNF-representation for n* if it is obtained by the translation of a monotone DNF-formula with n variables; that is, if $\mathcal{T}$ is of the following form, for some $\Gamma \subseteq \{r, \bar{r}\}^n$:

$$\{A \sqsubseteq \prod_{\rho_1 \cdots \rho_n \in \Gamma} \exists \rho_1 . \exists \rho_2 . \ldots \exists \rho_n . \top, \quad \bar{r} \sqsubseteq r\}.$$

A truth assignment I (for the variables $x_1 \ldots, x_n$) also corresponds to a concept expression

$$C_I := \exists \rho_1^i . \exists \rho_2^i . \ldots \exists \rho_n^i . \top,$$

where $\rho_j^i = r$ if I makes x_j true and $\rho_j^i = \bar{r}$ otherwise. Then

$$I \models \phi \text{ if, and only if, } \mathcal{T}_\phi \models A \sqsubseteq C_I$$

holds for all truth assignments I.

Note that $\bar{r}$ represents that a variable is false and r that a variable is true. Thus, the RI $\bar{r} \sqsubseteq r$ captures the monotonicity of the DNF formulas considered. For any fixed values n, s and t, we set

$$T(n, t, s) = \{\mathcal{T}_\phi \mid \phi \in M(n, t, s)\}.$$

Note that the TBoxes in $T(n, t, s)$ are exactly those TBoxes that have a DNF-representation for n and satisfy additionally the conditions that the DNF represented by $\mathcal{T}_\phi$ has exactly t conjunctions each conjunction of which has exactly s variables.

by Angluin (1990), where such a strategy is explicitly constructed for any class having approximate fingerprints.

We describe now the strategy for the oracle $\mathfrak{O}'$ to answer equivalence queries so that no learning algorithm is able to exactly identify members of $T(n, t, s)$ based on the answers to polynomially many equivalence queries of polynomial size. If the TBox in the equivalence query is 'obviously' not within the class $T(n, t, s)$, then we will explicitly produce a counterexample that the oracle can return. If, on the other hand, the TBox $\mathcal{H}$ from the equivalence query is 'similar' to TBoxes that have a DNF-representation for n, then we approximate $\mathcal{H}$ by a TBox $\mathcal{H}'$ that has a DNF-representation for n and return the counterexample $A \sqsubseteq C_I$ corresponding to the truth assignment I that the oracle $\mathfrak{O}$ from Theorem 69 would return when given ψ.

In detail the strategy is as follows. Assume q is the given polynomial in Theorem 69 and that t_0, s_0 and the strategy of the oracle $\mathfrak{O}$ are chosen so that for sufficiently large n no learning algorithm for DNF formulas that asks at most $q(n)$ equivalence queries, each bounded in size by $q(n)$, can distinguish all members of $M(n, t_0, s_0)$. Choose a sufficiently large n. Let $\mathcal{H}$ be an equivalence TBox query issued by a learning algorithm. Then $\mathfrak{O}'$ does the following:

1. If $\mathcal{H}$ entails some $A \sqsubseteq \exists \rho_1 . \exists \rho_2 . \ldots \exists \rho_{n+1} . \top$ with $\rho_i \in \{r, \bar{r}\}$ for $1 \leq i \leq n+1$, then return this CI as a negative counterexample;

2. If $\mathcal{H}$ entails some $\exists \rho_1 . \top \sqsubseteq \exists \rho_2 . \top$ such that $\{\rho_1, \rho_2\} \subseteq \{r, \bar{r}, r^-, \bar{r}^-\}$ and $\{\bar{r} \sqsubseteq r\} \not\models \exists \rho_1 . \top \sqsubseteq \exists \rho_2 . \top$, then return this CI as a negative counterexample;

3. If $\mathcal{H} \models \exists \rho_1 . \top \sqsubseteq \exists \rho_2 . \exists \rho_3 . \top$ such that $\{\rho_1, \rho_2, \rho_3\} \subseteq \{r, \bar{r}\}$, then return this CI as a negative counterexample;

4. If there exists no $\rho_1, \ldots, \rho_n \in \{r, \bar{r}\}^n$ such that $\mathcal{H} \models A \sqsubseteq \exists \rho_1 . \cdots \exists \rho_n . \top$ then return $A \sqsubseteq \underbrace{\exists r \cdots \exists r}_{n} . \top$ as a positive counterexample.

5. Suppose now that none of the above applies. We say that a sequence $\rho_1, \ldots, \rho_n \in \{r, \bar{r}\}^n$ is *r-minimal for $\mathcal{H}$* if $\mathcal{H} \models A \sqsubseteq \exists \rho_1 . \cdots \exists \rho_n . \top$ and whenever $\rho_i = r$, for $1 \leq i \leq n$, we have $\mathcal{H} \not\models \exists \rho_1 . \cdots \exists \rho_{i-1} . \exists \bar{r} . \exists \rho_{i+1} . \cdots \exists \rho_n . \top$. We obtain a TBox $\mathcal{H}'$ with a DNF representation by setting

$$\mathcal{H}' = \{A \sqsubseteq \prod_{\substack{\rho_1, \ldots, \rho_n \text{ is} \\ r\text{-minimal for } \mathcal{H}}} \exists \rho_1 . \cdots \exists \rho_n . \top, \quad \bar{r} \sqsubseteq r\}.$$

   Observe that for any sequence $\rho_1, \ldots, \rho_n \in \{r, \bar{r}\}^n$ we have $\mathcal{H} \models A \sqsubseteq \exists \rho_1 . \cdots \exists \rho_n . \top$ if, and only if, $\mathcal{H}' \models A \sqsubseteq \exists \rho_1 . \cdots \exists \rho_n . \top$. We convert $\mathcal{H}'$ into its corresponding monotone DNF formula $\phi_{\mathcal{H}'}$ by reversing the translation from monotone DNF formulas into DL-Lite$_{\mathcal{R}}^{\exists}$ TBoxes of the above form in the obvious way. Note that the size of $\phi_{\mathcal{H}'}$ is linear in the size of $\mathcal{H}'$. Given $\phi_{\mathcal{H}'}$ the oracle $\mathfrak{O}$ returns a (positive or negative) counterexample (a truth assignment) I. Then return the counterexample in the form of the CI $A \sqsubseteq C_I$.

Observe that the answers given in Points 1 to 3 are correct in the sense that if an inclusion α is returned as a negative example then $\mathcal{T} \not\models \alpha$ for any $\mathcal{T} \in T(n, t, s)$. Point 4 is trivially correct, since any monotone DNF is satisfied by the truth assignment that makes every variable true. We analyse the size of the TBox $\mathcal{H}'$ computed in Point 5.

Lemma 70 *Assume that Points 1 to 4 do not apply to $\mathcal{H}$. Then the number of sequences $\rho_1, \dots, \rho_n \in \{r, \bar{r}\}^n$ which are r-minimal for $\mathcal{H}$ is bounded by $|\mathcal{H}|$.*

Proof We first show that if $\rho_1, \dots, \rho_n \in \{r, \bar{r}\}^n$ is r-minimal for $\mathcal{H}$, then there exists a CI $A \sqsubseteq C \in \mathcal{H}$ such that

(∗) there are concept expressions $C_0, \dots, C_n$ with $C_0 = C$ and $\exists \rho_{i+1}.C_{i+1}$ a top-level conjunct of C_i, for all $i < n$.

For the proof we require the canonical model $\mathcal{I}_{A,\mathcal{H}}$ of A and $\mathcal{H}$ (Lemma 12). Denote the root of $\mathcal{I}_{A,\mathcal{H}}$ by ρ_A. Let $\rho_1, \dots, \rho_n \in \{r, \bar{r}\}^n$ be r-minimal for $\mathcal{H}$. Then there are $d_0, \dots, d_n \in \Delta^{\mathcal{I}_{A,\mathcal{H}}}$ with $d_0 = \rho_A$ such that $(d_i, d_{i+1}) \in \rho_i^{\mathcal{I}_{A,\mathcal{H}}}$ for all $i < n$. By the canonical model construction and the assumption that Points 2 and 3 do not hold, there either exists $d_i \in A^{\mathcal{I}_{A,\mathcal{H}}}$ or there is a CI $A \sqsubseteq C \in \mathcal{H}$ such that (∗) holds. We show that the first condition does not hold. Assume for a prove by contradiction that $d_i \in A^{\mathcal{I}_{A,\mathcal{H}}}$. By Lemma 12, $\mathcal{H} \models A \sqsubseteq \exists \rho_1 \cdots \exists \rho_i.A$. But then $\mathcal{H} \models A \sqsubseteq \exists (\rho_1 \cdots \rho_i)^n.\top$ for all $n > 0$ which contradicts the assumption that Point 1 does not apply to $\mathcal{H}$.

It follows that the number of distinct r-minimal sequences is bounded by the number of distinct sequences $C_0, \dots, C_n$ with $A \sqsubseteq C_0 \in \mathcal{H}$ and $\exists \rho_{i+1}.C_{i+1}$ a top-level conjunct of C_i for all $i < n$. Thus, the number of distinct r-minimal sequences is bounded by $|\mathcal{H}|$. ❑

It follows from Lemma 70 that the size of the TBox $\mathcal{H}'$ computed in Point 5 is bounded by $4n|\mathcal{H}| + 2$.

Theorem 71 *DL-Lite$^{\exists}_{\mathcal{R}}$ TBoxes (even without inverse roles) are not polynomial query learnable using only equivalence queries.*

Proof Suppose that the query complexity of a learning algorithm $\mathfrak{A}$ for DL-Lite$^{\exists}_{\mathcal{R}}$ TBoxes in $\Sigma = \{A, r, \bar{r}\}$ is bounded at every stage of computation by a polynomial $p(x, y)$, where x is the size of the target TBox, and y is the maximal size of a counterexample returned by the oracle up to the current stage of computation. Let $q(n) = (p(n^2, 4n + 6))^2$, and let constants t_0 and s_0 be as guaranteed by Lemma 69. We claim that, for sufficiently large n, $\mathfrak{A}$ cannot distinguish some $\mathcal{T}_\phi$ and $\mathcal{T}_\psi$ for $\phi, \psi \in M(n, t_0, s_0)$.

Assuming that $n > 11$ (the maximal size of counterexamples given under Point 2 and 3), the largest counterexample returned by our strategy described above is of the form $A \sqsubseteq \exists \rho_1 . \cdots \exists \rho_{n+1}.\top$, so for sufficiently large n the maximal size of any counterexample in any run of $\mathfrak{A}$ is bounded by $4n + 6 = 4(n + 1) + 2$. Similarly, the size of every potential target TBox $\mathcal{T}_\phi \in T(n, t_0, s_0)$ does not exceed $t_0 \cdot (4n + 2)$ and, as t_0 is a constant, for sufficiently large n it is bounded by n^2. Thus, for sufficiently large n the total query complexity of $\mathfrak{A}$ on any input from $T(n, t_0, s_0)$ is bounded by $p(n^2, 4n + 6)$. Obviously, the size of each query is bounded by the query complexity of the learning algorithm. So, the size of a DNF equivalence query forwarded to the strategy $\mathfrak{O}$ guaranteed by Lemma 69 is bounded by $4n \times p(n^2, 4n + 6) + 2 \leq q(n)$, and there will be at most $q(n)$ queries forwarded. But then $\mathfrak{O}$ can return answers such that some ϕ and ψ from $M(n, t_0, s_0)$ cannot be distinguished. It remains to observe that $\mathfrak{A}$ cannot distinguish $\mathcal{T}_\phi$ and $\mathcal{T}_\psi$. ❑

7. Related Work

Some related work has already been discussed in the introduction to this paper. Here we discuss in more detail related work from *ontology learning* in general and *exact learning of ontologies* in particular. We start with the former.

Ontology Learning. Research in ontology learning has a rich history that we cannot discuss here in full detail. The collection edited by Lehmann and Völker (2014) and surveys authored by Cimiano et al. (2010) and Wong et al. (2012) provide an excellent introduction to the state of the art in this field. The techniques applied in ontology learning range from information extraction and text mining to interactive learning and inductive logic programming (ILP). Of particular relevance for this paper are the approaches to learning logical expressions (rather than subsumption hierarchies between concept names). For example, the work of Lehmann and Haase (2009), Lehmann and Hitzler (2010), and Bühmann et al. (2014) applies techniques from ILP to learn description logic concept expressions. ILP is applied as well by Lisi (2011) for learning logical rules for ontologies. The learning of fuzzy DLs has been considered by Lisi and Straccia (2015). Other machine learning methods which have been applied to learn ontology axioms include Association Rule Mining (ARM) (Völker and Niepert, 2011; Fleischhacker et al., 2012; Völker et al., 2015) and Formal Concept Analysis (FCA) (Rudolph, 2004; Baader et al., 2007; Distel, 2011; Borchmann, 2014; Ganter et al., 2016). Recently, learnability of lightweight DL TBoxes from finite sets of interpretations has been investigated (Klarman and Britz, 2015).

Exact Learning of Description Logic Concept Expressions. Rather than aiming to learn a TBox here one is interested in learning a target concept expression C_*. This was first studied by Cohen and Hirsh (1994a,b) and Frazier and Pitt (1996). The standard learning protocol is as follows:

- a membership query asks whether a concept expression C is subsumed by the target concept expression C_* (in symbols, $\emptyset \models C \sqsubseteq C_*$?);
- an equivalence query asks whether a concept expression C is equivalent to the target concept expression C_* (in symbols, $\emptyset \models C \equiv C_*$?). If C and C_* are not equivalent then the oracle gives a counterexample, that is, a concept expression C' such that either $\emptyset \models C' \sqsubseteq C_*$ and $\emptyset \not\models C' \sqsubseteq C$ or $\emptyset \not\models C' \sqsubseteq C_*$ and $\emptyset \models C' \sqsubseteq C$.

Cohen and Hirsh (1994a,b) and Frazier and Pitt (1996) consider concept expressions in (variations of) the now largely historic description logic Classic (Borgida et al., 1989; Patel-Schneider et al., 1991; Borgida and Patel-Schneider, 1994). The expressive power of Classic and its variants is incomparable to the expressive power of modern lightweight description logics. Classic only shares conjunction and unqualified existential restrictions of the form $\exists r.\top$ with the DLs considered in this paper. It additionally admits *value restrictions* $\forall r.C$ whose interpretation is given as

$$(\forall r.C)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} | d' \in C^{\mathcal{I}} \text{ for all } d' \text{ with } (d, d') \in r^{\mathcal{I}}\}$$

and *unqualified number restrictions* $(\leq n\ r)$ and $(\geq n\ r)$ interpreted as

$$\begin{aligned} (\leq nr)^{\mathcal{I}} &= \{d \in \Delta^{\mathcal{I}} \mid |\{d' \mid (d, d') \in r^{\mathcal{I}}\}| \leq n\} \\ (\geq nr)^{\mathcal{I}} &= \{d \in \Delta^{\mathcal{I}} \mid |\{d' \mid (d, d') \in r^{\mathcal{I}}\}| \geq n\} \end{aligned}$$

as well as various constructors using individual names. For example, if $a_1, \ldots, a_n$ are names for individual objects, then ONE-OF$(a_1, \ldots, a_n)$ is a CLASSIC concept denoting the set $\{a_1^{\mathcal{I}}, \ldots, a_n^{\mathcal{I}}\}$, where $a_i^{\mathcal{I}}$ denotes the individual with name a_i in interpretation $\mathcal{I}$. It is proved by Cohen and Hirsh (1994a,b) and Frazier and Pitt (1996) that in many fragments of CLASSIC concept expressions cannot be learned polynomially using only membership or equivalence queries but that they can be learned in polynomial time using both. Exact learning of concept expressions in modern lightweight description logics has not yet been investigated.

Exact Learning of TBoxes using Concept Inclusions as Queries. First results on exact learning of description logic TBoxes using concept inclusions as queries were presented by Konev et al. (2013, 2014). This paper is an extension. In contrast to the work of Konev et al. (2013, 2014), we make the distinction between polynomial time and polynomial query learnability which enables us to formulate and prove results on a more fine grained level. TBoxes in DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$, for which we prove polynomial query learnability, were not considered by Konev et al. (2013, 2014). The current paper is also closely related to the PhD thesis of the third author (Ozaki, 2016). In addition to the results presented here, it is shown there that even in the extension of $\mathcal{EL}_{\mathsf{lhs}}$ with role inclusions, TBoxes can be learned in polynomial time. The learning algorithm is a non-trivial extension of the algorithm presented here for $\mathcal{EL}_{\mathsf{lhs}}$ TBoxes.

Exact Learning of TBoxes using Certain Answers. In recent years, data access mediated by ontologies has become one of the most important applications of DLs (Poggi et al., 2008; Bienvenu et al., 2014; Kontchakov and Zakharyaschev, 2014; Bienvenu and Ortiz, 2015) and references therein. The idea is to use a TBox to specify semantics and background knowledge for the data and use it for deriving more complete answers to queries over the data. In this context, the data is stored in an *ABox* consisting of a finite set of assertions of the form $A(a)$ or $r(a, b)$, where A is a concept names, r a role name, and a, b are individual names. Given a query $q(\vec{x})$ (typically a conjunctive query), a TBox $\mathcal{T}$, and an ABox $\mathcal{A}$, a tuple of individual names $\vec{a}$ from $\mathcal{A}$ and of the same length as $\vec{x}$ is called a certain answer to $q(\vec{x})$ over $\mathcal{A}$ w.r.t. $\mathcal{T}$, in symbols $\mathcal{T}, \mathcal{A} \models q(\vec{a})$, if every model $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$ satisfies $q(\vec{a})$. Motivated by this setup, Konev et al. (2016) and Ozaki (2016) study polynomial learnability of TBoxes using membership queries that ask whether a tuple of individuals names is a certain answer to a query over an ABox w.r.t. the target TBox. This is a natural alternative to learning using concept inclusions since domain experts are often more familiar with querying data in a particular domain than with the logical notion of subsumption between concept expressions. In detail, the learning protocol is as follows:

- a membership query takes the form $(\mathcal{A}, q(\vec{a}))$ and asks whether the tuple $\vec{a}$ of individual names is a certain answer to the query $q(\vec{x})$ over the ABox $\mathcal{A}$ w.r.t. the target TBox $\mathcal{T}$;

- an equivalence query asks whether a TBox $\mathcal{H}$ is equivalent to the target TBox $\mathcal{T}$. If $\mathcal{T}$ and $\mathcal{H}$ are not equivalent then a counterexample of the form $(\mathcal{A}, q(\vec{a}))$ is given such that $\mathcal{T}, \mathcal{A} \models q(\vec{a})$ and $\mathcal{H}, \mathcal{A} \not\models q(\vec{a})$ (a positive counterexample) or $\mathcal{T}, \mathcal{A} \not\models q(\vec{a})$ and $\mathcal{H}, \mathcal{A} \models q(\vec{a})$ (a negative counterexample).

In the learning protocol above we have not yet specified the class of queries from which the $q(\vec{x})$ are drawn and which strongly influences the classes of TBoxes that can be learned. In the context of data access using TBoxes the two most popular classes of queries are:

- conjunctive queries (CQs), that is, existentially quantified conjunctions of atoms; and

- instance queries (IQs), which take the form $C(x)$ or $r(x, y)$ with C a concept expression from the DL under consideration and r a role name.

Konev et al. (2016) and Ozaki (2016) study exact learning of TBoxes in the languages $\mathcal{EL}$, $\mathcal{EL}_{\mathsf{lhs}}$ and DL-Lite$^{\exists}_{\mathcal{R}}$ for both IQs and CQs in queries. The positive learnability results are proved by polynomial reductions to the learnability results presented in this paper and also by Ozaki (2016). The basic link between learning using concept inclusions as queries and learning by certain answers is as follows: if $\mathcal{T}$ is a TBox and C, D are concept expressions in any of the DLs discussed above then one can regard the labelled tree T_C corresponding to C as an ABox $\mathcal{A}_C$ with root ρ_C and it holds that $\mathcal{T} \models C \sqsubseteq D$ if, and only if, $\mathcal{T}, \mathcal{A}_C \models D(\rho_C)$. The converse direction (obtaining a concept expression from an ABox) is more involved since ABoxes are not tree-shaped and an additional unfolding step is needed to compute a corresponding concept expression. Using this link it is proved by Konev et al. (2016) and Ozaki (2016) that DL-Lite$^{\exists}_{\mathcal{R}}$ and $\mathcal{EL}_{\mathsf{lhs}}$ TBoxes with role inclusions can be learned with polynomially many queries using certain answers to IQs. It is also proved that $\mathcal{EL}$ is still not learnable with polynomially many queries using certain answers with neither IQs nor CQs as the query language and that DL-Lite$^{\exists}_{\mathcal{R}}$ TBoxes cannot be learned with polynomially many queries using certain answers with CQs as the query language.

Exact Learning in (other) Fragments of FO Horn. We discuss results on exact learning of finite sets of FO Horn clauses or fragments of this logic, where a *FO Horn clause* is a universally quantified clause with at most one positive literal (Page Jr, 1993; Arimura, 1997; Reddy and Tadepalli, 1998; Arias and Khardon, 2002; Arias et al., 2007; Selman and Fern, 2011). Depending on what is used as membership queries and as counterexamples to equivalence queries, one can distinguish between exact learning FO Horn clauses using interpretations and using entailments. As learning using entailments is closer to our approach we focus on that setting. The exact learning protocol is then as follows:

- a membership query asks whether an FO Horn clause is entailed by the target set T of FO Horn clauses;

- an equivalence query asks whether a set H of FO Horn clauses is equivalent to the target set T. If H and T are not equivalent then a counterexample is given, that is, an FO Horn clause entailed by T but not by H (a positive counterexample) or vice versa.

Considering how terms (with function symbols allowed) can appear in an FO Horn clause, two main restrictions have been studied in the literature:

1. *Range restricted clauses*: when the set of terms in the positive literal (if existent) is a subset of the terms in the negative literals and their subterms; and

2. *Constrained clauses*: when the set of terms and subterms in the positive literal (if existent) is a superset of the terms in the negative literals.

For example, the FO Horn clause $\forall x(\neg P(f(x)) \vee P(x))$ is range restricted but not constrained and the FO Horn clause $\forall x(\neg P(x) \vee P(f(x)))$ is constrained but not range restricted, where P is a predicate symbol and f a function symbol. Reddy and Tadepalli (1998) and Arimura (1997), it is shown that under certain acyclicity conditions FO Horn with range restricted clauses and, respectively, constrained clauses are polynomial time learnable from entailments if the arity of predicates is bounded by a constant. A learning algorithm for a fragment of FO Horn (called closed FO Horn) that subsumes the two languages defined above is presented by Arias and Khardon (2002). The algorithm is polynomial in the number of clauses, terms and predicates and the size of the counterexamples, but exponential not only in the arity of predicates but also in the number of variables per clause. In fact, it is an open question whether there exists a learning algorithm for closed FO Horn that is polynomial in the number of variables per clause.

We relate the learnability results for FO Horn to the learnability results for lightweight description logics presented in this paper. Observe that most DLs (and in particular all DLs investigated in this paper) can be translated into FO (Baader et al., 2003). For example, a translation of the $\mathcal{EL}_{\mathsf{lhs}}$ CI $\exists r.A \sqsubseteq B$ is $\forall x \forall y(\neg r(x,y) \vee \neg A(y) \vee B(x))$ and a translation of the DL-Lite$^{\exists}_{\mathcal{R}}$ CI $A \sqsubseteq \exists r.A$ is $\forall x(A(x) \rightarrow \exists y.(r(x,y) \wedge A(y)))$. Under this translation, every $\mathcal{EL}_{\mathsf{lhs}}$ TBox can be regarded as a set of range restricted FO Horn clauses, where the arity of predicates is bounded by 2. In contrast, since in DL-Lite$^{\exists}_{\mathcal{R}}$ existential quantifiers can be nested in the right side of CIs, DL-Lite$^{\exists}_{\mathcal{R}}$ CIs cannot be translated into FO Horn clauses. We can now summarise the relationship between our learnability results for $\mathcal{EL}_{\mathsf{lhs}}$, DL-Lite$^{\exists}_{\mathcal{R}}$ and DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$ and the results on exact learnability of FO Horn from entailments as follows: since the arity of DL predicates is at most 2 and since no function symbols are admitted in DLs, none of the DLs considered in this paper can express the fragments of FO Horn discussed above. On the other hand, we do not impose an acyclicity condition on the TBoxes (in contrast to the work by Reddy and Tadepalli (1998); Arimura (1997)) and our algorithms are polynomial in the number of variables permitted in any clause (in contrast to the work by Arias and Khardon (2002)). Thus, the results discussed above for FO Horn do not translate into polynomial learning algorithms for $\mathcal{EL}_{\mathsf{lhs}}$ and are not applicable to DL-Lite$^{\exists}_{\mathcal{R}}$ nor DL-Lite$^{\exists}_{\mathcal{R},\mathsf{horn}}$. Our results thus cover new fragments of FO that have not yet been considered for exact learning. This is not surprising, given the fact that the fragments of FO considered previously were not motivated by applications in ontology learning.

Also related to exact learning of Horn FO is recent work on exact learning of schema mappings in data exchange (ten Cate et al., 2012). Schema mappings are tuples (S,T,M) where S is a source schema (a finite set of predicates), T is a target schema (a finite set of predicates), and M is a finite set of sentences of the form $\forall \vec{x}(\varphi(\vec{x}) \rightarrow \exists \vec{y} \psi(\vec{x},\vec{y}))$ where $\varphi(\vec{x})$ and $\psi(\vec{x},\vec{y})$ are conjunctions of atoms over S and T, respectively (Fagin et al., 2005). (S,T,M) is a GAV schema mapping if $\vec{y}$ is empty and $\psi(\vec{x},\vec{y})$ is an atom. The authors study exact learnability of GAV schema mappings from data examples (I,J) consisting of a database I over the source schema S and a database J over the target schema T. Such a data example satisfies M if $I \cup J \models M$. The authors present both polynomial query learnability results for protocols using membership and equivalence queries and non-polynomial query learnability results if either only membership or only equivalence queries are allowed. These results are not applicable to the setting considered in this paper since the learning protocol uses data examples instead of entailments.

8. Conclusion

We have presented the first study of learnability of DL ontologies in Angluin et al's framework of exact learning, obtaining both positive and negative results. Several research questions remain to be explored. One immediate question is whether acyclic $\mathcal{EL}$ TBoxes can be learned in polynomial time using queries and counterexamples of the form $A \equiv C$ and $A \sqsubseteq C$ only. Note that our non-polynomial query learnability result for acyclic $\mathcal{EL}$ TBoxes relies heavily on counterexamples that are not of this form. Another immediate question is whether the extension of $\mathcal{EL}_{\mathsf{lhs}}$ with inverse roles (which is a better approximation of OWL2 RL than $\mathcal{EL}_{\mathsf{lhs}}$ itself) can still be learned in polynomial time, or at least with polynomially many queries of polynomial size. Other interesting research directions are non-polynomial time learning algorithms for $\mathcal{EL}$ TBoxes and the admission of different types of membership queries and counterexamples in the learning protocol. For example, one could replace CIs as counterexamples with interpretations.

Acknowledgments

We would like to thank the reviewers for their helpful comments. Lutz was supported by the DFG project Prob-DL (LU1417/1-1). Konev and Wolter were supported by the EPSRC project EP/H043594/1. Ozaki was supported by the Science without Borders scholarship programme (245288/2012-0) and cfaed.

Appendix A. Proofs for Section 6

We supply proofs for Lemma 62 and Lemma 64. In addition, we prove a claim used in the proof of Lemma 67. We start by giving the proof of Lemma 62.

Lemma 62 *For every DL-Lite$^{\exists}_{\mathcal{R}}$ CI $B \sqsubseteq D$ over the signature of $\mathcal{T}_{\boldsymbol{\sigma}}$,*

- *either $\mathcal{T}_{\boldsymbol{\sigma}} \models B \sqsubseteq D$ for every $\mathcal{T}_{\boldsymbol{\sigma}} \in S_N$*
- *or there is at most one $\mathcal{T}_{\boldsymbol{\sigma}} \in S_N$ such that $\mathcal{T}_{\boldsymbol{\sigma}} \models B \sqsubseteq D$.*

Proof Assume the CI $B \sqsubseteq D$ is given. If $B \neq A$ or M does not occur in D, then the claim can be readily checked. Thus, we assume that $B = A$ and M occurs in D. Assume there exists $\boldsymbol{\sigma}_0$ such that $\mathcal{T}_{\boldsymbol{\sigma}_0} \models A \sqsubseteq D$ (if no such $\boldsymbol{\sigma}_0$ exists, we are done). For any $\boldsymbol{\sigma}$, let $\mathcal{I}_{A,\mathcal{T}_{\boldsymbol{\sigma}}}$ be the canonical model of A and $\mathcal{T}_{\boldsymbol{\sigma}}$ (Lemma 12). Apply the following restricted form of parent/child merging exhaustively to the concept expression D:

- if there are nodes $d, d_1, d_2 \in T_D$ with $l(d_1,d) = \sigma$ and $l(d,d_2) = \sigma^-$ for some $\sigma \in \{r,s\}$, then replace D by the resulting concept expression after d_1 and d_2 are merged in D.

Let D' be the resulting concept expression. Recall from Lemma 12 that $\mathcal{T}_{\boldsymbol{\sigma}} \models A \sqsubseteq D$ iff there is a homomorphism from T_D to $\mathcal{I}_{A,\mathcal{T}_{\boldsymbol{\sigma}}}$ mapping ρ_D to ρ_A. Using the fact that $\mathcal{I}_{A,\mathcal{T}_{\boldsymbol{\sigma}}}$ is a ditree interpretation, one can readily check that any homomorphism h from T_D to $\mathcal{I}_{A,\mathcal{T}_{\boldsymbol{\sigma}}}$ mapping ρ_D to ρ_A factors through $T_{D'}$ and that D' is an $\mathcal{EL}$ concept expression. Thus, if there is an additional $\boldsymbol{\sigma}' \neq \boldsymbol{\sigma}_0$ such that $\mathcal{T}_{\boldsymbol{\sigma}'} \models A \sqsubseteq D$, then there are two homomorphisms

$h_{\boldsymbol{\sigma}_0}$ and $h_{\boldsymbol{\sigma}'}$ with the same domain $T_{D'}$ into $\mathcal{I}_{A,\mathcal{T}_{\boldsymbol{\sigma}_0}}$ and $\mathcal{I}_{A,\mathcal{T}_{\boldsymbol{\sigma}'}}$ and mapping the root of $T_{D'}$ to the roots of $\mathcal{I}_{A,\mathcal{T}_{\boldsymbol{\sigma}_0}}$ and $\mathcal{I}_{A,\mathcal{T}_{\boldsymbol{\sigma}'}}$, respectively. Since M occurs in D' and D' is an $\mathcal{EL}$ concept expression we find a sequence $D_0, \ldots, D_m$ with $D_0 = D'$ and $D_m = M$ such that $\exists s_{i+1}.D_{i+1}$ is a top-level conjunct of D_i for $s_i \in \{r, s\}$ and all $i < m$. But then $s_1 \cdots s_m = \boldsymbol{\sigma}_0$ and $s_1 \cdots s_m = \boldsymbol{\sigma}'$ and we have derived a contradiction to the assumption that $\boldsymbol{\sigma}_0$ and $\boldsymbol{\sigma}'$ are distinct. ❑

To prove Lemma 64 we require the following observation.

Lemma 72 *For any acyclic $\mathcal{EL}$ TBox $\mathcal{T}$, any CI $A \sqsubseteq C \in \mathcal{T}$ and any concept expression of the form $\exists t.D$ we have $\mathcal{T} \models A \sqsubseteq \exists t.D$ if, and only if, $\mathcal{T} \models C \sqsubseteq \exists t.D$.*

We are now ready to prove Lemma 64.

Lemma 64 *For every $\mathcal{EL}$ CI $C \sqsubseteq D$ over Σ_n:*

- *either $\mathcal{T}_L \models C \sqsubseteq D$ for every $L \in \mathfrak{L}_n$*
- *or the number of $L \in \mathfrak{L}_n$ such that $\mathcal{T}_L \models C \sqsubseteq D$ does not exceed $|C|$.*

Proof We prove the lemma by induction on the structure of D. We assume throughout the proof that there exists some $L_0 \in \mathfrak{L}_n$ such that $\mathcal{T}_{L_0} \models C \sqsubseteq D$.

Base case: D is a concept name. We make the following case distinction.

- $D \in \{X_i, A_i, B_i \mid 1 \leq i \leq n\}$ or $D = M$. By Lemma 65, C is of the form $Z \sqcap C'$, for some concept name Z, and $\mathcal{T}_{L_0} \models Z \sqsubseteq D$. But then $Z = D$ and it follows that $\mathcal{T}_L \models C \sqsubseteq D$ for every $L \in \mathfrak{L}_n$.
- $D = X_0$. By Lemma 65, C is of the form $Z \sqcap C'$, for some concept name Z, and $\mathcal{T}_{L_0} \models Z \sqsubseteq X_0$. This is the case if either $Z = X_0$, or $Z \in \{A, A_1, B_1, \ldots, A_n, B_n\}$. In either case, $\mathcal{T}_L \models C \sqsubseteq X_0$ for every $L \in \mathfrak{L}_n$.
- $D = A$. If C is of the form $A \sqcap C'$ or for all i such that $1 \leq i \leq n$, A_i or B_i is a conjunct of C, then $\mathcal{T}_L \models C \sqsubseteq A$ for every $L \in \mathfrak{L}_n$. Assume now that C is not of this form. Then for some j such that $1 \leq j \leq n$, C is neither of the form $A \sqcap C'$ nor of the form $A_j \sqcap C'$ nor of the form $B_j \sqcap C'$. Let $L = (\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_n) \in \mathfrak{L}_n$ be such that $\mathcal{T}_L \models C \sqsubseteq A$. Notice that $\mathcal{T}_L \models C \sqsubseteq A$, for $L = (\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_n) \in \mathfrak{L}_n$, if, and only if, $\mathcal{T}_L \models C \sqsubseteq X_0 \sqcap \exists \boldsymbol{\sigma}_1.M \sqcap \cdots \sqcap \exists \boldsymbol{\sigma}_n.M$. By the claim in the proof of Lemma 66, for such a $\mathcal{T}_L$ we must have $\emptyset \models C \sqsubseteq \exists \boldsymbol{\sigma}_j.M$. Clearly, the number of $L = (\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_n) \in \mathfrak{L}_n$ with $\emptyset \models C \sqsubseteq \exists \boldsymbol{\sigma}_j.M$ does not exceed $|C|$.

  Thus, either $\mathcal{T}_L \models C \sqsubseteq A$ for every $L \in \mathfrak{L}_n$ or the number of $L \in \mathfrak{L}_n$ such that $\mathcal{T}_L \models C \sqsubseteq A$ does not exceed $|C|$.

Induction step. If $D = D_1 \sqcap D_2$, then $\mathcal{T}_L \models C \sqsubseteq D$ if, and only if, $\mathcal{T} \models C \sqsubseteq D_i$, $i = 1, 2$. By induction hypothesis, for $i = 1, 2$ either $\mathcal{T}_L \models C \sqsubseteq D_i$ for every $L \in \mathfrak{L}_n$ or there exist at most $|C|$ different $L \in \mathfrak{L}_n$ such that $\mathcal{T}_L \models C \sqsubseteq D_i$. Thus either $\mathcal{T}_L \models C \sqsubseteq D$ for every $L \in \mathfrak{L}_n$ or the number of $L \in \mathfrak{L}_n$ such that $\mathcal{T}_L \models C \sqsubseteq D$ also does not exceed $|C|$.

Now assume that $D = \exists t.D'$. Suppose that $\mathcal{T}_L \models C \sqsubseteq D$ for some $L \in \mathfrak{L}_n$. Then, by Lemma 65, either there exists a conjunct Z of C, Z a concept name, such that $\mathcal{T}_L \models Z \sqsubseteq \exists t.D'$ or there exists a conjunct $\exists t.C'$ of C with $\mathcal{T}_L \models C' \sqsubseteq D'$. We analyse for every conjunct of C of the form Z or $\exists t.C'$ the number of $L \in \mathfrak{L}_n$ such that $\mathcal{T}_L \models Z \sqsubseteq \exists t.D'$ or $\mathcal{T}_L \models \exists t.C' \sqsubseteq \exists t.D'$, respectively.

(i) Let Z be a conjunct of C such that Z is a concept name and $\mathcal{T}_L \models Z \sqsubseteq \exists t.D'$. Notice that $Z \neq M$ as there is no $L \in \mathfrak{L}_n$ such that $\mathcal{T}_L \models M \sqsubseteq \exists t.D'$. We consider the remaining cases.

- $Z = X_i$, for some $i \geq 0$. It is easy to see that for $L, L' \in \mathfrak{L}_n$ we have $\mathcal{T}_L \models X_i \sqsubseteq \exists t.D'$ if, and only if $\mathcal{T}_{L'} \models X_i \sqsubseteq \exists t.D'$. Thus, $\mathcal{T}_L \models Z \sqsubseteq \exists t.D'$ for every $L \in \mathfrak{L}_n$.
- $Z \in \{A_i, B_i \mid 1 \leq i \leq n\}$. By Lemma 72, $\mathcal{T}_L \models Z \sqsubseteq \exists t.D'$ if, and only if, $\mathcal{T}_L \models X_0 \sqcap \exists \boldsymbol{\sigma}_i.M \sqsubseteq \exists t.D'$. By Lemma 65, either $\mathcal{T}_L \models X_0 \sqsubseteq \exists t.D'$ or $\mathcal{T}_L \models \exists \boldsymbol{\sigma}_i.M \sqsubseteq \exists t.D'$. If $\mathcal{T}_L \models X_0 \sqsubseteq \exists t.D'$ then, as above, for $\mathcal{T}_L \models C \sqsubseteq \exists t.D'$ every $L \in \mathfrak{L}_n$. Suppose that $\exists t.D'$ is such that $\mathcal{T}_L \not\models X_0 \sqsubseteq \exists t.D'$ and $\mathcal{T}_L \models \exists \boldsymbol{\sigma}_i.M \sqsubseteq \exists t.D'$. By inductive applications of Lemma 65, this is only possible if $\exists t.D' = \exists \boldsymbol{\sigma}_i.M$. Thus, there is exactly one $L \in \mathfrak{L}_n$ (namely, $L = L_0$) such that $\mathcal{T}_L \models Z \sqsubseteq \exists \boldsymbol{\sigma}_i.M$.
- $Z = A$. Suppose that for some $L = (\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_n) \in \mathfrak{L}_n$ we have $\mathcal{T}_L \models A \sqsubseteq \exists t.D'$. Equivalently, $\mathcal{T}_L \models X_0 \sqcap \exists \boldsymbol{\sigma}_1.M \sqcap \ldots \boldsymbol{\sigma}_n.M \sqsubseteq \exists t.D'$. By Lemma 65, either $\mathcal{T}_L \models X_0 \sqsubseteq \exists t.D'$ or $\mathcal{T}_L \models \exists \boldsymbol{\sigma}_i.M \sqsubseteq \exists t.D'$ for some i with $1 \leq i \leq n$. Thus, as above, unless $\mathcal{T}_L \models X_0 \sqsubseteq \exists t.D'$ we have $\exists t.D'$ is $\exists \boldsymbol{\sigma}_i.M$. But then $L = L_0$.

(ii) Let $\exists t.C'$ be a conjunct of C with $\mathcal{T}_L \models C' \sqsubseteq D'$. The induction hypothesis implies that the number of $L \in \mathfrak{L}_n$ such that $\mathcal{T}_L \models C' \sqsubseteq D'$ does not exceed $|C'|$.

To summarise, either $\mathcal{T}_L \models C \sqsubseteq \exists t.D'$ for every $L \in \mathfrak{L}_n$ or for every conjunct C_0 of C of the form Z or $\exists t.C'$, the number of $L \in \mathfrak{L}_n$ such that $\mathcal{T}_L \models C_0 \sqsubseteq \exists t.D'$ does not exceed $|C_0|$. Hence the number of $L \in \mathfrak{L}_n$ such that $\mathcal{T}_L \models C \sqsubseteq \exists t.D'$ does not exceed $|C|$. ❑

The next result is used in the proof of Lemma 67.

Lemma 73 *For any $0 \leq i \leq n$ and concept expression D over Σ_n, if $\mathcal{T}_0 \not\models X_i \sqsubseteq D$ then there exists a sequence of role names $t_1, \ldots t_l \in \{r, s\}^l$ such that $\emptyset \models D \sqsubseteq \exists t_1. \cdots \exists t_l.Y$ and $\mathcal{T}_0 \not\models X_i \sqsubseteq \exists t_1. \cdots \exists t_l.Y$, where Y is either $\top$ or a concept name and $0 \leq l \leq n - i + 1$.*

Proof We prove the lemma by induction on i from $i = n$ to $i = 0$. If $i = n$, then $\mathcal{T}_0 \not\models X_i \sqsubseteq D$ if either $\emptyset \models D \sqsubseteq \exists t.\top$, for some role name t, or $\emptyset \models D \sqsubseteq Y$, for some concept name $Y \neq X_i$.

Suppose that the lemma is proved for $0 < j \leq n$ and let $i = j - 1$. We proceed by induction on the structure of D. If D is a concept name, we are done as $\mathcal{T}_0 \models X_i \sqsubseteq Z$ does not hold for any concept name $Z \neq X_i$. If D is of the form $\exists t.D'$, where $t \in \{r, s\}$, then $\mathcal{T}_0 \not\models X_{i+1} \sqsubseteq D'$, and so, by induction hypothesis, there exists a sequence of role names $t_1, \ldots, t_l$, with $l \leq n - i$, such that $\mathcal{T}_0 \not\models X_{i+1} \sqsubseteq \exists t_1. \cdots \exists t_l.Y$ and $\emptyset \models D' \sqsubseteq \exists t_1. \cdots \exists t_l.Y$. But then, by Lemma 72 and Lemma 65, $\mathcal{T}_0 \not\models X_i \sqsubseteq \exists t.\exists t_1. \cdots \exists t_l.Y$ and $\emptyset \models \exists t.D' \sqsubseteq \exists t.\exists t_1. \cdots \exists t_l.Y$. If D is of the form $D = D_1 \sqcap D_2$, there there exists D_i, $i = 1, 2$, such that $\mathcal{T}_0 \not\models X_i \sqsubseteq D_i$ and the lemma holds by induction hypothesis. ❑

JMLR 18(201):1-63, 2018

References

Dana Angluin. Learning propositional Horn sentences with hints. Technical report, Yale University, 1987a.

Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1987b.

Dana Angluin. Negative results for equivalence queries. *Machine Learning*, 5:121–150, 1990.

Dana Angluin, Michael Frazier, and Leonard Pitt. Learning conjunctions of Horn clauses. *Machine Learning*, 9:147–164, 1992.

Marta Arias. *Exact learning of first-order expressions from queries*. PhD thesis, Tufts University, 2004.

Marta Arias and José L. Balcázar. Construction and learnability of canonical Horn formulas. *Machine Learning*, 85(3):273–297, 2011.

Marta Arias and Roni Khardon. Learning closed Horn expressions. *Information and Computation*, 178(1):214–240, 2002.

Marta Arias, Roni Khardon, and Jérôme Maloberti. Learning Horn expressions with LOGAN-H. *Journal of Machine Learning Research*, 8:549–587, 2007.

Hiroki Arimura. Learning acyclic first-order Horn sentences from entailment. In *International Workshop on Algorithmic Learning Theory*, pages 432–445, 1997.

Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev. The DL-Lite family and relations. *Journal of Artificial Intelligence Research (JAIR)*, 36:1–69, 2009.

Franz Baader, Ralf Küsters, and Ralf Molitor. Computing least common subsumers in description logics with existential restrictions. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 96–103, 1999.

Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, New York, NY, USA, 2003. ISBN 0-521-78176-0.

Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 364–369, 2005.

Franz Baader, Bernhard Ganter, Baris Sertkaya, and Ulrike Sattler. Completing description logic knowledge bases using formal concept analysis. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 230–235, 2007.

Franz Baader, Carsten Lutz, and Sebastian Brandt. Pushing the EL envelope further. In *Proceedings of the Fourth OWLED Workshop on OWL: Experiences and Directions*, 2008.

Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.

Meghyn Bienvenu and Magdalena Ortiz. Ontology-mediated query answering with data-tractable description logics. In *Reasoning Web. Semantic Technologies for Advanced Query Answering - 11th International Summer School*, pages 218–307, 2015.

Meghyn Bienvenu, Balder ten Cate, Carsten Lutz, and Frank Wolter. Ontology-based data access: A study through disjunctive datalog, CSP, and MMSNP. *ACM Trans. Database Syst.*, 39(4):33:1–33:44, 2014.

Daniel Borchmann and Felix Distel. Mining of $\mathcal{EL}$-GCIs. In *The 11th IEEE International Conference on Data Mining Workshops*, Vancouver, Canada, 2011.

Daniel Borchmann. *Learning terminological knowledge with high confidence from erroneous data*. PhD thesis, Higher School of Economics, 2014.

Alexander Borgida and Peter F Patel-Schneider. A semantics and complete algorithm for subsumption in the classic description logic. *Journal of Artificial Intelligence Research*, 1: 277–308, 1994.

Alexander Borgida, Ronald J. Brachman, Deborah L. McGuinness, and Lori Alperin Resnick. CLASSIC: A structural data model for objects. In *Proceedings of the 1989 ACM SIGMOD International Conference on Management of Data*, pages 58–67, 1989.

Ronald J. Brachman and James G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, 1985.

Lorenz Bühmann, Daniel Fleischhacker, Jens Lehmann, André Melo, and Johanna Völker. Inductive lexical learning of class expressions. In *Knowledge Engineering and Knowledge Management - 19th International Conference (EKAW)*, pages 42–53, 2014.

Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *Journal of Automated reasoning*, 39(3):385–429, 2007.

Philipp Cimiano, Johanna Völker, and Paul Buitelaar. Ontology construction. In *Handbook of Natural Language Processing, Second Edition.*, pages 577–604. Chapman and Hall/CRC, 2010.

William W Cohen and Haym Hirsh. The learnability of description logics with equality constraints. *Machine Learning*, 17(2-3):169–199, 1994a.

William W. Cohen and Haym Hirsh. Learning the CLASSIC description logic: Theoretical and experimental results. In *Principles of Knowledge Representation and Reasoning (KR)*, pages 121–133, 1994b.

Felix Distel. *Learning description logic knowledge bases from data using methods from formal concept analysis*. PhD thesis, Dresden University of Technology, 2011.

Ronald Fagin, Phokion G Kolaitis, Renée J Miller, and Lucian Popa. Data exchange: semantics and query answering. *Theoretical Computer Science*, 336(1):89–124, 2005.

JMLR 18(201):1-63, 2018

Daniel Fleischhacker, Johanna Völker, and Heiner Stuckenschmidt. Mining RDF data for property axioms. In *On the Move to Meaningful Internet Systems: OTM 2012*, pages 718–735. Springer, 2012.

Michael Frazier and Leonard Pitt. Learning from entailment: An application to propositional Horn sentences. In *International Conference on Machine Learning (ICML)*, pages 120–127, 1993.

Michael Frazier and Leonard Pitt. Classic learning. *Machine Learning*, 25(2-3):151–193, 1996.

Bernhard Ganter, Sergei A Obiedkov, Sebastian Rudolph, and Gerd Stumme. *Conceptual exploration*. Springer, 2016.

Ernesto Jiménez-Ruiz, Evgeny Kharlamov, Dmitriy Zheleznyakov, Ian Horrocks, Christoph Pinkel, Martin G Skjæveland, Evgenij Thorstensen, and Jose Mora. BootOX: practical mapping of RDBs to OWL 2. In *International Semantic Web Conference, (ISWC)*, pages 113–132. 2015.

Stanislav Kikot, Roman Kontchakov, and Michael Zakharyaschev. On (in)tractability of OBDA with OWL 2 QL. In *Proceedings of the 24th International Workshop on Description Logics (DL 2011)*, 2011.

Szymon Klarman and Katarina Britz. Ontology learning from interpretations in lightweight description logics. In *Inductive Logic Programming*, 2015.

Boris Konev, Michel Ludwig, Dirk Walther, and Frank Wolter. The logical difference for the lightweight description logic EL. *Journal of Artificial Intelligence Research (JAIR)*, 44: 633–708, 2012.

Boris Konev, Carsten Lutz, and Frank Wolter. Exact learning of Tboxes in EL and DL-Lite. In *Informal Proceedings of the 26th International Workshop on Description Logics*, pages 341–352, 2013.

Boris Konev, Carsten Lutz, Ana Ozaki, and Frank Wolter. Exact learning of lightweight description logic ontologies. In *Principles of Knowledge Representation and Reasoning (KR)*, 2014.

Boris Konev, Ana Ozaki, and Frank Wolter. A model for learning description logic ontologies based on exact learning. In *Conference on Artificial Intelligence (AAAI)*, pages 1008–1015, 2016.

Roman Kontchakov and Michael Zakharyaschev. An introduction to description logics and query rewriting. In *Reasoning Web. Semantic Technologies for Advanced Query Answering - 10th International Summer School*, pages 195–244, 2014.

Markus Krötzsch. OWL 2 profiles: An introduction to lightweight ontology languages. In *Reasoning Web. Semantic Technologies for Advanced Query Answering - 8th International Summer School*, pages 112–183, 2012.

Jens Lehmann and Christoph Haase. Ideal downward refinement in the EL description logic. In *International Conference on Inductive Logic Programming (ILP)*, pages 73–87, 2009.

Jens Lehmann and Pascal Hitzler. Concept learning in description logics using refinement operators. *Machine Learning*, 78(1-2):203–250, 2010.

Jens Lehmann and Johanna Völker. *Perspectives on Ontology Learning*, volume 18. IOS Press, 2014.

Francesca A. Lisi. Al-quin: An onto-relational learning system for semantic web mining. *International Journal on Semantic Web and Information Systems*, 7(3):1–22, 2011.

Francesca A. Lisi and Umberto Straccia. Learning in description logics with fuzzy concrete domains. *Fundamenta Informaticae*, 140(3-4):373–391, 2015.

Carsten Lutz, Robert Piro, and Frank Wolter. Description logic TBoxes: Model-theoretic characterizations and rewritability. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 983–988, 2011.

Yue Ma and Felix Distel. Learning formal definitions for snomed ct from text. In *Artificial Intelligence in Medicine Conference (AIME)*, pages 73–77, 2013.

Ana Ozaki. *Exact Learning of Description Logic Ontologies*. PhD thesis, University of Liverpool, 2016.

Charles David Page Jr. *Anti-unification in constraint logics: foundations and applications to learnability in first-order logic, to speed-up learning, and to deduction*. PhD thesis, University of Illinois at Urbana-Champaign, 1993.

Peter F. Patel-Schneider, Deborah L. McGuinness, and Alexander Borgida. The CLASSIC knowledge representation system: Guiding principles and implementation rationale. *SIGART Bulletin*, 2(3):108–113, 1991.

Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Linking data to ontologies. *J. Data Semantics*, 10: 133–173, 2008.

Chandra Reddy and Prasad Tadepalli. Learning first-order acyclic Horn programs from entailment. *Inductive Logic Programming*, pages 23–37, 1998.

Sebastian Rudolph. Exploring relational structures via FLE. In *International Conference on Conceptual Structures, ICCS*, pages 196–212, 2004.

Stefan Schlobach, Zhisheng Huang, Ronald Cornet, and Frank Van Harmelen. Debugging incoherent terminologies. *Journal of Automated Reasoning*, 39(3):317–349, 2007.

Joseph Selman and Alan Fern. Learning first-order definite theories via object-based queries. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, pages 159–174, 2011.

JMLR 18(201):1-63, 2018

Heiner Stuckenschmidt, Christine Parent, and Stefano Spaccapietra, editors. *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, volume 5445 of *Lecture Notes in Computer Science*. Springer, 2009.

Balder ten Cate, Víctor Dalmau, and Phokion G. Kolaitis. Learning schema mappings. In *International Conference on Database Theory (ICDT)*, pages 182–195, 2012.

Johanna Völker and Mathias Niepert. Statistical schema induction. In *The Semantic Web: Research and Applications*, pages 124–138. Springer, 2011.

Johanna Völker, Daniel Fleischhacker, and Heiner Stuckenschmidt. Automatic acquisition of class disjointness. *Journal of Web Semantics*, 35:124–139, 2015.

Hai Wang, Matthew Horridge, Alan L. Rector, Nick Drummond, and Julian Seidenberg. Debugging OWL-DL ontologies: A heuristic approach. In *4th International Semantic Web Conference ISWC*, pages 745–757, 2005.

Wilson Wong, Wei Liu, and Mohammed Bennamoun. Ontology learning from text: A look back and into the future. *ACM Computing Surveys*, 44(4):20:1–20:36, 2012.

Journal of Machine Learning Research 18 (2018) 1-26 Submitted 3/17; Revised 11/17; Published 4/18

Sparse Concordance-assisted Learning for Optimal Treatment Decision

Shuhan Liang SLIANG4@NCSU.EDU
Department of Statistics
North Carolina State University
Raleigh, NC 27695, USA

Wenbin Lu LU@STAT.NCSU.EDU
Department of Statistics
North Carolina State University
Raleigh, NC 27695, USA

Rui Song RSONG@NCSU.EDU
Department of Statistics
North Carolina State University
Raleigh, NC 27695, USA

Lan Wang WANGX346@UMN.EDU
School of Statistics
University of Minnesota
Minneapolis, MN 55455, USA

Editor: Jie Peng

Abstract

To find optimal decision rule, Fan et al. (2016) proposed an innovative concordance-assisted learning algorithm which is based on maximum rank correlation estimator. It makes better use of the available information through pairwise comparison. However the objective function is discontinuous and computationally hard to optimize. In this paper, we consider a convex surrogate loss function to solve this problem. In addition, our algorithm ensures sparsity of decision rule and renders easy interpretation. We derive the L_2 error bound of the estimated coefficients under ultra-high dimension. Simulation results of various settings and application to STAR*D both illustrate that the proposed method can still estimate optimal treatment regime successfully when the number of covariates is large.

Keywords: concordance-assisted learning, optimal treatment regime, L_1 norm, support vector machine, variable selection.

1. Introduction

A treatment regime is a decision rule that tailors treatment for each individual. Instead of randomly assigning treatments, we can select a specific treatment among a few options for each patient based on his or her clinical, genetic and other health information. A decision rule is a procedure to decide which treatment should be picked and it is a function of available information for each patient. Optimal treatment regime aims to find the decision rule that would yield the most favorable outcome. Besides treatment type, treatments of interest also include different treatment combinations and dosage level variation. In reality, it often occurs that large number of patient level covariates are available. However, many of them have no qualitative interaction with treatment effect. Covariates may also be correlated with each other. Under such circumstances, variable selection for optimal treatment regime is necessary to avoid overfitting and increase model interpretation.

Many learning algorithms have been proposed to estimate optimal treatment regime (Qian et al. 2012). Watkins and Dayan (1992) modeled the conditional expectation of outcome (Q-function) and obtained the optimal treatment regime through maximizing the Q-function. Qian and Murphy (2011) extended Q-learning using l_1-penalized least square (PLS). The estimator derived from the two-step procedure may not be consistent if the conditional mean is misspecified. Instead of modeling the outcome, Murphy (2003) proposed the advantage learning (A-learning) algorithm, which is based on modeling contrast function. A contrast function is the difference in potential outcome given different treatments. Lu et al. (2011) considered model selection for estimating optimal treatment regime via penalized least square. A-learning is more robust than Q-learning since it does not require a correct specification of the baseline function but a correct model of interaction term is still needed. In literature, it is common to adopt parametric models for Q-function or contract function. As a result, the corresponding decision rule derived from Q-learning or A-learning may be biased.

To further reduce the impact of model misspecification, Zhang et al. (2012) proposed a value function estimator using inverse probability weighting. The optimal decision rule is derived by maximizing the value function estimator. Zhao et al. (2012) proposed the outcome weighted learning (OWL) algorithm. The OWL approximates optimal treatment decision estimation by transforming an objective function in Zhang et al. (2012) to a classification loss. Larger reward observed indicates higher chance that the optimal decision rule would recommend the same treatment as the patient actually received. Song et al. (2015) extended this method to penalized outcome weighted learning (POWL). Penalty functions include lasso (Tibshirani 1996) and SCAD (Fan and Li 2001).

Value search methods suffer from slow convergence and computation difficulties. Fan et al. (2016) proposed a novel concordance-assisted learning (CAL) algorithm to estimate optimal decision rule. Concordance function is motivated by maximizing value function using pairwise comparison between patients. Since concordance function can be estimated by a much smoother function, better asymptotic results can be obtained.

In this paper, we show that concordance-assisted learning algorithm can be transformed to a classification problem. We replace 0-1 loss by a continuous surrogate function. In order to improve the accuracy of optimal treatment regime and interpretation of decision rule, we conduct variable selection by adding lasso penalty to the objective function. We derive error bound of the proposed estimator under ultra-high dimension. We illustrate that the proposed estimator has better performance than existing popular methods under different scenarios together with a clinical trial study.

In Section 2, we reviewed and developed the concordance-assisted learning algorithm. We continued to derive the L_2 error bound of coefficient estimation in Section 3. Section 4 demonstrates the performance of sparse concordance-assisted learning at different settings. We present results of the proposed method for the STAR*D clinical trial in Section 5. The proofs of all lemmas and theorems are provided in the Appendix.

2. Method

In this section, we first introduce notations and explain its usage. It is followed by a concordance-assisted learning overview. We then propose the sparse concordance-assisted learning algorithm and provide an algorithm to calculate the proposed estimator using Douglas-Rachford splitting method.

2.1 Notation

Let $\boldsymbol{X}_i = (X_{i1}, X_{i2}, ..., X_{ip})^T$ denote the vector of covariates measured for the i-th patient, A_i the assigned treatment and Y_i the outcome after treatment. Let $\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_n)^T$ denote the feature matrix. Assume that $(\boldsymbol{X}_i, A_i, Y_i)$ are independent, identically distributed. Y is a continuous variable and larger value of Y indicates better treatment effect. Denote $g(\boldsymbol{X})$ as the individualized treatment regime (ITR), P^g as the joint distribution of $(\boldsymbol{X}, A = g(\boldsymbol{X}), Y)$, $E^g(Y)$ as the expected outcome if all treatments follow $g(\boldsymbol{X})$. From now on we consider the case of a binary treatment, i.e., A takes values in {0,1}. Denote $\mu(a, \boldsymbol{X}) = E(Y|A = a, \boldsymbol{X})$, Zhang et al. (2012) shows that $g^{opt}(\boldsymbol{X}) = I\{\mu(1, \boldsymbol{X}) > \mu(0, \boldsymbol{X})\}$. Here $g^{opt}(\boldsymbol{X})$ represents optimal treatment regime.

We also assume stable unit treatment value assumption (SUTVA) and no-unmeasured-confounders assumption holds. SUTVA(Rubin 1980), i.e., $Y = I(A = 0)Y^*(0) + I(A = 1)Y^*(1)$, assumes that no interference exists between treatments of different units and no same treatment variation exists for different units. $Y_i^*(a)$ is the potential outcome after receiving treatment a for subject i. The no-unmeasured-confounders condition, i.e., $\{Y_i^*(0), Y_i^*(1)\} \perp A_i|\boldsymbol{X}_i$, implies all variables that affect treatment assignment or treatment-specific outcomes are observed. The second assumption holds in a randomized trial.

2.2 Concordance-assisted Learning Overview

Concordance-assisted learning estimates optimal treatment regime by comparing the outcome gain of different treatments between individuals. Maximum Rank Correlation (MRC) estimator (Kendall 1938, Han 1987, Cavanagh and Sherman 1998) is chosen to estimate the concordance function. CAL further relaxes parametric assumptions and allows for more flexibility. Fan et al. (2016) showed that under certain conditions, optimal treatment regime estimated by concordance-assisted learning is the same as optimal treatment regime estimated by maximizing value functions.

The true optimal decision rule may not be linear, however, throughout the paper, we only search the optimal decision rule within the class of linear decision rules, i.e. $g(\boldsymbol{X}) = I(\boldsymbol{\beta}^T\boldsymbol{X} \geq \beta_0)$. This is partly because that linear decision rules are much easier to compute and interpret compared with nonlinear decision rules, and they generally can achieve high accuracy. CAL is a two-step procedure that first estimates the prescriptive index, i.e., a set of decision rules with fixed covariate weights by maximizing the concordance function:

$$C(\boldsymbol{\beta}) = E\Big\{\{[Y_i^*(1) - Y_i^*(0)] - [Y_j^*(1) - Y_j^*(0)]\}I(\boldsymbol{\beta}^T\boldsymbol{X}_i > \boldsymbol{\beta}^T\boldsymbol{X}_j)\Big\}$$

and then threshold estimator is optimized based on the prescriptive index estimator. Let $D(\boldsymbol{X}_i)$ be the expected outcome gain of treatment 1 for the i^{th} subject, i.e. $D(\boldsymbol{X}_i) = E(Y_i|A_i = 1, \boldsymbol{X}_i) - E(Y_i|A_i = 0, \boldsymbol{X}_i)$. Concordance function is motivated by the following

idea: for pairwise subjects i and j, larger $D(\boldsymbol{X}_i) - D(\boldsymbol{X}_j)$, which means subject i would benefit more by taking treatment 1 compared to subject j, requires larger $\boldsymbol{\beta}^T\boldsymbol{X}_i$ compared to $\boldsymbol{\beta}^T\boldsymbol{X}_j$.

Define $w_i = \frac{\{Y_i - \nu(\boldsymbol{X}_i)\}\{A_i - \pi(\boldsymbol{X}_i)\}}{\pi(\boldsymbol{X}_i)\{1-\pi(\boldsymbol{X}_i)\}}$,here $\nu(\boldsymbol{X}_i)$ is any arbitrary function and $\pi(\boldsymbol{X}_i) = P(A_i = 1|\boldsymbol{X}_i)$ is the propensity score. In practice we choose $\nu(\boldsymbol{X}_i)$ to be the mean response of the patients who receive treatment 0. Given $\boldsymbol{X}_i$, w_i is an unbiased estimator of $D(\boldsymbol{X}_i)$. The proof is given in Appendix C. Concordance-assisted learning can be summarized as follows:

1 Estimate the prescriptive index:

$$\hat{\boldsymbol{\beta}} = \underset{||\boldsymbol{\beta}||=1}{argmax} \frac{1}{n(n-1)} \sum_{i \neq j} (w_i - w_j) I(\boldsymbol{\beta}^T\boldsymbol{X}_i > \boldsymbol{\beta}^T\boldsymbol{X}_j).$$

2 Estimate the threshold using the inverse probability weighted estimator (IPW) proposed by Zhang et al. (2012):

$$\hat{\beta}_0 = \underset{\beta_0}{argmax} \frac{1}{n} \sum_{i=1}^{n} \frac{\{Y_i - \nu(\boldsymbol{X}_i)\} I\{A_i = \hat{g}(\boldsymbol{X}_i)\}}{A_i\pi(\boldsymbol{X}_i) + (1 - A_i)\{1 - \pi(\boldsymbol{X}_i)\}},$$
$$\hat{g}(\boldsymbol{X}_i) = I(\hat{\boldsymbol{\beta}}^T\boldsymbol{X}_i > \beta_0).$$

Although in general the concordance-based estimator does not always lead to the actual optimal decision rule, under certain conditions, concordance-based estimator is the maximizer of value function (Fan et al. 2016). Concordance-based estimator has attractive properties, including faster convergence rates, known asymptotic distribution (normal) and easy optimization. It is a very promising approach for optimal treatment regime estimation. In the next section, we will introduce sparse concordance-assisted learning (SCAL). Compared to CAL, it is easier to optimize and can achieve satisfactory accuracy under high dimension.

2.3 Sparse Concordance-assisted Learning

Notice that solving for $\hat{\boldsymbol{\beta}}$ is equivalent to minimize:

$$\frac{1}{n(n-1)} \sum_{i \neq j} (w_i - w_j) I(\boldsymbol{\beta}^T\boldsymbol{X}_i < \boldsymbol{\beta}^T\boldsymbol{X}_j), \quad \text{subject to } ||\boldsymbol{\beta}|| = 1. \tag{1}$$

(1) is equivalent to minimizing (see Appendix B):

$$\sum_{w_i > w_j} (w_i - w_j) I(\boldsymbol{\beta}^T\boldsymbol{X}_i < \boldsymbol{\beta}^T\boldsymbol{X}_j), \quad \text{subject to } ||\boldsymbol{\beta}|| = 1.$$

This alternative expression reduces computation cost and ensures the convexity of the objective function. We replace the indicator loss function with the hinge loss. Hinge loss is

a convex upper bound of the 0-1 loss function. It is often used for support vector machine (Cortes and Vapnik 1995), a popular classification method with good performance (Gordon 2004). The optimization of hinge loss function can be solved in polynomial time. Due to the high dimension of p, we use lasso penalty to estimate the optimal treatment regime and perform variable selection simultaneously. Lasso penalty also helps reduce the variance of the fitted coefficients (Zhu et al. 2004). The prescriptive index estimated by the sparse concordance-assisted learning algorithm (SCAL) is:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{2}{n(n-1)} \sum_{w_i > w_j} (w_i - w_j)\Big[1 - \boldsymbol{\beta}^T(\boldsymbol{X}_i - \boldsymbol{X}_j)\Big]_+ + \lambda \sum_{j=1}^{p} |\beta_j|.$$

We then estimate the threshold parameter β_0 by:

$$\hat{\beta}_0 = \underset{\beta_0}{\operatorname{argmax}} \frac{1}{n} \sum_{i=1}^{n} w_i I(\hat{\boldsymbol{\beta}}^T \boldsymbol{X}_i > \beta_0).$$

The threshold parameter β_0 is estimated through grid search. In practice, the search range is $[\min(\hat{\boldsymbol{\beta}}^T \boldsymbol{X}_i), \max(\hat{\boldsymbol{\beta}}^T \boldsymbol{X}_j)]$, $1 \le i, j \le n$.

We sort the subjects by descending order of w_i. Therefore,

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(w_i - w_j)I(w_i - w_j > 0) = \sum_{i=1}^{n}\sum_{j=i+1}^{n}(w_i - w_j). \tag{2}$$

The objective function of SCAL can be written as:

$$\hat{\boldsymbol{\beta}} = argmin_{\boldsymbol{\beta}} \frac{2}{n(n-1)} \sum_{i=1}^{\binom{n}{2}} \delta w_i (1 - \boldsymbol{\beta}^T \boldsymbol{D}_i)_+ + \lambda \sum_{j=1}^{p} |\beta_j|.$$

$$\boldsymbol{D} = \begin{pmatrix} \boldsymbol{X}_1^T - \boldsymbol{X}_2^T \\ \boldsymbol{X}_1^T - \boldsymbol{X}_3^T \\ \cdots \\ \boldsymbol{X}_1^T - \boldsymbol{X}_n^T \\ \boldsymbol{X}_2^T - \boldsymbol{X}_3^T \\ \boldsymbol{X}_2^T - \boldsymbol{X}_4^T \\ \cdots \end{pmatrix}, \quad \delta w = \begin{pmatrix} w_1 - w_2 \\ w_1 - w_3 \\ \cdots \\ w_1 - w_n \\ w_2 - w_3 \\ w_2 - w_4 \\ \cdots \end{pmatrix}.$$

The optimization problem in step 1 is a weighted L_1-SVM problem. The objective function is convex and piecewise linear and many algorithms have been proposed to solve this problem. It can be solved by various linear programming and convex packages. Zhu et al. (2004) proposed an algorithm to compute the whole solution path. Iterative algorithm like Spingarn's Method is another good way to solve this problem. We use three methods: CVX, a package for specifying and solving convex programs (Michael and Stephen 2014, Grant and Boyd 2008), GLPK (GNU Linear Programming Kit, glp 2016) and the method proposed by Spingarn (1985) to find the minimizer. Spingarn's method of partial inverses

implements Douglas-Rachford splitting for equality constrained convex problem (Douglas and Rachford 1956). We add ancillary variables $\boldsymbol{\theta}$ and reformulate (1) as:

$$min f_1(\boldsymbol{\beta}) + f_2(\boldsymbol{\theta})$$
$$subject\ to\ \boldsymbol{\theta} = \boldsymbol{D}\boldsymbol{\beta}$$

where $f_1(\boldsymbol{\beta}) = \lambda||\boldsymbol{\beta}||_1, f_2(\boldsymbol{\theta}) = \sum_{i=1}^{\binom{n}{2}} \delta w_i (1 - \boldsymbol{D}_i\boldsymbol{\beta})_+$.
The iterative algorithm is as follows:

Repeat

1. $\boldsymbol{V}_1^+ = \Big(prox_{tf_1}(\boldsymbol{\beta}), prox_{tf_2}(\boldsymbol{\theta})\Big)$, where
   $[prox_{tf_1}(\boldsymbol{\beta})]_j = S(\beta_j, t\lambda)$,
   S is soft thresholding operator: $S(x, \lambda) = sgn(x)(|x| - \lambda)_+$.
   $[prox_{tf_2}(\boldsymbol{\beta})]_i = \begin{cases} 1 & \theta \in [1 - t\delta w_i, 1], \\ \theta_i & \theta > 1, \\ \theta_i + t\delta w_i & \theta < 1 - t\delta w_i. \end{cases}$
2. $\boldsymbol{V}_2^+ = \begin{pmatrix} \boldsymbol{I} \\ \boldsymbol{D} \end{pmatrix} \boldsymbol{R}^{-T}\boldsymbol{R}^{-1}\Big[P_1(2\boldsymbol{V}_1^+ - \boldsymbol{V}_3) + \boldsymbol{D}^T P_2(2\boldsymbol{V}_1^+ - \boldsymbol{V}_3)\Big]$,
   $P_1(\boldsymbol{\beta}, \boldsymbol{\theta}) = \boldsymbol{\beta}$, $P_2(\boldsymbol{\beta}, \boldsymbol{\theta}) = \boldsymbol{\theta}$, $\boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{I} + \boldsymbol{D}^T\boldsymbol{D}$ is Cholesky decomposition.
3. $\boldsymbol{V}_3^+ = \boldsymbol{V}_3 + \boldsymbol{V}_2^+ - \boldsymbol{V}_1^+$.

until convergence.
Output: $\boldsymbol{\beta}$

We keep step-size parameter t fixed at 1. The convergence is guaranteed (Spingarn 1985). The iterative algorithm greatly reduces memory and time cost. Under the case of ultra-high dimension, preprocessing requires $O(p^3)$ work to form and compute Cholesky decomposition and $O(p^2)$ work per iteration. In summary, CVX is the least computational efficient way to estimate prescriptive index and Spingarn's Method is the only approach that can handle STAR*D trial in terms of its scale.

3. Error Bound for Order-2 U Statistics

Define $\boldsymbol{\beta}^* = argmin_{\boldsymbol{\beta}} L(\boldsymbol{\beta})$ where

$$L(\boldsymbol{\beta}) = E\Big\{(w_i - w_j)I(w_i - w_j > 0)[1 - (\boldsymbol{X}_i - \boldsymbol{X}_j)^T\boldsymbol{\beta}]_+\Big\}.$$

Then the gradient vector and Hessian matrix of the loss function $L(\boldsymbol{\beta})$ are:

$$S(\boldsymbol{\beta}) = -E\Big\{(w_i - w_j)I(w_i - w_j > 0)I[1 - (\boldsymbol{X}_i^T - \boldsymbol{X}_j^T)\boldsymbol{\beta} \ge 0](\boldsymbol{X}_i - \boldsymbol{X}_j)\Big\},$$
$$H(\boldsymbol{\beta}) = E\Big\{(w_i - w_j)I(w_i - w_j > 0)Dirac\delta[1 - (\boldsymbol{X}_i^T - \boldsymbol{X}_j^T)\boldsymbol{\beta}](\boldsymbol{X}_i - \boldsymbol{X}_j)(\boldsymbol{X}_i^T - \boldsymbol{X}_j^T)\Big\},$$

where $Dirac\delta$ is the Dirac delta function. Denote the index set of active features as $T = \{1 \le j \le p : \beta_j^* \neq 0\}$ and $|T| = q$. $\hat{\boldsymbol{\beta}}(\lambda) = \underset{\boldsymbol{\beta}}{argmin}\ l_n(\boldsymbol{\beta}, \lambda)$ is an estimator of $\boldsymbol{\beta}^*$, where

$$l_n(\boldsymbol{\beta}, \lambda) = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1}^{n} (w_i - w_j) I(w_i - w_j > 0)[1 - (\boldsymbol{X}_i^T - \boldsymbol{X}_j^T)\boldsymbol{\beta}]_+ + \lambda ||\boldsymbol{\beta}||_1.$$

We assume the following regularity conditions:

(A1) The densities of $\boldsymbol{X}_i$, i = 1, 2, ... are continuous and have common support in $\boldsymbol{R}^p$, and there exists a constant $M_0 > 0$ such that $|\boldsymbol{X}_{ij}| \le M_0$, $i \in R^+,\ j \in 1, ..., p$
(A2) Denote $\boldsymbol{Z}_{ij} = \boldsymbol{X}_i^T - \boldsymbol{X}_j^T$ with probability density function $f^*(z)$. There exists $B(0, \delta_0)$, a ball centered at 0 with radius $\delta_0 > 0$ such that $E[(w_i - w_j) I(w_i - w_j > 0) | \boldsymbol{Z}_{ij} = \boldsymbol{z}_{ij}] f^*(\boldsymbol{z}_{ij}) > C_3$ for every $\boldsymbol{z}_{ij} \in B(0, \delta_0)$.
(A3) $\int E[(w_i - w_j) I(w_i - w_j > 0) | \boldsymbol{z}] z_k f^*(\boldsymbol{z}) d\boldsymbol{z} \neq 0$ for some k.
(A4) There exists a constant M_1 s.t. $\underset{\boldsymbol{d} \in \boldsymbol{R}^p : ||\boldsymbol{d}||_0 \le 2q}{max} \frac{\boldsymbol{d}^T \boldsymbol{D}^T \boldsymbol{D} \boldsymbol{d}}{\frac{n(n-1)}{2} ||\boldsymbol{d}||_2^2} \le M_1$ almost surely.
(A5) Denote $\bar{c} = \frac{c-1}{c+1}$ where c is a constant satisfying $\lambda \ge c||S(\hat{\boldsymbol{\beta}}^*)||_\infty$, T is the set of significant coefficients (non-zero coefficients). There exists a constant $M_2 > 0$ such that $\underset{\boldsymbol{d} \in \boldsymbol{R}^p : ||d||_0 \le q, ||\boldsymbol{d}_T||_1 \ge \bar{c}||\boldsymbol{d}_{T^c}||_1}{min} \frac{\boldsymbol{d}^T H(\boldsymbol{\beta}^*) \boldsymbol{d}}{||\boldsymbol{d}||_2^2} \ge M_2$.
(A6) $q = O(n^{c_1})$ for some $0 \le c_1 < \frac{1}{2}$.
(A7) There exists a constant M_3 such that for any w_M, $P(|w_i| > w_M) < \exp(-\frac{w_M}{M_3})$.

Condition (A1) ensures $H(\boldsymbol{\beta})$ is well-defined and continuous in $\boldsymbol{\beta}$. Condition (A2) is similar to condition (A2) in Koo et al. (2008). It guarantees $L(\boldsymbol{\beta}) \to \infty$ as $||\boldsymbol{\beta}|| \to \infty$ and further guarantees the existence of $\boldsymbol{\beta}^*$. Condition (A3) implies that $\boldsymbol{\beta}^* \neq 0$. Condition (A4) gives the upper bound of restricted eigenvalue (RE). It can guarantee the Gram matrix is positive definite over a subset of vectors (Bickel et al. 2009). Condition (A5) gives the lower bound for restricted eigenvalue of $H(\boldsymbol{\beta}^*)$. Condition (A6) restricts the divergence rate for the number of none-zero variables. Condition (A7) is a popular distribution assumption in literature.

Lemma 1: *Assume condition (A1) and (A7) satisfied. Suppose* $\lambda = c\sqrt{32A(\alpha)(\log p)^3/n}$, *c is some given constant,* α *is a small probability and* $A(\alpha)$ *is a constant such that* $(2 + n)p^{-A(\alpha)^{1/3} M_0^{-2/3} M_3^{-2/3} + 1} \le \alpha$, *we have*

$$P(\lambda \ge c||\hat{S}(\boldsymbol{\beta}^*)||_\infty) \ge 1 - \alpha.$$

Lemma 2: *Assume conditions (A1), (A4), (A6) and (A7) are satisfied,* $p > n$. *Let*

$$\begin{aligned} B(h) = \frac{2}{n(n-1)} \Big| & \sum_{i=1}^{n} \sum_{j=1}^{n} (w_i - w_j) I(w_i - w_j > 0)[1 - (\boldsymbol{X}_i^T - \boldsymbol{X}_j^T)(\boldsymbol{\beta}^* - h)]_+ \\ & - \sum_{i=1}^{n} \sum_{j=1}^{n} (w_i - w_j) I(w_i - w_j > 0)[1 - (\boldsymbol{X}_i^T - \boldsymbol{X}_j^T)\boldsymbol{\beta}^*]_+ \\ & - \sum_{i=1}^{n} \sum_{j=1}^{n} E\{(w_i - w_j) I(w_i - w_j > 0)[1 - (\boldsymbol{X}_i^T - \boldsymbol{X}_j^T)(\boldsymbol{\beta}^* - h)]_+ \\ & - \sum_{i=1}^{n} \sum_{j=1}^{n} (w_i - w_j) I(w_i - w_j > 0)[1 - (\boldsymbol{X}_i^T - \boldsymbol{X}_j^T)\boldsymbol{\beta}^*]_+ \} \Big|. \end{aligned}$$

Then for sufficiently large n,

$$P\Big(\underset{||\boldsymbol{h}||_0 \le q, ||\boldsymbol{h}||_2 \neq 0}{sup} \frac{B(\boldsymbol{h})}{||\boldsymbol{h}||_2} \ge (1 + C_2\sqrt{M_1}) q \sqrt{\frac{32 \log p}{n}} [M_3 q(C_2^2 - 2) \log p + M_3 \log 2n] \Big) \le 3p^{-q(C_2^2 - 2)}$$

Lemma 2 guarantees that with high probability,

$$\frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1}^{n} (w_i - w_j) I(w_i - w_j > 0) \Big\{ [1 - (\boldsymbol{X}_i^T - \boldsymbol{X}_j^T)(\boldsymbol{\beta}^* - h)]_+ - [1 - (\boldsymbol{X}_i^T - \boldsymbol{X}_j^T)\boldsymbol{\beta}^*]_+ \Big\}.$$

is within a small range of its expectation. From now on, we choose h to be $h = \boldsymbol{\beta}^* - \hat{\boldsymbol{\beta}}(\lambda)$.

Lemma 3: *For* $\lambda \ge c||S(\hat{\boldsymbol{\beta}}^*)||_\infty$,

$$||\boldsymbol{h}_T||_1 \ge \bar{c}||\boldsymbol{h}_{T^c}||_1,$$

where $\bar{c} = \frac{c-1}{c+1}$, *T is the set of significant coefficients (non-zero coefficients) and* $|T| \le q$.

Theorem 4: *Suppose (A1) - (A7) hold, then* $\hat{\boldsymbol{\beta}}$ *satisfies*

$$||\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*||_2 \le \sqrt{1 + \frac{1}{\bar{c}}} [\frac{2\lambda\sqrt{q}}{M_2} + \frac{2C_4}{M_2} q^2 \sqrt{\frac{(\log p)^3}{n}} (\frac{5}{4} + \frac{1}{\bar{c}})]$$

with probability at least $1 - 3p^{-q(C_2^2 - 2) + 1}$, *where* C_4 *is a constant.*

When $\lambda = c\sqrt{32A(\alpha)(\log p)^3/n}$, the first term has order $\sqrt{(\log p)^3 q/n}$ and the second term has order $q^2\sqrt{(\log p)^3/n}$. Therefore, with high probability,

$$||\hat{\boldsymbol{\beta}}(\lambda) - \boldsymbol{\beta}^*||_2 = O_p(q^2\sqrt{(\log p)^3/n}).$$

The proofs of the lemmas and theorems are given in the Appendix. We first show the existence of $\boldsymbol{\beta}^*$ and it is non-trivial (formally stated and proved in Appendix A). Lemma 1

indicates a certain λ to bound the infinity norm of gradient vector with large probability. Lemma 2, and Lemma 3 are stepping stones on the path to proving Theorem 4. To be specific, Lemma 2 establishes the relationship between difference in loss functions and its expectation. Lemma 3. Similar results can be found in Zhang et al. (2016) and Peng et al. (2016). Compared to their proofs, our proof uses Hoeffding's inequality for u-statistics of order 2. Another difference is that an unbounded weight exists in our objective function. To handle this challenge, we make assumptions on the tail distribution of weights and adjust the bound correspondingly.

4. Simulation Studies

In this section, we demonstrate the numerical performance of the proposed method. We simulate data from a randomized experiment and evaluate the estimated optimal treatment regimes using sparse concordance-assisted learning and penalized outcome weighted learning. Objective function of POWL is:

$$\frac{1}{n}\sum_{i=1}^{n}\Big\{\frac{Y_i}{A_i\pi(\boldsymbol{X_i})+(1-A_i)[1-\pi(\boldsymbol{X_i})]}[1-(2A_i-1)g(\boldsymbol{X}_i)]_+\Big\}+\lambda\sum_{j=1}^{p}|\beta_j|. \qquad (3)$$

Here g is a linear function, i.e. $g(\boldsymbol{\beta},\boldsymbol{X}_i)=\boldsymbol{\beta}^T\boldsymbol{X}_i+\beta_0$. Notice that the solution to (2) will remain the same if every Y_i is added to a constant c. In order to guarantee the objective function is convex and the optimization problem is feasible, we add a constant to all Y_i to make sure the smallest response is positive. The constant is chosen so that the smallest shifted response is 0.01. POWL is implemented using convex toolbox in MATLAB. We compute the IPW estimator using Monte Carlo simulations with 1000 replicates and select the tuning parameter λ with the largest $\hat{Y}_{\hat{opt}}$.

To evaluate the estimated decision rule, we report the mean outcome following the estimated optimal treatment regime (Estimated Value) and the percentage of correct decision (PCD) of the estimated optimal treatment regime. The mean of value function following estimated treatment regime is calculated by plugging estimated decisions in the real model using Monte Carlo simulations with 1000 replicates. The mean of value function following the true optimal treatment regime (True Value) is also listed. In addition, we report the mean square error of $\hat{\boldsymbol{\beta}}$. For variable selection, we report correct number of zero coefficients (Corr0) and incorrect number of zero coefficients (Incorr0) compared to the true optimal treatment regime. Results are evaluated and compared under various settings. The associated sample standard deviations are included in the parentheses.

4.1 Low Dimension

We follow the first simulation scenario in Zhao et al. (2012): $X_{i1}, X_{i2}, ..., X_{i50}$ are generated independently from a uniform distribution on $[-1,1]$, $i=1,\cdots,n$. The treatment indicator A is generated from Bernoulli distribution with $p=0.5$. The conditional density of the response Y given $\boldsymbol{X}$ and A is normal, with mean $Q_0(\boldsymbol{X}_i)=1+2X_{i1}+X_{i2}+0.5X_{i3}+0.442(1-X_{i1}-X_{i2})(2A_i-1)$ and variance 1. Here only X_{i1} and X_{i2} have linear interaction

with treatment. We ran 100 simulations with n=30, 100 and 200 respectively to estimate the individualized treatment rule using SCAL and POWL. Table 1 summarizes the results.

From Table 1 we have the following observations. First, sparse concordance-assisted learning leads to more accurate estimates of $\boldsymbol{\beta}$ and better variable selection results. Sparse concordance-assisted learning achieves smaller mean square error (MSE) smaller Incorr0 and smaller Corr0 than that of penalized outcome weighted learning. Although the model size of POWL is smaller and closer to the real model size, its value function estimation is smaller. This further demonstrates that SCAL can select covariates that have strong interaction with treatment and compensate for the influence of model complexity.

The mean of value function following the estimated treatment regime gets closer to the real optimal value as sample size increases. We also notice that SCAL estimator does not vary much from sample to sample: both PCD and value function estimated by SCAL have smaller variance.

In general two methods lead to comparable results. The difference between methods are small, and this is especially true when the sample size is large. When n=30, SCAL leads to much closer value function estimation to true optimal value function estimation than that of POWL. But when n=200, the difference between value functions estimated from SCAL and POWL is only 1.39% of the true value function estimation. It is not surprising since compared to concordance-assisted learning, outcome weighted learning uses information less efficiently, which, can be made up of by increasing available information.

	n	MSE	Incorr0(0)	Corr0(48)	PCD	Estimated Value	True Value
POWL	30	1.60	1.70	42.23	0.615(0.02)	1.09(0.02)	1.44
	100	1.27	1.94	46.64	0.768(0.02)	1.27(0.02)	1.44
	200	1.09	1.99	47.78	0.786(0.02)	1.30(0.03)	1.44
SCAL	30	1.40	0.73	35.79	0.659(0.01)	1.16(0.01)	1.44
	100	0.52	0.11	41.97	0.764(0.01)	1.31(0.01)	1.44
	200	0.19	0.01	46.03	0.749(0.01)	1.32(0.01)	1.44

Table 1: Simulation results of sparse concordance-assisted learning (SCAL) and penalized outcome weighted learning (POWL): low-dimensional case

4.2 High Dimension

We consider the following six models to generate simulation data:

Model I: $\boldsymbol{Y}=\boldsymbol{X}\gamma_1+\boldsymbol{X}\boldsymbol{\beta}A+\epsilon$, $\gamma_1=(3,-1,1,\mathbf{0_{p-2}})^T$, $\boldsymbol{\beta}=(2,1.8,0,0,0,-1.6,\mathbf{0_{p-6}})^T$.

Model II: $\boldsymbol{Y}=3-0.5(\boldsymbol{X}\gamma_1)^2+0.625(\boldsymbol{X}\gamma_2)^2+\boldsymbol{X}\boldsymbol{\beta}A+\epsilon$, $\gamma_1=(1,0.5,\mathbf{0_{p-2}})^T$, $\gamma_2=(0,1,\mathbf{0_{p-2}})^T$,$\boldsymbol{\beta}=(2,1.8,0,0,0,-1.6,\mathbf{0_{p-6}})^T$.

Model III: $\boldsymbol{Y}=1-\sin(\boldsymbol{X}\gamma_1)+\sin(\boldsymbol{X}\gamma_2)+\boldsymbol{X}\boldsymbol{\beta}A+\epsilon$, $\gamma_1=(1,\mathbf{0_{p-1}})^T$,$\gamma_2=(0,1,\mathbf{0_{p-2}})^T$, $\boldsymbol{\beta}=(2,1.8,0,0,0,-1.6,\mathbf{0_{p-6}})^T$.

Model IV: $\boldsymbol{Y}=\boldsymbol{X}\gamma_1+(\boldsymbol{X}\boldsymbol{\beta})^3A+\epsilon$, $\gamma_1=(3,-1,1,\mathbf{0_{p-2}})^T$, $\boldsymbol{\beta}=(1,0.9,0,0,0,-0.8,\mathbf{0_{p-6}})^T$.

Model V: $\boldsymbol{Y}=3-0.5(\boldsymbol{X}\gamma_1)^2+0.625(\boldsymbol{X}\gamma_2)^2+(\boldsymbol{X}\boldsymbol{\beta})^3A+\epsilon$, $\gamma_1=(1,0.5,\mathbf{0_{p-2}})^T$, $\gamma_2=(0,1,\mathbf{0_{p-2}})^T$,$\boldsymbol{\beta}=(1,0.9,0,0,0,-0.8,\mathbf{0_{p-6}})^T$.

Model VI: $\boldsymbol{Y}=1-\sin(\boldsymbol{X}\gamma_1)+\sin(\boldsymbol{X}\gamma_2)+(\boldsymbol{X}\boldsymbol{\beta})^3A+\epsilon$, $\gamma_1=(1,\mathbf{0_{p-1}})^T$,$\gamma_2=(0,1,\mathbf{0_{p-2}})^T$, $\boldsymbol{\beta}=(1,0.9,0,0,0,-0.8,\mathbf{0_{p-6}})^T$.

There are three baseline functions: Models I and III share the same linear baseline function; Models II and IV share the same higher order polynomial baseline function; Models III and VI share the same complex baseline function. In the first three models there is a linear interaction between covariates and treatment; in the last three models treatment and a cubic function of prescriptive index are interacted with each other. All six models have the same important variables $\boldsymbol{X}_{i1}$, $\boldsymbol{X}_{i2}$ and $\boldsymbol{X}_{i6}$. Covariates $\boldsymbol{X}_i = (X_{i1}, X_{i2}, ..., X_{ip})^T$ are generated from a multivariate normal distribution: each entry is standard normal and the correlation between covariates is $Corr(X_{ij}, X_{ik}) = \rho^{|j-k|}$ for $1 \leq j \neq k \leq p$. ρ is chosen to be 0 and 0.2 respectively. The error term ϵ is generated from standard normal distribution. We ran 100 simulations for each scenario with n=100 and p=500, 1000 respectively.

We consider randomized studies where A is generated from Bernoulli distribution with p = 0.5. The performance of variable selection and treatment regime estimation of both methods are summarized in Tables 2 and 3. Conclusions are similar to low-dimensional case. SCAL selects more important variables and fewer unimportant variables than POWL. SCAL also provides more accurate decision rule estimation. Its value function estimate is 0.73 higher than POWL, which is 18.3% of true optimal value function estimate and its PCD is 22.4% higher. The comparison between MSE of $\boldsymbol{\beta}$ estimate further supports the advantage of SCAL.

The performance of SCAL continues to improve as magnitude of interaction between treatment and covariates increases. However we are unable to see this trend from POWL. In general, SCAL can recover important variables better under cubic prescriptive index and treatment interaction than under linear interaction. When $\rho = 0$ and $p = 500$, comparing Model 1 with Model 4, we can see that MSE and Incorr0 dropped a lot. On the contrary, for POWL under the same circumstances, MSE remains almost the same and Incorr0 even increases. Results of Corr0 as well as PCD agree with this statement.

SCAL has demonstrated its performance under high dimension. It also shows the potential to identify important variables when p goes even larger. Performance of variable selection and optimal treatment regime estimation become slightly worse when p increases from 500 to 1000. In reality it is common that hundreds of covariates are available for each patient. Reliable results can still be obtained by SCAL under such circumstances.

The PCD slightly increases when the correlation between covariates increases, suggesting that correlated covariate structure can reduce the impact of falsely selected unimportant variables and missing important variables. When ρ increases from 0 to 0.5, $p = 500$, the PCD of Model 1 increases 4.2% for SCAL and 0.9% for POWL. Due to the fact that correlation exists in most of the real-world data, SCAL proves itself to be a desirable approach.

Next, we consider observational studies where the propensity score is estimated from data. To be specific, the treatment indicator A is generated from Bernoulli($\frac{1}{1+e^{-u}}$), where $u(\boldsymbol{X}) = 0.01 - 0.5 * X_1 + 0.4 * X_{10}$. Here, we consider the same high-dimensional settings with $p = 500$ and $\rho = 0.2$, and the propensity score is estimated using the l_1-penalized logistic regression. The results of SCAL and POWL are summarized in Table 4. Overall, SCAL outperforms POWL in terms of variable selection and estimating optimal treatment regime in all cases as observed for randomization studies.

p	ρ	Model	MSE	Incorr0(0)	Corr0(497/997)	PCD	Estimated Value	True Value
500	0	Model 1	0.61	0.75	482.62	0.744(0.01)	3.80(0.02)	4.21
		Model 2	0.56	0.57	485.34	0.763(0.01)	3.79(0.03)	4.16
		Model 3	0.44	0.49	488.12	0.786(0.01)	1.92(0.02)	2.22
		Model 4	0.35	0.35	486.81	0.801(0.01)	5.67(0.04)	5.93
		Model 5	0.32	0.25	487.00	0.810(0.01)	5.63(0.05)	5.88
		Model 6	0.29	0.20	485.33	0.820(0.01)	3.74(0.04)	3.94
	0.2	Model 1	0.50	0.71	487.37	0.783(0.01)	4.00(0.02)	4.31
		Model 2	0.47	0.68	488.52	0.788(0.01)	3.86(0.02)	4.16
		Model 3	0.38	0.46	490.32	0.816(0.01)	2.08(0.02)	2.32
		Model 4	0.28	0.25	487.45	0.831(0.01)	6.53(0.02)	6.68
		Model 5	0.28	0.23	487.86	0.832(0.06)	6.36(0.05)	6.53
		Model 6	0.25	0.18	485.98	0.845(0.01)	4.56(0.02)	4.68
1000	0	Model 1	0.67	0.89	981.92	0.741(0.01)	3.81(0.02)	4.26
		Model 2	0.56	0.70	985.66	0.755(0.01)	3.84(0.03)	4.20
		Model 3	0.49	0.58	988.22	0.782(0.01)	1.93(0.01)	2.26
		Model 4	0.38	0.34	986.01	0.797(0.01)	5.70(0.03)	5.97
		Model 5	0.34	0.20	985.83	0.800(0.01)	5.68(0.04)	5.92
		Model 6	0.32	0.20	985.25	0.813(0.01)	3.76(0.03)	3.98
	0.2	Model 1	0.56	0.94	989.91	0.762(0.01)	4.01(0.02)	4.35
		Model 2	0.51	0.64	986.95	0.779(0.01)	3.88(0.02)	4.20
		Model 3	0.43	0.54	989.14	0.805(0.01)	2.09(0.02)	2.36
		Model 4	0.31	0.27	987.40	0.819(0.01)	6.55(0.02)	6.72
		Model 5	0.30	0.24	985.37	0.827(0.01)	6.41(0.02)	6.57
		Model 6	0.26	0.18	987.92	0.832(0.01)	4.59(0.02)	4.73

Table 2: Simulation results of sparse concordance-assisted learning (SCAL): high dimensional case

p	ρ	Model	MSE	Incorr0(0)	Corr0(497/997)	PCD	Estimated Value	True Value
500	0	Model 1	1.81	2.37	449.06	0.521(0.01)	3.03(0.02)	4.21
		Model 2	1.76	2.40	456.11	0.520(0.01)	2.98(0.01)	4.16
		Model 3	1.76	2.33	450.64	0.521(0.01)	1.04(0.01)	2.22
		Model 4	1.81	2.53	450.63	0.516(0.01)	3.00(0.05)	5.93
		Model 5	1.77	2.53	453.97	0.517(0.01)	2.95(0.05)	5.88
		Model 6	1.78	2.53	454.53	0.517(0.01)	1.02(0.06)	3.94
	0.2	Model 1	1.76	2.30	456.63	0.528(0.01)	3.08(0.02)	4.31
		Model 2	1.76	2.38	458.01	0.528(0.07)	2.92(0.02)	4.16
		Model 3	1.75	2.24	458.19	0.530(0.01)	1.09(0.02)	2.32
		Model 4	1.80	2.48	453.13	0.520(0.01)	3.13(0.07)	6.68
		Model 5	1.81	2.53	456.69	0.519(0.01)	2.98(0.07)	6.53
		Model 6	1.79	2.52	453.29	0.521(0.01)	1.15(0.08)	4.68
1000	0	Model 1	1.79	2.67	948.08	0.512(0.01)	3.04(0.01)	4.26
		Model 2	1.82	2.68	947.05	0.512(0.01)	2.99(0.01)	4.20
		Model 3	1.84	2.66	949.14	0.514(0.01)	1.06(0.01)	2.26
		Model 4	1.83	2.78	950.22	0.508(0.01)	2.88(0.04)	5.97
		Model 5	1.83	2.78	947.72	0.508(0.01)	2.86(0.04)	5.92
		Model 6	1.82	2.78	947.83	0.507(0.01)	0.91(0.04)	3.98
	0.2	Model 1	1.83	2.68	950.87	0.517(0.01)	3.06(0.02)	4.35
		Model 2	1.82	2.66	946.94	0.516(0.01)	2.91(0.01)	4.20
		Model 3	1.78	2.66	951.12	0.517(0.01)	1.08(0.02)	2.36
		Model 4	1.79	2.77	948.49	0.508(0.01)	2.93(0.05)	6.72
		Model 5	1.81	2.76	947.55	0.507(0.01)	2.76(0.05)	6.57
		Model 6	1.76	2.77	952.16	0.510(0.01)	0.96(0.05)	4.73

Table 3: Simulation results of penalized outcome weighted learning(POWL): high dimensional case

p=500	ρ	Model	MSE	Incorr0(0)	Corr0(497)	PCD	Estimated Value	True Value
SCAL	0.2	Model 1	0.82	0.85	484.96	0.732(0.01)	3.83(0.04)	4.33
		Model 2	0.73	0.77	487.52	0.744(0.01)	3.72(0.04)	4.18
		Model 3	0.67	0.70	489.03	0.768(0.01)	1.93(0.04)	2.34
		Model 4	0.41	0.47	483.77	0.807(0.01)	6.42(0.04)	6.70
		Model 5	0.32	0.31	483.05	0.815(0.01)	6.36(0.04)	6.55
		Model 6	0.36	0.41	486.40	0.818(0.01)	4.46(0.05)	4.71
POWL	0.2	Model 1	1.64	2.49	460.97	0.505(0.01)	2.99(0.01)	4.33
		Model 2	1.56	2.70	467.86	0.509(0.01)	2.87(0.01)	4.18
		Model 3	1.62	2.56	467.20	0.512(0.01)	1.04(0.01)	2.34
		Model 4	1.52	2.66	470.31	0.500(0.01)	2.92(0.04)	6.70
		Model 5	1.51	2.70	472.09	0.501(0.01)	2.79(0.04)	6.55
		Model 6	1.48	2.66	471.58	0.500(0.01)	0.93(0.05)	4.71

Table 4: Simulation results for observational studies: sparse concordance-assisted learning (SCAL) and penalized outcome weighted learning (POWL):

5. Application to STAR*D Study

We apply the proposed method to STAR*D Study, the largest and longest study ever conducted to assess effectiveness of depression treatments. 4041 outpatients who are diagnosed with major depressive disorder (MDD), representing of various ethnic and socioeconomic groups are collected. There are four levels in this clinical trial and at each level different treatments are evaluated and compared. See Fava et al. (2003) for design and measurement details of STAR*D study.

In the data analysis, we focus on patients who received bupropion (BUP) or sertraline (SER) in the second level to illustrate our method. Among the 309 selected subjects, 153 of them received bupropion (BUP) and 166 received sertraline (SER). In order to be consistent with our previous notation, we use 0 to represent SER and 1 to represent BUP. We consider all 305 covariates collected from enrollment, IVR call, ROA interviews, clinic visit and other events (such as suicide, non-serious adverse event and protocol deviation) to recommend individualized treatment for each patient. We choose negative 16-item Quick Inventory of Depressive Symptomatology-Clinician-Rated (QIDS-C16) as our response variable. QIDS-C16 is reverse coded so that it satisfies larger outcome indicates better treatment effect. The negative QIDS-C16 is in the range of -24 to 0.

We apply SCAL using Spingarn's method to this data set. λ_{opt} is tuned using 5-fold cross validation. A pre-defined range (0, 4) is searched and λ is chosen based on the IPW estimator. Propensity score is estimated by proportion of subjects who receive treatment 1 in the training data set. For comparison, we also evaluate the performance of POWL using 5-fold cross validation. The response is shifted with the smallest value to be 0.01. Note that the adjusted response is only used to optimize objective function; Value function is estimated using original response.

To compare the estimated treatment regimes on STAR*D data, we draw bootstrap samples over 1,000 times and estimate the 95% confident interval of difference between expected outcome following estimated treatment regime from SCAL and the non-dynamic treatment regimes. The expected outcome difference between SCAL and POWL is also calculated. See Table 5 for estimate of value function based on estimated optimal treatment regime and 95% confident interval of the differences.

Treatment Regime	Estimated Value	Diff	95% CI on Diff
Optimal Regime(SCAL)	-6.77		
Optimal Regime(POWL)	-9.46	2.69	(1.18, 4.24)
BUP	-10.52	3.75	(2.38, 5.19)
SER	-10.74	3.97	(2.57, 5.50)

Table 5: Estimated values, difference in estimated values and its 95% CI

The estimated value function by SCAL is significantly larger than either of the non-dynamic treatment regimes. It is also significantly larger than the expected outcome following the optimal treatment regime based on POWL at $\alpha = 0.05$. Compared to POWL, SCAL achieves a reasonable model size. It keeps good balance of including important features and controlling the complexity of the model. Estimate based on POWL is too sparse and many covariates which have interaction with treatment effects are missed. Table 6 and

Table 7 are summaries of treatment actually received versus estimated optimal treatment. We can see that estimated optimal treatment regime based on SCAL tends to be more balance. Based on SCAL, among all 171 subjects who were assigned to SER, 92 of them should stay in the same treatment and 79 of them should be assigned to BUP. While the result of POWL indicates 111 of them were assigned to the optimal treatment and only 56 patients should switch to BUP.

	Estimated treatment: SER	Estimated treatment: BUP
Randomized treatment: SER	92	75
Randomized treatment: BUP	79	73

Table 6: Summary of treatment recommended by SCAL

	Estimated treatment: SER	Estimated treatment: BUP
Randomized treatment: SER	111	56
Randomized treatment: BUP	106	46

Table 7: Summary of treatment recommended by POWL

6. Conclusion

We propose a variable selection method based on concordance-assisted learning for estimating optimal treatment regime. Our method can minimize the weighted misclassification rate and select prescriptive index simultaneously. The proposed method gives much more accurate decision rule and value function estimation than existing popular methods under various simulation settings. Moreover, inputs that are correlated with treatments effects are also successfully identified. Sparse concordance-assisted learning achieves promising result in constructing real-world decision. We also study the error bound of SCAL in ultra-high dimension.

The proposed method does not require model specification except for propensity score. It is based on an estimate of contrast function which can be defined easily under binary treatment circumstance. SCAL can solve problems in other fields as well. One popular example is to determine the best move in a game. SCAL can be implemented to choose the best action. In the future, one interesting direction of our study would be to extend the definition of contrast function when more than two treatment arms are available. We may also replace linear support vector machine by other kernels to see whether a better treatment regime can be found.

Acknowledgments

We would like to acknowledge support for this project from National Institute of Mental Health for providing the STAR*D data. We would also like to thank Dr. Eric Chi for his help in developing Spingarn's method.

Appendix A.

In this appendix we first state and prove two fundamental lemmas.

Lemma 0.1: *If condition (A1) and (A2) are satisfied, $\boldsymbol{\beta}^*$ exists.*

Proof. of Lemma 0.1:

$$\begin{aligned}
&E\Big\{(w_i-w_j)I(w_i-w_j>0)[1-\boldsymbol{\beta}^*(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)]_+\Big\}\\
=&E\Big(E\{(w_i-w_j)I(w_i-w_j>0)[1-\boldsymbol{\beta}^*(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)]_+|\boldsymbol{X}_i^T-\boldsymbol{X}_j^T\}\Big)
\end{aligned}$$

$$\begin{aligned}
L(\boldsymbol{\beta})=&\int E\Big[(w_i-w_j)I(w_i-w_j>0)|\boldsymbol{z}\Big](1-\boldsymbol{z}^T\boldsymbol{\beta})_+f^*(\boldsymbol{z})d\boldsymbol{z}\\
\geq&\int I(\boldsymbol{z}^T\boldsymbol{\beta}\leq 0)E\Big[(w_i-w_j)I(w_i-w_j>0)|\boldsymbol{z}\Big](1-\boldsymbol{z}^T\boldsymbol{\beta})f^*(\boldsymbol{z})d\boldsymbol{z}\\
\geq&\int I(\boldsymbol{z}^T\boldsymbol{\beta}\leq 0)E\Big[(w_i-w_j)I(w_i-w_j>0)|\boldsymbol{z}\Big](-\boldsymbol{z}^T\boldsymbol{\beta})f^*(\boldsymbol{z})d\boldsymbol{z}\\
\geq&C_3\int_{B(0,\delta_0)}I(\boldsymbol{z}^T\boldsymbol{\beta}\leq 0)(-\boldsymbol{z}^T\boldsymbol{\beta})d\boldsymbol{z}\\
=&C_3||\boldsymbol{\beta}||\int_{B(0,\delta_0)}I(\boldsymbol{z}^Tw\leq 0)(-\boldsymbol{z}^Tw)d\boldsymbol{z}\\
\geq&C_3||\boldsymbol{\beta}||vol\Big(B(0,\delta_0)\bigcap\{-\boldsymbol{z}^Tw\geq\epsilon\}\Big)\epsilon,
\end{aligned}$$

where $w=\boldsymbol{\beta}/||\boldsymbol{\beta}||$ and vol is short for volume. Note that $vol\Big(B(0,\delta_0)\bigcap\{-\boldsymbol{z}^Tw\geq\epsilon\}\Big)>0$ for some $\epsilon<\delta_0$ and $vol\Big(B(0,\delta_0)\bigcap\{-\boldsymbol{z}^Tw\geq\epsilon\}\Big)$ is independent of $\boldsymbol{\beta}$. Thus $L(\beta)\to\infty$ as $||\boldsymbol{\beta}||\to\infty$. Since $L(\boldsymbol{\beta})$ is convex in $\boldsymbol{\beta}$, the solution $\boldsymbol{\beta}^*$ exists.■

Lemma 0.2: *Condition (A3) implies $\boldsymbol{\beta}^*\neq 0$.*

Proof. of Lemma 0.2:

Without loss of generality, suppose $\int E[(w_i-w_j)I(w_i-w_j>0)|\boldsymbol{z}]z_kf^*(\boldsymbol{z})d\boldsymbol{z}>0$, then for $\beta_k^*>0$,

$$\begin{aligned}
L(0,...,\beta_k^*,0,...0)=&\int E[(w_i-w_j)I(w_i-w_j>0)|\boldsymbol{z}](1-\beta_k^*z_k)I(1-\beta_k^*z_k>0)f^*(\boldsymbol{z})d\boldsymbol{z}\\
=&\int E[(w_i-w_j)I(w_i-w_j>0)|\boldsymbol{z}]I(z_k<\frac{1}{\beta_k^*})f^*(\boldsymbol{z})d\boldsymbol{z}\\
&-\beta_k^*\int E[(w_i-w_j)I(w_i-w_j>0)|\boldsymbol{z}]z_kI(z_k<\frac{1}{\beta_k^*})f^*(\boldsymbol{z})d\boldsymbol{z}.
\end{aligned}$$

The second term is non-negative for a sufficient small $\beta_k^*>0$.
Therefore,

$$L(0,...,\beta_k^*,0,...0)<\int E[(w_i-w_j)I(w_i-w_j>0)|\boldsymbol{z}]f^*(\boldsymbol{z})d\boldsymbol{z}=L(0,...,0).\blacksquare$$

Now we prove all lemmas and theorems from Section 3.

Proof of Lemma 1: Recall that

$$\hat{S}(\boldsymbol{\beta})=\frac{-2}{n(n-1)}\sum_{i=1}^n\sum_{j=i+1}^n(w_i-w_j)I[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)\boldsymbol{\beta}\geq 0](\boldsymbol{X}_i-\boldsymbol{X}_j).$$

Note that

$$\begin{aligned}
&\frac{2}{n(n-1)}\sum_{i=1}^n\sum_{j=i+1}^n(w_i-w_j)I[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)\boldsymbol{\beta}\geq 0](X_{ik}-X_{jk})\\
&\qquad=\frac{1}{n(n-1)}\sum_{i\neq j}q_1\Big((w_i,\boldsymbol{X}_i),(w_j,\boldsymbol{X}_j)\Big),
\end{aligned}$$

where

$$q_1\Big((w_i,\boldsymbol{X}_i),(w_j,\boldsymbol{X}_j)\Big)=\begin{cases}(w_i-w_j)I[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)\boldsymbol{\beta}\geq 0](\boldsymbol{X}_{ik}-\boldsymbol{X}_{jk}) & if\ i<j,\\ q_1\Big((w_j,\boldsymbol{X}_j),(w_i,\boldsymbol{X}_i)\Big) & if\ i>j.\end{cases}$$

By Hoeffding's inequality for u-statistics of order 2, which can be found in Peel et al. (2010):

$$\begin{aligned}
P\Big(\sqrt{32A(\alpha)(\log p)^3/n}\leq&\frac{2}{n(n-1)}\Big|\sum_{i=1}^n\sum_{j=i+1}^n(w_i-w_j)I[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)\boldsymbol{\beta}\geq 0](X_{ik}-X_{jk})\Big|\Big)\\
&\leq 2\exp\Big(-\frac{32A(\alpha)n(\log p)^3}{2n(4M_0w_M)^2}\Big)+n\exp(-\frac{w_M}{M_3}).
\end{aligned}$$

Let $w_M=A(\alpha)^{1/3}\log pM_0^{-2/3}M_3^{1/3}$, then

$$\exp\Big(-\frac{32A(\alpha)n(\log p)^3}{32nM_0^2w_M^2}\Big)=\exp(-\frac{w_M}{M_3}),$$

$$\begin{aligned}
P\Big(\sqrt{32A(\alpha)(\log p)^3/n}\leq&\frac{2}{n(n-1)}\Big|\sum_{i=1}^n\sum_{j=i+1}^n(w_i-w_j)I[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)\boldsymbol{\beta}\geq 0](X_{ik}-X_{jk})\Big|\Big)\\
&\leq(2+n)p^{-A(\alpha)^{1/3}M_0^{-2/3}M_3^{-2/3}},
\end{aligned}$$

$$P\Big(c\sqrt{32A(\alpha)(\log p)^3/n} \le c||\hat{S}(\boldsymbol{\beta}^*)||_\infty\Big)$$
$$\le \sum_{k=1}^{p} P\Big(\sqrt{32A(\alpha)(\log p)^3/n} \le$$
$$\frac{2}{n(n-1)}\Big|\sum_{i=1}^{n}\sum_{j=i+1}^{n}(w_i-w_j)I[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)\boldsymbol{\beta}\ge 0](\boldsymbol{X}_{ik}-\boldsymbol{X}_{jk})\Big|\Big)$$
$$\le (2+n)p^{-A(\alpha)^{1/3}M_0^{-2/3}M_3^{-2/3}+1} \le \alpha.\blacksquare$$

Proof of Lemma 2:
Let $q_2\Big((w_i,\boldsymbol{X}_i),(w_j,\boldsymbol{X}_j)\Big) =$

$$\begin{cases}(w_i-w_j)[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)(\boldsymbol{\beta}^*-h)]_+-(w_i-w_j)[1-(X_i^T-X_j^T)\boldsymbol{\beta}^*]_+ & if\ i<j,\\ q_2\Big((w_j,\boldsymbol{X}_j),(w_i,\boldsymbol{X}_i)\Big) & if\ i>j.\end{cases}$$

and $\hat{U}_{q_2}(\mathcal{W}_n,\boldsymbol{\mathcal{X}}_n)=\frac{1}{n(n-1)}\sum_{i\neq j}q_2\Big((w_i,\boldsymbol{X}_i),(w_j,\boldsymbol{X}_j)\Big)$. It is a u-statistics of order 2.

$$\Big|(w_i-w_j)[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)(\boldsymbol{\beta}^*-h)]_+-(w_i-w_j)[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)\boldsymbol{\beta}^*]_+\Big| \le \Big|2w_M(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)h\Big|$$

holds with at least probability $1-ne^{-\frac{w_M}{M_3}}$, $\forall\ 1\le i,j\le n$.

By Hoeffding's inequality,

$$P\Big(\sup_{h\in R^p:||\boldsymbol{h}||_0\le q,||\boldsymbol{h}||_2\neq 0}\frac{B(\boldsymbol{h})}{||\boldsymbol{h}||_2}\ge\frac{t}{\sqrt{n}}|\boldsymbol{X}\Big)\le 2\exp(-\frac{t^2}{32M_0^2w_M^2q})$$

holds with at least probability $1-ne^{-\frac{w_M}{M_3}}$, $\forall\ 1\le i,j\le n$. Let $t=C\sqrt{32\log p}qw_M$, then

$$P\Big(\sup_{h\in R^p:||\boldsymbol{h}||_0\le q,||\boldsymbol{h}||_2\neq 0}\frac{B(\boldsymbol{h})}{||\boldsymbol{h}||_2}\ge C\sqrt{\frac{32\log p}{n}}qw_M\Big)\le 2p^{-\frac{qC^2}{M_0^2}},$$
$$P\Big(\sup_{\boldsymbol{h}\in N}\frac{B(\boldsymbol{h})}{||\boldsymbol{h}||_2}\ge C\sqrt{\frac{32\log p}{n}}qw_M\Big)\le 2(\frac{3}{\epsilon}p^{1-\frac{C^2}{M_0^2}})^q,$$

where N is a ϵ-net to cover $\{\boldsymbol{h}\in R^p,||\boldsymbol{h}||_0\le q,||\boldsymbol{h}||_2\neq 0\}$, for any $\boldsymbol{h}_1$, $\boldsymbol{h}_2$ within the same ϵ-ball and $||\boldsymbol{h}_1||_2\neq 0$ and $||\boldsymbol{h}_2||_2\neq 0$, $\Big|\frac{\boldsymbol{h}_1}{||\boldsymbol{h}_1||_2}-\frac{\boldsymbol{h}_2}{||\boldsymbol{h}_2||_2}\Big|\le\epsilon$ holds.

We also have

$$\sup_{\boldsymbol{h}_1,\boldsymbol{h}_2\in R^p,||\boldsymbol{h}_1-\boldsymbol{h}_2||_0\le 2q,||\boldsymbol{h}_1||_2\neq 0,||\boldsymbol{h}_2||_2\neq 0}\Big|\frac{B(\boldsymbol{h}_1)}{||\boldsymbol{h}_1||_2}-\frac{B(\boldsymbol{h}_2)}{||\boldsymbol{h}_2||_2}\Big|\le\frac{8w_M}{n(n-1)}\Big|\Big|D(\frac{\boldsymbol{h}_1}{||\boldsymbol{h}_1||_2}-\frac{\boldsymbol{h}_2}{||\boldsymbol{h}_2||_2})\Big|\Big|_1$$
$$\le\frac{8w_M}{\sqrt{\frac{n(n-1)}{2}}}\Big|\Big|D(\frac{\boldsymbol{h}_1}{||\boldsymbol{h}_1||_2}-\frac{\boldsymbol{h}_2}{||\boldsymbol{h}_2||_2})\Big|\Big|_2\le 8w_M\sqrt{M_1}\epsilon$$

holds with probability at least $1-ne^{-\frac{w_M}{M_3}}$.

$$\sup_{||\boldsymbol{h}||_0\le q,||\boldsymbol{h}||_2\neq 0}\frac{B(\boldsymbol{h})}{||\boldsymbol{h}||_2}\le\sup_{\boldsymbol{h}\in N}\frac{B(\boldsymbol{h})}{||\boldsymbol{h}||_2}+8w_M\sqrt{M_1}\epsilon$$

holds with probability at least $1-ne^{-\frac{w_M}{M_3}}$.

Let $\epsilon=q\sqrt{\frac{32\log p}{n}}\frac{1}{8\sqrt{M_1}}$, we have that

$$P\Big(\sup_{||\boldsymbol{h}||_0\le q,||\boldsymbol{h}||_2\neq 0}\frac{B(\boldsymbol{h})}{||\boldsymbol{h}||_2}\ge Cq\sqrt{\frac{32\log p}{n}}w_M\Big)\le$$
$$P\Big(\sup_{\boldsymbol{h}\in N}\frac{B(\boldsymbol{h})}{||\boldsymbol{h}||_2}\ge(C-1)q\sqrt{\frac{32\log p}{n}}w_M\Big)+2n\exp(-\frac{w_M}{M_3}).$$

Since $p>n$ and take $C=1+C_2M_0$ for some $C_2\ge\sqrt{2}$, for sufficiently large n:

$$P\Big(\sup_{||\boldsymbol{h}||_0\le q,||\boldsymbol{h}||_2\neq 0}\frac{B(\boldsymbol{h})}{||\boldsymbol{h}||_2}\ge(1+C_2M_0)q\sqrt{\frac{32\log p}{n}}w_M\Big)\le 2p^{-q(C_2^2-2)}+\exp(\log 2n-\frac{w_M}{M_3}).$$

Take $w_M=M_3q(C_2^2-2)\log p+M_3\log 2n$, we have that $P\Big(\sup_{||\boldsymbol{h}||_0\le q,||\boldsymbol{h}||_2\neq 0}\frac{B(\boldsymbol{h})}{||\boldsymbol{h}||_2}\ge(1+C_2M_0)q\sqrt{\frac{32\log p}{n}}[M_3q(C_2^2-2)\log p+M_3\log 2n]\Big)\le 3p^{-q(C_2^2-2)}.\blacksquare$

Proof of Lemma 3.
By the definition of $\boldsymbol{\beta}^*$, we have

$$\begin{aligned}&\frac{2}{n(n-1)}\sum_{i=1}^{n}\sum_{j=i+1}^{n}(w_i-w_j)[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)\hat{\boldsymbol{\beta}}]_+ +\lambda||\hat{\boldsymbol{\beta}}||_1\le\\ &\frac{2}{n(n-1)}\sum_{i=1}^{n}\sum_{j=i+1}^{n}(w_i-w_j)[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)\boldsymbol{\beta}^*]_+ +\lambda||\boldsymbol{\beta}^*||_1\\ &\frac{2}{n(n-1)}\sum_{i=1}^{n}\sum_{j=i+1}^{n}(w_i-w_j)[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)\hat{\boldsymbol{\beta}}]_+\\ &-\frac{2}{n(n-1)}\sum_{i=1}^{n}\sum_{j=i+1}^{n}(w_i-w_j)[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)\boldsymbol{\beta}^*]_+\le\lambda||\boldsymbol{\beta}^*||_1-\lambda||\hat{\boldsymbol{\beta}}||_1.\end{aligned}\tag{4}$$

We can also show that

$$||\boldsymbol{\beta}^*||_1-||\hat{\boldsymbol{\beta}}||_1\le||\boldsymbol{h}_T||_1-||\boldsymbol{h}_{T^c}||_1,\tag{5}$$

$$\frac{2}{n(n-1)}\sum_{i=1}^{n}\sum_{j=i+1}^{n}(w_i-w_j)[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)\hat{\boldsymbol{\beta}}]_+$$
$$-\frac{2}{n(n-1)}\sum_{i=1}^{n}\sum_{j=i+1}^{n}(w_i-w_j)[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)\boldsymbol{\beta}^*]_+$$
$$\geq \hat{S}^T(\boldsymbol{\beta}^*)\boldsymbol{h}\geq -||\boldsymbol{h}||_1||\hat{S}(\boldsymbol{\beta}^*)||_\infty\geq -\frac{\lambda}{c}(||\boldsymbol{h}_T||_1+||\boldsymbol{h}_{T^c}||_1). \quad (6)$$

Based on (4),(6) and (6), we have

$$||\boldsymbol{h}_T||_1\geq\frac{c-1}{c+1}||\boldsymbol{h}_{T^c}||_1.\blacksquare$$

Proof. of Theorem 4:
Assume $|h_1|\geq|h_2|\geq ...\geq|h_p|$, then for the partition: $S_0=\{1,2,..,q\},S_1=\{q+1,q+2,..,2q\},...,$

$$||\boldsymbol{h}_{S_0}||_1\geq||\boldsymbol{h}_T||_1\geq\bar{c}||\boldsymbol{h}_{T^C}||_1\geq\bar{c}||\boldsymbol{h}_{S_0^C}||_1$$

holds. We have

$$\frac{2}{n(n-1)}\sum_{i=1}^{n}\sum_{j=i+1}^{n}(w_i-w_j)[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)(\boldsymbol{\beta}^*-\boldsymbol{h})]_+-$$
$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}(w_i-w_j)[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)\boldsymbol{\beta}^*]_+$$
$$=\frac{2}{n(n-1)}\sum_{i=1}^{n}\sum_{j=i+1}^{n}(w_i-w_j)[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)(\boldsymbol{\beta}^*-\boldsymbol{h}_{S_0})]_+-$$
$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}(w_i-w_j)[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)\boldsymbol{\beta}^*]_+$$
$$+\frac{2}{n(n-1)}\sum_{l\geq1}\Big\{\sum_{i=1}^{n}\sum_{j=i+1}^{n}(w_i-w_j)[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)(\boldsymbol{\beta}^*-\sum_{k=0}^{l}\boldsymbol{h}_{S_k})]_+$$
$$-\sum_{i=1}^{n}\sum_{j=i+1}^{n}(w_i-w_j)[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)(\boldsymbol{\beta}^*-\sum_{k=0}^{l-1}\boldsymbol{h}_{S_k})]_+\Big\}.$$

By Lemma 2 we have with at least probability $1-3p^{-q(C_2^2-2)}$,

$$E\Big\{(w_i-w_j)I(w_i-w_j>0)[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)(\boldsymbol{\beta}^*-\sum_{k=1}^{l}\boldsymbol{h}_{S_k})]_+\Big\}$$
$$-E\Big\{(w_i-w_j)I(w_i-w_j>0)[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)(\boldsymbol{\beta}^*-\sum_{k=1}^{l-1}\boldsymbol{h}_{S_k})]_+\Big\}$$
$$\leq\frac{2}{n(n-1)}\sum_{i=1}^{n}\sum_{j=i+1}^{n}(w_i-w_j)[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)(\boldsymbol{\beta}^*-\sum_{k=1}^{l}\boldsymbol{h}_{S_k})]_+$$
$$-\frac{2}{n(n-1)}\sum_{i=1}^{n}\sum_{j=i+1}^{n}(w_i-w_j)[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)(\boldsymbol{\beta}^*-\sum_{k=1}^{l-1}\boldsymbol{h}_{S_k})]_++C_4q^2\sqrt{\frac{(\log p)^3}{n}}||\boldsymbol{h}_{S_l}||_2.$$

It holds for every l, therefore

$$\frac{2}{n(n-1)}E\Big\{\sum_{i=1}^{n}\sum_{j=i+1}^{n}(w_i-w_j)[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)(\boldsymbol{\beta}^*-\boldsymbol{h})]_+-$$
$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}(w_i-w_j)[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)\boldsymbol{\beta}^*]_+\Big\}$$
$$\leq\frac{2}{n(n-1)}\sum_{i=1}^{n}\sum_{j=i+1}^{n}(w_i-w_j)[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)(\boldsymbol{\beta}^*-\boldsymbol{h})]_+$$
$$-\frac{2}{n(n-1)}\sum_{i=1}^{n}\sum_{j=i+1}^{n}(w_i-w_j)[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)\boldsymbol{\beta}^*]_++C_4q^2\sqrt{\frac{(\log p)^3}{n}}\sum_{j\geq0}||\boldsymbol{h}_{S_j}||_2$$
$$\leq\lambda(||\boldsymbol{h}_T||_1-||\boldsymbol{h}_{T^C}||_1)+C_4q^2\sqrt{\frac{(\log p)^3}{n}}||\boldsymbol{h}_{S_0}||_2+\sum_{j\geq1}C_4q^2\sqrt{\frac{(\log p)^3}{n}}||\boldsymbol{h}_{S_j}||_2$$
$$\leq\lambda\sqrt{q}||\boldsymbol{h}_{S_0}||_2+C_4q^2\sqrt{\frac{(\log p)^3}{n}}(\frac{5}{4}+\frac{1}{\bar{c}})||\boldsymbol{h}_{S_0}||_2$$

holds with probability at least $1-3p^{-q(C_2^2-2)+1}$.

The last inequality holds by,

$$\sum_{j\geq1}||\boldsymbol{h}_{S_j}||_2\leq\sum_{j\geq1}\frac{||\boldsymbol{h}_{S_j}||_1}{\sqrt{q}}+\frac{\sqrt{q}}{4}|h_q|\leq\frac{||\boldsymbol{h}_{S_0^C}||_1}{\sqrt{q}}+\frac{||\boldsymbol{h}_{S_0}||_1}{4\sqrt{q}}$$
$$\leq(\frac{1}{\sqrt{q}\bar{c}}+\frac{1}{4\sqrt{q}})||\boldsymbol{h}_{S_0}||_1\leq(\frac{1}{4}+\frac{1}{\bar{c}})||\boldsymbol{h}_{S_0}||_2.$$

And more details can be found in Cai et al. (2010).
By Taylor expansion of $L(\boldsymbol{\beta})$ around $\boldsymbol{\beta}^*$:

$$\begin{aligned}\frac{2}{n(n-1)}E\Big\{&\sum_{i=1}^{n}\sum_{j=i+1}^{n}(w_i-w_j)[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)(\boldsymbol{\beta}^*-\boldsymbol{h})]_+-\\&\sum_{i=1}^{n}\sum_{j=i+1}^{n}(w_i-w_j)[1-(\boldsymbol{X}_i^T-\boldsymbol{X}_j^T)\boldsymbol{\beta}^*]_+\Big\}\\=&\frac{1}{2}\boldsymbol{h}^TH(\boldsymbol{\beta}^*)\boldsymbol{h}+o_p(||\boldsymbol{h}||_2^2)\geq\frac{1}{2}M_2||\boldsymbol{h}||_2^2+o_p(||\boldsymbol{h}||_2^2).\end{aligned}$$

Then we have

$$\frac{1}{2}M_2||\boldsymbol{h}||_2^2+o_p(||\boldsymbol{h}||_2^2)\leq\lambda\sqrt{q}||\boldsymbol{h}_{S_0}||_2+C_4q^2\sqrt{\frac{(\log p)^3}{n}}(\frac{5}{4}+\frac{1}{\bar{c}})||\boldsymbol{h}_{S_0}||_2.$$

On the other hand,

$$\begin{aligned}||\boldsymbol{h}||_2^2=&||\mathbf{h}_{S_0}||_2^2+\sum_{j\geq1}||\boldsymbol{h}_{S_j}||_2^2\leq||\boldsymbol{h}_{S_0}||_2^2+|h_q|\sum_{j\geq1}||\boldsymbol{h}_{S_j}||_1\\\leq&||\boldsymbol{h}_{S_0}||_2^2+\frac{1}{\bar{c}}||\boldsymbol{h}_{S_0}||_1|h_q|\leq(1+\frac{1}{\bar{c}})||\boldsymbol{h}_{S_0}||_2^2.\end{aligned}$$

Therefore,

$$\begin{aligned}&||\boldsymbol{h}_{S_0}||_2^2\leq||\boldsymbol{h}||_2^2\leq(1+\frac{1}{\bar{c}})||\boldsymbol{h}_{S_0}||_2^2,\\&o(||\boldsymbol{h}||_2^2)=o(||\boldsymbol{h}_{S_0}||_2^2).\end{aligned}$$

Hence

$$\begin{aligned}&\frac{1}{2}M_2||\boldsymbol{h}_{S_0}||_2+o_p(||\boldsymbol{h}_{S_0}||_2)\leq\lambda\sqrt{q}+C_4q^2\sqrt{\frac{(\log p)^3}{n}}(\frac{5}{4}+\frac{1}{\bar{c}}),\\&||\boldsymbol{h}||_2+o_p(||\boldsymbol{h}||_2)\leq\sqrt{1+\frac{1}{\bar{c}}}[\frac{2\lambda\sqrt{q}}{M_2}+\frac{2C_4}{M_2}q^2\sqrt{\frac{(\log p)^3}{n}}(\frac{5}{4}+\frac{1}{\bar{c}})].\end{aligned}$$

That is,

$$||\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}^*||_2\leq\sqrt{1+\frac{1}{\bar{c}}}[\frac{2\lambda\sqrt{q}}{M_2}+\frac{2C_4}{M_2}q^2\sqrt{\frac{(\log p)^3}{n}}(\frac{5}{4}+\frac{1}{\bar{c}})]$$

with probability at least $1-3p^{-q(C_2^2-2)+1}$.■

Appendix B.

We now show that

$$\begin{aligned}&min\sum_{i\neq j}(w_i-w_j)I(\boldsymbol{\beta}^T\boldsymbol{X}_i<\boldsymbol{\beta}^T\boldsymbol{X}_j),\\&\text{subject to }||\boldsymbol{\beta}||=1.\end{aligned}$$

is equivalent to

$$\begin{aligned}&min\sum_{w_i>w_j}(w_i-w_j)I(\boldsymbol{\beta}^T\boldsymbol{X}_i<\boldsymbol{\beta}^T\boldsymbol{X}_j),\\&\text{subject to }||\boldsymbol{\beta}||=1.\end{aligned}$$

Suppose $\boldsymbol{\beta}^T\boldsymbol{X}_i\neq\boldsymbol{\beta}^T\boldsymbol{X}_j,\forall i,j$, we have

$$\begin{aligned}&\sum_{i\neq j}(w_i-w_j)I(\boldsymbol{\beta}^T\boldsymbol{X}_i<\boldsymbol{\beta}^T\boldsymbol{X}_j)\\&\sum_{w_i>w_j}(w_i-w_j)I(\boldsymbol{\beta}^T\boldsymbol{X}_i<\boldsymbol{\beta}^T\boldsymbol{X}_j)+\sum_{w_i>w_j}(w_j-w_i)I(\boldsymbol{\beta}^T\boldsymbol{X}_j<\boldsymbol{\beta}^T\boldsymbol{X}_i)\\&\sum_{w_i>w_j}(w_i-w_j)[I(\boldsymbol{\beta}^T\boldsymbol{X}_i-\boldsymbol{\beta}^T\boldsymbol{X}_j<0)-I(\boldsymbol{\beta}^T\boldsymbol{X}_i-\boldsymbol{\beta}^T\boldsymbol{X}_j>0)]\\&\sum_{w_i>w_j}(w_i-w_j)[I(\boldsymbol{\beta}^T\boldsymbol{X}_i-\boldsymbol{\beta}^T\boldsymbol{X}_j<0)-1+I(\boldsymbol{\beta}^T\boldsymbol{X}_i-\boldsymbol{\beta}^T\boldsymbol{X}_j<0)]\\&\sum_{w_i>w_j}(w_i-w_j)[2I(\boldsymbol{\beta}^T\boldsymbol{X}_i-\boldsymbol{\beta}^T\boldsymbol{X}_j<0)-1].\blacksquare\end{aligned}$$

Appendix C.

We now show that given $\boldsymbol{X}_i$, w_i is an unbiased estimator of $D(\boldsymbol{X}_i)$. Specifically,

$$\begin{aligned}&E\{\frac{[Y_i-\nu(\boldsymbol{X}_i)][A_i-\pi(\boldsymbol{X}_i)]}{\pi(\boldsymbol{X}_i)[1-\pi(\boldsymbol{X}_i)]}|\boldsymbol{X}_i\}\\=&E\left\{\frac{[Y_i-\nu(\boldsymbol{X}_i)][1-\pi(\boldsymbol{X}_i)]}{\pi(\boldsymbol{X}_i)[1-\pi(\boldsymbol{X}_i)]}|A_i=1,\boldsymbol{X}_i\right\}\pi(\boldsymbol{X}_i)\\+&E\left\{\frac{[Y_i-\nu(\boldsymbol{X}_i)][-\pi(\boldsymbol{X}_i)]}{\pi(\boldsymbol{X}_i)[1-\pi(\boldsymbol{X}_i)]}|A_i=0,\boldsymbol{X}_i\right\}[1-\pi(\boldsymbol{X}_i)]\\=&E[Y_i|A_i=1,\boldsymbol{X}_i]-E[Y_i|A_i=0,\boldsymbol{X}_i]=D(\boldsymbol{X}_i).\blacksquare\end{aligned}$$

References

Gnu linear programming kit,version 4.58. `http://www.gnu.org/software/glpk/glpk.html`, February 2016.

P. Bickel, Y. Ritov, and A. Tsybakov. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009.

T. Cai, L. Wang, and G. Xu. New bounds for restricted isometry constants. *Information Theory, IEEE Transactions on*, 56(9):4388–4394, 2010.

C. Cavanagh and R. Sherman. Rank estimators for monotonic index models. *Journal of Econometrics*, 84(2):351–381, 1998.

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

J. Douglas and H. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American mathematical Society*, 82 (2):421–439, 1956.

C. Fan, W. Lu, R. Song, and Y. Zhou. Concordance-assisted learning for estimating optimal individualized treatment regimes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2016.

J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 10(1):1348–1360, 2001.

M. Fava, J. Rush, M. H Trivedi, A. Nierenberg, M. Thase, F. Sackeim, H.and Quitkin, S. Wisniewski, P. Lavori, J. Rosenbaum, et al. Background and rationale for the sequenced treatment alternatives to relieve depression (STAR* D) study. *Psychiatric Clinics of North America*, 26(2):457–494, 2003.

G. Gordon. Support vector machines and kernel methods. *Lecture*, 2004.

M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In *Recent Advances in Learning and Control*, pages 95–110. Springer, 2008.

A. Han. Non-parametric analysis of a generalized regression model: the maximum rank correlation estimator. *Journal of Econometrics*, 35(2):303–316, 1987.

M. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

J. Koo, Y. Lee, Y. Kim, and C. Park. A bahadur representation of the linear support vector machine. *Journal of Machine Learning Research*, 9(Jul):1343–1368, 2008.

W. Lu, H. Zhang, and D. Zeng. Variable selection for optimal treatment decision. *Statistical methods in medical research*, 22(5):493–504, 2011.

G. Michael and B. Stephen. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, March 2014.

S. Murphy. Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(2):331–355, 2003.

T. Peel, S. Anthoine, and L. Ralaivola. Empirical Bernstein inequalities for u-statistics. In *Advances in Neural Information Processing Systems*, pages 1903–1911, 2010.

B. Peng, L. Wang, and Y. Wu. An error bound for L1-norm support vector machine coefficients in ultra-high dimension. *Journal of Machine Learning Research*, 17(236): 1–26, 2016.

M. Qian and S. Murphy. Performance guarantees for individualized treatment rules. *Annals of statistics*, 39(2):1180–1210, 2011.

M. Qian, I. Nahum-Shani, and S. Murphy. Dynamic treatment regimes. In *Modern Clinical Trial Analysis*, pages 127–148. Springer, 2012.

D. Rubin. Randomization analysis of experimental data: The Fisher randomization test comment. *Journal of the American Statistical Association*, 75(371):591–593, 1980.

R. Song, W. Wang, D. Zeng, and M. Kosorok. Penalized q-learning for dynamic treatment regimens. *Statistica Sinica*, 25(3):901, 2015.

J. Spingarn. Applications of the method of partial inverses to convex programming: decomposition. *Mathematical Programming*, 32(2):199–223, 1985.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

C. Watkins and P. Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

B. Zhang, A. Tsiatis, E. Laber, and M. Davidian. A robust method for estimating optimal treatment regimes. *Biometrics*, 68(4):1010–1018, 2012.

X. Zhang, Y. Wu, L. Wang, and R. Li. Variable selection for support vector machines in moderately high dimensions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(1):53–76, 2016.

Y. Zhao, D. Zeng, J. Rush, and M. Kosorok. Estimating individualized treatment rules using outcome weighted learning. *Journal of the American Statistical Association*, 107 (499):1106–1118, 2012.

J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani. 1-norm support vector machines. *Advances in neural information processing systems*, 16(1):49–56, 2004.

Journal of Machine Learning Research 18 (2018) 1-78 Submitted 3/17; Revised 11/17; Published 4/18

Post-Regularization Inference for Time-Varying Nonparanormal Graphical Models

Junwei Lu JUNWEIL@PRINCETON.EDU
Department of Operations Research and Financial Engineering
Princeton University
Princeton, NJ 08544, USA

Mladen Kolar MKOLAR@CHICAGOBOOTH.EDU
Booth School of Business
The University of Chicago
Chicago, IL 60637, USA

Han Liu HANLIU@PRINCETON.EDU
Department of Operations Research and Financial Engineering
Princeton University
Princeton, NJ 08544, USA

Editor: Kenji Fukumizu

Abstract

We propose a novel class of time-varying nonparanormal graphical models, which allows us to model high dimensional heavy-tailed systems and the evolution of their latent network structures. Under this model we develop statistical tests for presence of edges both locally at a fixed index value and globally over a range of values. The tests are developed for a high-dimensional regime, are robust to model selection mistakes and do not require commonly assumed minimum signal strength. The testing procedures are based on a high dimensional, debiasing-free moment estimator, which uses a novel kernel smoothed Kendall's tau correlation matrix as an input statistic. The estimator consistently estimates the latent inverse Pearson correlation matrix uniformly in both the index variable and kernel bandwidth. Its rate of convergence is shown to be minimax optimal. Our method is supported by thorough numerical simulations and an application to a neural imaging data set.

Keywords: graphical model selection, nonparanormal graph, time-varying network analysis, hypothesis test, regularized rank-based estimator

1. Introduction

We consider the problem of inferring time-varying undirected graphical models from high dimensional non-Gaussian distributions. Undirected graphical models have been widely used as a powerful tool for exploring the dependency relationships between variables. We are interested in graphical models which have non-static graphical structures and can handle heavy-tail distributions as well as data contaminated with outliers. To that end, we develop a class of time-varying nonparanormal models, which can be used to explore Markov dependencies of a random vector $\boldsymbol{X}$ given the index variable Z. Specifically, we assume the random variables $(\boldsymbol{X}, Z)$ follow the following joint distribution: the conditional distribution

of $\boldsymbol{X} \mid Z = z$ follows a nonparanormal distribution

$$\boldsymbol{X} \mid Z = z \sim \mathsf{NPN}_d(\mathbf{0}, \boldsymbol{\Sigma}(z), f) \tag{1}$$

where $f = \{f_1, \ldots, f_d\}$ is a set of d univariate, strictly increasing functions and Z is a random variable with a continuous density. A variable follows a nonparanormal distribution $\boldsymbol{Y} \sim \mathsf{NPN}_d(\boldsymbol{\mu}, \boldsymbol{\Sigma}, f)$ if $f(\boldsymbol{Y}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ (Liu et al., 2009). Graphical modeling with nonparanormal distribution is studied in Liu et al. (2009), Liu et al. (2012a) and Xue and Zou (2012), however, their graph structure is static. Time-varying graphical models are studied in Talih and Hengartner (2005), Xuan and Murphy (2007), Zhou et al. (2010), Kolar and Xing (2011), Kolar and Xing (2012), Yin et al. (2010), Kolar et al. (2010b), Ahmed and Xing (2009), Kolar et al. (2010a) and Kolar and Xing (2009). However, these papers assume that conditionally on the index, the distribution of $\boldsymbol{X}$ is parametric, which is not adequate for applications to heavy-tailed data sets in finance, neuroscience and genomics (Qiu et al., 2016). Moreover, inferential methods for the time-varying graphical models have not been developed so far.

Primary motivation for proposing the model in (1) and developing corresponding estimation and inferential procedures comes from an application in neuroscience. Graphical models are widely used to estimate and explore functional connectivity between different brain regions from functional magnetic resonance imaging (fMRI) data (Wang et al., 2010; Bullmore and Bassett, 2011; Smith et al., 2011). There is evidence that that the brain connectivity network evolves over time (Bartzokis et al., 2001; Gelfand et al., 2003) and current techniques are not adequate for capturing evolving nature of brain networks. For example, work of Kolar et al. (2010a) assumes that data are Gaussian, which is rarely satisfied in practice. Qiu et al. (2016) need replicated observations at each time point, which are not available in most of the time-varying fMRI data sets. Furthermore, current procedures are solely focused on estimation of networks, while the question of inference and quantification of uncertainty is left unanswered. We address these drawbacks in the current work.

The focus of the paper is on the inferential analysis about parameters in the model given in (1), as well as the Markov dependencies between observed variables. The inference procedures we develop are uniformly valid in a high-dimensional regime and robust to model selection mistakes. In particular, the inference does not depend on the oracle support recovery properties of the estimator. As a foundation for inference, we develop an estimation procedure for the high-dimensional latent time-varying inverse correlation matrix based on a novel kernel smoothed Kendall's tau statistic. The estimator is uniformly consistent in both the bandwidth and the index variable, and furthermore is optimal in a minimax sense. Obtaining rate of convergence for the estimator is technically challenging and requires development of new uniform bounds for the U-processes, careful characterization of the leading terms in the expansion of the estimator in the presence of high-dimensionality and approximation errors arising from the local approximation of nonlinear curves. The proof that the rate of convergence is optimal involves application of Le Cam's method on a carefully chosen function valued high dimensional matrices class. These technical details are novel and of independent interest, as discussed in Section 4.2.

We consider three types of hypothesis tests: (1) the edge presence test for whether there is an edge in the Markov graph at z_0, (2) the super graph test for whether the true graph is a subgraph of a fixed graph at z_0, and (3) the uniform edge presence test for whether

the true graph is a subgraph of a given graph over a range of index values. The first test was studied in the context of static Gaussian graphical models in Janková and van de Geer (2015, 2017) and Ren et al. (2015), however, their approach cannot handle time-varying models. The second test was considered in Wasserman et al. (2014), Yang et al. (2014b) and Gu et al. (2015) also in the context of static graphical models. Cai et al. (2013b) considered a statistical test for whether the correlation matrix is an identity, which is a special case of the super graph test. The third test is a generalization of the above two local tests to a global test over a range of index values, which allows for identifying whether certain connections in graphical models exist for a period of time. We illustrate the super graph test in an application to the ADHD-200 data set containing fMRI data from subjects with and without attention deficit hyperactive disorder (ADHD) (Biswal et al., 2010), which allows us to uncover how the brain networks change with age.

This paper makes two major contributions to the literature on statistical inference for graphical models. First, we develop a general inferential procedure for a wide family of high dimensional graphical model estimation methods. Many existing high dimensional inference methods are specifically designed for concrete estimators. For example, Zhang and Zhang (2013), van de Geer et al. (2014), Javanmard and Montanari (2014), and Ning and Liu (2017) design inference procedures for specific M-estimators, while Neykov et al. (2015) developed an inferential procedure for the method of moments estimators like Dantzig selector (Candés and Tao, 2007) and CLIME (Cai et al., 2011). Barber and Kolar (2018) design a procedure for constructing confidence intervals in high-dimensional elliptical copula models. In contrast to that, we propose a nonparametric score-type statistic, which uses any estimator of $\mathbf{\Sigma}(z)^{-1}$ with fast enough rate of convergence as an input. Therefore, our inference procedure does not depend on a particular estimate of $\mathbf{\Sigma}(z)^{-1}$ and can be applied to both M-estimators, like graphical Lasso, and method of moments estimators, like CLIME. Second, to the best of our knowledge, this paper considers for the first time presence of the edges test uniformly over the index z for high dimensional graphical models. Computing quantiles of the test statistic requires development a new Gaussian multiplier bootstrap procedure for a U-process.

1.1 Related Literature

High-dimensional Gaussian graphical models are studied in Meinshausen and Bühlmann (2006), Yuan and Lin (2007), Rothman et al. (2008), Friedman et al. (2008), d'Aspremont et al. (2008), Fan et al. (2009), Lam and Fan (2009), Yuan (2010), Cai et al. (2011), Liu and Wang (2017), and Zhao and Liu (2014), with an extension to covariate-adjusted graphical models given in Dondelinger et al. (2010), Li et al. (2012), Cai et al. (2013a), Chen et al. (2016), and Yin and Li (2013). Semiparametric extensions using copulas are developed in Liu et al. (2009), Liu et al. (2012a), Xue and Zou (2012), and Liu et al. (2012b), and extended for mixed data in Fan et al. (2015). Guo et al. (2011a), Guo et al. (2011b), Lee and Hastie (2015), Cheng et al. (2017), Yang et al. (2012), and Yang et al. (2014a) study the mixed exponential family graphical models where a node conditional distribution is a member of an exponential family distribution. Danaher et al. (2014), Qiu et al. (2016), Mohan et al. (2014) consider joint estimation of multiple graphical models.

All of the above mentioned work assumes that the graphical structure is static. However, in analysis of many complex systems, such an assumption is not valid. There are two major types of time-varying graphical model: directed and undirected. The directed time-varying graphical models are mainly studied in the context of autoregressive models with time-varying parameters (Punskaya et al., 2002; Fujita et al., 2007; Rao et al., 2007; Grzegorczyk and Husmeier, 2011; Song et al., 2009; Robinson and Hartemink, 2010; Jia and Huan, 2010; Lébre et al., 2010; Husmeier et al., 2010; Wang et al., 2011; Grzegorczyk and Husmeier, 2012; Dondelinger et al., 2013; Lébre et al., 2010). For the time-varying undirected graphical models, Zhou et al. (2010), Kolar and Xing (2011), Yin et al. (2010), Kolar et al. (2010b) and Kolar et al. (2010a) consider the kernel- smoothed type estimator for graphical models. Kolar and Xing (2012) assume the graphical model evolves in a piecewise-constant fashion and estimate it by the temporally smoothed ℓ_1 penalized regression. Talih and Hengartner (2005) and Xuan and Murphy (2007) consider a Bayesian framework to model the time-varying of graphs and estimate the graph by Markov chain Monte Carlo methods. All the above works show the statistical rates for graphical models for fixed time points. Our work contributes to this literature by studying the uniform properties of estimators and developing inferential procedures. For estimation, we prove a rate of convergence that is uniform over z and show it matches the minimax rate. Instead of the kernel smoothed sample covariance, we propose the kernel smoothed U-statistics as a robust estimator. For inference, we study the presence of edges for a range of index values instead of the local tests in the literature.

Our paper also contributes to the literature on high dimensional inference. Hypothesis testing and confidence intervals for the high dimensional M-estimators are studied in Zhang and Zhang (2013), van de Geer et al. (2014); Javanmard and Montanari (2014), Belloni et al. (2013), Belloni et al. (2016), Javanmard and Montanari (2014) and Meinshausen (2015). Lu et al. (2015) considered the confidence bands for the high dimensional nonparametric models. Neykov et al. (2015) proposed the inference for high dimensional method of moments estimators. Lee et al. (2016) and Tibshirani et al. (2016) consider the conditional inference based on post-selection methods. Our work considers a new inferential framework involving both discrete graph structures and the nonparametric index variable, which provides more flexibility in the modeling of modern data sets.

Finally, we develop novel probabilistic tools to study the high dimensional U-statistics. Classical analysis for fixed dimensional U-statistics is built on the Hoeffding decomposition (Hoeffding, 1948). Concentration inequalities for high dimensional U-statistics are studied in Giné et al. (2000) and Adamczak (2006). However, existing methods based on uniform entropy numbers are too loose to be applicable (Nolan and Pollard, 1987). We develop a new peeling method to control suprema of our kernel smoothed U-process uniformly over three aspects: dimension, index variable and bandwidth. The uniform consistency over the bandwidth shown in the paper also generalizes a data-driven bandwidth-tuning method to U-statistics from the kernel-type estimator considered in Einmahl and Mason (2005). This provides more flexibility in the tuning procedure of our method. Moreover, to study the limiting distribution of the U-statistics, we generalize the Gaussian multiplier bootstrap proposed in Chernozhukov et al. (2013) and Chernozhukov et al. (2014a) to nonparametric U-process by considering a new type of nonlinear Gaussian multiplier U-processes.

1.2 Notation

Let $[n]$ denote the set $\{1, \ldots, n\}$ and let $\mathbb{1}\{\cdot\}$ denote the indicator function. For a vector $a \in \mathbb{R}^d$, we let $\text{supp}(a) = \{j \,:\, a_j \neq 0\}$ be the support set (with an analogous definition for matrices $A \in \mathbb{R}^{n_1 \times n_2}$), $||a||_q$, $q \in [1, \infty)$, the ℓ_q-norm defined as $||a||_q = (\sum_{i\in[n]} |a_i|^q)^{1/q}$ with the usual extensions for $q \in \{0, \infty\}$, that is, $||\mathbf{a}||_0 = |\text{supp}(\mathbf{a})|$ and $||\mathbf{a}||_\infty = \max_{i\in[n]} |a_i|$. For a matrix $\mathbf{A} \in \mathbb{R}^{n_1 \times n_2}$, we use the notation $\text{vec}(A)$ to denote the vector in $\mathbb{R}^{n_1 n_2}$ formed by stacking the columns of $\mathbf{A}$. We write $\mathbf{A} = [\mathbf{A}_{jk}]$ if the (j,k)-th entry of $\mathbf{A}$ is $\mathbf{A}_{jk}$. Let $\boldsymbol{A}_{\setminus j,k}$ be the sub-vector of the k-th column $\mathbf{A}$ with $\mathbf{A}_{jk}$ removed. We denote the Frobenius norm of $\mathbf{A}$ by $||\mathbf{A}||_F^2 = \sum_{i\in[n_1],j\in[n_2]} \mathbf{A}_{ij}^2$, the max-norm $\|\mathbf{A}\|_{\max} = \max_{i\in[n_1],j\in[n_2]} |\mathbf{A}_{ij}|$, the ℓ_1-norm $\|\mathbf{A}\|_1 = \max_{j\in[n_2]} \sum_{i\in[n_1]} |\mathbf{A}_{ij}|$, and the operator norm $\|\mathbf{A}\|_2 = \sup_{\|\mathbf{v}\|_2=1} \|\mathbf{A}\mathbf{v}\|_2$. The Hadamard product of two matrices is the matrix $\mathbf{A} \circ \mathbf{B}$ with elements $(\mathbf{A} \circ \mathbf{B})_{jk} = \mathbf{A}_{jk} \cdot \mathbf{B}_{jk}$. Given two functions f and g, we denote their convolution as $(f * g)(t) = \int f(t-x)g(x)dx$. For $1 \leq p < \infty$, let $\|f\|_p = (\int f^p)^{1/p}$ denote the L^p-norm of f and $\|f\|_\infty = \sup_x |f(x)|$. The total variation of f is defined as $\text{TV}(f) = \|f'\|_1$. For two sequences of numbers $\{a_n\}_{n=1}^\infty$ and $\{b_n\}_{n=1}^\infty$, we use $a_n = O(b_n)$ or $a_n \lesssim b_n$ to denote that $a_n \leq C b_n$ for some finite positive constant C, and for all n large enough. If $a_n = O(b_n)$ and $b_n = O(a_n)$, we use the notation $a_n \asymp b_n$. The notation $a_n = o(b_n)$ is used to denote that $a_n/b_n \to 0$ as n goes infinity. We also define $a \vee b = \max(a, b)$ and $a \wedge b = \min(a, b)$ for any two scalars a and b. We use $\xrightarrow{P}$ for convergence in probability and $\rightsquigarrow$ for convergence in distribution. Throughout the paper, we let c, C be two generic absolute constants, whose values will vary at different locations.

We use the notation $\mathbb{E}_n[\cdot]$ to denote the empirical average, $\mathbb{E}_n[f] = n^{-1} \sum_{i\in[n]} f(X_i)$. We also use $\mathbb{G}_n[f] = \sqrt{n}\left(\mathbb{E}_n[f(X_i)] - \mathbb{E}[f(X_i)]\right) = n^{-1/2} \sum_{i\in[n]} \left(f(X_i) - \mathbb{E}[f(X_i)]\right)$. For a bivariate function $H(x, x')$, we define the U-statistic $\mathbb{U}_n[H] = [n(n-1)]^{-1} \sum_{i\neq i'} H(X_i, X_{i'})$.

Appendix J collects all the notation in a table format with reference to where they appear first.

2. Preliminaries

We start by providing background on the nonparanormal distribution and discuss how it relates to the time-varying nonparanormal graphical model in (1). The nonparanormal distribution was introduced in Liu et al. (2009). A random variable $\boldsymbol{X} = (X_1, \ldots, X_d)^T$ is said to follow a nonparanormal distribution if there exists a set of monotone univariate functions $f = \{f_1, \ldots, f_d\}$ such that $f(\boldsymbol{X}) := (f_1(X_1), \ldots, f_d(X_d))^T \sim N(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is a latent correlation matrix satisfying $\text{diag}(\boldsymbol{\Sigma}) = 1$. We denote $\boldsymbol{X} \sim \mathsf{NPN}_d(\mathbf{0}, \boldsymbol{\Sigma}, f)$.

Given n independent copies of $\boldsymbol{X} \sim \mathsf{NPN}_d(\mathbf{0}, \boldsymbol{\Sigma}, f)$, Liu et al. (2012a) study how to estimate the latent correlation matrix $\boldsymbol{\Sigma}$. The key idea lies in relating the Kendall's tau correlation matrix with the Pearson correlation. The Kendall's tau correlation between X_j and X_k, two coordinates of $\boldsymbol{X}$, is defined as

$$\tau_{jk} = \mathbb{E}\left[\text{sign}\left((X_j - \widetilde{X}_j)(X_k - \widetilde{X}_k)\right)\right],$$

where $(\widetilde{X}_j, \widetilde{X}_k)$ is an independent copy of (X_j, X_k). It can be related to the latent correlation matrix using the fact that $\tau_{jk} = (2/\pi) \arcsin\left(\boldsymbol{\Sigma}_{jk}\right)$ when $\boldsymbol{X}$ follows a nonparanormal distribution (Fang et al., 1990). The inverse covariance matrix $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$ encodes the graph

structure of a nonparanormal distribution (Liu et al., 2009). Specifically $\boldsymbol{\Omega}_{jk} = 0$ if and only if X_j is independent of X_k conditionally on $\boldsymbol{X}_{\setminus\{j,k\}}$.

The above observations lead naturally to the following estimation procedure for $\boldsymbol{\Omega}$. We estimate the Kendall's tau correlation matrix $\widehat{\mathbf{T}} = [\widehat{\tau}_{jk}] \in \mathbb{R}^{d\times d}$ elementwise using the following U-statistic

$$\widehat{\tau}_{jk} = \frac{2}{n(n-1)} \sum_{1\leq i<i'\leq n} \text{sign}(X_{ij} - X_{i'j})\,\text{sign}(X_{ik} - X_{i'k}).$$

An estimate of the latent correlation matrix is given as $\widehat{\boldsymbol{\Sigma}} = \sin\left(\pi\widehat{\mathbf{T}}/2\right)$, where $\sin(\cdot)$ is applied elementwise. Finally, the estimate of the latent correlation matrix $\widehat{\boldsymbol{\Sigma}}$ is used as a plug-in statistic in the CLIME estimator (Cai et al., 2011), or calibrated CLIME estimator (Zhao and Liu, 2014), to obtain the inverse covariance estimator $\widehat{\boldsymbol{\Omega}}$.

The CLIME estimator solves the following optimization program

$$\widehat{\boldsymbol{\Omega}}_j^{\text{CLIME}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^d}{\text{argmin}} \|\boldsymbol{\beta}\|_1 \qquad \text{subject to} \qquad \|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta} - \mathbf{e}_j\|_\infty \leq \lambda, \tag{2}$$

where $\mathbf{e}_j$ is the j-th canonical basis in $\mathbb{R}^d$ and the penalty parameter λ that controls the sparsity of the resulting estimator is commonly chosen as $\lambda \asymp \|\boldsymbol{\Omega}\|_1 \sqrt{\log d/n}$ (Cai et al., 2011). Note that the tuning parameter depends on the unknown $\boldsymbol{\Omega}$ through $\|\boldsymbol{\Omega}\|_1$, which makes practical selection of λ difficult. The calibrated CLIME is a tuning-insensitive estimator, which alleviates this problem. The calibrated CLIME estimator solves

$$\left(\widehat{\boldsymbol{\Omega}}_j^{\text{CCLIME}}, \widehat{\kappa}_j\right) = \underset{\boldsymbol{\beta}\in\mathbb{R}^d, \kappa\in\mathbb{R}}{\text{argmin}} \|\boldsymbol{\beta}\|_1 + \gamma\kappa \qquad \text{subject to} \qquad \|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta} - \mathbf{e}_j\|_\infty \leq \lambda\kappa,\ \|\boldsymbol{\beta}\|_1 \leq \kappa, \tag{3}$$

where γ is any constant in $(0, 1)$ and the tuning parameter can be chosen as $\lambda = C\sqrt{\log d/n}$ with C being a universal constant independent of the problem parameters. In what follows, we will adapt the calibrated clime to estimation of the parameters of the model in (1).

2.1 Time-Varying Nonparanormal Graphical Model

The time-varying nonparanormal graphical model in (1) is an extension of the nonparanormal distribution. For every fixed value of the index variable $Z = z$, we have a static nonparanormal distribution $\boldsymbol{X} \mid Z = z \sim \mathsf{NPN}_d(\mathbf{0}, \boldsymbol{\Sigma}(z), f)$ that can be easily interpreted. However, as the index variable changes, the conditional distribution of $\boldsymbol{X} \mid Z$ can change in an unspecified way. In this sense, time-varying nonparanormal graphical models extend nonparanormal graphical models in the same way varying coefficient models extend linear regression models.

Let $Y = (\boldsymbol{X}, Z)$ denote a random pair distributed according to the time-varying nonparanormal distribution. Specifically $Z \sim f_Z(z)$ with $f_Z(\cdot)$ being a continuous density function supported on $[0, 1]$ and $\boldsymbol{X} \mid Z = z \sim \mathsf{NPN}_d(\mathbf{0}, \boldsymbol{\Sigma}(z), f)$ for all $z \in [0, 1]$. For any fixed $z \in [0, 1]$, we denote the inverse of the correlation matrix as $\boldsymbol{\Omega}(z) = \boldsymbol{\Sigma}^{-1}(z)$. Both $f_Z(z)$ and each entry of $\boldsymbol{\Omega}(z)$ are second-order differentiable (we will formalize assumptions in Section 4). We denote the undirected graph encoding the conditional independence of $\boldsymbol{X} \mid Z = z$ as $G^*(z) = (V, E^*(z))$, with $(j, k) \in E^*(z)$ when $\boldsymbol{\Omega}_{jk}(z) \neq 0$. As in the static

case, we relate the Kendall's tau correlation matrix with the latent correlation matrix. Let $\mathbf{T}(z) = [\tau_{jk}(z)]_{jk}$ be the Kendall's tau correlation matrix corresponding to $\boldsymbol{X} \mid Z = z$ with elements defined as

$$\tau_{jk}(z) = \mathbb{E}\left[\operatorname{sign}\left((X_j - \widetilde{X}_j)(X_k - \widetilde{X}_k)\right) \mid Z = z\right],$$

where $\widetilde{\boldsymbol{X}}$ is an independent copy of $\boldsymbol{X}$ conditionally on $Z = z$. Given n independent copies of $Y = (\boldsymbol{X}, Z)$, $\{Y_i = (\boldsymbol{X}_i, Z_i)\}_{i\in[n]}$, we estimate an element of the Kendall's tau correlation matrix using the following kernel estimator

$$\widehat{\tau}_{jk}(z) = \frac{n^{-2}\sum_{i<i'}\omega_z(Z_i, Z_{i'})\operatorname{sign}(X_{ij} - X_{i'j})\operatorname{sign}(X_{ik} - X_{i'k})}{n^{-2}\sum_{i<i'}\omega_z(Z_i, Z_{i'})}, \text{ where} \tag{4}$$

$$\omega_z(Z_i, Z_{i'}) = K_h(Z_i - z)\, K_h(Z_{i'} - z) \tag{5}$$

with the kernel function $K(\cdot)$ being a symmetric density function, $K_h(\cdot) = h^{-1}K(\cdot/h)$ and $h > 0$ is the bandwidth parameter. We can choose the kernel function as long as it satisfies some regularity conditions, which will be specified in Assumption 4.3. The kernel U-statistic in (4) is a generalization of classical kernel regression (Opsomer and Ruppert, 1997; Fan and Jiang, 2005). For example, given i.i.d. samples $\{Y_i, Z_i\}_{i=1}^n$ from the model $Y = f(Z) + \epsilon$, the Nadaraya-Watson estimator (Bierens, 1988) is

$$\widehat{f}(z) = \frac{n^{-1}\sum_{i=1}^n K_h(Z_i - z)Y_i}{n^{-1}\sum_{i=1}^n K_h(Z_i - z)}, \tag{6}$$

where we take the weighted average of Y_i's and the weight $K_h(Z_i - z)$ is related to the distance between Z_i and z. In order to normalize the weights, we add the denominator in (6), which is the kernel density estimator for the density of Z. Comparing (4) with the Nadaraya-Watson estimator, since the kernel U-statistic involves both $\boldsymbol{X}_i$ and $\boldsymbol{X}_{i'}$, we need to multiply the weights as (5) to ensure that both Z_i and $Z_{i'}$ are in the neighborhood of z. We also normalize the weights by dividing the denominator $n^{-2}\sum_{i<i'}\omega_z(Z_i, Z_{i'})$ which is the estimator of $f_Z^2(z)$. The denominator $n^{-2}\sum_{i<i'}\omega_z(Z_i, Z_{i'})$ is also related to the density $f_Z(z)$. In fact, it is the estimator of $f_Z^2(z)$. Intuitively, we can see it from

$$\mathbb{E}\Big[\frac{1}{n^2}\sum_{i<i'}\omega_z(Z_i, Z_{i'})\Big] = \iint K(t_1)K(t_2)f_Z(z + t_1h)f_Z(z + t_2h)dt_1dt_2 = f_Z^2(z) + O(h^2).$$

Lemma 14 provides the details and is given in the supplementary material.

Based on the above estimator, we obtain the corresponding latent correlation matrix for any index value $z \in (0, 1)$ as

$$\widehat{\boldsymbol{\Sigma}}(z) = \sin\left(\frac{\pi}{2}\widehat{\mathbf{T}}(z)\right), \tag{7}$$

where $\widehat{\mathbf{T}}(z) = [\widehat{\tau}_{jk}(z)] \in \mathbb{R}^{d\times d}$.

Finally, similar to the static case, we can plug $\widehat{\boldsymbol{\Sigma}}(z)$ into a procedure that gives an estimate of the inverse correlation matrix, such as the CLIME in (2) or calibrated CLIME in (3). Due to practical advantages of the calibrated CLIME, we will use it for our simulations. However, at this point we note that the inferential framework only requires the estimator of the inverse correlation matrix to converge at a fast enough rate. Therefore, in what follows, we denote $\widehat{\boldsymbol{\Omega}}(z)$ a generic estimator of $\boldsymbol{\Omega}(z)$. Concrete statistical properties required of the calibrated CLIME will be discussed in details in Section 4.2.

3. Inferential Methods

In this section, we develop a framework for statistical inference about the parameters in a time-varying nonparanormal graphical model. We focus on the following three testing problems:

- **Edge presence test:** $H_0 : \boldsymbol{\Omega}_{jk}(z_0) = 0$ for a fixed $z_0 \in (0, 1)$ and $j, k \in [d]$;
- **Super-graph test:** $H_0 : G^*(z_0) \subset G$ for a fixed $z_0 \in (0, 1)$ and a fixed graph G;
- **Uniform edge presence test:** $H_0 : G^*(z) \subset G$ for all $z \in [z_L, z_U] \subset (0, 1)$ and a fixed graph G.

For all the testing problems, the alternative hypotheses is the negation of the null. The edge presence test is concerned with a local hypothesis that X_j and X_k are conditionally independent given $\boldsymbol{X}_{\setminus\{j,k\}}$ for a particular value of the index z_0. Equivalently, under the null hypothesis of the edge presence test, the nodes j and k are not connected at a particular index value z_0. The null hypothesis under the super-graph test postulates that the true graph is a subgraph of a given graph G for $Z = z_0$. It can also be interpreted as multiple-edge presence tests, since

$$H_0 : \boldsymbol{\Omega}_{jk}(z_0) = 0, \qquad \text{for all } (j, k) \in E^c, \tag{8}$$

where E is the edge set of the graph G. The null hypothesis under the uniform edge presence test postulates that the true graph is a subgraph of G for all index values in the range $[z_L, z_U]$. It is a generalization of the first two local tests to a global test over a range of index values. Similar to the super-graph test, this hypothesis is equivalent to the following

$$H_0 : \sup_{z\in[z_L, z_U]} |\boldsymbol{\Omega}_{jk}(z)| = 0, \qquad \text{for all } (j, k) \in E^c.$$

If the graph G consists of the edge set $E = \{(a, b) \in V \times V \mid (a, b) \neq (j, k)\}$, the uniform edge presence test becomes a uniform single-edge test $H_0 : \sup_{z\in[z_L, z_U]} |\boldsymbol{\Omega}_{jk}(z)| = 0$.

Next, we provide details on how to construct tests for the above three hypothesis.

3.1 Edge Presence Testing

We consider the hypothesis $H_0 : \boldsymbol{\Omega}_{jk}(z_0) = 0$, for a fixed $z_0 \in (0, 1)$ and $j, k \in [d]$. In order to construct a test for this hypothesis, we introduce the score function

$$\widehat{S}_{z|(j,k)}(\boldsymbol{\beta}) = \widehat{\boldsymbol{\Omega}}_j^T(z)\big(\widehat{\boldsymbol{\Sigma}}(z)\boldsymbol{\beta} - \mathbf{e}_k\big). \tag{9}$$

The argument $\boldsymbol{\beta}$ of the score function corresponds to the k-th column of $\boldsymbol{\Omega}(z)$. Our test is based on the score function evaluated at $\widehat{\boldsymbol{\Omega}}_{k\setminus j}$ which is an estimator of $\boldsymbol{\Omega}_k(z)$ under the null hypothesis, defined as

$$\widehat{\boldsymbol{\Omega}}_{k\setminus j}(z) = \big(\widehat{\boldsymbol{\Omega}}_{1k}(z), \ldots, \widehat{\boldsymbol{\Omega}}_{(j-1)k}(z), 0, \widehat{\boldsymbol{\Omega}}_{(j+1)k}(z), \ldots, \widehat{\boldsymbol{\Omega}}_{dk}(z)\big)^T \in \mathbb{R}^d,$$

where $\widehat{\boldsymbol{\Omega}}(z)$ is an estimator of $\boldsymbol{\Omega}(z)$. That is, we use $\widehat{S}_{z|(j,k)}\big(\widehat{\boldsymbol{\Omega}}_{k\setminus j}(z)\big)$ as the score statistic. We establish statistical properties of this statistic later. We first develop intuition for

why $\widehat{S}_{z|(j,k)}\big(\widehat{\mathbf{\Omega}}_{k\backslash j}(z)\big)$ is a good testing statistic for $H_0 : \mathbf{\Omega}_{jk}(z_0) = 0$. If we replace $\widehat{\mathbf{\Omega}}$ and $\widehat{\mathbf{\Sigma}}$ by the truth $\mathbf{\Omega}$ and $\mathbf{\Sigma}$ in (9), we can find the score statistic $\widehat{S}_{z|(j,k)}\big(\widehat{\mathbf{\Omega}}_{k\backslash j}(z)\big) \approx \mathbf{\Omega}_j^T(z)\big(\mathbf{\Sigma}(z)\mathbf{\Omega}_{k\backslash j}(z) - \mathbf{e}_k\big) = \mathbf{\Omega}_{jk}(z)$. Therefore, the score statistic is close to zero under the null, and $\widehat{S}_{z|(j,k)}\big(\widehat{\mathbf{\Omega}}_{k\backslash j}(z_0)\big) \approx \widehat{\mathbf{\Omega}}_{jk}(z_0)$ under the alternative. In specific, under H_0, the testing statistic is close to

$$\widehat{S}_{z|(j,k)}\big(\widehat{\mathbf{\Omega}}_{k\backslash j}\big) \approx \mathbf{\Omega}_j^T(z)\big(\widehat{\mathbf{\Sigma}}(z) - \mathbf{\Sigma}(z)\big)\mathbf{\Omega}_k(z) \approx \mathbf{\Omega}_j^T(z)\big[\dot{\mathbf{\Sigma}}(z) \circ \big(\widehat{\mathbf{T}}(z) - \mathbf{T}(z)\big)\big]\mathbf{\Omega}_k(z), \quad (10)$$

where $\dot{\mathbf{\Sigma}}_{jk}(z) = (\pi/2)\cos\big((\pi/2)\tau_{jk}(z)\big)$ is the derivative of $\mathbf{\Sigma}_{jk}(\cdot)$ and "$\approx$" denotes equality up to a smaller order term. The first approximation in (10) is due to replacing $\widehat{\mathbf{\Omega}}$ with the truth and the second approximation is due to the Taylor expansion. A rigorous derivation of this argument can be found in Appendix B.1. We note that the right-hand side of (10) is a linear function of $\widehat{\mathbf{T}}(z)$, which is a U-statistic. By applying the central limit theorem for U-statistics, we will show the asymptotic normality of $\widehat{S}_{z|(j,k)}\big(\widehat{\mathbf{\Omega}}_{k\backslash j}\big)$. See Theorem 1 for details.

Under the null, we have that

$$\sqrt{nh}\cdot\sigma_{jk}^{-1}(z_0)\widehat{S}_{z_0|(j,k)}\big(\widehat{\mathbf{\Omega}}_{k\backslash j}(z_0)\big) \rightsquigarrow N(0,1),$$

where $\sigma_{jk}^2(z_0) = f_Z^{-4}(z_0)\mathrm{Var}(\mathbf{\Omega}_j^S(z_0)\mathbf{\Theta}_{z_0}\mathbf{\Omega}_k(z_0))$, and $\mathbf{\Theta}_z$ is a random matrix with elements

$$(\mathbf{\Theta}_z)_{jk} = \pi\cos\big((\pi/2)\tau_{jk}(z)\big)\tau_{jk}^{(1)}(Y), \quad \text{where} \quad (11)$$

$$\tau_{jk}^{(1)}(\boldsymbol{x}, z) = \sqrt{h}\cdot\mathbb{E}\left[K_h\left(z - z_0\right)K_h\left(Z - z_0\right)\left(\mathrm{sign}(X_j - x_j)\,\mathrm{sign}(X_k - x_k) - \tau_{jk}(z_0)\right)\right], \quad (12)$$

where the expectation is taken for Z and $\boldsymbol{X}$. For simplicity, we denote $y = (\boldsymbol{x}^T, z)^T$ and write $\tau_{jk}^{(1)}(y) := \tau_{jk}^{(1)}(\boldsymbol{x}, z)$. The form of the asymptotic variance comes from the Hoeffding decomposition of the U-statistics in (4), with $\tau_{jk}^{(1)}$ being the leading term of the decomposition. Technical details will be provided in Section 4.1 and Section B.1. The score statistic is a generalization of Rao's score tests in fixed dimensional parametric models. We can apply a one-step debiased estimator of $\mathbf{\Omega}_{jk}(z)$ from the score statistic $\widehat{S}_{z|(j,k)}\big(\widehat{\mathbf{\Omega}}_{k\backslash j}(z)\big)$ following the procedure similar to (Ning and Liu, 2017). They show that the one-step estimation procedure is asymptotically equivalent to the score statistic and we choose to use score statistic in this paper. Janková and van de Geer (2017) considered a similar debiasing procedure specific for the nodewise regression estimator. On the other hand, we will show in Theorem 1 that the score statistic $\widehat{S}_{z|(j,k)}\big(\widehat{\mathbf{\Omega}}_{k\backslash j}(z)\big)$ can be applied to any estimator $\widehat{\mathbf{\Omega}}$ has sharp enough statistical rate.

In order to use the score function as a test statistic, we need to estimate its asymptotic variance $\sigma_{jk}^2(z_0)$. For any $1 \le s \le n$ and $1 \le j, k \le d$, let

$$q_{s,jk}(z) = \frac{\sqrt{h}}{n-1}\sum_{s'\neq s}\left[\omega_z(Z_s, Z_{s'})\left(\mathrm{sign}\left((X_{sj} - X_{s'j})(X_{sk} - X_{s'k})\right) - \widehat{\tau}_{jk}(z)\right)\right], \quad (13)$$

$$\widehat{\mathbf{\Theta}}_{jk}^{(s)}(z) = \pi\cos\left((\pi/2)\widehat{\tau}_{jk}(z)\right)q_{s,jk}(z).$$

With this notation, the leave-one-out Jackknife estimator for $\sigma_{jk}^2(z_0)$ is given as

$$\widehat{\sigma}_{jk}^2(z_0) = [\mathbb{U}_n(\omega_{z_0})]^{-2}\cdot\frac{1}{n}\sum_{s=1}^{n}\left(\widehat{\mathbf{\Omega}}_j^S(z_0)\widehat{\mathbf{\Theta}}^{(s)}(z_0)\widehat{\mathbf{\Omega}}_k(z_0)\right)^2, \quad (14)$$

where the matrix $\widehat{\mathbf{\Theta}}^{(s)}(z) = [\widehat{\mathbf{\Theta}}_{jk}^{(s)}(z)]$. As we have remarked in Section 2.1, $\widehat{\sigma}_{jk}^2(z_0)$ is an estimator for $f_Z^2(z)$. We divide $[\mathbb{U}_n(\omega_{z_0})]^{-2}$ in (14) in order to normalized the weights $\omega_z(Z_s, Z_{s'})$ in the U-statistics $q_{s,jk}(z)$ defined in (13). The Jackknife estimator is widely used when estimating the variance of a U-statistics, which is not an average of independent random variables. The leave-one-out statistic $q_{s,jk}(z)$ in (13) is estimates the expectation in (12) by leaving Y_s out of the summation in $q_{s,jk}(z)$.

Finally, a level α test for $H_0 : \mathbf{\Omega}_{jk}(z_0) = 0$ is given as

$$\psi_{z_0|(j,k)}(\alpha) = \begin{cases} 1 & \text{if } \sqrt{nh}\cdot\left|\widehat{S}_{z_0|(j,k)}\big(\widehat{\mathbf{\Omega}}_{k\backslash j}(z_0)\big)/\widehat{\sigma}_{jk}(z_0)\right| > \Phi^{-1}(1-\alpha/2); \\ 0 & \text{if } \sqrt{nh}\cdot\left|\widehat{S}_{z_0|(j,k)}\big(\widehat{\mathbf{\Omega}}_{k\backslash j}(z_0)\big)/\widehat{\sigma}_{jk}(z_0)\right| \le \Phi^{-1}(1-\alpha/2), \end{cases}$$

where $\Phi(\cdot)$ is the cumulative distribution function of a standard normal distribution. The single-edge presence test is a cornerstone of more general hypothesis tests described in the next two sections. The properties of the test are given in Theorem 1.

3.2 Super-Graph Testing

In this section, we discuss super-graph testing. Recall that for a fixed z_0 and a predetermined graph $G = (V, E)$, the null hypothesis is

$$H_0 : G^*(z_0) \subset G. \quad (15)$$

From (8), we have that the super-graph test can be seen as a multiple test for presence of several edges. Therefore, we propose the following testing statistic based on the score function in (9):

$$S(z_0) = \sqrt{nh}\cdot\mathbb{U}_n(\omega_{z_0})\max_{(j,k)\in E^c}\widehat{S}_{z_0|(j,k)}\big(\widehat{\mathbf{\Omega}}_{k\backslash j}(z_0)\big). \quad (16)$$

In order to estimate the quantile of $S(z_0)$, we develop a novel Gaussian multiplier bootstrap for U-statistics. Let $\{\xi_i\}_{i\in[n]}$ be n independent copies of $N(0,1)$. Let $\widehat{\mathbf{T}}^B(z) = \big[\widehat{\tau}_{jk}^B(z)\big]$ where

$$\widehat{\tau}_{jk}^B(z) = \frac{\sum_{i\neq i'}K_h\left(Z_i - z\right)K_h\left(Z_{i'} - z\right)\mathrm{sign}\left((X_{ij} - X_{i'j})(X_{ik} - X_{i'k})\right)(\xi_i + \xi_{i'})}{\sum_{i\neq i'}K_h\left(Z_i - z\right)K_h\left(Z_{i'} - z\right)(\xi_i + \xi_{i'})}, \quad (17)$$

$$\widehat{\mathbf{\Sigma}}^B(z) = \sin\left(\frac{\pi}{2}\widehat{\mathbf{T}}^B(z)\right). \quad (18)$$

The Gaussian multiplier bootstrap statistic in (19) is motivated by the method developed in Chernozhukov et al. (2013), who proposed a bootstrap procedure to estimate a quantile of the supremum of high dimensional empirical processes. Their method, however, cannot be directly applied to our kernel Kendall's tau estimator in (4), which is a ratio of two

U-statistics. If we compare (17) with (4), we add $\xi_i + \xi_{i'}$ into the bootstrap estimator in order to simulate the distribution of $\widehat{\tau}_{jk}(z)$ in (4).

The bootstrap estimator of the test statistic $S(z_0)$ in (16) is

$$S^B(z_0) = \sqrt{nh} \cdot \mathbb{U}_n[\omega^B_{z_0}] \max_{(j,k)\in E^c} \widehat{\mathbf{\Omega}}^S_j(z_0)\big(\widehat{\mathbf{\Sigma}}^B(z_0)\widehat{\mathbf{\Omega}}_{k\backslash j} - \mathbf{e}_k^T\big), \text{ where} \tag{19}$$

$$\mathbb{U}_n[\omega^B_{z_0}] = \frac{2}{n(n-1)} \sum_{i\neq i'} K_h\left(Z_i - z_0\right) K_h\left(Z_{i'} - z_0\right)(\xi_i + \xi_{i'}). \tag{20}$$

Here we multiply $\mathbb{U}_n[\omega^B_{z_0}]$ in (19) in order to eliminate the denominator in (17) such that the leading term of the bootstrap statistic $S^B(z_0)$ is a Gaussian multiplier bootstrap U-statistic. The correlation estimator in (7) has an additional sin transform. Therefore, a new nonlinear type of multiplier bootstraps in (17) and (19) are introduced to overcome these problems and novel technical tools are then developed to study statistical properties. See Theorem 3 for the statement of statistical properties.

Denote the conditional $(1-\alpha)$-quantile of $S^B(z_0)$ given $\{Y_i\}_{i=1}^n$ as $\widehat{c}_T(1-\alpha, \{Y_i\}_{i=1}^n)$. The level-$\alpha$ super-graph test is constructed as

$$\psi_{z_0|G}(\alpha) = \begin{cases} 1 & \text{if } S(z_0) > \widehat{c}_T(1-\alpha, \{Y_i\}_{i=1}^n); \\ 0 & \text{if } S(z_0) \leq \widehat{c}_T(1-\alpha, \{Y_i\}_{i=1}^n). \end{cases} \tag{21}$$

Note that the quantile $\widehat{c}_T(1-\alpha, \{Y_i\}_{i=1}^n)$ can be estimated by a Monte-Carlo method.

An alternative approach for the super-graph test in (15) is the multiple hypothesis testing. We can apply the Holm's multiple testing procedure (Holm, 1979) to control the family-wise error for the hypotheses set $\{H_{0,(jk)}\}_{(j,k)\in E^c}$ where $H_{0,(jk)} : \mathbf{\Omega}_{jk}(z_0) = 0$ and $G^*(z_0) \subset G = (V, E)$. However, it is not straightforward to obtain the nominal probability for the family-wise error in Holm's method. Moreover, we will show in Theorem 3 that our testing procedure is nominal.

3.3 Uniform Edge Presence Testing

In this section, we develop the uniform presence test for which the null hypothesis is given as

$$H_0 : G^*(z) \subset G \text{ for all } z \in [z_L, z_U].$$

This test is a generalization of the edge presence test to the uniform version over both edges and index. We again use the score function in (9) to construct the test statistic

$$W_G = \sqrt{nh} \sup_{z\in[z_L,z_U]} \max_{(j,k)\in E^c} \mathbb{U}_n[\omega_z]\widehat{S}_{z|(j,k)}\big(\widehat{\mathbf{\Omega}}_{k\backslash j}(z)\big) \tag{22}$$

and estimate a quantile of W_G by developing a Gaussian multiplier bootstrap. Let

$$W^B_G = \sqrt{nh} \sup_{z\in[z_L,z_U]} \max_{(j,k)\in E^c} \mathbb{U}_n[\omega^B_z] \cdot \widehat{\mathbf{\Omega}}^T_j(z)\big(\widehat{\mathbf{\Sigma}}^B(z)\widehat{\mathbf{\Omega}}_{k\backslash j}(z) - \mathbf{e}_k^T\big), \tag{23}$$

where $\widehat{\mathbf{\Sigma}}^B(z)$ is defined in (18). Let $\widehat{c}_W(1-\alpha, \{Y_i\}_{i=1}^n)$ denote the conditional $(1-\alpha)$-quantile of W^B_G given $\{Y_i\}_{i=1}^n$. Similar to (21), the level α uniform edge presence test is constructed as

$$\psi_G(\alpha) = \begin{cases} 1 & \text{if } W_G > \widehat{c}_W(1-\alpha, \{Y_i\}_{i=1}^n); \\ 0 & \text{if } W_G \leq \widehat{c}_W(1-\alpha, \{Y_i\}_{i=1}^n). \end{cases} \tag{24}$$

Theorem 4 provides statistical properties of the test. The statistics W_G in (22) and W^B_G in (23) involve taking supreme over $z \in [z_L, z_U]$. In practice, we approximate the suprema by evenly dividing $[z_L, z_U]$ into discrete grids and taking the maximum of the statistic over these discrete values in $[z_L, z_U]$.

4. Theoretical Properties

In this section, we establish the validity of tests proposed in the previous sections. Validity of tests rely on existence of estimators for the latent inverse correlation matrix with fast enough convergence. We show that the calibrated CLIME satisfy the testing requirements and, in addition, show that it achieves the minimax rate of convergence for a large class of models.

To facilitate the argument, we need the regularity and smoothness of the density function of Z and the time-varying correlation matrix $\mathbf{\Sigma}(z)$. Let us first introduce the Hölder class $\mathcal{H}(\gamma, L)$ of smooth functions. The Hölder class $\mathcal{H}(\gamma, L)$ on $(0,1)$ is the set of $\ell = \lfloor\gamma\rfloor$ times differentiable functions $g : \mathcal{X} \mapsto \mathbb{R}$ whose derivative $g^{(\ell)}$ satisfies

$$|g^{(\ell)}(x) - g^{(\ell)}(y)| \leq L|x-y|^{\gamma-\ell}, \quad \text{for any } x, y \in \mathcal{X}$$

and $\lfloor\gamma\rfloor$ denotes the largest integer smaller than γ. In this paper, we need some regularity conditions for the functions in our model.

Assumption 4.1 (Density function of Z) *There exist constants $0 < \underline{f}_Z < \bar{f}_Z < \infty$ such that the marginal density f_Z of the index variable Z has its image in $[\underline{f}_Z, \bar{f}_Z]$ and $f_Z \in \mathcal{H}(2, \bar{f}_Z)$.*

Assumption 4.2 (Regularization of $\mathbf{\Sigma}_{jk}(\cdot)$) *The correlations $\mathbf{\Sigma}_{jk}(\cdot) \in \mathcal{H}(2, M_\sigma)$ for some constant $M_\sigma < \infty$ given any $1 \leq j, k \leq d$.*

The above two assumptions are standard assumptions on the marginal distribution of Z (Pagan and Ullah, 1999) and time-varying graphical models (see, for example, Kolar et al., 2010a).

Assumption 4.3 (Kernel function) *Through this paper, we assume the kernel function K, used in (4), is a symmetric density function supported on $[-1, 1]$ with bounded variation, i.e., $||K||_\infty \vee \mathrm{TV}(K) < \infty$,*

$$\int_{-1}^{1} K(u)du = 1 \text{ and } \int_{-1}^{1} uK(u)du = 0.$$

These properties are also required in Zhou et al. (2010). Many widely used kernels, including the uniform kernel $K(u) = 0.5\mathbb{1}(|u| < 1)$, the triangular kernel $K(u) = (1-|u|)\mathbb{1}(|u| < 1)$, and the Epanechnikov kernel $K(u) = 0.75(1-u^2)\mathbb{1}(|u| < 1)$, satisfy this assumption.

Finally, we list a generic assumption on the properties of $\widehat{\mathbf{\Sigma}}(z)$ in (7) and the an inverse correlation matrix estimator $\widehat{\mathbf{\Omega}}(z)$.

Assumption 4.4 (Statistical rates) *There are sequences $r_{1n}, r_{2n}, r_{3n} = o(1)$ such that*

$$\sup_{z\in(0,1)} \|\widehat{\mathbf{\Sigma}}(z) - \mathbf{\Sigma}(z)\|_{\max} \le r_{1n}, \quad \sup_{z\in(0,1)} \|\widehat{\mathbf{\Omega}}(z) - \mathbf{\Omega}(z)\|_1 \le r_{2n}, \text{ and}$$
$$\sup_{z\in(0,1)} \max_{j\in[d]} \|\widehat{\mathbf{\Sigma}}(z)\widehat{\mathbf{\Omega}}_j(z) - \mathbf{e}_j\|_\infty \le r_{3n},$$

with probability at least $1 - 1/d$.

Assumption 4.4 is a generic condition on the consistency of $\widehat{\mathbf{\Sigma}}(z)$ and $\widehat{\mathbf{\Omega}}(z)$. We aim to show that our testing methods are independent to any specific procedure to estimate $\mathbf{\Omega}(\cdot)$. The three rates in Assumption 4.4 are sufficient for the validity of our tests. Our inferential framework can thus be easily generalized to other inverse correlation matrix estimators as long as their rates satisfy Assumption 4.4. Under this assumption, the score statistic used for testing can be approximated by an asymptotically normal leading term. For our estimators $\widehat{\mathbf{\Sigma}}(\cdot)$ in (7) and $\widehat{\mathbf{\Omega}}(\cdot)$ in (2) or (3), we will show in Theorems 5 and 6 that

$$r_{1n} = O(\sqrt{\log(d/h)/(nh)}), \quad r_{2n} = O(s\sqrt{\log(d/h)/(nh)}) \text{ and } r_{3n} = O(\sqrt{\log(d/h)/(nh)}). \tag{25}$$

See Section 4.2 for more details.

4.1 Validity of Tests

In this section, we state theorems on asymptotic validity of the tests considered in Section 3. We first define the parameter space

$$\mathcal{U}_s(M,\rho) = \Big\{\mathbf{\Omega} \in \mathbb{R}^{d\times d} \Big| \, \mathbf{\Omega} \succ 1/\rho, \|\mathbf{\Omega}\|_2 \le \rho, \max_{j\in[d]} \|\mathbf{\Omega}_j\|_0 \le s, \|\mathbf{\Omega}\|_1 \le M\Big\}. \tag{26}$$

This matrix class was considered in the literature on inverse covariance matrix estimation (Cai et al., 2016) and time-varying covariance estimation (Chen and Leng, 2016).

The following theorem gives us the limiting distribution of the score function defined in (9).

Theorem 1 (Edge presence test) *For a fixed $z_0 \in (0,1)$, suppose $\mathbf{\Omega}(z_0) \in \mathcal{U}_s(M,\rho)$, Assumption 4.4 holds with $\sqrt{nh}\cdot(r_{2n}(r_{1n}+r_{3n})) = o(1)$ and the bandwidth h satisfies*

$$\sqrt{nh}\big(\log(dn)/(nh) + h^2\big) + s^3/\sqrt{nh} = o(1). \tag{27}$$

Furthermore, for a fixed and $j,k \in [d]$, suppose there exists $\theta_{\min} > 0$ such that

$$\mathbb{E}(\mathbf{\Omega}_j^T(z_0)\mathbf{\Theta}_{z_0}\mathbf{\Omega}_k(z_0))^2 \ge \theta_{\min}\|\mathbf{\Omega}_j(z_0)\|_2^2\|\mathbf{\Omega}_k(z_0)\|_2^2,$$

then under $H_0 : \mathbf{\Omega}_{jk}(z_0) = 0$, we have that

$$\sqrt{nh}\cdot\sigma_{jk}^{-1}(z_0)\widehat{S}_{z_0|(j,k)}\big(\widehat{\mathbf{\Omega}}_{k\backslash j}(z_0)\big) \rightsquigarrow N(0,1),$$

where $\sigma_{jk}^2(z_0) = f_Z^{-4}(z_0)\mathrm{Var}(\mathbf{\Omega}_j^T(z_0)\mathbf{\Theta}_{z_0}\mathbf{\Omega}_k(z_0))$ and $\mathbf{\Theta}_{z_0}$ is defined in (11).

We have two sets of scaling conditions in the above theorem. Under the condition $\sqrt{nh}(r_{2n}(r_{1n}+r_{3n})) = o(1)$, the first heuristic approximation "$\approx$" in (10) is valid. The condition in (27) guarantees that the leading term on the right hand side of (10) is asymptotically normal. In particular, the first term of (27) makes the second heuristic approximation in (10) valid and allows for control of the higher order term of the Hoeffding decomposition of the U-statistics in (4). If r_{1n}, r_{2n} and r_{3n} have the rates as in (25) (see Theorems 5 and 6 for more details), this condition becomes $\sqrt{nh}\cdot s(h^2 + \sqrt{\log(dn)/(nh)})^2 = o(1)$. We choose $h \asymp n^{-\nu}$, where $\nu > 1/5$ in order to remove the bias. The two scaling conditions in Theorem 1 can be replaced by $(s^3 + s\log(dn))/n^{(1-\nu)/2} = o(1)$. This is similar to the condition $s^2\log d/\sqrt{n}$ in the inference for the Lasso estimator (Zhang and Zhang, 2013; van de Geer et al., 2014; Javanmard and Montanari, 2014). Here, the slower $n^{-(1-\nu)/2}$ term originates from the nonparametric relationship between the index and correlation matrix. The additional s^2 term comes from the matrix structure and is ignorable if $s = o\Big(\sqrt{\log(dn)}\Big)$.

The following lemma shows that the asymptotic variance of the score function can be consistently estimated.

Lemma 2 *Suppose the conditions of Theorem 1 hold. If $r_{2n}/h = o(1)$ and $\log(dn)/(nh^3) = o(1)$, then the variance estimator $\widehat{\sigma}_{jk}^2(z_0)$ in (14) has $\widehat{\sigma}_{jk}^2(z_0) \xrightarrow{P} \sigma_{jk}^2(z_0)$.*

The proofs of Theorem 1 and Lemma 2 are deferred to Appendix B.1 and G.4 respectively. Stronger scaling conditions are needed for consistent estimation of variance as its estimator in (14) relies on controlling higher moments. Under the rates in (25) with $h \asymp n^{-\nu}$, for some $\nu > 1/5$, the scaling $(s^3 + s^2\log(dn))/n^{(1-\nu)/2} = o(1)$ suffices for the estimator to consistently estimate the variance.

We also have the following theorems on the asymptotic validity of the super-graph test.

Theorem 3 (Super-graph test) *Let $z_0 \in (0,1)$ and $j,k \in [d]$ be fixed. Assume the conditions of Theorem 1 hold. Suppose there exists $\theta_{\min} > 0$ such that for all $j \ne k \in [d]$, $\mathbb{E}(\mathbf{\Omega}_j^T(z_0)\mathbf{\Theta}_{z_0}\mathbf{\Omega}_k(z_0))^2 \ge \theta_{\min}\|\mathbf{\Omega}_j(z_0)\|_2^2\|\mathbf{\Omega}_k(z_0)\|_2^2$, and there exists a constant $\epsilon > 0$ such that $\sqrt{nh}(r_{2n}(r_{1n}+r_{3n})) = O(n^{-\epsilon})$ and*

$$\sqrt{nh}\big(\log(dn)/(nh) + h^2\big) + \log d/(nh^2) + (\log(dn))^7/(nh) = O(n^{-\epsilon}). \tag{28}$$

Let $G = (V,E)$ be any fixed graph and $G^*(z_0)$ is the Markov graph corresponding to the index value z_0. Under the null hypothesis $H_0 : G^*(z_0) \subset G$, the test $\psi_{z_0|G}(\alpha)$ defined in (21) satisfies

$$\sup_{\alpha\in(0,1)} \Big|\mathbb{P}_{H_0}\big(\psi_{z_0|G}(\alpha) = 1\big) - \alpha\Big| = O(n^{-c}) \tag{29}$$

for some universal constant c.

The next theorem shows the asymptotic validity of the uniform edge presence test.

Theorem 4 (Uniform edge presence test) *Assume that $\mathbf{\Omega}(z) \in \mathcal{U}(M,\rho)$ for any $z \in (0,1)$ and Assumption 4.4 is true. Suppose there exists $\theta_{\min} > 0$ such that for all $j \ne k \in [d]$ and $z \in (0,1)$, $\mathbb{E}(\mathbf{\Omega}_j^T(z)\mathbf{\Theta}_z\mathbf{\Omega}_k(z))^2 \ge \theta_{\min}\|\mathbf{\Omega}_j(z)\|_2^2\|\mathbf{\Omega}_k(z)\|_2^2$, and there exists a constant $\epsilon > 0$ such that $\sqrt{nh}(r_{2n}(r_{1n}+r_{3n})) = O(n^{-\epsilon})$ and*

$$\sqrt{nh}\big(\log(dn)/(nh) + h^2\big) + \log d/(nh^2) + (\log(dn))^8/(nh) = O(n^{-\epsilon}). \tag{30}$$

Under the null hypothesis $H_0 : G^*(z) \subset G$ *for all* $z \in [z_L, z_U]$*, the test* $\psi_G(\alpha)$ *defined in* (24) *satisfies*

$$\sup_{\alpha\in(0,1)} \left|\mathbb{P}_{H_0}\big(\psi_G(\alpha) = 1\big) - \alpha\right| = O(n^{-c}) \tag{31}$$

for some universal constant $c > 0$.

We defer the proof of Theorem 3 to Appendix B.2 and the proof of Theorem 4 to Appendix F.

Theorems 3 and 4 only depend on the estimation rates of $\widehat{\mathbf{\Sigma}}(z)$ and $\widehat{\mathbf{\Omega}}(z)$ through Assumption 4.4. This implies that our inferential framework does not rely on exact model selection. We have $O(n^{-\epsilon})$ in (28) and (30) instead of $o(1)$ in (27) to achieve the polynomial convergence rate for type I error in (29) and (31). Comparing the scaling condition in (28) with the one in (27), the second term in (28) is dominated by the first term under a mild bandwidth rate $h = o(n^{-1/3})$. The third term $(\log(dn))^7/(nh)$ in (28) comes from a Berry-Essen bound on the suprema of increasing dimensional U-processes. Such a scaling condition is similar to the one in Chernozhukov et al. (2013). They showed that for the empirical process $\boldsymbol{W} = (W_1, \ldots, W_d)^T$ having the same covariance as the centered Gaussian vector $\boldsymbol{U} = (U_1, \ldots, U_d)^T$, the following Berry-Essen bound holds

$$\sup_{t\in\mathbb{R}} \left|\mathbb{P}\big(\max_j W_j \le t\big) - \mathbb{P}\big(\max_j U_j \le t\big)\right| = O\left(\big((\log(dn))^7/n\big)^{1/6}\right).$$

Comparing with our condition that $(\log(dn))^7/(nh) = O(n^{-\epsilon})$, the additional term nh in the denominator comes from the nonparametric part of our estimator. Furthermore, Theorem 4 requires a stronger scaling condition in (30), where the term $(\log(dn))^8/(nh) = O(n^{-\epsilon})$ arises from the additional supremum over $z \in [z_L, z_U]$ in the uniform edge presence test.

4.2 Consistency of Estimation

In this section, we show that the Assumption 4.4 holds under mild conditions on the data generating process. We give explicit rates for r_{1n}, r_{2n} and r_{3n} under concrete estimation procedures. We first show the estimation rate of $\widehat{\mathbf{\Sigma}}$ given in (7). Next, we give the rate of convergence for $\mathbf{\Omega}(z)$ when using the (calibrated) CLIME estimator.

We establish rates of convergence that are uniform in bandwidth h. Uniform in bandwidth results are important as they ensure consistency of our estimators even when the bandwidth is chosen in a data-driven way, which is the case in practice, including the cross-validation over integrated squared error Hall (1992) and other risks (Muller and Stadtmuller, 1987; Ruppert et al., 1995; Fan and Gijbels, 1995). See Jones et al. (1996) for a survey of other methods. Existing literature on uniform in bandwidth consistency focuses on low dimensional problems (see, for example, Einmahl and Mason, 2005). High dimensional statistical methods usually have more tuning parameters and it is hard to guarantee that the selected bandwidth satisfies an optimal scaling condition. To the best of our knowledge this is the first result established in a high-dimensional regime that shows uniform consistency for a wide range of possible bandwidths.

The following theorem shows the rate of the covariance matrix estimator.

Theorem 5 *Assume* $\log d/n = o(1)$ *and the bandwidths* $0 < h_l < h_u < 1$ *satisfy*

$$h_l n/\log(dn) \to \infty \quad \text{and} \quad h_u = o(1).$$

There exists a universal constant $C_{\mathbf{\Sigma}} > 0$ *such that for any* $\delta \in (0,1)$*, we have*

$$\sup_{h\in[h_l,h_u]} \sup_{z\in(0,1)} \frac{\big\|\widehat{\mathbf{\Sigma}}(z) - \mathbf{\Sigma}(z)\big\|_{\max}}{h^2 + \sqrt{(nh)^{-1}\big[\log(d/h) \vee \log\big(\delta^{-1}\log\big(h_u h_l^{-1}\big)\big)\big]}} \le C_{\mathbf{\Sigma}} \tag{32}$$

with probability $1 - \delta$.

The proof of this theorem is deferred to Appendix A.1. Using (32), we can determine the rate of r_{1n} in Assumption 4.4. The supremum over the bandwidth h in (32) implies that if a data-driven bandwidth $\widehat{h}_n$ satisfies $\mathbb{P}(h_l \le \widehat{h}_n \le h_u) \longrightarrow 1$, then with high probability,

$$\sup_{z\in(0,1)} \big\|\widehat{\mathbf{\Sigma}}(z) - \mathbf{\Sigma}(z)\big\|_{\max} \le C_{\mathbf{\Sigma}} \left(\widehat{h}_n^2 + \sqrt{\frac{\log(d/\widehat{h}_n)}{n\widehat{h}_n}}\right). \tag{33}$$

The first term in the rate is the bias and the second is the variance. Our result is sharper than the rate $O\left(h^2\sqrt{\log d} + \sqrt{\log d/(nh)}\right)$ established in Lemma 9 of Chen and Leng (2016). The uniform consistency result $\sup_{h_l\le h\le h_u}\sup_{z\in(0,1)}\big\|\widehat{\mathbf{\Sigma}}(z) - \mathbf{\Sigma}(z)\big\|_{\max} = o_P(1)$ holds for a wide range of bandwidths satisfying $h_l n/\log(dn) \to \infty$ and $h_u = o(1)$, which allows flexibility for data-driven methods. In fact, such h_l is the smallest to make the variance (33) converge and h_u is the largest for the convergence of bias. When $d = 1$, the range $[h_l, h_u]$ is the same as for the kernel-type function estimators (Einmahl and Mason, 2005).

Due to the large capacity of the estimator $\widehat{\mathbf{\Sigma}}(z)$ in (32), which varies with both the bandwidth h and the index z, the routine proof based on uniform entropy numbers does not easily apply here. We use the peeling method (Van de Geer, 2000) by slicing the range of h into smaller intervals, for which the uniform entropy number is controllable. Finally, we assemble the interval specific bounds to obtain (32). See Section E.1 for more details.

Next, we give a result on the estimation consistency of the inverse correlation matrix. Let $\widehat{\mathbf{\Omega}}(z) = (\widehat{\mathbf{\Omega}}_1(z), \ldots, \widehat{\mathbf{\Omega}}_d(z))$ where each column $\widehat{\mathbf{\Omega}}_j(z)$ is constructed either by using the CLIME in (2) or calibrated CLIME in (3) We recommend using the calibrated CLIME in practice due to the tuning issues discussed in Section 2.1.

Theorem 6 *Suppose* $\mathbf{\Omega}(z) \in \mathcal{U}_s(M, \rho)$ *for all* $z \in (0,1)$*. Assume* $\log d/n = o(1)$*, the bandwidths* $0 < h_l < h_u < 1$ *satisfy* $h_l n/\log(dn) \to \infty$ *and* $h_u = o(1)$*. The regularization parameter* λ *is chosen to satisfy* $\lambda \ge \lambda_{n,h} := C_{\mathbf{\Sigma}}\big(h^2 + \sqrt{\log(d/h)/(nh)}\big)$*, where* $C_{\mathbf{\Sigma}}$ *is the constant in* (32)*, for the calibrated CLIME and* $\lambda \ge M\lambda_{n,h}$ *for the CLIME estimator. Then there exists a universal constant* $C > 0$ *such that*

$$\sup_{h\in[h_l,h_u]} \sup_{z\in(0,1)} \frac{1}{\lambda M^2}\big\|\widehat{\mathbf{\Omega}}(z) - \mathbf{\Omega}(z)\big\|_{\max} \le C; \tag{34}$$

$$\sup_{h\in[h_l,h_u]} \sup_{z\in(0,1)} \frac{1}{\lambda s M}\big\|\widehat{\mathbf{\Omega}}(z) - \mathbf{\Omega}(z)\big\|_1 \le C;$$

$$\sup_{z\in(0,1)} \max_{j\in[d]} \frac{1}{\lambda M} \cdot \|\widehat{\mathbf{\Omega}}_j^T\widehat{\mathbf{\Sigma}} - \mathbf{e}_j\|_\infty \le C, \tag{35}$$

with probability $1 - 1/d$.

The proof is deferred to Appendix A.2. From this theorem, we can see that (34) determines r_{2n} and (35) determines r_{3n} in Assumption 4.4. We can plug the r_{1n}, r_{2n} and r_{3n} given in (32), (34) and (35) into the condition $\sqrt{nh}(r_{2n}(r_{1n} + r_{3n})) = O(n^{-\epsilon})$ stated in Theorems 1, 3 and 4 to get an explicit condition for n, h, s, d.

Corollary 7 *Let* $\widehat{\mathbf{\Omega}}(z)$ *be the calibrated CLIME estimator with* $\lambda \geq \lambda_{n,h} = C_{\mathbf{\Sigma}}\big(h^2 + \sqrt{\log(d/h)/(nh)}\big)$ *or the CLIME estimator with* $\lambda \geq M\lambda_{n,h}$. *Then the Assumption 4.4 and all conditions on* r_{1n}, r_{2n} *and* r_{3n} *in Theorems 1, 3 and 4 can be replaced with* $\sqrt{nh} \cdot s\left(h^2 + \sqrt{\log(dn)/(nh)}\right)^2 = o(1)$.

Theorem 6 implies that if the bandwidth satisfies $h \asymp (\log(dn)/n)^{1/5}$, then

$$\sup_{z\in(0,1)} \big\|\widehat{\mathbf{\Omega}}(z) - \mathbf{\Omega}(z)\big\|_{\max} \leq CM^2\left(\frac{\log d + \log n}{n}\right)^{2/5};$$
$$\sup_{z\in(0,1)} \big\|\widehat{\mathbf{\Omega}}(z) - \mathbf{\Omega}(z)\big\|_{1} \leq CMs\left(\frac{\log d + \log n}{n}\right)^{2/5}$$

with probability $1 - 1/d$. When $\log d \gg \log n$, the optimal bandwidth for selection is larger than the standard scaling $h \asymp (\log n/n)^{1/5}$ for univariate nonparametric regression (Tsybakov, 2009). This is because we need to over-regularize each entry of $\widehat{\mathbf{\Sigma}}(z)$ to reduce the variance of entire matrix. The optimal bandwidth is also larger than the scaling for inference $h \asymp n^{-\nu}$ for some $\nu > 1/5$ in (27), since we also need to over-regularize to remove bias for inference.

4.2.1 Optimality of Estimation Rate

The following theorem shows that the rate in (34) is minimax optimal.

Theorem 8 *Consider the following class of the inverse correlation matrices*

$$\overline{\mathcal{U}}_s(M, \rho, L) := \{\mathbf{\Omega}(\cdot) \mid \mathbf{\Omega}(z) \in \mathcal{U}_s(M, \rho) \text{ for any } z \in (0,1), \text{ and } \mathbf{\Omega}_{jk}(\cdot) \in \mathcal{H}(2, L) \text{ for } j, k \in [d]\},$$

where $\mathcal{U}_s(M, \rho)$ *is defined in* (26). *We have the following two results on the minimax risk:*

1. *If* $\log(dn)/n = o(1)$, *then*

$$\inf_{\widehat{\mathbf{\Omega}}(z)} \sup_{\mathbf{\Omega}(\cdot)\in\overline{\mathcal{U}}_s(M,\rho,L)} \mathbb{E}\left[\sup_{z\in(0,1)} \|\widehat{\mathbf{\Omega}}(z) - \mathbf{\Omega}(z)\|_{\max}\right] \geq c\left(\frac{\log d + \log n}{n}\right)^{2/5}. \tag{36}$$

2. *If* $s^2 \log(dn)/n = o(1)$ *and* $s^{-v}d \leq 1$ *for some* $v > 2$, *then*

$$\inf_{\widehat{\mathbf{\Omega}}(z)} \sup_{\mathbf{\Omega}(\cdot)\in\overline{\mathcal{U}}_s(M,\rho,L)} \mathbb{E}\left[\sup_{z\in(0,1)} \|\widehat{\mathbf{\Omega}}(z) - \mathbf{\Omega}(z)\|_{1}\right] \geq cs\left(\frac{\log d + \log n}{n}\right)^{2/5}. \tag{37}$$

The proof is deferred to Appendix C. We prove it by applying Le Cam's lemma (Le Cam, 1973) and constructing a finite collection $\mathbf{\Omega}(\cdot)$ from the function value matrices space $\overline{\mathcal{U}}_s(M, \rho, L)$. If we take the dimension $d = 1$, our problem degenerates to the univariate twice differentiable function estimation. The risk on the left hand side of (36) becomes to the supreme norm between the estimated function and truth. The right hand side of (36) degenerates to $O((\log n/n)^{2/5})$ which matches the typical minimax rate for nonparametric regression in $\|\cdot\|_\infty$ risk (Tsybakov, 2009). This indicates the reason why we have the power $2/5$ in the rates.

5. Numerical Experiments

In this section, we explore finite sample performance of our estimation procedure and test using simulations. Furthermore, we also apply the super graph test to the brain image network.

5.1 Synthetic Data

We illustrate finite sample properties of the estimator in (3) on synthetic data. We consider three other competing methods: (1) the kernel Pearson CLIME estimator (Zhou et al., 2010; Yin et al., 2010), (2) the kernel graphical Lasso estimator (Zhou et al., 2010), and (3) the kernel neighborhood selection estimator (Kolar et al., 2010a). Zhou et al. (2010) and Yin et al. (2010) consider the correlation matrix estimator at z as a weighted summation of Pearson's correlations

$$\widehat{\mathbf{\Sigma}}_P(z) = \frac{\sum_{i=1}^n K_h(Z_i - z)\boldsymbol{X}_i\boldsymbol{X}_i^T}{\sum_{i=1}^n K_h(Z_i - z)}.$$

This estimate can be plugged into the calibrated CLIME estimator in (3) to obtain the kernel Pearson inverse correlation estimator $\widehat{\mathbf{\Omega}}_{(1)}(z)$. Zhou et al. (2010) proposed to estimate the inverse correlation matrix by plugging $\widehat{\mathbf{\Sigma}}_P$ into the graphical lasso (Yuan and Lin, 2007) objective

$$\widehat{\mathbf{\Omega}}_{(2)}(z) = \operatorname*{argmax}_{\mathbf{\Omega}\succ 0} \Big\{\log\det\mathbf{\Omega} - \operatorname{tr}\left(\mathbf{\Omega}\widehat{\mathbf{\Sigma}}_P(z)\right) - \lambda\|\mathbf{\Omega}\|_1\Big\},$$

resulting in the kernel graphical Lasso estimator. Kolar et al. (2010a) proposed to combine the neighborhood selection procedure Meinshausen and Bühlmann (2006) with local kernel smoothing. To estimate the j-th column of $\mathbf{\Omega}(z)$ they use

$$\widehat{\boldsymbol{\beta}}_j(z) = \operatorname*{argmin}_{\boldsymbol{\beta}\in\mathbb{R}^{d-1}} \frac{1}{nh}\sum_{i=1}^n K\left(\frac{Z_i - z}{h}\right)(\boldsymbol{X}_{ij} - \boldsymbol{X}_{i\setminus j}\boldsymbol{\beta})^2 + \lambda\|\boldsymbol{\beta}\|_1,$$

where $\boldsymbol{X}_{i\setminus j}$ is a $(d-1)$-dimensional vector with the j-th entry of $\boldsymbol{X}_i$ removed. The kernel neighborhood selection estimator $\widehat{\mathbf{\Omega}}_{(3)}(z)$ is then obtained as as

$$\big[\widehat{\mathbf{\Omega}}_{(3)}(z)\big]_{jj} = 1 \text{ and } \big[\widehat{\mathbf{\Omega}}_{(3)}(z)\big]_{\setminus j,j} = \widehat{\boldsymbol{\beta}}_j(z), \text{ for all } j \in [d].$$

We describe a procedure to generate the graph $G^*(z)$ and inverse correlation matrix $\mathbf{\Omega}(z)$ at each $z \in (0,1)$. At any index $z \in (0,1)$, we generate a graph with $d = 50$

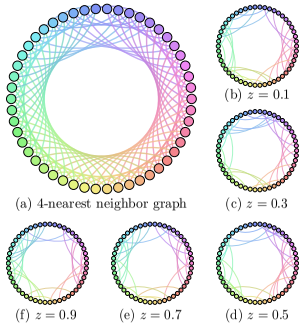


Figure 1: The 4-nearest neighbor graph and the time-varying graph for various index values z. The thickness of edges represents the magnitude of the inverse correlation matrix corresponding to that edge for a specific value z.

nodes and $e = 25$ by selecting edge from the 4-nearest neighbor graph illustrated in Figure 1(a). We first generate $Z_1, \ldots, Z_n$ independently from Uniform$(0, 1)$ with the sample size $n \in \{200, 500, 800\}$ and then generate $\mathbf{\Omega}(z)$ at each index value Z_i as follows.

1. Randomly select 25 edges from the 4-nearest neighbor graph and generate the initial graph $G^{(0)}$. The structure of the underlying graph will change at the following anchor points $z = 0.2, 0.4, 0.6, 0.8$. At the ℓ-th anchor point, we randomly remove 10 edges from the previous graph $G^{(\ell-1)}$ and randomly add 10 edges from the 4-nearest neighbor graph that did not belong to the previous graph $G^{(k-1)}$. We therefore generate 5 anchor graphs $G^{(0)}, \ldots, G^{(4)}$. On the ℓ-th interval $[(\ell-1)/5, \ell/5]$, the graph structure stays constant and equal to $G^{(\ell-1)}$ for $\ell = 1, \ldots, 5$.

2. Given the graph structure we generate $\mathbf{\Omega}(z)$. At the middle of each interval, that is, for index values $z = 0.1, 0.3, 0.5, 0.7, 0.9$, if the edge (j, k) belongs to the graph at that time, we randomly generate $\mathbf{\Omega}_{jk}(z)$ from Uniform$[\mu, 0.9]$, where μ is the minimal signal strength, otherwise we set $\mathbf{\Omega}_{jk}(z)$ to be zero. For $1 \leq \ell \leq 4$, if the edge (j, k) belongs to both $G^{(\ell)}$ and $G^{(\ell+1)}$, we let $\mathbf{\Omega}_{jk}(z_\ell)$ be generated from Uniform$[\mu, 0.9]$, otherwise we set it to be zero. We also generate $\mathbf{\Omega}_{jk}(0)$ from Uniform$[\mu, 0.9]$ if the edge (j, k) is in $G^{(0)}$ and similarly for $\mathbf{\Omega}_{jk}(1)$ using $G^{(4)}$. The signal strength is set to $\mu = 0.5$.

3. For any $z \in (0, 1)$, if $z \in [\ell/10, (\ell+1)/10]$ for some $\ell = 0, \ldots, 9$, $\mathbf{\Omega}_{jk}(z)$ is set to be an linear interpolation of $\mathbf{\Omega}_{jk}(\ell/10)$ and $\mathbf{\Omega}_{jk}((\ell+1)/10)$. We first rescale the diagonal of $\mathbf{\Omega}(z)$ by $\mathbf{\Omega}(z) + (1 - \lambda_{\min}(\mathbf{\Omega}(z)))\mathbf{I}_d$, where $\lambda_{\min}(\mathbf{\Omega}(z))$ is the minimum eigenvalue of $\mathbf{\Omega}(z)$ in order to make its minimum eigenvalue equal to one for each time $z \in (0, 1)$. We then get the covariance matrix $\mathbf{\Sigma}(z) = \mathbf{\Omega}^{-1}(z)$ and normalize its diagonal to all

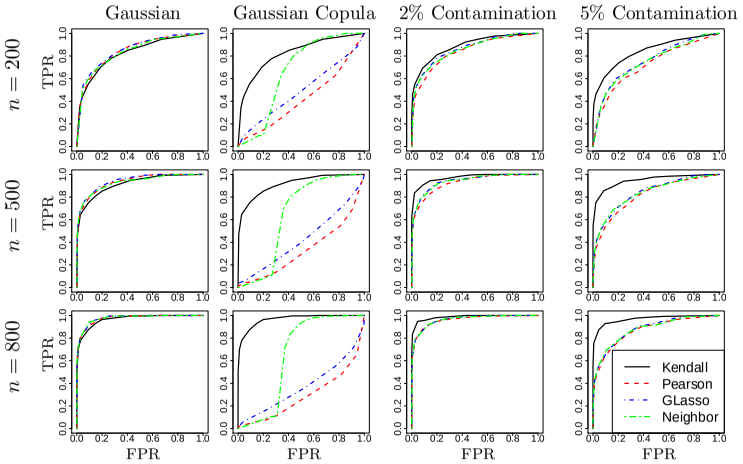


Figure 2: ROC curves for different estimators of the inverse correlation matrix estimator. The number of nodes is $d = 50$, the number of edges for each index value is $e = 25$ and the sample size is varied in $n \in \{200, 500, 800\}$. Each curve is obtained by averaging 100 simulation runs.

ones. An illustration of the time-varying graphical model is shown in Figure 1(b) - Figure 1(f).

Given the inverse correlation matrix $\mathbf{\Omega}(z)$, we consider four data generation schemes:

1. Gaussian, where $\boldsymbol{X} \mid Z = z \sim N(\mathbf{0}, \mathbf{\Sigma}(z))$;

2. Gaussian Copula, where $\boldsymbol{X} \mid Z = z \sim (\Phi(\boldsymbol{Y}_1(z)), \ldots, \Phi(\boldsymbol{Y}_d(z)))^T$, $\boldsymbol{Y} \mid Z = z \sim N(\mathbf{0}, \mathbf{\Sigma}(z))$ and $\Phi(\cdot)$ is the cumulative density function of standard normal distribution;

3. Gaussian with 2% contamination, where $\boldsymbol{X} \mid Z = z \sim N(\mathbf{0}, \mathbf{\Sigma}(z))$ and then 2% of locations are randomly chosen and replaced by $\pm N(3, 3)$;

4. Gaussian with 5% contamination, is uses the same data generating mechanism as before, but more of the observations are contaminated.

All the methods under consideration require a specification of the kernel function satisfying Assumption 4.3. We choose the Epanechnikov kernel $K(u) = 0.75(1 - u^2)\mathbb{1}(|u| < 1)$ which can be easily checked that it satisfies Assumption 4.3. The bandwidth parameter is set as $h = 0.35/n^{1/5}$ for all four methods in order to facilitate easier comparison. For the calibrated CLIME, we choose $\gamma = 0.5$ in (3). We then estimate a sequence of inverse correlation matrices and graphs by changing the penalty parameter λ over a large range of values. Let $\widehat{E}(z)$ be the estimated edge set and $E^*(z)$ is the true edge set at the index value

n	$\mu_0 = 0$	$\mu_0 = 0.4$	$\mu_0 = 0.6$	$\mu_0 = 0.9$
600	**0.108**	0.722	0.892	0.994
800	**0.090**	0.822	0.900	0.998
1,000	**0.056**	0.736	0.968	1.000

Table 1: Size (bold) and power of the local edge presence test $H_0 : \mathbf{\Omega}_{12}(0.5) = 0$ at significance level 95% with various signal strength μ_0. Results reported based on 500 simulation runs.

z. The true positive rate (TPR) and false positive rate (FPR) are defined as

$$\text{TPR}(z) = \frac{|\widehat{E}(z) \bigcap E^*(z)|}{|E^*(z)|} \qquad \text{and} \qquad \text{FPR}(z) = \frac{|\widehat{E}(z) \backslash E^*(z)|}{d(d-1)/2 - |E^*(z)|},$$

where $|S|$ denotes cardinality of the set S. To measure the quality of graph estimation for $z \in (0,1)$, we randomly choose 10 data points from $\{Z_i\}_{i\in[n]}$ and compute the averaged TPR and FPR on these 10 points as the overall TPR and FPR of the graph estimation. Figure 2 illustrates the ROC curves of the overall TPR and FPR for the four competing methods under four schemes for $n = 200, 500$ and 800. We can observe that the proposed estimator is slightly worse compared to other three estimators when the data are Gaussian conditionally on the index value. In this setting the data generating assumptions are satisfied for the other three procedures. On the other hand, when the data are generated according to the Gaussian copula distribution, the Gaussian assumption is violated and our estimator performs better compared to the other three estimators. The third and fourth columns of Figure 2 further illustrate that our estimator is more robust to corruption of data.

In addition to graph estimation accuracy, we consider the numerical performance of testing procedures proposed in the paper. We first focus on the local edge presence test introduced in Section 3.1. Consider the following null hypothesis $H_0 : \mathbf{\Omega}_{12}(0.5) = 0$. We choose the bandwidth $h = 0.9n^{-1/5}$, $\gamma = 0.5$ in (3) and the penalty parameter $\lambda = 0.2\big(h^2 + \sqrt{\log(d/h)/(nh)}\big)$ for the sample size $n \in \{600, 800, 1000\}$. The data generating process is the same as before, except that we set $\mathbf{\Omega}_{21}(z) = \mu_0$ for all $z \in (0,1)$ where $\mu_0 \in \{0, 0.4, 0.6, 0.9\}$. We use (21) to test the null hypothesis at significance level 95% and estimate the power based on 500 repetitions. The Q-Q plots of the testing statistic

	$n = 600$			$n = 800$			$n = 1,000$		
	$\mu = 0$	$\mu = 0.4$	$\mu = 0.9$	$\mu = 0$	$\mu = 0.4$	$\mu = 0.9$	$\mu = 0$	$\mu = 0.4$	$\mu = 0.9$
$k = 0$	**0.118**	0.954	0.958	**0.088**	0.958	0.976	**0.068**	0.977	0.992
$k = 2$	**0.128**	0.900	0.926	**0.090**	0.944	0.964	**0.070**	0.970	0.982
$k = 4$	**0.098**	**0.089**	**0.058**	**0.082**	**0.068**	**0.054**	**0.066**	**0.052**	**0.048**
$k = 8$	**0.090**	**0.074**	**0.054**	**0.066**	**0.058**	**0.048**	**0.054**	**0.052**	**0.050**

Table 2: Size (bold) and power of local edge present test $H_0 : G^*(z) \subset G$ for all $z \in (0,1)$ at significance level 95% with the super graph G being a k-nearest neighbor graph for $k \in \{0, 2, 4, 8\}$. Results reported based on 500 simulation runs.

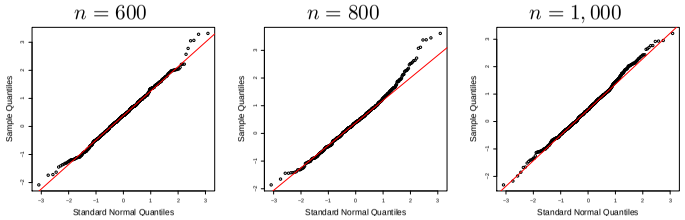


Figure 3: Q-Q plot of the testing statistic under the null hypothesis $H_0 : \mathbf{\Omega}_{12}(0.5) = 0$.

for various sample sizes are shown in Figure 3. The empirical distribution of the testing statistic is close to standard normal distribution. The results on the power of these tests are summarized in Table 5.1. When the signal strength $\mu_0 = 0$, the size of the test approaches the nominal value 0.05. When $\mu \geq 0.4$, the power approaches 1, showing validity of the proposed test.

We also conduct a simulation for the uniform edge presence test introduced in Section 3.3. We consider the null hypothesis $H_0 : G^*(z) \subset G$ for all $z \in (0,1)$, where the super graph G is a k-nearest neighbor graph for $k \in \{0, 2, 4, 8\}$. Here a k-nearest neighbor graph has edges only connecting a vertex to its closest k nodes. See Figure 1(a) for an illustration of a 4-nearest neighbor graph. When $k = 0$, it is a null graph whose adjacency matrix is identity. In order to illustrate the power of our test, we generate the nonzero entries of the inverse correlation matrix at anchor points from Uniform$[\mu, \max(\mu + 0.2, 0.9)]$ for $\mu = 0.4$ and 0.9. We also consider the setting where $\mathbf{\Omega}(z) = \mathbf{I}$ for all $z \in (0,1)$. The sample size is varied as $n \in \{600, 800, 1000\}$. The bandwidth and tuning parameter are set in the same way as for the local edge presence test. When computing the test statistic in (22) the suprema over $z \in (0,1)$ is approximated by taking the maximum of the statistic over 100 evenly spaced values in $(0,1)$, and similarly for the bootstrapped statistic in (23). The size and power of the test are summarized in Table 5.1. From the data generating mechanism, we see that the true graph is always a subgraph of the 4-nearest graph. Therefore, the null hypothesis is true when the number of nearest neighbors is $k \in \{4, 8\}$ or $\mathbf{\Omega}(z) = \mathbf{I}$. From the results reported in Table 5.1, we observe that the uniform edge presence test has the correct size as the sample size increases. The alternative hypothesis is true when $k \in \{0, 2\}$ and $\mathbf{\Omega}(z) \neq \mathbf{I}$. In this setting, the less edges the super graph has, the more powerful the test is. That is because we take maximum over more edges in (16).

5.2 Super Brain Test

We apply the super graph test in Section 3.2 to the ADHD-200 brain imaging data set (Biswal et al., 2010). The ADHD-200 data set is a collection of resting-state functional MRI (R-fMRI) of subjects with and without attention deficit hyperactive disorder (ADHD) (Eloyan et al., 2012; The ADHD-200 Consortium, 2012). The data set contains 776 subjects: 491 controls, 195 cases diagnosed with ADHD of various types and 88 subjects with withheld diagnosis for the purpose of a prediction competition. For each subject there are between 76 and 276 R-fMRI scans. We focus on 264 voxels as the regions of interest (ROI) extracted by Power et al. (2011). These voxels are representative of the functional areas corresponding

	$H_0 : G(8) \subset G(11.75)$	$H_0 : G(11.75) \subset G(20)$
Statistics	2.793	3.548
Quantiles	1.899	2.154

Table 3: Results on the super brain hypothesis tests at 95% significant level. The row named "Statistics" are the values of the testing statistic in (16). The row named "Quantiles" are the 95%-quantile estimators obtained by the bootstrap estimator in (19).

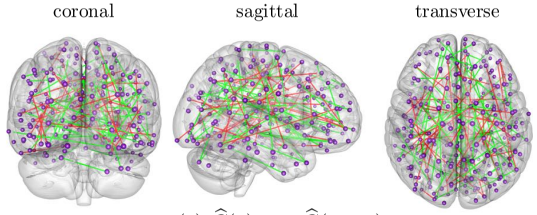
to the cerebral cortex and cerebellum. For each subject we also have covariates that include age, IQ, gender and handedness. For illustration purposes, we focus on 491 controls and explore how the structure of the neural network varies with age of subjects, which range from 7.08 to 21.83. Using the kernel smoothing technique to estimate the varying neural networks, Qiu et al. (2016) found that the connections tend to be denser at the age 21.83 than at ages 11.75 and 7.09. Such a discovery is also supported by Bartzokis et al. (2001) and Gelfand et al. (2003).

Based on these discoveries, a more interesting conjecture is whether the neural network is always growing with age. We can use our testing framework to investigate the claim that a neural network in a younger subject is a subgraph of a neural network in an older subject. Specifically, we choose three ages: 8, 11.75 and 20, and we are interested in the graphs $G(8)$, $G(11.75)$ and $G(20)$, where $G(z)$ is the true neural network at age z. The age 11.75 is chosen as the median age of subjects. We call the neural networks at ages 8, 11.75 and 20 as the junior, median and senior neural network, respectively. As we do not have access to the true networks, we will estimate them using the calibrated CLIME in (3). We first map the ages onto the interval $(0, 1)$ and set the tuning parameters for our procedure as $\gamma = 0.5$, the bandwidth $h = 0.002$ and the penalty parameter $\lambda = 0.03\big(h^2 + \sqrt{d/h}/(nh)\big)$, where the last two parameters are chosen through cross-validation. We estimate the neural networks $\widehat{G}(8)$, $\widehat{G}(11.75)$ and $\widehat{G}(20)$ by (3) at ages 8, 11.75 and 20. The estimated graphs are used as super graphs in defining the null hypothesis. In particular, we have the following two tests: $H_0 : G(8) \subset \widehat{G}(11.75)$ and $H_0 : G(11.75) \subset \widehat{G}(20)$. Although we use the random estimators as the super graphs in these hypotheses, the testing procedure is still valid. This is because of the small bandwidth $h = 0.002$ making the data used in network estimation independent to the data used in these two tests. Therefore, the test can still be reduced as a super graph test.

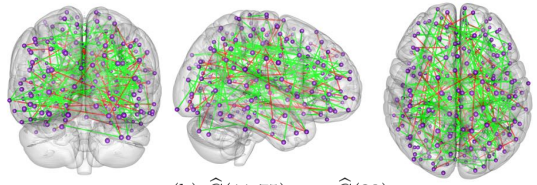
Figure 4 shows the differences among the junior, median and senior brains. We observe that even if a later neural network has more edges compared to the earlier one, many edges existing at an earlier stage disappear later in the development. This implies that the conjecture that neural networks grow with age is not supported by the data. Table 5.2 quantifies evidence against the null hypothesis.

6. Discussion

In this paper, we consider the time-varying graphical model under the framework of nonparanormal distribution. Although it contains many examples of heavy-tailed distributions, there are many other cases uncovered in this family. It will be interesting to explore the



(a) $\widehat{G}(8)$ v.s. $\widehat{G}(11.75)$



(b) $\widehat{G}(11.75)$ v.s. $\widehat{G}(20)$

Figure 4: The differences between junior, median and senior neural networks. $\widehat{G}(8)$, $\widehat{G}(11.75)$ and $\widehat{G}(20)$ denote the estimated graphs at age 8, 11.75 and 20. The first row is the difference between $\widehat{G}(8)$ and $\widehat{G}(11.75)$. The green lines are the edges existing in $\widehat{G}(11.75)$ but not in $\widehat{G}(8)$ and the red edges only exist in $\widehat{G}(8)$ but not $\widehat{G}(11.75)$. The second row is the difference between $\widehat{G}(11.75)$ and $\widehat{G}(20)$. The green edges only exist in $\widehat{G}(20)$ and the red edges only exist in $\widehat{G}(11.75)$.

estimation and inference methods for the time-varying graphical model under general heavy-tailed distributions with certain moment conditions. It is possible to incorporate existing methods including the Catoni's estimator (Catoni, 2012) and the median-of-means estimator (Hsu and Sabato, 2014) into the framework of this paper and conduct inference for the general heavy-tailed time-varying graphical models.

Acknowledgments

The authors are grateful for the support of NSF CAREER Award DMS1454377, NSF IIS1408910, NSF IIS1332109, NIH R01MH102339, NIH R01GM083084, and NIH R01HG06841. This work is also supported by an IBM Corporation Faculty Research Fund at the University of Chicago Booth School of Business. This work was completed in part with resources provided by the University of Chicago Research Computing Center.

Appendix A. Proof of Estimation Consistency

In the appendix, we use $c, C, c_1, C_1, c_2, C_2, \ldots$ to denote universal constants, independent of n and d, whose values may change from line to line.

In this section, we prove uniform rates of convergence for the covariance matrix estimator $\widehat{\mathbf{\Sigma}}(z)$ and the inverse covariance estimator $\widehat{\mathbf{\Omega}}(z)$. These rates are uniformly valid over both the index z and the kernel bandwidth h used for the estimator $\widehat{\tau}_{jk}(z)$ in (4).

A.1 Proof of Theorem 5

We apply the bias-variance decomposition for the kernel smoothed Kendall's tau statistic in the following two lemmas. The first lemma controls the variance term $|\widehat{\tau}_{jk}(z) - \mathbb{E}[\widehat{\tau}_{jk}(z)]|$ uniformly in $j, k \in [d]$, $z \in (0,1)$, and $h \in [h_l, h_u]$. The second lemma controls the bias term $|\mathbb{E}[\widehat{\tau}_{jk}(z)] - \tau_{jk}(z)|$.

Lemma 9 (Variance of Kendall's tau estimator) *Assume that $n^{-1}\log d = o(1)$ and the bandwidths $0 < h_l < h_u < 1$ satisfy $h_l n/\log(dn) \to \infty$ and $h_u = o(1)$. There exists a universal constant $C > 0$ such that, with probability $1-\delta$, for sufficiently large n,*

$$\sup_{j,k\in[d]} \sup_{h\in[h_l,h_u]} \sup_{z\in(0,1)} \frac{\sqrt{nh}}{\sqrt{\log(d/h) \vee \log\left(\delta^{-1}\log\left(h_u h_l^{-1}\right)\right)}} |\widehat{\tau}_{jk}(z) - \mathbb{E}\left[\widehat{\tau}_{jk}(z)\right]| \leq C.$$

Lemma 10 (Bias of Kendall's tau estimator) *Assume that the bandwidths $0 < h_l < h_u < 1$ satisfy $h_u = o(1)$. There exists a constant $C > 0$ such that*

$$\sup_{j,k\in[d]} \sup_{h\in[h_l,h_u]} \sup_{z\in(0,1)} \frac{|\mathbb{E}[\widehat{\tau}_{jk}(z)] - \tau_{jk}(z)|}{h^2 + 1/(nh)} \leq C.$$

We defer the proof of these two lemmas to Appendix D.

By the definition of $\widehat{\mathbf{\Sigma}}(z)$ in (7), for any $j, k \in [d]$ and $z \in (0,1)$, we have

$$\left|\widehat{\mathbf{\Sigma}}_{jk}(z) - \mathbf{\Sigma}_{jk}(z)\right| = \left|\sin\left(\pi\widehat{\tau}_{jk}(z)/2\right) - \sin\left(\pi\tau_{jk}(z)/2\right)\right| \leq \frac{\pi}{2}\left|\widehat{\tau}_{jk}(z) - \tau_{jk}(z)\right|, \tag{38}$$

where the last inequality is due to $|\sin(x) - \sin(y)| \leq |x - y|$ for any $x, y \in [-\pi/2, \pi/2]$. Therefore, the rate of $\widehat{\mathbf{\Sigma}}(z)$ can be bounded by the rate of $\widehat{\tau}_{jk}(z)$ up to a constant. Recall that $\mathbf{T} = [\tau_{jk}]_{jk}$ and $\widehat{\mathbf{T}} = [\widehat{\tau}_{jk}]_{jk}$. We have

$$\|\widehat{\mathbf{T}}(z) - \mathbf{T}(z)\|_{\max} \leq \sup_{j,k\in[d]} |\widehat{\tau}_{jk}(z) - \mathbb{E}\widehat{\tau}_{jk}(z)| + \sup_{j,k\in[d]} |\mathbb{E}\widehat{\tau}_{jk}(z) - \tau_{jk}(z)|.$$

Lemma 9 and Lemma 10 together with (38) give us

$$\sup_{h\in[h_l,h_u]} \sup_{z\in(0,1)} \frac{\left\|\widehat{\mathbf{\Sigma}}(z) - \mathbf{\Sigma}(z)\right\|_{\max}}{h^2 + \sqrt{(nh)^{-1}\left[\log(d/h) \vee \log\left(\delta^{-1}\log\left(h_u h_l^{-1}\right)\right)\right]}}$$

$$\leq \sup_{h\in[h_l,h_u]} \sup_{z\in(0,1)} \frac{\left\|\widehat{\mathbf{T}}(z) - \mathbf{T}(z)\right\|_{\max}}{h^2 + \sqrt{(nh)^{-1}\left[\log(d/h) \vee \log\left(\delta^{-1}\log\left(h_u h_l^{-1}\right)\right)\right]}} \leq 2\pi(C_1 + C_2),$$

with probability $1-\delta$, since $1/(nh) = o(1)$. We complete the proof of the theorem by setting $C_{\mathbf{\Sigma}} = 2\pi(C_1 + C_2)$.

A.2 Proof of Theorem 6

Using the uniform rate of convergence of $\widehat{\mathbf{\Sigma}}(z)$ established in Theorem 5, we establish the corresponding rate for $\widehat{\mathbf{\Omega}}(z)$ when estimated using the CLIME (Cai et al., 2011) or the calibrated CLIME (Zhao and Liu, 2014). Modifying the proofs in the above two papers, we can establish a bound on $\left\|\widehat{\mathbf{\Omega}} - \mathbf{\Omega}\right\|_{\max}$, if $\left\|\widehat{\mathbf{\Sigma}} - \mathbf{\Sigma}\right\|_{\max}$ is controlled. For simplicity, we recall the results for the calibrated CLIME estimator. Similar results for the CLIME estimator can be found in the proof of Theorem 6 in Cai et al. (2011).

Theorem 11 (Adapted from Zhao and Liu 2014) *Suppose $\mathbf{\Omega} \in \mathcal{U}_s(M, \rho)$ and the tuning parameter satisfies $s\lambda = o(1)$. On the event $\left\{\|\widehat{\mathbf{\Sigma}} - \mathbf{\Sigma}\|_{\max} \leq \lambda\right\}$, there exist universal constants C_1, C_2, C_3 such that the output of the calibrated CLIME satisfies*

$$\left\|\widehat{\mathbf{\Omega}} - \mathbf{\Omega}\right\|_{\max} \leq C_1 M^2 \lambda, \quad \left\|\widehat{\mathbf{\Omega}} - \mathbf{\Omega}\right\|_1 \leq C_2 s M \lambda \quad \text{and} \quad \max_{j\in[d]} \|\widehat{\mathbf{\Sigma}}\widehat{\mathbf{\Omega}}_j - \mathbf{e}_j\|_\infty \leq C_3 \lambda M.$$

The formal statements of Theorem IV.1 and Theorem IV.2 in Zhao and Liu (2014) are not the same as the statement above. The result of Theorem 11 is more general and directly follows from the proofs of Theorem IV.1 and Theorem IV.2 in Zhao and Liu (2014). For example, the last inequality in Theorem 11 follows from Equation (E.12) of Zhao and Liu (2014).

Let $\lambda_{n,h} = C_{\mathbf{\Sigma}}\left(h^2 + \sqrt{\log(dn)/(nh)}\right)$ and define the event

$$\mathcal{E} = \left\{\sup_{h\in[h_l,h_u]} \sup_{z\in(0,1)} \lambda_{n,h}^{-1}\left\|\widehat{\mathbf{\Sigma}}(z) - \mathbf{\Sigma}(z)\right\|_{\max} \leq 1\right\}.$$

Since the constants C_1, C_2 and C_3 in Theorem 11 are universal and the penalty parameter $\lambda \geq \lambda_{n,h}$, it follows that

$$\sup_{h\in[h_l,h_u]} \sup_{z\in(0,1)} \lambda^{-1}\left\|\widehat{\mathbf{\Omega}}(z) - \mathbf{\Omega}(z)\right\|_{\max} \leq C_1 M^2;$$

$$\sup_{h\in[h_l,h_u]} \sup_{z\in(0,1)} \lambda^{-1}\left\|\widehat{\mathbf{\Omega}}(z) - \mathbf{\Omega}(z)\right\|_1 \leq C_2 s M;$$

$$\sup_{z\in(0,1)} \max_{j\in[d]} \lambda^{-1} \cdot \|\widehat{\mathbf{\Omega}}_j^T\widehat{\mathbf{\Sigma}} - \mathbf{e}_j\|_\infty \leq C_3 M,$$

on the event $\mathcal{E}$. Theorem 5 gives us $\mathbb{P}(\mathcal{E}) \geq 1 - 1/d$, which completes the proof.

Appendix B. Asymptotic Properties of Testing Statistics

In this section, we prove asymptotic properties of the testing statistics for three kinds of hypothesis tests: (1) edge presence test in Theorem 1, (2) super-graph test in Theorem 3 and (3) uniform edge presence test in Theorem 4.

Let $\mathcal{S}_{j,z} = \{k \in [d] \mid \mathbf{\Omega}_{kj}(z) \neq 0\}$. For any index sets $\mathcal{S}, \mathcal{S}' \subset [d]$, we define $\mathbf{\Omega}_{\mathcal{S}\mathcal{S}'} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}'|}$ to be the submatrix of $\mathbf{\Omega}$ obtained from rows indexed by $\mathcal{S}$ and columns indexed by $\mathcal{S}'$. For any function $f(x)$, we define

$$\mathbb{G}_n^\xi[f] = \frac{1}{\sqrt{n}}\sum_{i=1}^n f(X_i)\cdot\xi_i, \tag{39}$$

where $\xi_1, \ldots, \xi_n \sim N(0,1)$. We also define $\mathbb{P}_\xi(\cdot) := \mathbb{P}(\cdot|\{Y_i\}_{i\in[n]})$ and $\mathbb{E}_\xi[\cdot] := \mathbb{E}[\cdot|\{Y_i\}_{i\in[n]}]$.

At a high-level, we establish the asymptotic results by carefully studying the Hoeffding decomposition of the kernel smoothed Kendall's tau estimator. From (4), we observe that $\widehat{\tau}_{jk}(z)$ is a quotient of two U-statistics. To study this quotient, we introduce some additional notation. For a fixed $j,k \in [d]$, we define the following bivariate function

$$g_{z|(j,k)}(y_i, y_{i'}) = \omega_z(z_i, z_{i'}) \operatorname{sign}(x_{ij} - x_{i'j}) \operatorname{sign}(x_{ik} - x_{i'k}), \tag{40}$$

for $y_i = (z_i, \boldsymbol{x}_i)$ and $y_{i'} = (z_{i'}, \boldsymbol{x}_{i'})$. Recalling the definition of $\omega_z(z_i, z_{i'})$ in (5), we have that

$$\widehat{\tau}_{jk}(z) = \mathbb{U}_n[g_{z|(j,k)}]/\mathbb{U}_n[\omega_z].$$

Let us define the Hoeffding decomposition of $g_{z|(j,k)}$ as

$$g^{(1)}_{z|(j,k)}(y) = \mathbb{E}[g_{z|(j,k)}(y,Y)] - \mathbb{E}\left[\mathbb{U}_n[g_{z|(j,k)}]\right], \tag{41}$$

$$g^{(2)}_{z|(j,k)}(y_1,y_2) = g_{z|(j,k)}(y_1,y_2) - g^{(1)}_{z|(j,k)}(y_1) - g^{(1)}_{z|(j,k)}(y_2) - \mathbb{E}\left[\mathbb{U}_n[g_{z|(j,k)}]\right]. \tag{42}$$

Then we can reformulate the centered U-statistic as

$$\mathbb{U}_n\left[g_{z|(j,k)}\right] - \mathbb{E}\left[\mathbb{U}_n\left[g_{z|(j,k)}\right]\right] = 2\mathbb{E}_n\left[g^{(1)}_{z|(j,k)}(Y_i)\right] + \mathbb{U}_n\left[g^{(2)}_{z|(j,k)}\right]. \tag{43}$$

In the above display, we decomposed the centered U-statistic into an empirical process and a higher order U-statistic. Our proof strategy is to study the asymptotic property of the leading empirical process term $2\mathbb{E}_n\left[g^{(1)}_{z|(j,k)}(Y_i)\right]$ and show that the higher order term can be ignored. Similarly, let

$$\omega^{(1)}_z(s) = \mathbb{E}[\omega_z(s,Z)] - \mathbb{E}\left[\mathbb{U}_n[\omega_z]\right], \tag{44}$$

$$\omega^{(2)}_z(s,t) = \omega_z(s,t) - \omega^{(1)}_z(s) - \omega^{(1)}_z(t) - \mathbb{E}\left[\mathbb{U}_n[\omega_z]\right], \tag{45}$$

which leads to the following Hoeffding decomposition

$$\mathbb{U}_n[\omega_z] - \mathbb{E}\left[\mathbb{U}_n[\omega_z]\right] = 2\mathbb{E}_n\left[\omega^{(1)}_z\right] + \mathbb{U}_n\left[\omega^{(2)}_z\right], \tag{46}$$

According to the heuristic approximation of $\widehat{S}_{z|(j,k)}\big(\widehat{\boldsymbol{\Omega}}_{k\setminus j}(z_0)\big)$ in (10), we have

$$\begin{aligned}
&\widehat{S}_{z|(j,k)}\big(\widehat{\boldsymbol{\Omega}}_{k\setminus j}(z)\big) \approx \boldsymbol{\Omega}_j^T(z)\big[\dot{\boldsymbol{\Sigma}}(z) \circ \big(\widehat{\mathbf{T}}(z) - \mathbf{T}(z)\big)\big]\boldsymbol{\Omega}_k(z) \\
&\approx [\mathbb{U}_n[\omega_z]]^{-1} \sum_{u,v\in[d]} \boldsymbol{\Omega}_{ju}(z)\boldsymbol{\Omega}_{kv}(z)\pi\cos\Big(\tau_{uv}(z)\frac{\pi}{2}\Big) \cdot \left[\mathbb{U}_n[g_{z|(u,v)}] - \mathbb{E}[\mathbb{U}_n[g_{z|(u,v)}]] \cdot \mathbb{U}_n[\omega_z]\right],
\end{aligned} \tag{47}$$

where the last "$\approx$" comes from (4) and Lemma 10. Combining (43) and (46) with (47),

$$\sqrt{nh} \cdot \widehat{S}_{z|(j,k)} \approx [\mathbb{U}_n[\omega_z]]^{-1}\mathbb{G}_n\big[J_{z|(j,k)}\big],$$

where the leading term of the Hoeffding decomposition is defined as

$$J_{z|(j,k)}(y') := \sum_{u,v\in[d]} \boldsymbol{\Omega}_{ju}(z)\boldsymbol{\Omega}_{kv}(z)\pi\cos\Big(\tau_{uv}(z)\frac{\pi}{2}\Big)\sqrt{h} \cdot \left[g^{(1)}_{z|(u,v)}(y') - \tau_{uv}(z)\omega^{(1)}_z(z')\right], \tag{48}$$

for any $y' = (z', \boldsymbol{x}')$. We find the asymptotic distribution of $[\mathbb{U}_n[\omega_z]]^{-1}\mathbb{G}_n\big[J_{z|(j,k)}\big]$ in the next part.

B.1 Proof of Theorem 1

Let the operator $u_n[\cdot]$ be defined as

$$u_n[H] = \sqrt{n} \cdot \big(\mathbb{U}_n[H] - \mathbb{E}[\mathbb{U}_n[H]]\big) \tag{49}$$

for any bivariate function $H(x,x')$. In order to prove Theorem 1, we need the convergence rate of terms related to the Kendall's tau estimator, especially the two lemmas below.

Lemma 12 *Suppose that $n^{-1}\log d = o(1)$ and the bandwidths $0 < h_l < h_u < 1$ satisfy $h_l n/\log(dn) \to \infty$ and $h_u = o(1)$. There exists a universal constant $C > 0$ such that with probability $1-\delta$,*

$$\sup_{j,k\in[d]} \sup_{h\in[h_l,h_u]} \sup_{z\in(0,1)} \left| \frac{\sqrt{h}}{\sqrt{\log(d/h) \vee \log\left(\delta^{-1}\log\left(h_u h_l^{-1}\right)\right)}} \Big(u_n\left[\omega_z\right] \vee u_n\left[g_{z|(j,k)}\right] \Big) \right| \le C, \tag{50}$$

for large enough n.

Lemma 13 *Suppose the bandwidths $0 < h_l < h_u < 1$ satisfy $h_u = o(1)$. Then*

$$\sup_{z\in(0,1)} \left|\mathbb{E}\left[\mathbb{U}_n\left[g_{z|(j,k)}\right]\right] - f_Z^2(z)\tau_{jk}(z)\right| = O(h^2), \tag{51}$$

$$\sup_{z\in(0,1)} \left|\mathbb{E}\left[\mathbb{U}_n\left[\omega_z\right]\right] - f_Z^2(z)\right| = O(h^2), \tag{52}$$

$$\sup_{z\in(0,1)} n^{-1}\mathbb{E}\left[u_n\left[g_{z|(j,k)}\right] \cdot u_n\left[\omega_z\right]\right] = O\big((nh)^{-1}\big), \tag{53}$$

$$\sup_{z\in(0,1)} n^{-1}\mathbb{E}\big[\left(u_n\left[\omega_z\right]\right)^2\big] = O\big((nh)^{-1}\big). \tag{54}$$

We defer the proof of the above two lemmas to Appendix E in the supplementary material.

Using Lemma 12 and Lemma 13, we have

$$\begin{aligned}
\inf_{z\in(0,1)} \mathbb{U}_n[\omega_z] &\ge \inf_{z\in(0,1)} \mathbb{E}[\mathbb{U}_n[\omega_z]] - \sup_{z\in(0,1)} n^{-1/2}|u_n[\omega_z]| \ge \underline{f}_Z^2/2, \\
\sup_{z\in(0,1)} \mathbb{U}_n[\omega_z] &\le \sup_{z\in(0,1)} \mathbb{E}[\mathbb{U}_n[\omega_z]] + \sup_{z\in(0,1)} n^{-1/2}|u_n[\omega_z]| \le 2\bar{\mathrm{f}}_Z^2,
\end{aligned} \tag{55}$$

with probability $1 - 1/d$ for sufficiently large n. The last inequality is due to the fact that f_Z is bounded from above and below, $h = o(1)$ and $\log(1/h)/nh = o(1)$.

Combining the above display with Lemma 19, we have

$$\begin{aligned}
&\left|\sqrt{nh} \cdot \widehat{S}_{z|(j,k)}\big(\widehat{\boldsymbol{\Omega}}_{k\setminus j}(z)\big) - [\mathbb{U}_n[\omega_z]]^{-1}\mathbb{G}_n\big[J_{z|(j,k)}\big]\right| \\
&\quad\le \left|\sqrt{nh} \cdot \mathbb{U}_n[\omega_z]\widehat{S}_{z|(j,k)}\big(\widehat{\boldsymbol{\Omega}}_{k\setminus j}(z)\big) - \mathbb{G}_n\big[J_{z|(j,k)}\big]\right| \cdot \Big(\inf_{z\in(0,1)} \mathbb{U}_n[\omega_z]\Big)^{-1} \le 2\underline{\mathrm{f}}_Z^{-2}n^{-c}.
\end{aligned} \tag{56}$$

Therefore, it suffices to derive the limiting distribution of $\mathbb{G}_n\big[J_{z|(j,k)}\big]$.

In order to apply central limit theorem to $\mathbb{G}_n\big[J_{z|(j,k)}\big]$ we check Lyapunov's condition. By the definition of $g^{(1)}_{z|(j,k)}$ in (41) and $\omega^{(1)}_z$ in (44), we have $\mathbb{E}[J_{z|(j,k)}(Y_i)] = 0$ for all $i \in [n]$. The matrix $\mathbf{\Theta}_z$ defined in (11) can be rewritten as

$$(\mathbf{\Theta}_z)_{jk} = \pi \cos\left(\tau_{jk}(z)\frac{\pi}{2}\right)\sqrt{h}\cdot\left[g^{(1)}_{z|(j,k)}(Y) - \tau_{jk}(z)\omega^{(1)}_z(Z)\right]. \tag{57}$$

In order to apply the Lyapunov condition to show the asymptotic normality, we begin to control the third moments of $\omega^{(1)}_z$ and $g^{(1)}_{z|(j,k)}$. We bound the third moment of $\omega^{(1)}_z(Z)$ by

$$\begin{aligned}
\sup_z \mathbb{E}\Big[\big|\omega^{(1)}_z(Z)\big|^3\Big] &= \sup_z |\mathbb{E}[K_h(z-Z)]|^3\mathbb{E}\Big[\big|K_h(z-Z) - \mathbb{E}[K_h(z-Z)]\big|^3\Big] \\
&\le \sup_z 8|\mathbb{E}[K_h(z-Z)]|^3\mathbb{E}\Big[\big|K_h(z-Z)\big|^3\Big] \\
&= \sup_z 8\big|f_Z(z) + O(h^2)\big|^3\cdot h^{-2}\int |K^3(t)|f_Z(z+th)dt \\
&\lesssim h^{-2}\cdot \bar{\mathrm{f}}_Z^4\int |K^3(t)|dt,
\end{aligned} \tag{58}$$

where we used that $(1+x)^3 \le 4(1+x^3)$ for $x > 0$ and $|\mathbb{E}[K_h(z-Z)]|^3 \le \mathbb{E}[|K_h(z-Z)|^3]$. By the definition of $g^{(1)}_{z|(j,k)}$, we also have

$$\mathbb{E}\Big[\big|g^{(1)}_{z|(j,k)}(Y)\big|^3\Big] \le 4\mathbb{E}\Big[\big|\mathbb{E}[g_{z|(j,k)}(Y',Y)\mid Y']\big|^3\Big] + 4\Big|\mathbb{E}\big[g_{z|(j,k)}(Y',Y)\big]\Big|^3, \tag{59}$$

where Y' is an independent copy of Y. Using (99),

$$\sup_z\Big|\mathbb{E}\big[g_{z|(j,k)}(Y',Y)\big]\Big|^3 = \sup_z\big|f_Z^2(z)\tau_{jk}(z) + O(h^2)\big|^3 \lesssim \bar{\mathrm{f}}_Z^6. \tag{60}$$

We now bound the conditional expectation in (59). From (A.3) of Mitra and Zhang (2014), denoting $\boldsymbol{x}' = (x'_1,\ldots,x'_d)^T$ and $y' = (z', \boldsymbol{x}')$, we have

$$\begin{aligned}
&\mathbb{E}\left[g_{z|(j,k)}(y',Y)|Z=s\right] = K_h(z'-z)K_h(s-z)\varphi\left(x'_j, x'_k, \mathbf{\Sigma}_{jk}(s)\right), \text{ where} \\
&\qquad \varphi(u,v,\rho) = 2\int \mathrm{sign}(u-x)\phi(x)\cdot\Phi\left(\frac{v-\rho x}{\sqrt{1-\rho^2}}\right)dx,
\end{aligned} \tag{61}$$

with $\phi(\cdot)$ and $\Phi(\cdot)$ being the probability density and cumulative distribution function of a standard normal variable, respectively. From (41), we have

$$\begin{aligned}
\mathbb{E}\left[g_{z|(j,k)}(y',Y)\right] &= \mathbb{E}\big[\mathbb{E}\left[g_{z|(j,k)}(y',Y)|Z\right]\big] \\
&= K_h(z'-z)\int K_h(s-z)\varphi\left(x'_j, x'_k, \mathbf{\Sigma}_{jk}(s)\right)f_Z(s)ds.
\end{aligned} \tag{62}$$

Let $\phi_\rho(x,y)$ be the density function of bivariate normal distribution with mean zero, variance one and correlation ρ. Notice that $\sup_{x,y,\rho}|\varphi(x,y,\rho)| \le 2$. Since the minimum eigenvalue

of $\mathbf{\Sigma}(z)$ is strictly positive for any z, there exists a $\gamma_\sigma < 1$ such that $\sup_z|\mathbf{\Sigma}_{jk}(z)| \le \gamma_\sigma < 1$ for any $j \ne k$. We also have $\sup_{x,y,\rho}|\phi_\rho(x,y)| \le (2\pi\sqrt{1-\gamma_\sigma^2})^{-1}$. By (62), for any $z \in (0,1)$

$$\begin{aligned}
&\mathbb{E}\Big[\big|\mathbb{E}[g_{z|(j,k)}(Y',Y)\mid Y']\big|^3\Big] \\
&= \frac{2}{\pi}\int h^{-2}K^3(t_1)f_Z(z+t_1h)\times \\
&\qquad\qquad \times\left|\int K(t_2)f_Z(z+t_2h)\varphi\left(u,v,\mathbf{\Sigma}_{jk}(z+t_2h)\right)dt_2\right|^3\phi_{\mathbf{\Sigma}_{jk}(z+t_1h)}(u,v)dt_1 \\
&\lesssim \frac{1}{h^2\pi\sqrt{1-\gamma_\sigma^2}}\int K^3(t_1)f_Z(z+t_1h)|f_Z(z)+O(h^2)|^3dt_1 \le h^{-2}\cdot\frac{\bar{\mathrm{f}}_Z^4\|K\|_3^3}{\pi\sqrt{1-\gamma_\sigma^2}}.
\end{aligned} \tag{63}$$

Combining (60), (63) with (59), we have

$$\mathbb{E}\Big[\big|g^{(1)}_{z|(j,k)}(Y)\big|^3\Big] \lesssim h^{-2}\left(\frac{\bar{\mathrm{f}}_Z^4\|K\|_3^3}{\pi\sqrt{1-\gamma_\sigma^2}}\right). \tag{64}$$

By the assumption of Theorem 4, there exists a $\theta_{\min} > 0$ such that

$$\mathrm{Var}(J_{z_0|(j,k)}(Y)) = \mathbb{E}(\mathbf{\Omega}_j^T(z_0)\mathbf{\Theta}_{z_0}\mathbf{\Omega}_k(z_0))^2 \ge \theta_{\min}\|\mathbf{\Omega}_j(z_0)\|_2^2\|\mathbf{\Omega}_k(z_0)\|_2^2. \tag{65}$$

We are now ready to check the Lyapunov's condition. We have

$$\begin{aligned}
\frac{\sum_{i=1}^n\mathbb{E}|J_{z_0|(j,k)}(Y_i)|^3}{n^{3/2}\mathrm{Var}^{3/2}(J_{z_0|(j,k)}(Y))} &\overset{(65)}{\le} \frac{(\theta_{\min}n)^{-3/2}}{\|\mathbf{\Omega}_j^T\|_2^3\|\mathbf{\Omega}_k\|_2^3}\sum_{i=1}^n\mathbb{E}|J_{z_0|(j,k)}(Y_i)|^3 \\
&\lesssim \frac{1}{n^{3/2}}\sum_{i=1}^n\mathbb{E}\|\mathrm{Vec}((\mathbf{\Theta}_{z_0})_{\mathcal{S}_{j,z_0},\mathcal{S}_{k,z_0}})\|_2^3.
\end{aligned} \tag{66}$$

Since $|\mathbf{\Theta}^{(i)}_{jk}| \le \pi\sqrt{h}|g^{(1)}_{z_0|(j,k)}(Y)| + \pi\sqrt{h}|\tau_{jk}(z_0)\omega^{(1)}_{z_0}(Z)|$, we have

$$\begin{aligned}
&\mathbb{E}\|\mathrm{Vec}((\mathbf{\Theta}_{z_0})_{\mathcal{S}_{j,z_0},\mathcal{S}_{k,z_0}})\|_2^3 \\
&\qquad \le |\mathcal{S}_{j,z_0}|^{3/2}|\mathcal{S}_{k,z_0}|^{3/2}\pi^3h^{3/2}\Big(\mathbb{E}\Big[\big|g^{(1)}_{z_0|(j,k)}(Y)\big|^3\Big] + |\tau_{jk}(z_0)|^3\mathbb{E}\Big[\big|\omega^{(1)}_{z_0}(Z)\big|^3\Big]\Big).
\end{aligned}$$

Using (58) and (64), together with $s^3/\sqrt{nh} = o(1)$,

$$\frac{\sum_{i=1}^n\mathbb{E}|J_{z_0|(j,k)}(Y_i)|^3}{n^{3/2}\mathrm{Var}^{3/2}(J_{z_0|(j,k)}(Y))} \lesssim \frac{s^3}{\sqrt{nh}} = o(1),$$

which implies that the Lyapunov's condition is satisfied. Moreover, by Lemma 12, for any $z_0 \in (0,1)$, $\mathbb{U}_n[\omega_{z_0}] - \mathbb{E}[\mathbb{U}_n[\omega_{z_0}]]$ converges to 0 in probability. Combining this with (52) and $h = o(1)$, we have that $\mathbb{U}_n[\omega_{z_0}]$ converges to $f_Z^2(z_0)$ in probability. Therefore, by the central limit theorem and Slutsky's theorem, for any $j,k \in [d]$,

$$\frac{[\mathbb{U}_n[\omega_{z_0}]]^{-1}\mathbb{G}_n\big[J_{z_0|(j,k)}\big]}{f_Z^{-2}(z_0)\big\{\mathbb{E}\big[\big(\mathbf{\Omega}_j^T\mathbf{\Theta}_{z_0}\mathbf{\Omega}_k\big)^2\big]\big\}^{1/2}} \rightsquigarrow N(0,1).$$

Combining with (56), the proof is complete.

B.2 Proof of Theorem 3

The strategy is to apply the theory for multiplier bootstrap developed in Chernozhukov et al. (2013) to the score function in (16). A similar strategy is applied to prove Theorem 4, whose proof is deferred to Section F.

Let $T_0(z) := \max_{(j,k)\in E^c} \mathbb{G}_n[J_{z|(j,k)}]$ and

$$S_0^B(z_0) = \max_{(j,k)\in E^c} \mathbb{G}_n^{\xi}\left[J_{z_0|(j,k)}\right] = \max_{(j,k)\in E^c} \frac{1}{\sqrt{n}} \sum_{i=1}^{n} J_{z_0|(j,k)}(Y_i)\cdot \xi_i$$

be the bootstrap counterpart to $T_0(z_0)$. Recall that $S^B(z)$ is defined in (19). We denote

$$\Delta_z := \max_{(j,k),(j',k')\in E^c} \left| \frac{1}{n}\sum_{i=1}^{n} \Big(J_{z|(j,k)}(Y_i)J_{z|j'k'}(Y_i) - \mathbb{E}[J_{z|(j,k)}(Y_i)J_{z|j'k'}(Y_i)] \Big) \right|.$$

In order to Use Theorem 3.2 in Chernozhukov et al. (2013), we check four conditions.

1. With probability $1-1/d$,
$$|S(z_0) - T_0(z_0)| \le \sup_{j,k\in[d]} \left| \sqrt{nh}\cdot \mathbb{U}_n[\omega_{z_0}]\widehat{S}_{z_0|(j,k)} \left(\widehat{\mathbf{\Omega}}_{k\backslash j}(z_0)\right) - \mathbb{G}_n\left[J_{z_0|(j,k)}\right] \right| \le n^{-c}.$$
2. With probability $1-1/d$, $\mathbb{P}_\xi\big(|S^B(z_0) - S_0^B(z_0)| \le n^{-c}\big) \ge 1 - 1/d$.
3. There exists a constant $c>0$, such that $\mathrm{Var}(\mathbb{G}_n\big[J_{z_0|(j,k)}\big]) > c$.
4. There exists a constant $c>0$, such that $\mathbb{P}(\Delta_{z_0} > n^{-c}) \le n^{-c}$.

The first condition is proven in Lemma 19. We defer the proof of the second condition in Lemma 20. The third condition is due to (65). The following of the proof verifies the last condition.

Define

$$\gamma_{z|(j,k,j',k')}(Y_i) = J_{z|(j,k)}(Y_i)J_{z_0|(j',k')}(Y_i) - \mathbb{E}[J_{z|(j,k)}(Y_i)J_{z|(j',k')}(Y_i)]. \tag{67}$$

We will apply Lemma A.1 in van de Geer (2008) on the concentration of empirical processes. For the self-consistence, we has restated the lemma in Lemma 34. In order to apply Lemma 34, we need to bound $\|\gamma_{z|(j,k,j',k')}\|_\infty$ and $n^{-1}\sum_{i=1}^n \mathbb{E}[\gamma^2_{z|(j,k,j',k')}(Z_i)]$. By the definition of $J_{z_0|(j,k)}$ in (48), we have for any $z_0\in(0,1)$,

$$\begin{aligned}
&\max_{(j,k),(j',k')\in E^c} \|\gamma_{z_0|(j,k,j',k')}\|_\infty \\
&\quad\le \max_{i\in[n]} \max_{(j,k)\in E^c} 2|J_{z_0|(j,k)}(Y_i)|^2 \\
&\quad\le \max_{(j,k)\in E^c} 2\|\mathbf{\Omega}_j(z_0)\|_1^2 \|\mathbf{\Omega}_k(z_0)\|_1^2 \cdot \pi^2 h \cdot \left(\|g^{(1)}_{z_0|(j,k)}(Y_i)\|_\infty + \|\omega^{(1)}_{z_0}(Z_i)\|_\infty \right)^2 \\
&\quad\le CM^2h^{-1},
\end{aligned} \tag{68}$$

where the second inequality follows from Hölder's inequality, similar to (66), and the final inequality is due to (84) and (94). Since the right hand size of (68) does not depend on z_0, we also have

$$\max_{z\in(0,1)} \max_{(j,k),(j',k')\in E^c} \|\gamma_{z|(j,k,j',k')}\|_\infty \le CM^2h^{-1} \text{ and} \tag{69}$$

$$\max_{z\in(0,1)} \max_{(j,k),(j',k')\in E^c} \mathbb{E}[\gamma^2_{z|(j,k,j',k')}(Z_i)] \le \max_{z\in(0,1)} \max_{(j,k),(j',k')\in E^c} \|\gamma_{z|(j,k,j',k')}\|_\infty^2 \le CM^4h^{-2}. \tag{70}$$

According to Lemma 34, the expectation of Δ_{z_0} is bounded by

$$\mathbb{E}[\Delta_{z_0}] \lesssim \sqrt{\frac{2M^4\log(2d)}{nh^2}} + \frac{M^2\log(2d)}{nh}.$$

Since $\log d/(nh^2) = o(n^{-\epsilon})$, there exists $c_1>0$ such that $\mathbb{E}[\Delta_{z_0}] \le n^{-2c_1}$ for sufficiently large n. By Markov's inequality, $\mathbb{P}(\Delta_{z_0} > n^{-c_1}) \le n^{c_1}\mathbb{E}[\Delta_{z_0}] \le n^{-c_1}$ for sufficiently large n, which verifies the last condition.

By Theorem 3.2 in Chernozhukov et al. (2013),

$$\sup_{\alpha\in(0,1)} \left| \mathbb{P}_{H_0}\big(\psi_{z_0|G}(\alpha)=1\big) - \alpha \right| \lesssim n^{-c},$$

for some constant $c>0$, which completes the proof.

Appendix C. Proof of Theorem 8

In this section, we prove the minimax rate of convergence for estimating time-varying inverse covariance matrices. Section C.1 proves the minimax rate in terms of $\|\cdot\|_{\max}$ norm, while Section C.2 establishes the minimax rate for the $\|\cdot\|_1$ norm.

At a high-level, both results will use Le Cam's lemma applied to a finite collection of time-varying inverse covariance matrices. Given a time-varying inverse covariance matrix $\mathbf{\Omega}(\cdot)$, let $\mathbb{P}_{\mathbf{\Omega}}$ be the joint distribution of $(\boldsymbol{X}_1, Z_1), \ldots, (\boldsymbol{X}_n, Z_n)$ where $(\boldsymbol{X}_i, Z_i)$ are independent copies of $(\boldsymbol{X}, Z)$ with $Z \sim \mathrm{Unif}((0,1))$ and $\boldsymbol{X} \mid Z \sim N(0, \mathbf{\Omega}(z)^{-1})$. Let $\mathcal{U}_0 = \{\mathbf{\Omega}_0(\cdot), \mathbf{\Omega}_1(\cdot), \ldots, \mathbf{\Omega}_m(\cdot)\}$ be a collection of time-varying inverse covariance matrices, which are going to be defined later. With these we define the mixture distribution $\overline{\mathbb{P}} = m^{-1}\sum_{\ell=1}^m \mathbb{P}_{\mathbf{\Omega}_\ell}$. For two measures $\mathbb{P}$ and $\mathbb{Q}$, the total variation is given as $\|\mathbb{P}\wedge\mathbb{Q}\| := \int (d\mathbb{P}/d\mu)\wedge(d\mathbb{Q}/d\mu)d\mu$, where $d\mu$ is the Lebesgue measure. Now, Le Cam's lemma (Le Cam, 1973) gives us the following lower bound.

Lemma 14 *Let $\widehat{\mathbf{\Omega}}(\cdot)$ be any estimator of $\mathbf{\Omega}(\cdot)$ based on the data generated from the distribution family $\{\mathbb{P}_{\mathbf{\Omega}} \mid \mathbf{\Omega}\in\mathcal{U}_0\}$. Then*

$$\max_{1\le\ell\le m} \mathbb{E}\left[\sup_{z\in(0,1)} \|\widehat{\mathbf{\Omega}}(z) - \mathbf{\Omega}_\ell(z)\|_{\max} \right] \ge r_{\min} \|\overline{\mathbb{P}} \wedge \mathbb{P}_{\mathbf{\Omega}_0}\|,$$

where $r_{\min} = \min_{1\le\ell\le m} \sup_{z\in(0,1)} \|\mathbf{\Omega}_0(z) - \mathbf{\Omega}_\ell(z)\|_{\max}$.

We will use the above lemma in the following two subsections.

C.1 Proof of Maximum Norm in (36)

We start by constructing the collection of inverse covariance matrices $\mathcal{U}_0$. Let $\mathbf{\Omega}_0(\cdot) \equiv \mathbf{I}$. Let $M_0 = \lceil c_0(n/\log(dn))^{1/5} \rceil$ where c_0 is some constant to be determined. Then

$$\mathcal{U}_0 = \Big\{ \mathbf{\Omega}_{(j,m)}(\cdot) \mid \mathbf{\Omega}^{-1}_{(j,m)}(z) = \mathbf{\Sigma}_{(j,m)}(z) = \mathbf{I} + \tau_m(z)\mathbf{E}_{jj}, z \in (0,1), j \in [d-1], m \in [M_0] \Big\},$$

where $\mathbf{E}_{jj} = \mathbf{e}_j \mathbf{e}_{j+1}^T + \mathbf{e}_{j+1}\mathbf{e}_j^T$, $\mathbf{e}_j$ is the j-th canonical basis in $\mathbb{R}^d$ and for any $m \in [M_0]$,

$$\tau_m(z) = Lh^2 K_0\Big(\frac{z - z_m}{h}\Big), \quad z_m = \frac{m - 1/2}{M_0}, \quad h = 1/M_0. \tag{71}$$

Here, $K_0(\cdot)$ is any function supported on $(-1/2, 1/2)$ and satisfying $\|K_0\|_{\sup} \le K_{\max}$. For example, consider $K_0(\cdot) = a\psi(2z)$, where $\psi(z) = \exp(-1/(1-z^2))\mathbb{1}(|z| \le 1)$, for some sufficiently small a (Tsybakov, 2009). It is easy to check that $\mathcal{U}_0 \subset \overline{\mathcal{U}}_s(M, \rho, L)$ if $n^{-1}\log(dn) = o(1)$.

For any $j \in [d-1]$, $m \in [M_0]$, we have

$$\sup_{z \in (0,1)} \|\mathbf{\Omega}_{(j,m)}(z) - \mathbf{\Omega}_0(z)\|_{\max} \ge \Big\| \frac{\tau_m(\cdot)}{1 - \tau_m^2(\cdot)} \Big\|_{\sup} \ge \|\tau_m\|_{\sup} \ge Lh^2 K_0(0)$$

by direct calculation, which gives us $r_{\min} \ge Lh^2 K_0(0) \asymp (\log(dn)/n)^{2/5}$.

In the remainder of the proof we show that for $\overline{\mathbb{P}} = ((d-1)M_0)^{-1} \sum_{j \in [d], m \in [M_0]} \mathbb{P}_{\mathbf{\Omega}_{(j,m)}}$, we have $\|\mathbb{P}_{\mathbf{\Omega}_0} \wedge \overline{\mathbb{P}}\| \ge 1/2$. Let f_{jm} be the density of $\mathbb{P}_{\mathbf{\Omega}_{(j,m)}}$ for $j \in [d-1], m \in [M_0]$ and f_0 the density of $\mathbb{P}_0$. Under our setting,

$$f_{jm}((\boldsymbol{x}_i, z_i)_{i=1}^n) = \prod_{i=1}^n g_{jm}(\boldsymbol{x}_i)\mathbb{1}\{z_i \in (0,1)\},$$

where g_{jm} is the density function of $N(0, \mathbf{\Sigma}_{(j,m)})$. Note that for any two densities f and f'

$$\int (f \wedge f') d\mu = 1 - \frac{1}{2}\int |f - f'| d\mu \ge 1 - \frac{1}{2}\Big(\int \frac{f^2}{f'} d\mu - 1 \Big)^{1/2}.$$

Therefore, it suffices to show that

$$\Delta := \int \Big(((d-1)M_0)^{-1} \sum_{j,m} f_{jm} \Big)^2 / f_0 \, d\mu - 1 \to 0. \tag{72}$$

Expanding the square of the mixture in (72), we have

$$\frac{1}{((d-1)M_0)^2} \Bigg\{ \sum_{j=1}^{d-1}\sum_{m=1}^{M_0} \int \frac{f_{jm}^2}{f_0} d\mu + \sum_{j=1}^{d-1} \sum_{m_1 \ne m_2} \int \frac{f_{jm_1} f_{jm_2}}{f_0} d\mu + \sum_{j_1 \ne j_2} \sum_{m_1, m_2 \in [M_0]} \int \frac{f_{j_1 m_1} f_{j_2 m_2}}{f_0} d\mu \Bigg\} - 1.$$

To proceed, we recall the following result of Ren et al. (2015) given in their equation (94). Let g_i be the density function of $N(0, \mathbf{\Sigma}_i)$ for $i = 0, 1, 2$. Then

$$\int \frac{g_1 g_2}{g_0} = \Big[\det\Big(\mathbf{I} - \mathbf{\Sigma}_0^{-1}(\mathbf{\Sigma}_1 - \mathbf{\Sigma}_0)\mathbf{\Sigma}_0^{-1}(\mathbf{\Sigma}_2 - \mathbf{\Sigma}_0) \Big) \Big]^{-1/2}. \tag{73}$$

Using the above display, for $j_1 \ne j_2$ and $m_1, m_2 \in [M_0]$, since $(\mathbf{\Sigma}_{j_1} - \mathbf{I})(\mathbf{\Sigma}_{j_2} - \mathbf{I}) = 0$, we have $\int f_{j_1 m_1} f_{j_2 m_2} / f_0 d\mu = 1$. For $|j_1 - j_2| = 1$, we have $\int f_{j_1 m_1} f_{j_2 m_2}/f_0 d\mu = [\det(\mathbf{I}_3 - \mathbf{M}_3)]^{-n/2} = 1$, where $\mathbf{M}_3 \in \mathbb{R}^{3\times 3}$ with $(\mathbf{M}_3)_{13} = \tau_{m_1}(z)\tau_{m_2}(z)$ and the other entries are zero. For any $m_1, m_2 \in [M_0]$ and $j \in [d-1]$, we have

$$\int \frac{f_{jm_1} f_{jm_2}}{f_0} d\mu = \prod_{1 \le i \le n} \int_0^1 \big(1 - \tau_{m_1}(z_i)\tau_{m_2}(z_i)\big)^{-1} dz_i \;\; = \Big[\int_0^1 \big(1 - \tau_{m_1}(z)\tau_{m_2}(z)\big)^{-1} dz \Big]^n. \tag{74}$$

If $m_1 \ne m_2$, since the supports of $\tau_{m_1}(\cdot)$ and $\tau_{m_2}(\cdot)$ are disjoint, (74) implies

$$\int f_{jm_1} f_{jm_2} / f_0 d\mu = 1.$$

Finally, if $m_1 = m_2 = m$, we have

$$\begin{aligned} \int \frac{f_{jm}^2}{f_0} d\mu &\le \Big[\int_0^1 \big(1 - \tau_m^2(z)\log^{4/5}(dn)\big)^{-1} dz \Big]^n \\ &\le \Big[\frac{M_0 - 1}{M_0} + \frac{1}{M_0}(1 - L^2h^4K_{\max}^2)^{-1} \Big]^n = \Big[1 + \frac{L^2h^4K_{\max}^2}{M_0(1 - L^2h^4K_{\max}^2)} \Big]^n. \end{aligned} \tag{75}$$

In summary,

$$\begin{aligned} \Delta &= \frac{1}{((d-1)M_0)^2} \sum_{j=1}^{d-1}\sum_{m=1}^{M} \Big(\int \frac{f_{jm}^2}{f_0} d\mu - 1 \Big) \\ &\le \exp\Big[-\log((d-1)M_0) + n\log\Big(1 + \frac{L^2h^4K_{\max}^2}{M_0(1 - L^2h^4K_{\max}^2)} \Big) \Big] - \frac{1}{(d-1)M_0}. \end{aligned}$$

Recall that $M_0 = \lceil c_0(n/\log(dn))^{1/5} \rceil$ and $h = 1/M_0$. We choose c_0 sufficiently large such that $c_0^5 > 1 - L^2K_{\max}^2$. Using $\log(1+x) \le x$ and $x/(1-x) \le 2x$ for $x \in (0, 1/2)$, we have

$$\Delta \le \exp\Big[-\log(d(c_0 n/\log(dn))^{1/5}) + c_0^{-5}L^2K_{\max}^2 \log(dn) \Big] - \frac{1}{c_0 dn^{1/5}} \to 0.$$

Since $r_{\min} \ge Lh^2K_0(0) \asymp (\log(dn)/n)^{2/5}$, the proof of (36) is complete by Lemma 14.

C.2 Proof of ℓ_1 Norm in (37)

In order to show the lower bound for $\|\cdot\|_1$, we need to construct a different $\mathcal{U}_0$. We still choose $\mathbf{\Omega}_0(\cdot) \equiv \mathbf{I}$. Let $\mathcal{B}$ be the set of matrices defined as

$$\mathcal{B} = \Big\{ \mathbf{B} \in \mathbb{R}^{d\times d} \Big| \, B = \begin{pmatrix} 0 & \boldsymbol{v} \\ \boldsymbol{v} & 0 \end{pmatrix} \text{ s.t. } \boldsymbol{v} \in \{0,1\}^{d-1}, \|\boldsymbol{v}\|_0 = s, \boldsymbol{v}_2 = 0 \Big\}.$$

With this, we define

$$\mathcal{U}_0' = \{\mathbf{\Omega}_{(j,m)}(\cdot) \mid \mathbf{\Omega}_{(j,m)}^{-1}(z) = \mathbf{\Sigma}_{(j,m)}(z) = \mathbf{I} + \tau_m(z)B_j, \text{ for } z \in (0,1), B_j \in \mathcal{B}, m \in [M_0]\},$$

where the index j corresponds to an element in the set $\mathcal{B}$ and the function $\tau_m(\cdot)$ is defined in (71). Let $D := |\mathcal{B}| = \binom{d-2}{s}$. We still choose $M_0 = \lceil c_0(n/\log(dn))^{1/5} \rceil$ for some constant c_0. It can be easily shown that $\mathcal{U}_0 \subset \overline{\mathcal{U}}_s(M, \rho, L)$ if $s^2 \log(dn)/n^{4/5} = o(1)$.

The rest of the proof is similar to the proof in Section C.1. For any $j \in [D]$, $m \in [M_0]$,

$$\sup_{z \in (0,1)} \|\mathbf{\Omega}_{(j,m)}(z) - \mathbf{\Omega}_0(z)\|_1 \geq s\|\tau_m\|_{\sup} \geq sLh^2 K_0(0),$$

giving $r_{\min} \geq sLh^2K_0(0) \asymp s(\log(dn)/n)^{2/5}$. We proceed to show $\|\mathbb{P}_{\mathbf{\Omega}_0} \wedge \overline{\mathbb{P}}\| \geq 1/2$, where $\overline{\mathbb{P}} = (DM_0)^{-1} \sum_{j \in [D], m \in [M_0]} \mathbb{P}_{\mathbf{\Omega}_{(j,m)}}$, by proving

$$\Delta := \int \Big(\frac{1}{DM_0} \sum_{j,m} f_{jm}\Big)^2 / f_0 \, d\mu - 1 \to 0.$$

We establish the above display by modifying the proof of Lemma 2 in Ren et al. (2015).

Let $J(j_1, j_2) = \text{Vec}(B_{j_1})^T \text{Vec}(B_{j_2})/2$, where $\text{Vec}(\mathbf{M}) = (\mathbf{M}_1^T, \ldots, \mathbf{M}_d^T)^T$ for any matrix $\mathbf{M} = [\mathbf{M}_1, \ldots, \mathbf{M}_d]$. From (73) and the definition of $\mathcal{B}$, we have

$$\begin{aligned} \int \frac{f_{j_1 m_1} f_{j_1 m_2}}{f_0} d\mu &= \Big[\int_0^1 \Big[\det\Big(\mathbf{I} - \tau_{m_1}(z)\tau_{m_2}(z) B_{j_1} B_{j2}\Big)\Big]^{-1/2} dz\Big]^n \\ &= \Big[\int_0^1 (1 - J(j_1, j_2)\tau_{m_1}(z)\tau_{m_2}(z))^{-1} dz\Big]^n. \end{aligned} \tag{76}$$

Similar to (74), when $m_1 \neq m_2$, (76) yields that $\int f_{j_1 m_1} f_{j_2 m_2}/f_0 d\mu = 1$. Similar to (75), we also have

$$\int \frac{f_{j_1 m} f_{j_2 m}}{f_0} d\mu \leq \Big[1 + \frac{2J(j_1, j_2) L^2 h^4 K_{\max}^2}{M_0}\Big]^n.$$

Since

$$\big|\{(j_1, j_2) \in [D]^2 \mid J(j_1, j_2) = j\}\big| = \binom{d-2}{s}\binom{s}{j}\binom{d-2-s}{s-j},$$

we have

$$\begin{aligned} \Delta &= \frac{1}{D^2 M_0^2} \sum_{m=1}^{M_0} \sum_{0 \leq j \leq s} \sum_{J(j_1, j_2) = j} \Big(\int \frac{f_{j_1 m} f_{j_2 m}}{f_0} d\mu - 1\Big) \\ &\leq \frac{1}{D^2 M_0^2} \sum_{m=1}^{M_0} \sum_{0 \leq j \leq s} \sum_{J(j_1, j_2) = j} \Big(\Big[1 + \frac{2jL^2 h^4 K_{\max}^2}{M_0}\Big]^n - 1\Big) \\ &\leq \frac{1}{D^2 M_0^2} \sum_{m=1}^{M_0} \sum_{0 \leq j \leq s} \binom{d-2}{s}\binom{s}{j}\binom{d-2-s}{s-j} \Big[1 + \frac{2jL^2 h^4 K_{\max}^2}{M_0}\Big]^n. \end{aligned}$$

Let $a_0 = 2c_0^{-5} L^2 K_{\max}^2$. Similar to the proof of Lemma 6 in Ren et al. (2015), we have

$$\Big[1 + \frac{2jL^2 h^4 K_{\max}^2 \log(dn)}{M_0}\Big]^n \leq \exp\big(2nM^{-1} jL^2 h^4 K_{\max}^2\big) \leq (nd)^{a_0 j}.$$

This further gives us

$$\begin{aligned} \Delta &\leq \frac{1}{M_0^2} \sum_{m=1}^{M_0} \sum_{1 \leq j \leq s} \binom{s}{j}\binom{d-2-s}{s-j} (nd)^{a_0 j} \Big/ \binom{d-2}{s} \\ &= \frac{1}{M_0^2} \sum_{m=1}^{M_0} \sum_{0 \leq j \leq s} \frac{1}{j!} \Big(\frac{s!(d-s)!}{(s-j)!}\Big)^2 \frac{(nd)^{a_0 j}}{(d-2)!(d-2s+j-2)!} \\ &\leq \frac{1}{M_0} \sum_{1 \leq j \leq s} \Big(\frac{s^2 (nd)^{a_0}}{d-s}\Big)^j, \end{aligned}$$

where the last inequality is due to $\frac{s!}{(s-j)!} \leq s^j$ and $\frac{(d-2)!(d-2s+j-2)!}{[(d-s)!]^2} \geq (d-s)^j$. By assumption $s^{-v} d \geq 1$ for some $v > 2$, and therefore

$$\Delta \leq \frac{1}{c_0(n/\log n)^{1/5}} \sum_{0 \leq j \leq s} (2d^{2/v-1}(dn)^{a_0})^j \leq \frac{2c_0(n/\log(dn))^{-1/5} d^{2/v-1+a_0} n^{a_0}}{(1 - 2d^{2/v-1+2a_0})},$$

for sufficiently large d, $d - s \geq d/2$, $d \geq n$. By choosing c_0 in $a_0 = 2c_0^{-5} L^2 K_{\max}^2$ sufficiently small, so that $a_0 < \min(1/5, 1/v - 1/2)$, we have $\Delta \to 0$. This completes the proof.

Appendix D. Convergence Rate of Kendall's Tau Estimator

In this section, we study the rate of convergence of kernel Kendall's tau estimator $\widehat{\tau}_{jk}(z)$ in (4) to $\tau_{jk}(z)$ uniformly in the bandwidth $h \in [h_l, h_u]$, the index $z \in (0,1)$ and the dimension $j, k \in [d]$. We start by establishing the bias-variance decomposition for $\widehat{\tau}_{jk}(z)$ and then show how to bound the bias and variance separately.

Recall that in (40), we define

$$g_{z|(j,k)}(y_i, y_{i'}) = \omega_z(z_i, z_{i'}) \operatorname{sign}(x_{ia} - x_{i'a}) \operatorname{sign}(x_{ib} - x_{i'b}),$$

and $\omega_z(z_i, z_{i'})$ is defined in (5). The estimator $\widehat{\tau}_{jk}(z)$ can be written as a quotient of two U-statistics

$$\widehat{\tau}_{jk}(z) = \mathbb{U}_n[g_{z|(j,k)}]/\mathbb{U}_n[\omega_z].$$

Recall that the operator $u_n[\cdot]$ is defined as

$$u_n[H] = \sqrt{n} \cdot \big(\mathbb{U}_n[H] - \mathbb{E}[\mathbb{U}_n[H]]\big)$$

for any bivariate function $H(x, x')$. With this, following an argument similar to that in Equation (3.45) of Pagan and Ullah (1999), we have the following decomposition

$$\begin{aligned} \widehat{\tau}_{jk}(z) &= \frac{\mathbb{E}[\mathbb{U}_n\left[g_{z|(j,k)}\right]] + \mathbb{U}_n\left[g_{z|(j,k)}\right] - \mathbb{E}[\mathbb{U}_n\left[g_{z|(j,k)}\right]]}{\mathbb{E}[\mathbb{U}_n[\omega_z]]} \left[1 + \frac{\mathbb{U}_n[\omega_z] - \mathbb{E}[\mathbb{U}_n[\omega_z]]}{\mathbb{E}[\mathbb{U}_n[\omega_z]]}\right]^{-1} \\ &= \frac{\mathbb{E}\left[\mathbb{U}_n\left[g_{z|(j,k)}\right]\right]}{\mathbb{E}[\mathbb{U}_n[\omega_z]]} + \frac{u_n\left[g_{z|(j,k)}\right]}{\sqrt{n}\,\mathbb{E}[\mathbb{U}_n[\omega_z]]} - \frac{\mathbb{E}\left[\mathbb{U}_n\left[g_{z|(j,k)}\right]\right] \cdot u_n\left[\omega_z\right]}{\sqrt{n}\left(\mathbb{E}[\mathbb{U}_n[\omega_z]]\right)^2} \\ &\quad + n^{-1} O\left(u_n\left[g_{z|(j,k)}\right] \cdot u_n\left[\omega_z\right] + \left(u_n\left[\omega_z\right]\right)^2\right), \end{aligned} \tag{77}$$

under the condition that

$$\left|\frac{u_n\left[\omega_z\right]}{\sqrt{n}\mathbb{E}[\mathbb{U}_n[\omega_z]]}\right| < 1 \quad \text{and} \quad \mathbb{E}[\mathbb{U}_n[\omega_z]] \neq 0. \tag{78}$$

Taking expectation on both sides of (77), we obtain

$$\mathbb{E}\left[\widehat{\tau}_{jk}(z)\right] = \frac{\mathbb{E}\left[\mathbb{U}_n\left[g_{z|(j,k)}\right]\right]}{\mathbb{E}\left[\mathbb{U}_n[\omega_z]\right]} + n^{-1}O\bigg(\mathbb{E}\Big[u_n\left[g_{z|(j,k)}\right]\cdot u_n\left[\omega_z\right]\Big] + \mathbb{E}\left[\left(u_n\left[\omega_z\right]\right)^2\right]\bigg). \tag{79}$$

With this, we are ready to prove Lemma 9 and Lemma 10.

D.1 Proof of Lemma 9

We first check that the condition in (78) is satisfied under the assumptions. Using (50) and (52), for sufficiently large n,

$$\sup_{j,k\in[d]}\sup_{z\in(0,1)}\left|\frac{u_n\left[\omega_z\right]}{\sqrt{n}\mathbb{E}[\mathbb{U}_n[\omega_z]]}\right| \lesssim \sqrt{\frac{\log(d/h)\vee\log\left(\delta^{-1}\log\left(h_uh_l^{-1}\right)\right)}{nh}}\cdot\frac{1}{f_Z^2(z)+O(h^2)} < 1.$$

Since f_Z is bounded from below, $\mathbb{E}[\mathbb{U}_n[\omega_z]] = f_Z^2(z)+O(h^2) \geq \mathrm{f}_Z/2 > 0$ for large enough n. Therefore, condition in (78) holds and the expansion in (77) is valid.

From (77) and (79), we have

$$\widehat{\tau}_{jk}(z,h) - \mathbb{E}\left[\widehat{\tau}_{jk}(z,h)\right] = \underbrace{\frac{u_n\left[g_{z|(j,k)}\right]}{\sqrt{n}\,\mathbb{E}[\mathbb{U}_n[\omega_z]]}}_{I_1} - \underbrace{\frac{\mathbb{E}\left[\mathbb{U}_n\left[g_{z|(j,k)}\right]\right]\cdot u_n\left[\omega_z\right]}{\sqrt{n}\left(\mathbb{E}[\mathbb{U}_n[\omega_z]]\right)^2}}_{I_2} + I_3, \tag{80}$$

where

$$I_3 = n^{-1}O\left(u_n\left[g_{z|(j,k)}\right]\cdot u_n\left[\omega_z\right] + \left(u_n\left[\omega_z\right]\right)^2 + \mathbb{E}\Big[u_n\left[g_{z|(j,k)}\right]\cdot u_n\left[\omega_z\right]\Big] + \mathbb{E}\left[\left(u_n\left[\omega_z\right]\right)^2\right]\right).$$

We bound I_1, I_2 and I_3 separately. Using (50) and (52), we have that

$$\begin{aligned}&\sup_{j,k\in[d]}\sup_{h\in[h_l,h_u]}\sup_{z\in(0,1)}\left|\frac{\sqrt{nh}\cdot|I_1|}{\sqrt{\log(d/h)\vee\log\left(\delta^{-1}\log\left(h_uh_l^{-1}\right)\right)}}\right|\\ &\leq \sup_{j,k\in[d]}\sup_{h\in[h_l,h_u]}\sup_{z\in(0,1)}\left|\frac{\sqrt{h}}{\sqrt{\log(d/h)\vee\log\left(\delta^{-1}\log\left(h_uh_l^{-1}\right)\right)}}\frac{|u_n\left[g_{z|(j,k)}\right]|}{f_Z^2(z)+O(h^2)}\right| \leq C,\end{aligned}$$

with probability $1-\delta$ for large enough n. Similarly, using (50), (51) and (52), we also have

$$\begin{aligned}&\sup_{j,k\in[d]}\sup_{h\in[h_l,h_u]}\sup_{z\in(0,1)}\left|\frac{\sqrt{nh}\cdot|I_2|}{\sqrt{\log(d/h)\vee\log\left(\delta^{-1}\log\left(h_uh_l^{-1}\right)\right)}}\right|\\ &\leq \sup_{j,k\in[d]}\sup_{h\in[h_l,h_u]}\sup_{z\in(0,1)}\left|\frac{\sqrt{h}}{\sqrt{\log(d/h)\vee\log\left(\delta^{-1}\log\left(h_uh_l^{-1}\right)\right)}}\frac{|u_n\left[\omega_z\right]|(f_Z^2(z)\tau_{jk}(z)+O(h^2))}{(f_Z^2(z)+O(h^2))^2}\right| \leq C,\end{aligned}$$

with probability $1-\delta$. Finally, using (50), (53) and (54), we have

$$\begin{aligned}&\sup_{j,k\in[d]}\sup_{h\in[h_l,h_u]}\sup_{z\in(0,1)}\left|\frac{nh\cdot|I_3|}{\log(d/h)\vee\log\left(\delta^{-1}\log\left(h_uh_l^{-1}\right)\right)}\right|\\ &\lesssim \sup_{j,k\in[d]}\sup_{h\in[h_l,h_u]}\sup_{z\in(0,1)}\left|\frac{2h}{\log(d/h)\vee\log\left(\delta^{-1}\log\left(h_uh_l^{-1}\right)\right)}\Big(u_n\left[\omega_z\right]\vee u_n\left[g_{z|(j,k)}\right]\Big)^2\right|\\ &\quad+\sup_{j,k\in[d]}\sup_{h\in[h_l,h_u]}\sup_{z\in(0,1)}\left|\frac{nh\cdot O((nh)^{-1})}{\log(d/h)\vee\log\left(\delta^{-1}\log\left(h_uh_l^{-1}\right)\right)}\right| \leq C,\end{aligned}$$

with probability $1-\delta$.

Combining the above three displays with (80) completes the proof.

D.2 Proof of Lemma 10

It follows from Lemma 13 and the decomposition in (79) that

$$\begin{aligned}&\sup_{j,k\in[d]}\sup_{h\in[h_l,h_u]}\sup_{z\in(0,1)}\frac{|\mathbb{E}[\widehat{\tau}_{jk}(z)]-\tau_{jk}(z)|}{h^2+1/(nh)}\\ &\leq \sup_{j,k\in[d]}\sup_{h\in[h_l,h_u]}\sup_{z\in(0,1)}\left[\frac{f_Z^2(z)\tau_{jk}(z)+O(h^2)}{f_Z^2(z)+O(h^2)}-\tau_{jk}(z)+O((nh)^{-1})\right]\frac{1}{h^2+(nh)^{-1}}\\ &\leq \sup_{j,k\in[d]}\sup_{h\in[h_l,h_u]}\sup_{z\in(0,1)}\left[\frac{O(h^2+(nh)^{-1})}{f_Z^2(z)+O(h^2)}\right]\frac{1}{h^2+(nh)^{-1}} \leq C.\end{aligned}$$

Appendix E. Concentration of U-statistics

In this section, we study certain properties of the U-statistics used in this paper. We state and prove Lemma 12 and Lemma 13, which were used to establish results on the rate of convergence of Kendall's tau estimator in Appendix D. In particular, we will prove the following two results in this section, with the notations on U-statistics and empirical processes defined in Appendix D.

E.1 Proof of Lemma 12

At a high-level, we will use the Hoeffding decomposition to represent the U-statistics $\mathbb{U}_n[\omega_z]$ and $\mathbb{U}_n[g_{z|(j,k)}]$ defined in (43) and (46). Next, we will use concentration inequalities for suprema of empirical processes and U-statistics to bound individual terms in the decomposition.

Recall that (43) and (46) give

$$n^{-1/2}\cdot u_n\left[g_{z|(j,k)}\right] = \mathbb{U}_n\left[g_{z|(j,k)}\right] - \mathbb{E}\left[\mathbb{U}_n\left[g_{z|(j,k)}\right]\right] = 2\mathbb{E}_n\left[g^{(1)}_{z|(j,k)}(Y_i)\right] + \mathbb{U}_n\left[g^{(2)}_{z|(j,k)}\right]$$

and

$$n^{-1/2}\cdot u_n\left[\omega_z\right] = \mathbb{U}_n[\omega_z] - \mathbb{E}\left[\mathbb{U}_n[\omega_z]\right] = 2\mathbb{E}_n\left[\omega_z^{(1)}\right] + \mathbb{U}_n\left[\omega_z^{(2)}\right].$$

In order to bound $u_n[\omega_z]$ and $u_n\left[g_{z|(j,k)}\right]$ it is sufficient to bound $g^{(1)}_{z|(j,k)}$, $g^{(2)}_{z|(j,k)}$, $\omega^{(1)}_z$ and $\omega^{(2)}_z$. We do so in the following lemmas.

Lemma 15 *We assume $n^{-1}\log d = o(1)$ and the bandwidths $0 < h_l < h_u < 1$ satisfy $h_l n/\log(dn) \to \infty$ and $h_u = o(1)$. There exist a universal constant $C > 0$ such that*

$$\sup_{j,k\in[d]}\sup_{h\in[h_l,h_u]}\sup_{z\in(0,1)}\left|\mathbb{G}_n\left[\frac{\sqrt{h}g^{(1)}_{z|(j,k)}}{\sqrt{\log(d/h)\vee\log\left(\delta^{-1}\log\left(h_u h_l^{-1}\right)\right)}}\right]\right| \le C \tag{81}$$

with probability $1-\delta$.

Proof The general strategy to show (81) can be separated into two steps.

Step 1. Splitting the supreme over $h \in [h_l, h_u]$ into smaller intervals such that the empirical process (81) is easier to bound for each interval. Let S be the smallest integer such that $2^S h_l \ge h_u$. Note that $S \le \log_2(h_u/h_l)$ and $[h_l, h_u] \subseteq \cup_{\ell=1}^S [2^{\ell-1}h_l, 2^\ell h_l] =: \mathcal{H}_\ell$. Then

$$\mathbb{P}\left[\sup_{z,j,k}\sup_{h\in[h_l,h_u]}\left|\mathbb{G}_n\left[\sqrt{h}g^{(1)}_{z|(j,k)}\right]\right| \ge t\right] \le \sum_{\ell=1}^S \mathbb{P}\left[\sup_{j,k\in[d]}\sup_{h\in\mathcal{H}_\ell}\sup_{z\in(0,1)}\left|\mathbb{G}_n\left[\sqrt{h}g^{(1)}_{z|(j,k)}\right]\right| \ge t\right]. \tag{82}$$

Step 2. We apply Talagrand's inequality (Bousquet, 2002) to each term in the summation on the right hand side of (82).

Consider the class of functions

$$\mathcal{F}_\ell = \left\{\sqrt{h}g^{(1)}_{z|(j,k)}\,\middle|\, h\in\mathcal{H}_\ell, z\in(0,1), j,k\in[d]\right\}.$$

In order to apply Talagrand's inequality to functions in the class $\mathcal{F}_\ell$, we have to check three conditions:

- The class $\mathcal{F}_\ell$ is uniformly bounded;
- Bounding the variance of $g^{(1)}_{z|(j,k)}$;
- Bounding the covering number of $\mathcal{F}_\ell$

We verify the above three conditions next. First, we show that the class $\mathcal{F}_\ell$ is uniformly bounded.

Recall that in (62), we show that

$$\mathbb{E}\left[g_{z|(j,k)}(y', Y)\right] = K_h(z'-z)\int K_h(s-z)\varphi\left(x'_j, x'_k, \mathbf{\Sigma}_{jk}(s)\right) f_Z(s)ds, \tag{83}$$

where $\varphi(\cdot)$ is defined in (61). Since $|\varphi(u,v,\rho)| \le 2$ for any u, v, and ρ, and $h \in \mathcal{H}_\ell$, the above display gives us

$$\begin{aligned}\sup_{z\in(0,1)}\max_{j,k\in[d]}\left\|g^{(1)}_{z|(j,k)}\right\|_\infty &\le \frac{2\|K\|_\infty}{h}\sup_{z\in(0,1)}\int K_h(s-z)f_Z(s)ds + \sup_{z\in(0,1)}\max_{j,k\in[d]}\mathbb{E}\left[\mathbb{U}_n\left[g_{z|(j,k)}\right]\right]\\ &\le \frac{2\|K\|_\infty}{h}\sup_{z\in(0,1)}\left(f_Z(z)+O(h^2)\right) + \bar{\mathrm{f}}_Z^2 + O(h^2) \le \frac{3\|K\|_\infty\bar{\mathrm{f}}_Z}{h},\end{aligned} \tag{84}$$

where the second inequality is due to (51). This results shows that the class $\mathcal{F}_\ell$ is uniformly bounded by $(2^\ell h_l)^{-1/2}3\|K\|_\infty\bar{\mathrm{f}}_Z$ and has the envelope $F_\ell = 3(2^{\ell-1}h_l)^{-1/2}\|K\|_\infty\bar{\mathrm{f}}_Z$.

Next, we bound the variance of $g^{(1)}_{z|(j,k)}$. Let $\bar{\varphi}_{u,v}(s) = \varphi(u,v,\mathbf{\Sigma}_{jk}(s))f_Z(s)$. We can rewrite (83) as

$$\mathbb{E}\left[g_{z|(j,k)}(y', Y)\right] = K_h(z'-z)\cdot(K_h * \bar{\varphi}_{u,v})(z),$$

where "$*$" denotes the convolution operator. Then we bound the convolution by

$$\sup_{u,v}\|K_h * \bar{\varphi}_{u,v}\|_\infty \le \sup_{u,v}\|K_h\|_1\|\bar{\varphi}_{u,v}\|_\infty \le 2\bar{\mathrm{f}}_Z, \tag{85}$$

where the first inequality follows using the Young's inequality for convolution. Now, for any fixed $h \in \mathcal{H}_\ell$ and $z \in (0,1)$, we have

$$\begin{aligned}&\sup_{z,j,k}\mathbb{E}\left[\left(\sqrt{h}g^{(1)}_{z|(j,k)}\right)^2\right] \le \sup_{z,j,k}\mathbb{E}\left[\max_{u,v}\left(\sqrt{h}g^{(1)}_{z|(u,v)}\right)^2\right]\\ &\le \sup_z 2\mathbb{E}\left[h\left(K_h(Z-z)\right)^2\right]\sup_{u,v}\|K_h * \bar{\varphi}_{u,v}\|^2_\infty + \sup_{z,j,k}2h\left\{\mathbb{E}\left[\mathbb{U}_n\left[g_{z|(j,k)}\right]\right]\right\}^2\\ &\le C\bar{\mathrm{f}}_Z^2\cdot\sup_z\left(\int\frac{1}{h}K^2\left(\frac{x-z}{h}\right)f_Z(x)dx\right) + Ch\bar{\mathrm{f}}_Z^2 + O(h^3)\\ &\le C\bar{\mathrm{f}}_Z^2\sup_z(f_Z(z)\|K\|_2^2 + O(h^2)) \le C\bar{\mathrm{f}}_Z^3||K||_2^2.\end{aligned} \tag{86}$$

The reason why we show a stronger result above on the expectation of maximal is because we need (86) in the proof of Section F.

Using the above display, we can bound the variance for any $\ell \in [S]$,

$$\sigma_\ell^2 := \sup_{f\in\mathcal{F}_\ell}\mathbb{E}\left[f^2\right] \le C\bar{\mathrm{f}}_Z^3||K||_2^2. \tag{87}$$

Finally, we need a bound on the covering number of $\mathcal{F}_\ell$. Using Lemma 28 we have that

$$\sup_Q N(\mathcal{F}_\ell, L_2(Q), \epsilon) \le \frac{Cd^2}{(2^{\ell-1}h_\ell)^{v+9}\epsilon^{v+6}}.$$

Theorem 3.12 of Koltchinskii (2011) then gives us that for some universal constant C,

$$\mathbb{E}\left[\sup_{f\in\mathcal{F}_\ell}|n\mathbb{E}_n[f]|\right] \le C\sqrt{n\log\left(\frac{Cd}{2^{\ell-1}h_\ell}\right)} + \frac{C}{(2^{\ell-1}h_l)^{1/2}}\log\left(\frac{Cd}{2^{\ell-1}h_\ell}\right).$$

Furthermore, since $h_l n/\log(dn) \to \infty$, the above display simplifies to

$$\mathbb{E}\left[\sup_{f\in\mathcal{F}_\ell}|n\mathbb{E}_n[f]|\right] \le C\sqrt{n\log\left(\frac{Cd}{2^{\ell-1}h_l}\right)}. \tag{88}$$

Using Theorem 2.3 of Bousquet (2002) together with (87), (88), and $n^{-1}\log\left(dh^{-1}\right) = o(1)$, we obtain

$$\sup_{j,k\in[d]}\sup_{h\in\mathcal{H}_\ell}\sup_{z\in(0,1)}\left|\mathbb{G}_n\left[\sqrt{h}g^{(1)}_{z|(j,k)}\right]\right| \le C\left(\sqrt{\log\left(\frac{Cd}{2^{\ell-1}h_l}\right)} + \sqrt{\log(1/\delta)}\right) \tag{89}$$

with probability $1-\delta$. Observe that $2^{\ell-1}h_l \geq h/2$ for any $h \in \mathcal{H}_\ell$. Therefore, combining (89), (82), we obtain that for some constant $C > 0$

$$\sup_{j,k\in[d]} \sup_{h\in[h_l,h_u]} \sup_{z\in(0,1)} \left| \mathbb{G}_n \left[\frac{\sqrt{h} g^{(1)}_{z|(j,k)}}{\sqrt{\log(d/h) \vee \log\left(\delta^{-1}\log\left(h_u h_l^{-1}\right)\right)}} \right] \right| \leq C,$$

with probability $1-\delta$. ■

Lemma 16 *We assume $n^{-1}\log d = o(1)$ and the bandwidths $0 < h_l < h_u < 1$ satisfy $h_l n/\log(dn) \to \infty$ and $h_u = o(1)$. There exists a universal constant $C > 0$ such that*

$$\sup_{j,k\in[d]} \sup_{h\in[h_l,h_u]} \sup_{z\in(0,1)} \left| \frac{nh}{\log(d/h) \vee \log\left(\delta^{-1}\log\left(h_u h_l^{-1}\right)\right)} \mathbb{U}_n \left[g^{(2)}_{z|(j,k)} \right] \right| \leq C$$

with probability $1-\delta$.

Proof The method to prove this lemma is similar to the proof of Lemma 15. The only difference is that instead of bounding the empirical process, we bound a suprema of a U-statistic process. Therefore, we will use Theorem 33 instead of Talagrand's inequality.

We apply the trick of splitting $[h_l, h_u]$ again. Let S be the smallest integer such that $2^S h_l \geq h_u$. We have that $S \leq \log_2(h_u/h_l)$ and $[h_l, h_u] \subseteq \cup_{\ell=1}^S [2^{\ell-1}h_l, 2^\ell h_l]$. For simplicity, we define $\mathcal{H}_\ell = [2^{\ell-1}h_l, 2^\ell h_l]$. Therefore,

$$\mathbb{P}\left[\sup_{z,j,k} \sup_{h\in[h_l,h_u]} \left| h\mathbb{U}_n \left[g^{(2)}_{z|(j,k)} \right] \right| \geq t \right] \leq \sum_{j=1}^S \mathbb{P}\left[\sup_{z,j,k} \sup_{h\in\mathcal{H}_\ell} \left| h\mathbb{U}_n \left[g^{(2)}_{z|(j,k)} \right] \right| \geq t \right]. \tag{90}$$

As $g^{(2)}_{z|(j,k)}$ is a degenerate kernel, we can use Theorem 33 to bound the right hand side of (90). Consider the class of functions

$$\mathcal{F}^{(2)}_{\ell|(j,k)} = \left\{ (2^{\ell-1}h_l L) h g^{(2)}_{z|(j,k)} \,\middle|\, h \in \mathcal{H}_\ell, z \in (0,1) \right\} \text{ and } \mathcal{F}^{(2)}_\ell = \left\{ f \in \mathcal{F}^{(2)}_{\ell|(j,k)} \mid j,k \in [d] \right\}.$$

where $L^{-1} := 7\|K\|_\infty^2 \bar{\mathrm{f}}_Z^2$. From the expansion in (42), with (84) and (51), we have

$$\sup_{z\in(0,1)} \max_{j,k\in[d]} \left\| g^{(2)}_{z|(j,k)} \right\|_\infty \leq C\|K\|_\infty^2 \bar{\mathrm{f}}_Z^2 h^{-2}, \qquad \text{for } h \in [h_l, h_u].$$

Therefore, the class $\mathcal{F}^{(2)}_\ell$ is uniformly bounded by 1 and has the envelope $F^{(2)}_\ell = 1$, which verifies the condition in (159). Furthermore, by (86) and (51), we can bound the variance

as

$$\begin{aligned}
&\sup_{z,j,k} \mathbb{E}\left[\left(h g^{(2)}_{z|(j,k)} \right)^2 \right] \\
&\quad \lesssim \sup_{z,j,k} \left\{ 4h^2 \mathbb{E}\left[\left(g_{z|(j,k)}(Y_i, Y_{i'}) \right)^2 \right] + h\mathbb{E}\left[\left(\sqrt{h} g^{(1)}_{z|(j,k)} \right)^2 \right] + 4\{ h\mathbb{E}[\mathbb{U}_n[g_{z|(j,k)}]] \}^2 \right\} \\
&\quad \lesssim \sup_z h^{-2}\mathbb{E}\left[K\left((Z_i - z)/h \right)^2 K\left((Z_{i'} - z)/h \right)^2 \right] + \bar{\mathrm{f}}_Z^3 h + \bar{\mathrm{f}}_Z^2 h^2 \\
&\quad \lesssim \sup_z \left(h^{-1} \int K\left((x-z)/h \right)^2 f_Z(x) dx \right)^2 + \bar{\mathrm{f}}_Z^3 h + \bar{\mathrm{f}}_Z^2 h^2 \\
&\quad \lesssim \bar{\mathrm{f}}_Z^2 \|K\|_2^4 + \bar{\mathrm{f}}_Z^3 h + \bar{\mathrm{f}}_Z^2 h^2 \lesssim \bar{\mathrm{f}}_Z^2 \|K\|_2^4.
\end{aligned}$$

Therefore, for any $\ell \in [S]$,

$$\sup_{f\in\mathcal{F}^{(2)}_\ell} \mathbb{E}\left[f^2 \right] \lesssim 2\bar{\mathrm{f}}_Z^2 \|K\|_2^4 (2^{\ell-1}h_l L)^2 := \sigma_\ell^2 < 1.$$

According to Lemma 28, we have that

$$\sup_Q N(\mathcal{F}^{(2)}_{\ell|(j,k)}, L_2(Q), \epsilon) \leq C \frac{(2^{\ell-1}h_l L)^{-4v-15}}{(2^{\ell-1}h_l)^{2v+18} \epsilon^{4v+15}}.$$

By setting $t = C(2^{\ell-1}h_l)[\log\left(C/(2^{\ell-1}h_l)\right) \vee \log(C/\delta)]$ in (160) for a sufficiently large constant C, as $h_l n/\log(dn) \to \infty$, we have

$$\begin{aligned}
n\sigma_\ell^2 = n(2^{\ell-1}h_l)^2 &\geq \log\left(1/(2^{\ell-1}h_l)\right) \vee \log(1/\delta) \\
&= \frac{t}{\sigma_\ell} \geq C' \left(\frac{\log(1/(2^{\ell-1}h_l))}{\log n} \right)^{3/2} \log\left(\frac{2}{2^{\ell-1}h_l} \right),
\end{aligned}$$

for some constant $C' > 0$. This verifies the condition (161). Now (160) gives us that with probability $1 - [(2^{\ell-1}h_\ell) \vee \delta]$,

$$\sup_{h\in\mathcal{H}_\ell} \sup_{z\in(0,1)} (2^{\ell-1}h_l L) h \mathbb{U}_n \left[g^{(2)}_{z|(j,k)} \right] \leq Cn^{-1}(2^{\ell-1}h_l)[\log\left(1/(2^{\ell-1}h_l)\right) \vee \log(1/\delta)]. \tag{91}$$

Since $2^{\ell-1}h_l \geq h/2$ for any $h \in \mathcal{H}_\ell$, applying (90) and (91) with the union bound over $j,k \in [d]$, we obtain

$$\sup_{j,k\in[d]} \sup_{h\in[h_l,h_u]} \sup_{z\in(0,1)} \left| \frac{nh}{\log(d/h) \vee \log\left(\delta^{-1}\log\left(h_u h_l^{-1}\right)\right)} \mathbb{U}_n \left[g^{(2)}_{z|(j,k)} \right] \right| \leq C, \tag{92}$$

with probability $1-(h_u \vee \delta)$. As $h_u = o(1)$, (92) follows with probability larger than $1-\delta$. ■

Lemma 17 *We suppose $n^{-1}\log d = o(1)$ and the bandwidths $0 < h_l < h_u < 1$ satisfy $h_l n/\log(dn) \to \infty$ and $h_u = o(1)$. There exists a universal constant $C > 0$ such that*

$$\sup_{h\in[h_l,h_u]}\sup_{z\in(0,1)}\left|\mathbb{G}_n\left[\frac{\sqrt{h}\omega_z^{(1)}}{\sqrt{\log(1/h)\vee\log\left(\delta^{-1}\log\left(h_u h_l^{-1}\right)\right)}}\right]\right| \le C$$

with probability $1-\delta$.

Proof The proof is similar to that of Lemma 15. We again decompose $[h_l, h_u] \subseteq \cup_{\ell=1}^{S}\mathcal{H}_\ell$, where $\mathcal{H}_\ell = [2^{\ell-1}h_l, 2^\ell h_l]$ and S is the smallest integer such that $2^S h_l \ge h_u$. By the union bound,

$$\mathbb{P}\left[\sup_{h\in[h_l,h_u]}\sup_{z\in(0,1)}\left|\mathbb{G}_n\left[h^2\omega_z^{(1)}\right]\right| \ge t\right] \le \sum_{j=1}^{S}\mathbb{P}\left[\sup_{h\in\mathcal{H}_\ell}\sup_{z\in(0,1)}\left|\mathbb{G}_n\left[h^2\omega_z^{(1)}\right]\right| \ge t\right]. \tag{93}$$

Talagrand's inequality (Bousquet, 2002) is applied once again to control (93). Consider the class of functions

$$\mathcal{K}_\ell = \left\{\sqrt{h}\omega_z^{(1)} \mid h\in\mathcal{H}_\ell, z\in(0,1)\right\}.$$

According to the definition of $\omega_z^{(1)}$ in (44), we have

$$\omega_z^{(1)}(s) = K_h(s-z)\mathbb{E}[K_h(z-Z)] - (\mathbb{E}[K_h(z-Z)])^2.$$

We can bound the supremum norm by

$$\sup_{z\in(0,1)}\|\omega_z^{(1)}\|_\infty \le 2\bar{\mathrm{f}}_Z\|K\|_\infty h^{-1}, \tag{94}$$

which implies that the envelope function is $F_\ell = 2\bar{\mathrm{f}}_Z\|K\|_\infty(2^{\ell-1}h_\ell)^{-1/2}$. Similar to (86), we can also bound the variance by

$$\begin{aligned}\sup_z\mathbb{E}\left[\left(\sqrt{h}\omega_z^{(1)}\right)^2\right] &\le \sup_z 2\mathbb{E}\left[h\left(K_h(Z-z)\right)^2\right]\cdot(\mathbb{E}[K_h(z-Z)])^2 + \sup_z 2h\left\{\mathbb{E}\left[\mathbb{U}_n\left[\omega_z\right]\right]\right\}^2\\ &\le 2(\bar{\mathrm{f}}_Z^3\|K\|_2^2 + O(h^2)) + 2h\bar{\mathrm{f}}_Z^2 + O(h^3) \le 2\bar{\mathrm{f}}_Z^3||K||_2^2 := \sigma_\ell^2.\end{aligned} \tag{95}$$

Using Lemma 29 we have

$$\sup_Q N(\mathcal{K}_\ell, L_2(Q), \epsilon) \le \frac{C}{(2^{\ell-1}h_\ell)^4\epsilon^{2v+1}}\cdot\frac{\|F_\ell\|_{L_2}}{2\|K\|_\infty(2^{\ell-1}h_\ell)^{-1/2}},$$

which combined with Theorem 3.12 of Koltchinskii (2011) with $h_l n/\log(dn)\to\infty$ implies that

$$\mathbb{E}\left[\sup_{f\in\mathcal{K}_\ell}|n\mathbb{E}_n[f]|\right] \le C\sqrt{n\log\left(\frac{C}{2^{\ell-1}h_l}\right)}.$$

Theorem 2.3 of Bousquet (2002) derives that for some constant $C > 0$

$$\sup_{h\in\mathcal{H}_\ell}\sup_{z\in(0,1)}\left|\mathbb{G}_n\left[\sqrt{h}\omega_z^{(1)}\right]\right| \le C\left(\sqrt{\log\left(\frac{C}{2^{\ell-1}h_l}\right)} + \sqrt{\log(1/\delta)}\right) \tag{96}$$

with probability $1-\delta$. Using the union bound in (96) with (93), we have

$$\sup_{h\in[h_l,h_u]}\sup_{z\in(0,1)}\left|\mathbb{G}_n\left[\frac{\sqrt{h}\omega_z^{(1)}}{\sqrt{\log(1/h)\vee\log\left(\delta^{-1}\log\left(h_u h_l^{-1}\right)\right)}}\right]\right| \le C,$$

for some constant $C > 0$ with probability $1-\delta$. ■

Lemma 18 *We assume $n^{-1}\log d = o(1)$ and the bandwidths $0 < h_l < h_u < 1$ satisfy $h_l n/\log(dn)\to\infty$ and $h_u = o(1)$. There exists a universal constant $C > 0$ such that for sufficiently large n,*

$$\sup_{h\in[h_l,h_u]}\sup_{z\in(0,1)}\left|\frac{nh}{\log(1/h)\vee\log\left(\delta^{-1}\log\left(h_u h_l^{-1}\right)\right)}\mathbb{U}_n\left[\omega_z^{(2)}\right]\right| \le C$$

with probability $1-\delta$.

Proof The proof is similar to that of Lemma 16. Instead of applying Lemma 29, we use Lemma 28. We omit the details of the proof. ■

Let

$$r(d,n,h,\delta,h_u,h_l) = \frac{\log(d/h)\vee\log\left(\delta^{-1}\log\left(h_u h_l^{-1}\right)\right)}{nh}.$$

Applying Lemma 15, Lemma 16, Lemma 17 and Lemma 18 to (43) and (46), we obtain that with probability $1-\delta$, there exists a constant $C > 0$ such that

$$\sup_{j,k\in[d]}\sup_{h\in[h_l,h_u]}\sup_{z\in(0,1)}\left|\frac{n^{-1/2}\cdot u_n\left[\omega_z\right]\vee u_n\left[g_{z|(j,k)}\right]}{r^{1/2}(d,n,h,\delta,h_u,h_l)+r(d,n,h,\delta,h_u,h_l)}\right| \le C/2. \tag{97}$$

Since $\log(dn)/(nh_l) = o(1)$ and $h_u = o(1)$, we have $r(d,n,h,\delta,h_u,h_l) \le r^{1/2}(d,n,h,\delta,h_u,h_l)$ for sufficiently large n. Using this in (97), we have Lemma 12 proved.

E.2 Proof of Lemma 13

The high-level idea for proving this lemma is to write the expectations as the integrals and apply Taylor expansions to the density functions and other nonparametric functions. Afterwards, we bound the remainder terms of the Taylor expansions.

We compute $\mathbb{E}\left[\mathbb{U}_n\left[g_{z|(j,k)}\right]\right]$ first. Using Corollary 32, we have

$$\mathbb{E}\left[\mathbb{U}_n\left[g_{z|(j,k)}\right]\right] = \frac{2}{\pi}\mathbb{E}\left[K_h(Z_1-z)K_h(Z_2-z)\arcsin\left(\frac{\mathbf{\Sigma}_{jk}(Z_1)+\mathbf{\Sigma}_{jk}(Z_2)}{2}\right)\right], \tag{98}$$

where Z_1, Z_2 are independent and equal to Z in distribution. After a change of variables, we can further expand the right hand side of (98) as

$$\frac{2}{\pi}\iint K(t_1)K(t_2)f_Z(z+t_1h)f_Z(z+t_2h)\arcsin\left(\frac{\mathbf{\Sigma}_{jk}(z+t_1h)+\mathbf{\Sigma}_{jk}(z+t_2h)}{2}\right)dt_1dt_2.$$

Using the Taylor series expansion of $\arcsin(\cdot)$, we have

$$\arcsin\left(\frac{\mathbf{\Sigma}_{jk}(z+t_1h)+\mathbf{\Sigma}_{jk}(z+t_2h)}{2}\right)$$
$$=\arcsin\left(\mathbf{\Sigma}_{jk}(z)\right)+\left(\frac{t_1h\dot{\mathbf{\Sigma}}_{jk}(z)+t_2h\dot{\mathbf{\Sigma}}_{jk}(z)}{2}+\frac{(t_1h)^2\ddot{\mathbf{\Sigma}}_{jk}(\bar{z})+(t_2h)^2\ddot{\mathbf{\Sigma}}_{jk}(\bar{z})}{4}\right)\frac{1}{\sqrt{1-\bar{\rho}^2}},$$

where $\bar{z}$ is between z and $z+t_1h$, and $\bar{\rho}$ is between $\mathbf{\Sigma}_{jk}(z)$ and $(\mathbf{\Sigma}_{jk}(z+t_1h)+\mathbf{\Sigma}_{jk}(z+t_2h))/2$. Since the minimum eigenvalue of $\mathbf{\Sigma}(z)$ is strictly positive for any z, there exists a $\gamma_\sigma<1$ such that $\sup_z|\mathbf{\Sigma}_{jk}(z)|\leq\gamma_\sigma<1$ for any $j\neq k$. We thus have $(1-\bar{\rho}^2)^{-1/2}\leq(1-\gamma_\sigma^2)^{-1/2}<\infty$. Similarly, we can expand $f_Z(z+th)=f_Z(z)+th\dot{f}_Z(z)+(th)^2\ddot{f}_Z(\widetilde{z})/2$, where $\widetilde{z}\in(z,z+th)$. Therefore, using regularity conditions on f_Z and $\mathbf{\Sigma}(\cdot)$, we obtain

$$\max_{j,k\in[d]}\sup_{z\in(0,1)}\left|\mathbb{E}\left[\mathbb{U}_n\left[g_{z|(j,k)}\right]\right]-f_Z^2(z)\tau_{jk}(z)\right|\leq\frac{M_\sigma\bar{f}_Z}{\sqrt{1-\gamma_\sigma^2}}h^2 \tag{99}$$

as desired. Proof of (52) follows in the same way since

$$\begin{aligned}\mathbb{E}\left[\mathbb{U}_n\left[K_h(Z_i-z)K_h(Z_{i'}-z)\right]\right]&=\iint K(t_1)K(t_2)f_Z(z+t_1h)f_Z(z+t_2h)dt_1dt_2\\&=f_Z^2(z)+O(h^2).\end{aligned}$$

Similarly, we can also bound the expectation of the cross product term

$$\begin{aligned}&n^{-1}\mathbb{E}\left[u_n\left[g_{z|(j,k)}\right]\times u_n\left[K_h(z_i-z)K_h(z_i-z)\right]\right]\\&=\mathbb{E}[\mathbb{U}_n\left[g_{z|(j,k)}\right]\mathbb{U}_n\left[K_h(z_i-z)K_h(z_{i'}-z)\right]]-\mathbb{E}[\mathbb{U}_n\left[g_{z|(j,k)}\right]]\mathbb{E}[\mathbb{U}_n\left[K_h(z_i-z)K_h(z_{i'}-z)\right]]\\&=\frac{4}{n^2(n-1)^2}\sum_{i\neq j,s\neq t}\mathbb{E}\left[K_h(z_i-z)K_h(z_j-z)g_{z|(j,k)}(y_s,y_t)\right]-(f_Z^2(z)\tau_{jk}(z)+O(h^2))^2\\&=\frac{24}{n^2(n-1)^2}\left\{\binom{n}{4}\mathbb{E}^2\left[K_h(z_i-z)\right]\mathbb{E}\left[\mathbb{U}_n\left[g_{z|(j,k)}\right]\right]\right.\\&\quad+\binom{n}{3}\mathbb{E}\left[K_h(z_i-z)\right]\mathbb{E}\left[\mathbb{U}_n\left[g_{z|(j,k)}(y_i,y_j)K_h(z_i-z)\right]\right]\\&\quad\left.+\binom{n}{3}\mathbb{E}\left[\mathbb{U}_n\left[g_{z|(j,k)}(y_i,y_j)K_h(z_i-z)K_h(z_j-z)\right]\right]\right\}-(f_Z^2(z)+O(h^2))^2=O\left(\frac{1}{nh}\right).\end{aligned}$$

and the expectation of the square term

$$\begin{aligned}&n^{-1}\mathbb{E}\left[\left(u_n\left[K_h(z_i-z)K_h(z_i-z)\right]\right)^2\right]\\&=\mathbb{E}[\mathbb{U}_n^2\left[K_h(z_i-z)K_h(z_{i'}-z)\right]]-\mathbb{E}^2[\mathbb{U}_n\left[K_h(z_i-z)K_h(z_{i'}-z)\right]]\\&=\frac{4}{n^2(n-1)^2}\sum_{i\neq j,s\neq t}\mathbb{E}\left[K_h(z_i-z)K_h(z_j-z)K_h(z_s-z)K_h(z_t-z)\right]-(f_Z^2(z)+O(h^2))^2\\&=\frac{24}{n^2(n-1)^2}\left\{\binom{n}{4}\mathbb{E}^4\left[K_h(z_i-z)\right]+\binom{n}{3}\mathbb{E}^2\left[K_h(z_i-z)\right]\mathbb{E}\left[K_h(z_i-z)\right]^2\right.\\&\qquad\left.+\binom{n}{3}\mathbb{E}\left[K_h(z_i-z)\right]^4\right\}-(f_Z^2(z)+O(h^2))^2=O\left(\frac{1}{nh}\right).\end{aligned}$$

This completes the proof.

Appendix F. Proof of Theorem 4

The proof is similar to the proof of Theorem 3. Instead of only bounding the maximum over $(j,k)\in E^c$, we also need to control the supremum over $z\in[z_L,z_U]$. Without loss of generality, we consider the case $E=\emptyset$ and $[z_L,z_U]=(0,1)$. We will apply the multiplier bootstrap theory for continuous suprema developed in Chernozhukov et al. (2014a). Let W_0 and its bootstrap counterpart W_0^B be defined as

$$W_0=\max_{j,k\in[d]}\sup_{z\in(0,1)}\frac{1}{\sqrt{n}}\sum_{i=1}^n J_{z|(j,k)}(Y_i)\text{ and }W_0^B=\max_{j,k\in[d]}\sup_{z\in(0,1)}\frac{1}{\sqrt{n}}\sum_{i=1}^n J_{z|(j,k)}(Y_i)\cdot\xi_i.$$

Step 1. We aim to approximate W_0 by a Gaussian process. In order to apply Theorem A.1 of Chernozhukov et al. (2014a), we need to study the variance of $J_{z|(j,k)}$ and covering number of the function class $\mathcal{J}=\left\{J_{z|(j,k)}\mid z\in(0,1),j,k\in[d]\right\}$.

By the definition of $J_{z|(j,k)}$ in (48), we have

$$\begin{aligned}\sup_{z,j,k}\mathbb{E}[J_{z|(j,k)}^2(Y)]&\leq\sup_{z,j,k}4\|\mathbf{\Omega}_j(z)\|_1^2\|\mathbf{\Omega}_k(z)\|_1^2\cdot\pi^2h\cdot\left(\mathbb{E}[\max_{u,v}g_{z|(u,v)}^{(1)}(Y_i)]^2+\mathbb{E}[\omega_z^{(1)}(Z_i)]^2\right)\\&\leq CM^2:=\sigma_J^2,\end{aligned}\tag{100}$$

where the last inequality is due to (86) and (95). Similar to (68), we also have for some constant $C>0$,

$$\sup_{z,j,k}\|J_{z|(j,k)}^2(Y)\|_\infty\leq C/h:=b_J.$$

Furthermore, by Lemma 30, the covering number of $\mathcal{J}$ satisfies

$$\sup_Q N(\mathcal{J},L_2(Q),b_J\epsilon)\leq\left(\frac{Cd}{\epsilon h}\right)^c.$$

Therefore $\mathcal{J}$ is a $\mathrm{VC}(b_J,C(d/h)^c,c)$ type class (see, for example, Definition 3.1 Chernozhukov et al., 2014a) and we can apply Theorem A.1 in Chernozhukov et al. (2014a). Let $K_n=C\left(\log n\vee\log(d/h)\right)$ for some sufficiently large $C>0$. Using Theorem A.1 of Chernozhukov et al. (2014a), there exists a random variable W^0 such that for any $\gamma\in(0,1)$,

$$\mathbb{P}\Big(|W_0-W^0|\geq\frac{K_n/\sqrt{h}}{(\gamma n)^{1/2}}+\frac{(\sigma_J/\sqrt{h})^{1/2}K_n^{3/4}}{\gamma^{1/2}n^{1/4}}+\frac{h^{-1/6}\sigma_J^{2/3}K_n^{2/3}}{\gamma^{1/3}n^{1/6}}\Big)\leq C\Big(\gamma+\frac{\log n}{n}\Big).$$

Choosing $\gamma=(\log^4(dn)/(nh))^{1/8}$, we have

$$\mathbb{P}\big(|W_0-W^0|>C(\log^4(dn)/(nh))^{1/8}\big)\leq C(\log^4(dn)/(nh))^{1/8}.\tag{101}$$

Step 2. We next bound the difference between W_0^B and W^0. Define

$$\psi_n=\sqrt{\frac{\sigma_J^2K_n}{n}}+\left(\frac{\sigma_J^2K_n^3}{nh}\right)^{1/4}\qquad\text{and}\qquad\gamma_n(\delta)=\frac{1}{\delta}\left(\frac{\sigma_J^2K_n^3}{nh}\right)^{1/4}+\frac{1}{n}.$$

Since $K_n/h\lesssim(\log(dn))/h\lesssim n\sigma_J^2$, Theorem A.2 of Chernozhukov et al. (2014a) gives us

$$\mathbb{P}\Big(|W_0^B-W^0|>\psi_n+\delta\,\Big|\,\{Y_i\}_{i\in[n]}\Big)\leq C\gamma_n(\delta),$$

with probability $1-3/n$. Choosing $\delta = (\log(dn))^3/(nh))^{1/8}$, with probability $1-3/n$,

$$\mathbb{P}\Big(|W_0^B - W^0| > C((\log(dn))^3/(nh))^{1/8} \mid \{Y_i\}_{i\in[n]}\Big) \leq C((\log d)^3/(nh))^{1/8}. \tag{102}$$

Step 3. The last step is to assemble the above results to quantify the difference between W and W^B. According to Lemma 19 and Lemma 20, there exists a constant $c>0$ such that

$$\mathbb{P}(|W - W_0| > n^{-c}) \leq n^{-c} \qquad \text{and} \qquad \mathbb{P}(\mathbb{P}_\xi\big(|W_G^B - W_0^B| > n^{-c}\big) > n^{-c}\big) \leq n^{-c}. \tag{103}$$

Let $q_n := [(\log(dn))^3/(nh)]^{1/8}$. From (103) and (101), we have

$$\mathbb{P}\big(|W - W^0| > q_n\big) \leq q_n. \tag{104}$$

Define the event

$$\mathcal{W} = \Big\{\mathbb{P}\big(|W_G^B - W^0| > q_n \,\Big|\, \{Y_i\}_{i\in[n]}\big) \leq q_n\Big\}. \tag{105}$$

Using (103) and (102), $\mathbb{P}(\mathcal{W}) \geq 1-n^{-c}$. Recall that $\widehat{c}_W(1-\alpha, \{Y_i\}_{i=1}^n)$ is $(1-\alpha)$-quantile of W_G^B conditionally on $\{Y_i\}_{i=1}^n$. Let

$$W_0^B(\{Y_i\}_{i=1}^n) = \max_{j,k\in[d]} \sup_{z\in(0,1)} n^{-1/2}\sum_{i=1}^n J_{z|(j,k)}(Y_i)\cdot\xi_i$$

and define $W^B(\{Y_i\}_{i=1}^n)$ similarly. Let $\widehat{\sigma}^2_{z,j,k} = n^{-1}\sum_{i=1}^n J^2_{z|(j,k)}(Y_i)$, $\underline{\sigma}_J = \inf_{z,j,k}\widehat{\sigma}_{z,j,k}$ and $\overline{\sigma}_J = \sup_{z,j,k}\widehat{\sigma}_{z,j,k}$. Using the triangle inequality, we have

$$\begin{aligned}
&\mathbb{P}\big(W \leq \widehat{c}_W(1-\alpha, \{Y_i\}_{i=1}^n)\big) \overset{(104)}{\geq} \mathbb{P}\big(W^0 \leq \widehat{c}_W(1-\alpha, \{Y_i\}_{i=1}^n) - q_n\big) - q_n \\
&\overset{(105)}{\geq} \mathbb{P}\big(W_0^B(\{Y_i\}_{i=1}^n) \leq \widehat{c}_W(1-\alpha, \{Y_i\}_{i=1}^n) - 2q_n\big) - 2q_n \\
&\geq \mathbb{P}\big(W^B(\{Y_i\}_{i=1}^n) \leq \widehat{c}_W(1-\alpha, \{Y_i\}_{i=1}^n)\big) \\
&\qquad - C(\underline{\sigma}_J, \overline{\sigma}_J)q_n(\mathbb{E}_\xi[W_0^B] + \sqrt{1\vee\log(n\underline{\sigma}_J)}) - Cq_n,
\end{aligned} \tag{106}$$

where the last inequality follows from the anti-concentration for suprema of Gaussian processes, given in Lemma A.1 of Chernozhukov et al. (2014b), and $C(\underline{\sigma}_J, \overline{\sigma}_J)$ is a constant that only depends on $\underline{\sigma}_J$ and $\overline{\sigma}_J$.

In the following of the proof, we will bound the right hand side of (106). We first show the constant $C(\underline{\sigma}_J, \overline{\sigma}_J)$ is independent to h, n or d and then bound $\mathbb{E}_\xi[W^B]$. We bound $\mathbb{E}_\xi[W^B]$ by bounding $\mathbb{E}_\xi[W_0^B]$ and $\mathbb{E}_\xi[|W^B - W_0^B|]$. Since W_0^B is a supreme of a Gaussian process given data, we can bound its expectation by quantifying its conditional variance and related covering number. We first bound its conditional variance $n^{-1}\sum_{i=1}^n J^2_{z|(j,k)}(Y_i)$. Using the notation defined in (67), we have

$$\sup_{z,j,k}\frac{1}{n}\sum_{i=1}^n (J^2_{z|(j,k)}(Y_i) - \mathbb{E}[J^2_{z|(j,k)}(Y_i)]) = \sup_{z,j,k}\frac{1}{n}\sum_{i=1}^n \gamma_{z|(j,k,j,k)}(Y_i).$$

Therefore, (69) and (70) give an upper bound and variance of $(J^2_{z|(j,k)}(Y_i) - \mathbb{E}[J^2_{z|(j,k)}(Y_i)])$. Define the function class $\mathcal{J}_{(2)} = \Big\{J^2_{z|(j,k)} \mid z\in(0,1), j,k\in[d]\Big\}$. Using Lemma 26 and Lemma 30,

$$\sup_Q N(\mathcal{J}_{(2)}, \|\cdot\|_{L_2(Q)}, \epsilon/\sqrt{h}) \leq \left(\frac{Cd}{h\epsilon}\right)^c.$$

As the upper bound, variance and covering number are quantified above, similar to (88), we apply Theorem 3.12 of Koltchinskii (2011) to get

$$\mathbb{E}\left[\sup_{z,j,k}\frac{1}{n}\sum_{i=1}^n (J^2_{z|(j,k)}(Y_i) - \mathbb{E}[J^2_{z|(j,k)}(Y_i)])\right] \lesssim \sqrt{\frac{\log(2d/h)}{nh^2}} + \frac{\log(2d/h)}{nh}.$$

Under the assumptions of the theorem, $\sqrt{\log(2d/h)/(nh^2)} + \log(2d/h)/(nh) = O(n^{-2c})$ and the Markov's inequality give

$$\mathbb{P}\left(\sup_{z,j,k}\frac{1}{n}\sum_{i=1}^n (J^2_{z|(j,k)}(Y_i) - \mathbb{E}[J^2_{z|(j,k)}(Y_i)]) > n^{-c}\right) \leq Cn^{-c}.$$

Combining with (100),

$$\overline{\sigma}_J^2 = \sup_{z,j,k}\frac{1}{n}\sum_{i=1}^n J^2_{z|(j,k)}(Y_i) \leq \sigma_J^2 + n^{-c} \leq 2\sigma_J^2,$$

with probability $1-Cn^{-c}$. By the assumption in Theorem 4,

$$\inf_{z,j,k}\mathbb{E}[J^2_{z|(j,k)}(Y)] = \inf_{z,j,k}\mathrm{Var}(\mathbf{\Omega}_j(z)\mathbf{\Theta}_z\mathbf{\Omega}_k(z)) > c > 0.$$

Therefore, we have

$$\begin{aligned}
\underline{\sigma}_J &= \inf_{z,j,k}\frac{1}{n}\sum_{i=1}^n J^2_{z|(j,k)}(Y_i) \\
&\geq \inf_{z,j,k}\mathbb{E}[J^2_{z|(j,k)}(Y)] - \sup_{z,j,k}\frac{1}{n}\sum_{i=1}^n (J^2_{z|(j,k)}(Y_i) - \mathbb{E}[J^2_{z|(j,k)}(Y_i)]) \geq c/2 > 0
\end{aligned}$$

with probability $1-Cn^{-c}$. The constant $C(\underline{\sigma}_J, \overline{\sigma}_J)$ does not depend on n, d and h.

Combining Lemma 2.2.8 in van der Vaart and Wellner (1996) and Lemma 30, we have

$$\mathbb{E}_\xi[W_0^B] \leq C\sigma_J\sqrt{\log\left(Cd\sigma_J^{-1}/h\right)} \leq C\sqrt{\log(d/h)}.$$

From Lemma 20, we have $\mathbb{E}_\xi[W^B] \leq \mathbb{E}_\xi[W_0^B] + \mathbb{E}_\xi[|W^B - W_0^B|] \leq C\sqrt{\log(d/h)}$, since $q_n\sqrt{\log(d/h)} = (\log^8(d/h)/(nh))^{1/8} = O(n^{-c})$. Due to (106) and the fact that $\mathbb{P}(\mathcal{W}) \geq 1-n^{-c}$, we have $\mathbb{P}\big(W \leq c_W(1-\alpha)\big) \geq 1-\alpha-3n^{-c}$. Similarly, we can also show that $\mathbb{P}\big(W \geq \widehat{c}_W(1-\alpha, \{Y_i\}_{i=1}^n)\big) \geq \alpha - 3n^{-c}$, which completes the proof.

Appendix G. Auxiliary Lemmas for Score Statistics

In this section, we provide the auxiliary results for proving auxiliary lemmas on the asymptotic properties on the score statistics.

G.1 Approximation Error for Score Statistics

In this section, we prove Lemma 19 and Lemma 20, which approximate the score statistics by a leading linear term.

Lemma 19 *Under the same conditions as Theorem 4, there exists a universal constant $c > 0$ such that*

$$\sup_{j,k\in[d]}\sup_{z\in(0,1)}\left|\sqrt{nh}\cdot\mathbb{U}_n[\omega_z]\widehat{S}_{z|(j,k)}\big(\widehat{\mathbf{\Omega}}_{k\backslash j}(z)\big)-\mathbb{G}_n\big[J_{z|(j,k)}\big]\right|\leq n^{-c}, \tag{107}$$

with probability $1-c/d$.

Proof We have

$$\begin{aligned}
&\left|\sqrt{nh}\cdot\mathbb{U}_n[\omega_z]\widehat{S}_{z|(j,k)}\left(\widehat{\mathbf{\Omega}}_{k\backslash j}(z)\right)-\mathbb{G}_n\left[J_{z|(j,k)}\right]\right| \\
&\quad\leq\underbrace{\sqrt{nh}\cdot\mathbb{U}_n[\omega_z]\left|\widehat{S}_{z|(j,k)}\left(\widehat{\mathbf{\Omega}}_{k\backslash j}(z)\right)-\mathbf{\Omega}_j^T(z)\left(\widehat{\mathbf{\Sigma}}(z)\mathbf{\Omega}_k(z)-\mathbf{e}_k^T\right)\right|}_{I} \\
&\qquad+\underbrace{\left|\sqrt{nh}\cdot\mathbb{U}_n[\omega_z]\mathbf{\Omega}_j^T(z)\left(\widehat{\mathbf{\Sigma}}(z)\mathbf{\Omega}_k(z)-\mathbf{e}_k^T\right)-\mathbb{G}_n\left[J_{z|(j,k)}\right]\right|}_{II}
\end{aligned}$$

A bound on I is obtained in Lemma 21. Here, we proceed to obtain a bound on II.

To simplify the notation, we sometimes omit the argument z_0, that is, we write $\mathbf{\Omega}(z_0)$ as $\mathbf{\Omega}$ and similarly for other parameters indexed by z. Applying the Taylor expansion to $\sin(\cdot)$ we obtain

$$\begin{aligned}
&\sqrt{nh}\cdot\mathbf{\Omega}_j^T(z)\left(\widehat{\mathbf{\Sigma}}(z)\mathbf{\Omega}_k(z)-\mathbf{e}_k\right)=\sqrt{nh}\cdot\mathbf{\Omega}_j^T(z)\left(\widehat{\mathbf{\Sigma}}(z)-\mathbf{\Sigma}(z)\right)\mathbf{\Omega}_k(z) \\
&=\sqrt{nh}\sum_{a,b\in[d]}\mathbf{\Omega}_{ja}\mathbf{\Omega}_{bk}\left(\sin\left(\widehat{\tau}_{ab}\frac{\pi}{2}\right)-\sin\left(\tau_{ab}\frac{\pi}{2}\right)\right) \\
&=\underbrace{\sqrt{nh}\sum_{a,b\in[d]}\mathbf{\Omega}_{ja}\mathbf{\Omega}_{bk}\cos\left(\tau_{ab}\frac{\pi}{2}\right)\frac{\pi}{2}\left(\widehat{\tau}_{ab}-\tau_{ab}\right)}_{T_1} \\
&\qquad\underbrace{-\frac{\sqrt{nh}}{2}\sum_{a,b\in[d]}\mathbf{\Omega}_{ja}\mathbf{\Omega}_{bk}\sin\left(\widetilde{\tau}_{ab}\frac{\pi}{2}\right)\left(\frac{\pi}{2}\left(\widehat{\tau}_{ab}-\tau_{ab}\right)\right)^2}_{T_2},
\end{aligned} \tag{108}$$

where $\widetilde{\tau}_{ij}=(1-\alpha)\widehat{\tau}_{ij}+\alpha\tau_{ij}$ for some $\alpha\in(0,1)$.

In order to study the properties of T_1 and T_2, we first analyze the $\sqrt{nh}(\widehat{\tau}_{ab}-\tau_{ab})$ shared by both terms. Let

$$r_{z|(a,b)}(Y_i,Y_{i'}):=g_{z|(a,b)}(Y_i,Y_{i'})-\tau_{ab}(z)\omega_z(Z_i,Z_{i'}), \tag{109}$$

where $g_{z|(a,b)}$ is defined in (40). From (4), we have

$$\sqrt{nh}\big(\widehat{\tau}_{ab}(z)-\tau_{ab}(z)\big)=\frac{\sqrt{nh}\cdot\sum_{i\neq i'}r_{z|(a,b)}(Y_i,Y_{i'})}{\sum_{i\neq i'}\omega_z(Z_i,Z_{i'})}.$$

We divide the U-statistic in the numerator into two parts

$$\frac{1}{\sqrt{nh}(n-1)}\sum_{i\neq i'}r_{z|(a,b)}(Y_i,Y_{i'})=\sqrt{h}\cdot u_n[r_{z|(a,b)}]+\sqrt{nh}\cdot\mathbb{E}[r_{z|(a,b)}(Y_i,Y_{i'})].$$

For any $z\in(0,1)$ and $a,b\in[d]$, using Lemma 13 we obtain

$$\begin{aligned}
&\sqrt{nh}\cdot\mathbb{E}[r_{z|(a,b)}(Y_i,Y_{i'})] \\
&\quad=\sqrt{nh}\cdot\mathbb{E}\left[\mathbb{U}_n\left[g_{z|(a,b)}\right]\right]-\sqrt{nh}\cdot\tau_{ab}(z)\mathbb{E}\left[\mathbb{U}_n\left[K_h(Z_i-z)K_h(Z_{i'}-z)\right]\right] \\
&\quad=\sqrt{nh}\big[f_Z^2(z)\tau_{ab}(z)+O(h^2)\big]-\sqrt{nh}\tau_{ab}(z)\big[f_Z^2(z)+O(h^2)\big] \\
&\quad=O(\sqrt{nh^5}).
\end{aligned} \tag{110}$$

For the leading term $\sqrt{h}\cdot u_n[r_{z|(a,b)}]$, we combine (109) with (43) and (46), to obtain

$$\sqrt{h}\cdot u_n[r_{z|(a,b)}]=2\sqrt{nh}\cdot\mathbb{E}_n\left[g^{(1)}_{z|(a,b)}-\tau_{ab}(z)\omega_z^{(1)}\right]+\sqrt{nh}\cdot\mathbb{U}_n\left[g^{(2)}_{z|(a,b)}-\tau_{ab}(z)\omega_z^{(2)}\right],$$

where $g^{(1)}_{z|(a,b)}$, $g^{(2)}_{z|(a,b)}$, $\omega_z^{(1)}$, $\omega_z^{(2)}$ are defined in (41), (42), (44), (45) respectively. With this,

$$\begin{aligned}
T_1&=[\mathbb{U}_n[\omega_z]]^{-1}\sum_{a,b\in[d]}\mathbf{\Omega}_{ja}\mathbf{\Omega}_{bk}\frac{\pi}{2}\cos\left(\tau_{ab}\frac{\pi}{2}\right)\left(\sqrt{h}\cdot u_n[r_{z|(a,b)}]+\sqrt{nh}\cdot\mathbb{E}[r_{z|(a,b)}(Y_i,Y_{i'})]\right) \\
&=[\mathbb{U}_n[\omega_z]]^{-1}(T_{11}+T_{12}+T_{13}),
\end{aligned}$$

where

$$\begin{aligned}
T_{11}&=\sqrt{nh}\sum_{a,b\in[d]}\mathbf{\Omega}_{ja}\mathbf{\Omega}_{bk}\pi\cos\left(\tau_{ab}\frac{\pi}{2}\right)\mathbb{E}_n\left[g^{(1)}_{z|(a,b)}-\tau_{ab}(z)\omega_z^{(1)}\right], \\
T_{12}&=\sqrt{nh}\sum_{a,b\in[d]}\mathbf{\Omega}_{ja}\mathbf{\Omega}_{bk}\frac{\pi}{2}\cos\left(\tau_{ab}\frac{\pi}{2}\right)\mathbb{U}_n\left[g^{(2)}_{z|(a,b)}-\tau_{ab}(z)\omega_z^{(2)}\right] \quad\text{and} \\
T_{13}&=\sqrt{nh}\sum_{a,b\in[d]}\mathbf{\Omega}_{ja}\mathbf{\Omega}_{bk}\frac{\pi}{2}\cos\left(\tau_{ab}\frac{\pi}{2}\right)\mathbb{E}[r_{z|(a,b)}(Y_i,Y_{i'})].
\end{aligned}$$

From (48), we have that $T_{11}=\mathbb{G}_n[J_{z|(j,k)}]$. We proceed to bound the other terms. Using Lemma 16 and Lemma 18, we have

$$\sup_{z,j,k}|T_{12}|\leq\sup_{z,j,k}\sqrt{nh}\cdot\|\mathbf{\Omega}_j\|_1\|\mathbf{\Omega}_k\|_1\max_{a,b\in[d]}\left(\left|\mathbb{U}_n\left[g^{(2)}_{z|(a,b)}\right]\right|+\left|\mathbb{U}_n\left[\omega_z^{(2)}\right]\right|\right)\lesssim\frac{\log(d/h)}{\sqrt{nh}}, \tag{111}$$

with probability $1-1/d$. Using (110), we can bound T_{13} as

$$\sup_{z,j,k}|T_{13}|\lesssim\frac{\pi}{2}\|\mathbf{\Omega}_j\|_1\|\mathbf{\Omega}_k\|_1\cdot(\sqrt{nh^5})\leq\pi M^2\sqrt{nh^5}. \tag{112}$$

The final step is to bound T_2. Using Lemma 9 and Lemma 10,

$$\sup_{z,j,k}|T_2| \le \frac{\pi^2}{8}\|\mathbf{\Omega}_j\|_1\|\mathbf{\Omega}_k\|_1 \max_{z,a,b}\sqrt{nh}\cdot|\widehat{\tau}_{ab}(z)-\tau_{ab}(z)|^2$$
$$\le CM^2\sqrt{nh}\cdot\left(\frac{\log(d/h)}{nh}+h^4+\frac{1}{n^2h^2}\right) \quad (113)$$

with probability $1-1/d$.

Combining (55), (111), (112), and (113) with (108), we finally have

$$\sup_{j,k\in[d]}\sup_{z\in(0,1)}\left|\sqrt{nh}\cdot\mathbb{U}_n[\omega_z]\widehat{S}_{z|(j,k)}\big(\widehat{\mathbf{\Omega}}_{k\backslash j}(z)\big)-\mathbb{G}_n\big(J_{z|(j,k)}\big)\right|$$
$$\le \sup_{j,k\in[d]}\sup_{z\in(0,1)}\big(|T_{12}|+|T_{13}|+|\mathbb{U}_n[\omega_z]||T_2|\big)\lesssim\frac{\log(d/h)}{\sqrt{nh}}+\sqrt{nh^5}.$$

with probability $1-1/d$. Under the assumptions of the lemma, there exists a constant $c>0$ such that $\log(d/h)/\sqrt{nh}=o(n^{-c})$ and $\sqrt{nh^5}=o(n^{-c})$, which completes the proof. ■

The following lemma states a result analogous to Lemma 19 for the bootstrap test statistic.

Lemma 20 *Under the same conditions as Theorem 4, there exists a universal constant $c>0$ such that for sufficiently large n,*

$$\mathbb{P}_\xi\left(\sup_{j,k\in[d]}\sup_{z\in(0,1)}\left|\sqrt{nh}\cdot\mathbb{U}_n[\omega_z^B]\widehat{S}^B_{z|(j,k)}\big(\widehat{\mathbf{\Omega}}_{k\backslash j}(z)\big)-\mathbb{G}_n^\xi\big[J_{z|(j,k)}\big]\right|\le n^{-c}\right)\ge 1-c/d.$$

Proof We have

$$\left|\sqrt{nh}\cdot\mathbb{U}_n[\omega_z^B]\widehat{S}^B_{z|(j,k)}\left(\widehat{\mathbf{\Omega}}_{k\backslash j}(z)\right)-\mathbb{G}_n^\xi\left(J_{z|(j,k)}\right)\right|$$
$$\le\underbrace{\sqrt{nh}\cdot\mathbb{U}_n[\omega_z^B]\left|\widehat{S}^B_{z|(j,k)}\left(\widehat{\mathbf{\Omega}}^B_{k\backslash j}(z)\right)-\mathbf{\Omega}_j^T(z)\left(\widehat{\mathbf{\Sigma}}(z)\mathbf{\Omega}_k(z)-\mathbf{e}_k^T\right)\right|}_{I}$$
$$+\underbrace{\left|\sqrt{nh}\cdot\mathbb{U}_n[\omega_z^B]\mathbf{\Omega}_j^T(z)\left(\widehat{\mathbf{\Sigma}}^B(z)\mathbf{\Omega}_k(z)-\mathbf{e}_k^T\right)-\mathbb{G}_n^\xi\left[J_{z|(j,k)}\right]\right|}_{II}.$$

Lemma 22 gives a bound on I. Here, we focus on obtaining a bound on II. That is, we show that $\sqrt{nh}\cdot\mathbb{U}_n[\omega_z^B]\mathbf{\Omega}_j^T(z)\big(\widehat{\mathbf{\Sigma}}^B(z)\mathbf{\Omega}_k(z)-\mathbf{e}_k^T\big)$ is close to the linear leading term

$$T_0^B:=\mathbb{G}_n^\xi\left[J_{z|(j,k)}\right]=\frac{1}{\sqrt{n}}\sum_{i=1}^n J_{z|(j,k)}(Y_i)\cdot\xi_i.$$

Similar to (108), we have

$$\max_{j,k\in[d]}\sup_{z\in(0,1)}\sqrt{nh}\cdot\mathbb{U}_n[\omega_z^B]\mathbf{\Omega}_j(z_0)^T\Big(\widehat{\mathbf{\Sigma}}^B(z)\mathbf{\Omega}_k(z)-\mathbf{e}_k^T\Big)=T_1^B+T_2^B,$$

where

$$T_1^B=\sqrt{nh}\sum_{a,b\in[d]}\mathbf{\Omega}_{ja}\mathbf{\Omega}_{bk}\cos\left(\tau_{ab}\frac{\pi}{2}\right)\frac{\pi}{2}\mathbb{U}_n[\omega_z^B]\left(\widehat{\tau}_{ab}^B-\tau_{ab}\right)$$

and

$$T_2^B=-\frac{\sqrt{nh}}{2}\sum_{a,b\in[d]}\mathbf{\Omega}_{ja}\mathbf{\Omega}_{bk}\sin\left(\tau_{ab}\frac{\pi}{2}\right)\mathbb{U}_n[\omega_z^B]\left(\frac{\pi}{2}\left(\widehat{\tau}_{ab}^B-\tau_{ab}\right)\right)^2.$$

We bound T_2^B first. Note that

$$\underline{\mathrm{f}}_Z^2-\sup_{z\in(0,1)}|\mathbb{U}_n[\omega_z^B]-f_Z^2(z)|\le\inf_{z\in(0,1)}\mathbb{U}_n[\omega_z^B]\le\sup_{z\in(0,1)}\mathbb{U}_n[\omega_z^B]\le\bar{\mathrm{f}}_Z^2+\sup_{z\in(0,1)}|\mathbb{U}_n[\omega_z^B]-f_Z^2(z)|.$$

Using Lemma 24 and $\log(h^{-1})/(nh^2)=o(1)$, we have

$$\underline{\mathrm{f}}_Z^2/2\le\inf_{z\in(0,1)}\mathbb{U}_n[\omega_z]\le\sup_{z\in(0,1)}\mathbb{U}_n[\omega_z]\le 2\bar{\mathrm{f}}_Z^2, \quad (114)$$

with probability at least $1-1/n$. Similar to (113), (114) and the Hölder's inequality give us

$$|T_2^B|\le\frac{\pi^2}{4\underline{\mathrm{f}}_Z^2}\|\mathbf{\Omega}_j\|_1\|\mathbf{\Omega}_k\|_1\max_{a,b\in[d]}\sqrt{nh}\cdot|\mathbb{U}_n[\omega_z^B](\widehat{\tau}_{ab}^B-\tau_{ab})|^2.$$

Under the assumptions, $h\asymp n^{-\delta}$ for $\delta\in(1/5,1/4)$, and Lemma 23 gives us

$$\mathbb{P}_\xi\Big(\max_{z,j,k}|T_2^B|\le C\log(d/h)/\sqrt{nh^3}\Big)\ge 1-1/d \quad (115)$$

for some constant $C>0$ with probability $1-c/d$.

Next, we handle the difference between T_1^B and T_0^B. We denote

$$\Delta W_{z|(a,b)}=\mathbb{U}_n[\omega_z^B]\left(\widehat{\tau}_{ab}^B(z)-\tau_{ab}(z)\right)-\frac{2}{n}\sum_{i=1}^n\left[g^{(1)}_{z|(a,b)}(Y_i)-\tau_{ab}(z)\omega_z^{(1)}(Z_i)\right]\xi_i. \quad (116)$$

Using the Hölder's inequality, we have

$$|T_1^B-T_0^B|\le\left|\sqrt{nh}\sum_{a,b\in[d]}\mathbf{\Omega}_{ja}\mathbf{\Omega}_{bk}\cos\left(\tau_{ab}\frac{\pi}{2}\right)\frac{\pi}{2}\cdot\Delta W_{z|(a,b)}\right|\le CM^2\sqrt{nh}\max_{a,b}|\Delta W_{z|(a,b)}|.$$

Combining with (123), there exists a constant $c>0$ such that with probability $1-c/d$

$$\mathbb{P}_\xi\big(\sup_{z\in(0,1)}\max_{j,k\in[d]}|T_1^B-T_0^B|\ge C\sqrt{\log(d/h)/(nh^2)}\big)\le 1/d. \quad (117)$$

If $\log(d/h)/(nh^2)\asymp n^{-c}$, (120), (115) and (117) give us with probability $1-c/d$

$$\mathbb{P}_\xi\Big(\sup_{z\in(0,1)}\max_{j,k\in[d]}|\widehat{S}^B_{z|(j,k)}\big(\widehat{\mathbf{\Omega}}_{k\backslash j}(z)\big)-T_0^B|\le n^{-c}\Big)\ge 1-1/d,$$

which completes the proof of the lemma. ■

G.2 First Step Approximation Results

Here we establish results needed for the first step in the proofs of Lemma 19 and Lemma 20.

Lemma 21 *Under the same conditions as Theorem 4, there exists a universal constant $c > 0$ such that*

$$\sup_{j,k\in[d]}\sup_{z\in(0,1)}\sqrt{nh}\cdot\Big|\widehat{S}_{z|(j,k)}\Big(\widehat{\mathbf{\Omega}}_{k\backslash j}(z)\Big)-\mathbf{\Omega}_j^T(z)\Big(\widehat{\mathbf{\Sigma}}(z)\mathbf{\Omega}_k(z)-\mathbf{e}_k^T\Big)\Big|\le n^{-c}$$

with probability $1-c/d$.

Proof For any matrix $\mathbf{A}=(\mathbf{A}_1,\ldots,\mathbf{A}_d)\in\mathbb{R}^{d\times d}$, we define

$$\mathbf{A}_{-j}:=(\mathbf{A}_1,\ldots,\mathbf{A}_{j-1},\mathbf{A}_{j+1},\ldots,\mathbf{A}_d)\in\mathbb{R}^{d\times(d-1)},$$

which is a submatrix of $\mathbf{A}$ with column j removed. We also denote

$$\begin{aligned}\boldsymbol{\gamma}^*(z)&=(\mathbf{\Omega}_{k1}(z),\ldots,\mathbf{\Omega}_{k(j-1)}(z),\mathbf{\Omega}_{k(j+1)}(z)\ldots\mathbf{\Omega}_{kd}(z))^T\in\mathbb{R}^{d-1}\quad\text{and}\\ \widehat{\boldsymbol{\gamma}}(z)&=(\widehat{\mathbf{\Omega}}_{k1}(z),\ldots,\widehat{\mathbf{\Omega}}_{k(j-1)}(z),\widehat{\mathbf{\Omega}}_{k(j+1)}(z),\ldots,\widehat{\mathbf{\Omega}}_{kd}(z))^T\in\mathbb{R}^{d-1}.\end{aligned}$$

To simplify the notation, we sometimes omit the varying variable z in the proof.

We start by decomposing the score function into two parts. We will then identify the leading term and bound the remainder. With the above introduced notation, we have

$$\begin{aligned}&\sqrt{nh}\cdot\widehat{S}_{z|(j,k)}\big(\widehat{\mathbf{\Omega}}_{k\backslash j}\big)\\&\quad=\sqrt{nh}\cdot\widehat{\mathbf{\Omega}}_j^T\big(\widehat{\mathbf{\Sigma}}_{-j}\widehat{\boldsymbol{\gamma}}-\mathbf{e}_k^T\big)\\&\quad=\sqrt{nh}\cdot\mathbf{\Omega}_j^T\big(\widehat{\mathbf{\Sigma}}_{-j}\boldsymbol{\gamma}^*-\mathbf{e}_k^T\big)+\underbrace{\sqrt{nh}\cdot\big(\widehat{\mathbf{\Omega}}_j^T-\mathbf{\Omega}_j^T\big)\big(\widehat{\mathbf{\Sigma}}_{-j}\boldsymbol{\gamma}^*-\mathbf{e}_k^T\big)}_{I_1}+\underbrace{\sqrt{nh}\cdot\widehat{\mathbf{\Omega}}_j^T\widehat{\mathbf{\Sigma}}_{-j}\big(\widehat{\boldsymbol{\gamma}}-\boldsymbol{\gamma}^*\big)}_{I_2}.\end{aligned}$$

The first term in the display above is the leading term on the right hand side of (107) as desired. The remaining part of the proof is to bound I_1 and I_2. By the Hölder's inequality, we have

$$|I_1|=\sqrt{nh}\cdot\big(\widehat{\mathbf{\Omega}}_j-\mathbf{\Omega}_j\big)^T\big(\widehat{\mathbf{\Sigma}}_{-j}-\mathbf{\Sigma}_{-j}\big)\boldsymbol{\gamma}^*\le\sqrt{nh}\cdot\big\|\widehat{\mathbf{\Omega}}_j-\mathbf{\Omega}_j\big\|_1\big\|\widehat{\mathbf{\Sigma}}-\mathbf{\Sigma}\big\|_{\max}\|\mathbf{\Omega}_k\|_1.$$

Assumption 4.4 together with the display above gives

$$\sup_{z\in(0,1)}\max_{j,k\in[d]}|I_1|\le M\sqrt{nh}\cdot r_{1n}r_{2n}\tag{118}$$

with probability $1-1/d$. Next, using the Hölder's inequality we have

$$|I_2|\le\sqrt{nh}\cdot\|\widehat{\mathbf{\Omega}}_j^T\widehat{\mathbf{\Sigma}}_{-j}\|_\infty\big\|\widehat{\boldsymbol{\gamma}}-\boldsymbol{\gamma}^*\big\|_1.$$

From Assumption 4.4, we have $\|\widehat{\mathbf{\Omega}}_j^T\widehat{\mathbf{\Sigma}}_{-j}\|_\infty\le\|\widehat{\mathbf{\Omega}}_j^T\widehat{\mathbf{\Sigma}}\|_\infty\le r_{3n}$ with probability $1-1/d$ and, therefore,

$$\sup_{z\in(0,1)}\max_{j,k\in[d]}|I_2|\le\sqrt{nh}\cdot r_{3n}r_{2n}\tag{119}$$

with probability $1-1/d$. Combining (118) and (119), we have

$$\sup_{j,k\in[d]}\sup_{z\in(0,1)}\big(|I_1|+|I_2|\big)\le\sqrt{nh}\cdot r_{2n}(r_{1n}+r_{3n})\le n^{-c}$$

with probability $1-2/d$, which completes the proof. ■

Lemma 22 *Under the same conditions as Theorem 4, there exists a universal constant $c>0$ such that with probability $1-1/d$,*

$$\mathbb{P}_\xi\Big(\max_{j,k\in[d]}\sup_{z\in(0,1)}\sqrt{nh}\cdot\Big|\widehat{S}^B_{z|(j,k)}\big(\widehat{\mathbf{\Omega}}_{k\backslash j}(z)\big)-\mathbf{\Omega}_j^T(z)\Big(\widehat{\mathbf{\Sigma}}^B(z)\mathbf{\Omega}_k(z)-\mathbf{e}_k^T\Big)\Big|\le n^{-c}\Big)\ge1-1/d.\tag{120}$$

Proof The proof is similar to the proof of Lemma 21. Compared to the proof in Lemma 21, there are two differences: (1) we need to bound $\mathbb{U}_n[\omega_z^B]\|\widehat{\mathbf{\Sigma}}^B-\mathbf{\Sigma}\|_{\max}$ instead of $\|\widehat{\mathbf{\Sigma}}-\mathbf{\Sigma}\|_{\max}$ and (2) obtain a rate for $\mathbb{U}_n[\omega_z^B]\|\widehat{\mathbf{\Omega}}_j^T\widehat{\mathbf{\Sigma}}^B\|_\infty$ instead of $\|\widehat{\mathbf{\Omega}}_j^T\widehat{\mathbf{\Sigma}}\|_\infty$.

According to Lemma 23 and (114), we have

$$\begin{aligned}&\mathbb{P}_\xi\Big(\sup_{z\in(0,1)}\mathbb{U}_n[\omega_z^B]\|\widehat{\mathbf{\Sigma}}^B-\mathbf{\Sigma}\|_{\max}>C\sqrt{\log(d/h)/(nh^2)}\Big)\\&\le\mathbb{P}_\xi\Big(\max_{j,k\in[d]}\sup_{z\in(0,1)}\Big|\mathbb{U}_n[\omega_z^B]\big(\widehat{\tau}^B_{jk}(z)-\tau_{jk}(z)\big)\Big|>C\sqrt{\log(d/h)/(nh^2)}\Big)\le1/d.\end{aligned}\tag{121}$$

Next, (114) and the Hölder's inequality, give us

$$\mathbb{U}_n[\omega_z^B]\|\widehat{\mathbf{\Omega}}_j^T\widehat{\mathbf{\Sigma}}^B\|_\infty\le2\bar{\mathrm{f}}_Z\Big(\|\widehat{\mathbf{\Omega}}_j^T\widehat{\mathbf{\Sigma}}\|_\infty+\big(\|\widehat{\mathbf{\Omega}}_j-\mathbf{\Omega}_j\|_1+\|\mathbf{\Omega}_j\|_1\big)\mathbb{U}_n[\omega_z^B]\|\widehat{\mathbf{\Sigma}}^B-\widehat{\mathbf{\Sigma}}\|_{\max}\Big).$$

Therefore, by Assumption 4.4, (121) and Theorem 6, with probability $1-1/d$,

$$\mathbb{P}_\xi\Big(\mathbb{U}_n[\omega_z^B]\|\widehat{\mathbf{\Omega}}_j\widehat{\mathbf{\Sigma}}^B\|_\infty>C\big(M\lambda+M\sqrt{\log(d/h)/(nh^2)}\big)\Big)\le1/d.\tag{122}$$

Compared to the rate on I_1 in (118), we have

$$I_{12}^B:=\sqrt{nh}\cdot\big(\widehat{\mathbf{\Omega}}_j-\mathbf{\Omega}_j\big)^T\big(\widehat{\mathbf{\Sigma}}^B_{-j}-\mathbf{\Sigma}_{-j}\big)\boldsymbol{\gamma}^*\le\sqrt{nh}\cdot\big\|\widehat{\mathbf{\Omega}}_j-\mathbf{\Omega}_j\big\|_1\big\|\widehat{\mathbf{\Sigma}}^B-\mathbf{\Sigma}\big\|_{\max}\|\boldsymbol{\gamma}^*\|_1.$$

For $\lambda=\kappa\sqrt{\log(dn)}\cdot(h^2+1/\sqrt{nh})$ and $h=n^{-\delta}$, for $1/5<\delta<1/4$, by (121), we have with probability $1-1/d$, there exists a constant c such that

$$\mathbb{P}_\xi\Big(\sup_{z,j,k}|I_{12}^B|>sM^2\lambda\sqrt{\log(d/h)/h}\Big)\le1/d.$$

Instead of I_2 in (119), we define $I_2^B:=\sqrt{nh}\cdot\mathbb{U}_n^B(\omega_z)\|\widehat{\mathbf{\Omega}}_j\widehat{\mathbf{\Sigma}}^B\|_\infty\big\|\widehat{\boldsymbol{\gamma}}-\boldsymbol{\gamma}^*\big\|_1$. By (122) and Theorem 6, we have with probability $1-1/d$,

$$\mathbb{P}_\xi\Big(\sup_{z,j,k}|I_2^B|\le C\sqrt{nh}\cdot sM^2\lambda(\lambda+\sqrt{\log(d/h)(nh^2)})\Big)\le1/d.$$

For $\lambda = C_{\mathbf{\Sigma}}\big(h^2 + \sqrt{\log(d/h)/(nh)}\,\big)$, as $s\log d/\sqrt{nh^3} = o(n^{-c})$ and $\sqrt{nh^5} = o(n^{-c})$, we have with probability $1 - 1/d$,

$$\mathbb{P}_\xi\left(\sup_{j,k\in[d]}\sup_{z\in(0,1)}\sqrt{nh}\cdot\Big|\widehat{S}^B_{z|(j,k)}\big(\widehat{\mathbf{\Omega}}_{k\backslash j}(z)\big) - \mathbf{\Omega}_j^T(z)\Big(\widehat{\mathbf{\Sigma}}^B(z)\mathbf{\Omega}_k(z) - \mathbf{e}_k^T\Big)\Big| > n^{-c}\right)$$
$$\leq \mathbb{P}_\xi\left(\sup_{j,k\in[d]}\sup_{z\in(0,1)}\big(|I^B_{12}| + |I^B_2|\big) > n^{-c}\right) \leq 1/d,$$

following the same proof of Lemma 21. The proof is therefore complete. ∎

G.3 Properties of Bootstrap Score Statistics

In this section, we focus on establishing certain properties of the Gaussian multiplier bootstrap statistics introduced in this paper. The main goal is to prove Lemma 20, which states the approximation rate of a leading linear term to the bootstrap score statistic. To that end, we establish a rate of convergence for the bootstrap Kendall's tau estimator $\widehat{\tau}^B_{jk}(z)$ parallel to the results for $\widehat{\tau}^B_{jk}(z)$ in Lemma 12.

Recall from (17) and (20) that

$$\widehat{\tau}^B_{jk}(z) = \frac{\sum_{i\neq i'} K_h\left(Z_i - z\right) K_h\left(Z_{i'} - z\right)\operatorname{sign}(X_{ij} - X_{i'j})\operatorname{sign}(X_{ik} - X_{i'k})(\xi_i + \xi_{i'})}{\sum_{i\neq i'} K_h\left(Z_i - z\right) K_h\left(Z_{i'} - z\right)(\xi_i + \xi_{i'})},$$

and

$$\mathbb{U}_n[\omega^B_z] = \frac{2}{n(n-1)}\sum_{i\neq i'} K_h\left(Z_i - z_0\right) K_h\left(Z_{i'} - z_0\right)(\xi_i + \xi_{i'}).$$

The following lemma presents a convergence rate of the bootstrap Kendall's tau estimator $\widehat{\tau}^B_{jk}(z)$.

Lemma 23 *Under the conditions of Lemma 20, with probability* $1 - c/d$,

$$\mathbb{P}_\xi\left(\max_{j,k\in[d]}\sup_{z\in(0,1)}\Big|[\mathbb{U}_n[\omega^B_z]]\big(\widehat{\tau}^B_{jk}(z) - \tau_{jk}(z)\big)\Big| > C\sqrt{\log(d/h)/(nh^2)}\right) \leq 1/d$$

and

$$\mathbb{P}_\xi\left(\max_{j,k\in[d]}\sup_{z\in(0,1)}\sqrt{nh}|\Delta W_{z|(j,k)}| > C\sqrt{\log(d/h)/(nh^2)}\right) \leq 1/d, \tag{123}$$

with $\Delta W_{z|(j,k)}$ *defined in* (116).

Proof We first introduce some notation to simplify the proof. Let

$$W_{z|(j,k)}(Y_i) = \frac{2}{n-1}\sum_{i'\neq i}\omega_z(Z_i, Z_{i'})\Big(\operatorname{sign}(X_{ij} - X_{i'j})\operatorname{sign}(X_{ik} - X_{i'k}) - \tau_{jk}(z)\Big). \tag{124}$$

From the definition of $\widehat{\tau}^B_{jk}(z)$ in (17), conditionally on $\{Y_i\}_{i\in[n]}$, we have

$$\sqrt{n}\cdot\mathbb{U}_n[\omega^B_{z_0}]\Big(\widehat{\tau}^B_{jk}(z) - \tau_{jk}(z)\Big) = \frac{1}{\sqrt{n}}\sum_{i=1}^n W_{z|(j,k)}(Y_i)\xi_i \sim N\Big(0, \frac{1}{n}\sum_{i=1}^n W^2_{z|(j,k)}(Y_i)\Big). \tag{125}$$

Since the bootstrap process in (124) is a Gaussian process, we bound its supreme using the Borell's inequality (see Proposition A.2.1, van der Vaart and Wellner (1996)). The Borell's inequality requires us bound the following three quantities:

1. the variance of $n^{-1}\sum_{i=1}^n W^2_{z|(j,k)}(Y_i)$;
2. the supremum norm of $n^{-1}\sum_{i=1}^n W^2_{z|(j,k)}(Y_i)$;
3. the L^2 norm covering number of the function class

$$\mathcal{F}_W = \left\{\omega^{(i)}_{z|(j,k)} \mid z\in(0,1), j,k\in[d]\right\}, \tag{126}$$

under the empirical measure $\mathbb{P}_n = n^{-1}\sum_{i=1}^n \delta_{Y_i}$, where

$$\omega^{(i)}_{z|(j,k)} := \frac{1}{n-1}\sum_{i'\neq i}\omega_z(Z_i, Z_{i'})\operatorname{sign}(X_{ij} - X_{i'j})\operatorname{sign}(X_{ik} - X_{i'k}).$$

To bound the variance, we first study $W_{z|(j,k)}(Y_i)$ for each single $i\in[n]$. For any bivariate function $f(y_1, y_2)$, define the operator

$$\mathbb{G}^{(i)}_{n-1}[f] = \frac{1}{\sqrt{n-1}}\sum_{i'\neq i}^n\left(f(Y_{i'}, Y_i) - \mathbb{E}\left[f(Y_{i'}, Y_i)\mid Y_i\right]\right).$$

Now, can be written as

$$W_{z|(j,k)}(Y_i) = \mathbb{E}[W_{z|(j,k)}(Y_i)] + \frac{2}{\sqrt{n-1}}\underbrace{\Big(\mathbb{G}^{(i)}_{n-1}(g_{z|(j,k)}) - \tau_{jk}(z)\mathbb{G}^{(i)}_{n-1}(\omega_z)\Big)}_{J^{(1)}(Y_i)}$$
$$+ \underbrace{2\left(g^{(1)}_{z|(j,k)}(Y_i) - \tau_{jk}(z)\omega^{(1)}_z(Z_i)\right)}_{J^{(2)}(Y_i)}. \tag{127}$$

From (84) and (94), we have that almost surely

$$\max_{i\in[n]}\sup_{j,k\in[d]}\sup_{z\in(0,1)} J^{(2)}(Y_i) \leq Ch^{-1}. \tag{128}$$

Using Lemma 13, we have

$$\sup_{z,j,k}\mathbb{E}[W_{z|(j,k)}(Y_i)] \leq \sup_{z,j,k} 2\Big(\big|\mathbb{E}\left[\mathbb{U}_n\left[g_{z|(j,k)}\right]\right] - f^2_Z(z)\tau_{jk}(z)\big| + \big|\mathbb{E}\left[\mathbb{U}_n\left[\omega_z\right]\right] - f^2_Z(z)\big|\Big) \leq Ch^2. \tag{129}$$

Similar to the proof of Lemma 15 and Lemma 17, with probability $1-\delta$, we have

$$\max_{i\in[n]}\sup_{j,k\in[d]}\sup_{z\in(0,1)}\left|\mathbb{G}_{n-1}^{(i)}\left[\frac{h^{3/2}\big(g_{z|(j,k)}-\tau_{jk}(z)\omega_z\big)}{\sqrt{\log(d/h)\vee\log(n/\delta)}}\right]\right|\le C. \tag{130}$$

Plugging (128), (129) and (130) into (127), with probability $1-1/d$, we have

$$\max_{j,k\in[d]}\sup_{z\in(0,1)}\frac{1}{n}\sum_{i=1}^{n}W_{z|(j,k)}^2(Y_i)\le\max_{i\in[n]}\max_{j,k\in[d]}\sup_{z\in(0,1)}W_{z|(j,k)}^2(Y_i)\le Ch^{-2}, \tag{131}$$

as $\log(d/h)/(nh)=o(1)$ and $h=o(1)$.

Next, we bound the covering number of the function class $\mathcal{F}_W$ defined in (126). For some M_0 to be determined later, let $\{K_h(z_\ell-\cdot)\}_{\ell\in[M_0]}$ be the ϵ-net of

$$\mathcal{K}=\{K\left((s-\cdot)/h\right)\mid s\in(0,1)\}.$$

That is, for any $z\in(0,1)$, there exists a z_ℓ such that $\|K_h(z_\ell-\cdot)-K_h(z-\cdot)\|_{L^2(\mathbb{P}_n)}^2\le\epsilon$. For this z_ℓ, we also have

$$\begin{aligned}\|\omega_{z|(j,k)}^{(i)}-\omega_{z_\ell|(j,k)}^{(i)}\|_{L^2(\mathbb{P}_n)}^2&\le\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{n-1}\sum_{i'\neq i}|\omega_z(Z_i,Z_{i'})-\omega_{z'}(Z_i,Z_{i'})|\right)^2\\&\le\frac{1}{n}\sum_{i=1}^{n}(K_h(z-Z_i))^2\left(\frac{1}{n-1}\sum_{i'\neq i}|K_h(z-Z_{i'})-K_h(z_\ell-Z_{i'})|^2\right)\\&\quad+\frac{1}{n}\sum_{i=1}^{n}\big(K_h(z-Z_i)-K_h(z_\ell-Z_i)\big)^2\left(\frac{1}{n-1}\sum_{i'\neq i}|K_h(z_\ell-Z_{i'})|^2\right)\\&\le C\epsilon^2h^{-2}.\end{aligned}$$

Therefore, as $M_0\le(C/\epsilon)^v$, we have $N(\mathcal{F}_W,\|\cdot\|_{L^2(\mathbb{P}_n)},h^{-1}\epsilon)\le d^2(C/\epsilon)^{3v}$.

Similarly, for the function class

$$\mathcal{F}_W'=\big\{W_{z|(j,k)}(Y_i)\mid z\in(0,1),j,d\in[d]\big\}$$

we have

$$N(\mathcal{F}_W',\|\cdot\|_{L^2(\mathbb{P}_n)},h^{-1}\epsilon)\le N(\mathcal{F}_W,\|\cdot\|_{L^2(\mathbb{P}_n)},h^{-1}\epsilon)\le d^2(C/\epsilon)^{6v}. \tag{132}$$

This bound follows by combining the the fact that $\tau_{jk}(\cdot)$ is Lipschitz (since $\mathbf{\Sigma}_{jk}(\cdot)\in\mathcal{H}(2,L)$) with Lemma 26 and Lemma 27.

Now, the Dudley's inequality (see Lemma 2.2.8 in van der Vaart and Wellner, 1996), together with the fact that $\mathbb{U}_n[\omega_{z_0}^B](\widehat{\tau}_{jk}^B(z)-\tau_{jk}(z))$ is normally distributed conditionally on data (see (125)), the upper bound and variance bound in (131) and the covering number on $\mathcal{F}_W'$ above, gives us

$$\mathbb{E}\left[\sup_{z\in(0,1)}\max_{j,k\in[d]}\mathbb{U}_n[\omega_{z_0}^B]\Big(\widehat{\tau}_{jk}^B(z)-\tau_{jk}(z)\Big)\right]\le C\sqrt{\frac{\log(d/h)}{nh^2}}.$$

Using the Borell's inequality, on the event that (131) is true, we have

$$\mathbb{P}_\xi\left(\sup_{z\in(0,1)}\max_{j,k\in[d]}\mathbb{U}_n[\omega_{z_0}^B]\Big(\widehat{\tau}_{jk}^B(z)-\tau_{jk}(z)\Big)\ge C\sqrt{\frac{\log(d/h)}{nh^2}}\right)\ge1/d.$$

Since (131) is true with probability $1-1/d$, the first part of the lemma is proved.

Similarly, we can bound $\Delta W_{z|(j,k)}$. By (127), we have

$$\Delta W_{z|(j,k)}=\bar{\xi}\cdot\mathbb{E}[W_{z|(j,k)}(Y)]+\frac{1}{n}\sum_{i=1}^{n}\frac{2J^{(1)}(Y_i)}{\sqrt{n-1}}\xi_i, \tag{133}$$

where $\bar{\xi}=n^{-1}\sum_{i=1}^n\xi_i$. From the concentration of sub-Gaussian random variables, we have $\mathbb{P}(|\bar{\xi}|<C\sqrt{\log d/n})\ge1-1/d$. Combining with (129), we have

$$\mathbb{P}_\xi\left(\sup_{z\in(0,1)}\max_{j,k\in[d]}\bar{\xi}\cdot\mathbb{E}[W_{z|(j,k)}(Y_i)]\ge\sqrt{\frac{Ch^4\log d}{n}}\right)\le1/d. \tag{134}$$

According to (130), with probability $1-1/d$, we have

$$\sup_{z\in(0,1)}\max_{j,k\in[d]}\frac{1}{n}\sum_{i=1}^{n}\left(\frac{2J^{(1)}(Y_i)}{\sqrt{n-1}}\right)^2\le\max_{i\in[n]}\sup_{z\in(0,1)}\max_{j,k\in[d]}\left(\frac{2J^{(1)}(Y_i)}{\sqrt{n-1}}\right)^2\le\frac{C\log(d/h)}{nh^3}. \tag{135}$$

Define the function class $\widetilde{F}_W=\{J^{(1)}(\cdot)\mid z\in(0,1),j,k\in[d]\}$. By the definition of $J^{(1)}$ in (127), we apply Lemma 26 to the covering number of function classes consisting of $W_{z|(j,k)}$, $g_{z|(j,k)}^{(1)}$ and $\omega_z^{(1)}$ in (132), (144) and (153) to obtain

$$N(\widetilde{\mathcal{F}}_W,\|\cdot\|_{L^2(\mathbb{P}_n)},h^{-1}\epsilon)\le d^2(C/\epsilon)^{cv}.$$

The Borell's inequality, on the event that (135) is true, gives us

$$\mathbb{P}_\xi\left(\max_{j,k\in[d]}\sup_{z\in(0,1)}\frac{1}{n}\sum_{i=1}^{n}J^{(1)}(Y_i)\xi_i\ge C\sqrt{\frac{\log(d/h)}{n^2h^3}}\right)\le1/d. \tag{136}$$

Plugging (134) and (136) into (133), the proof of the second part is complete. ■

The following lemma presents a convergence rate of $\mathbb{U}^B(\omega_z)$ to $f_Z^2(z)$.

Lemma 24 *Under the conditions of Lemma 20, with probability* $1-1/n$,

$$\mathbb{P}_\xi\left(\sup_{z\in(0,1)}\big|\mathbb{U}_n[\omega_z^B]]-f_Z^2(z)\big|>C\sqrt{\log(1/h)/(nh^2)}\right)\le1/n.$$

Proof The proof is similar to that of Lemma 23. We define

$$\overline{W}_{z|(j,k)}(Z_i)=\frac{2}{n-1}\sum_{i'\neq i}\Big(\omega_z(Z_i,Z_{i'})-f_Z^2(z)\Big).$$

Conditionally on the data $\{Y_i\}_{i\in[n]}$,

$$\sqrt{n}\cdot(\mathbb{U}_n[\omega_z^B]-f_Z^2(z))=\frac{1}{\sqrt{n}}\sum_{i=1}^n\overline{W}_{z|(j,k)}(Z_i)\xi_i\sim N\Big(0,\frac{1}{n}\sum_{i=1}^n\overline{W}^2_{z|(j,k)}(Z_i)\Big).$$

Note that

$$\overline{W}_{z|(j,k)}(Y_i)=\mathbb{E}[\overline{W}_{z|(j,k)}(Y_i)]+2(n-1)^{-1/2}\mathbb{G}^{(i)}_{n-1}[\omega_z]+2\omega_z^{(1)}(Z_i).$$

Similar to the proof of Lemma 17 and (130), with probability $1-\delta$,

$$\max_{j,k\in[d]}\sup_{z\in(0,1)}\frac{1}{n}\sum_{i=1}^n\overline{W}^2_{z|(j,k)}(Y_i)\le\max_{i\in[n]}\max_{j,k\in[d]}\sup_{z\in(0,1)}\overline{W}^2_{z|(j,k)}(Y_i)\le Ch^{-2}. \tag{137}$$

Using Lemmas 29, 26 and 27, we can bound the covering number of the function class

$$\mathcal{F}''_W=\{\overline{W}_{z|(j,k)}(Y_i)|z\in(0,1)\}$$

by $N(\mathcal{F}''_W,\|\cdot\|_{L^2(\mathbb{P}_n)},h^{-1}\epsilon)\le(C/\epsilon)^{6v}$.

The remained of the proof follows the proof of Lemma 23. Using the Dudley's and Borell's inequality (see Lemma 2.2.8 and Proposition A.2.1 van der Vaart and Wellner, 1996)), on the event that (137) is true, we have

$$\mathbb{P}_\xi\left(\sup_{z\in(0,1)}\big(\mathbb{U}_n[\omega_z^B]-f_Z^2(z)\big)\ge C\sqrt{\frac{\log(n/h)}{nh^2}}\right)\ge 1/n.$$

The lemma follows since (137) holds with probability $1-1/n$. ∎

G.4 Proof of Lemma 2

Recall that we defined the matrix $\mathbf{\Theta}^{(i)}$ in (57) with elements

$$\mathbf{\Theta}^{(i)}_{jk}(z)=\pi\cos\big((\pi/2)\tau_{jk}(z)\big)\cdot\frac{1}{n-1}\sum_{i'\neq i}\tau^{(1)}_{jk}(Y_{i'}),$$

and $\tau^{(1)}_{jk}$ defined in (12). The strategy of the proof is to establish

$$\frac{1}{n}\sum_{i=1}^n\Big(\widehat{\mathbf{\Omega}}_j^T(z_0)\widehat{\mathbf{\Theta}}^{(i)}(z_0)\widehat{\mathbf{\Omega}}_k(z_0)\Big)^2\xrightarrow{P}\mathrm{Var}(\mathbf{\Omega}_j^T(z_0)\mathbf{\Theta}_{z_0}\mathbf{\Omega}_k(z_0))\qquad\text{and}\tag{138}$$

$$[\mathbb{U}_n[\omega_{z_0}]]^2\xrightarrow{P}f_Z^4(z_0).\tag{139}$$

The lemma then follows from the Slutsky's theorem.

We first establish (138). Let

$$\Delta_1=\frac{1}{n}\sum_{i=1}^n\Big(\widehat{\mathbf{\Omega}}_j^T(z_0)\widehat{\mathbf{\Theta}}^{(i)}(z_0)\widehat{\mathbf{\Omega}}_k(z_0)\Big)^2-\frac{1}{n}\sum_{i=1}^n\Big(\mathbf{\Omega}_j^T(z_0)\widehat{\mathbf{\Theta}}^{(i)}(z_0)\widehat{\mathbf{\Omega}}_k(z_0)\Big)^2\text{ and}$$

$$\Delta_2=\frac{1}{n}\sum_{i=1}^n\Big(\mathbf{\Omega}_j^T(z_0)\widehat{\mathbf{\Theta}}^{(i)}(z_0)\widehat{\mathbf{\Omega}}_k(z_0)\Big)^2-\frac{1}{n}\sum_{i=1}^n\Big(\mathbf{\Omega}_j^T(z_0)\widehat{\mathbf{\Theta}}^{(i)}(z_0)\mathbf{\Omega}_k(z_0)\Big)^2.$$

We can bound Δ_1 as

$$\begin{aligned}|\Delta_1|&=\left|(\widehat{\mathbf{\Omega}}_j(z_0)-\mathbf{\Omega}_j(z_0))^T\cdot\frac{1}{n}\sum_{i=1}^n\widehat{\mathbf{\Theta}}^{(i)}(z_0)\widehat{\mathbf{\Omega}}_k(z_0)\widehat{\mathbf{\Omega}}_k^T(z_0)\widehat{\mathbf{\Theta}}^{(i)}(z_0)\cdot(\widehat{\mathbf{\Omega}}_j(z_0)+\mathbf{\Omega}_j(z_0))^T\right|\\&\le\big\|\widehat{\mathbf{\Omega}}_j(z_0)-\mathbf{\Omega}_j(z_0)\big\|_1\big\|\widehat{\mathbf{\Omega}}_j(z_0)+\mathbf{\Omega}_j(z_0)\big\|_1\big\|\widehat{\mathbf{\Omega}}_k(z_0)\big\|_1^2\big\|\widehat{\mathbf{\Theta}}^{(i)}(z_0)\big\|^2_{\max}\\&=O_P\big(M^4r_{2n}(2\pi)^2h^{-1}\big)=o_P(1),\end{aligned}\tag{140}$$

where the second equality is by Assumption 4.4 and $\big\|\widehat{\mathbf{\Theta}}^{(i)}(z_0)\big\|_{\max}\le 2\pi h^{-1/2}$ with probability $1-c/d$. Similarly,

$$|\Delta_2|=O_P\big(M^4r_{2n}(2\pi)^2h^{-1}\big)=o_P(1).\tag{141}$$

Next, $\max_{i\in[n]}h^{1/2}\|\widehat{\mathbf{\Theta}}^{(i)}-\mathbf{\Theta}^{(i)}\|_{\max}\le\Delta_{31}+\Delta_{32}$, where

$$\Delta_{31}=\max_{i\in[n],j,k\in[d]}\pi\left|\cos\left((\pi/2)\widehat{\tau}_{jk}(z_0)\right)-\cos\left((\pi/2)\tau_{jk}(z_0)\right)\right|\cdot\left|\frac{h^{1/2}}{n-1}\sum_{i'\neq i}\tau^{(1)}_{jk}(Y_{i'})\right|;$$

$$\Delta_{32}=\max_{i\in[n],j,k\in[d]}\pi\left|\cos\left((\pi/2)\tau_{jk}(z_0)\right)\right|\cdot h^{1/2}\left|q_{i,jk}(z_0)-\frac{1}{n-1}\sum_{i'\neq i}\tau^{(1)}_{jk}(Y_{i'})\right|,$$

where $q_{i,jk}$ is defined in (13). Using Lemma 9 and Lemma 10, with probability $1-c/d$

$$\Delta_{31}\le\pi^2\|K\|_\infty\cdot\max_{j,k}|\widehat{\tau}_{jk}(z_0)-\widehat{\tau}_{jk}(z_0)|\lesssim\left(h^2+\sqrt{\log(dn)/(nh)}\right).$$

Let $\bar{q}_{i,jk}=q_{i,jk}+\widehat{\tau}_{jk}(z_0)$. By the Hoeffding's inequality and union bound, we have

$$\mathbb{P}\left(\max_{i,j,k}h^{3/2}|\bar{q}_{i,jk}-\mathbb{E}[\bar{q}_{i,jk}]|>t\,\middle|\,Y_i\right)\le 2nd^2\exp\left(-(n-1)t^2/2\right).$$

Setting $t=\sqrt{\log(dn)/n}$ and taking the expectation above, with probability $1-c/d$,

$$\max_{i,j,k}h^{1/2}|\bar{q}_{i,jk}-\mathbb{E}[\bar{q}_{i,jk}]|\le\sqrt{\log(dn)/n}.$$

Similarly, with probability $1-c/d$,

$$\Delta_{32}\le\pi h^{1/2}\max_{j,k}|\widehat{\tau}_{jk}(z_0)-\widehat{\tau}_{jk}(z_0)|+\max_{i,j,k}h^{1/2}|\bar{q}_{i,jk}-\mathbb{E}[\bar{q}_{i,jk}]|\lesssim\sqrt{\log(dn)/n}+h^{3/2}.$$

Therefore,

$$\begin{aligned}\Delta_3&:=\frac{1}{n}\sum_{i=1}^n\Big(\mathbf{\Omega}_j^T(z_0)\widehat{\mathbf{\Theta}}^{(i)}(z_0)\mathbf{\Omega}_k(z_0)\Big)^2-\frac{1}{n}\sum_{i=1}^n\Big(\mathbf{\Omega}_j^T(z_0)\mathbf{\Theta}^{(i)}(z_0)\mathbf{\Omega}_k(z_0)\Big)^2\\&\le 4M^4\pi h^{-1/2}\max_{i\in[n]}\|\widehat{\mathbf{\Theta}}^{(i)}-\mathbf{\Theta}^{(i)}\|_{\max}=O_P\Big(h+\sqrt{\log(dn)/(nh^3)}\Big)=o_P(1).\end{aligned}\tag{142}$$

Finally, by the law of large numbers,

$$\frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{\Omega}_j^T(z_0)\mathbf{\Theta}^{(i)}(z_0)\mathbf{\Omega}_k(z_0)\right)^2 \xrightarrow{P} \mathbb{E}[\mathbf{\Omega}_j^T(z_0)\mathbf{\Theta}^{(i)}(z_0)\mathbf{\Omega}_k(z_0)]^2 = \mathrm{Var}(\mathbf{\Omega}_j^T(z_0)\mathbf{\Theta}_{z_0}\mathbf{\Omega}_k(z_0)).$$

Combining (140), (141) and (142), we prove (138).

Using Lemma 24 and the continuous mapping theorem, we also prove (139). By the Slutsky's theorem, we have $\widehat{\sigma}_{jk}(z_0) \xrightarrow{P} \sigma_{jk}(z_0)$.

Appendix H. Results on Covering Number

In this section, we present several results on the covering number of certain function classes. The first two lemmas, Lemmas 26 and 27, are preliminary technical lemmas that will be used to prove Lemmas 28, 29 and 30. Lemma 26 provides bounds on the covering numbers for function classes generated from products and additions of two function classes. Lemma 27 provides a bound on the covering number of a class of constant functions.

Before presenting these lemmas, we first state a result on the covering number of kernel functions.

Lemma 25 *(Lemma 22 of Nolan and Pollard, 1987). Let $K : \mathbb{R} \mapsto \mathbb{R}$ be a bounded variation function. The following function class*

$$\mathcal{K} = \left\{ K\left(\frac{s-\cdot}{h}\right) \,\middle|\, h > 0, s \in \mathbb{R} \right\}$$

indexed by the kernel bandwidth satisfies the uniform entropy condition

$$\sup_Q N(\mathcal{K}, L_2(Q), \epsilon) \le C\epsilon^{-v}, \quad \text{for all } \epsilon \in (0,1), \tag{143}$$

for some $C > 0$ and $v > 0$.

The following lemma is about the covering number of summation and product of functions.

Lemma 26 *Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two function classes satisfying*

$$N(\mathcal{F}_1, \|\cdot\|_{L_2(Q)}, a_1\epsilon) \le C_1\epsilon^{-v_1} \qquad \text{and} \qquad N(\mathcal{F}_2, \|\cdot\|_{L_2(Q)}, a_2\epsilon) \le C_2\epsilon^{-v_2}$$

for some $C_1, C_2, a_1, a_2, v_1, v_2 > 0$ and any $0 < \epsilon < 1$. Define $\|\mathcal{F}_\ell\|_\infty = \sup\{\|f\|_\infty, f \in \mathcal{F}_\ell\}$ for $\ell = 1, 2$ and $U = \|\mathcal{F}_1\|_\infty \vee \|\mathcal{F}_2\|_\infty$. For the function classes $\mathcal{F}_\times = \{f_1 f_2 \mid f_1 \in \mathcal{F}_1, f_2 \in \mathcal{F}_2\}$ and $\mathcal{F}_+ = \{f_1 + f_2 \mid f_1 \in \mathcal{F}_1, f_2 \in \mathcal{F}_2\}$, we have for any $\epsilon \in (0,1)$,

$$N(\mathcal{F}_\times, \|\cdot\|_{L_2(Q)}, \epsilon) \le C_1C_2\left(\frac{2a_1U}{\epsilon}\right)^{v_1}\left(\frac{2a_2U}{\epsilon}\right)^{v_2};$$
$$N(\mathcal{F}_+, \|\cdot\|_{L_2(Q)}, \epsilon) \le C_1C_2\left(\frac{2a_1}{\epsilon}\right)^{v_1}\left(\frac{2a_2}{\epsilon}\right)^{v_2}.$$

Proof For any $\epsilon \in (0,1)$, let $N_1 = \{f_{11}, \ldots, f_{1N_1}\}$ and $N_2 = \{f_{21}, \ldots, f_{2N_2}\}$ be the $\epsilon/(2U)$-net of $\mathcal{F}_1$ and $\mathcal{F}_2$ respectively with

$$N_1 \le C_1\left(\frac{2a_1U}{\epsilon}\right)^{v_1} \text{ and } N_2 \le C_2\left(\frac{2a_2U}{\epsilon}\right)^{v_2}.$$

Define the set $N = \{f_{1j}f_{2k} \mid f_{1j} \in N_1, f_{2k} \in N_2\}$. We now show that N is an ϵ-net for $\mathcal{F}_\times$. For any $f_1f_2 \in \mathcal{F}$, there exist two functions $f_{1j} \in N_1$ and $f_{2k} \in N_2$ such that $\|f_1 - f_{1j}\|_{\|\cdot\|_{L_2(Q)}} \le \epsilon/(2U)$ and $\|f_2 - f_{2k}\|_{\|\cdot\|_{L_2(Q)}} \le \epsilon/(2U)$. Moreover, we have $f_{1j}f_{2k} \in N$ and

$$\|f_1f_2 - f_{1j}f_{2k}\|_{\|\cdot\|_{L_2(Q)}} \le \|\mathcal{F}_2\|_\infty\|f_1 - f_{1j}\|_{\|\cdot\|_{L_2(Q)}} + \|\mathcal{F}_1\|_\infty\|f_2 - f_{2k}\|_{\|\cdot\|_{L_2(Q)}} \le \epsilon.$$

Therefore N is the ϵ-net for $\mathcal{F}_\times$. Similarly, we also have

$$\|(f_1 + f_2) - (f_{1j} + f_{2k})\|_{\|\cdot\|_{L_2(Q)}} \le \|f_1 - f_{1j}\|_{\|\cdot\|_{L_2(Q)}} + \|f_2 - f_{2k}\|_{\|\cdot\|_{L_2(Q)}} \le \epsilon/U.$$

So $N' = \{f_{1j} + f_{2k} \mid f_{1j} \in N_1, f_{2k} \in N_2\}$ is the ϵ/U-net of $\mathcal{F}_+$. We finally complete the proof by showing that

$$|N'| = |N| = N_1N_2 \le C_1C_2\left(\frac{2a_1U}{\epsilon}\right)^{v_1}\left(\frac{2a_2U}{\epsilon}\right)^{v_2}.$$

■

Lemma 27 *Let $f(s)$ be a Lipschitz function defined on $[a,b]$ such that $|f(s) - f(s')| \le L_f|s - s'|$ for any $s, s' \in [a,b]$. We define the constant function class $\mathcal{F}_c = \{g_s(\cdot) \equiv f(s) \mid s \in [a,b]\}$. For any probability measure Q, the covering number of $\mathcal{F}_c$ satisfies for any $\epsilon \in (0,1)$,*

$$N(\mathcal{F}_c, \|\cdot\|_{L_2(Q)}, \epsilon) \le L_f \cdot \frac{|b-a|}{\epsilon}.$$

Proof Let $N = \{a + i\epsilon/L_f \mid i = 0, \ldots, \lfloor L_f|b-a|/\epsilon \rfloor\}$. For any $g_{s_0} \in \mathcal{F}_c$, there exists a $s \in N$ such that $|s_0 - s| \le \epsilon/L_f$ and we have

$$\|g_{s_0} - g_s\|_{L^2(Q)} = |f(s_0) - f(s)| \le L_f|s_0 - s| \le \epsilon.$$

Therefore $\{g_s \mid s \in N\}$ is the ϵ-net of $\mathcal{F}_c$. As $|N| \le L_f|b-a|/\epsilon$, the lemma is proved. ■

The following lemma presents the covering number of function classes consisting of $g^{(1)}_{z|(j,k)}(\cdot)$ or $g^{(2)}_{z|(j,k)}(\cdot)$ defined in (41) and (42).

Lemma 28 *For some $0 < \underline{h} < \overline{h} < 1$, we consider the class of functions*

$$\mathcal{F}^{(1)} = \left\{\sqrt{h}\cdot g^{(1)}_{z|(j,k)} \,\middle|\, h \in [\underline{h}, \overline{h}], z \in (0,1), j, k \in [d]\right\}; \tag{144}$$
$$\mathcal{F}^{(2)} = \left\{h\cdot g^{(2)}_{z|(j,k)} \,\middle|\, h \in [\underline{h}, \overline{h}], z \in (0,1), j, k \in [d]\right\},$$

where $g^{(1)}_{z|(j,k)}$ *and* $g^{(2)}_{z|(j,k)}$ *are defined in* (41) *and* (42). *There exist constants* $C_{(1)}$ *and* $C_{(2)}$ *such that for any* $\epsilon \in (0,1)$

$$\sup_Q N(\mathcal{F}^{(1)}, \|\cdot\|_{L_2(Q)}, \epsilon) \leq \frac{d^2 C_{(1)}}{\underline{\mathrm{h}}^{v+9}\epsilon^{v+6}} \quad \text{and} \quad \sup_Q N(\mathcal{F}^{(2)}, \|\cdot\|_{L_2(Q)}, \epsilon) \leq \frac{d^2 C_{(2)}}{\underline{\mathrm{h}}^{2v+18}\epsilon^{4v+15}}.$$

Proof Recall that

$$g^{(1)}_{z|(j,k)}(y) = \mathbb{E}[g_{z|(j,k)}(y,Y)] - \mathbb{E}\left[\mathbb{U}_n(g_{z|(j,k)})\right] \qquad \text{and}$$
$$g^{(2)}_{z|(j,k)}(y_1,y_2) = g_{z|(j,k)}(y_1,y_2) - g^{(1)}_{z|(j,k)}(y_1) - g^{(1)}_{z|(j,k)}(y_2) - \mathbb{E}\left[\mathbb{U}_n(g_{z|(j,k)})\right].$$

Our proof strategy for bounding the covering numbers of $\mathcal{F}^{(1)}$ and $\mathcal{F}^{(2)}$ is to decompose them into the following three auxiliary function classes:

$$\mathcal{F}^{(1)}_{1,jk} = \left\{\mathbb{E}[g_{z|(j,k)}(y,Y)] \mid h \in [\underline{\mathrm{h}}, \overline{\mathrm{h}}], z \in (0,1)\right\};$$
$$\mathcal{F}^{(1)}_{2} = \{\mathbb{E}\left[\mathbb{U}_n(g_{h,z})\right] \mid h \in [\underline{\mathrm{h}}, \overline{\mathrm{h}}], z \in (0,1)\};$$
$$\mathcal{F}^{(2)}_{jk} = \{g_{z|(j,k)}(y_1,y_2) \mid h \in [\underline{\mathrm{h}}, \overline{\mathrm{h}}], z \in (0,1)\}.$$

Observe that we can write

$$\mathcal{F}^{(1)} = \{\sqrt{h}\cdot(f_1 - f_2) \mid h \in [\underline{\mathrm{h}}, \overline{\mathrm{h}}], f_1 \in \mathcal{F}^{(1)}_{1,jk}, f_2 \in \mathcal{F}^{(1)}_2, j,k \in [d]\};$$
$$\mathcal{F}^{(2)} = \{h\cdot(f_1 - f_2 - f_3 - f_4) \mid h \in [\underline{\mathrm{h}}, \overline{\mathrm{h}}], f_1 \in \mathcal{F}^{(2)}_{jk}, f_2, f_3 \in \mathcal{F}^{(1)}_{1,jk}, f_4 \in \mathcal{F}^{(1)}_2, j,k \in [d]\}.$$

Therefore, we can apply Lemma 26 on the addition and product of functions classes and Lemma 27 on the constant functions to bound the covering numbers of $\mathcal{F}^{(1)}$ and $\mathcal{F}^{(2)}$.

Covering number of $\mathcal{F}^{(1)}_{1,jk}$. We bound the covering number of $\mathcal{F}^{(1)}_{1,jk}$ first. Recall from (62) that for $y' = (z', \boldsymbol{x}')$

$$\mathbb{E}\left[g_{z|(j,k)}(y',Y)\right] = K_h(z'-z)\int K_h(s-z)\varphi\left(x'_j, x'_k, \boldsymbol{\Sigma}_{jk}(s)\right) f_Z(s)ds.$$

We have $\mathcal{F}^{(1)}_{1,jk} = \left\{f_1 \cdot f_2 \mid f_1 \in \{K_h(\cdot - z)\}, f_2 \in \mathcal{F}^{(1)}_{3,jk}\right\}$ where

$$\mathcal{F}^{(1)}_{3,jk} = \left\{q_{z,h}(x,y) = \int K_h(s-z)\varphi\left(x,y,\boldsymbol{\Sigma}_{jk}(s)\right) f_Z(s)ds, h \in [\underline{\mathrm{h}}, \overline{\mathrm{h}}], z \in (0,1)\right\}. \quad (145)$$

Let $\bar{\varphi}_{x,y}(s) = \varphi\left(x,y,\boldsymbol{\Sigma}_{jk}(s)\right) f_Z(s)$. Then $\mathcal{F}^{(1)}_{3,jk}$ is the class of functions generated by the convolution $q_{z,h}(x,y) = (K_h * \bar{\varphi}_{x,y})(z)$. The L_1 norm of the derivative of K_h can be bounded by

$$\|K'_h\|_1 = \int \frac{1}{h^2}\left|K'\left(\frac{t}{h}\right)\right| dt = h^{-1}\int |K'(t)|dt = h^{-1}\mathrm{TV}(K), \quad (146)$$

where $\mathrm{TV}(K)$ is the total variation of the kernel K. Similarly we have for any $h \in [\underline{\mathrm{h}}, \overline{\mathrm{h}}]$,

$$\left\|\frac{\partial}{\partial h}K_h\right\|_1 \leq \int h^{-2}|K(t/h)|dt + \int h^{-3}|K'(t/h)|dt = h^{-1}\|K\|_1 + h^{-2}\mathrm{TV}(K).$$

We can apply a similar argument as in (85) and derive that

$$\sup_{z_0,h,x,y}\left|\frac{\partial}{\partial z}q_{z,h}(x,y)\Big|_{z=z_0}\right| = \sup_{h,x,y}\|K'_h * \bar{\varphi}_{x,y}\|_\infty \leq \sup_{h,x,y}\|K'_h\|_1\|\bar{\varphi}_{x,y}\|_\infty \leq 2\underline{\mathrm{h}}^{-1}\mathrm{TV}(K)\bar{\mathrm{f}}_Z, \quad (147)$$

where the first equality is due to the property of the derivative of a convolution, the first inequality is because of Young's inequality and the last inequality is by (146). Similarly, we have

$$\begin{aligned}\sup_{z,h_0,x,y}\left|\frac{\partial}{\partial h}q_{z,h}(x,y)\Big|_{h=h_0}\right| &= \sup_{h_0,x,y}\|\nabla_{h=h_0}K_h * \bar{\varphi}_{x,y}\|_\infty \\ &\leq \sup_{h_0,x,y}\|\nabla_{h=h_0}K_h\|_1\|\bar{\varphi}_{x,y}\|_\infty \\ &\leq 2\bar{\mathrm{f}}_Z(\underline{\mathrm{h}}^{-1}\|K\|_1 + \underline{\mathrm{h}}^{-2}\mathrm{TV}(K)).\end{aligned} \quad (148)$$

Therefore for any $z_1, z_2 \in (0,1)$, $h_1, h_2 \in [\underline{\mathrm{h}}, \overline{\mathrm{h}}]$, denoting $C_h := 2\bar{\mathrm{f}}_Z[(\underline{\mathrm{h}}^{-1} + \underline{\mathrm{h}}^{-2})\mathrm{TV}(K) + \underline{\mathrm{h}}^{-1}\|K\|_1]$, we have

$$\sup_{x,y}|q_{z_1,h_1}(x,y) - q_{z_2,h_2}(x,y)| \leq C_h\max(|z_1 - z_2|, |h_1 - h_2|).$$

Given any measure Q on $\mathbb{R}^2$, let $\mathcal{Z}$ be the ϵ/C_h-net of $(0,1)\times[\underline{\mathrm{h}}, \overline{\mathrm{h}}]$ under $\|\cdot\|_\infty$. For any $z \in (0,1), h \in [\underline{\mathrm{h}}, \overline{\mathrm{h}}]$, choose $(z_0, h_0) \in \mathcal{Z}$ such that $\max(|z - z_0|, |h - h_0|) \leq \epsilon/C_h$ and we have

$$\|q_{z,h} - q_{z_0,h_0}\|_{\|\cdot\|_{L_2(Q)}} \leq \|q_{z,h} - q_{z_0,h_0}\|_\infty \leq \epsilon.$$

This shows that $\{q_{z,h} \mid (z,h) \in \mathcal{Z}\}$ is the ϵ-net of $\mathcal{F}^{(1)}_{3,jk}$ and

$$\sup_Q N(\mathcal{F}^{(1)}_{3,jk}, \|\cdot\|_{L_2(Q)}, \epsilon) \leq |\mathcal{Z}| \leq \left(\frac{C_h}{\epsilon}\right)^2. \quad (149)$$

According to the formulation in (62) and Lemma 26 and (143), we have

$$\sup_Q N(\mathcal{F}^{(1)}_{1,jk}, \|\cdot\|_{L_2(Q)}, \epsilon) \leq C\left(\frac{1}{\underline{\mathrm{h}}\epsilon}\right)^v\left(\frac{\overline{\mathrm{h}} - \underline{\mathrm{h}}}{\underline{\mathrm{h}}^2\epsilon}\right)\left(\frac{C_h}{\underline{\mathrm{h}}\epsilon}\right)^2 \leq \frac{CC_h^2}{\underline{\mathrm{h}}^{v+4}\epsilon^{v+3}}. \quad (150)$$

Covering number of $\mathcal{F}^{(1)}_2$. According to (51) and Assumptions (**T**) and (**D**), we have that for any $z_1, z_2 \in (0,1), h_1, h_2 \in [\underline{\mathrm{h}}, \overline{h}]$

$$\begin{aligned}\left|\mathbb{E}\left[\mathbb{U}_n(g^{(1)}_{z_1|(j,k)})\right] - \mathbb{E}\left[\mathbb{U}_n(g^{(1)}_{z_2|(j,k)})\right]\right| &\leq \left|f^2_Z(z_1)\tau_{jk}(z_1) - f^2_Z(z_2)\tau_{jk}(z_2)\right| + C|h_1^2 - h_2^2| \\ &\leq \bar{\mathrm{f}}^2_Z C_\tau|z_1 - z_2| + 2C\overline{\mathrm{h}}|h_1 - h_2|.\end{aligned}$$

Using Lemma 27, we have

$$\sup_Q N(\mathcal{F}^{(1)}_2, \|\cdot\|_{L_2(Q)}, \epsilon) \leq \left(\frac{\bar{\mathrm{f}}^2_Z C_\tau + 2C\overline{\mathrm{h}}}{\epsilon}\right)^2. \quad (151)$$

Covering number of $\mathcal{F}^{(1)}$. Observe that the function $g(h) = \sqrt{h}$ is Lipschitz on $[\underline{h}, \overline{h}]$ with Lipschitz constant $(\underline{h})^{-1/2}/2$. Combining (150) and (151) with Lemma 26, we have

$$\sup_Q N(\mathcal{F}^{(1)}, \|\cdot\|_{L_2(Q)}, \epsilon) \leq d^2 \cdot \frac{2CC_h^2(\bar{f}_Z^2 C_\tau + 2C\overline{h})^2}{\underline{h}^{v+5}\epsilon^{v+6}}. \tag{152}$$

Here the additional d^2 on the right hand side of (152) is because we also take supreme over $j, k \in [d]$ in the definition of $\mathcal{F}^{(1)}$. As $C_h \leq 2\bar{f}_Z(2\mathrm{TV}(K) + \|K\|_1) \cdot \underline{h}^{-2}$, defining

$$C_{(1)} := 4C\bar{f}_Z(2\mathrm{TV}(K) + \|K\|_1)(\bar{f}_Z^2 C_\tau + 2C)^2$$

and the first part of lemma in (144) is proved.

Covering numbers of $\mathcal{F}_{jk}^{(2)}$ and $\mathcal{F}^{(2)}$. Let $N_K = \{K_h(x-z)K_h(y-z) \mid z \in \mathcal{Z}_K, h \in \mathcal{H}_K\}$ be the ϵ-net of the function class $\mathcal{K}^2 = \{K_h(x-z)K_h(y-z) \mid h \in [\underline{h}, \overline{h}], z \in (0,1)\}$. According to Lemma 26 and (143), we have $|N_K| \leq C^2(2\|K\|_\infty/\epsilon)^{2v}$. Given any $g_{h_0,z_0} \in \mathcal{F}_{jk}^{(2)}$, there exist $h_1 \in \mathcal{Z}_K, z_1 \in \mathcal{H}_K$ such that

$$\|g_{h_0,z_0} - g_{h_1,z_1}\|_{\|\cdot\|_{L_2(Q)}} \leq \|K_{h_0}(\cdot - z_0)K_{h_0}(\cdot - z_0) - K_{h_1}(\cdot - z_1)K_{h_1}(\cdot - z_1)\|_{\|\cdot\|_{L_2(Q)}} \leq \epsilon.$$

Therefore $N_g = \{g_{z|(j,k)}(y_1, y_2) \mid z \in \mathcal{Z}_K, h \in \mathcal{H}_K\}$ is an ϵ-net of $\mathcal{H}$ and $|N_g| = |N_K| \leq C^2(2\|K\|_\infty/\epsilon)^{2v}$. Applying Lemma 26 again with (151) and (152), we have

$$\begin{aligned}\sup_Q N(\mathcal{F}^{(2)}, \|\cdot\|_{L_2(Q)}, \epsilon) &\leq d^2C^2\left(\frac{\overline{h}-\underline{h}}{\epsilon}\right)\left(\frac{2\|K\|_\infty}{\epsilon}\right)^{2v}\left(\frac{C_{(1)}}{\underline{h}^{v+9}\epsilon^{v+6}}\right)^2\left(\frac{\bar{f}_Z^2 C_\tau + 2C\overline{h}}{\epsilon}\right)^2 \\ &\leq \frac{d^2 C_{(2)}}{\underline{h}^{2v+18}\epsilon^{4v+15}},\end{aligned}$$

where $C_{(2)} := C^2 4^v \|K\|_\infty^{2v} C_{(1)}^2 (\bar{f}_Z^2 C_\tau + 2C)^2$. Therefore, we complete the proof of the lemma. ■

Similar to Lemma 28, we can also establish the covering number for function classes consisting of $\omega_z^{(1)}$ or $\omega_z^{(2)}$ in the following lemma.

Lemma 29 *For some $0 < \underline{h} < \overline{h} < 1$, we consider the class of functions*

$$\begin{aligned}\mathcal{K}^{(1)} &= \left\{\sqrt{h} \cdot \omega_z^{(1)} \,\middle|\, h \in [\underline{h}, \overline{h}], z \in (0,1)\right\}; \\ \mathcal{K}^{(2)} &= \left\{h \cdot \omega_z^{(2)} \,\middle|\, h \in [\underline{h}, \overline{h}], z \in (0,1)\right\},\end{aligned} \tag{153}$$

where $\omega_z^{(1)}$ and $\omega_z^{(2)}$ are defined in (44) *and* (45)*. There exist constants $C'_{(1)}$ and $C'_{(2)}$ such that for any $\epsilon \in (0,1)$*

$$\sup_Q N(\mathcal{K}^{(1)}, \|\cdot\|_{L_2(Q)}, \epsilon) \leq \frac{C'_{(1)}}{\underline{h}^{2v+7}\epsilon^{v+3}} \quad \text{and} \quad \sup_Q N(\mathcal{K}^{(2)}, \|\cdot\|_{L_2(Q)}, \epsilon) \leq \frac{C'_{(2)}}{\underline{h}^{4v+14}\epsilon^{4v+8}}.$$

Proof The proof is similar to Lemma 28. We first bound $\sup_Q N(\mathcal{K}^{(1)}, \|\cdot\|_{L_2(Q)}, \epsilon)$. We have

$$\begin{aligned}\omega_z^{(1)}(s) &= \mathbb{E}[K_h(s-z)K_h(Z-z)] - \mathbb{E}\left[\mathbb{U}_n(K_h(Z_i - z)K_h(Z_{i'} - z))\right] \\ &= K_h(s-z)\mathbb{E}[K_h(z-Z)] - \{\mathbb{E}[K_h(z-Z)]\}^2.\end{aligned}$$

We first study the covering number of the function class

$$\mathcal{C}_K = \{\mathbb{E}[K_h(z-Z)] = (K_h * f_Z)(z) \mid h \in [\underline{h}, \overline{h}], z \in (0,1)\},$$

where "$*$" denotes the convolution. Just as (145), $\mathcal{C}_K$ is also generated by convolutions. Similar to (147) and (148), we have

$$\begin{aligned}\sup_{z_0,h}\left|\nabla_z \mathbb{E}[K_h(z-Z)]\Big|_{z=z_0}\right| &\leq 2\underline{h}^{-1}\mathrm{TV}(K)\bar{f}_Z \text{ and} \\ \sup_{z,h_0}\left|\nabla_h \mathbb{E}[K_h(z-Z)]\Big|_{h=h_0}\right| &\leq 2\bar{f}_Z(\underline{h}^{-1}\|K\|_1 + \underline{h}^{-2}\mathrm{TV}(K)).\end{aligned}$$

Therefore, following the derivation of (149), for any $z_1, z_2 \in (0,1)$, $h_1, h_2 \in [\underline{h}, \overline{h}]$, we have

$$\left|\mathbb{E}[K_{h_1}(z_1 - Z)] - \mathbb{E}[K_{h_2}(z_2 - Z)]\right| \leq C_h \max(|z_1 - z_2|, |h_1 - h_2|),$$

where $C_h := 2\bar{f}_Z[(\underline{h}^{-1} + \underline{h}^{-2})\mathrm{TV}(K) + \underline{h}^{-1}\|K\|_1]$. From Lemma 27, we have

$$\sup_Q N(\mathcal{C}_K, \|\cdot\|_{L_2(Q)}, \epsilon) \leq C_h/\epsilon \quad \text{and}$$

$$\sup_Q N\big(\{(\mathbb{E}[K_h(z-Z)])^2 \mid z, h\}, \|\cdot\|_{L_2(Q)}, \epsilon\big) \leq C_h/\epsilon.$$

Using the fact that the function $q(h) = 1/\sqrt{h}$ is Lipschitz on $[\underline{h}, \overline{h}]$ with Lipschitz constant $(\underline{h})^{-3/2}/2$ together with Lemma 26, Lemma 27 and (143), we have

$$\sup_Q N(\mathcal{K}^{(1)}, \|\cdot\|_{L_2(Q)}, \epsilon) \leq C\left(\frac{1}{\underline{h}^2\epsilon}\right)^v\left(\frac{C_h}{\underline{h}\epsilon}\right)\left(\frac{C_h}{\epsilon}\right)\left(\frac{\overline{h}-\underline{h}}{\underline{h}^{3/2}\epsilon}\right) \leq \frac{C'_{(1)}}{\underline{h}^{2v+7}\epsilon^{v+3}},$$

where $C'_{(1)} := [2\bar{f}_Z(2\mathrm{TV}(K) + \|K\|_1)C]^2$. Function class $\mathcal{K}^{(2)}$ contains functions in the form

$$\omega_z^{(2)}(s,t) = K_h(s-z)K_h(t-z) - \omega_z^{(1)}(s) - \omega_z^{(1)}(t) - \mathbb{E}\left[\mathbb{U}_n(K_h(Z_i - z)K_h(Z_i - z))\right].$$

By Lemma 26, it suffices to study the covering number of

$$\mathcal{C}'_K = \{K_h(s-z)K_h(t-z) \mid h \in [\underline{h}, \overline{h}], z \in (0,1)\}.$$

Using Lemma 26 and (143) again, we have

$$\sup_Q N(\mathcal{C}'_K, \|\cdot\|_{L_2(Q)}, \epsilon) \leq C^2\left(\frac{2\|K\|_\infty}{\epsilon}\right)^{2v},$$

and therefore combining with the covering number in (153) and (151)

$$\sup_Q N(\mathcal{K}^{(2)}, \|\cdot\|_{L_2(Q)}, \epsilon) \leq C^2\left(\frac{2\|K\|_\infty}{\epsilon}\right)^{2v}\left(\frac{C'_{(1)}}{\underline{h}^{2v+7}\epsilon^{v+3}}\right)^2\left(\frac{C_h}{\epsilon}\right)\left(\frac{\overline{h}-\underline{h}}{\epsilon}\right) \leq \frac{C'_{(2)}}{\underline{h}^{4v+14}\epsilon^{4v+8}},$$

where $C'_{(2)} := C^2 4^v \|K\|_\infty^{2v}(C'_{(1)})^2(\bar{f}_Z^2 C_\tau + 2C)^2$. This completes the proof. ■

Lemma 30 *Suppose $\mathbf{\Omega}(z) \in \mathcal{U}(c, M, \rho)$ for all $z \in (0,1)$. Consider the class of functions $\mathcal{J} = \left\{J_{z|(j,k)} \mid z \in (0,1), j, k \in [d]\right\}$, where $J_{z|(j,k)}$ is defined in (48). There exists positive constants C and c such that*

$$\sup_Q N(\mathcal{J}, \|\cdot\|_{L_2(Q)}, \epsilon/\sqrt{h}) \leq \left(\frac{Cd}{h\epsilon}\right)^c.$$

Proof We denote for any $u, v \in [d]$ and $z \in (0,1)$ that

$$\mathbf{\Phi}_{uv}(z; Y) = \pi \cos\left(\tau_{uv}(z)\frac{\pi}{2}\right)\sqrt{h} \cdot \left[g^{(1)}_{z|(u,v)}(Y) - \tau_{uv}(z)\omega^{(1)}_z(Z)\right] \tag{154}$$

and the matrix $\mathbf{\Phi}(z; Y) = [\mathbf{\Phi}_{uv}(z; Y)]_{u,v\in[d]}$. In order to bound the covering number of $\mathcal{J}$, we define a larger function class

$$\mathcal{J}' = \left\{\mathbf{\Omega}_j^T(z)\mathbf{\Phi}(x; \cdot)\mathbf{\Omega}_k(z) \mid z, x \in (0,1), j, k \in [d]\right\}.$$

Given any measure Q, $j, k \in [d]$ and $x_1, x_2, z_1, z_2 \in (0,1)$, we first bound the difference

$$\begin{aligned}
&\|\mathbf{\Omega}_j^T(z_1)\mathbf{\Phi}(x_1; Y)\mathbf{\Omega}_k(z_1) - \mathbf{\Omega}_j^T(z_2)\mathbf{\Phi}(x_2; Y)\mathbf{\Omega}_k(z_2)\|^2_{L_2(Q)} \\
&\leq 3\|\mathbf{\Omega}_j(z_1) - \mathbf{\Omega}_j(z_2)\|_1^2 \max_{u,v}\|\mathbf{\Phi}_{uv}(x_1; Y)\|^2_{L_2(Q)}\|\mathbf{\Omega}_k(z_1)\|_1^2 \\
&\quad + 3\|\mathbf{\Omega}_j(z_2)\|_1^2 \max_{u,v}\|\mathbf{\Phi}_{uv}(z_1; Y) - \mathbf{\Phi}_{uv}(z_2; Y)\|^2_{L_2(Q)}\|\mathbf{\Omega}_k(z_1)\|_1^2 \\
&\quad + 3\|\mathbf{\Omega}_j(z_2)\|_1^2 \max_{u,v}\|\mathbf{\Phi}_{uv}(x_2; Y)\|^2_{L_2(Q)}\|\mathbf{\Omega}_k(z_1) - \mathbf{\Omega}_k(z_2)\|_1^2.
\end{aligned} \tag{155}$$

Since $\mathbf{\Omega}(z) \in \mathcal{U}(c, M, \rho)$ for any $z \in (0,1)$, we have $\sup_z \|\mathbf{\Omega}_j(z)\|_1 \leq M$. Next, using Theorem 2.5 of Stewart et al. (1990), for any $z_1, z_2 \in (0,1)$,

$$\|\mathbf{\Omega}(z_1) - \mathbf{\Omega}(z_2)\|_2 \leq \|\mathbf{\Omega}(z_1)\|_2\|\mathbf{\Omega}(z_2)(\mathbf{\Sigma}(z_1) - \mathbf{\Sigma}(z_2))\|_2.$$

Since $\mathbf{\Omega}(z) \in \mathcal{U}(c, M, \rho)$ for any $z \in (0,1)$, we further have

$$\|\mathbf{\Omega}(z_1) - \mathbf{\Omega}(z_2)\|_1 \leq \sqrt{d}\|\mathbf{\Omega}(z_1) - \mathbf{\Omega}(z_2)\|_2 \leq \rho^2\|\mathbf{\Sigma}(z_1) - \mathbf{\Sigma}(z_2)\|_2 \leq \rho^2 d^{3/2}\|\mathbf{\Sigma}(z_1) - \mathbf{\Sigma}(z_2)\|_{\max}.$$

Since $\mathbf{\Sigma}_{jk}(\cdot) \in \mathcal{H}(2, M_\sigma)$, we have

$$\begin{aligned}
\|\mathbf{\Omega}(z_1) - \mathbf{\Omega}(z_2)\|_1 &\leq \rho^2 d^{3/2}\|\mathbf{\Sigma}(z_1) - \mathbf{\Sigma}(z_2)\|_{\max} \\
&\leq \rho^2 d^{3/2}\|\mathbf{T}(z_1) - \mathbf{T}(z_2)\|_{\max} \leq \rho^2 M_\sigma d^{3/2}|z_1 - z_2|.
\end{aligned} \tag{156}$$

We next study the covering number of the function class $\mathcal{J}_{uv} = \{\mathbf{\Phi}_{uv}(z; \cdot) \mid z \in (0,1)\}$. By (84) and (94), we have

$$\max_{u,v,x}\|\mathbf{\Phi}_{uv}(x; Y)\|^2_{L_2(Q)} \leq \max_{u,v,x}\|\mathbf{\Phi}_{uv}(x; Y)\|^2_\infty \leq Ch^{-1}. \tag{157}$$

According to the definition in (154), $\mathbf{\Phi}_{uv}(z; \cdot)$ is obtained from products and summations of functions with known covering numbers, quantified in Lemmas 28 and 29. By Lemmas 27 and 26 and fixing the bandwidth $h = \underline{h} = \overline{h}$, $\sup_Q N(\mathcal{J}_{uv}, \|\cdot\|_{L_2(Q)}, \epsilon) \leq C/(h\epsilon)^{v_1}$ for any $u, v \in [d]$. Notice that the construction of covering sets in the proofs of Lemmas 28, and 29 is independent to the indices j, k. Therefore, we can construct a set $N_{(2)} \subset (0,1)$ with $|N_{(2)}| \leq C/(h\epsilon)^c$ such that for any $x \in (0,1)$, there exists a $x_\ell \in N_{(2)}$ with

$$\max_{u,v}\|\mathbf{\Phi}_{uv}(x; Y) - \mathbf{\Phi}_{uv}(x_\ell; Y)\|_{L_2(Q)} \leq \epsilon. \tag{158}$$

With this, we construct the covering set for $\mathcal{J}'$ as

$$N_{(3)} = N_{(2)} \times \{\ell\epsilon\sqrt{h} | \ell = 0, \ldots, \lfloor 1/(\epsilon\sqrt{h})\rfloor\}.$$

For any $(x, z) \in (0,1)^2$, we select $(x_\ell, z_\ell) \in N_{(3)}$ such that (158) holds and $|z - z_\ell| \leq \epsilon\sqrt{h}$. Therefore, by (155), (156) and (157), we have

$$\|\mathbf{\Omega}_j^T(z)\mathbf{\Phi}(x; Y)\mathbf{\Omega}_k(z) - \mathbf{\Omega}_j^T(z_\ell)\mathbf{\Phi}(x_\ell; Y)\mathbf{\Omega}_k(z_\ell)\|^2_{L_2(Q)} \leq Cd^3M^4\epsilon^2$$

and $\sup_Q N(\mathcal{J}', \|\cdot\|_{L_2(Q)}, d^{3/2}M^2\epsilon) \leq d^2|N_{(3)}| = (Cd/(h\epsilon))^c$. ■

Appendix I. Some Useful Results

Lemma 31 *Let $(Y_1, Y_2, Y_3, Y_4)^T \sim N_4(\mathbf{0}, \mathbf{K})$ with $\mathbf{K} = [K_{ab}]_{ab}$. We then have*

$$\mathbb{E}\left[\operatorname{sign}(Y_1 - Y_2)\operatorname{sign}(Y_3 - Y_4)\right] = \frac{2}{\pi}\arcsin\left(\frac{K_{13} + K_{24} - K_{23} - K_{14}}{\sqrt{(K_{11} + K_{22} - 2K_{12})(K_{33} + K_{44} - 2K_{34})}}\right).$$

Proof Observe that $(Y_1 - Y_2, Y_3 - Y_4)^T$ is distributed according to a bivariate Gaussian distribution with mean zero and Pearson correlation coefficient

$$\operatorname{Corr}\left[\mathrm{Y}_1 - \mathrm{Y}_2, \mathrm{Y}_3 - \mathrm{Y}_4\right] = \frac{\mathrm{K}_{13} + \mathrm{K}_{24} - \mathrm{K}_{23} - \mathrm{K}_{14}}{\sqrt{(\mathrm{K}_{11} + \mathrm{K}_{22} - 2\mathrm{K}_{12})(\mathrm{K}_{33} + \mathrm{K}_{44} - 2\mathrm{K}_{34})}}.$$

The result follows directly from the correspondence between Pearson correlation and Kendall's tau (Fang et al., 1990). ■

Corollary 32 *Let $(\boldsymbol{X}_1, Z_1), (\boldsymbol{X}_2, Z_2)$ be independently distributed according to the model in (1). Then we have*

$$\mathbb{E}\left[\operatorname{sign}(X_{1a} - x_{2a})\operatorname{sign}(X_{1b} - x_{2b}) \mid Z_1 = z_1, Z_2 = z_2\right] = \frac{2}{\pi}\arcsin\left(\frac{\mathbf{\Sigma}_{ab}(z_1) + \mathbf{\Sigma}_{ab}(z_2)}{2}\right).$$

Proof Follows directly from Lemma 31 by observing that

$$\operatorname{sign}(X_{1a} - X_{2a}) = \operatorname{sign}(f(X_{1a}) - f(X_{2a})),$$

since f is monotone, and using the fact that $f(\boldsymbol{X}_i)$ follows a Gaussian distribution. ■

Let $H : S^2 \mapsto \mathbb{R}$ be a symmetric kernel function. In the setting of our paper, we have $S^2 = \mathbb{R}^2 \times (0,1)$. A kernel is completely degenerate if

$$\mathbb{E}[H(Y_1, Y_2) \mid Y_2] = 0.$$

U-statistic based on the kernel H is called degenerate of order 1. See, for example, Serfling (2001).

Theorem 33 (Theorem 2, Major (2006)) *Let $\{Y_i\}_{i\in[n]}$ be independent and identically distributed random variables on a probability space $(S, \mathcal{S}, \mu)$. Let $\mathcal{F}$ be a separable space (with respect to μ) of $\mathcal{S}$-measurable μ-degenerate kernel functions that satisfies*

$$N(\epsilon, \mathcal{F}, L_2(\mu)) \le A\epsilon^{-v}, \quad \text{for all } 1 \ge \epsilon > 0,$$

where A and v are some fixed constants. Furthermore, we assume that the envelope function is bounded by 1, that is,

$$\sup_{y_1,y_2} |H(y_1, y_2)| \le 1, \quad \text{for all } H \in \mathcal{F} \text{ and} \tag{159}$$

$$\sup_{H\in\mathcal{F}} \mathbb{E}\left[H^2(Y_1, Y_2)\right] \le \sigma^2$$

for some $0 < \sigma \le 1$. Then there exist constants C_1, C_2, C_3 that depend only on A and v, such that

$$\mu\left[\sup_{H\in\mathcal{F}} \left|\frac{1}{n}\sum_{i\neq i'} H(Y_i, Y_{i'})\right| \ge t\right] \le C_1 \exp\left(-C_2 \frac{t}{\sigma}\right) \tag{160}$$

for all t such that

$$n\sigma^2 \ge \frac{t}{\sigma} \ge C_3\Big(v + \frac{\log A}{\log n}\Big)^{3/2} \log\left(\frac{2}{\sigma}\right). \tag{161}$$

Lemma 34 (Lemma A.1, van de Geer 2008) *Let $X_1, \ldots, X_n$ be independent random variables. The $\gamma_1, \ldots, \gamma_d$ be real-valued bounded functions satisfying*

$$\mathbb{E}[\gamma_j(Z_i)] = 0, \text{ for any } i \in [n], j \in [d]; \quad \|\gamma_k\|_\infty \le \eta_n \quad \text{and} \quad \max_{j\in[d]} \frac{1}{n}\sum_{i=1}^n \mathbb{E}\gamma_j^2(Z_i) \le \tau_n^2.$$

We have

$$\mathbb{E}\left[\max_{j\in[d]} \left|\frac{1}{n}\sum_{i=1}^n \gamma_j(Z_i)\right|\right] \le \sqrt{\frac{2\tau_n^2 \log(2d)}{n}} + \frac{\eta_n \log(2m)}{n}.$$

Appendix J. Notation Table

The following table collects notation used in the paper, their meanings and where they first appear.

Notation	Meaning	Page No.
$\boldsymbol{X}$	d-dimensional nonparanormal random vector	2
Z	index variable	2
Y	$Y = (\boldsymbol{X}, Z)$ denotes the full data variable	2
$\boldsymbol{\Sigma}(z)$	true correlation matrix	2
f	marginal transform	2
$\boldsymbol{\Omega}(z)$	inverse correlation matrix	2
$[n], [d]$	discrete sets $[n] = \{1, \ldots, n\}$, $[d] = \{1, \ldots, d\}$	5
$\mathbf{A} \circ \mathbf{B}$	Hadamard product $(\mathbf{A} \circ \mathbf{B})_{jk} = \mathbf{A}_{jk} \cdot \mathbf{B}_{jk}$	5
$f * g$	convolution $(f * g)(t) = \int f(t-x)g(x)dx$	5
$\mathrm{TV}(f)$	total variation of f	5
$\vee, \wedge$	$a \vee b = \max(a, b)$ and $a \wedge b = \min(a, b)$	5
$c, C, c_1, C_1, \ldots$	universal constants whose values may change from line to line	5
$\rightsquigarrow$	converge in distribution	5
$\xrightarrow{P}$	converge in probability	5
$\mathbb{1}(\cdot)$	indicator function	5
$\mathbb{E}_n[f]$	$= n^{-1}\sum_{i\in[n]} f(X_i)$	5
$\mathbb{G}_n[f]$	$= n^{-1/2}\sum_{i\in[n]} (f(X_i) - \mathbb{E}[f(X_i)])$	5
$\mathbb{U}_n[H]$	$= [n(n-1)]^{-1}\sum_{i\neq i'} H(X_i, X_{i'})$	5
$G^*(z)$	true graph	6
$E^*(z)$	true edge set	6
$\mathbf{e}_j$	the j-th canonical basis in $\mathbb{R}^d$	6
$\widehat{\boldsymbol{\Omega}}(z)$	a generic inverse correlation estimator	6
$\widehat{\boldsymbol{\Omega}}^{\mathrm{CLIME}}(z)$	CLIME inverse correlation estimator	6
$\widehat{\boldsymbol{\Omega}}^{\mathrm{CCLIME}}(z)$	calibrated CLIME inverse correlation estimator	6
λ	$\|\cdot\|_\infty$-tuning parameter calibrated CLIME	6
γ	$\|\cdot\|_1$-tuning parameter calibrated CLIME	6
$\tau_{jk}(z)$	Kendall's tau correlation	7
$\widehat{\tau}_{jk}(z)$	Kendall's tau estimator	7
h	bandwidth	7
K_h	The rescaled kernel $K_h(\cdot) = h^{-1}K(\cdot/h)$	7
ω_z	double kernel function $\omega_z(Z_i, Z_{i'}) = K_h(Z_i - z)K_h(Z_{i'} - z)$	7
$\widehat{\boldsymbol{\Sigma}}(z)$	correlation matrix estimator	7
$\widehat{\mathbf{T}}(z)$	Kendall's tau matrix estimator	7
$[z_L, z_U]$	interval for uniform edge test	8
$\widehat{S}_{z\mid(j,k)}(\boldsymbol{\beta})$	$= \widehat{\boldsymbol{\Omega}}_j^T(z)\big(\widehat{\boldsymbol{\Sigma}}(z)\boldsymbol{\beta} - \mathbf{e}_k\big)$	8
f_Z	the density of Z	9
σ_{jk}^2	variance of the score statistic	9
$\widehat{\boldsymbol{\Theta}}_z$	random matrix for the definition of σ_{jk}^2	9
$\tau_{jk}^{(1)}(y)$	$= \sqrt{h} \cdot \mathbb{E}\Big[K_h(z - z_0)K_h(Z - z_0)\big(\mathrm{sign}(X_j - x_j)\mathrm{sign}(X_k - x_k) - \tau_{jk}(z_0)\big)\Big]$	9
ξ_i	i.i.d. sample of standard normal $N(0,1)$	10
$\widehat{\tau}_{jk}^B(z)$	bootstrap Kendall's tau estimator	10
$\widehat{\boldsymbol{\Sigma}}_{jk}^B(z)$	bootstrap correlation estimator	10
$\mathcal{H}(\gamma, L)$	the Hölder class on $(0,1)$	12
$\underline{f}_Z, \bar{f}_Z$	lower and upper bounds of f_Z	12
M_σ	the Hölder constant such that $\boldsymbol{\Sigma}_{jk}(\cdot) \in \mathcal{H}(2, M_\sigma)$	12
r_{1n}, r_{2n}, r_{3n}	the statistical rates of $\widehat{\boldsymbol{\Sigma}}(z)$ and $\widehat{\boldsymbol{\Omega}}(z)$	12
M	upper bound of column ℓ_1-norm of $\boldsymbol{\Omega}(z)$	13
ρ	maximum eigenvalue of $\boldsymbol{\Omega}(z)$	13

$\mathcal{U}_s(M,\rho)$	the class of matrices $\mathbf{\Omega} \succ 1/\rho, \|\mathbf{\Omega}\|_2 \le \rho, \max_{j\in[d]} \|\mathbf{\Omega}_j\|_0 \le s, \|\mathbf{\Omega}\|_1 \le M$	13
h_l, h_u	lower and upper bounds of bandwidth	15
$C_{\mathbf{\Sigma}}$	universal constant in the rate of $\widehat{\mathbf{\Sigma}}(z)$	16
$\overline{\mathcal{U}}_s(M,\rho,L)$	time-varying matrix class $\mathbf{\Omega}(z) \in \mathcal{U}_s(M,\rho)$	17
$g_{z\|(j,k)}(y_i, y_{i'})$	$= \omega_z(z_i, z_{i'}) \operatorname{sign}(x_{ia} - x_{i'a}) \operatorname{sign}(x_{ib} - x_{i'b})$	27
$g^{(1)}_{z\|(j,k)}(y)$	$= \mathbb{E}[g_{z\|(j,k)}(y,Y)] - \mathbb{E}\left[\mathbb{U}_n[g_{z\|(j,k)}]\right]$	27
$g^{(2)}_{z\|(j,k)}(y_1,y_2)$	$= g_{z\|(j,k)}(y_1,y_2) - g^{(1)}_{z\|(j,k)}(y_1) - g^{(1)}_{z\|(j,k)}(y_2) - \mathbb{E}\left[\mathbb{U}_n[g_{z\|(j,k)}]\right]$	27
$\omega^{(1)}_z(s)$	$= \mathbb{E}[\omega_z(s,Z)] - \mathbb{E}\left[\mathbb{U}_n[\omega_z]\right]$	27
$\omega^{(2)}_z(s,t)$	$= \omega_z(s,t) - \omega^{(1)}_z(s) - \omega^{(1)}_z(t) - \mathbb{E}\left[\mathbb{U}_n[\omega_z]\right]$	27
$\mathbb{G}^{\xi}_n[f]$	$= n^{-1/2} \sum_{i=1}^n f(X_i) \cdot \xi_i$	26
$\mathbb{P}_\xi$, $\mathbb{E}_\xi$	$\mathbb{P}_\xi(\cdot) := \mathbb{P}(\cdot \| \{Y_i\}_{i\in[n]})$ and $\mathbb{E}_\xi[\cdot] := \mathbb{E}[\cdot \| \{Y_i\}_{i\in[n]}]$	26
$J_{z\|(j,k)}(y')$	$= \sum_{u,v\in[d]} \mathbf{\Omega}_{ju}(z)\mathbf{\Omega}_{kv}(z)\pi \cos\left(\tau_{uv}(z)\frac{\pi}{2}\right) \sqrt{h} \cdot \left[g^{(1)}_{z\|(u,v)}(y') - \tau_{uv}(z)\omega^{(1)}_z(z')\right]$	27
$u_n[H]$	$= \sqrt{n} \cdot \left(\mathbb{U}_n[H] - \mathbb{E}[\mathbb{U}_n[H]]\right)$	28
$N(\mathcal{F}, d, \epsilon)$	covering number of function class $\mathcal{F}$ of ϵ-ball in metric d	61

References

R. Adamczak. Moment inequalities for U-statistics. *Ann. Probab.*, 34(6):2288–2314, 2006.

A. Ahmed and E. P. Xing. Recovering time-varying networks of dependencies in social and biological studies. *Proc. Natl. Acad. Sci. U.S.A.*, 106(29):11878–11883, 2009.

R. F. Barber and M. Kolar. Rocket: Robust confidence intervals via kendall's tau for transelliptical graphical models. *To appear in Ann. Statist.*, 2018.

G. Bartzokis, M. Beckson, P. H. Lu, K. H. Nuechterlein, N. Edwards, and J. Mintz. Age-related changes in frontal and temporal lobe volumes in men: a magnetic resonance imaging study. *Archives of General Psychiatry*, 58(5):461–465, 2001.

A. Belloni, V. Chernozhukov, and C. B. Hansen. Inference on treatment effects after selection amongst high-dimensional controls. *Rev. Econ. Stud.*, 81(2):608–650, 2013.

A. Belloni, V. Chernozhukov, and Y. Wei. Post-selection inference for generalized linear models with many controls. *J. Bus. Econom. Statist.*, 34(4):606–619, 2016.

H. J. Bierens. The nadaraya-watson kernel regression function estimator. 1988.

B. B. Biswal, M. Mennes, X.-N. Zuo, S. Gohel, C. Kelly, S. M. Smith, C. F. Beckmann, J. S. Adelstein, R. L. Buckner, S. Colcombe, and M. P. Milham. Toward discovery science of human brain function. *Proc. Natl. Acad. Sci. U.S.A.*, 107(10):4734–4739, 2010.

O. Bousquet. A Bennett concentration inequality and its application to suprema of empirical processes. *Comptes Rendus Mathematique*, 334(6):495–500, 2002.

E. T. Bullmore and D. S. Bassett. Brain graphs: graphical models of the human brain connectome. *Annual Review of Clinical Psychology*, 7:113–140, 2011.

T. T. Cai, W. Liu, and X. Luo. A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *J. Am. Stat. Assoc.*, 106(494):594–607, 2011.

T. T. Cai, H. Li, W. Liu, and J. Xie. Covariate-adjusted precision matrix estimation with an application in genetical genomics. *Biometrika*, 100(1):139–156, 2013a.

T. T. Cai, W. Liu, and Y. Xia. Two-sample covariance matrix testing and support recovery in high-dimensional and sparse settings. *J. Am. Stat. Assoc.*, 108(501):265–277, 2013b.

T. T. Cai, W. Liu, and H. H. Zhou. Estimating sparse precision matrix: optimal rates of convergence and adaptive estimation. *Ann. Statist.*, 44(2):455–488, 2016.

E. J. Candés and T. Tao. The Dantzig selector: Statistical estimation when p is much larger than n. *Ann. Stat.*, 35(6):2313–2351, 2007.

O. Catoni. Challenging the empirical mean and empirical variance: A deviation study. *Ann. Inst. H. Poincaré Probab. Statist.*, 48(4):1148–1185, 2012.

M. Chen, Z. Ren, H. Zhao, and H. Zhou. Asymptotically normal and efficient estimation of covariate-adjusted Gaussian graphical model. *J. Amer. Statist. Assoc.*, 111(513):394–406, 2016.

Z. Chen and C. Leng. Dynamic covariance models. *J. Amer. Statist. Assoc.*, 111(515): 1196–1208, 2016.

J. Cheng, T. Li, E. Levina, and J. Zhu. High-dimensional mixed graphical models. *J. Comput. Graph. Statist.*, 26(2):367–378, 2017.

V. Chernozhukov, D. Chetverikov, and K. Kato. Gaussian approximations and multiplier bootstrap for maxima of sums of high-dimensional random vectors. *Ann. Stat.*, 41(6): 2786–2819, 2013.

V. Chernozhukov, D. Chetverikov, and K. Kato. Anti-concentration and honest, adaptive confidence bands. *Ann. Stat.*, 42(5):1787–1818, 2014a.

V. Chernozhukov, D. Chetverikov, and K. Kato. Gaussian approximation of suprema of empirical processes. *Ann. Statist.*, 42(4):1564–1597, 2014b.

P. Danaher, P. Wang, and D. M. Witten. The joint graphical Lasso for inverse covariance estimation across multiple classes. *J. R. Stat. Soc. B*, 76(2):373–397, 2014.

A. d'Aspremont, O. Banerjee, and L. El Ghaoui. First-order methods for sparse covariance selection. *SIAM J. Matrix Anal. Appl.*, 30(1):56–66, 2008.

F. Dondelinger, S. Lebre, and D. Husmeier. Heterogeneous continuous dynamic Bayesian networks with flexible structure and inter-time segment information sharing. In J. Fürnkranz and T. Joachims, editors, *Proc. of ICML*, Haifa, Israel, 2010.

F. Dondelinger, S. Lébre, and D. Husmeier. Non-homogeneous dynamic Bayesian networks with Bayesian regularization for inferring gene regulatory networks with gradually time-varying structure. *Mach. Learn.*, 90:191–230, 2013.

U. Einmahl and D. M. Mason. Uniform in bandwidth consistency of kernel-type function estimators. *Ann. Stat.*, 33(3):1380–1403, 2005.

A. Eloyan, J. Muschelli, M. B. Nebel, H. Liu, F. Han, T. Zhao, A. D. Barber, S. Joel, J. J. Pekar, S. H. Mostofsky, and B. Caffo. Automated diagnoses of attention deficit hyperactive disorder using magnetic resonance imaging. *Front. Syst. Neurosci.*, 6, 2012.

J. Fan and I. Gijbels. Data-driven bandwidth selection in local polynomial fitting: variable bandwidth and spatial adaptation. *J. Roy. Statist. Soc. Ser. B*, 57(2):371–394, 1995.

J. Fan and J. Jiang. Nonparametric inferences for additive models. *J. Am. Stat. Assoc.*, 100(471):890–907, 2005.

J. Fan, Y. Feng, and Y. Wu. Network exploration via the adaptive Lasso and SCAD penalties. *Ann. Appl. Stat.*, 3(2):521–541, 2009.

J. Fan, H. Liu, Y. Ning, and H. Zou. High dimensional semiparametric latent graphical model for mixed data. *J. R. Stat. Soc. B*, 2015.

K. T. Fang, S. Kotz, and K. W. Ng. *Symmetric multivariate and related distributions*, volume 36 of *Monographs on Statistics and Applied Probability*. Chapman and Hall, Ltd., London, 1990.

J. H. Friedman, T. J. Hastie, and R. J. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

A. Fujita, J. R. Sato, H. M. Garay-Malpartida, P. A. Morettin, M. C. Sogayar, and C. E. Ferreira. Time-varying modeling of gene expression regulatory networks using the wavelet dynamic vector autoregressive method. *Bioinformatics*, 23(13):1623–1630, 2007.

A. E. Gelfand, H.-J. Kim, C. Sirmans, and S. Banerjee. Spatial modeling with spatially varying coefficient processes. *J. Am. Stat. Assoc.*, 98(462):387–396, 2003.

E. Giné, R. Latała, and J. Zinn. Exponential and moment inequalities for u-statistics. In *High Dimensional Probability II*, pages 13–38. Springer, 2000.

M. Grzegorczyk and D. Husmeier. Non-homogeneous dynamic bayesian networks for continuous data. *Mach. Learn.*, 83(3):355–419, 2011.

M. Grzegorczyk and D. Husmeier. Bayesian regularization of non-homogeneous dynamic bayesian networks by globally coupling interaction parameters. In N. Lawrence and M. Girolami, editors, *Proc. of AISTATS*, pages 467–476, 2012.

Q. Gu, Y. Cao, Y. Ning, and H. Liu. Local and global inference for high dimensional Gaussian copula graphical models. *arXiv preprint arXiv:1502.02347*, 2015.

J. Guo, E. Levina, G. Michailidis, and J. Zhu. Joint estimation of multiple graphical models. *Biometrika*, 98(1):1–15, 2011a.

J. Guo, E. Levina, G. Michailidis, and J. Zhu. Asymptotic properties of the joint neighborhood selection method for estimating categorical markov networks. Technical report, University of Michigan, 2011b.

P. Hall. On global properties of variable bandwidth density estimators. *Ann. Statist.*, 20 (2):762–778, 1992.

W. Hoeffding. A class of statistics with asymptotically normal distribution. *Ann. Math. Statist.*, 19(3):293–325, 1948.

S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

D. Hsu and S. Sabato. Heavy-tailed regression with a generalized median-of-means. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 37–45, 2014.

D. Husmeier, F. Dondelinger, and S. Lébre. Inter-time segment information sharing for non-homogeneous dynamic bayesian networks. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Proc. of NIPS*, pages 901–909, 2010.

J. Janková and S. A. van de Geer. Confidence intervals for high-dimensional inverse covariance estimation. *Electron. J. Statist.*, 9(1):1205–1229, 2015.

J. Janková and S. A. van de Geer. Honest confidence regions and optimality in high-dimensional precision matrix estimation. *TEST*, 26(1):143–162, 2017.

A. Javanmard and A. Montanari. Confidence intervals and hypothesis testing for high-dimensional regression. *J. Mach. Learn. Res.*, 15(Oct):2869–2909, 2014.

Y. Jia and J. Huan. Constructing non-stationary dynamic bayesian networks with a flexible lag choosing mechanism. *BMC Bioinformatics*, 11(Suppl 6):S27, 2010.

C. Jones, J. Marron, and S. Sheather. Progress in data-based bandwidth selection for kernel density estimation. *Computational Statistics*, (11):337–381, 1996.

M. Kolar and E. P. Xing. Sparsistent estimation of time-varying discrete markov random fields. *ArXiv e-prints, arXiv:0907.2337*, 2009.

M. Kolar and E. P. Xing. On time varying undirected graphs. In *Proc. of AISTATS*, 2011.

M. Kolar and E. P. Xing. Estimating networks with jumps. *Electron. J. Stat.*, 6:2069–2106, 2012.

M. Kolar, L. S. A. Ahmed, and E. P. Xing. Estimating Time-varying networks. *Ann. Appl. Stat.*, 4(1):94–123, 2010a.

M. Kolar, A. P. Parikh, and E. P. Xing. On sparse nonparametric conditional covariance selection. In J. Fürnkranz and T. Joachims, editors, *Proc. 27th Int. Conf. Mach. Learn.*, Haifa, Israel, 2010b.

V. Koltchinskii. *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems*, volume 2033 of *Lecture Notes in Mathematics*. Springer, Heidelberg, 2011. Lectures from the 38th Probability Summer School held in Saint-Flour, 2008, École d'Été de Probabilités de Saint-Flour.

C. Lam and J. Fan. Sparsistency and rates of convergence in large covariance matrix estimation. *Ann. Stat.*, 37:4254–4278, 2009.

L. Le Cam. Convergence of estimates under dimensionality restrictions. *Ann. Statist.*, 1: 38–53, 1973.

S. Lébre, J. Becq, F. Devaux, M. Stumpf, and G. Lelandais. Statistical inference of the time-varying structure of gene-regulation networks. *BMC Systems Biology*, 4(1):130, 2010.

J. D. Lee and T. J. Hastie. Learning the structure of mixed graphical models. *J. Comput. Graph. Statist.*, 24(1):230–253, 2015.

J. D. Lee, D. L. Sun, Y. Sun, and J. E. Taylor. Exact post-selection inference, with application to the lasso. *Ann. Statist.*, 44(3):907–927, 2016.

B. Li, H. Chun, and H. Zhao. Sparse estimation of conditional graphical models with application to gene networks. *J. Am. Stat. Assoc.*, 107(497):152–167, 2012.

H. Liu and L. Wang. TIGER: a tuning-insensitive approach for optimally estimating Gaussian graphical models. *Electron. J. Stat.*, 11(1):241–294, 2017.

H. Liu, J. D. Lafferty, and L. A. Wasserman. The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *J. Mach. Learn. Res.*, 10:2295–2328, 2009.

H. Liu, F. Han, M. Yuan, J. D. Lafferty, and L. A. Wasserman. High-dimensional semiparametric Gaussian copula graphical models. *Ann. Stat.*, 40(4):2293–2326, 2012a.

H. Liu, F. Han, and C.-H. Zhang. Transelliptical graphical models. In P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Proc. of NIPS*, pages 809–817. 2012b.

J. Lu, M. Kolar, and H. Liu. Post-regularization confidence bands for high dimensional nonparametric models with local sparsity. *arXiv preprint arXiv:1503.02978*, 2015.

P. Major. An estimate on the supremum of a nice class of stochastic integrals and u-statistics. *Probab. Theory Related Fields*, 134(3):489–537, 2006.

N. Meinshausen. Group bound: confidence intervals for groups of variables in sparse high dimensional regression without assumptions on the design. *J. R. Stat. Soc. Ser. B. Stat. Methodol.*, 77(5):923–945, 2015.

N. Meinshausen and P. Bühlmann. High dimensional graphs and variable selection with the Lasso. *Ann. Stat.*, 34(3):1436–1462, 2006.

R. Mitra and C.-H. Zhang. Multivariate analysis of nonparametric estimates of large correlation matrices. *ArXiv e-prints, arXiv:1403.6195*, 2014.

K. Mohan, P. London, M. Fazel, D. M. Witten, and S.-I. Lee. Node-based learning of multiple Gaussian graphical models. *J. Mach. Learn. Res.*, 15:445–488, 2014.

H.-G. Muller and U. Stadtmuller. Variable bandwidth kernel estimators of regression curves. *Ann. Statist.*, 15(1):182–201, 1987.

M. Neykov, Y. Ning, J. S. Liu, and H. Liu. A unified theory of confidence regions and testing for high dimensional estimating equations. *arXiv preprint arXiv:1510.08986*, 2015.

Y. Ning and H. Liu. A general theory of hypothesis tests and confidence regions for sparse high dimensional models. *Ann. Statist.*, 45(1):158–195, 2017.

D. Nolan and D. Pollard. U-processes: Rates of convergence. *Ann. Statist.*, 15(2):780–799, 1987.

J. D. Opsomer and D. Ruppert. Fitting a bivariate additive model by local polynomial regression. *Ann. Stat.*, 25(1):186–211, 1997.

A. Pagan and A. Ullah. *Nonparametric Econometrics (Themes in Modern Econometrics)*. Cambridge University Press, 1999.

J. D. Power, A. L. Cohen, S. M. Nelson, G. S. Wig, K. A. Barnes, J. A. Church, A. C. Vogel, T. O. Laumann, F. M. Miezin, B. L. Schlaggar, and S. E. Petersen. Functional network organization of the human brain. *Neuron*, 72(4):665–678, 2011.

E. Punskaya, C. Andrieu, A. Doucet, and W. J. Fitzgerald. Bayesian curve fitting using mcmc with applications to signal segmentation. *IEEE Trans. Signal Proces.*, 50(3):747–758, 2002.

H. Qiu, F. Han, H. Liu, and B. Caffo. Joint estimation of multiple graphical models from high dimensional time series. *J. R. Stat. Soc. Ser. B. Stat. Methodol.*, 78(2):487–504, 2016.

A. Rao, I. A. O. Hero, D. J. States, and J. D. Engel. Inferring time-varying network topologies from gene expression data. *EURASIP J. Bioinformatics Syst. Bio.*, 2007(1): 51947, 2007.

Z. Ren, T. Sun, C.-H. Zhang, and H. H. Zhou. Asymptotic normality and optimalities in estimation of large gaussian graphical models. *Ann. Statist.*, 43(3):991–1026, 2015.

J. W. Robinson and A. J. Hartemink. Learning non-stationary dynamic bayesian networks. *J. Mach. Learn. Res.*, 11:3647–3680, 2010.

A. J. Rothman, P. J. Bickel, E. Levina, and J. Zhu. Sparse permutation invariant covariance estimation. *Electron. J. Stat.*, 2:494–515, 2008.

D. Ruppert, S. J. Sheather, and M. P. Wand. An effective bandwidth selector for local least squares regression. *J. Amer. Statist. Assoc.*, 90(432):1257–1270, 1995.

R. J. Serfling. *Approximation Theorems of Mathematical Statistics*. Wiley-Interscience, 2001.

S. M. Smith, K. L. Miller, G. Salimi-Khorshidi, M. Webster, C. F. Beckmann, T. E. Nichols, J. D. Ramsey, and M. W. Woolrich. Network modelling methods for fmri. *Neuroimage*, 54(2):875–891, 2011.

L. Song, M. Kolar, and E. P. Xing. Time-varying dynamic bayesian networks. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1732–1740. Curran Associates, Inc., 2009.

G. W. Stewart, J.-g. Sun, and H. B. Jovanovich. *Matrix perturbation theory*, volume 175. Academic press New York, 1990.

M. Talih and N. Hengartner. Structural learning with time-varying components: Tracking the cross-section of the financial time series. *J. R. Stat. Soc. B*, 67(3):321–341, 2005.

The ADHD-200 Consortium. The ADHD-200 consortium: a model to advance the translational potential of neuroimaging in clinical neuroscience. *Front. Syst. Neurosci.*, 6, 2012.

R. J. Tibshirani, J. Taylor, R. Lockhart, and R. Tibshirani. Exact post-selection inference for sequential regression procedures. *J. Amer. Statist. Assoc.*, 111(514):600–620, 2016.

A. B. Tsybakov. *Introduction To Nonparametric Estimation*. Springer Series in Statistics. Springer, New York, 2009.

S. A. Van de Geer. *Applications of empirical process theory*, volume 91. Cambridge University Press Cambridge, 2000.

S. A. van de Geer. High-dimensional generalized linear models and the lasso. *Ann. Stat.*, 36(2):614–645, 2008.

S. A. van de Geer, P. Bühlmann, Y. Ritov, and R. Dezeure. On asymptotically optimal confidence regions and tests for high-dimensional models. *Ann. Stat.*, 42(3):1166–1202, 2014.

A. W. van der Vaart and J. A. Wellner. *Weak Convergence and Empirical Processes: With Applications to Statistics*. Springer, 1996.

J. Wang, X. Zuo, and Y. He. Graph-based network analysis of resting-state functional mri. *Frontiers in systems neuroscience*, 4:16, 2010.

Z. Wang, E. E. Kuruoglu, X. Yang, Y. Xu, and T. S. Huang. Time varying dynamic bayesian network for nonstationary events modeling and online inference. *IEEE Trans. Signal Proces.*, 59(4):1553–1568, 2011.

L. A. Wasserman, , and A. Rinaldo. Berry-Esseen bounds for estimating undirected graphs. *Electron. J. Stat.*, 8:1188–1224, 2014.

X. Xuan and K. Murphy. Modeling changing dependency structure in multivariate time series. In *Proc. of ICML*, ICML '07, pages 1055–1062, New York, NY, USA, 2007. ACM.

L. Xue and H. Zou. Regularized rank-based estimation of high-dimensional nonparanormal graphical models. *Ann. Stat.*, 40(5):2541–2571, 2012.

E. Yang, G. I. Allen, Z. Liu, and P. Ravikumar. Graphical models via generalized linear models. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1358–1366. Curran Associates, Inc., 2012.

E. Yang, Y. Baker, P. Ravikumar, G. I. Allen, and Z. Liu. Mixed graphical models via exponential families. In *Proc. 17th Int. Conf, Artif. Intel. Stat.*, pages 1042–1050, 2014a.

Z. Yang, Y. Ning, and H. Liu. On semiparametric exponential family graphical models. *arXiv preprint arXiv:1412.8697*, 2014b.

J. Yin, Z. Geng, R. Li, and H. Wang. Nonparametric covariance model. *Stat. Sinica*, 20: 469–479, 2010.

J. Yin and H. Li. Adjusting for high-dimensional covariates in sparse precision matrix estimation by 1-penalization. *J. Multivar. Anal.*, 116:365–381, 2013.

M. Yuan. High dimensional inverse covariance matrix estimation via linear programming. *J. Mach. Learn. Res.*, 11:2261–2286, 2010.

M. Yuan and Y. Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

C.-H. Zhang and S. S. Zhang. Confidence intervals for low dimensional parameters in high dimensional linear models. *J. R. Stat. Soc. B*, 76(1):217–242, 2013.

T. Zhao and H. Liu. Calibrated precision matrix estimation for high dimensional elliptical distributions. *IEEE Trans. Inf. Theory*, 60:7874–7887, 2014.

S. Zhou, J. D. Lafferty, and L. A. Wasserman. Time varying undirected graphs. *Mach. Learn.*, 80(2-3):295–319, 2010.

Journal of Machine Learning Research 18 (2018) 1-33 Submitted 7/17; Revised 12/17; Published 4/18

Permuted and Augmented Stick-Breaking Bayesian Multinomial Regression

Quan Zhang quan.zhang@mccombs.utexas.edu
Mingyuan Zhou mingyuan.zhou@mccombs.utexas.edu
Department of Information, Risk, and Operations Management
McCombs School of Business
The University of Texas at Austin
Austin, TX 78712, USA

Editor: David M. Blei

Abstract

To model categorical response variables given their covariates, we propose a permuted and augmented stick-breaking (paSB) construction that one-to-one maps the observed categories to randomly permuted latent sticks. This new construction transforms multinomial regression into regression analysis of stick-specific binary random variables that are mutually independent given their covariate-dependent stick success probabilities, which are parameterized by the regression coefficients of their corresponding categories. The paSB construction allows transforming an arbitrary cross-entropy-loss binary classifier into a Bayesian multinomial one. Specifically, we parameterize the negative logarithms of the stick failure probabilities with a family of covariate-dependent softplus functions to construct nonparametric Bayesian multinomial softplus regression, and transform Bayesian support vector machine (SVM) into Bayesian multinomial SVM. These Bayesian multinomial regression models are not only capable of providing probability estimates, quantifying uncertainty, increasing robustness, and producing nonlinear classification decision boundaries, but also amenable to posterior simulation. Example results demonstrate their attractive properties and performance.

Keywords: Discrete choice models, logistic regression, nonlinear classification, softplus regression, support vector machines

1. Introduction

Inferring the functional relationship between a categorical response variable and its covariates is a fundamental problem in physical and social sciences. To address this problem, it is common to use either multinomial logistic regression (MLR) (McFadden, 1973; Greene, 2003; Train, 2009) or multinomial probit regression (Albert and Chib, 1993; McCulloch and Rossi, 1994; McCulloch et al., 2000; Imai and van Dyk, 2005), both of which can be expressed as a latent-utility-maximization model that lets an individual make the decision by comparing its random utilities across all categories at once. In this paper, we address the problem via a new stick-breaking construction of the multinomial distribution, which defines a one-to-one random mapping between the category and stick indices. Rather than assuming an individual compares its random utilities across all categories at once, we assume an individual makes a sequence of stick-specific binary random decisions. The choice of the individual is the category mapped to the stick that is the first to choose "1," or the category mapped to stick S if all the first $S-1$ sticks choose "0." This framework transforms the problem of regression analysis of categorical variables into the problem of inferring the one-to-one mapping between the category and stick indices, and performing regression analysis of binary stick-specific random variables.

Both MLR and the proposed stick-breaking models link a categorical response variable to its covariate-dependent probability parameters. While MLR is invariant to the permutation of category labels, given a fixed category-stick mapping, the proposed stick-breaking models purposely destruct that invariance. We are motivated to introduce this new framework for discrete choice modeling mainly to facilitate efficient Bayesian inference via data augmentation, introduce nonlinear decision boundaries, and relax a well-recogonized restrictive model assumption of MLR, as described below.

An important motivation is to extend efficient Bayesian inference available to binary regression to multinomial one. In the proposed stick-breaking models, the binary stick-specific random variables of an individual are conditionally independent given their stick-specific covariate-dependent probabilities. Under this setting, one can solve a multinomial regression by solving conditionally independent binary ones. The only requirement is that the underlying binary regression model uses the cross entropy loss. In other words, we require each stick-specific binary random variable to be linked via the Bernoulli distribution to its corresponding stick-specific covariate-dependent probability parameter.

Another important motivation is to improve the model capacity of MLR, which is a linear classifier in the sense that if the total number of categories is S, then MLR uses the intersection of $S-1$ linear hyperplanes to separate one class from the others. By choosing nonlinear binary regression models, we are able to enhance the capacities of the proposed stick-breaking models. We are also motivated to relax the *independence of irrelevant alternative* (IIA) assumption, an inherent property of MLR that requires the probability ratio of any two choices to be independent of the presence or characteristics of any other alternatives (McFadden, 1973; Greene, 2003; Train, 2009). By contrast, the proposed stick-breaking models make the probability ratio of two choices depend on other alternatives, as long as the two sticks that both choices are mapped to are not next to each other.

In light of these considerations, we will first extend the softplus regressions recently proposed in Zhou (2016), a family of cross-entropy-loss binary classifiers that can introduce nonlinear decision boundaries and can recover logistic regression as a special case, to construct Bayesian multinomial softplus regressions (MSRs). We then consider a multinomial generalization of the widely used support vector machine (SVM) (Boser et al., 1992; Cortes and Vapnik, 1995; Schölkopf et al., 1999; Cristianini and Shawe-Taylor, 2000), a max-margin binary classifier that uses the hinge loss. While there has been significant effort in extending binary SVMs into multinomial ones (Crammer and Singer, 2002; Lee et al., 2004; Liu and Yuan, 2011), the resulted extensions typically only provide the predictions of deterministic class labels. By contrast, we extend the Bayesian binary SVMs in Sollich (2002) and Mallick et al. (2005) under the proposed framework to construct Bayesian multinomial SVMs (MSVMs), which naturally provide predictive class probabilities.

We will show that the proposed Bayesian MSRs and MSVMs, which all generalize the stick-breaking construction to perform Bayesian multinomial regression, are not only capable of placing nonlinear decision boundaries between different categories, but also amenable

to posterior simulation via data augmentation. Another attractive feature shared by all these proposed Bayesian algorithms is that they can not only predict class probabilities but also quantify model uncertainty. In addition, we will show that robit regression, a robust cross-entropy-loss binary classifier proposed in Liu (2004), can be extended into a robust Bayesian multinomial classifier under the proposed stick-breaking construction.

The remainder of the paper is organized as follows. In Section 2 we briefly review MLR and discuss the restrictions of its stick-breaking construction. In Section 3 we propose the permuted and augmented stick breaking (paSB) to construct Bayesian multi-class classifiers, present the inference, and show how the IIA assumption is relaxed. Under the paSB framework, we show how to transform softplus regressions and support vector machines into Bayesian multinomial regression models in Sections 4 and 5, respectively. We provide experimental results in Section 6 and conclude the paper in Section 7.

2. Multinomial Logistic Regression and Stick Breaking

In this section we first briefly review multinomial logistic regression (MLR). We then use the stick-breaking construction to show how to generate a categorical random variable as a sequence of dependent binary variables, and further discuss a naive approach to transform binary logistic regression under stick breaking into multinomial regression. In the following discussion, we use $i \in \{1, \ldots, N\}$ to index the individual/observation, $s \in \{1, \ldots, S\}$ to index the choice/category, and the prime symbol to denote the transpose operation.

2.1 Multinomial Logistic Regression

MLR that parameterizes the probability of each category given the covariates as

$$P(y_i = s \,|\, \boldsymbol{x}_i, \{\boldsymbol{\beta}_s\}_{1,S}) = p_{is},\ p_{is} = e^{\boldsymbol{x}_i'\boldsymbol{\beta}_s} / \big(\textstyle\sum_{j=1}^{S} e^{\boldsymbol{x}_i'\boldsymbol{\beta}_j}\big) \tag{1}$$

is widely used, where $\boldsymbol{x}_i \in \mathbb{R}^{P+1}$ consists of $x_{i1} = 1$ and P covariates, and $\boldsymbol{\beta}_s \in \mathbb{R}^{P+1}$ consists of the regression coefficients for the sth category (McCullagh and Nelder, 1989; Albert and Chib, 1993; Holmes and Held, 2006). Without loss of generality, one may choose category S as the reference category by setting all the elements of $\boldsymbol{\beta}_S$ as 0, making $e^{\boldsymbol{x}_i'\boldsymbol{\beta}_S} = 1$ almost surely (a.s.). For MLR, if data i is assigned to the category with the largest p_{is}, then one may consider that category s resides within a convex polytope (Grünbaum, 2013), defined by the set of solutions to $S-1$ inequalities as $\boldsymbol{x}'(\boldsymbol{\beta}_j - \boldsymbol{\beta}_s) \le 0$, where $j \in \{1, \ldots, s-1, s+1, \ldots, S\}$.

Despite its popularity, MLR is a linear classifier in the sense that it uses the intersection of $S-1$ linear hyperplanes to separate one class from the others. As a classical discrete choice model in econometrics, it makes the independence of irrelevant alternatives (IIA) assumption, implying that the unobserved factors for choice making are both uncorrelated and having the same variance across all alternatives (McFadden, 1973; Train, 2009). Moreover, while its log-likelihood is convex and there are efficient iterative algorithms to find the maximum likelihood or maximum a posteriori solutions of $\boldsymbol{\beta}_s$, the absence of conjugate priors on $\boldsymbol{\beta}_s$ makes it difficult to derive efficient Bayesian inference. For Bayesian inference, Polson et al. (2013) have introduced the Pólya-Gamma data augmentation for logit models, and combined it with the data augmentation technique of Holmes and Held (2006) for the multinomial likelihood to develop a Gibbs sampling algorithm for MLR. This algorithm, however, has to update $\boldsymbol{\beta}_s$ one at a time while conditioning on all $\boldsymbol{\beta}_j$ for $j \neq s$. Thus it may not only lead to slow convergence and mixing, especially when the number of categories S is large, but also prevent us from parallelizing the sampling of $\{\boldsymbol{\beta}_s\}_{1,S}$ within each MCMC iteration.

2.2 Stick Breaking

Suppose y_i is a random variable drawn from a categorical distribution with a finite vector of probability parameters $(p_{i1}, \ldots, p_{iS})$, where $S < \infty$, $p_{is} \ge 0$, and $\sum_{s=1}^{S} p_{is} = 1$. Instead of directly using $y_i \sim \sum_{s=1}^{S} p_{is}\delta_s$, one may consider generating y_i using the multinomial stick-breaking construction that sequentially draws binary random variables

$$b_{is} \,\big|\, \{b_{ij}\}_{j<s} \sim \text{Bernoulli}\Big[\Big(1 - \textstyle\sum_{j<s} b_{ij}\Big)\,\pi_{is}\Big],\quad \pi_{is} = \dfrac{p_{is}}{1 - \sum_{j<s} p_{ij}} \tag{2}$$

for $s = 1, 2, \ldots, S$. Note that $\pi_{iS} = 1$ and $b_{iS} = 1 - \sum_{j=1}^{S-1} b_{ij}$ by construction. Defining $y_i = s$ if and only if $b_{is} = 1$ and $b_{ij} = 0$ for all $j \neq s$, then one has a strick-breaking representation for the multinomial probability parameter as

$$P(y_i = s \,|\, \{\pi_{is}\}_{1,S}) = P(b_{is} = 1) \textstyle\prod_{j \neq s} P(b_{ij} = 0) = \pi_{is} \prod_{j<s} (1 - \pi_{ij}), \tag{3}$$

which, as expected, recovers p_{is} by substituting the definitions of π_{is} shown in (2).

The finite stick-breaking construction in (3) can be further generalized to an infinite setting, as widely used in Bayesian nonparametrics (Hjort et al., 2010). For example, the stick-break construction of Sethuraman (1994) represents the length of the kth stick using the product of k stick-specific probabilities that are independent, and identically distributed (i.i.d.) beta random variables. It represents a size-biased random permutation of a Dirichlet process (DP) (Ferguson, 1973) random draw, which includes countably infinite atoms whose weights sum to one. The stick-breaking construction of Sethuraman (1994) has also been generalized to represent a draw from a random probability measure that is more general than the DP (Pitman, 1996; Ishwaran and James, 2001; Wang et al., 2011a).

Related to this paper, one may further consider making the stick-specfic probabilities depend on the covariates (Dunson and Park, 2008; Chung and Dunson, 2009; Ren et al., 2011). For example, the logistic stick-breaking process of Ren et al. (2011) uses the product of k covariate-dependent logistic functions to parameterize the probability of the kth stick. To implement a stick-breaking process mixture model, truncated stick-breaking representations with a finite number of sticks are commonly used, with inference developed via both Gibbs sampling (Ishwaran and James, 2001; Dunson and Park, 2008; Rodriguez and Dunson, 2011) and variational approximation (Blei and Jordan, 2006; Kurihara et al., 2007; Ren et al., 2011).

Another related work is the order-based dependent Dirichlet processes of Grifin and Steel (2006), which use an ordered stick-breaking construction for mixture modeling, encouraging the data samples close to each other in the covariate space to share similar orders of the sticks and hence similar mixture weights. We will show that the proposed stick-breaking construction is distinct in that all data samples share the same category-stick mapping inferred from the data, with the category labels mapped to lower-indexed sticks subject to fewer geometric constraints on their decision boundaries.

2.3 Logistic Stick Breaking

The stick-breaking construction parameterizes each p_{is} with the product of s probability parameters and links each y_i with a unit-norm binary vector $(b_{i1}, \ldots, b_{iS})$, where $b_{iy_i} = 1$ and $b_{ij} = 0$ a.s. if $j \neq y_i$. Following the logistic stick-breaking construction of Ren et al. (2011), one may represent p_{is} with (3) and parameterize the logit of each π_{is} with a latent Gaussian variable w_{is} as $\pi_{is} = e^{w_{is}}/\sum_{j=1}^{S} e^{w_{ij}}$. To model observed or latent multinomial variables, a stick-breaking procedure, closely related to that of Ren et al. (2011), is used in Khan et al. (2012) to transform the modeling of multinomial probability parameters into the modeling of the logits of binomial probability parameters using Gaussian latent variables. As shown in Linderman et al. (2015), this procedure allows using the Pólya-Gamma data augmentation, without requiring the assistance of the technique of Holmes and Held (2006), to construct Gibbs sampling that simultaneously updates all categories in each MCMC iteration, leading to improved performance over the one proposed in Polson et al. (2013).

The simplification brought by the stick-breaking representation, which stochastically arranges its categories in decreasing order, comes with a clear change in that it removes the invariance of the multinomial distribution to label permutation. While the loss of invariance to label permutation may not pose a major issue for Bayesian mixture models inferred with MCMC (Jasra et al., 2005; Kurihara et al., 2007), it appears to be a major obstacle when applying stick breaking for multinomial regression, where the performance is often found to be sensitive to how the labels of the S categories are ordered. In particular, if one constructs a logistic stick breaking model by letting $\text{logit}(\pi_{is}) = w_{is} = \boldsymbol{x}_i'\boldsymbol{\beta}_s$, which means $\pi_{is} = (1 + e^{-\boldsymbol{x}_i'\boldsymbol{\beta}_s})^{-1}$, then one has

$$p_{is} = \left(1 + e^{-\boldsymbol{x}_i'\boldsymbol{\beta}_s}\right)^{-1} \prod_{j<s} \left(1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}_j}\right)^{-1},$$

which clearly tends to impose fewer geometric constraints on the classification decision boundaries of a category with a smaller s. For example, $p_{i1} = \left(1 + e^{-\boldsymbol{x}_i'\boldsymbol{\beta}_1}\right)^{-1}$ is larger than 50% if $\boldsymbol{x}_i'\boldsymbol{\beta}_1 > 0$ while $p_{i2} = \left(1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}_1}\right)^{-1}\left(1 + e^{-\boldsymbol{x}_i'\boldsymbol{\beta}_2}\right)^{-1}$ is possible to be larger than 50% only if both $\boldsymbol{x}_i'\boldsymbol{\beta}_1 < 0$ and $\boldsymbol{x}_i'\boldsymbol{\beta}_2 > 0$. We will use an example to illustrate this type of geometric constraints in Section 6.1.

Under the logistic stick-breaking construction, not only could the performance be sensitive to how the S different categories are ordered, but the imposed geometric constraints could also be overly restrictive even if the categories are appropriately ordered. Below we address the first issue by introducing a permuted and augmented stick-breaking representation for a multinomial model, and the second issue by adding the ability to model nonlinearity.

3. Permuted and Augmented Stick Breaking

To turn the seemingly undesirable sensitivity of the stick-breaking construction to label permutation into a favorable model property, when label asymmetry is desired, and mitigate performance degradation, when label symmetry is desired, we introduce a permuted and augmented stick-breaking (paSB) construction for a multinomial distribution, making it straightforward to extend an arbitrary binary classifier with cross entropy loss into a

Bayesian multinomial one. The paSB construction infers a one-to-one mapping between the labels of the S categories and the indices of the S latent sticks, transforming the problem from modeling a multinomial random variable into modeling S conditionally independent binary ones. It not only allows for parallel computation within each MCMC iteration, but also improves the mixing of MCMC in comparison to the one used in Polson et al. (2013), which updates one regression-coefficient vector conditioning on all the others, as will be shown in Section 6.5. Note that the number of distinct one-to-one label-stick mappings is $S!$, which quickly becomes too large to exhaustively search for the best mapping as S increases. Our experiments will show that the proposed MCMC algorithm can quickly escape from a purposely poorly initialized mapping and subsequently switch between many different mappings that all lead to similar performance, suggesting an effective search space that is considerably smaller than $S!$.

3.1 Category-Stick Mapping and Data Augmentation

The proposed paSB construction randomly maps a category to one and only one of the S latent sticks and makes the augmented Bernoulli random variables $\{b_{is}\}_{1,S}$ conditionally independent to each other given $\{\pi_{is}\}_{1,S}$. Denote $\boldsymbol{z} = (z_1, \ldots, z_S)$ as a permutation of $(1, \ldots, S)$, where $z_s \in \{1, \ldots, S\}$ is the index of the stick that category s is mapped to. Given the label-stick mapping $\boldsymbol{z}$, let us denote $p_{is}(\boldsymbol{z})$ as the multinomial probability of category s, and $\pi_{iz_s}(\boldsymbol{x}_i, \boldsymbol{\beta}_s)$ as the covariate-dependent stick probability that is associated with the covariates of observation i and the stick that category s is mapped to. For notational convenience, we will write $\pi_{iz_s}(\boldsymbol{x}_i, \boldsymbol{\beta}_s)$ as π_{iz_s} and $\pi_{ij}(\boldsymbol{x}_i, \boldsymbol{\beta}_{s:z_s=j})$ as π_{ij}. We emphasize that here the sth regression-coefficient vector $\boldsymbol{\beta}_s$ is always associated with both category s and the corresponding stick probabily π_{iz_s}, a construction that will facilitate the inference of the label-stick mapping $\boldsymbol{z}$. The following Theorem shows how to generate a categorical random variable of S categories with a set of S conditionally independent Bernoulli random variables. This is key to transforming the problem from solving multinomial regression into solving S binary regressions independently.

Theorem 1 *Suppose* $y_i \sim \sum_{s=1}^{S} p_{is}(\boldsymbol{z})\delta_s$, *where* $[p_{i1}(\boldsymbol{z}), \ldots, p_{iS}(\boldsymbol{z})]$ *is a multinomial probability vector whose elements are constructed as*

$$p_{is}(\boldsymbol{z}) = (\pi_{iz_s})^{\mathbf{1}(z_s \neq S)} \prod_{j<z_s} (1 - \pi_{ij}), \tag{4}$$

then y_i *can be equivalently generated under the permuted and augmented stick-breaking (paSB) construction as*

$$y_i \sim \sum_{s=1}^{S} \left\{ [\mathbf{1}(b_{iz_s} = 1)]^{\mathbf{1}(z_s \neq S)} \prod_{j<z_s} \mathbf{1}(b_{ij} = 0) \right\} \delta_s\,, \tag{5}$$

$$b_{ij} \sim \text{Bernoulli}(\pi_{ij}), \quad j \in \{1, \ldots, S\}. \tag{6}$$

Distinct from the conventional stick breaking in (2) that maps category s to stick s and makes b_{is} depend on $b_{ij}, j = 1, \ldots, s-1$, under the new construction in (5)-(6), the S categories are now randomly permuted and then one-to-one mapped to S sticks, and

the augmented binary random variables $\{b_{ij}\}_j$ become mutually independent given $\{\pi_{ij}\}_j$. Given y_i, we still have $b_{ij} = 0$ for $j < z_{y_i}$ and $b_{iz_{y_i}} = 1$ a.s., but impose no restriction on any b_{ij} for $j > z_{y_i}$, whose conditional posteriors given y_i and π_{ij} remain the same as their priors. These changes are key to appropriately ordering the latent sticks, more flexibly parameterizing π_{iz_s} and hence $p_{is}(\boldsymbol{z})$, and maintaining tractable inference.

With paSB, the problem of inferring the functional relationship between the categorical response y_i and the corresponding covariates $\boldsymbol{x}_i$ is now transformed into the problem of modeling S conditionally independent binary regressions as

$$b_{iz_s} \,|\, \boldsymbol{x}_i, \boldsymbol{\beta}_s \sim \text{Bernoulli}[\pi_{iz_s}(\boldsymbol{x}_i, \boldsymbol{\beta}_s)],\ i = 1, \ldots, N,\ s = 1, \ldots, S.$$

Note that the only requirement for the binary regression model under paSB is that it uses the Bernoulli likelihood. In other words, it uses the cross entropy loss (Murphy, 2012) as

$$-\sum_{i=1}^{N} \ln P(b_{iz_s} \,|\, \boldsymbol{x}_i, \boldsymbol{\beta}_s) = \sum_{i=1}^{N} \big\{ -b_{iz_s} \ln \pi_{iz_s}(\boldsymbol{x}_i, \boldsymbol{\beta}_s) - (1 - b_{iz_s}) \ln[1 - \pi_{iz_s}(\boldsymbol{x}_i, \boldsymbol{\beta}_s)] \big\}.$$

A basic choice is paSB logistic regression that lets

$$\pi_{iz_s}(\boldsymbol{x}_i, \boldsymbol{\beta}_s) = 1/(1 + e^{-\boldsymbol{x}_i \boldsymbol{\beta}_s}),$$

which becomes the same as the logistic stick breaking construction described in Section 2.3 if $z_s = s$ for all $s \in \{1, \ldots, S\}$. Another choice is paSB-robit regression that extends robit regression of Liu (2004), a robust binary classifier using cross entropy loss, into a robust Bayesian multinomial classifier. In robit regression, observation i is labeled as 1 if $\boldsymbol{x}_i'\boldsymbol{\beta} + \varepsilon_i > 0$ and as 0 otherwise, where ε_i are independently drawn from a t-distribution with κ degrees of freedom, denoted as $\varepsilon_i \overset{iid}{\sim} t_\kappa$. Consequently, the conditional class probability function of robit regression is $P(y_i = 1 \,|\, \boldsymbol{x}_i, \boldsymbol{\beta}) = F_\kappa(\boldsymbol{x}_i'\boldsymbol{\beta})$, where F_κ is the cumulative density function of t_κ. The robustness is attributed to the heavy-tail property of $F_\kappa(\boldsymbol{x}_i'\boldsymbol{\beta})$, which, if $\kappa < 7$, imposes less penalty than the conditional class probability function of logistic regression does on misclassified observations that are far from the decision boundary. Applying Theorem 1, the category probability of paSB-robit regression with κ degrees of freedom is shown in (4), where $\pi_{iz_s}(\boldsymbol{x}_i, \boldsymbol{\beta}_s) = F_\kappa(\boldsymbol{x}_i'\boldsymbol{\beta}_s)$. The paSB-robit regression provides a simple solution to robust multiclass classification; with $\{b_{ij}\}_{i,j}$ defined in Theorem 1, we run independent binary robit regressions using the Gibbs sampler proposed in Liu (2004).

In addition to paSB, we define permuted and augmented reverse stick breaking (parSB) in the following Corollary.

Corollary 2 *Suppose* $y_i \sim \sum_{s=1}^{S} p_{is}\delta_s$ *and*

$$p_{is}(\boldsymbol{z}) = (1 - \pi_{iz_s})^{\mathbf{1}(z_s \neq S)} \textstyle\prod_{j<z_s} \pi_{ij}\,,$$

then y_i *can also be generated under the permuted and augmented reverse stick-breaking (parSB) representation as*

$$y_i \sim \sum_{s=1}^{S} \Big\{ [\mathbf{1}(b_{iz_s} = 0)]^{\mathbf{1}(z_s \neq S)} \textstyle\prod_{j<z_s} \mathbf{1}(b_{ij} = 1) \Big\} \delta_s\,, \qquad (7)$$
$$b_{ij} \sim \text{Bernoulli}(\pi_{ij}), \quad j \in \{1, \ldots, S\}.$$

Generally speaking, if $\pi_{iz_s}(\boldsymbol{x}_i, -\boldsymbol{\beta}_s) = 1 - \pi_{iz_s}(\boldsymbol{x}_i, \boldsymbol{\beta}_s)$, which is the case for logistic stick breaking and robit stick breaking, where π_{iz_s} are defined as $(1 + e^{-\boldsymbol{x}_i'\boldsymbol{\beta}_s})^{-1}$ and $F_\kappa(\boldsymbol{x}_i'\boldsymbol{\beta}_s)$, respectively, and Bayesian multinomial SVMs to be discussed in Section 5, then there is no need to introduce parSB as an addition to paSB. Otherwise, there are potential benefits, such as for softplus regressions to be introduced in Section 4, to combine parSB with paSB.

3.2 Inference of Stick Variables and Category-Stick Mapping

Below we first describe Gibbs sampling for the augmented stick variables $\{b_{ij}\}_{1,S}$, and then introduce a Metropolis-Hastings (MH) step to infer the category-stick mapping $\boldsymbol{z}$. Given the category label y_i, stick probability π_{ij}, and $\boldsymbol{z}$, we sample b_{ij} as

$$(b_{ij} \,|\, y_i, \pi_{ij}, \boldsymbol{z}) \sim \mathbf{1}(j = z_{y_i}) + \mathbf{1}(j > z_{y_i})\text{Bernoulli}(\pi_{ij}),$$

for $j = 1, \ldots, S-1$, and let

$$b_{iS} = \mathbf{1}(z_{y_i} = S).$$

This means we let $b_{ij} = 0$ if $j < z_{y_i}$, $b_{ij} = 1$ if $j = z_{y_i}$, draw b_{ij} from Bernoulli(π_{ij}) if $z_{y_i} < j < S$, and let $b_{iS} = 1$ if and only if $z_{y_i} = S$. Note that stick S is used as a reference stick and π_{iS} is not used in defining $p_{is}(\boldsymbol{z})$ in (4). Despite having no impact on computing $\{p_{is}\}_{1,S}$, we infer π_{iS} (*i.e.*, sample the regression-coefficient vector $\boldsymbol{\beta}_{s':z_{s'}=S}$) under the likelihood $\prod_{i=1}^{N} \text{Bernoulli}(b_{iS}; \pi_{iS})$ and use it in a Metropolis-Hastings step, as described in (8) shown below, to decide whether to switch the mappings of two different categories, if one of which is mapped to the reference stick S. Once we have an MCMC sample of $\{b_{ij}\}_{1,S}$, we then essentially solve independently S binary classification problems, the jth of which can be expressed as $b_{ij} \,|\, \boldsymbol{x}_i, \boldsymbol{\beta}_{s:z_s=j} \sim \text{Bernoulli}[\pi_{ij}(\boldsymbol{x}_i, \boldsymbol{\beta}_{s:z_s=j})]$.

Analogously, for parSB, $\{b_{ij}\}_{1,S}$ can be sampled as $(b_{ij} \,|\, y_i, \pi_{ij}, \boldsymbol{z}) \sim \mathbf{1}(j < z_{y_i}) + \mathbf{1}(j > z_{y_i})\text{Bernoulli}(\pi_{ij})$ for $j = 1, \ldots, S-1$, and $b_{iS} = 1 - \mathbf{1}(z_{y_i} = S)$, which means we let $b_{ij} = 1$ if $j < z_{y_i}$, let $b_{ij} = 0$ if $j = z_{y_i}$, draw b_{ij} from Bernoulli(π_{ij}) if $z_{y_i} < j < S$, and let $b_{iS} = 0$ if and only if $z_{y_i} = S$.

Since stick-breaking multinomial classification is not invariant to the permutation of its class labels, it may perform substantially worse than it could be if the inherent geometric constraints implied by the current ordering of the labels make it difficult to adapt the decision boundaries to the data. Our solution to this problem is to infer the one-to-one mapping between the category labels and stick indices from the data. We construct a Metropolis-Hastings (MH) step within each Gibbs sampling iteration, with a proposal of switching two sticks that categories c and c', $1 \leq c < c' \leq S$, are mapped to, by changing the current category-stick one-to-one mapping from $\boldsymbol{z} = (z_1, \ldots, z_c, \ldots, z_{c'}, \ldots, z_S)$ to $\boldsymbol{z}' = (z_1', \ldots, z_S') := (z_1, \ldots, z_{c'}, \ldots, z_c, \ldots, z_S)$. Assuming a uniform prior on $\boldsymbol{z}$ and proposing (c, c') uniformly at random from one of the $\binom{S}{2} = S(S-1)/2$ possibilities, we would accept the proposal with probability

$$\min\left\{ \prod_i \frac{\prod_{s=1}^{S} [p_{is}(\boldsymbol{z}')]^{\mathbf{1}(y_i=s)}}{\prod_{s=1}^{S} [p_{is}(\boldsymbol{z})]^{\mathbf{1}(y_i=s)}},\ 1 \right\} = \min\left\{ \prod_i \frac{\prod_{s=1}^{S} \left[(\pi_{iz_s'})^{\mathbf{1}(z_s' \neq S)} \prod_{j<z_s'} (1 - \pi_{ij}) \right]^{\mathbf{1}(y_i=s)}}{\prod_{s=1}^{S} \left[(\pi_{iz_s})^{\mathbf{1}(z_s \neq S)} \prod_{j<z_s} (1 - \pi_{ij}) \right]^{\mathbf{1}(y_i=s)}},\ 1 \right\}. \qquad (8)$$

3.3 Sequential Decision Making

Random utility models, including both the logit and probit models as special examples, are widely used to infer the functional relationship between a categorical response variable and its covariates. For discrete choice analysis in econometrics (Hanemann, 1984; Greene, 2003; Train, 2009), these models assume that among a set of S alternatives, an individual makes the choice that maximizes his/her utility $U_{is} = V_{is} + \varepsilon_{is}$, where V_{is} and ε_{is} represent the observable and unobservable parts of U_{is}, respectively. If V_{is} is set as $V_{is} = \boldsymbol{x}_i'\boldsymbol{\beta}_s$, then marginalizing out $\boldsymbol{\varepsilon}_i = (\varepsilon_{i1}, \ldots, \varepsilon_{iS})'$ leads to MLR if all ε_{is} follow the extreme value distribution (McFadden, 1973; Greene, 2003; Train, 2009), and multinomial probit regression if all $\boldsymbol{\varepsilon}_i$ follow a multivariate normal distribution (Albert and Chib, 1993; McCulloch and Rossi, 1994; McCulloch et al., 2000; Imai and van Dyk, 2005).

Instead of examining the utilities of all choices before making the decision, the paSB construction is characterized by a sequential decision making process, described as follows. In step one, an individual decides whether to select the choice mapped to stick 1, or to select a choice among the remaining alternatives, *i.e.*, choices $\{s : z_s \in \{2, \ldots, S\}\}$. If the individual selects the choice mapped to stick 1, then the sequential process is terminated. Otherwise this choice is eliminated and the individual proceeds to step two, in which he/she would follow the same procedure to either select the choice mapped to stick 2 or proceed to the next step to select a choice among the remaining alternatives, *i.e.*, choices $\{s : z_s \in \{3, \ldots, S\}\}$. The individual, reconsidering none of the eliminated choices, will keep making a *one-vs-remaining* decision at each step until the termination of the sequential decision making process.

This unique sequential decision making procedure relaxes the independence of irrelevant alternatives (IIA) assumption, as described in the following Lemma.

Lemma 3 *Under the paSB construction, the probability ratio of two choices are influenced by the success probabilities of the sticks that lie between these two choices' corresponding sticks. In other words, the probability ratio of two choices will be influenced by some other choices if they are not mapped to adjacent sticks.*

As in Lemma 3, the paSB construction could adjust how two choices' probability ratio depends on the other alternatives by controlling the distance between the two sticks that they are mapped to, and hence provide a unique way to relax the IIA assumption. While the widely used MLR can be considered as a random-utility-maximization model with the IIA assumption, the paSB multinomial logistic model performs sequential random utility maximization that relaxes this assumption, as described in Lemma 5 in the Appendix.

4. Bayesian Multinomial Softplus Regression

Logistic regression is a cross-entropy-loss binary classifier that can be straightforwardly extended to paSB multinomial logistic regression (paSB-MLR). However, it is a linear classifier that uses a single hyperplane to separate one class from the other. To introduce nonlinear classification decision boundaries, we consider extending softplus regression of Zhou (2016), a multi-hyperplane binary classifier that uses the cross entropy loss, into multinomial softplus regression (MSR) under paSB.

Softplus regression uses the interaction of multiple hyperplanes to construct a union of convex-polytope-like confined spaces to enclose the data labeled as "1," which are hence separated from the data labeled as "0". It is constructed under a Bernoulli-Poisson link (Zhou, 2015) that thresholds at one a latent Poisson count, with the distribution of the Poisson rate defined as the convolution of the probability density functions of K experts, each of which corresponds to the stack of T gamma distributions with covariate-dependent scale parameters. The number of experts K and the number of layers T can be considered as the two model parameters that determine the nonlinear capacity of the model. More specifically, for expert k, denoting r_k as its weight and $\boldsymbol{\beta}_k^{t+1}$ as its tth regression-coefficient vector, the conditional class probability can be expressed as

$$P(y_i = 1 \,|\, \boldsymbol{x}_i, \{r_k, \{\boldsymbol{\beta}_k^{(t+1)}\}_{1,T}\}_{1,K}) = 1 - \prod_{k=1}^{K}(1 - p_{ik}),$$
$$p_{ik} = 1 - \left(1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(T+1)}} \ln\left\{1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(T)}} \ln\left[1 + \ldots \ln\left(1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(2)}}\right)\right]\right\}\right)^{-r_k};$$

when $K = T = 1$, the conditional class probability reduces to

$$P(y_i = 1 \,|\, \boldsymbol{x}_i, r, \boldsymbol{\beta}) = 1 - \left(\frac{1}{1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}}}\right)^{r};$$

and when $K = T = r = 1$, it becomes the same as that of binary logistic regression. Note that a gamma process, a random draw from which is expressed as $G = \sum_{k=1}^{\infty} r_k \delta_{\{\boldsymbol{\beta}_k^{t+1}\}_{1,T}}$, can be used to support a potentially countably infinite number of experts for softplus regression. For this reason, one can set K as large as permitted by computation and relies on the gamma process's inherent shrinkage mechanism to turn off unneeded model capacity (not all K experts will be used if K is set to be sufficiently large).

4.1 paSB and parSB Extensions of Softplus Regressions

We first follow Zhou (2016) to define

$$\varsigma(x_1, \ldots, x_t) = \ln\left(1 + e^{x_t} \ln\left\{1 + e^{x_{t-1}} \ln\left[1 + \ldots \ln\left(1 + e^{x_1}\right)\right]\right\}\right)$$

as the stack-softplus function. Note that if $t = 1$, the stack-softplus function reduces to softplus function $\varsigma(x) = \ln(1 + e^x)$, which is often considered as a smoothed version of the rectifier function, expressed as $\text{rectifier}(x) = \max(0, x)$, that has become the dominant nonlinear activation function for deep neural networks (Nair and Hinton, 2010; Glorot et al., 2011; Krizhevsky et al., 2012; LeCun et al., 2015). We then parameterize $\lambda_{iz_s} = -\ln(1 - \pi_{iz_s})$, the negative logarithms of the failure probabilities of the stick that category s is mapped to, as

$$\lambda_{iz_s} = \sum_{k=1}^{\infty} r_{sk}\, \varsigma\left(\boldsymbol{x}'\boldsymbol{\beta}_{sk}^{(2)}, \ldots, \boldsymbol{x}'\boldsymbol{\beta}_{sk}^{(T+1)}\right), \tag{9}$$

where the countably infinite atoms $(\boldsymbol{\beta}_{sk}^{(2)}, \ldots, \boldsymbol{\beta}_{sk}^{(T+1)})$ and their weights $\{r_{sk}\}_k$ constitute a draw from a gamma process $G_s \sim \text{GaP}(G_0, 1/c_s)$ (Ferguson, 1973), with G_0 as a finite and

continuous base distribution over a complete separable metric space Ω and $1/c_s$ as a scale parameter. In other words, we let $b_{iz_s} \sim \text{Bernoulli}(\pi_{iz_s})$ or

$$b_{iz_s} \sim \text{Bernoulli}\left[1-\prod_{k=1}^{\infty}\left(1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}_{sk}^{(T+1)}}\ln\Big\{1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}_{sk}^{(T)}}\ln\Big[1+\ldots\ln\Big(1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}_{sk}^{(2)}}\Big)\Big]\Big\}\right)^{-r_{sk}}\right]. \tag{10}$$

As shown in Theorem 10 of Zhou (2016), b_{iz_s} can be equivalently generated from a hierarchical model that convolves countably infinite stacked gamma distributions, with covariate-dependent scales, as

$$\begin{aligned}
\theta_{isk}^{(T)} &\sim \text{Gamma}\left(r_{sk}, e^{\boldsymbol{x}_i'\boldsymbol{\beta}_{sk}^{(T+1)}}\right),\\
&\cdots\\
\theta_{isk}^{(t)} &\sim \text{Gamma}\left(\theta_{isk}^{(t+1)}, e^{\boldsymbol{x}_i'\boldsymbol{\beta}_{sk}^{(t+1)}}\right),\\
&\cdots\\
\theta_{isk}^{(1)} &\sim \text{Gamma}\left(\theta_{isk}^{(2)}, e^{\boldsymbol{x}_i'\boldsymbol{\beta}_{sk}^{(2)}}\right),\\
b_{iz_s} = \mathbf{1}(m_{is}\geq 1),\ m_{is} &= \sum_{k=1}^{\infty} m_{isk}^{(1)},\ m_{isk}^{(1)} \sim \text{Pois}(\theta_{isk}^{(1)}),
\end{aligned} \tag{11}$$

the marginalization of whose latent variables lead to (10). Note the gamma distribution $\theta \sim \text{Gamma}(r, 1/c)$ is defined such that $\mathbb{E}[\theta] = r/c$ and $\text{var}[\theta] = r/c^2$, and the hierarchical structure in (11) can also be related to the augmentable gamma belief network proposed in Zhou et al. (2016). We consider the combination of (11) and either paSB in (5) or parSB in (7) as the Bayesian nonparametric hierarchical model for multinomial softplus regression (MSR) that is defined below.

Definition 1 (Multinomial Softplus Regression) *With a draw from a gamma process for each category that consists of countably infinite atoms $\boldsymbol{\beta}_{sk}^{(2:T+1)}$ with weights $r_{sk} > 0$, where $\boldsymbol{\beta}_{sk}^{(t)} \in \mathbb{R}^{P+1}$, given the covariate vector $\boldsymbol{x}_i$ and category-stick mapping $\boldsymbol{z}$, MSR parameterizes p_{is}, the multinomial probability of category s, under the paSB construction as*

$$\begin{aligned}
p_{is}(\boldsymbol{z}) &= \left[1-\textstyle\prod_{k=1}^{\infty}\left(1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}_{sk}^{(T+1)}}\ln\Big\{1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}_{sk}^{(T)}}\ln\Big[1+\ldots\ln\Big(1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}_{sk}^{(2)}}\Big)\Big]\Big\}\right)^{-r_{sk}}\right]^{\mathbf{1}(z_s\neq S)}\\
&\times\textstyle\prod_{j:z_j<z_s}\left[\prod_{k=1}^{\infty}\left(1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}_{jk}^{(T+1)}}\ln\Big\{1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}_{jk}^{(T)}}\ln\Big[1+\ldots\ln\Big(1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}_{jk}^{(2)}}\Big)\Big]\Big\}\right)^{-r_{jk}}\right],
\end{aligned}$$

and parameterizes p_{is} under the parSB construction as

$$\begin{aligned}
p_{is}(\boldsymbol{z}) &= \left[\prod_{k=1}^{\infty}\left(1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}_{sk}^{(T+1)}}\ln\Big\{1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}_{sk}^{(T)}}\ln\Big[1+\ldots\ln\Big(1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}_{sk}^{(2)}}\Big)\Big]\Big\}\right)^{-r_{sk}}\right]^{\mathbf{1}(z_s\neq S)}\\
&\times\textstyle\prod_{j:z_j<z_s}\left[1-\prod_{k=1}^{\infty}\left(1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}_{jk}^{(T+1)}}\ln\Big\{1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}_{jk}^{(T)}}\ln\Big[1+\ldots\ln\Big(1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}_{jk}^{(2)}}\Big)\Big]\Big\}\right)^{-r_{jk}}\right].
\end{aligned}$$

For the convenience of implementation, we truncate the number of atoms of the gamma process at K by choosing a discrete base measure for each category as $G_{s0} = \sum_{k=1}^{K}\frac{\gamma_{s0}}{K}\delta_{\boldsymbol{\beta}_{sk}^{(2:T+1)}}$, under which we have $r_{sk} \sim \text{Gamma}(\gamma_{s0}/K, 1/c_{s0})$ as the prior distribution for the weight of expert k in category s. For each category, we expect only some of its K experts to have non-negligible weights if K is set large enough, and we may use $\sum_k \mathbf{1}\big(\sum_i m_{isk}^{(1)} > 0\big)$, where $m_{isk}^{(1)}$ is defined in (11), to measure the number of active experts inferred from the data.

4.2 Geometric Constraints for MSR

Since by definition we have $p_{is}(\boldsymbol{z}) = \pi_{iz_s}\big(1-\sum_{j<s}p_{is}(\boldsymbol{z})\big) = \pi_{iz_s}\prod_{j<z_s}(1-\pi_{ij})$ in MSR, it is clear that if π_{ij} for all $j < z_s$ are small and π_{iz_s} is the first one to have a large probability value close to one, y_i will be likely assigned to category s regardless of how large the values of $\{\pi_{ij}\}_{j>z_s}$ are. To motivate the use of the seemingly over-parameterized sum-stack-softplus function in (9), we first consider the simplest case of $K = T = 1$. Without loss of generality, let us assume that the category-stick mapping is fixed at $\boldsymbol{z} = (1, \ldots, S)$.

Lemma 4 *For paSB-MSR with $K = T = 1$ and $\boldsymbol{z} = (1, \ldots, S)$, the set of solutions to $p_{is}(\boldsymbol{z}) > p_0$ in the covariate space are bounded by a convex polytope defined by the intersection of s linear hyperplanes.*

Note that the binary softplus regression with $K = T = 1$ is closely related to logistic regression, and reduces to logistic regression if $r = 1$ (Zhou, 2016). With Lemma 4, it is clear that even if an optimal category-stick mapping $\boldsymbol{z}$ is provided, paSB-MSR with $K = T = 1$ may still clearly underperform MLR. This is because category s uses a single hyperplane to separate itself from the remaining $S - s$ categories, and hence uses the interaction of at most s hyperplanes to separate itself from the other $S - 1$ categories. By contrast, MLR uses a convex polytope bounded by at most $S - 1$ hyperplanes for each of the S categories.

When $K > 1$ and/or $T > 1$, an exact theoretical analysis is beyond the scope of this paper. Instead we provide some qualitative analysis by borrowing related geometric-constraint analysis for softplus regressions in Zhou (2016). Note that Equation (10) indicates that a noisy-or model (Pearl, 2014; Srinivas, 1993), commonly appearing in causal inference, is used at each step of the sequential one-vs-remaining decision process; at each step, the binary outcome of an observation is attributed to the disjunctive interaction of many possible hidden causes. Roughly speaking, to enclose category s to separate it from the remaining $S-s$ categories in the covariate space, paSB-MSR with $K > 1$ and $T = 1$ uses the complement of a convex-polytope-bounded space, paSB-MSR with $K = 1$ and $T > 1$ uses a convex-polytope-like confined space, and paSB-MSR with both $K > 1$ and $T > 1$ uses a union of convex-polytope-like confined spaces. For parSB-MSR with $K + T > 1$, the interpretation is the same except a convex polytope in paSB will be replaced with the complement of a convex polytope, and vise versa. In contrast to SVMs using the kernel trick, MSRs using the original covariates might be more appealing in research areas, like biostatistics and sociology, where the interpretation of regression coefficients and investigation of causal relationships are of interest. In addition, we find that the classification capability of MSRs could be further enhanced with data transformation, as will be discussed in Section 6.4.

5. Bayesian Multinomial Support Vector Machine

Support vector machines (SVMs) are max-margin binary classifiers that typically minimize a regularized hinge loss objective function as

$$l(\boldsymbol{\beta}, \nu) = \sum_{i=1}^{N} \max(1 - b_i \boldsymbol{x}_i'\boldsymbol{\beta}, 0) + \nu R(\boldsymbol{\beta}),$$

where $b_i \in \{-1, 1\}$ represents the binary label for the ith observation, $R(\boldsymbol{\beta})$ is a regularization function that is often set as the L_1 or L_2 norm of $\boldsymbol{\beta}$, ν is a tuning parameter, and $\boldsymbol{x}_i'$ is the ith row of the design matrix $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)'$. For linear SVMs, $\boldsymbol{x}_i$ is the covariate vector of the ith observation, whereas for nonlinear SVMs, one typically set the (i, j)th element of $\mathbf{X}$ as the kernel distance between the covariate vector of the ith observation and the jth support vector. The decision boundary of a binary SVM is $\{\boldsymbol{x} : \boldsymbol{x}'\boldsymbol{\beta} = 0\}$ and an observation is assigned the label $y_i = \text{sign}(\boldsymbol{x}'\boldsymbol{\beta})$, which means $b_i = 1$ if $\boldsymbol{x}'\boldsymbol{\beta} \geq 0$ and $b_i = -1$ if $\boldsymbol{x}'\boldsymbol{\beta} < 0$.

5.1 Bayesian Binary SVMs

It is shown in Polson and Scott (2011) that the exponential of the negative of the hinge loss can be expressed as a location-scale mixture of normals as

$$\begin{aligned} L(b_i \,|\, \boldsymbol{x}_i, \boldsymbol{\beta}) &= \exp\left[-2\max(1 - b_i \boldsymbol{x}_i'\boldsymbol{\beta}, 0)\right] \\ &= \int_0^\infty \frac{1}{\sqrt{2\pi\omega_i}} \exp\left[-\frac{1}{2}\frac{(1 + \omega_i - b_i \boldsymbol{x}_i'\boldsymbol{\beta})^2}{\omega_i}\right] d\omega_i. \end{aligned}$$

Consequently, $L(\boldsymbol{b} \,|\, \mathbf{X}, \boldsymbol{\beta}) = \prod_i L(b_i \,|\, \boldsymbol{x}_i, \boldsymbol{\beta}) = \exp\{-2\sum_i \max(1 - b_i \boldsymbol{x}_i'\boldsymbol{\beta}, 0)\}$ can be regarded as a pseudo likelihood in the sense that it is unnormalized with respect to $\boldsymbol{b} = (b_1, \ldots, b_N)' \in \{-1, 1\}^N$. This location-scale normal mixture representation of the hinge loss allows developing close-form Gibbs sampling update equations for the regression coefficients $\boldsymbol{\beta}$ via data augmentation, as discussed in detail in Polson and Scott (2011) and further generalized in Henao et al. (2014) to construct nonlinear SVMs amenable to Bayesian inference. While data augmentation has made it feasible to develop Bayesian inference for SVMs, it has not addressed a common issue that SVMs provide the predictions of deterministic class labels but not class probabilities. For this reason, below we discuss how to allow SVMs to predict class probabilities while maintaining tractable Bayesian inference via data augmentation.

Following Sollich (2002) and Mallick et al. (2005), by defining the joint distribution of $\boldsymbol{\beta}$ and $\{\boldsymbol{x}_i\}_i$ to be proportional to $\prod_i [L(1 \,|\, \boldsymbol{x}_i, \boldsymbol{\beta}) + L(-1 \,|\, \boldsymbol{x}_i, \boldsymbol{\beta})]$, one may define the conditional distribution of the binary label $b_i \in \{-1, 1\}$ as

$$P(b_i \,|\, \boldsymbol{x}_i, \boldsymbol{\beta}) = \begin{cases} \dfrac{1}{1 + e^{-2b_i \boldsymbol{x}_i'\boldsymbol{\beta}}}, & \text{for } |\boldsymbol{x}_i'\boldsymbol{\beta}| \leq 1; \\ \dfrac{1}{1 + e^{-b_i[\boldsymbol{x}_i'\boldsymbol{\beta} + \text{sign}(\boldsymbol{x}_i'\boldsymbol{\beta})]}}, & \text{for } |\boldsymbol{x}_i'\boldsymbol{\beta}| > 1; \end{cases} \tag{12}$$

which defines a probabilistic inference model that has the same maximum a posteriori (MAP) solution as that of a binary SVM for a given data set. Note that for MAP inference, the penalty term $\nu R(\boldsymbol{\beta})$ of the regularized hinge loss can be related to a corresponding prior distribution imposed on $\boldsymbol{\beta}$, such as Gaussian, Laplace, and spike-and-slab priors (Polson and Scott, 2011).

5.2 paSB Multinomial Support Vector Machine

Generalizing previous work in constructing Bayesian binary SVMs, we propose multinomial SVM (MSVM) under the paSB framework that is distinct from previously proposed MSVMs (Crammer and Singer, 2002; Lee et al., 2004; Liu and Yuan, 2011). A Bayesian MSVM that predicts class probabilities has also been proposed before in Zhang and Jordan (2006), which, however, does not have a data augmentation scheme to sample the regression coefficients in closed form, and consequently, relies on a random-walk Metropolis-Hastings procedure that may be difficult to tune.

Redefining the label sample space from $b_i \in \{-1, 1\}$ to $b_i \in \{0, 1\}$, we may rewrite (12) as $b_i \,|\, \boldsymbol{x}_i, \boldsymbol{\beta} \sim \text{Bernoulli}[\pi_{i,\text{svm}}(\boldsymbol{x}_i, \boldsymbol{\beta})]$, where

$$\pi_{i,\text{svm}}(\boldsymbol{x}_i, \boldsymbol{\beta}) = \begin{cases} \dfrac{1}{1 + e^{-2\boldsymbol{x}_i\boldsymbol{\beta}}}, & \text{for } |\boldsymbol{x}_i'\boldsymbol{\beta}| \leq 1; \\ \dfrac{1}{1 + e^{-\boldsymbol{x}_i\boldsymbol{\beta} - \text{sign}(\boldsymbol{x}_i'\boldsymbol{\beta})}}, & \text{for } |\boldsymbol{x}_i'\boldsymbol{\beta}| > 1. \end{cases} \tag{13}$$

The Bernoulli likelihood based cross-entropy-loss binary classifier, whose covariate-dependent probabilities are parameterized as in (13), is exactly what we need to extend the binary SVM into a multinomial classifier under paSB introduced in Theorem 1. More specifically, given the category-stick mapping $\boldsymbol{z}$, with the success probabilities of the stick that category s is mapped to parameterized as $\pi_{iz_s,\text{svm}}(\boldsymbol{x}_i, \boldsymbol{\beta}_s)$ and binary stick variables drawn as $b_{iz_s} \sim \text{Bernoulli}[\pi_{iz_s,\text{svm}}(\boldsymbol{x}_i, \boldsymbol{\beta}_s)]$, we have the following definition.

Definition 2 (paSB multinomial SVM) *Under the paSB construction, given the covariate vector $\boldsymbol{x}_i$ and category-stick mapping $\boldsymbol{z}$, multinomial support vector machine (MSVM) parameterizes p_{is}, the multinomial probability of category s, as*

$$p_{is}(\boldsymbol{z}) = [\pi_{iz_s,\text{svm}}(\boldsymbol{x}_i, \boldsymbol{\beta}_s)]^{\mathbf{1}(z_s \neq S)} \prod\nolimits_{j:z_j < z_s} \pi_{iz_j,\text{svm}}(\boldsymbol{x}_i, \boldsymbol{\beta}_j).$$

Note that there is no need to introduce parSB-MSVM in addition to paSB-MSVM, since by definition, we have $\pi_{iz_s,\text{svm}}(\boldsymbol{x}_i, -\boldsymbol{\beta}_s) = 1 - \pi_{iz_s,\text{svm}}(\boldsymbol{x}_i, \boldsymbol{\beta}_s)$ for all s.

6. Example Results

Constructed under the paSB framework, a multinomial regression model of S categories is characterized by not only how the S stick-specific binary classifiers with cross entropy loss parameterize their covariate-dependent probability parameters, but also how its S categories are one-to-one mapped to S latent sticks. To investigate the unique properties of a paSB multinomial regression model, we will study the benefits of both inferring an appropriate mapping $\boldsymbol{z}$ and increasing the modeling capacity of the underlying binary regression model. For illustration purpose, we will focus on multinomial softplus regression (MSR) whose capacity and complexity are both explicitly controlled by K and T.

6.1 Influence of Binary Regression Model Capacity

We first consider the Iris data set with $S = 3$ categories. We choose the sepal and petal lengths as the two dimensional covariates to illustrate the performance of MSR under four different settings. We fix $\boldsymbol{z} = (1, 2, 3)$, which means category s is mapped to stick s for all s, but choose different model capacities by varying K and T.

Examining the relative 2D spatial locations of the observations, where the blue, black, and gray points are labeled as category 1, 2, and 3, respectively, one can imagine that setting $\boldsymbol{z} = (2, 1, 3)$, which means mappings categories 2, 1, and 3 to the 1st, 2nd, and 3rd sticks, respectively, will already lead to excellent class separations for MSR with $K = T = 1$, according to the analysis in Section 4.2 and also confirmed by our experimental results (not shown for brevity). More specifically, with the 2nd, 1st, and 3rd categories mapped to the 1st, 2nd, and 3rd sticks, respectively, one can first use a single hyperplane to separate category 2 (black points) from both categories 1 (blue points) and 3 (gray points), and then use another hyperplane to separate category 1 (blue points) from category 3 (gray points).

However, when the mapping is fixed at $\boldsymbol{z} = (1, 2, 3)$, as shown in the first row of Figure 1, MSR with $K = T = 1$ performs poorly and fails to separate out category 1 (blue points) right in the beginning. This is not surprising since MSR with $K = T = 1$ is only equipped with a single hyperplane to separate the category that the first stick is mapped to (category $z_1 = 1$ in this case) from the others, whereas for this data set it is apparent at least two hyperplanes are required to separate the blue from the black and gray points. MSR with $K = 5$ and $T = 1$ also fails to work with $\boldsymbol{z} = (1, 2, 3)$, as shown in the third row of Figure 1, which is also not surprising since it can only use the complementary of a convex-polytope-bound confined space to enclose category $z_1 = 1$, but the blue points can not be enclosed in such a manner. Despite purposely enforcing an unfavorable category-stick mapping, once we increase T, the performance quickly improves, which is expected since $T > 1$ allows using a single (if $K = 1$ as in the second row) or a union (if $K > 1$ as in the fourth row) of convex-polytope-like confined spaces to separate one category from the others (by enclosing the positively labeled observations in each stick-specific binary classification task).

The results in Figure 1 show that even an unoptimized category-stick mapping, which is unfavorable to MSR with small K and/or T, is enforced, empowering each stick-specific binary regression model with a higher capacity (using larger K and/or T) can still allow MSR to achieve excellent separations. It is also simple to show that for the data set in Figure 1, even if one chooses low-capacity stick-specific binary regression models by setting $T = 1$, one can still achieve good performance with MSR if the category-stick mapping is set as $\boldsymbol{z} = (2, 1, 3)$, $\boldsymbol{z} = (3, 1, 2)$, $\boldsymbol{z} = (2, 3, 1)$, or $\boldsymbol{z} = (3, 2, 1)$. That is to say, as long as it is not category 1 (blue points) that is mapped to stick 1, MSR with $T = 1$ is able to provide satisfactory performance.

6.2 Influence of Category-Stick Mapping and its Inference

The Iris data set in Figure 1 provides an instructive example to show not only the importance of increasing the model capacity if a poor category-stick mapping is imposed, but also the importance of optimizing the category-stick mapping if the capacities of these stick-specific binary regression models are limited. To further illustrate the benefits of inferring an appropriate category-stick mapping $\boldsymbol{z}$, we consider the *square* data set shown in Figure 2.

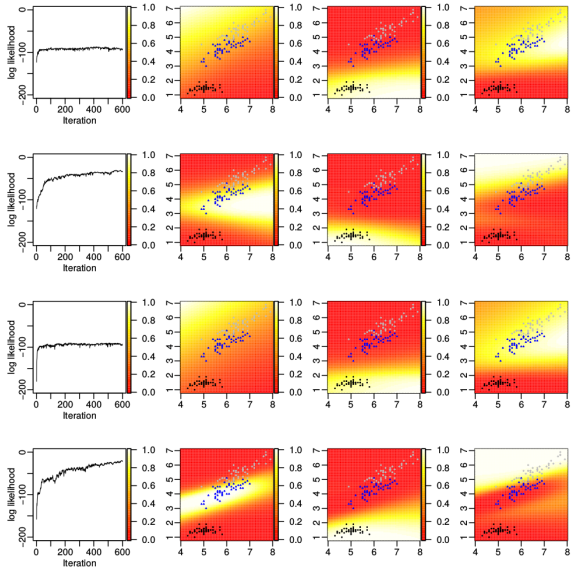


Figure 1: Log-likelihood plots and predictive probability heat maps for the 2-D iris data with a fixed category-stick mapping $\boldsymbol{z} = (1, 2, 3)$. Blue, black, and gray points are labeled as categories 1, 2, and 3, respectively. For the first row, $K = 1$ and $T = 1$, second row, $K = 1$ and $T = 3$, third row, $K = 5$ and $T = 1$, and fourth row, $K = 5$ and $T = 3$. The log-likelihood plots are shown in Column 1, and the predictive probability heat maps of categories 1 (blue), 2 (black), and 3 (gray) are shown in Columns 2, 3, and 4, respectively.

We show that for MSR, even if both K and T are sufficiently large to allow each stick-specific binary regression model to have a high enough capacity, whether an optimal category-stick mapping is selected may still clearly matter for the performance.

As shown in the first three rows of Figure 2, with $K = T = 10$, three different $\boldsymbol{z}$'s are considered and $\boldsymbol{z} = (1, 2, 3)$ (shown in the first row) is found to perform the best. As shown in the fourth row, we sample $\boldsymbol{z}$ using (8) within each MCMC iteration and achieve a result that seems as good as fixing $\boldsymbol{z} = (1, 2, 3)$. In fact, we find that our inferred mappings switch between $\boldsymbol{z} = (1, 2, 3)$ and $\boldsymbol{z} = (1, 3, 2)$ during MCMC iterations, indicating that the Markov chain is mixing well. These results suggest the importance of both learning the mapping $\boldsymbol{z}$ from the data and allowing the stick-specific binary classifiers to have enough capacities to model nonlinear classification decision boundaries.

When sampling $\boldsymbol{z} = (z_1, \ldots, z_S)$ that the S categories are mapped to, although $S!$ permutations of $(1, \ldots, S)$ can become enormous as S increases, the effective search space could be much smaller if many different mappings imply similar likelihoods and if these extremely poor mappings can be easily avoided. Rather than searching for the best mapping,

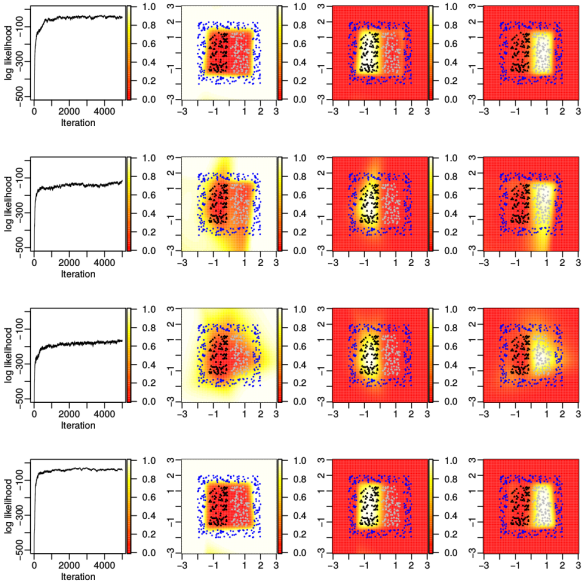


Figure 2: Log-likelihood plots and predictive probability heat maps for the square data with $K = T = 10$. The blue, black, and gray points are labeled as categories 1, 2, and 3, respectively. We fix the category-stick mapping as $\boldsymbol{z} = (1, 2, 3)$ for Row 1, $(2, 1, 3)$ for Row 2, and $(3, 1, 2)$ for Row 3, and sample $\boldsymbol{z}$ for Row 4. The log-likelihood plots are shown in Column 1, and the predictive probability heat maps of categories 1 (blue), 2 (black), and 3 (gray) are shown in Columns 2, 3, and 4, respectively.

the proposed MH step, proposing two indices z_j and $z_{j'}$ to switch in each iteration, is a simple but effective strategy to escape from the mappings that lead to poor fits. Note that the probability of a z_j not being proposed to switch after t MCMC iterations is $[(S-2)/S]^t$. Even if S is as large as 100, this probability is less than 10^{-8} at $t = 1000$. Also note the iteration at which z_j is proposed to switch at the first time follows a geometric distribution, with success probability $2/S$. Thus $S/2$ is the expected number of iterations for a z_j to be proposed to switch once.

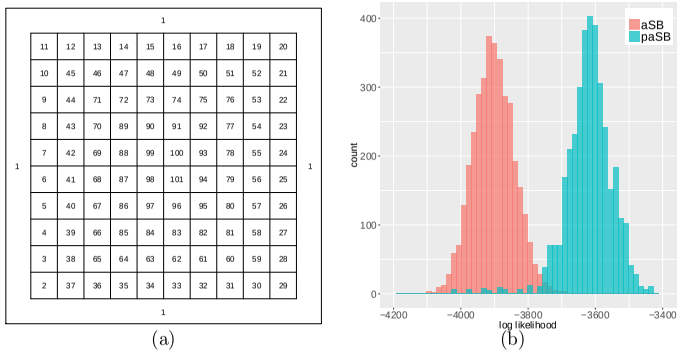
To demonstrate the efficiency of our permutation scheme, we construct square101, a synthetic two-dimensional data set consisting of 101 categories. We generate 8000 data points that are uniformly at random distributed within the 12×12 spatial region occupied by all 101 categories. The decision boundaries of different classes are displayed in Figure 3(a), where the data points placed within the outside square frame, whose outer and inner dimensions are 12 and 10, respectively, are assigned to category 1, and these placed within the sth unit square, where $s \in \{2, \ldots, 101\}$, inside the square frame are assigned to category s.

Although it is almost impossible to search for the best category-stick mapping $\boldsymbol{z}$ giving rise to the highest likelihood from all $101! \approx 10^{160}$ possible mappings, we show our permutation scheme is very effective in escaping from poor mappings, leading to a performance that is comparable to the best of those obtained with pre-fixed suboptimal mappings. More specifically, applying the analysis in Section 4.2 to Figure 3(a), we expect an aSB-MSR to perform well under a fixed suboptimal category-stick mapping $\boldsymbol{z}$, where $z_1 = 1$, which means the outside square frame is mapped to stick 1, and the squares closer to the inner boundary of the square frame are mapped to the sticks broken at earlier stages; the mapping $\boldsymbol{z} = (1, 2, \cdots, 101)$ is such an example. In other words, we first separate the frame from all the other squares, and then sequentially separate the squares from the remainders; the closer a square is from the frame, the earlier it is separated. The total number of suboptimal mappings $\boldsymbol{z}$'s constructed in this manner is as large as $36! \times 28! \times 20! \times 12! \times 4! \approx 10^{99.5}$.

First, we uniformly at random generate 3600 different suboptimal mappings $\boldsymbol{z}$'s under this construction, run aSB-MSR with $K = T = 4$, and plot the histogram of the 3600 log-likelihoods in Figure 3(b). Second, we start from 3600 randomly initialized $\boldsymbol{z}$, run paSB-MSR with $K = T = 4$, and also plot the histogram of the 3600 log-likelihoods in Figure 3(b). For each run, we choose 20,000 MCMC iterations and collect the last 1000 MCMC samples. Each log-likelihood is averaged over those of the corresponding model's collected MCMC samples. As in Figure 3(b), the log-likelihood from a paSB-MSR is in general clearly larger than that of an aSB-MSR with a fixed suboptimal $\boldsymbol{z}$, and there is little overlap between their corresponding histograms. Further examining the 3600 $\boldsymbol{z}$'s inferred by paSB at its last MCMC iteration shows that 3482 of them have $z_1 = 1$ and all of them have $z_1 \leq 5$. Suppose $z_1 \notin \{1, 2, 3, 4, 5\}$ at the current iteration, which means category 1 is mapped to none of the first five sticks, then the probability of not only selecting stick z_1, but also switching it with one of the first five sticks in the MH proposal is $\frac{1}{101} \times \frac{5}{100}$. Thus the probability that category 1 has never been proposed to mapped to one of the first five sticks after t iterations is $[1 - 5/(101 \times 100)]^t$, which becomes as small as 0.005% at $t = 20,000$, demonstrating the effectiveness of our permutation scheme in dealing with a large number of categories. Note we have also tried 3600 aSB-MSR, each of which is provided with a randomly initialized $\boldsymbol{z}$. The log-likelihoods, however, are all far below -4000 and hence not included for comparison. This phenomenon is not surprising, as the probability for a randomly initialized $\boldsymbol{z}$ to be suboptimal is as tiny as $36! \times 28! \times 20! \times 12! \times 4!/101! \approx 10^{-60.5}$.

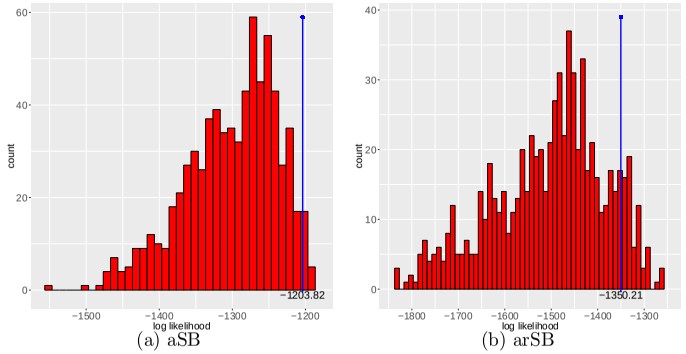
Figure 4 empirically demonstrates the effectiveness of permuting $\boldsymbol{z}$ on the satimage data set, using MSRs with $K = 5$, $T = 3$, and $\boldsymbol{z}$ fixed at each of the $6! = 720$ possible one-to-one category-stick mappings. Panels (a) and (b) show the log-likelihood histograms for MSRs constructed under augmented SB (aSB) and augmented reversed SB (arSB), respectively. Both histograms are clearly left skewed, indicating under both aSB and arSB, only a small proportion of the 720 different category-stick mappings lead to very poor fits. The blue vertical lines at -1203.82 in (a) and -1350.21 in (b) are the log-likelihoods by paSB and parSB, respectively, in both of which the category-stick mapping $\boldsymbol{z}$ is updated by a MH step in each MCMC iteration. Only 20 (97) out of 720 aSB-MSRs (arSB-MSRs) have a higher likelihood than paSB-MSR (parSB-MSR).

Since in the stick-breaking construction, the binary classifier that separates a category mapped to a smaller-indexed stick from the others utilizes fewer constraints, the classification can be poor if the complexity of the decision boundary goes beyond the nonlinear



(a) (b)

Figure 3: (a) Illustration of the square101 data and (b) log-likelihood histograms, by aSB-MSR with 3600 random suboptimal category-stick mappings and by paSB-MSR with 3600 randomly initialized category-stick mappings.



(a) aSB (b) arSB

Figure 4: Log-likelihood histograms for MSRs using all 720 possible category-stick mappings, constructed under (a) augmented stick breaking (aSB) and (b) augmented and reversed stick breaking (arSB). The blue lines in (a) and (b) correspond to the log-likelihoods of paSB-MSR and parSB-MSR, respectively.

modeling capacity of the binary classifier. However, even with a low-capacity binary classifier, the performance could be significantly improved if that difficult-to-separate category is mapped to a larger-indexed stick, for which there are fewer categories left to be separated in its "one-vs-remaining" binary classification problem. Examining the $\boldsymbol{z}$'s associated with the 100 lowest log-likelihoods in Figure 4, we find there are 51 mappings belonging to the set $\{\boldsymbol{z} : z_5 = 1 \text{ or } z_6 = 1\}$ in aSB, and 77 belonging to $\{\boldsymbol{z} : z_3 = 1 \text{ or } z_6 = 1\}$ in arSB. It suggests that separating Categories 5 or 6 (Categories 3 or 6) from all the other categories might be beyond the capacity of a binary softplus regression with $K = 5$ and $T = 3$ under

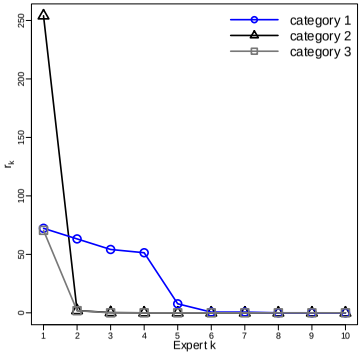


Figure 5: Inferred expert weights r_k in descending order for each category of the square data with $K = T = 10$.

the aSB (arSB) construction. But if breaking the sticks associated with these categories at late stages, we only need to separate them from fewer remaining categories, which could be much easier. We have further examined the other 620 arrangements, and found no evident patterns. These observations suggest that the effective search space of the mapping $\boldsymbol{z}$ is considerably smaller than $S!$, and the proposed MH step is effective in escaping from poor category-stick mappings.

In paSB-MSVM, we use a Gaussian radial basis function kernel, whose kernel width is cross validated from a set of predefined candidates. We find its performance to be sensitive to the setting of the kernel width, which is a common issue for SVMs (Cherkassky and Ma, 2004; Soares et al., 2004; Chang et al., 2005). If an appropriate kernel width could be identified through cross validation, we find that learning the mapping $\boldsymbol{z}$ becomes less important for paSB-MSVM to perform well. However, we find that if the kernel width is not well selected, which can happen if all candidate kernel widths are far from the optimal value, the binary classifier for each category may not have enough capacity for nonlinear classification and the learning of the category-stick mapping $\boldsymbol{z}$ could then become important.

6.3 Turning Off Unneeded Model Capacities

While one can adjust both K and T to control the capacity of binary softplus regression, for MSR, the total number of experts K is a truncation level that can be set as large as permitted by the computation budget. This is because the truncated gamma process used by each stick-specific binary softplus regression shrinks the weights of unnecessary experts towards zeros. Figure 5 shows in decreasing order the inferred weights of the experts belonging to each of the 3 categories of the square data set. These weights are inferred by MSR with $K = T = 10$ and the learning of $\boldsymbol{z}$, as in the fourth row of Figure 2. It is clear from Figure 5 that only a small number of experts are inferred with non-negligible weights in the posterior, and the number of active experts and their weights indicate the complexity

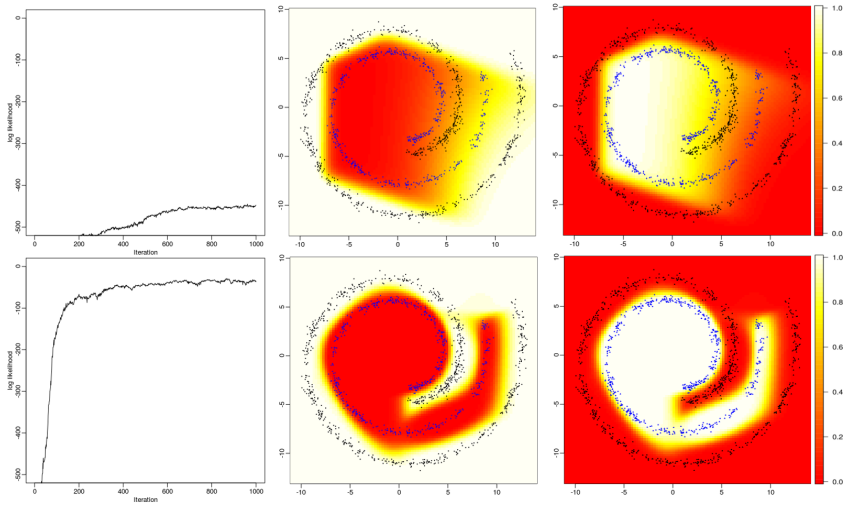


Figure 6: First row: classification of a 2-D swiss roll data by paSB-MSR with $K = 5$, $T = 3$, using the original covariates. Second row: paSB-MSR with $K = 5$, $T = 1$ trained on the covariates transformed via the paSB-MSR used in the first row. In each row, the left column plot the log-likelihood against MCMC iteration, and the middle and right columns show the predictive probability heatmaps for Category 1 (black points) and Category 2 (blue points), respectively.

of the corresponding classification decision boundaries shown in the fourth row of Figure 2. We note that while T is a parameter to be set by the user, we find increasing it increases model capacity, without observing clear signs of overfitting for all the data considered here.

6.4 MSR with Data Transformation

Kernel SVMs transform the data to make different categories more linearly separable in the transformed covariate space. While kernel SVMs may provide high nonlinear modeling capacity, its performance could be sensitive to the kernel width, which often needs to be cross validated, and its number of support vectors often increases linearly in the size of the training set. By contrast, MSRs rely on the interactions of linear hyperplanes to construct nonlinear decision boundaries, as discussed in Section 4.2, and hence may have insufficient capacity for highly complex nonlinearity. However, we may simply stack another MSR on a previously trained MSR to quickly enhance its nonlinear modeling capacity. In particular, we may first run a MSR to obtain a finite set of hyperplanes denoted by $\tilde{\boldsymbol{\beta}}_{jk}^{(t+1)}$. We may

then augment the original covariate vector $\boldsymbol{x}_i$ as

$$\tilde{\boldsymbol{x}}_i := \left[\boldsymbol{x}_i', \log\left(1 + e^{\boldsymbol{x}_i'\tilde{\boldsymbol{\beta}}_{11}^{(2)}}\right), \cdots, \log\left(1 + e^{\boldsymbol{x}_i'\tilde{\boldsymbol{\beta}}_{jk}^{(t+1)}}\right), \cdots, \log\left(1 + e^{\boldsymbol{x}_i'\tilde{\boldsymbol{\beta}}_{SK}^{(T+1)}}\right)\right]' \tag{14}$$

and run another MSR with the transformed covariates $\tilde{\boldsymbol{x}}_i$.

For illustration, we show the efficacy of this data-transformation strategy on a 2-D swiss roll data in Figure 6. The first row shows the results of MSR with $K = 5$ and $T = 3$, using the original covariates $\boldsymbol{x}_i$, while the second row shows MSR with $K = 5$ and $T = 1$, using the transformed covariates $\tilde{\boldsymbol{x}}_i$ defined by (14), where the regression coefficient vectors $\tilde{\boldsymbol{\beta}}_{jk}^{(t+1)}$ are learned using the MSR illustrated in the first row. It is evident that the classification is greatly improved in terms of both training log-likelihood and out-of-sample predictions.

6.5 Results on Benchmark Data Sets

To further evaluate the performance of the proposed paSB multinomial regression models, we consider paSB multinomial logistic regression (paSB-MLR), paSB multinomial robit with $\kappa = 6$ degrees of freedom (paSB-robit), paSB multinomial support vector machine (paSB-MSVM), and MSRs. We compare their performance with those of L_2 regularized multinomial logistic regression (L_2-MLR), support vector machine (SVM), and adaptive multi-hyperplane machine (AMM), and consider the following benchmark multi-class classification data sets: iris, wine, glass, vehicle, waveform, segment, dna, and satimage. We also include the synthetic square data shown in Figure 2 for comparison. For SVM we use the LIBSVM package, which trains $S(S-1)/2$ one-vs-one binary classifiers and makes prediction using majority voting (Chang and Lin, 2011). We run LIBSVM in `R` with package `e1071` (Meyer et al., 2015). We consider MSRs with (K, T) as $(1, 1)$, $(1, 3)$, $(5, 1)$, and $(5, 3)$, respectively. We also consider MSR with data transformation (DT-MSR), in which we first train a MSR with $K = 5$ and $T = 3$ to transform the covariates and then stack another MSR with $K = 5$ and $T = 1$. We provide detailed descriptions on the data and experimental settings in the Appendix.

With the number of categories in parentheses right after the data set names, we summarize in Table 1 the classification error rates by various models, where those of MSRs are calculated by averaging over paSB and parSB. Table 1 shows that an MSR with K or T sufficiently large generally outperforms paSB-MLR, paSB-robit, L_2-MLR, and AMM, and using another MSR on the transformed covariates can in general further reduce the error rate. This is especially evident when there are nonlinearly separable categories, as indicated by a clearly higher error rate of L_2-MLR in contrast to that of SVM. One may notice that paSB-robit, paSB-MLR, and MSR with $K = T = 1$ are similar to L_2-MLR in terms of performance, suggesting the effectiveness of the proposed permutation scheme, which helps mitigate the potential adverse effects of having asymmetric class labels. One may also note that paSB-robit outperforms paSB-MLR on glass, vehicle, waveform, dna, and satimage, indicating there are benefits in using a robust classifier on these data sets. Comparable error rates of paSB-MSVM to SVM and better performance of MSRs on most data sets demonstrate the success of the paSB framework in transforming a binary classifier with cross entropy loss into a Bayesian multinomial one.

To further check whether a paSB model is attractive when fast out-of-sample prediction is desired, we consider using only the MCMC sample that has the highest training likelihood

Data (S)	paSB-MLR	paSB-robit	paSB-MSVM	$K=1$ $T=1$	$K=1$ $T=3$	$K=5$ $T=1$	$K=5$ $T=3$	DT-MSR	L_2-MLR	SVM	AMM
square(3)	59.52	67.46	**0**	57.14	15.08	**0**	**0**	**0**	62.29	4.76	16.67
iris(3)	4.00	5.33	**3.33**	4.67	4.00	4.00	4.00	**3.33**	**3.33**	4.00	4.67
wine(3)	4.44	5.00	2.78	2.78	**2.22**	2.78	**2.22**	2.78	3.89	2.78	3.89
glass(6)	35.35	34.88	29.30	33.49	26.05	31.16	32.09	**26.37**	33.02	28.84	37.67
vehicle(4)	23.23	21.25	17.32	22.44	17.32	17.72	15.75	**14.96**	22.83	18.50	21.89
waveform(4)	17.87	16.42	15.76	19.84	16.62	15.67	**15.04**	15.56	15.60	15.22	18.54
segment(7)	7.36	8.03	7.98	6.20	6.49	6.45	**5.63**	7.65	8.56	6.20	12.47
dna(3)	5.06	4.05	5.31	4.13	4.47	4.55	4.22	**3.88**	5.98	4.97	5.43
satimage(6)	20.65	17.25	8.90	16.65	14.45	12.85	12.00	9.85	17.80	**8.50**	15.31

Table 1: Comparison of the classification error rates (%) of paSB-MLR, paSB-robit, paSB-MSVM, MSRs with various K and T (columns 5 to 8), MSR with data transformation (DT-MSR), L_2-MLR, SVM, and AMM.

among the collected ones for all paSB models, and summarize in Table 4 of the Appendix the classification error rates of various models, with the number of inferred support vectors or active hyperplanes included in parenthesis. Following the definition of active experts in Zhou (2016), we define for MSRs the number of active hyperplanes as $T\sum_s^S \widetilde{K}_s$ where $\widetilde{K}_s$ is the number of active experts for class s. The number of active hyperplanes determines the computational complexity for out-of-sample prediction with a single MCMC sample, which is $O(T\sum_s^S \widetilde{K}_s)$. Since the error rates of MSRs in Table 4 are calculated by averaging over both paSB and parSB, the number of active hyperplanes is $T\sum_s^S(\widetilde{K}_s^{(paSB)} + \widetilde{K}_s^{(parSB)})$.

Shown in Figure 8 in the Appendix are boxplots of the number of each category's active experts for MSR with $K = 5$ and $T = 3$. Except for several categories of satimage that require all $K = 5$ experts for parSB-MSR, $K = 5$ is large enough to provide the needed model capacity under all the other scenarios. As shown in Table 4, MSRs with sufficiently large K and/or T are comparable to both SVM and paSB-MSVM in terms of the error rates, while clearly outperforming them in terms of the number of (active) hyperplanes/support vectors and hence computational complexity for out-of-sample predictions. While MSR with $K = T = 1$, paSB-MLR, and paSB-robit generally perform worse than SVM in terms of the error rates, they use much fewer hyperplanes and hence have significantly lower computation for out-of-sample predictions. In summary, MSR whose upper-bound for the number of active expects K and number of layers for each expert T can both be adjusted to control its capacity of modeling nonlinearity, can achieve a good compromise between the accuracy and computational complexity for out-of-sample prediction of multinomial class probabilities, and can be further improved by training an additional MSR on the transformed covariates.

We further measure how well the Gibbs sampler is mixing using effective sample size (ESS) for both paSB-MLR and Bayesian multinomial logistic regression (Bayes MLR) of Polson et al. (2013). For both algorithms we let $\boldsymbol{\beta}_j \sim \mathcal{N}\left(0, \text{diag}(\alpha_{j0}^{-1}, \ldots, \alpha_{jV}^{-1})\right)$, where $\alpha_{jv} \sim \text{Gamma}(0.001, 1/0.001)$. The ESS (Holmes and Held, 2006) of a parameter or a function of parameters is defined as $\text{ESS} = L/\left[1 + 2\sum_{h=1}^{\infty}\rho(h)\right]$, where L is the number of post-burn-in samples, $\rho(h)$ is the hth autocorrelation of the parameter or the function of parameters. It describes how quickly an MCMC algorithm generates independent samples.

	10% quantile				median				90% quantile			
	training		testing		training		testing		training		testing	
	Bayes MLR	paSB-MLR	Bayes MLR	paSB-MLR	Bayes MLR	paSB-MLR	Bayes MLR	paSB-MLR	Bayes MLR	paSB-MLR	Bayes MLR	paSB-MLR
square	411.46	194.54	421.62	256.48	913.17	948.53	858.46	952.33	924.48	973.60	927.93	976.33
iris	82.30	90.33	73.75	84.40	149.47	218.21	156.25	174.16	331.71	854.45	341.39	793.00
wine	194.41	314.41	58.41	56.30	467.73	859.67	506.66	643.90	926.55	991.46	958.12	994.77
glass	70.18	162.69	67.81	138.90	137.18	335.14	122.27	329.21	359.75	686.43	347.40	615.02
vehicle	77.71	103.30	74.05	101.64	133.66	230.83	127.22	230.48	426.44	460.07	414.48	453.87
waveform	123.77	120.01	120.99	113.94	199.00	203.38	191.77	209.61	310.84	499.05	291.96	478.59
segment	114.91	104.77	95.63	91.31	281.11	294.17	270.63	355.46	742.63	844.11	752.74	814.62
dna	217.99	238.48	63.66	67.26	481.65	736.58	505.59	772.32	911.68	986.60	927.30	991.63
satimage	53.51	66.19	54.04	66.33	82.50	90.50	81.87	90.11	160.84	168.85	156.83	157.31

Table 2: Comparison of the ESS of the conditional class probability between Bayes MLR and paSB-MLR.

Since the Gibbs sampler of Bayes MLR samples one $\boldsymbol{\beta}_j$ conditioning on all $\boldsymbol{\beta}_{j'}$ for $j' \neq j$, which may lead to strong dependencies between different categories and hence slow down the mixing of the Markov chain. By contrast, the $\boldsymbol{\beta}_j$'s are conditionally independent given the augmented variables b_{ij}'s in paSB-MLR, which may lead to faster mixing. For both Bayes MLR and paSB-MLR, we consider five independent random trials, in each of which we randomly initialize the model parameters, run 10,000 Gibbs sampling iterations, and collect the last 1,000 MCMC samples of $\boldsymbol{\beta}_j$. We use the `mcmcse` package (Flegal et al., 2016) to estimate the ESS of each p_{ij} in a random trial using the 1,000 collected MCMC samples. For the training set, we calculate the 10% quantile, median, and 90% quantile of the ESSs of all p_{ij} for each random trial, and then report their averages over the five random trials in Table 2. For the testing set, we follow the same steps and report the results in Table 2. While paSB-MLR underperforms Bayes MLR on some of the data sets for the 10% ESS quantile, they consistently outperform Bayes MLR on all data sets for both the ESS median and 90% ESS quantile, for both training and testing.

6.6 Robustness of paSB-Robit Regression

We use the contaminated vehicle data to demonstrate the robustness of paSB-robit. As discussed by Liu (2004), the heavy-tailed conditional class probability function of robit regression can robustify the decision boundary when there exist outliers. We use the vehicle training set as inliers, synthesize outliers that are far from inliers, combine both as the new training set, and keep the testing set unchanged. We generate different numbers of outliers so that the ratio of outliers to inliers varies from 0, 0.1, 0.2, 0.3, to 0.5, at each of which we randomly simulate 10 different sets of outlier covariates. We provide the details on how we generate outliers in the Appendix.

We compare L_2-MLR and paSB-robit with $\kappa = 1$ degree of freedom on the contaminated vehicle data. Figure 7 shows the prediction error rate (mean $\pm$ standard deviation) of the testing set for different outlier-inlier ratios. When there are no outliers, both approaches delivers comparable performances. As the ratio increases, paSB-robit with $\kappa = 1$ more and more clearly outperforms L_2-MLR, which justifies the robustness of paSB-robit.

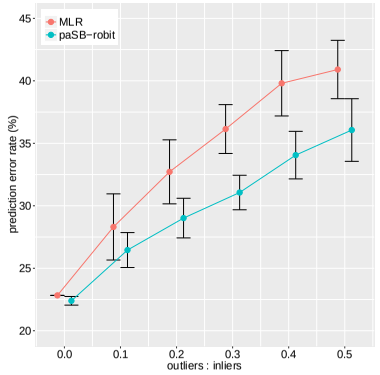


Figure 7: Prediction error rates (%, mean ± standard deviation) for different ratios of outliers to inliers.

7. Conclusions

To transform a cross-entropy-loss binary classifier into a Bayesian multinomial regression model and derive efficient Bayesian inference, we develop a permuted and augmented stick-breaking construction. With permutation, we one-to-one map the categories to sticks to escape from poor category-stick mappings that impose restrictive geometric constraints on the decision boundaries, and with augmentation, we link a category outcome to conditionally independent stick-specific covariate-dependent Bernoulli random variables. We illustrate this general framework by extending binary softplus regression, robit regression, and support vector machine into multinomial ones. Experiment results validate our contributions and show that the proposed multinomial softplus regressions achieve a good compromise between interpretability, complexity, and predictability.

Acknowledgments

The authors would like to thank the editor and two anonymous referees for their insightful and constructive comments and suggestions, and Texas Advanced Computing Center for computational support.

Appendix A. Additional Lemma and Proofs

Proof [Proof of Theorem 1] The conditional probability of y_i given $\{z_s, \pi_{is}\}_{1,S}$ can be expressed as

$$P(y_i = s \,|\, \{z_s, \pi_{is}\}_{1,S}) = \textstyle\sum_{b_{ij}:j>z_s} [P(b_{iz_s}=1)]^{\mathbf{1}(z_s\neq S)} \left[\prod_{j<z_s} P(b_{ij}=0)\right] \left[\prod_{j>z_s} P(b_{ij})\right]$$
$$= [P(b_{iz_s}=1)]^{\mathbf{1}(z_s\neq S)} \left[\prod_{j<z_s} P(b_{ij}=0)\right] \textstyle\sum_{b_{ij}:j>z_s} \left[\prod_{j>z_s} P(b_{ij})\right],$$

which becomes the same as (4) by applying (5) and $\sum_{b_{ij}:j>z_s}\left[\prod_{j>z_s} P(b_{ij})\right]=1$. ■

Proof [Proof of Lemma 3] Under the paSB construction, the probability ratio of categories (choices) s and $s+d$ is a function of the stick success probabilities $\pi_{z_s}, \pi_{z_{(s+1)}}, \cdots, \pi_{z_{(s+d)}}$. More specifically,

$$\frac{p_{i(s+d)}(\boldsymbol{z})}{p_{is}(\boldsymbol{z})} = \frac{\pi_{iz_{(s+d)}}^{\mathbf{1}(z_{(s+d)}\neq S)} \left[\prod_{z_s\leq j<z_{(s+d)}}(1-\pi_{ij})\right]^{\delta(z_s\leq z_{(s+d)})}}{\pi_{iz_s}^{\mathbf{1}(z_s\neq S)} \left[\prod_{z_{(s+d)}\leq j<z_s}(1-\pi_{ij})\right]^{\delta(z_s>z_{(s+d)})}}.$$

■

Proof [Proof of Lemma 4] Since $p_{is}(\boldsymbol{z}) = \left[1-\left(1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}_s}\right)^{-r_s}\right]^{\mathbf{1}(s\neq S)} \prod_{j<s}\left(1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}_j}\right)^{-r_j}$ when $K=T=1$ and $\boldsymbol{z}=(1,\ldots,S)$, the set of solutions to $p_{is}>p_0$ are bounded by the set of solutions to $\left(1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}_j}\right)^{-r_s}>p_0$, $j\in\{1,\ldots,s-1\}$, and $1-\left(1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}_s}\right)^{-r_s}>p_0$, and hence bounded by the convex polytope defined by the set of solutions to the s inequalities as

$$\boldsymbol{x}_i'[(-1)^{\mathbf{1}(j=s)}\boldsymbol{\beta}_j] < (-1)^{\mathbf{1}(j=s)} \ln\left\{\left[p_0^{\mathbf{1}(j\neq s)}(1-p_0)^{\mathbf{1}(j=s)}\right]^{-\frac{1}{r_j}}-1\right\}, \quad j\in\{1,\ldots,s\}.$$

■

Lemma 5 *Without loss of generality, let us assume that the category-stick mapping is fixed at $\boldsymbol{z}=(1,2,\cdots,S)$. The paSB multinomial logistic model that assigns choice $s\in\{1,\ldots,S\}$ for individual i with probability $p_{is}=(\pi_{is})^{\mathbf{1}(s\neq S)}\prod_{j<s}(1-\pi_{ij})$, where $\pi_{is}=1/(1+e^{-W_{is}})$, can be considered as a sequential random utility maximization model. This model selects choice s once $U_{is}>\sum_{j\geq s}U_{ij}$ is observed for $s=1,\ldots,S$, where U_{is} are defined as*

$$\begin{aligned} U_{i1} &= U_{i2}+\cdots+U_{iS}+W_{i1}+\varepsilon_{i1},\\ &\cdots\\ U_{is} &= \sum_{j>s}U_{ij}+W_{is}+\varepsilon_{is},\\ &\cdots\\ U_{i(S-1)} &= W_{i(S-1)}+\varepsilon_{i(S-1)},\\ U_{iS} &= 0, \end{aligned}$$

and $\varepsilon_{is}\overset{i.i.d.}{\sim}\text{Logistic}(0,1)$ are independent, and identically distributed (i.i.d.) random variables following the standard logistic distribution.

Proof [Proof of Lemma 5] Note that $P(\varepsilon<x)=1/(1+e^{-x})$ if $\varepsilon\sim\text{Logistic}(0,1)$. First consider the choice of individual i be $y_i=1$, which would happen with probability

$$P(y_i=1)=P\left(U_{i1}>\sum\nolimits_{j\geq 1}U_{ij}\right)=P(\varepsilon_{i1}>-W_{i1})=1/(1+e^{-W_{i1}})=\pi_{i1}=p_{i1}.$$

	square	iris	wine	glass	vehicle	waveform	segment	dna	satimage
Train size	294	120	142	171	592	500	231	2000	4435
Test size	126	30	36	43	254	4500	2079	1186	2000
Covariate number	2	4	13	9	18	21	19	180	36
Category number	3	3	3	6	4	3	7	3	6

Table 3: Multi-class classification data sets used in experiments for model comparison.

Then for $s = 2, \cdots, S-1$,

$$\begin{aligned} P(y_i = s) &= P(y_i = s \,|\, y_i > s-1) P(y_i > s-1) \\ &= P\left(U_{is} > \sum\nolimits_{j>s} U_{ij}\right) \prod\nolimits_{j \le s-1} P\left(U_{ij} < \sum\nolimits_{j'>j} U_{ij'}\right) \\ &= P(\varepsilon_{is} > -W_{is}) \prod\nolimits_{j \le s-1} P(\varepsilon_{ij} < -W_{ij}) \\ &= \pi_{is} \prod\nolimits_{j \le s-1} (1 - \pi_{ij}) \\ &= p_{is}. \end{aligned}$$

Finally, $P(y_i = S) = 1 - \sum_{j<S} P(y_i = j) = \prod_{j<S}(1 - \pi_{ij}) = p_{iS}$. ■

Appendix B. Experimental Settings and Additional Results

The table below summarizes the sizes of both training and testing sets, and the number of covariates and categories. The training and testing sets are predefined for vehicle, dna, and satimage. Note that the training and validation sets are combined as training. We divide the other data sets into training and testing as follows. For iris, wine, and glass, five random partitions are taken such that for each partition the training set accounted for 80% of the whole data set while the testing set 20%. The classification error rate is calculated by averaging the error rates of all five random partitions. For square, waveform, and segment, only one random partition is taken, where 70% of the square data set are used as training and the remaining 30% as testing, and 10% of both the waveform and segment datas are used as training and the remaining 90% as testing.

We compare paSB-MLR, paSB-robit with $\kappa = 6$, paSB-MSVM, and MSR with three other models, including L_2 regularized multinomial logistic regression (L_2-MLR), support vector machine (SVM), and adaptive multi-hyperplane machine (AMM). For paSB-robit, we run 8,000 iterations and discard the first 5,000 as burn-in (this setting is unchanged for experiments in Section 6.6). For paSB-MSVM, we use the spike-and-slab prior to select the kernel bases and set 0.5 as the probability of spike at 0, which is referred to as a uniform prior by Polson and Scott (2011). A Gaussian radial basis function (RBF) kernel is used and the kernel width is selected by 3-fold cross validation from $(2^{-10}, 2^{-9}, \ldots, 2^{10})$. We run 1000 MCMC iterations and discard the first 500 as burn-in samples. For MSR, we try both paSB and parSB with (K, T) set as $(1,1)$, $(1,3)$, $(5,1)$, or $(5,3)$. We run 10000 MCMC iterations and discard the first 5000 as burn-in samples. The predictive probability is calculated by averaging the Monte Carlo average predictive probabilities from paSB and

	paSB-MLR	paSB-robit	paSB-MSVM	$K=1$ $T=1$	$K=1$ $T=3$	$K=5$ $T=1$	$K=5$ $T=3$	DT-MSR	L_2-MLR	SVM	AMM
square	53.17(2)	57.94(2)	0(256)	57.14(4)	13.49(12)	1.59(10)	0.79(33)	1.59(40)	62.29(2)	4.76(22)	16.67(7)
iris	2(2)	6.67(2)	3.33(97.6)	4.67(4)	4.67(12)	4(5.4)	3.33(12)	4.67(16.2)	3.33(2)	4(35)	4.67(8.6)
wine	8.33(2)	4.86(2)	2.14(125.8)	4.45(4)	4.45(12)	6.67(4)	3.34(12)	3.89(16.4)	3.89(2)	2.78(77.2)	3.89(7.8)
glass	39.07(5)	34.41(5)	30.23(137.6)	35.35(10)	30.7(30)	33.02(10.4)	35.81(33.2)	32.56(47.8)	33.02(5)	28.84(118)	37.67(23.8)
vehicle	25.98(3)	22.44(3)	17.71(592)	23.62(6)	21.65(18)	18.9(12)	16.93(33)	18.11(45)	22.83(3)	18.50(256)	21.89(17)
waveform	18.78(2)	16.84(2)	16.56(500)	19.73(4)	17.11(12)	17.07(6)	17.11(18)	16.49(23)	15.60(2)	15.22(212)	18.54(11.6)
segment	7.07(6)	9.81(6)	9.86(231)	8.61(12)	8.37(36)	7.31(13)	7.79(36)	8.85(50)	8.56(6)	6.20(93)	12.47(11.4)
dna	6.58(2)	4.30(2)	7.25(1701)	5.56(4)	5.73(12)	6.07(7)	5.82(12)	4.72(17)	5.98(2)	4.97(1142)	5.43(18.6)
satimage	21.35(5)	16.40(5)	9.5(4315)	15.8(10)	14.7(30)	13.25(34)	11.95(102)	11.55(105)	17.80(5)	8.50(1652)	15.31(16.8)

Table 4: Comparison of classification error rates (%) of paSB-MLR, paSB-robit, paSB-MSVM, MSRs with various K and T (results column 3 to 6), MSR with data transformation (DT-MSR), L_2-MLR, SVM, and AMM, using the collected MCMC sample with the highest log-likelihood. The number of active hyperplanes/support vectors used for out-of-sample predictions are shown in parenthesis.

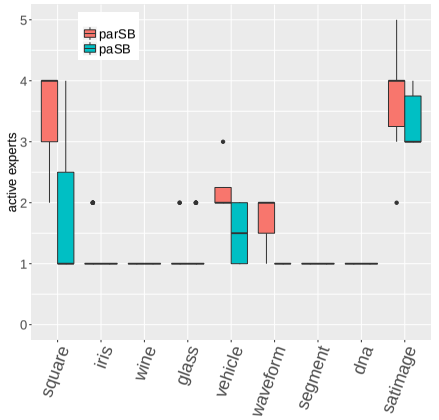


Figure 8: Boxplots of the number of active experts inferred by paSB/parSB MSRs with $K = 5$ and $T = 3$.

parSB MSRs. An observation in the testing set is classified to the category associated with the largest predictive probability.

For MSRs on benchmark data with data transformation, in the first step, we run paSB-MSR and parSB-MSR with $K = 5$ and $T = 3$ on the original covariates for 3,000 iterations to learn $\boldsymbol{z}$, $\{r_{jk}\}$, $\{\boldsymbol{\beta}_{jk}^{(t)}\}$. We then transform the covariates by Equation (6.4), where $\boldsymbol{\beta}_{jk}^{(t)}$ are associated with active experts. In the last step, we use $\boldsymbol{z}$ learned in the first step, and run paSB-MSR and parSB-MSR with $K = 5$ and $T = 1$ for 10,000 iterations and collect the last 5,000 samples to compute the predictive probabilities.

We use the L_2-MLR provided in the LIBLINEAR package (Fan et al., 2008) to train a linear classifier, where a bias term is included and the regularization parameter is five-

fold cross-validated on the training set from $(2^{-10}, 2^{-9}, \ldots, 2^{15})$. We run the LIBLINEAR package in R via `R` package `LiblineaR` (Helleputte, 2015). We also classify an observation to the category associated with the largest predictive probability. For SVM, we use the LIBSVM package (Chang and Lin, 2011) and run it in R with `R` package `e1071` (Meyer et al., 2015). A Gaussian RBF kernel is used and three-fold cross validation is adopted to tune both the regularization parameter and kernel width from $(2^{-10}, 2^{-9}, \ldots, 2^{10})$ on the training set. For paSB-MSVM, we use three-fold cross validation on the training set to select a kernel width from $(2^{-10}, 2^{-9}, \ldots, 2^{10})$. We choose the default LIBSVM settings for all the other parameters. We consider adaptive multi-hyperplane machine (AMM) of Wang et al. (2011b), as implemented in the BudgetSVM[1] (Version 1.1) software package (Djuric et al., 2013). We use the batch version of the algorithm. Important parameters of the AMM include both the regularization parameter ν and training epochs E. As also mentioned by Kantchelian et al. (2014), we do not observe the testing errors of AMM to strictly decrease as E increased. Thus, in addition to cross validating the regularization parameter ν on the training set from $\{10^{-7}, 10^{-6}, \ldots, 10^{-2}\}$, as done in Wang et al. (2011b), for each ν, we try $E \in \{5, 10, 20, 50, 100\}$ sequentially until the cross-validation error begins to decrease, *i.e.*, under the same ν, we choose $E = 20$ if the cross-validation error of $E = 50$ is greater than that of $E = 20$. We use the default settings for all the other parameters, and calculate average classification error rates.

We add an outlier to the vehicle data in Section 6.6 as follows. There are 18 covariates, whose values range from -1 to 1, in this data set. To simulate an outlier, since the MLR regression coefficients associated with the 4th, 5th, and 6th covariates all have large absolute values, we first draw three uniform random numbers from $(-3, -2) \cup (2, 3)$ and assign them to these three covariates, and then assign each of the 12 remaining covariates a uniform random number from $(-1, 1)$. Finally, we draw the category label y uniformly from $\{1, 2, 3, 4\}$.

1. `http://www.dabi.temple.edu/budgetedsvm/`

References

J. H. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *J. Amer. Statist. Assoc.*, 88(422):669–679, 1993.

D. M. Blei and M. I. Jordan. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143, 2006.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152. ACM, 1992.

C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

Q. Chang, Q. Chen, and X Wang. Scaling Gaussian RBF kernel width to improve SVM classification. In *2005 International Conference on Neural Networks and Brain*, volume 1, pages 19–22. IEEE, 2005.

V. Cherkassky and Y. Ma. Practical selection of SVM parameters and noise estimation for SVM regression. *Neural networks*, 17(1):113–126, 2004.

Y. Chung and D. B. Dunson. Nonparametric Bayes conditional distribution modeling with variable selection. *J. Amer. Statist. Assoc.*, 104:1646–1660, 2009.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *J. Mach. Learn. Res.*, 2:265–292, 2002.

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

N. Djuric, L. Lan, S. Vucetic, and Z. Wang. Budgetedsvm: A toolbox for scalable SVM approximations. *J. Mach. Learn. Res.*, 14:3813–3817, 2013.

D. B. Dunson and J.-H. Park. Kernel stick-breaking processes. *Biometrika*, 95(2):307–323, 2008.

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *J. Mach. Learn. Res.*, pages 1871–1874, 2008.

T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *Ann. Statist.*, 1(2): 209–230, 1973.

James M. Flegal, John Hughes, and Dootika Vats. *mcmcse: Monte Carlo Standard Errors for MCMC*. Riverside, CA and Minneapolis, MN, 2016. R package version 1.2-1.

X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *AISTATS*, pages 315–323, 2011.

W. H. Greene. *Econometric Analysis*. Upper Saddle River, New Jersey: Prentice Hall, 2003.

J. Grifin and M. Steel. Order-based dependent Dirichlet processes. *J. Amer. Statist. Assoc.*, 2006.

B. Grünbaum. *Convex Polytopes*. Springer New York, 2013.

W. M. Hanemann. Discrete/continuous models of consumer demand. *Econometrica: Journal of the Econometric Society*, pages 541–561, 1984.

T. Helleputte. *LiblineaR: Linear Predictive Models Based on the LIBLINEAR C/C++ Library*, 2015. R package version 1.94-2.

R. Henao, X. Yuan, and L. Carin. Bayesian nonlinear support vector machines and discriminative factor modeling. In *NIPS*, pages 1754–1762, 2014.

N. L. Hjort, C. Holmes, P. Müller, and S. G. Walker. *Bayesian Nonparametrics*, volume 28. Cambridge University Press, 2010.

C. C. Holmes and L. Held. Bayesian auxiliary variable models for binary and multinomial regression. *Bayesian Analysis*, 1(1):145–168, 2006.

K. Imai and D. A. van Dyk. A Bayesian analysis of the multinomial probit model using marginal data augmentation. *Journal of Econometrics*, 124(2):311–334, 2005.

H. Ishwaran and L. F. James. Gibbs sampling methods for stick-breaking priors. *J. Amer. Statist. Assoc.*, 96(453), 2001.

A. Jasra, C. C. Holmes, and D. A. Stephens. Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling. *Statistical Science*, pages 50–67, 2005.

A. Kantchelian, M. C. Tschantz, L. Huang, P. L. Bartlett, A. D. Joseph, and J. D. Tygar. Large-margin convex polytope machine. In *NIPS*, pages 3248–3256, 2014.

M. E. Khan, S. Mohamed, B. M. Marlin, and K. P. Murphy. A stick-breaking likelihood for categorical data analysis with latent Gaussian models. In *AISTATS*, pages 610–618, 2012.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

K. Kurihara, M. Welling, and Y. W. Teh. Collapsed variational Dirichlet process mixture models. In *IJCAI*, volume 7, pages 2796–2801, 2007.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

Y. Lee, Y. Lin, and G. Wahba. Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *J. Amer. Statist. Assoc.*, 99(465):67–81, 2004.

S. Linderman, M. Johnson, and R. P. Adams. Dependent multinomial models made easy: Stick-breaking with the Pólya-Gamma augmentation. In *NIPS*, pages 3438–3446, 2015.

Chuanhai Liu. Robit regression: a simple robust alternative to logistic and probit regression. *Applied Bayesian Modeling and Casual Inference from Incomplete-Data Perspectives*, pages 227–238, 2004.

Y. Liu and M. Yuan. Reinforced multicategory support vector machines. *Journal of Computational and Graphical Statistics*, 20(4):901–919, 2011.

B. K. Mallick, D. Ghosh, and M. Ghosh. Bayesian classification of tumours by using gene expression data. *J. R. Stat. Soc: Series B*, 67(2):219–234, 2005.

P. McCullagh and J. A. Nelder. *Generalized linear models*, volume 37. CRC press, 1989.

R. McCulloch and P. E Rossi. An exact likelihood analysis of the multinomial probit model. *Journal of Econometrics*, 64(1):207–240, 1994.

R. E. McCulloch, N. G. Polson, and P. E. Rossi. A Bayesian analysis of the multinomial probit model with fully identified parameters. *Journal of Econometrics*, 99(1):173–193, 2000.

D. McFadden. Conditional logit analysis of qualitative choice behavior. *Frontiers in Econometrics*, pages 105–142, 1973.

D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, and F. Leisch. *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*, 2015. URL `https://CRAN.R-project.org/package=e1071`. R package version 1.6-7.

K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.

V. Nair and G. E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *ICML*, pages 807–814, 2010.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 2014.

J. Pitman. Some developments of the Blackwell-Macqueen urn scheme. *Statistics, Probability, and Game Theory: Papers in Honor of David Blackwell*, 30:245–267, 1996.

N. G. Polson and S. L. Scott. Data augmentation for support vector machines. *Bayesian Analysis*, 6(1):1–23, 2011.

N. G. Polson, J. G. Scott, and J. Windle. Bayesian inference for logistic models using Pólya–Gamma latent variables. *J. Amer. Statist. Assoc.*, 108(504):1339–1349, 2013.

L. Ren, L. Du, L. Carin, and D. Dunson. Logistic stick-breaking process. *J. Mach. Learn. Res.*, 12:203–239, 2011.

A. Rodriguez and D. B. Dunson. Nonparametric Bayesian models through probit stick-breaking processes. *Bayesian Analysis*, 6(1), 2011.

B. Schölkopf, C. J. C. Burges, and A. J. Smola. *Advances in Kernel Methods: Support Vector Learning*. MIT Press, 1999.

J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, pages 639–650, 1994.

C. Soares, P. B. Brazdil, and P. Kuba. A meta-learning method to select the kernel width in support vector regression. *Machine Learning*, 54(3):195–209, 2004.

P. Sollich. Bayesian methods for support vector machines: Evidence and predictive class probabilities. *Machine Learning*, 46(1-3):21–52, 2002.

S. Srinivas. A generalization of the noisy-or model. In *UAI*, pages 208–215, 1993.

K. E. Train. *Discrete Choice Methods with Simulation*. Cambridge university press, 2009.

C. Wang, J. Paisley, and D. M. Blei. Online variational inference for the hierarchical Dirichlet process. In *AISTATS*, 2011a.

Z. Wang, N. Djuric, K. Crammer, and S. Vucetic. Trading representability for scalability: adaptive multi-hyperplane machine for nonlinear classification. In *KDD*, pages 24–32, 2011b.

Z. Zhang and M. I. Jordan. Bayesian multicategory support vector machines. In *UAI*, 2006.

M. Zhou. Infinite edge partition models for overlapping community detection and link prediction. In *AISTATS*, pages 1135–1143, 2015.

M. Zhou. Softplus regressions and convex polytopes. *arXiv:1608.06383*, 2016.

M. Zhou, Y. Cong, and B. Chen. Augmentable gamma belief networks. *J. Mach. Learn. Res.*, 17(163):1–44, 2016.

Journal of Machine Learning Research 18 (2018) 1-35 Submitted 7/17; Revised 12/17; Published 4/18

Steering Social Activity: A Stochastic Optimal Control Point Of View*

Ali Zarezade ZAREZADE@CE.SHARIF.EDU
Sharif University of Technology
Teheran, Iran

Abir De ADE@MPI-SWS.ORG
Max Planck Institute for Software Systems
Kaiserslautern, Germany

Utkarsh Upadhyay UTKARSHU@MPI-SWS.ORG
Max Planck Institute for Software Systems
Kaiserslautern, Germany

Hamid R. Rabiee RABIEE@SHARIF.EDU
Sharif University of Technology
Teheran, Iran

Manuel Gomez-Rodriguez MANUELGR@MPI-SWS.ORG
Max Planck Institute for Software Systems
Kaiserslautern, Germany

Editor: Nikos Vlassis

Abstract

User engagement in online social networking depends critically on the level of *social activity* in the corresponding platform—the number of online *actions*, such as posts, shares or replies, taken by their users. Can we design data-driven algorithms to increase social activity? At a user level, such algorithms may increase activity by helping users decide when to take an action to be more likely to be *noticed* by their peers. At a network level, they may increase activity by *incentivizing* a few influential users to take more actions, which in turn will trigger additional actions by other users.

In this paper, we model social activity using the framework of marked temporal point processes, derive an alternate representation of these processes using stochastic differential equations (SDEs) with jumps and, exploiting this alternate representation, develop two efficient online algorithms with provable guarantees to steer social activity both at a user and at a network level. In doing so, we establish a previously unexplored connection between optimal control of jump SDEs and doubly stochastic marked temporal point processes, which is of independent interest. Finally, we experiment both with synthetic and real data gathered from Twitter and show that our algorithms consistently steer social activity more effectively than the state of the art.

Keywords: marked temporal point processes, stochastic optimal control, stochastic differential equations with jumps, social networks, information networks

. *Preliminary version of this work appeared in Zarezade et al. (2017).

1. Introduction

People perform a wide variety of online actions in a growing number of online social networking sites. In most cases, these online actions can be broadly categorized into *exogenous* actions, taken by users at their own initiative, and *endogenous* actions, taken by users as a *response* to previous actions by other users. For example, users often post small pieces of information at their own initiative, which can then trigger additional posts, shares or replies by other users. In this paper, our goal is designing online algorithms that, by steering exogenous actions, are able to increase the number of endogenous actions and thus the overall user activity.

We first address the above problem from the perspective of an individual user whose posts compete for *attention* with dozens, if not hundreds, of posts *simultaneously* shared by other users that her *followers* follow (Backstrom et al., 2011; Gomez-Rodriguez et al., 2014). In this context, recent empirical studies have shown that stories at the top of their followers' feed are more likely to be *noticed* and consequently liked, shared or replied to (Hodas and Lerman, 2012; Kang and Lerman, 2015; Lerman and Hogg, 2014). Therefore, we design an online algorithm, RedQueen, to help a user decide when-to-post to increase her chances to stay at the top, increasing her *visibility*. Then, we tackle the problem from the perspective of an entire online social networking site, where an endless stream of stories posted by its users are constantly eliciting a variable number of likes, shares or replies by other users (Goel et al., 2012; Farajtabar et al., 2014; Rizoiu et al., 2017). Here, we design another online algorithm, Cheshire, to find how much should we incentivize a small number of influential users to post more over time to increase the overall number of additional posts, shares or replies in the site.

More specifically, we represent users' actions using the framework of marked temporal point processes, which characterizes the continuous time interval between actions using conditional intensity functions (Aalen et al., 2008), and model endogenous and exogenous actions using multidimensional Hawkes processes (Hawkes, 1971). Then, we derive an alternate representation of these processes using stochastic differential equations (SDEs) with jumps and, exploiting this alternate representation, address the design of both algorithms from the perspective of optimal control for SDEs with jumps (Hanson, 2007). Our approach differs from previous literature in two key technical aspects, which are of independent interest:

I. The control signals are conditional (posting) intensities, which are used to sample stochastic events (*i.e.*, stories to post). In contrast, previous work considered the control signal to be a time-varying real vector. As a consequence, our approach requires another layer of stochasticity.

II. The (posting) intensities are stochastic Markov processes and thus the dynamics are doubly stochastic. This requires us to redefine the cost-to-go to incorporate the instantaneous value of these intensities as additional arguments. Previous work has typically considered constant intensities until very recently (Wang et al., 2016, 2017).

These technical aspects have implications beyond steering social activity since they enable us to establish a previously unexplored connection between optimal control of jump SDEs

and doubly stochastic temporal point processes (*e.g.*, Hawkes processes), which have been increasingly used to model social activity (Farajtabar et al., 2014, 2015b; Zhao et al., 2015).

In both cases, we find that the solution to the corresponding optimal control problem is surprisingly simple. At a user level, the optimal posting intensity for a user to achieve high visibility depends linearly on the position of her most recent post on each of her follower's feed and, at a network level, the optimal level of incentivized actions depends linearly on the current level of overall actions. Moreover, at a user level, the coefficients of the linear relationship only depend on tunable parameters, which are predefined, and, at a network level, the coefficients can be found by solving a matrix Riccati differential equation (Garrett, 2013) and a first order differential equation, which has a closed form solution. As a consequence, both RedQueen and Cheshire are simple and highly efficient online procedures, which can be implemented in a few lines of code (refer to Algorithms 1 and 3). Finally, we perform experiments on both synthetic and real-data gathered from Twitter and show that our algorithms can consistently steer social activity at a user and at a network level more effectively than the state of the art.

1.1 Related Work

Our work relates to previous work on the when-to-post problem, the activity shaping problem, stochastic optimal control, and temporal point processes.

The "when-to-post" problem was first studied by Spasojevic et al. (2015), who performed a large empirical study on the best times for a user to post in Twitter and Facebook, measuring attention a user elicits by means of the number of responses to her posts. Moreover, they designed several heuristics to pinpoint at the times that elicit the greatest attention in a training set and showed that these times also lead to more responses in a held-out set. Since then, algorithmic approaches to the "when-to-post" problem with provable guarantees have been largely lacking. Only very recently, Karimi et al. (2016) introduced a convex optimization framework to find optimal broadcasting strategies, measuring attention a user elicits as the time that at least one of her posts is among the k most recent stories received in her followers' feed. However, their algorithm requires expensive data pre-processing, it does not adapt to changes in the users' feeds dynamics, and in practice, it is less effective than our algorithm RedQueen, as shown in Section 5.

The "activity shaping" problem was first studied by Farajtabar et al. (2015a), who derived a time dependent linear relation between the *intensity* of exogenous actions and the overall intensity of actions in a social network under a model of actions based on multidimensional Hawkes processes and, exploiting this connection, developed a convex optimization framework for activity shaping. One of the main shortcomings of their framework is that it provides deterministic exogenous intensities that do not adapt to changes in the users' intensities and, as a consequence, it is less effective than our algorithm Cheshire, as shown in Section 5. More recently, Farajtabar et al. (2016) developed a heuristic method that splits the time window of interest into stages and adapts to changes in the users' intensities at the beginning of each stage. However, their method is suboptimal, it does not have *provable* guarantees, and it achieves lower performance than our method.

In the traditional control literature (Hanson, 2007), two key aspects of our approach—intensities as control signals and stochastic intensities—have been largely understudied.

Only very recently, Wang et al. (2016) and Wang et al. (2017) have considered these aspects. However, in Wang et al. (2016), the intensities are not stochastic and the resulting algorithm needs to know the future actions, hindering its applicability, and, in Wang et al. (2017), the solution is open-loop and the control policy depends on the expectation of the uncontrolled dynamics, which needs to be calculated approximately by a time consuming sampling process. In contrast, our framework consider double stochastic intensities, our solution is closed-loop, our control policies only depend on the current state of the dynamics, and the feedback coefficients only need to be calculated once off-line.

Temporal point processes have been increasingly used for representation and modeling in a wide range of applications in social and information systems, *e.g.*, information propagation (Gomez-Rodriguez et al., 2011; Du et al., 2013; Zhao et al., 2015), opinion dynamics (De et al., 2016), product competition (Valera and Gomez-Rodriguez, 2015; Zarezade et al., 2016b), spatiotemporal social activity (Jankowiak and Gomez-Rodriguez, 2017; Zarezade et al., 2016a), information reliability (Tabibian et al., 2017), or human learning (Mavroforakis et al., 2017). However, in such context, algorithms based on stochastic optimal control of temporal point processes have been lacking.

2. Preliminaries

In this section, we first revisit the framework of temporal point processes (Aalen et al., 2008) and then describe how to use such framework to represent actions and feeds (Karimi et al., 2016) as well as to model endogenous and exogenous actions in social networks (Farajtabar et al., 2014).

2.1 Temporal Point Processes

A univariate temporal point process is a stochastic process whose realization consists of a sequence of discrete events localized in time, $\mathcal{H} = \{t_i \in \mathbb{R}^+ \,|\, i \in \mathbb{N}^+, t_i < t_{i+1}\}$. It can also be represented as a counting process $N(t)$, which counts the number of events before time t, *i.e.*,

$$N(t) = \sum_{t_i \in \mathcal{H}} \mathbb{I}(t - t_i \geq 0),$$

where $\mathbb{I}(\cdot)$ is the indicator function. Then, we can characterize the counting process using the conditional intensity function $\lambda^*(t)$, which is the conditional probability of observing an event in an infinitesimal window $[t, t + dt)$ given the history of event times up to time t, $\mathcal{H}(t) = \{t_i \in \mathcal{H} \,|\, t_i < t\}$, *i.e.*,

$$\lambda^*(t)\, dt = \mathbb{P}\left\{\text{event in } [t, t + dt) \,|\, \mathcal{H}(t)\right\} = \mathbb{E}[dN(t) \,|\, \mathcal{H}(t)],$$

where $dN(t) := N(t+dt) - N(t) \in \{0, 1\}$, the sign * means that the intensity may depend on the history $\mathcal{H}(t)$. Moreover, given a function $f(t)$, it will be useful to define the convolution with respect to $dN(t)$ as

$$f(t) \star dN(t) := \int_0^t f(t - s) dN(s) = \sum_{t_i \in \mathcal{H}(t)} f(t - t_i).$$

One can readily extend the above definitions to multivariate (or multidimensional) temporal point processes, which have been recently used to represent many different types of event data produced in social networks, such as the times of tweets (Farajtabar et al., 2014), retweets (Zhao et al., 2015), or links (Farajtabar et al., 2015b). More specifically, a realization of an m-dimensional temporal point process, $\mathcal{H} = \{(t_i, u_i) \,|\, i \in \mathbb{N}^+, t_i \in \mathbb{R}_+, u_i \in [m], t_i < t_{i+1}\}$, consists of m sequences of discrete events localized in time, $\mathcal{H} = \cup_{u\in[m]}\mathcal{H}_u$, where $\mathcal{H}_u = \{(t_i, u_i) \in \mathcal{H} \,|\, u_i = u\}$. Equivalently, it can be represented by an m-dimensional counting process $\mathbf{N}(t)$, where $N_u(t)$ counts the number of events in the u-th sequence before time t, and this counting process can be characterized by m intensity functions, *i.e.*,

$$\boldsymbol{\lambda}^*(t)dt = \mathbb{E}[d\boldsymbol{N}(t)|\mathcal{H}(t)],$$

where $\mathcal{H}(t) = \{(t_i, u_i) \in \mathcal{H} \,|\, t_i < t\}$, $d\boldsymbol{N}(t) := (dN_u(t))_{u\in[m]} := (N_u(t + dt) - N_u(t))_{u\in[m]}$, and $\boldsymbol{\lambda}^*(t) := (\lambda_u^*(t))_{u\in[m]}$ denotes the associated intensities, which may depend on history $\mathcal{H}(t)$. Finally, given a function $f(t)$, one can define the convolution with respect to $d\boldsymbol{N}(t)$ as

$$f(t) \star d\boldsymbol{N}(t) := \int_0^t f(t-s)d\boldsymbol{N}(s) = \left(\sum_{t_i\in\mathcal{H}_u(t)} f(t-t_i)\right)_{u\in[m]}.$$

In the remainder of the paper, to simplify the notation, we drop the sign * from the intensities.

2.2 Representation of Actions and Feeds

Given a (directed) social network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = n$ users, we assume that each user i can take a variety of online actions, *e.g.*, posting, sharing or replying, and she will be exposed to the online actions taken by the users she *follows* through her feed. Here, we assume that user i follows user j if and only if $(i, j) \in \mathcal{E}$.

Then, we represent the times when users take online actions as a multidimensional counting process $\boldsymbol{N}(t)$, where the i-th dimension, $N_i(t)$, counts the number of actions taken by user i up to time t. Here, we denote the history of times of the actions taken by user i by time t as $\mathcal{H}_i(t)$, the entire history of times as $\mathcal{H}(t) = \cup_{i\in\mathcal{V}}\mathcal{H}_i(t)$, and characterize this multidimensional process using n intensity functions, *i.e.*, $\mathbb{E}[d\boldsymbol{N}(t)|\mathcal{H}(t)] = \boldsymbol{\lambda}(t)\, dt$.

Given the adjacency matrix $\boldsymbol{A} \in \{0, 1\}^{n\times n}$, where $\boldsymbol{A}_{ij} = 1$ indicates that user j follows user i, we can represent the times of the actions users are exposed to through their feeds as a sum of counting processes, $\boldsymbol{A}^T\boldsymbol{N}(t)$, and calculate the corresponding conditional intensities as $\boldsymbol{\gamma}(t) = \boldsymbol{A}^T\boldsymbol{\lambda}(t)$. Here, we denote the history of times of the actions user j is exposed to by time t as $\mathcal{F}_j(t) := \cup_{i\in\mathcal{N}(j)}\mathcal{H}_i(t)$, where $\mathcal{N}(j)$ is the set of users that j follows.

Finally, from the perspective of a user i, it is useful to define the multidimensional counting process $\boldsymbol{M}_{\setminus i}(t) = \boldsymbol{A}^T\boldsymbol{N}(t) - \boldsymbol{A}_i N_i(t)$, in which the j-th dimension, $M_{j\setminus i}(t)$, counts the number of actions taken by other users that user j follows up to time i, and $\boldsymbol{A}_i$ is the i-th row of the adjacency matrix $\boldsymbol{A}$. Moreover, for each dimension, the conditional intensity is given by $\gamma_{j\setminus i}(t) = \gamma_j(t) - \lambda_i(t)$ and the history is given by $\mathcal{F}_{j\setminus i}(t) := \mathcal{F}_j(t)\setminus\mathcal{H}_i(t)$. When there is no ambiguity, we will omit the subscript i and write $\boldsymbol{M}_{\setminus i}(t) = \boldsymbol{M}(t)$ and $\boldsymbol{\gamma}_{\setminus i}(t) = \boldsymbol{\gamma}(t)$.

2.3 Modeling Endogenous and Exogenous Actions

Following previous work (Farajtabar et al., 2014, 2015b; Zhao et al., 2015; De et al., 2016; Rizoiu et al., 2017), we model endogenous and exogenous actions using multidimensional Hawkes processes (Hawkes, 1971). More specifically, we proceed as follows.

From the perspective of an individual user i, we assume we observe the online actions her followers are exposed to through their feeds. Then, consider the following functional form for the intensities $\boldsymbol{\gamma}_{\setminus i}(t) = \boldsymbol{\gamma}(t)$ of the followers' feeds from Section 2.2:

$$\boldsymbol{\gamma}(t) = \boldsymbol{\gamma}_0(t) + \boldsymbol{D}\,\kappa_\omega(t) \star d\boldsymbol{M}(t) = \boldsymbol{\gamma}_0(t) + \boldsymbol{D}\int_0^t \kappa_\omega(t-s)\, d\boldsymbol{M}(s) \tag{1}$$

where $\boldsymbol{\gamma}_0(t)$ are time-varying functions that model exogenous actions, *i.e.*, actions users take at their own initiative, the second term, with $\boldsymbol{D} = \mathrm{diag}(\alpha_j), \alpha_j \geq 0$, models endogenous actions, *i.e.*, actions users take as *response* to previous actions (their own as well as the ones taken by others), and $\kappa_\omega(t) = e^{-\omega t}\,\mathbb{I}(t \geq 0)$ is an exponential kernel modeling the decay on influence over time[1].

From the perspective of an entire online social networking site, we assume we observe the online actions taken by all the users[2]. Thus, we consider the following functional form for the users' intensities $\boldsymbol{\lambda}(t)$ from Section 2.2:

$$\boldsymbol{\lambda}(t) = \boldsymbol{\lambda}_0(t) + \boldsymbol{B}\,\kappa_\omega(t) \star d\boldsymbol{N}(t) = \boldsymbol{\lambda}_0(t) + \boldsymbol{B}\int_0^t \kappa_\omega(t-s)\, d\boldsymbol{N}(s), \tag{2}$$

where $\boldsymbol{\lambda}_0(t) \geq 0$ are time-varying functions that model exogenous actions and the second term models endogenous actions—actions users take as response to the actions taken by the users they follow. Here, we parameterize the strength of *influence* between users using a sparse nonnegative *influence matrix* $\boldsymbol{B} = (\beta_{uv}) \in \mathbb{R}_+^{m\times m}$, where β_{uv} means user u's actions directly triggers *follow-ups* from user v and we assume $\beta_{uv} = 0$ if $(u, v) \notin \mathcal{E}$.

In both cases, the second term makes the intensity dependent on the history and a stochastic process itself. Moreover, the following alternative representation of multidimensional Hawkes processes will be useful to design our stochastic optimal control algorithms (proven in Appendix A):

Proposition 1 *Let $\boldsymbol{N}(t)$ be an m-dimensional Hawkes process with an associated intensity $\boldsymbol{\lambda}(t) = \boldsymbol{\lambda}_0(t) + \boldsymbol{C}\int_0^t \kappa_\omega(t-s)\, d\boldsymbol{N}(s)$, where $\lambda_0(t) \geq 0$ is a differentiable time-varying function, $\kappa_\omega(t) = e^{-\omega t}\,\mathbb{I}(t \geq 0)$ is an exponential triggering kernel, and $\omega \geq 0$ and $\boldsymbol{C} \in \mathbb{R}_+^{m\times m}$ are given parameters. Then, the tuple $(\boldsymbol{N}(t), \boldsymbol{\lambda}(t))$ is a doubly stochastic Markov process, whose dynamics can be defined by the following jump SDEs:*

$$d\boldsymbol{\lambda}(t) = \left[\boldsymbol{\lambda}_0'(t) + \omega\boldsymbol{\lambda}_0(t) - \omega\boldsymbol{\lambda}(t)\right] dt + \boldsymbol{C}\, d\boldsymbol{N}(t), \tag{3}$$

1. Exponential kernels have been shown to provide relatively good predictive performance (Farajtabar et al., 2014, 2015b; Zhao et al., 2015; De et al., 2016; Rizoiu et al., 2017), however, we acknowledge that power-law may have higher predictive performance in some scenarios, as recently shown by Mishra et al. (2016). Here, we opt for exponential kernels since, in such case, the dynamics of the corresponding intensity can be expressed by a linear SDE with jumps and this will be helpful in the derivation of our stochastic optimal control algorithms.
2. We acknowledge that there may be scenarios in which one only has access to a subset of the online actions taken by the users. However, in those cases, one could resort to an increasing number of methods to fit multidimensional Hawkes processes from incomplete observations, *e.g.*, refer to Xu et al. (2017)

with the initial condition $\boldsymbol{\lambda}(0) = \boldsymbol{\lambda}_0(0)$.

Remark. From the perspective of an individual user, we assume the user has a local view, *i.e.*, she only observes the online actions her followers are exposed to through their feeds. Alternatively, if the user would have a global view, one could naturally opt for the following functional form for the intensities $\boldsymbol{\gamma}(t)$ of the followers' feeds:

$$\boldsymbol{\gamma}(t) = \boldsymbol{A}^T \boldsymbol{\lambda}(t) = \boldsymbol{A}^T \boldsymbol{\lambda}_0(t) + \boldsymbol{A}^T \boldsymbol{B} \int_0^t \kappa_\omega(t-s)\, d\boldsymbol{N}(s). \tag{4}$$

However, for simplicity, from the perspective of an individual user, we will assume the user has a local view and thus consider the functional form defined by Eq. 1. Considering an individual user has a global view is an interesting, albeit challenging venue for future work.

3. A Local Problem: When-to-post

In this section, we take the perspective of an individual user and design an online algorithm to help her decide when-to-post to be *noticed*—achieve high *visibility*. To this aim, we first define our visibility measure and then derive a jump stochastic differential equation that links our measure to the user's posting intensity and the feeds' intensities due to other users her followers follow. Finally, we formally state the when-to-post problem for this visibility measure and address the problem from the perspective of stochastic optimal control of SDEs with jumps. For ease of exposition, we assume users can only take one type of exogenous and endogenous action—posting stories. Moreover, throughout the section, we refer to users who posts stories as *broadcasters*.

3.1 Visibility Definition and Dynamics

Given a broadcaster i and one of her followers j, we define the visibility function $r_{ij}(t)$ as the position or *rank* of the most recent story posted by i in j's feed by time t, which clearly depends on the feed ranking mechanism in the corresponding social network. Here, for simplicity, we assume each user's feed ranks stories in reverse chronological order[3]. However, our framework can be easily extended to any feed ranking mechanisms, as long as its rank dynamics can be expressed as a jump SDE[4].

Under the reverse chronological ordering assumption, position at time t is simply the number of stories that other broadcasters posted in j's feed from the time of the most recent story posted by i until t. Then, when a new story arrives to a user's feed, it appears at the top of the feed and the other stories are shifted down by one. If we denote the time of the most recent story posted by i by time t as $\tau_i(t) = \max\{t_k \in \mathcal{H}_i(t)\}$, then the visibility is formally defined as

$$r_{ij}(t) = M_{j\setminus i}(t) - M_{j\setminus i}(\tau_i(t)). \tag{5}$$

Note that, if the last story posted by i is at the top of j's feed at time t, then $r_{ij}(t) = 0$.

3. At the time of writing, Weibo rank stories in reverse chronological order by default and Twitter and Facebook allow choosing such an ordering.
4. This would require either having access to the corresponding feed ranking mechanism or reverse engineering it, which is out of the scope of this work.

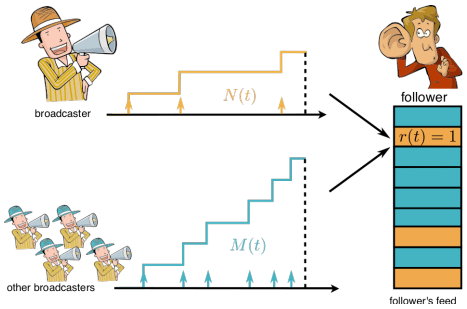


Figure 1: The dynamics of visibility. A broadcaster i posts $N_i(t) = N(t)$ stories with intensity $\lambda_i(t) = u(t)$. Her stories accumulate in her follower j's feed, competing for attention with $M_{j\setminus i}(t) = M(t)$ other stories posted by other broadcasters user j follows with intensity $\gamma_{j\setminus i}(t) = \gamma(t)$. The visibility function $r_{ij}(t) = r(t)$ is the position or rank of the most recent story posted by broadcaster i in the follower j's feed by time t.

Dynamics of visibility. Given a broadcaster i with broadcasting counting process $N_i(t)$ and one of her followers j with feed counting process due to other broadcasters $M_{j\setminus i}(t)$, the rank of i in j's feed $r_{ij}(t)$ satisfies the following equation:

$$r_{ij}(t+dt) = \underbrace{(r_{ij}(t)+1)dM_{j\setminus i}(t)(1-dN_i(t))}_{\text{1. Increases by one}} + \underbrace{0}_{\text{2. Becomes zero}} + \underbrace{r_{ij}(t)(1-dM_{j\setminus i}(t))(1-dN_i(t))}_{\text{3. Remains the same}},$$

where each term models one of the three possible situations:

1. The other broadcasters post a story in $(t, t+dt]$, $dM_{j\setminus i}(t) = 1$, and broadcaster i does not post, $dN_i(t) = 0$. The position of the last story posted by i in j's feed steps down by one, *i.e.*, $r_{ij}(t+dt) = r_{ij}(t) + 1$.
2. Broadcaster i posts a story in $(t, t+dt]$, $dN_i(t) = 1$, and the other broadcasters do not, $dM_{j\setminus i}(t) = 0$. No matter what the previous rank was, the new rank is $r_{ij}(t+dt) = 0$ since the newly posted story appears at the top of j's feed.
3. No one posts any story in $(t, t+dt]$, $dN_i(t) = 0$ and $dM_{j\setminus i}(t) = 0$. The rank remains the same, *i.e.*, $r_{ij}(t+dt) = r_{ij}(t)$.

We skip the case in which $M_{j\setminus i}(t) = 1$ and $dN_i(t) = 1$ in the same time interval $(t, t+dt]$ because, by the Blumenthal zero-one law (Blumenthal, 1957), it has zero probability. Now, by rearranging terms and using that $dN_i(t)dM_{j\setminus i}(t) = 0$, we uncover the following jump SDE for the visibility (or rank) dynamics:

$$dr_{ij}(t) = -r_{ij}(t)\, dN_i(t) + dM_{j\setminus i}(t), \tag{6}$$

where $dr_{ij}(t) = r_{ij}(t+dt) - r_{ij}(t)$. Figure 1 illustrates the concept of visibility for one broadcaster and one follower.

3.2 Problem Formulation

Given a broadcaster i and her followers $\mathcal{N}(i)$, our goal is to find the optimal posting intensity $\lambda_i(t) = u(t)$ that minimizes the expected value of a particular nondecreasing convex loss

function $\ell(\boldsymbol{r}(t), u(t))$ of the broadcaster's visibility on each of her follower's feed, $\boldsymbol{r}(t) = [r_{ij}(t)]_{j \in \mathcal{N}(i)}$, and the intensity itself, $u(t)$, over a time window $(t_0, t_f]$, *i.e.*,

$$\begin{aligned} \underset{u(t_0,t_f]}{\text{minimize}} \quad & \mathbb{E}_{(N_i, \boldsymbol{M}_{\setminus i})(t_0,t_f]} \left[\phi(\boldsymbol{r}(t_f)) + \int_{t_0}^{t_f} \ell(\boldsymbol{r}(s), u(s)) ds \right] \\ \text{subject to} \quad & u(t) \geq 0 \quad \forall t \in (t_0, t_f], \end{aligned} \tag{7}$$

where $u(t_0, t_f]$ denotes the broadcaster i's intensity from t_0 to t_f, the expectation is taken over all possible realizations of the counting processes associated to the broadcaster i and all other broadcasters from t_0 to t_f, denoted as $(N_i, \boldsymbol{M}_{\setminus i})(t_0, t_f]$, and $\phi(\boldsymbol{r}(t_f))$ is an arbitrary penalty function[5]. Here, by considering a loss which is nondecreasing in the rank $r(t)$ and the posting intensity $u(t)$, we penalize times when the position of the most recent story on each of the follower's feeds is high (*i.e.*, the most recent story does not stay *at the top*) and we limit the posting intensity, which in turn limits the number of stories the broadcaster can post. Finally, note that the optimal intensity $u(t)$ for broadcaster i at time t may depend on the visibility $\boldsymbol{r}(t)$ with respect to each of her followers and, thus, the associated counting process $N_i(t)$ may be doubly stochastic.

3.3 Stochastic Optimal Control Algorithm

In this section, we tackle the when-to-post problem defined by Eq. 7 from the perspective of stochastic optimal control of jump SDEs (Hanson, 2007). More specifically, we first derive a solution to the problem considering only one follower, provide an efficient practical implementation of the solution and then generalize it to the case of multiple followers. We conclude this section by deriving a solution to the problem given an (idealized) oracle that knows the times of all stories in the followers' feeds *a priori*, which we will use as baseline in our experiments (refer to Section 5.1).

Optimizing for one follower. Given a broadcaster i with $N_i(t) = N(t)$ and $\lambda_i(t) = u(t)$ and only one of her followers j with $M_{j \setminus i}(t) = M(t)$ and $\gamma_{j \setminus i}(t) = \gamma(t)$, we can rewrite the when-to-post problem defined by Eq. 7 as

$$\begin{aligned} \underset{u(t_0,t_f]}{\text{minimize}} \quad & \mathbb{E}_{(N,M)(t_0,t_f]} \left[\phi(r(t_f)) + \int_{t_0}^{t_f} \ell(r(s), u(s))\, ds \right] \\ \text{subject to} \quad & u(t) \geq 0 \quad \forall t \in (t_0, t_f], \end{aligned} \tag{8}$$

where, using Eqs. 1, 3 and 6, the dynamics of $M(t)$ and $r(t)$ are given by the following two coupled jump SDEs:

$$\begin{aligned} d\gamma(t) &= \left[\gamma_0'(t) + \omega\gamma_0(t) - \omega\gamma(t)\right] dt + \alpha\, dM(t), \\ dr(t) &= -r(t)\, dN(t) + dM(t), \end{aligned} \tag{9}$$

with initial conditions $r(t_0) = r_0$ and $\gamma(t_0) = \gamma_0$, and the dynamics of $N(t)$ are given by the intensity $u(t)$ that we aim to optimize.

5. The final penalty function $\phi(\boldsymbol{r}(t_f))$ is necessary to derive the optimal intensity $u^*(t)$ in Section 3.3. However, the actual optimal intensity $u^*(t)$ will not depend on the particular choice of terminal condition.

Next, we will define a novel optimal cost-to-go function that accounts for the above unique aspects of our problem, showing that the Bellman's principle of optimality still follows, and finally find the optimal solution using the corresponding Hamilton-Jacobi-Bellman (HJB) equation.

Definition 2 *The optimal cost-to-go $J(r(t), \gamma(t), t)$ is defined as the minimum of the expected value of the cost of going from state $r(t)$ with intensity $\gamma(t)$ at time t to a final state at time t_f,* i.e.*,*

$$J(r(t), \gamma(t), t) = \min_{u(t,t_f]} \mathbb{E}_{(N,M)(t,t_f]} \left[\phi(r(t_f)) + \int_t^{t_f} \ell(r(s), u(s))\, ds \right], \tag{10}$$

where the expectation is taken over all trajectories of the control and noise jump process, N and M, in the $(t, t_f]$ interval, given the initial values of $r(t)$, $\gamma(t)$ and $u(t)$.

To find the optimal control $u(t, t_f]$ and cost-to-go J, we break the problem into smaller subproblems, using the Bellman's principle of optimality (Bertsekas, 1995), which the above definition allows (proven in Appendix B):

Lemma 3 (Bellman's Principle of Optimality) *The optimal cost satisfies the following recursive equation:*

$$J(r(t), \gamma(t), t) = \min_{u(t,t+dt]} \left\{ \mathbb{E}\left[J(r(t+dt), \gamma(t+dt), t+dt) \right] + \ell(r(t), u(t))\, dt \right\}, \tag{11}$$

where the expectation is taken over all trajectories of the control and noise jump processes, N and M, in $(t, t+dt]$. Then, we use the Bellman's principle of optimality to derive a partial differential equation on J, often called the Hamilton-Jacobi-Bellman (HJB) equation (Hanson, 2007). To do so, we first assume J is continuous and then rewrite Eq. 11 as

$$\begin{aligned} J(r(t), \gamma(t), t) &= \min_{u(t,t+dt]} \left\{ \mathbb{E}\left[J(r(t), \gamma(t), t) + dJ(r(t), \gamma(t), t) \right] + \ell(r(t), u(t))\, dt \right\} \\ 0 &= \min_{u(t,t+dt]} \left\{ \mathbb{E}\left[dJ(r(t), \gamma(t), t) \right] + \ell(r(t), u(t))\, dt \right\}, \end{aligned} \tag{12}$$

where $dJ(r(t), \gamma(t), t) = J(r(t+dt), \gamma(t+dt), t+dt) - J(r(t), \gamma(t), t)$. Then, we differentiate J with respect to time t, $r(t)$ and $\gamma(t)$ using to following Lemma (proven in the Appendix C):

Lemma 4 *The differential $dJ(r(t), \gamma(t), t)$ of the cost-to-go function $J(r(t), \gamma(t), t)$, as defined by Eq. 10, is given by*

$$\begin{aligned} dJ(r(t), \gamma(t), t) = {} & J_t(r(t), \gamma(t), t) dt + \left[\gamma_0'(t) + \omega\gamma_0(t) - \omega\gamma(t)\right] J_\gamma(r(t), \gamma(t), t) dt \\ & + \left[J(0, \gamma(t), t) - J(r(t), \gamma(t), t) \right] dN(t) \\ & + \left[J(r(t) + 1, \gamma(t) + \alpha, t) - J(r(t), \gamma(t), t) \right] dM(t), \end{aligned}$$

where J_t and J_γ are derivatives of J with respect to t and γ, respectively.

Algorithm 1: RedQueen for fixed s, q and one follower.

Input: parameters q and s ;
Output: Returns time for the next post ;
$t \leftarrow \infty$; $\tau \leftarrow othersNextPost(\,)$;
while $\tau < t$ **do**
  $\Delta \sim Sample(\sqrt{s/q})$;
  $t \leftarrow \min(t,\, \tau + \Delta)$;
  $\tau \leftarrow othersNextPost(\,)$;
return t;

Next, if we plug in the above equation in Eq. 12, it follows that

$$\begin{aligned} 0 = \min_{u(t,t+dt]} \Big\{ & J_t(r(t),\gamma(t),t)dt + \left[\gamma_0'(t) + \omega\gamma_0(t) - \omega\gamma(t)\right] J_\gamma(r(t),\gamma(t),t)dt \\ & + [J(0,\gamma(t),t) - J(r(t),\gamma(t),t)]\mathbb{E}[dN(t)] \\ & + [J(r(t)+1,\gamma(t)+\alpha,t) - J(r(t),\gamma(t),t)]\mathbb{E}[dM(t)] + \ell(r(t),u(t))\,dt \Big\}. \end{aligned}$$

Now, using $\mathbb{E}[dN(t)] = u(t)dt$ and $\mathbb{E}[dM(t)] = \gamma(t)dt$, and rearranging terms, the HJB equation follows:

$$\begin{aligned} 0 = {} & J_t(r(t),\gamma(t),t) + \left[\gamma_0'(t) + \omega\gamma_0(t) - \omega\gamma(t)\right] J_\gamma(r(t),\gamma(t),t) \\ & + [J(r(t)+1,\gamma(t)+\alpha,t) - J(r(t),\gamma(t),t)]\gamma(t) \\ & + \min_{u(t,t+dt]} \ell(r(t),u(t)) + [J(0,\gamma(t),t) - J(r(t),\gamma(t),t)]u(t). \end{aligned} \tag{13}$$

To be able to continue further, we need to define the loss ℓ and the penalty ϕ. Following the literature on stochastic optimal control (Hanson, 2007), we consider the following quadratic forms, which will turn out to be a tractable choice[6]:

$$\phi(r(t_f)) = \frac{1}{2} r^2(t_f) \quad \text{and} \quad \ell(r(t),u(t)) = \frac{1}{2} s(t)\, r^2(t) + \frac{1}{2} q\, u^2(t),$$

where $s(t)$ is a time significance function $s(t) \geq 0$, which favors some periods of times (*e.g.*, times in which the follower is online[7]), and q is a given parameter, which trade-offs visibility and number of broadcasted posts.

Under these definitions, we take the derivative with respect to $u(t)$ of Eq. 13 and uncover the relationship between the optimal intensity and the optimal cost:

$$u^*(t) = q^{-1}\left[J(r(t),\gamma(t),t) - J(0,\gamma(t),t)\right]. \tag{14}$$

Finally, we substitute the above expression in Eq. 13 and find that the optimal cost J needs to satisfy the following nonlinear differential equation:

$$\begin{aligned} 0 = {} & J_t(r(t),\gamma(t),t) + \frac{1}{2} s(t)\, r^2(t) - \frac{1}{2} q^{-1}\left[J(r(t),\gamma(t),t) - J(0,\gamma(t),t)\right]^2 \\ & + \left[\gamma_0'(t) + \omega\gamma_0(t) - \omega\gamma(t)\right] J_\gamma(r(t),\gamma(t),t) + [J(r(t)+1,\gamma(t)+\alpha,t) - J(r(t),\gamma(t),t)]\gamma(t) \end{aligned} \tag{15}$$

6. Considering other losses with a specific semantic meaning (*e.g.*, $\mathbb{I}(r(t) \leq k)$) is a challenging direction for future work.
7. Such information may be hidden but one can use the followers' posting activity or geographic location as a proxy (Karimi et al., 2016).

with $J(r(t_f),\gamma(t_f),t_f) = \phi(r(t_f))$ as the terminal condition. The following technical lemma provides us with a solution to the above equation (proven in Appendix D):

Lemma 5 *In the space of m-degree polynomials, the following polynomial is the only solution to Eq. 15:*

$$J(r(t),\gamma(t),t) = f(t) + \sqrt{s(t)q}\, r(t) + \sum_{j=1}^{m} g_j(t)\gamma^j(t),$$

where $f(t)$ and $g_j(t)$ are time-varying functions which can be found by solving a linear system of differential equations.

Given the above Lemma and Eq. 14, the optimal intensity is readily given by following theorem:

Theorem 6 *The optimal intensity for the when-to-post problem defined by Eq. 8 with quadratic loss and penalty function is given by $u^*(t) = \sqrt{s(t)/q}\, r(t)$.*

The optimal intensity only depends on the position of the most recent post by broadcaster i in her follower's feed and, thus, allows for a very efficient procedure to sample posting times, which exploits the superposition theorem (Kingman, 1992). The key idea is as follows: at any given time t, we can view the process defined by the optimal intensity as a superposition of $r(t)$ inhomogeneous Poisson processes with intensity $\sqrt{s(t)/q}\, r(t)$ which starts at jumps of the rank $r(t)$, and find the next sample by computing the minimum across all samples from these processes. Algorithm 1 summarizes our (sampling) method, which we name RedQueen (Carroll, 1917). Within the algorithm, $Sample(\sqrt{s/q})$ samples from a Poisson process with intensity $\sqrt{s/q}$ and $othersNextPost(\,)$ returns the time of the next event by other broadcasters in the followers' feeds, once the event happens. In practice, we only need to know if the event happens before we post. Remarkably, it only needs to sample $M(t_f)$ times from a (in)homogeneous Poisson process (if significance is time varying) and requires $O(1)$ space.

Optimizing for multiple followers. Given a broadcaster i with $N_i(t) = N(t)$ and $\lambda_i(t) = u(t)$ and her followers $\mathcal{N}(i)$ with $\boldsymbol{M}_{\setminus i}(t) = \boldsymbol{M}(t)$ and $\boldsymbol{\gamma}_{\setminus i}(t) = \boldsymbol{\gamma}(t)$, we can write the dynamics of $\boldsymbol{M}(t)$ and $\boldsymbol{r}(t)$, which we need to solve Eq. 7, using Eqs. 1, 3 and 6:

$$\begin{aligned} d\boldsymbol{\gamma}(t) &= \left[\boldsymbol{\gamma}_0'(t) + w\boldsymbol{\gamma}_0(t) - w\boldsymbol{\gamma}(t)\right] dt + \boldsymbol{D}\, d\boldsymbol{M}(t) \\ d\boldsymbol{r}(t) &= -\boldsymbol{r}(t)\, dN(t) + d\boldsymbol{M}(t). \end{aligned}$$

Then, consider the following quadratic forms for the loss ℓ and the penalty ϕ:

$$\begin{aligned} \phi(\boldsymbol{r}(t_f)) &= \sum_{i=1}^{n} \frac{1}{2} r_i^2(t_f) \\ \ell(\boldsymbol{r}(t),u(t),t) &= \sum_{i=1}^{n} \frac{1}{2} s_i(t) r_i^2(t) + \frac{1}{2} q\, u^2(t). \end{aligned}$$

Algorithm 2: Optimal posting times with an oracle.

Input: Initial state r_0, interval widths $w_1, \ldots, w_{m+1}$, parameter q and significance $s(t) = s$;
Output: Overall cost $J(r_0, 0)$, optimal control $u_0^*, \ldots, u_m^*$;
for $r \leftarrow r_0 + m$ **to** 0 **do**
  $J(r, m+1) \leftarrow \frac{1}{2} r^2$;
for $k \leftarrow m$ **to** 0 **do**
  for $r \leftarrow r_0 + k - 1$ **to** 0 **do**
    $J(r,k) = \min\{\frac{1}{2}q + J(0, k+1),\ \frac{1}{2} s w_{k+1} (r+1)^2 + J(r+1, k+1)\}$;
for $k \leftarrow 0$ **to** m **do**
  if $\frac{1}{2}q + J(0, k+1) < \frac{1}{2} s w_{k+1} (r_k + 1)^2 + J(r_k + 1, k+1)$ **then**
    $u_k^* \leftarrow 1$; $r_{k+1} \leftarrow 0$;
  else
    $u_k^* \leftarrow 0$; $r_{k+1} \leftarrow r_k + 1$;
return $J(r_0, 0)$, $u_0^*, \ldots, u_m^*$

where $s_i(t)$ is the time significance function for follower i, as defined above, and q is a given parameter. Proceeding similarly as in the case of one follower, we can show that:

$$u^*(t) = \sum_{i=1}^{n} \sqrt{s_i(t)/q}\, r_i(t), \tag{16}$$

which only depends on the position of the most recent post by broadcaster i in her followers' feeds. Finally, we can readily adapt RedQueen (Algorithm 1) to efficiently sample the posting times using the above intensity—it only needs to sample $|\cup_{j \in \mathcal{N}(i)} \mathcal{F}_{j \setminus i}(t_f)|$ values from an (in)homogeneous Poisson process (if significance is time varying) and requires $O(|\mathcal{N}(i)|)$ space.

Optimizing with an oracle. In this section, we consider a broadcaster i with $N_i(t) = N(t)$ and $\lambda_i(t) = u(t)$, only one of her followers j with $M_{j \setminus i}(t) = M(t)$, and a constant significance $s(t) = s$. The derivation can be easily adapted to the case of multiple followers and time-varying significance.

Suppose there is an (idealized) oracle that reveals $M(t)$ from t_0 to t_f, *i.e.*, the history $\mathcal{F}_{j \setminus i}(t_f) = \mathcal{F}(t_f)$ is given, and $M(t_f) = |\mathcal{F}(t_f)| = m$. Then, we can rewrite Eq. 7 as

$$\begin{aligned} \underset{u(t_0, t_f]}{\text{minimize}} \quad & \mathbb{E}_{N(t_0,t_f]} \left[\phi(r(t_f)) + \int_{t_0}^{t_f} \ell(r(s), u(s))\, ds \right] \\ \text{subject to} \quad & u(t) \geq 0 \quad \forall t \in (t_0, t_f], \end{aligned}$$

where the expectation is only taken over all possible realizations of the counting process $N(t_0, t_f]$ since $M(t_0, t_f]$ is revealed by the oracle and, thus, deterministic.

Similar to the previous sections, assume the loss ℓ and penalty ϕ are quadratic. The best times for user i to post will always coincide with one of the times in $\mathcal{F}(t_f)$ since, given a posting time $\tau_i \in (t_k, t_{k+1})$, where $t_k, t_{k+1} \in \mathcal{F}(t_f)$, one can reduce the cost by $(1/2)q(\tau_i - t_k) r^2(t_k)$ by choosing instead to post at t_k. As a consequence, we can discretize the dynamics of $r(t)$ in times $\mathcal{F}(t_f)$, and write $r_{k+1} = r_k + 1 - (r_k + 1)u_k$, where $r_k = r(t_k^-)$,

$u_k = u(t_k^+) \in \{0, 1\}$, $t_k \in \mathcal{F}(t_f)$. We can easily see that r_k is bounded by $0 \leq r_k < r_0 + m$. Similarly, we can derive the optimal cost-to-go in discrete-time as

$$J(r_k, k) = \min_{u_k, \ldots, u_m} \frac{1}{2} r_{m+1}^2 + \sum_{i=k}^{m} \frac{1}{2} q\, w_{i+1}\, r_{i+1}^2 + \frac{1}{2} s\, u_i^2,$$

where $w_i = t_i - t_{i-1}$. Next, we can break the minimization and use Bellman's principle of optimality,

$$J(r_k, k) = \min_{u_k} \frac{1}{2} q\, w_{k+1}\, r_{k+1}^2 + \frac{1}{2} s\, u_k^2 + J(r_{k+1}, k+1),$$

and, since $u_k \in \{0, 1\}$, the above recursive equation can be written as

$$J(r_k, k) = \min \left\{ \frac{1}{2} s + J(0, k+1),\ \frac{1}{2} q\, w_{k+1} (r_k + 1)^2 + J(r_k + 1, k+1) \right\}.$$

Finally, we can find the optimal control u_k^*, $k = 0, \ldots, m$ and cost $J(r_0, 0)$ by backtracking from the terminal condition $J(r_{m+1}, m+1) = r_{m+1}^2/2$ to the initial state r_0, as summarized in Algorithm 2, which can be adapted to multiple followers. Note that, in this case, the optimal strategy is not stochastic and consists of a set of optimal posting times, as one could have guessed. However, for multiple followers, the complexity of the algorithm is $O(m^2)$, where $m = |\cup_{j \in \mathcal{N}(i)} \mathcal{F}_{j \setminus i}(t_f)|$.

4. A Global Problem: Activity Maximization

In this section, we take the perspective of an entire online social networking site and design an online algorithm to find how much should we incentivize a small number of influential users to post more over time to increase the overall number of additional posts, shares or replies in the site. To this aim, we first describe how to formally model such incentive mechanism in social networks. Then, we state the activity shaping problem and address the problem from the perspective of stochastic optimal control of SDEs with jumps, similarly as in Section 3.

4.1 Triggering Additional Endogenous Actions

Given a social network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = n$ users, we trigger additional endogenous user actions by directly incentivizing (*e.g.*, paying) for $\boldsymbol{P}(t)$ actions, where $P_i(t)$ counts the number of directly incentivized actions taken by user $i \in \mathcal{V}$ before time t and we can characterize $\boldsymbol{P}(t)$ by means of n intensity functions $\boldsymbol{u}(t)$, *i.e.*, $\mathbb{E}[d\boldsymbol{P}(t)] = \boldsymbol{u}(t)dt$. Then, if we assume the strength of influence $\boldsymbol{B}$ between users is the same both for organic and incentivized actions[8], as previous work (Farajtabar et al., 2014, 2016), we can rewrite the

8. This assumption seems reasonable in some scenarios, *e.g.*, it is often difficult to notice whether an influential user is being paid for posting a message. However, one could relax this assumption by considering a different influence matrix $C \neq B$ for the additional endogenous actions in Eq. 17, change the SDE, Eq. 18 and HJB, Eq. 22 accordingly, and derived the new optimal control in Theorem 11.

users' intensities $\boldsymbol{\lambda}(t)$, given by Eq. 2, as

$$\boldsymbol{\lambda}(t) = \boldsymbol{\lambda}_0(t) + \boldsymbol{B}\int_0^t \kappa_\omega(t-s)\, d\boldsymbol{N}(s) + \underbrace{\boldsymbol{B}\int_0^t \kappa_\omega(t-s)\, d\boldsymbol{P}(s)}_{\text{Additional endogenous actions}}. \tag{17}$$

Here, note that treating directly incentivized actions as a different counting process $\boldsymbol{P}(t)$ prevents the intensities $\boldsymbol{\lambda}(t)$ from including their intensity but instead including only the intensity of their follow-ups (*i.e.*, additional endogenous actions), which we aim to maximize. Then, it is easy to derive the following alternative representation, similarly as in Proposition 1, which we will use in our stochastic optimal control algorithm:

Proposition 7 *Let $\boldsymbol{N}(t)$ and $\boldsymbol{P}(t)$ be two multidimensional counting processes with associated intensities $\boldsymbol{\lambda}(t)$, given by Eq. 17, and $\boldsymbol{u}(t)$, respectively. Then, the dynamics of the intensity $\boldsymbol{\lambda}(t)$ can be expressed using the following jump SDEs:*

$$d\boldsymbol{\lambda}(t) = \left[\boldsymbol{\lambda}_0'(t) + \omega\boldsymbol{\lambda}_0(t) - \omega\boldsymbol{\lambda}(t)\right] dt + \boldsymbol{B}\, d\boldsymbol{N}(t) + \boldsymbol{B}\, d\boldsymbol{P}(t) \tag{18}$$

with the initial condition $\boldsymbol{\lambda}(0) = \boldsymbol{\lambda}_0(0)$.

In the remainder of the section, for ease of exposition, we assume $\boldsymbol{\lambda}_0(t) = \boldsymbol{\lambda}_0$ and thus $\boldsymbol{\lambda}_0'(t) = 0$.

4.2 Problem Formulation

Given a social network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = n$ users, our goal is to find the optimal users' intensities for directly incentivized actions $\boldsymbol{u}(t)$ (for short, control intensities) that minimize the expected value of a particular loss function $\ell(\boldsymbol{u}(t), \boldsymbol{\lambda}(t))$ of the control intensities and the users' intensities for not directly incentivized actions over a time window $(t_0, t_f]$, *i.e.*,

$$\begin{aligned} &\underset{\boldsymbol{u}(t_0,t_f]}{\text{minimize}} \quad \mathbb{E}_{(\boldsymbol{N},\boldsymbol{P})(t_0,t_f]}\left[\phi(\boldsymbol{\lambda}(t_f)) + \int_{t_0}^{t_f} \ell(\boldsymbol{\lambda}(s), \boldsymbol{u}(s))\, ds\right] \\ &\text{subject to} \quad u_i(t) \geq 0,\ \forall t \in (t_0, t_f],\ i = 1, \ldots, n \end{aligned} \tag{19}$$

where $\boldsymbol{u}(t_0, t_f]$ denotes the control intensities from t_0 to t_f, the dynamics of $\boldsymbol{N}(t)$ are given by Eq. 18, and the expectation is taken over all possible realizations of the two counting processes $\boldsymbol{N}(t)$ and $\boldsymbol{P}(t)$ during interval $(t_0, t_f]$. Here, by considering a loss that is nonincreasing (nondecreasing) with respect to the intensities $\boldsymbol{\lambda}(t)$ ($\boldsymbol{u}(t)$), we will trade-off number of directly incentivized actions and number of additional endogenous actions. Finally, note that the optimal intensities $\boldsymbol{u}(t)$ at time t may depend on the intensities $\boldsymbol{\lambda}(t)$ and thus the associated counting process $\boldsymbol{P}(t)$ may be doubly stochastic.

4.3 Stochastic Optimal Control Algorithm

In this section, we proceed similarly as in Section 3.3 and tackle the activity maximization problem defined by Eq. 19 from the perspective of stochastic optimal control of SDEs with jumps (Hanson, 2007). More specifically, we first define a novel optimal cost-to-go function specially designed for this global problem and then derive and solve the corresponding HJB equation to find the optimal control intensities.

Definition 8 *The optimal cost-to-go $J(\boldsymbol{\lambda}(t), t)$ is defined as the minimum of the expected value of the cost of going from the state with intensity $\boldsymbol{\lambda}(t)$ at time t to the final state at time t_f,* i.e.,

$$J(\boldsymbol{\lambda}(t), t) = \min_{\boldsymbol{u}(t,t_f]} \mathbb{E}_{(\boldsymbol{N},\boldsymbol{P})(t,t_f]}\left[\phi(\boldsymbol{\lambda}(t_f)) + \int_t^{t_f} \ell(\boldsymbol{\lambda}(s), \boldsymbol{u}(s))\, ds\right], \tag{20}$$

where the expectation is taken over all trajectories of the counting processes $\boldsymbol{N}$ and $\boldsymbol{P}$ in the $(t, t_f]$ interval, given the initial values of $\boldsymbol{\lambda}(t)$ and $\boldsymbol{u}(t)$.

Next, we use the Bellman's principle of optimality (Bertsekas, 1995), which the above cost-to-go $J(\boldsymbol{\lambda}(t), t)$ also allows for (proof is similar to that of Lemma 3)), to break the problem into smaller subproblems and rewrite Eq. 20 as

$$\begin{aligned} J(\boldsymbol{\lambda}(t), t) &= \min_{\boldsymbol{u}(t,t+dt]} \left\{\mathbb{E}\left[J(\boldsymbol{\lambda}(t), t) + dJ(\boldsymbol{\lambda}(t), t)\right] + \ell(\boldsymbol{u}(t), \boldsymbol{\lambda}(t))\, dt\right\} \\ 0 &= \min_{\boldsymbol{u}(t,t+dt]} \left\{\mathbb{E}\left[dJ(\boldsymbol{\lambda}(t), t)\right] + \ell(\boldsymbol{u}(t), \boldsymbol{\lambda}(t))\, dt\right\}, \end{aligned} \tag{21}$$

where $dJ(\boldsymbol{\lambda}(t), t) = J(\boldsymbol{\lambda}(t+dt), t+dt) - J(\boldsymbol{\lambda}(t), t)$. Then, we explicitly differentiate J with respect to time t and $\lambda(t)$ using the following Lemma (proven in the Appendix E):

Lemma 9 *The differential $dJ(\boldsymbol{\lambda}(t), t)$ of the cost-to-go function $J(\boldsymbol{\lambda}(t), t)$, as defined by Eq. 20, is given by*

$$\begin{aligned} J_t(\boldsymbol{\lambda}(t), t)\, dt + [w\boldsymbol{\lambda}_0 - w\boldsymbol{\lambda}(t)]^T \nabla_{\boldsymbol{\lambda}} J(\boldsymbol{\lambda}(t), t)\, dt + \sum_i \left[J(\boldsymbol{\lambda}(t) + \boldsymbol{b}_i, t) - J(\boldsymbol{\lambda}(t), t)\right] dN_i(t) \\ + \sum_i \left[J(\boldsymbol{\lambda}(t) + \boldsymbol{b}_i, t) - J(\boldsymbol{\lambda}(t), t)\right] dP_i(t), \end{aligned}$$

where $\boldsymbol{b}_i$ is the i'th column of $\boldsymbol{B}$, J_t is derivative of J with respect to t and $\nabla_{\boldsymbol{\lambda}} J$ is the gradient of J with respect to $\boldsymbol{\lambda}(t)$.

Next, if we plug the above equation in Eq. 21 and use that $\mathbb{E}[N_i(t)] = \lambda_i(t)\, dt$ and $\mathbb{E}[P_i(t)] = u_i(t)\, dt$, the HJB equation follows:

$$0 = J_t(\boldsymbol{\lambda}(t), t) + [w\boldsymbol{\lambda}_0 - w\boldsymbol{\lambda}(t)]^T \nabla_{\boldsymbol{\lambda}} J(\boldsymbol{\lambda}(t), t) + \boldsymbol{\lambda}^T(t)\, \Delta_B J + \min_{\boldsymbol{u}(t)} \ell(\boldsymbol{u}(t), \boldsymbol{\lambda}(t)) + \boldsymbol{u}^T(t)\, \Delta_B J, \tag{22}$$

where $\Delta_B J$ denotes a vector whose i'th element is given by $(\Delta_B J)_i = J(\boldsymbol{\lambda}(t) + \boldsymbol{b}_i, t) - J(\boldsymbol{\lambda}(t), t)$.

To solve the above equation, we need to define the loss and penalty functions, ℓ and ϕ. Similarly as in Section 3, we consider the following quadratic forms, which will turn out to be a tractable choice:

$$\ell(\boldsymbol{u}(t), \boldsymbol{\lambda}(t)) = -\frac{1}{2}\boldsymbol{\lambda}^T(t)\, \boldsymbol{Q}\, \boldsymbol{\lambda}(t) + \frac{1}{2}\boldsymbol{u}^T(t)\, \boldsymbol{S}\, \boldsymbol{u}(t) \quad \text{and} \quad \phi(\boldsymbol{\lambda}(t_f)) = -\frac{1}{2}\boldsymbol{\lambda}^T(t_f)\, \boldsymbol{F}\, \boldsymbol{\lambda}(t_f),$$

where $\boldsymbol{Q}$, $\boldsymbol{F}$ and $\boldsymbol{S}$ are given symmetric matrices[9] with $q_{ij} \geq 0$, $f_{ij} \geq 0$ and $s_{ij} \geq 0$ for all $i, j \in [n]$. These matrices allow us to trade-off the number of directly incentivized actions

9. In practice, we will consider diagonal matrices.

Algorithm 3: CHESHIRE: it returns user i and time τ for the next incentivized action

1: **Initialization:**
2: Compute $\boldsymbol{H}(t)$ and $\mathbf{g}(t)$;
3: $\mathbf{u}(t) \leftarrow -\boldsymbol{S}^{-1}\big[\boldsymbol{B}^T(\boldsymbol{g}(t) + \boldsymbol{H}(t)\boldsymbol{\lambda}_0) + \frac{1}{2}\operatorname{diag}(\boldsymbol{B}^T\boldsymbol{H}(t)\boldsymbol{B})\big]$;
4: **General subroutine:**
5: $(i, \tau) \leftarrow \mathit{Sample}(\mathbf{u}(t))$;
6: $(j, s) \leftarrow \mathit{NextAction}(\,)$;
7: **while** $s < \tau$ **do**
8: $\quad \boldsymbol{\lambda}_N(t) \leftarrow \boldsymbol{B}\mathbf{e}_j\kappa_\omega(t - s)$;
9: $\quad \mathbf{u}_N(t) \leftarrow -\boldsymbol{S}^{-1}\boldsymbol{B}^T\boldsymbol{H}(t)\boldsymbol{\lambda}_N(t)$;
10: $\quad (k, r) \leftarrow \mathit{Sample}(\mathbf{u}_N(t))$;
11: $\quad$ **if** $r < \tau$ **then**
12: $\quad\quad \tau \leftarrow r$;
13: $\quad\quad i \leftarrow k$;
14: $\quad \mathbf{u}(t) \leftarrow \mathbf{u}(t) + \mathbf{u}_N(t)$;
15: $\quad (j, s) \leftarrow \mathit{NextAction}(\,)$;
16: $\boldsymbol{\lambda}_P(t) \leftarrow \boldsymbol{B}\mathbf{e}_i\kappa_\omega(t - \tau)$;
17: $\mathbf{u}_P(t) \leftarrow -\boldsymbol{S}^{-1}\boldsymbol{B}^T\boldsymbol{H}(t)\boldsymbol{\lambda}_P(t)$;
18: $\mathbf{u}(t) \leftarrow \mathbf{u}(t) + \mathbf{u}_P(t)$;
19: **return** (i, τ)

over time and the number of additional endogenous actions, both over time and at time t_f. Under these definitions, we can find the relationship between the optimal intensity and the optimal cost by solving the minimization in the HJB equation:

$$\underset{\boldsymbol{u}(t)}{\text{minimize}} \quad \boldsymbol{u}^T(t)\,\Delta_B J + \frac{1}{2}\boldsymbol{u}^T(t)\,\boldsymbol{S}\,\boldsymbol{u}(t) \quad \text{subject to} \quad u_i(t) \geq 0,\ i = 1, \ldots, n.$$

By taking the differentiation with respect to $\boldsymbol{u}(t)$, the solution of the unconstrained minimization is given by

$$\boldsymbol{u}^*(t) = -\boldsymbol{S}^{-1}\Delta_B J, \tag{23}$$

which is the same as the solution to the constrained problem given that $s_{ij} \geq 0$, by definition, and $(\Delta_B J)_i \leq 0$, as proved in the Appendix F. Then, we substitute Eq. 23 into Eq. 22 and find that the optimal cost J needs to satisfy the following partial differential equation:

$$0 = J_t(\boldsymbol{\lambda}(t), t) + [w\boldsymbol{\lambda}_0 - w\boldsymbol{\lambda}(t)]^T\nabla_{\boldsymbol{\lambda}}J(\boldsymbol{\lambda}(t), t) + \boldsymbol{\lambda}^T(t)\,\Delta_B J - \frac{1}{2}\boldsymbol{\lambda}^T(t)\,\boldsymbol{Q}\,\boldsymbol{\lambda}(t) - \frac{1}{2}\Delta_B J^T\boldsymbol{S}^{-1}\Delta_B J, \tag{24}$$

with $J(\boldsymbol{\lambda}(t_f), t_f) = \phi(\boldsymbol{\lambda}(t_f))$ as the terminal condition. The following lemma provides us with a solution to the above equation (proven in Appendix G):

Lemma 10 *In the space of m-degree degree polynomials, the following quadratic form is the only solution to Eq. 24:*

$$J(\boldsymbol{\lambda}(t), t) = f(t) + \boldsymbol{g}(t)^T\boldsymbol{\lambda}(t) + \frac{1}{2}\boldsymbol{\lambda}(t)^T\boldsymbol{H}(t)\boldsymbol{\lambda}(t).$$

where $\boldsymbol{g}(t)$ and $\boldsymbol{H}(t)$ can be found by solving the following differential equations:

$$\dot{\boldsymbol{H}}(t) = (\omega\boldsymbol{I} - \boldsymbol{B})^T\boldsymbol{H}(t) + \boldsymbol{H}(t)(\omega\boldsymbol{I} - \boldsymbol{B}) + \boldsymbol{H}(t)\boldsymbol{B}\boldsymbol{S}^{-1}\boldsymbol{B}^T\boldsymbol{H}(t) + \boldsymbol{Q}$$

$$\dot{\boldsymbol{g}}(t) = [\omega\boldsymbol{I} - \boldsymbol{B}^T + \boldsymbol{H}(t)\boldsymbol{B}\boldsymbol{S}^{-1}\boldsymbol{B}^T]\boldsymbol{g}(t) - \omega\boldsymbol{H}(t)\boldsymbol{\lambda}_0 + \frac{1}{2}\left[\boldsymbol{H}(t)\boldsymbol{B}\boldsymbol{S}^{-1} - \boldsymbol{I}\right]\operatorname{diag}(\boldsymbol{B}^T\boldsymbol{H}(t)\boldsymbol{B}).$$

In the above lemma, note that the first differential equation is a matrix Riccati differential equation, which can be solved using many well-known efficient numerical solvers (Garrett, 2013), and the second one is a first order differential equation which has closed form solution. Both equations are solved backward in time with final conditions $\boldsymbol{g}(t_f) = \mathbf{0}$ and $\boldsymbol{H}(t_f) = -\boldsymbol{F}$.

Finally, given the above cost-to-go function, the optimal intensity is given by the following theorem:

Theorem 11 *The optimal intensity for the activity maximization problem defined by Eq. 19 with quadratic loss and penalty function is given by*

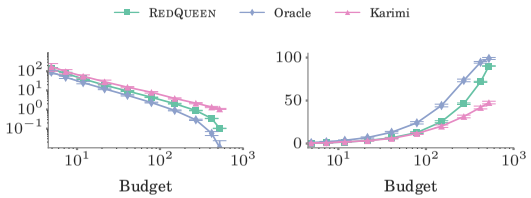
$$\boldsymbol{u}^*(t) = -\boldsymbol{S}^{-1}\big[\boldsymbol{B}^T\boldsymbol{g}(t) + \boldsymbol{B}^T\boldsymbol{H}(t)\boldsymbol{\lambda}(t) + \frac{1}{2}\operatorname{diag}(\boldsymbol{B}^T\boldsymbol{H}(t)\boldsymbol{B})\big].$$

Since the optimal intensity is linear in $\boldsymbol{\lambda}(t)$, it allows for an efficient procedure to sample the times of the users' directly incentivized actions: at any given time t, we can view the multidimensional control signal $\boldsymbol{u}(t)$ as a superposition of inhomogeneous multidimensional Poisson processes, one per non incentivized action, which start when the actions take place. Algorithm 3 summarizes our method, which we name CHESHIRE (Carroll, 1917).

Within the algorithm, *NextAction*() returns the time of the next (non directly incentivized) action in the network as well as the identity of the user who takes the action, once the action happens, $\boldsymbol{e}_j$ is an indicator vector where the entry corresponding to user j is 1, and *Sample*($\boldsymbol{u}(t)$) samples from a multidimensional inhomogeneous Poisson process with intensity $\boldsymbol{u}(t)$ and it returns both the sampled time and dimension (*i.e.*, user). Moreover, note that the algorithm initially plans a user i and time τ for the next directly incentivized action, however, if before τ, a new organic action takes place at $s < \tau$ and the intensity $\boldsymbol{\lambda}(t)$ changes, then the algorithm updates the user and time for the next directly incentivized action using the superposition principle. To sample from a multidimensional inhomogeneous Poisson process, there exist multiple methods *e.g.*, refer to Lewis and Shedler (1979). Finally, note that one can precompute most of the quantities the algorithm needs, *e.g.*, lines 2-3, $\boldsymbol{B}\boldsymbol{e}_j$ in line 8, and $\boldsymbol{S}^{-1}\boldsymbol{B}^T\boldsymbol{H}(t)$ in line 9. Given these precomputations, the algorithm only needs to perform $O(n)$ operations and sample $\mathbf{1}^T\boldsymbol{N}(t_f)$ times from an inhomogeneous Poisson process.

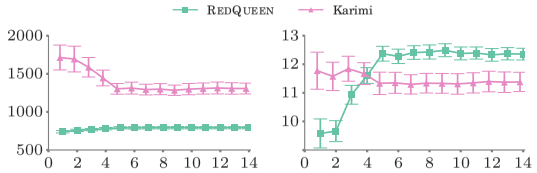
5. Experiments

In this section, we validate REDQUEEN (Algorithm 1) and CHESHIRE (Algorithm 3) using both synthetic and real data gathered from Twitter and compare their performance with several state of the art methods and competitive baselines He et al. (2012).



(a) Position over time (b) Time at the top

Figure 2: Optimizing for one follower. Performance of RedQueen in comparison with the oracle and the method by Karimi et al. (2016) against number of broadcasted events. The feeds counting processes $M(t)$ due to other broadcasters are Hawkes processes with $\gamma_0 = 10$, $\alpha = 1$ and $w = 10$. In all cases, the time horizon $t_f - t_0$ is chosen such that the number of stories posted by other broadcasters is ∼1000. Error bars are too small to be seen.



(a) Position over time (b) Time at the top

Figure 3: Optimizing for multiple followers. Performance of RedQueen in comparison with the method by Karimi et al. (2016) against number of followers. The feeds counting processes $M(t)$ due to other broadcasters follow piecewise constant intensities, where the intensity of each follower remains constant within each piece, it varies as a half-sinusoid across pieces and it starts with a random initial phase. The performance of both methods stays constant upon addition of more followers.

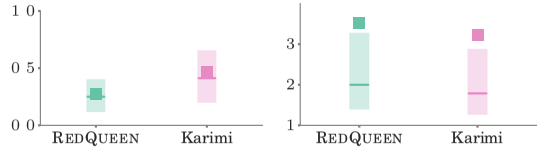
5.1 When-to-post

5.1.1 Experiments on synthetic data

Experimental setup. We evaluate the performance via two quality measures: position over time, $\int_0^T r(t)dt$, and time at the top, $\int_0^T \mathbb{I}(r(t)<1)dt$ and compare the performance of RedQueen against the oracle, described in Section 3.3, and the method by Karimi et al. (2016), which, to the best of our knowledge, is the state of the art. Unless otherwise stated, we set the significance $s_i(t) = 1, \forall t, i$ and use the parameter q to control the number of posts by RedQueen[10].

Optimizing for one follower. We first experiment with one broadcaster and one follower against an increasing number of events (or budget). We generate the counting processes

10. The expected number of posts by RedQueen are a decreasing function of q. Hence, we can use binary search to guess q and then use averaging over multiple simulation runs to estimate the number of posts made.



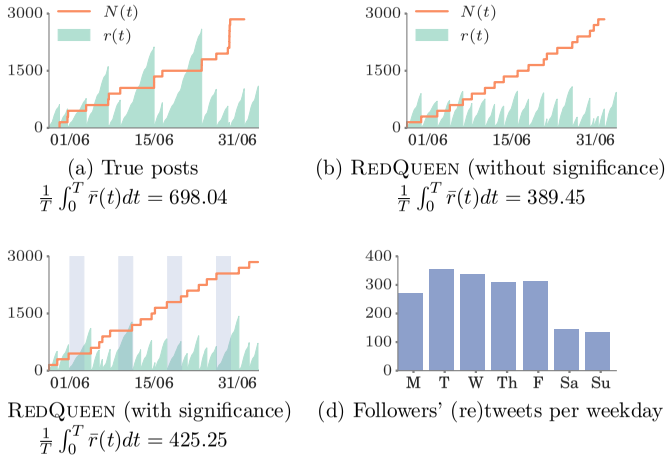
(a) Position over time (b) Time at the top

Figure 4: Performance of RedQueen and the method by Karimi et al. (2016) for 2000 Twitter users, picked at random. The solid horizontal line (square) shows the median (mean) quality measure, normalized with respect to the value achieved by the users' actual *true* posts, and the box limits correspond to the 25%-75% percentiles.

$M(t)$ due to other broadcasters using Hawkes processes, as defined by Eq. 1. We perform 10 independent simulation runs and compute the average and standard error (or standard deviation) of the quality measures. Fig. 2 summarizes the results, which show that our method: (i) consistently outperforms the method by Karimi et al. by large margins; (ii) achieves at most 3× higher position over time than the oracle as long as the budget is <30% of the posted events by all other broadcasters; and, (iii) achieves >40% of the value of time at the top that the oracle achieves.

Optimizing for multiple followers. Next, we experiment with one broadcaster and multiple followers. In this case, we generate the counting processes $M(t)$ due to other broadcasters using piece-wise constant intensity functions. More specifically, we simulate the feeds of each follower for 1 day, using 24 1-hour long segments, where the rate of posts remains constant per follower in each segment and the rate itself varies as a half-sinusoid (*i.e.*, from $\sin 0$ to $\sin \pi$), with each follower starting with a random initial phase. This experimental setup reproduces volume changes throughout the day across followers' feeds in different time-zones and closely resembles the settings in previous work (Karimi et al., 2016). The total number of posts by the RedQueen broadcaster is kept nearly constant and is used as the budget for the other baselines. Additionally, for Karimi's method, we provide as input the true empirical rate of tweets per hour for each user. Here, we do not compare with the oracle since, due to its quadratic complexity, it does not scale.

Figure 3 summarizes the results. In terms of position over time, RedQueen outperforms Karimi's method by a factor of 2. In terms of time at the top, RedQueen achieves ∼18% lower values than Karimi's method for 1-4 followers but ∼10% higher values for >5 followers. A potential reason Karimi's method performs best in terms of time at the top for a low number of followers and piecewise constant intensities is that, while the number of followers is low, there are segments which are clearly favorable and thus Karimi's method concentrates posts on those. However, as the number of followers increases, there are no clear favorable segments and advance planning does not give Karimi's method any advantage. On the other hand, RedQueen, due to its online nature, is able to adapt to transient variations in the feeds.



(a) True posts
$\frac{1}{T}\int_0^T \bar{r}(t)dt = 698.04$

(b) RedQueen (without significance)
$\frac{1}{T}\int_0^T \bar{r}(t)dt = 389.45$

(c) RedQueen (with significance)
$\frac{1}{T}\int_0^T \bar{r}(t)dt = 425.25$

(d) Followers' (re)tweets per weekday

Figure 5: A broadcaster chosen from real data. Panels compares the position over time $\bar{r}(t) = \sum_{i=0}^{\mathcal{N}} r(t)/\mathcal{N}$ (in green; lower is better) for the most recent tweet posted by a real user against the most recent one *posted* by a simulation run of RedQueen without and with significance. Here, the orange staircases represent the counts $N(t)$ of the tweets posted by the real user and RedQueen over time. The shaded area in panel (c) highlights weekends. We can see that RedQueen avoided tweeting on weekends, when the followers are less likely to be active/logged-in, as seen in panel (d).

5.1.2 Experiments on real data

Dataset description and experimental setup. We use data gathered from Twitter as reported in previous work (Cha et al., 2010), which comprises profiles of 52 million users, 1.9 billion directed follow links among these users, and 1.7 billion public tweets posted by the collected users. The follow link information is based on a snapshot taken at the time of data collection, in September 2009. Here, we focus on the tweets published during a two month period, from July 1, 2009 to September 1, 2009, in order to be able to consider the social graph to be approximately static, and sample 2000 users uniformly at random as broadcasters and record all the tweets they posted. Then, for each of these broadcasters, we track down their followers and record all the (re)tweets they posted as well as reconstruct their timelines by collecting all the (re)tweets published by the people they follow. We assign equal significance to each follower but filter out those who follow more than 500 people since, otherwise, they would dominate the optimal strategy. Finally, we tune q such that the total number of tweets posted by our method is equal to the number of tweets the broadcasters tweeted during the two month period (with a tolerance of 10%).

Solution quality. We only compare the performance of our method against the method by Karimi et al. (2016) since the oracle does not scale to the size of real data. Moreover,

for the method by Karimi et al., we divide the two month period into ten segments of approximately one week to fit the piecewise constant intensities of the followers' timelines, which the method requires. Fig. 4 summarizes the results by means of box plots, where position over time and time at the top are normalized with respect to the value achieved by the broadcasters' actual *true* posts during the two month period. That means, if $y = 1$, the optimized intensity achieves the same position over time or time at the top as the broadcaster's true posts. In terms of position over time and time at the top, RedQueen consistently outperforms competing methods by large margins and achieves 0.28× lower average position and 3.5× higher time at the top, in average, than the broadcasters' true posts—in fact, it achieves lower position over time (higher time at the top) for 100% (99.1%) of the users.

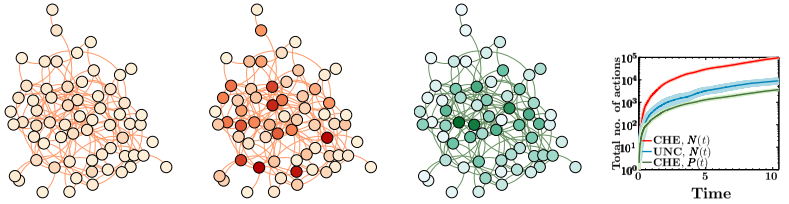
Time significance. We look at the actual broadcasting strategies for one real user and investigate the effect of a time varying significance. We define $s_i(t)$ to be the probability that follower i is online on that weekday, estimated empirically using the (re)tweets the follower posted as in Karimi et al. (2016). Fig. 5 compares the position over time for the most recent tweet posted by a real user against the most recent one *posted* by a simulation run of RedQueen with and without time varying significance. We can see that without significance information, RedQueen posts at nearly an even pace. However, when we supply empirically estimated significance, RedQueen avoids tweeting at times the followers are unlikely to be active, *i.e.*, the weekends, denoted by the shaded areas in panel (c) of Fig. 5. Due to this, the average position (maximum position) falls from 389.45 (1085.17) to 425.25 (1431.0), but is still lower than 698.04 (2597.9) obtained by the user's original posting schedule.

5.2 Activity Maximization

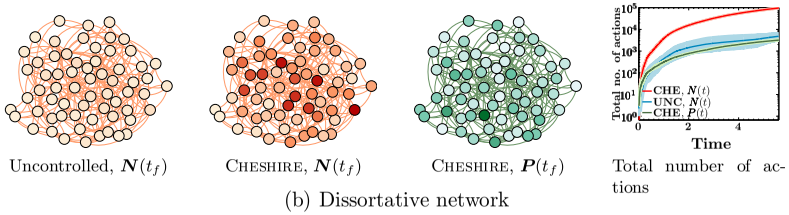
5.2.1 Experiments on Synthetic Data

In this section, we first shed light on Cheshire's sampling strategy in two small Kronecker networks (Leskovec et al., 2010) by recording, on the one hand, the number of directly incentivized actions per node and, on the other hand, the number of not directly incentivized actions per node in comparison with an uncontrolled setup. Then, we compare the performance of our method against several baselines and state of the art methods (Farajtabar et al., 2014, 2016) on a variety of large Kronecker networks and provides a scalability analysis.

Sampling strategy. We generate two small Kronecker networks with 64 nodes, a small core-periphery (parameter matrix $[0.96, 0.3; 0.3, 0.96]$) and a dissortative network (param. matrix $[0.3, 0.96; 0.96, 0.3]$), shown in Figure 6. For each network, we draw $\boldsymbol{B}$ from a uniform distribution $U(0, 10)$, $\boldsymbol{\lambda}_0$ also from a uniform distribution $U(0, 10)$ for 20% of the nodes and $\boldsymbol{\lambda}_0 = 0$ for the remaining 80%, and set $\omega = 16$, $t_0 = 0$ and $t_f = 5.5$. Then, we compare the number of actions $\boldsymbol{N}(t)$ over time under uncontrolled dynamics (Eq. 2; without Cheshire) and controlled dynamics (Eq. 17; with Cheshire). In both cases, we perform 20 simulation runs and sample non directly incentivized actions using Ogata's thinning algorithm (Ogata, 1981). In the case of controlled dynamics, we sample directly incentivized actions using Algorithm 3.



(a) Core-Periphery network

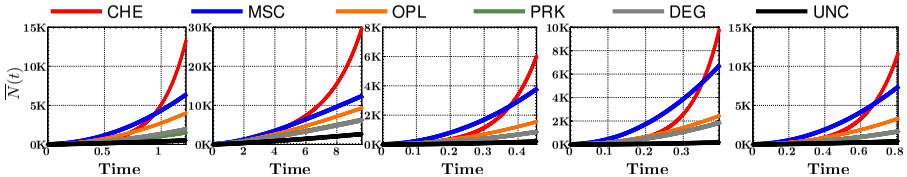


(b) Dissortative network

Figure 6: Activity on two 64-node networks, G_1 (top) and G_2 (bottom) under uncontrolled (Eq. 2; without Cheshire) and controlled (Eq. 17; with Cheshire) dynamics. The first and second columns visualize the final number of non incentivized actions $\boldsymbol{N}(t_f)$ under uncontrolled and controlled dynamics, where darker red corresponds to higher number of actions. The third column visualizes the final number of incentivized actions $\boldsymbol{P}(t_f)$ under controlled dynamics, where darker green corresponds to higher number of actions. The fourth column shows the temporal evolution of the number of incentivized and non incentivized actions across the whole networks for controlled and uncontrolled dynamics, where the solid line is the average across simulation runs and the shadowed region represents the standard error. By incentivizing a relatively small number of actions (∼3,600 actions), Cheshire is able to increase the overall number of (non incentivized) actions dramatically (∼96,000 vs ∼4,800 actions).

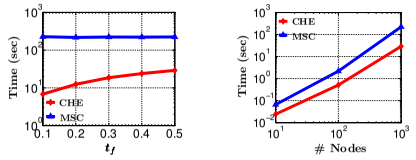
Figure 6 summarizes the results in terms of the number of non directly incentivized and incentivized actions, which show that: (i) a relatively small number of incentivized actions (fourth column; ∼3,600 actions) result in a dramatic increase in the overall number of (non directly incentivized) actions with respect the uncontrolled setup (fourth column; ∼96,000 vs ∼4,800 actions); (ii) the majority of the incentivized actions concentrate in a small set of influential nodes (third column; nodes in dark green); and, (iii) the variance of the overall number of actions (fourth column; shadowed regions) is consistently reduced when using Cheshire, in other words, the networks become more robust.

Performance. We experiment with five different types of large Kronecker networks (Leskovec et al., 2010) with 512 nodes: (i) homophily (assortative) networks (parameter matrix $[0.96, 0.3; 0.3, 0.96]$); (ii) heterophily (dissortative) networks ($[0.3, 0.96; 0.96, 0.3]$); (iii) random networks ($[0.7, 0.7; 0.7, 0.7]$); (iv) hierarchical networks ($[0.9, 0.1; 0.1, 0.9]$); and, (v)



(a) Core-periphery (b) Hierarchical (c) Homophily (d) Heterophily (e) Random

Figure 7: Performance over time of Cheshire against several competitors for several types of Kronecker networks. Performance is measured in terms of overall number of tweets $\bar{N}(t) = \sum_{u \in \mathcal{V}} \mathbb{E}[N_u(t)]$. In all cases, we tune the parameters Q, S and F such that the total number of incentivized tweets *posted* by our method is equal to the budget used in the competing methods and baselines.



(a) Running time vs t_f (b) Running time vs #nodes

Figure 8: Scalability of Cheshire against several competitors. Panel (a) shows the running time against the cut-off time t_f for a 1,000 node Kronecker network. Panel (b) shows the running time for several Kronecker networks of increasing size with $t_f = 0.5$. In both panels, the average degree per node is 10. The experiments are carried out in a single machine with 24 cores and 64 GB of main memory.

core-periphery networks ($[0.9, 0.5; 0.5, 0.3]$). For each network, we draw $\boldsymbol{\lambda}_0$ and $\boldsymbol{B}$ from a uniform distribution $U(0, 1)$ and set $\omega = 100$. We compare the performance of our algorithm, Cheshire, with two state of the art methods, which we denote as OPL (Farajtabar et al., 2014) and MSC (Farajtabar et al., 2016), and three baselines, which we denote as PRK, DEG, and UNC. OPL is a provably optimal algorithm for the offline setting and MSC is a suboptimal algorithm that discretizes the time window of interest in several rounds and, at any given round, it computes the control signal using the feedback from previous rounds to maximize the activity. PRK and DEG distribute the users' control intensities $\mathbf{u}(t)$ proportionally to the user's page rank and outgoing degree in the network, respectively. UNC simply considers all control intensities to be zero, *i.e.*, it represents the uncontrolled dynamics.

For the large Kronecker networks, Figure 7 compares the performance of our algorithm against others in terms of overall average number of tweets $\bar{N}(t) = \sum_{u \in \mathcal{V}} \mathbb{E}[N_u(t)]$ for a fixed budget $\bar{P}(t_f) = \sum_{u \in \mathcal{V}} \mathbb{E}[P_u(t_f)] \approx 6.1\text{K}$. We find that: (i) our algorithm consistently outperforms competing methods by large margins at time t_f; (ii) it triggers up to 50%–100% more posts than the second best performer by time t_f; (iii) MSC tends to use the

Dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{H}(T_{\text{Data}})\rvert$	$T = T_{\text{simulation}}$
Elections	231	1108	1584	120.2
Verdict	1059	10691	17452	22.11
Club	703	4154	9409	19.23
Sports	703	4154	7431	21.53
TV Show	947	10253	13203	12.11

Table 1: Real datasets statistics

budget too early, as a consequence, although it initially beats our method, it eventually gets outperformed by time t_f; and, (iv) the baselines PRK and DEG have an underwhelming performance, suggesting that the network structure alone is not an accurate measure of influence.

Scalability. Figure 8 shows that our algorithm scales to large networks and is almost an order of magnitude faster than the second best performer, MSC (Farajtabar et al., 2016). For example, our algorithm takes ∼30 seconds to steer a network with 1,000 nodes and average degree of 10 while MSC takes ∼4 minutes.

5.2.2 Experiments on Real Data

In this section, we experiment with data gathered from Twitter and show that our model can maximize the number of online actions more effectively than several baselines and state of the art methods (Farajtabar et al., 2014, 2016).

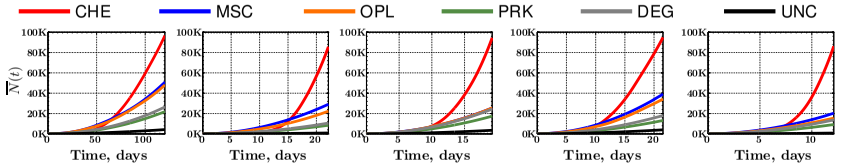
Experimental setup. We experiment with five Twitter data sets about current real-world events, where actions are tweets (and retweets). To create each data set, we used the Twitter search API[11] to collect all the tweets (corresponding to a 2-3 weeks period around the event date) that contain hashtags related to:

- **Elections:** British election, from May 7 to May 15, 2015.
- **Verdict:** Verdict for the corruption-case against Jayalalitha, an Indian politician, from May 6 to May 17, 2015.
- **Club:** Barcelona getting the first place in La-liga, from May 8 to May 16, 2016.
- **Sports:** Champions League final in 2015, between Juventus and Real Madrid, from May 8 to May 16, 2015.
- **TV Show:** The promotion on the TV show "Games of Thrones", from May 4 to May 12, 2015.

We then built the follower-followee network for the users that posted the collected tweets using the Twitter rest API[12]. Finally, we filtered out users that posted less than 200 tweets during the account lifetime, follow less than 100 users, or have less than 50 followers. An account of the data set statistics is given in Table 1.

Similarly as in the synthetic experiments, we compare the performance of our algorithm with two state of the art methods, OPL (Farajtabar et al., 2014) and MSC (Farajtabar

11. https://dev.twitter.com/rest/public/search
12. https://dev.twitter.com/rest/public



(a) Elections, $\bar{P}(t_f) \approx 3K$ (b) Verdict, $\bar{P}(t_f) \approx 14K$ (c) Club, $\bar{P}(t_f) \approx 5K$ (d) Sports, $\bar{P}(t_f) \approx 5K$ (e) Series, $\bar{P}(t_f) \approx 13K$

Figure 9: Performance over time of Cheshire against several competitors for each Twitter data set. Performance is measured in terms of overall number of tweets $\bar{N}(t) = \sum_{u \in \mathcal{V}} \mathbb{E}[N_u(t)]$. In all cases, we tune the parameters $\boldsymbol{Q}$, $\boldsymbol{S}$ and $\boldsymbol{F}$ to be diagonal matrices such that the total number of incentivized tweets *posted* by our method is equal to the budget used in the competing methods and baselines. Cheshire (in red) consistently outperforms competing methods over time and it triggers up to 100%–400% more posts than the second best performer (in blue) as time goes by.

et al., 2016), and three baselines, PRK, DEG and UNC. More in detail, we proceed as follows.

For each data set, we estimate the influence matrix $\boldsymbol{B}$ of the multidimensional Hawkes process defined by Eq. 2 using maximum likelihood, as elsewhere (Farajtabar et al., 2014; Valera and Gomez-Rodriguez, 2015). Moreover, we set the decay parameter ω of the corresponding exponential triggering kernel $\kappa(t)$ by cross-validation. Then, we perform 20 simulation runs for each method and baseline, where we sample non directly incentivized actions from the multidimensional Hawkes process learned from the corresponding Twitter data set using Ogata's thinning algorithm (Ogata, 1981). For the competing methods and baselines, the control intensities $\boldsymbol{u}(t)$ are deterministic and thus we only need to sample incentivized actions from inhomogeneous Poisson processes (Lewis and Shedler, 1979). For our method, the control intensities are stochastic and thus we sample incentivized actions using Algorithm 3. Here, we compare their performance in terms of the (empirical) average number of tweets $\mathbb{E}[\boldsymbol{N}(t)]$. In the above procedure, for a fair comparison, we tune the parameters $\boldsymbol{Q}$, $\boldsymbol{S}$ and $\boldsymbol{F}$ to be diagonal matrices such that the total number of incentivized tweets *posted* by our method is equal to the budget used in the state of the art methods and baselines.

Results. We first compare the performance of our algorithm against others in terms of overall average number of tweets $\bar{N}(t) = \sum_{u \in \mathcal{V}} \mathbb{E}[N_u(t)]$ for a fixed budget $\bar{P}(t_f) = \sum_{u \in \mathcal{V}} P_u(t_f)$. Figure 9 summarizes the results, which show that: (i) our algorithm consistently outperforms competing methods by large margins; (ii) it triggers up to 100%–400% more posts than the second best performer as time goes by; and, (iii) the baselines PRK and DEG have an underwhelming performance, suggesting that the network structure alone is not an accurate measure of influence.

Next, we evaluate the performance of our algorithm against others with respect to the available budget. To this aim, we compute the average time $\bar{t}_{30K}$ required by each method to reach a milestone of 30,000 tweets against the number of directly incentivized tweets

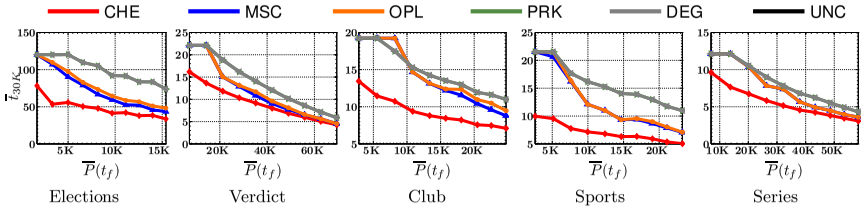


Figure 10: Performance vs. number of directly incentivized tweets for each Twitter data set. Performance is measured in terms of the average time $\bar{t}_{30K}$ required by each method to reach a milestone of 30,000 tweets. Cheshire (in red) consistently reaches the milestone faster than the competing methods, *e.g.*, 20%–50% faster than the second best performer (in blue) for low budgets.

$\bar{P}(t_f)$ (*i.e.*, the budget). Here, we do not report the results for the uncontrolled dynamics (UNC) since it did not reach the milestone after 10× the time the slowest competitor took to reach it. Figure 10 summarizes the results, which show that: (i) our algorithm consistently reaches the milestone faster than the competing methods; (ii) it exhibits a greater competitive advantage when the budget is low; and, (iii) it reaches the milestone 20%–50% faster than the second best performer for low budgets.

6. Conclusions

In this paper, we developed efficient online algorithms for steering social activity, both at a user (local) and network (global) level, based on stochastic optimal control of stochastic differential equations (SDEs) with jumps. In doing so, we established a previously unexplored connection between optimal control of jump SDEs and doubly stochastic marked temporal point processes, which is of independent interest. We experimented with synthetic and real-world data gathered from Twitter and showed that our algorithms can consistently steer social activity more effectively than the state of the art.

Our work also opens many venues for future work. For example, from the perspective of an individual user, we considered social networks that sort stories in the users' feeds in reverse chronological order (*e.g.*, Twitter, Weibo). Extending our methodology to social networks that sort stories algorithmically (*e.g.*, Facebook) is a natural next step. Moreover, we assume that only one broadcaster is using RedQueen. A very interesting follow-up would be augmenting our framework to consider multiple broadcasters under cooperative, competitive and adversarial environments. From the perspective of an entire social networking site, our experimental evaluation is based on simulation, using models whose parameters ($\boldsymbol{B}$, ω) are learned from data. It would be very interesting to evaluate our method using actual interventions in a social network. From both perspectives, our algorithms optimize quadratic losses and assume the model parameters do not change over time, however, it would be useful to derive optimal broadcasting intensities for other losses with well-defined semantics and time-varying model parameters. Finally, optimal control of jump SDEs with doubly stochastic temporal point processes can be potentially applied to design online algorithms for a wide variety of control problems in social and information systems, such as human learning (Reddy et al., 2016) or rumor control (Friggeri et al., 2014).

Appendix A. Proof of Proposition 1

Using the left continuity of Poisson processes and the definition of derivative $d\boldsymbol{\lambda}(t) = \boldsymbol{\lambda}(t + dt) - \boldsymbol{\lambda}(t)$ we can find the dynamics of the process using Ito's calculus (Hanson, 2007) as follows:

$$
\begin{aligned}
d\boldsymbol{\lambda}(t) &= \boldsymbol{\lambda}_0'(t)\,dt + \boldsymbol{C}\int_0^{t+dt} \kappa_\omega(t+dt-s)\,d\boldsymbol{N}(s) - \boldsymbol{C}\int_0^{t} \kappa_\omega(t-s)\,d\boldsymbol{N}(s) \\
&= \boldsymbol{\lambda}_0'(t)\,dt + \boldsymbol{C}\int_0^{t+dt} (\kappa_\omega(t-s) + \kappa_\omega'(t-s)\,dt)\,d\boldsymbol{N}(s) - \boldsymbol{C}\int_0^{t} \kappa_\omega(t-s)\,d\boldsymbol{N}(s) \\
&= \boldsymbol{\lambda}_0'(t)\,dt + \boldsymbol{C}\int_t^{t+dt} \kappa_\omega(t-s)\,d\boldsymbol{N}(s) + dt\,\boldsymbol{C}\int_0^{t+dt} \kappa_\omega'(t-s)\,d\boldsymbol{N}(s) \\
&= \boldsymbol{\lambda}_0'(t)\,dt + \boldsymbol{C}\,\kappa_\omega(0)\,d\boldsymbol{N}(t) - \omega\,dt\,\boldsymbol{C}\int_0^{t+dt} \kappa_\omega(t-s)\,d\boldsymbol{N}(s) \\
&= \boldsymbol{\lambda}_0'(t)\,dt + \boldsymbol{C}\,d\boldsymbol{N}(t) - \omega\,dt\,\boldsymbol{C}\int_0^{t} \kappa_\omega(t-s)\,d\boldsymbol{N}(s) \\
&= \left[\boldsymbol{\lambda}_0'(t) + \omega\boldsymbol{\lambda}_0(t) - \omega\boldsymbol{\lambda}(t)\right]dt + \boldsymbol{C}\,d\boldsymbol{N}(t).
\end{aligned}
$$

Appendix B. Proof of Lemma 3

$$
\begin{aligned}
&J(\gamma(t), r(t), t) = \min_{u(t,t_f]} \mathbb{E}_{(N,M)(t,t_f]}\left[\phi(r(t_f)) + \int_t^{t_f} \ell(r(\tau), u(\tau))\,d\tau\right] \\
&= \min_{u(t,t_f]} \mathbb{E}_{(N,M)(t,t_f]}\left[\phi(r(t_f)) + \int_t^{t+dt} \ell(r(\tau), u(\tau))\,d\tau + \int_{t+dt}^{t_f} \ell(r(\tau), u(\tau))\,d\tau\right] \\
&= \min_{u(t,t_f]} \mathbb{E}_{(N,M)(t,t+dt]}\left[\mathbb{E}_{(N,M)(t+dt,t_f]}\left[\phi(r(t_f)) + \ell(t,r,u)\,dt + \int_{t+dt}^{t_f} \ell(r(\tau), u(\tau))\,d\tau\right]\right] \\
&= \min_{u(t,t+dt]} \min_{u(t+dt,t_f]} \mathbb{E}_{(N,M)(t,t+dt]}\left[\ell(r(t),\gamma(t),t)\,dt + \mathbb{E}_{(N,M)(t+dt,t_f]}\left[\phi(r(t_f)) + \int_{t+dt}^{t_f} \ell(r(\tau), u(\tau))\,d\tau\right]\right] \\
&= \min_{u(t,t+dt]} \mathbb{E}_{(N,M)(t,t+dt]}\left[J(\gamma(t+dt), r(t+dt), t+dt)\right] + \ell(r(t), u(t))\,dt.
\end{aligned}
$$

Appendix C. Proof of Lemma 4

According to the definition of differential,

$$
\begin{aligned}
dJ(r(t),\gamma(t),t) &:= J(r(t+dt),\gamma(t+dt),t+dt) - J(r(t),\gamma(t),t) \\
&= J(r(t)+dr(t),\gamma(t)+d\gamma(t),t+dt) - J(r(t),\gamma(t),t).
\end{aligned}
$$

To evaluate the first term in the right hand side of the above equality we substitute $dr(t)$ and $d\gamma(t)$ using Eq. 9. Then, using the zero-one jump law (Kingman, 1992) we can write:

$$\begin{aligned} dJ = {} & J\big(0, \gamma(t) + m(t)dt, t + dt\big)\, dN(t) + J\big(r(t) + g(t), \gamma(t) + m(t)dt + \alpha, t + dt\big)\, dP(t) \\ & + J\big(r, \gamma(t) + m(t)dt, t + dt\big)\,(1 - dN(t))(1 - dP(t)) - J\big(r(t), \gamma(t), t\big), \end{aligned}$$

where $m(t) = \gamma_0'(t) + \omega\gamma_0(t) - \omega\gamma(t)$. Then, we can expand the first three terms in the right hand sides:

$$\begin{aligned} J\big(0, \gamma + m(t)dt, t + dt\big) &= J(0, \gamma(t), t) + J_\gamma(0, \gamma(t), t)m(t)dt + J_t(0, \gamma(t), t)dt \\ J\big(r(t) + 1, \gamma(t) + m(t)dt + \alpha, t + dt\big) &= J(r(t) + 1, \gamma(t) + \alpha, t) + J_\gamma(r + 1, \gamma(t) + \alpha, t)m(t)dt \\ J\big(r(t), \gamma(t) + m(t)dt, t + dt\big) &= J(r(t), \gamma(t), t) + J_\gamma(r(t), \gamma(t), t)m(t)dt + J_t(r(t), \gamma(t), t)dt, \end{aligned}$$

using that the bilinear differential form $dt\, dN(t) = 0$ (Hanson, 2007) and $dN(t)dP(t) = 0$ by the zero-one jump law. Finally,

$$\begin{aligned} dJ(r(t), \gamma(t), t) = {} & J_t(r(t), \gamma(t), t)dt + \big[\gamma_0'(t) + \omega\gamma_0(t) - \omega\gamma(t)\big]\, J_\gamma(r(t), \gamma(t), t)dt \\ & + [J(0, \gamma(t), t) - J(r(t), \gamma(t), t)]dN(t) \\ & + [J(r(t) + 1, \gamma(t) + \alpha, t) - J(r(t), \gamma(t), t)]dP(t), \end{aligned}$$

which concludes the proof.

Appendix D. Proof of Lemma 5

Consider the following proposal for the cost-to-go:

$$J(r(t), \gamma(t), t) = \sum_{i=0}^{n} \sum_{j=0}^{m} f_{ij}(t) r^i(t) \gamma^j(t),$$

where m and n are arbitrary large numbers and $f_{ij}(t)$ are time-varying functions. Now, substitute this proposal in to Eq. 15:

$$\begin{aligned} 0 = {} & \sum_{i=1}^{n} f_{i0}'\, r^i + f_{i0}\,(r+1)^i\gamma - f_{i0}\, r^i\gamma + \sum_{j=1}^{m} f_{0j}'\gamma^j + j(\gamma_0' + w\gamma_0 - w\gamma)f_{0j}\gamma^{j-1} + f_{0j}(\gamma+\alpha)^j\gamma - f_{0j}\gamma^{j+1} \\ & + \sum_{i=1}^{n}\sum_{j=1}^{m} j(\gamma_0' + w\gamma_0 - w\gamma)f_{ij} r^i\gamma^{j-1} + \sum_{i=1}^{n}\sum_{j=1}^{m} f_{ij}(r+1)^i(\gamma+\alpha)^j\gamma - f_{ij} r^i\gamma^{j+1} \\ & - \frac{1}{2}q^{-1}\Big[\sum_{i=1}^{n} f_{i0} r^i + \sum_{i=1}^{n}\sum_{j=1}^{m} f_{ij} r^i\gamma^j\Big]^2 + \frac{1}{2}s\, r^2 + f_{00}' \end{aligned}$$

where for notational simplicity we omitted the time argument of functions. To find the unknown functions $f_{ij}(t)$, we equate the coefficient of different variables. If we consider the coefficient of r^{2n}, we have $f_{n0}(t) = 0$. We can continue this argument for $n-1, n-2, \cdots, 2$ to show that $\forall i \geq 2$; $f_{i0}(t) = 0$. Similar reasoning for coefficients of $r^{2i}\gamma^{2j}$ shows that $\forall j, i \geq 2$; $f_{ij}(t) = 0$. Finally, the coefficient of r^2 is $s(t)/2 - q^{-1}f_{10}^2(t)/2 = 0$, so $f_{10}(t) = \sqrt{s(t)q}$. If we rename $f_{0j}(t)$ to $g_j(t)$ and $f_{00}(t)$ to $f(t)$, then it follows that

$$J(r(t), \gamma(t), t) = f(t) + \sqrt{s(t)q}\, r(t) + \sum_{j=1}^{m} g_j(t)\gamma^j(t).$$

We can continue the previous method to find the remaining coefficients and completely define the cost-to-go function. If we equate the coefficient of γ^j to zero we would have a system of first oder differential equation which its j'th row for $j > 1$ is

$$g_j'(t) - g_{j-1}(t) + j(\alpha - w)g_j(t) + (j+1)\big(\gamma_0'(t) + w\gamma_0(t) + \frac{j}{2}\alpha^2\big)g_{j+1}(t) + \sum_{k=2}^{m-j}\binom{j+k}{k+1}\alpha^{k+1}g_{j+k}(t) = 0$$

We can also evaluate the corresponding differential equation for $j = 1$. When $\gamma_0(t) = \gamma_0$, we can express these differential equations using a matrix differential equation $\boldsymbol{g}'(t) = A\boldsymbol{g}(t)$. and its solution is $\boldsymbol{g}(t) = c_1 e^{\zeta_1 t}\boldsymbol{u}_1 + c_2 e^{\zeta_2 t}\boldsymbol{u}_2 + \cdots + c_n e^{\zeta_n t}\boldsymbol{u}_n$ where ζ_i and $\boldsymbol{u}_i$ are eigenvalue and eigenvector of matrix A and c_i is a constant found using the terminal conditions. Since in triangular matrices diagonal entries are eigenvalues, we have $\boldsymbol{g}(t) = \sum_{j=1}^{m} c_i e^{j(w-\alpha)}\boldsymbol{u}_i$. We can approximate general time varying $\gamma_0(t)$ using piecewise function and repeat the above procedure for each piece.

Appendix E. Proof of Lemma 9

According to the definition of differential,

$$dJ(\boldsymbol{\lambda}(t), t) := J(\boldsymbol{\lambda}(t + dt), t + dt) - J(\boldsymbol{\lambda}(t), t) = J(\boldsymbol{\lambda}(t) + d\boldsymbol{\lambda}(t), t + dt) - J(\boldsymbol{\lambda}(t), t).$$

To evaluate the first term in the right hand side of the above equality we substitute $d\boldsymbol{\lambda}(t)$ using Eq. 18. Then, using the zero-one jump law (Kingman, 1992) we can write:

$$\begin{aligned} & J(\boldsymbol{\lambda}(t) + d\boldsymbol{\lambda}(t), t + dt) = J(\boldsymbol{\lambda}(t) + m(t)\, dt + \boldsymbol{B}\, d\boldsymbol{N}(t) + \boldsymbol{B}\, d\boldsymbol{P}(t), t + dt) \\ & = \sum_i J(\boldsymbol{\lambda}(t) + m(t)\, dt + \boldsymbol{b}_i, t + dt)dN_i(t) + \sum_i J(\boldsymbol{\lambda}(t) + m(t)\, dt + \boldsymbol{b}_i, t + dt)dP_i(t) \\ & \quad + J(\boldsymbol{\lambda}(t) + m(t)\, dt, t + dt)\prod_i [1 - dN_i(t)][1 - dP_i(t)] \\ & = J(\boldsymbol{\lambda}(t) + m(t)\, dt, t + dt)[1 - \sum_i dN_i(t) + dP_i(t)] + \sum_i J(\boldsymbol{\lambda}(t) + m(t)\, dt + \boldsymbol{b}_i, t + dt)[dN_i(t) + dP_i(t)] \\ & = J(\boldsymbol{\lambda}(t) + m(t)\, dt, t + dt) + \sum_i [J(\boldsymbol{\lambda}(t) + m(t)\, dt + \boldsymbol{b}_i, t + dt) - J(\boldsymbol{\lambda}(t) + m(t)\, dt, t + dt)][dN_i(t) + dP_i(t)] \end{aligned}$$

where $m(t) := \omega\boldsymbol{\lambda}_0 - \omega\boldsymbol{\lambda}(t)$ and we used that the bilinear differential form $dt\, dN(t) = 0$ (Hanson, 2007). By total derivative rule, it follows that

$$\begin{aligned} J(\boldsymbol{\lambda}(t) + m(t)\, dt + \boldsymbol{b}_i, t + dt) &= J(\boldsymbol{\lambda}(t) + \boldsymbol{b}_i, t) + \nabla_{\boldsymbol{\lambda}} J(\boldsymbol{\lambda}(t) + \boldsymbol{b}_i, t)m(t)\, dt + J_t(\boldsymbol{\lambda}(t) + \boldsymbol{b}_i, t)\, dt \\ J(\boldsymbol{\lambda}(t) + m(t)\, dt, t + dt) &= J(\boldsymbol{\lambda}(t), t) + \nabla_{\boldsymbol{\lambda}} J(\boldsymbol{\lambda}(t), t)m(t)\, dt + J_t(\boldsymbol{\lambda}(t), t)\, dt. \end{aligned}$$

Then, the differential is given by:

$$dJ(\boldsymbol{\lambda}(t) + d\boldsymbol{\lambda}(t), t + dt) = J(\boldsymbol{\lambda}(t), t) + (\omega\boldsymbol{\lambda}_0 - \omega\boldsymbol{\lambda}(t))^T \nabla_{\boldsymbol{\lambda}} J(\boldsymbol{\lambda}(t), t)\, dt + J_t(\boldsymbol{\lambda}(t), t)\, dt + \sum_i \left[J(\boldsymbol{\lambda}(t) + \boldsymbol{b}_i, t) - J(\boldsymbol{\lambda}(t), t)\right] \left[dN_i(t) + dP_i(t)\right],$$

which completes the proof.

Appendix F. Proof of $(\Delta_B J)_i \leq 0$

Lets $t < s$, then according to the definition we can write,

$$\boldsymbol{\lambda}(s) = \boldsymbol{\lambda}_0 + \int_0^s \kappa_\omega(s-\tau)\, \boldsymbol{B}\, d\boldsymbol{N}(\tau) = \boldsymbol{\lambda}_0 + \int_0^t \kappa_\omega(s-\tau)\, \boldsymbol{B}\, d\boldsymbol{N}(\tau) + \int_t^s \kappa_\omega(s-\tau)\, \boldsymbol{B}\, d\boldsymbol{N}(\tau).$$

For the exponential kernel $\kappa_\omega(t) = e^{-wt}$ we have,

$$\int_0^t \kappa_\omega(s-\tau)\, \boldsymbol{B}\, d\boldsymbol{N}(\tau) = \int_0^t e^{-w(s-\tau)}\, \boldsymbol{B}\, d\boldsymbol{N}(\tau) = e^{-w(s-t)} \int_0^t e^{-w(t-\tau)}\, \boldsymbol{B}\, d\boldsymbol{N}(\tau) = e^{-w(s-t)}(\boldsymbol{\lambda}(t) - \boldsymbol{\lambda}_0)$$

so given the value of $\boldsymbol{\lambda}(t)$ at time t then we can write $\boldsymbol{\lambda}(s)$ for later times as

$$\boldsymbol{\lambda}(s) = \boldsymbol{\lambda}_0 + e^{-w(s-t)}(\boldsymbol{\lambda}(t) - \boldsymbol{\lambda}_0) + \int_t^s \kappa_\omega(s-\tau)\, \boldsymbol{B}\, d\boldsymbol{N}(\tau)$$

Lets consider a process $\boldsymbol{\xi}(s)$ with intensity value at time t equal to $\boldsymbol{\lambda}(t) + \boldsymbol{b}_i$ as,

$$\boldsymbol{\xi}(s) = \boldsymbol{\lambda}_0 + e^{-w(s-t)}(\boldsymbol{\lambda}(t) + \boldsymbol{b}_i - \boldsymbol{\lambda}_0) + \int_t^s \kappa_\omega(s-\tau)\, \boldsymbol{B}\, d\boldsymbol{N}(\tau).$$

Since $\boldsymbol{b}_i \succeq 0$, then given the same history in interval (t, s), we have $\boldsymbol{\xi}(s) \succeq \boldsymbol{\lambda}(s)$. Then, we have:

$$\ell(\boldsymbol{\xi}(s), \boldsymbol{u}(s)) \leq \ell(\boldsymbol{\lambda}(s), \boldsymbol{u}(s)).$$

Now, taking the integration, then expectation (over all histories) and finally the minimization from the above inequality does not change the direction of inequality. So it readily follows the required result.

$$J(\boldsymbol{\lambda}(t) + \boldsymbol{b}_i, t) \leq J(\boldsymbol{\lambda}(t), t).$$

Appendix G. Proof of Lemma 10

Consider the following proposal of degree three for the cost-to-go function:

$$J(\boldsymbol{\lambda}(t), t) = f(t) + \sum_i g_i(t)\lambda_i(t) + \sum_i \sum_j \lambda_i(t)\lambda_j(t) H_{ij}(t) + \sum_i \sum_j \sum_k \lambda_i(t)\lambda_j(t)\lambda_k(t) H_{ijk}(t)$$

If we plug this proposal in Eq. 22, and evaluate the coefficient of fourth degree terms like $\lambda_i^2(t)\lambda_j^2(t)$ and equate them to zero, then we can find the unknown coefficients $H_{ijk}(t)$'s as follows:

$$\forall i, j, t: \ \sum_k \big(\sum_\ell B_{\ell k}^2\big) H_{ijk}^2(t) = 0$$

Since the sum of positives terms is zero if and only if they all be zero, then $H_{ijk}(t)$ and consequently the terms with degree three in the proposal are all zero. So the proposal reduces to a quadratic proposal.

It is quite straightforward to extend this argument for proposals with order $m > 3$ and by equating the degree $2m-1$ terms, similarly conclude that coefficients of degree m terms in the proposal are zero. If we repeat this argument for $m-1, \ldots, 3$, we deduce that any proposal with arbitrary degree $m \geq 2$, would result in a quadratic optimal cost.

References

O. Aalen, O. Borgan, and H. K. Gjessing. *Survival and event history analysis: a process point of view*. Springer, 2008.

L. Backstrom, E. Bakshy, J. M. Kleinberg, T. M. Lento, and I. Rosenn. Center of attention: How facebook users allocate attention across friends. *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media*, 2011.

D. P. Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena Scientific Belmont, MA, 1995.

R. M. Blumenthal. An extended markov property. *Transactions of the American Mathematical Society*, 85(1):52–72, 1957.

L. Carroll. *Through the looking glass: And what Alice found there*. Rand, McNally, 1917.

M. Cha, H. Haddadi, F. Benevenuto, and P. K. Gummadi. Measuring User Influence in Twitter: The Million Follower Fallacy. *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media*, 2010.

A. De, I. Valera, N. Ganguly, S. Bhattacharya, and M. Gomez-Rodriguez. Learning and forecasting opinion dynamics in social networks. In *Advances in Neural Information Processing Systems*, 2016.

N. Du, L. Song, M. Gomez-Rodriguez, and H. Zha. Scalable influence estimation in continuous-time diffusion networks. In *Advances in Neural Information Processing Systems*, 2013.

M. Farajtabar, N. Du, M. Gomez-Rodriguez, I. Valera, H. Zha, and L. Song. Shaping social activity by incentivizing users. In *Advances in Neural Information Processing Systems*, 2014.

M. Farajtabar, M. Gomez-Rodriguez, N. Du, M. Zamani, H. Zha, and L. Song. Back to the past: Source identification in diffusion networks from partially observed cascades. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, 2015a.

M. Farajtabar, Y. Wang, M. Gomez-Rodriguez, S. Li, H. Zha, and L. Song. COEVOLVE: A joint point process model for information diffusion and network co-evolution. In *Advances in Neural Information Processing Systems*, 2015b.

M. Farajtabar, X. Ye, S. Harati, L. Song, and H. Zha. Multistage campaigning in social networks. In *Advances in Neural Information Processing Systems*, 2016.

A. Friggeri, L. Adamic, D. Eckles, and J. Cheng. Rumor cascades. In *Proceedings of the 8th International AAAI Conference on Weblogs and Social Media*, 2014.

C. K. Garrett. *Numerical integration of matrix Riccati differential equations with solution singularities*. PhD thesis, The University of Texas at Arlington, 2013.

S. Goel, D. J. Watts, and D. G. Goldstein. The structure of online diffusion networks. In *Proceedings of the 13th ACM conference on electronic commerce*, 2012.

M. Gomez-Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the temporal dynamics of diffusion networks. In *Proceedings of the 28th International Conference on Machine Learning*, 2011.

M. Gomez-Rodriguez, K. Gummadi, and B. Schölkopf. Quantifying information overload in social media and its impact on social contagions. In *Proceedings of the 8th International AAAI Conference on Weblogs and Social Media*, 2014.

F. B. Hanson. *Applied stochastic processes and control for Jump-diffusions: modeling, analysis, and computation*, volume 13. SIAM, 2007.

A. G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.

X. He, G. Song, W. Chen, and Q. Jiang. Influence blocking maximization in social networks under the competitive linear threshold model. In *SDM*, pages 463–474. SIAM, 2012.

N. Hodas and K. Lerman. How visibility and divided attention constrain social contagion. *SocialCom*, 2012.

M. Jankowiak and M. Gomez-Rodriguez. Uncovering the spatiotemporal patterns of collective social activity. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 822–830. SIAM, 2017.

J. H. Kang and K. Lerman. Vip: Incorporating human cognitive biases in a probabilistic model of retweeting. In *Social Computing, Behavioral-Cultural Modeling, and Prediction*, pages 101–110. Springer, 2015.

M. Karimi, E. Tavakoli, M. Farajtabar, L. Song, and M. Gomez-Rodriguez. Smart Broadcasting: Do you want to be seen? In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016.

J. F. C. Kingman. *Poisson processes*. Oxford University Press, 1992.

K. Lerman and T. Hogg. Leveraging position bias to improve peer recommendation. *PloS one*, 9(6), 2014.

J. Leskovec, D. Chakrabarti, J. M. Kleinberg, C. Faloutsos, and Z. Ghahramani. Kronecker graphs: An approach to modeling networks. *Journal of Machine Learning Research*, 11 (2):985–1042, 2010.

P. A. Lewis and G. S. Shedler. Simulation of nonhomogeneous poisson processes by thinning. *Naval research logistics quarterly*, 26(3):403–413, 1979.

C. Mavroforakis, I. Valera, and M. Gomez-Rodriguez. Modeling the dynamics of online learning activity. In *Proceedings of the 26th International World Wide Web Conference*, 2017.

S. Mishra, M. Rizoiu, and L. Xie. Feature driven and point process approaches for popularity prediction. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 1069–1078, 2016.

Y. Ogata. On lewis' simulation method for point processes. *Information Theory, IEEE Transactions on*, 27(1):23–31, 1981.

S. Reddy, I. Labutov, S. Banerjee, and T. Joachims. Unbounded human learning: Optimal scheduling for spaced repetition. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016.

M. A. Rizoiu, L. Xie, S. Sanner, M. Cebrian, H. Yu, and P. V. Hentenryck. Expecting to be hip: Hawkes intensity processes for social media popularity. In *Proceedings of the 26th International Conference on World Wide Web*, 2017.

N. Spasojevic, Z. Li, A. Rao, and P. Bhattacharyya. When-to-post on social networks. In *Proceedings of the 21st ACM SIGKDD international conference on Knowledge discovery and data mining*, 2015.

B. Tabibian, I. Valera, M. Farajtabar, L. Song, B. Schoelkopf, and M. Gomez-Rodriguez. Distilling information reliability and source trustworthiness from digital traces. In *Proceedings of the 26th International World Wide Web Conference*, 2017.

I. Valera and M. Gomez-Rodriguez. Modeling diffusion of competing products and conventions in social media. In *Proceedings of the IEEE International Conference on Data Mining*, 2015.

Y. Wang, E. Theodorou, A. Verma, and L. Song. A stochastic differential equation framework for guiding online user activities in closed loop. *arXiv preprint arXiv:1603.09021*, 2016.

Y. Wang, G. Williams, E. Theodorou, and L. Song. Variational policy for guiding point processes. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

H. Xu, D. Luo, and H. Zha. Learning hawkes processes from short doubly-censored event sequences. In *Proceedings of the 34th International Conference in Machine Learning*, 2017.

A. Zarezade, S. Jafarzadeh, and H. R Rabiee. Spatio-temporal modeling of check-ins in location-based social networks. *arXiv preprint arXiv:1611.07710*, 2016a.

A. Zarezade, A. Khodadadi, M. Farajtabar, H. R. Rabiee, and H. Zha. Correlated Cascades: Compete or Cooperate. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, 2016b.

A. Zarezade, U. Upadhyay, H. R. Rabiee, and M. Gomez-Rodriguez. Redqueen: An online algorithm for smart broadcasting in social networks. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*, 2017.

Q. Zhao, M. Erdogdu, H. He, A. Rajaraman, and J. Leskovec. Seismic: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21st ACM SIGKDD international conference on Knowledge discovery and data mining*, 2015.

Journal of Machine Learning Research 18 (2018) 1-61 Submitted 9/16; Revised 10/17; Published 4/18

The Search Problem in Mixture Models

Avik Ray AVIK@UTEXAS.EDU
Department of Electrical and Computer Engineering
University of Texas at Austin
Austin, TX 78701, USA

Joe Neeman NEEMAN@IAM.UNI-BONN.DE
Department of Mathematics
Rheinische Friedrich-Wilhelms-Universität Bonn
D-53115 Bonn, Germany

Sujay Sanghavi SANGHAVI@MAIL.UTEXAS.EDU
Department of Electrical and Computer Engineering
University of Texas at Austin
Austin, TX 78701, USA

Sanjay Shakkottai SHAKKOTT@AUSTIN.UTEXAS.EDU
Department of Electrical and Computer Engineering
University of Texas at Austin
Austin, TX 78701, USA

Editor: Animashree Anandkumar

Abstract

We consider the task of learning the parameters of a *single* component of a mixture model, for the case when we are given *side information* about that component; we call this the "search problem" in mixture models. We would like to solve this with computational and sample complexity lower than solving the overall original problem, where one learns parameters of all components.

Our main contributions are the development of a simple but general model for the notion of side information, and a corresponding simple matrix-based algorithm for solving the search problem in this general setting. We then specialize this model and algorithm to four common scenarios: Gaussian mixture models, LDA topic models, subspace clustering, and mixed linear regression. For each one of these we show that if (and only if) the side information is informative, we obtain parameter estimates with greater accuracy, and also improved computation complexity than existing moment based mixture model algorithms (e.g. tensor methods). We also illustrate several natural ways one can obtain such side information, for specific problem instances. Our experiments on real data sets (NY Times, Yelp, BSDS500) further demonstrate the practicality of our algorithms showing significant improvement in runtime and accuracy.

Keywords: mixture models, search, side information, semi-supervised, method of moments

1. Introduction

Mixture models denote the statistical setting where observed samples can come from one of several distinct underlying populations—each typically with its own probability distribution—but are not labeled as separate in the data presented. They have been used to model a wide variety of phenomena, and have seen great success in practice, going back as far as Pearson (1894). In this paper we consider (what we call) the **search problem** in the mixture model setting: given some *special side information* about one of the mixture components, is it possible to efficiently learn the parameters of that component only? Given that there are known methods for learning the entire set of parameters of various mixture models, "efficient" here means more efficient (statistically and/or computationally) than existing methods for learning all the parameters.

As an example, we consider the "latent Dirichlet allocation" model for document generation. In this model, "underlying population" means the set of topics in a document, which determines the frequencies of different words in the document. "Side information" could be a word that is more common in the topic of interest than it is in any other topic: for example, the word "semi-supervised" might work if the topic of interest is machine learning.

Side information could also consist of a small number of labelled examples. We might have a small collection of documents about machine learning and also a much larger corpus that includes documents from many topics. Our methods will allow us to leverage the large, unlabelled corpus to obtain good estimates for word frequencies in machine learning articles—and these estimates will be much better than anything that could be learned from the small labelled sample.

Main contributions: We propose a general setting for side information in mixture models, and show how to solve the search problem by estimating certain matrices of moments. We prove error bounds on the resulting estimates; our rates have a sharp dependence on the sample size (although they are possibly not sharp in the other parameters).

We then specialize our approach to four popular families of mixture models: Gaussian mixture models with spherical covariances, latent Dirichlet allocation for topic models, mixed linear regression, and subspace clustering. We give concrete algorithms for these four families. Our results also include new moment derivations for mixed linear regression and subspace clustering models.

Finally, we simulate our algorithm on both real and synthetic data sets for the Gaussian mixture model, topic model, and subspace clustering applications. For synthetic data set we compare its performance to the tensor decomposition methods discussed by Anandkumar et al. (2014) in both GMM and LDA models, and k-means for subspace clustering. We show that our methods outperform the baseline when the side information is informative. We also demonstrate the practical applicability of our algorithms on three real data sets—the NY Times data set of news articles, Yelp data set of business reviews, and BSDS500 data set of images. In the first two text corpus, we show our algorithm recovers more coherent topics than topic modeling algorithm by Arora et al. (2013). In the BSDS500 data set, we demonstrate how our algorithm can be used for parallel image segmentation. In all three cases, our algorithm also exhibits significant computational gains over competing unsupervised and semi-supervised algorithms.

1.1 Related Work

There is a vast literature on mixture models; too much to even summarize here. We will therefore focus this section on two more closely related areas: method of moments estimators for mixture models, and learning with side information.

Mixture models and method of moments: A common method for learning mixture models is the EM algorithm of Dempster et al. (1977), which outputs a complete set of model parameters. However, EM may converge slowly (or not at all) [Redner and Walker 1984]; this weakness of EM has spurred a resurgence in method-of-moments estimators for mixture models. Although these methods go back to the pioneering work of Pearson (1894) on Gaussian mixture models, the last several years have seen important advances. Moitra and Valiant (2010), and Hardt and Price (2015) showed that Gaussian mixture models with two components can be learned in polynomial time. Hsu and Kakade (2013) considered mixtures of more Gaussians, but constrained to have spherical covariances. They gave a method based on third-order tensor decompositions, which was later generalized to other models in Anandkumar et al. (2014).

Learning with side information: As has been observed many times, often in practice one has access to a set of data that is somewhat richer than standard models of data in learning theory. The term *side information* is used as a catch-all for extra data that doesn't fit into pre-existing models; as such, the literature contains many incomparable models of side information.

Xing et al. (2002) and Yang et al. (2010) took unsupervised clustering as their starting point. For them, side information arrived as pairs of points that were known to belong to the same cluster; they showed how this extra information could substantially improve the performance of the k-means algorithm.

Kuusela and Ocone (2004) developed a framework for side information in the PAC learning model, in which extra samples with a particular dependence on the original samples could sometimes give a substantial benefit.

Many different types of metadata have been proposed for the *latent Dirichlet allocation* (LDA) model of document generation. Mcauliffe and Blei (2008) introduced the *supervised LDA* model, in which each document comes with an additional response variable from a generalized linear model. On the other hand Rosen-Zvi et al. (2004) proposed the *author-topic model*, in which the metadata (author names) affects the distribution of the documents themselves. From a more experimental point of view, Lu and Zhai (2008) used long, detailed product reviews as side information for categorizing short snippets and blog entries.

The notion of *semi-supervised learning* (see the book by Chapelle et al. (2006)) is also related to our framework of side information. In semi-supervised learning, the learner has access to a small number of labelled examples and a large number of unlabelled examples. This setting is useful for us too, although our general method does not strictly require data of this form.

2. Basic Idea and Algorithm

We now first briefly describe the basic mixture model setting, and then describe our method. These descriptions cover several popular specific examples for mixture models, and we detail the application to each of them in Section 3.

Setting: We are interested in the standard statistical setting of (parametric) mixture models: that is, samples are drawn i.i.d. from a distribution f given by

$$f(x) \;=\; \sum_{i=1}^{k} \alpha_i \, g(x;\mu_i).$$

Here g corresponds to a known parametric class of distributions, and k is the number of mixture components. The corresponding parameter vectors are $\mu_1, \ldots, \mu_k$, and their mixture weights / probabilities are $\alpha_1, \ldots, \alpha_k$. So, for example, in the case of the standard (spherical) Gaussian mixture model, $g(x;\mu_i)$ is the Gaussian pdf $\mathcal{N}(\mu_i, I)$. Thus each sample can be considered to be drawn by first selecting a mixture component μ_i with probability α_i, and then drawing the sample x according to $g(x;\mu_i)$. We assume all the μ_i's are *linearly independent*. This is a common assumption for learning mixture models using spectral methods.

Search problem: The standard parameter estimation problem is to find all the μ_i vectors given samples. In this paper we are interested in the search problem: we are given *side information* about one of the vectors—say μ_1, without loss of generality—and we would like to recover *only* μ_1. Of course, we would like to do this with sample and computational complexity lower than what would be required to estimate all parameter vectors (i.e., lower complexity than the standard case).

Side information: Our general procedure requires the following model for side information: we assume that we have access to a vector v such that the inner product with the parameter vector μ_1—the special one we are searching for—is higher than the inner product with any of the other μ_i; i.e. there exists $\delta > 0$ such that;

$$\langle \mu_1, v\rangle \geq \; (1+\delta)\langle \mu_i, v\rangle \quad \text{for all } i \neq 1$$

Section 3 shows how to obtain such side information in some specific models of interest: spherical Gaussian mixture models, mixed linear regression, subspace clustering and the LDA topic model.

We remark that it's also possible (and perhaps more intuitive in some situations) to ask for side information satisfying $|\langle \mu_1, v\rangle| \geq (1+\delta)|\langle \mu_i, v\rangle|$. However, our assumption above is slightly weaker, since for any v satisfying the latter assumption, either v or $-v$ satisfies the former assumption. Later, we show the above condition is sufficient for uniquely identifying the required parameter μ_1 (but it may not be necessary). We refer side information vector v as *informative* about μ_1 if it satisfies the above condition.

2.1 General Procedure

The main idea behind method of moments is to use samples to estimate certain moments of the distribution $f(x)$, using which we can recover the parameters of interest. For many mixture models (including the four common examples we detail), it is possible to easily and directly estimate using first and second order moments, given sufficient samples, the vector

$$m := \sum_{i=1}^{k} \alpha_i \mu_i. \tag{1}$$

and the matrix

$$A := \sum_{i=1}^{k} \alpha_i \mu_i \mu_i^T. \tag{2}$$

For example, in many models the estimate of vector m is simply the sample mean, and matrix A can be derived from the sample covariance matrix. The exact procedure for estimating m and A varies according to the particular parametric model g. The fact that m and A (and also higher-order tensors) can be estimated from samples is well known for many models, see Anandkumar et al. (2014) for a treatment of several different models, and for other pointers to the literature.

Typically, all mixture model components cannot be identified from just the first and second order moments (or m and A). It is often necessary to compute even higher order moment terms. In our search problem, given the side information, **we develop** procedures to estimate an alternative matrix B, using higher order moments, given by

$$B := \sum_{i=1}^{k} \alpha_i \langle \mu_i, v \rangle \mu_i \mu_i^T \tag{3}$$

Again, the exact procedure for estimating B from samples depends on the particular parametric model g.

For this section, we assume we are able to estimate A, B, m to within some accuracy. We will use the notation $\hat{A}, \hat{B}, \hat{m}$ to denote these finite sample estimates of A, B, m respectively, and n denotes the number of samples used to compute these estimates. With this in hand, we outline two general procedures for estimating μ_1 (i.e. the component that we are interested in). The first procedure is based on a whitening step, much like the one that is used in the spectral algorithms in Hsu and Kakade (2013); Anandkumar et al. (2012), and tensor decomposition methods of Anandkumar et al. (2014) (please see remarks in Section 3 for the differences for specific models). The second procedure uses a line search instead, and may be computationally favorable when k is large, because it avoids the need to invert a $k \times k$ matrix. Both Algorithms 1 and 2 take as input the estimates $\hat{A}, \hat{B}, \hat{m}$ (where $\hat{B}$ is constructed using side information vector v) and they output estimates of the first mixture component $\hat{\mu}_1$, and also the proportion of the first component $\hat{\alpha}_1$.

2.1.1 The Whitening Method

Our main result about Algorithm 1 is that if $\hat{A}$ and $\hat{B}$ are good estimates of A and B then Algorithm 1 outputs good estimates for μ_1 and α_1. In order to interpret Theorem 1 as an error rate, note that if all parameters but ϵ are fixed then the error is $O(\epsilon)$. Since standard concentration results yield $\epsilon = O(n^{-1/2})$, where n is the number of samples; our error rate in terms of n is also $O(n^{-1/2})$. This rate is sharp, since it is also the rate for estimating the mean of a single Gaussian vector (i.e. a GMM with only one component).

Theorem 1 *Suppose that $\mu_1, \ldots, \mu_k$ are linearly independent, and that $\hat{A}$ is positive semi-definite. Also suppose that $\langle \mu_1, v \rangle \geq (1+\delta)\langle \mu_i, v \rangle$ for all $i \neq 1$. Assume that*

$$\max\{\|A - \hat{A}\|, \|B - \hat{B}\|, \|m - \hat{m}\|\} \leq \epsilon < \sigma_k(A)/4,$$

Algorithm 1 Extracting a mixture component from side information: the whitening method.

Input: $\hat{A}, \hat{B}, \hat{m}$

Output: $\hat{\mu}_1, \hat{\alpha}_1$

1: let $\{\sigma_j, v_j\}$ be the singular values and singular vectors of $\hat{A}$, in non-increasing order
2: let V be the $d \times k$ matrix whose jth column is v_j
3: let D be the $k \times k$ diagonal matrix with $D_{jj} = \sigma_j$
4: let u be the largest eigenvector of $D^{-1/2} V^T \hat{B} V D^{-1/2}$
5: let $w = V D^{1/2} u$
6: let E be the span of $\{V D^{1/2} v : v \perp u\}$
7: write $V V^T \hat{m}$ (uniquely) as $aw + y$, where $y \in E$
8: return w/a and a^2

and that the right hand side of (4) *is at most α_1. Then*

$$\|\mu_1 - \hat{\mu}_1\| \leq CR|\alpha_1^{-1/2} - \hat{\alpha}_1^{-1/2}| + C\frac{\sqrt{\sigma_1(A)}}{\sqrt{\alpha_1}}\eta \quad , \text{ and}$$

$$|\alpha_1 - \hat{\alpha}_1| \leq \frac{C\sqrt{\alpha_1}(\alpha_1 R + \eta)}{\sigma_k(A)} \left(\eta + R\frac{\epsilon}{\sigma_k(A)} + \epsilon \right) \tag{4}$$

where $\eta = \frac{\epsilon \sigma_1}{\delta \sigma_k^{5/2}}$, $R = \max_i \|\mu_i\|$, $\sigma_1(A) \geq \cdots \geq \sigma_k(A) > 0$ are the non-zero singular values of $A = \sum_i \alpha_i \mu_i \mu_i^T$, and C is a universal constant.

Our error bounds are somewhat complicated, and depend on many different parameters, so let us elaborate on them slightly. First of all, the dependence on $\sigma_1(A)$ and $\sigma_k(A)$ is of the order $\|\mu_1 - \hat{\mu}_1\| \lesssim \sigma_1(A)^{3/2}/\sigma_k(A)^{5/2}$, which is probably an artifact of the analysis, and not the true behavior of the algorithm. On the other hand, our dependence on ϵ is optimal: we have $|\alpha_1 - \hat{\alpha}_1| \lesssim \epsilon$ and $\|\mu_1 - \hat{\mu}_1\| \lesssim \epsilon$. Note also that our bound has no explicit dependence on k; this feature comes from the fact that our method is targeted at a single mixture component. By comparison, other methods typically give bounds in which the *averaged* per-mixture-component error does not depend on k. In terms of dependence on k, therefore, our bounds are better than previous bounds if there is only one component of interest.

Finally, let us remark on the assumption that the right hand side of (4) is at most α_1. This amounts to an assumption that ϵ is sufficiently small compared to all the other parameters. Without this assumption, the bound in (4) would not be very interesting, since $|\alpha_1 - \hat{\alpha}_1| \leq \alpha_1$ is too weak to give useful information about $\hat{\alpha}_1$ (it could even be zero).

We defer the actual analysis of Algorithm 1 to the appendix, but we will motivate the algorithm and give the basic idea of the proof by showing that if $\hat{A}, \hat{B}$, and $\hat{m}$ are equal to A, B and m respectively then Algorithm 1 outputs μ_1 and α_1 exactly.

Lemma 2 *Let m, A, and B be defined by in* (1), (2), *and* (3), *where $\mu_1, \ldots, \mu_k$ are linearly independent. If $\langle \mu_1, v \rangle > \langle \mu_i, v \rangle$ for all $i \neq 1$ and we apply Algorithm 1 to A, B, and m, then it returns μ_1 and α_1.*

Proof Let V and D be as defined in Algorithm 1. Since A has rank k,

$$\sum_{i=1}^{k} \alpha_i D^{-1/2} V^T \mu_i \mu_i^T V D^{-1/2} = D^{-1/2} V^T A V D^{-1/2} = I_k.$$

Defining $u_i := \sqrt{\alpha_i} D^{-1/2} V^T \mu_i$, we have $\sum_i u_i u_i^T = I_k$, which implies that the u_i are orthonormal in $\mathbb{R}^k$. Now,

$$D^{-1/2} V^T B V D^{-1/2} = \sum_{i=1}^{k} \alpha_i \langle \mu_i, v \rangle D^{-1/2} V^T \mu_i \mu_i^T V D^{-1/2} = \sum_{i=1}^{k} \langle \mu_i, v \rangle u_i u_i^T.$$

Since $\langle \mu_1, v \rangle$ was assumed to be larger than all other $\langle \mu_i, v \rangle$, it follows that u_1 is the largest eigenvector of $D^{-1/2} V^T B V D^{-1/2}$. Now, if $w = V D^{1/2} u_1$ then $w = \sqrt{\alpha_1}\mu_1$.

Now, note that since the μ_i are linearly independent, there is a unique way to write $m = V V^T m = \sum_i \alpha_i \mu_1$ as $aw + y$, where y belongs to the span of $\{\mu_2, \ldots, \mu_k\}$ (which is the same as the span of $\{V D^{1/2} u_i : i \geq 2\}$. Moreover, the unique choice of a that allows this representation must satisfy $aw = \alpha_1 \mu_1$, which implies that $a = \sqrt{\alpha_1}$. Therefore, $w/a = \mu_1$ and $a^2 = \alpha_1$. ■

The proof of Lemma 2 is crucial to understanding the algorithm, and also the broader message of this article: if we can get hold of two different normalizations of something, then we can learn something about it. In the proof of Lemma 2, this happens twice: first, we use the fact that A and B contain the same components (but with differing normalizations) to extract the span of a single component of interest. The differing normalization is crucial, because A by itself does not uniquely determine the set $\{\mu_1, \ldots, \mu_k\}$, much less single out a specific component of interest.

In the second step of Lemma 2, we know $\sqrt{\alpha_1}\mu_1$, which is not enough to determine either α_1 or μ_1. However, we also have access to m, which involves a contribution of $\alpha_1 \mu_1$. Exploiting the difference between these two normalizations, we recover both α_1 and μ_1.

2.1.2 The Cancellation Method

Our second method avoids the matrix inversion in Algorithm 1, preferring a line search instead.

In the above Algorithm 2, we assume $\langle \mu_1, v \rangle > 0$. When this is not the case and B is a negative semi-definite matrix, we simply have to change the line search step to search for the smallest $\lambda < 0$ such that $\widehat{V}\widehat{V}^T(\widehat{A} - \lambda \widehat{B})\widehat{V}\widehat{V}^T$ is PSD. Theorem 3 shows that with m, A, B estimated up to $O(\epsilon)$ error, the parameter estimation error in Algorithm 2 is also bounded as $O(\epsilon)$.

Theorem 3 *Suppose $\{\mu_1, \ldots, \mu_k\}$ are linearly independent and v satisfies $\langle \mu_1, v \rangle \geq (1 + \delta)\langle \mu_i, v \rangle$ for all $i \neq 1$. Suppose that $\max\{\|\widehat{A} - A\|, \|\widehat{B} - B\|, \|\hat{m} - m\|\} < \epsilon$, and $\lambda_1 := 1/\langle \mu_1, v \rangle$. Then Algorithm 2 returns $\hat{\mu}_1, \hat{\alpha}_1$ with*

$$\begin{aligned} \|\hat{\mu}_1 - \mu_1\| &< \frac{C\epsilon}{\alpha_1^2 a_1^2}\left(\sigma_1(A)\left(1 + \frac{\alpha_1 a_1}{\sigma_{k-1}(Z_{\lambda_1})}\right) + \frac{\sigma_1(A)\eta_3 R}{\sigma_{k-1}(Z_{\lambda_1})}\right) \\ |\hat{\alpha}_1 - \alpha_1| &< \frac{C\sigma_1(A)\epsilon}{\alpha_1 a_1^3}\left(\eta_1 + \frac{\eta_2 R \eta_3}{\sigma_{k-1}(Z_{\lambda_1})}\right) \end{aligned}$$

Algorithm 2 Extracting a mixture component from side information: the cancellation method.

Input: $\hat{A}, \hat{B}, \hat{m}$
Output: $\hat{\mu}_1, \hat{\alpha}_1$

1: let $\widehat{V}$ be the $d \times k$ matrix of k largest eigenvectors of $\hat{A}$;
2: search over λ to find the largest $\lambda = \lambda^*$ such that $\widehat{V}\widehat{V}^T(\widehat{A} - \lambda\widehat{B})\widehat{V}\widehat{V}^T$ is PSD;
3: let $\widehat{Z}_{\lambda^*} = \widehat{A} - \lambda^* \widehat{B}$, and let $\{v_2, \ldots, v_k\}$ be the top $k-1$ singular vectors of $\widehat{Z}_{\lambda^*}$
4: let $V_{1:(k-1)}$ be the $d \times (k-1)$ matrix with columns $\{v_2, \ldots, v_k\}$
5: let $x_1 = \hat{m} - V_{1:(k-1)} V_{1:(k-1)}^T \hat{m}$
6: let $v_1 = x_1 / \|x_1\|$
7: compute $c_i = v_1^T \widehat{A} v_i$ for $i = 1$ to k
8: let $a_i = c_i / \|x_1\|$ for $i = 1$ to k
9: return $\hat{\mu}_1 = \sum_{i=1}^{k} a_i v_i$ and $\hat{\alpha}_1 = c_1 / a_1^2$

where $\eta_1 := \max\{\alpha_1 a_1 (2a_1 + 1), 20\}$, $\eta_2 := \max\{\alpha_1 a_1^2, 10\}$, $\eta_3 = \max\{1, \lambda_1, \sigma_1(B)\}$, $R = \max \|\mu_i\|$, $a_1 = \|\mu_1 - \prod_{\mathcal{V}} \mu_1\|$, where $\mathcal{V} = span\{\mu_2, \ldots, \mu_k\}$, and C is an universal constant.

Again, we will defer the actual analysis to the appendix, and instead show that Algorithm 2 returns the exact answer when fed exact initial data. We will do this in two lemmas: Lemmas 4 and 5.

Lemma 4 *Let $Z = \sum_{i=1}^{k} \gamma_i \mu_i \mu_i^T$ where $\{\mu_1, \ldots, \mu_k\}$ are linearly independent, $\mu_i \in \mathbb{R}^d, \gamma_i \in \mathbb{R}$ and $d > k$. If $\gamma_1 < 0$ and $\gamma_i > 0$ for all $i \neq 1$ then Z is not positive semi-definite.*

Proof Let Π denote the projection onto the orthogonal complement of $\text{span}\{\mu_2, \ldots, \mu_k\}$. Let $x = \Pi \mu_1$, and note that $\langle x, \mu_1 \rangle > 0$ but $\langle x, \mu_i \rangle = 0$ for all $i \neq 1$. Hence, $x^T Z x = \gamma_1 \langle x, \mu_1 \rangle^2 < 0$ and so Z is not positive semi-definite. ■

Lemma 5 *Let m, A, and B be defined by in (1), (2), and (3), where $\mu_1, \ldots, \mu_k$ are linearly independent. If $\langle \mu_1, v \rangle > \langle \mu_i, v \rangle$ for all $i \neq 1$ and we apply Algorithm 2 to A, B, and m, then it returns μ_1 and α_1.*

Proof Define $w_i = \langle \mu_i, v \rangle$ and let $\gamma_i = \alpha_i(1 - \lambda w_i)$, so that

$$Z_\lambda = A - \lambda B = \sum_{i=1}^{k} \gamma_i \mu_i \mu_i^T.$$

Note that, in our case where $\widehat{A} = A$, and $\widehat{B} = B$, columns of $\widehat{V}$ simply form a common orthonormal bases of the row/column space of both matrices A, B. Therefore the matrix $\widehat{V}\widehat{V}^T(A - \lambda B)\widehat{V}\widehat{V}^T = A - \lambda B = Z_\lambda$. Now for $\lambda > \frac{1}{w_1}$, $\gamma_1 < 0$ and for all $\lambda \leq \frac{1}{w_1}$, $\gamma_i \geq 0$ for all i since $w_1 > w_i$, for every $i \neq 1$. By Lemma 4, $\lambda^* = \frac{1}{w_1}$ is the largest λ such that Z_λ is PSD; hence,

$$Z_{\lambda^*} = \sum_{i=2}^{k} \alpha_i (1 - \lambda^* w_i) \mu_i \mu_i^T.$$

From Lemma 26 in Appendix E.2 it follows that $k-1$ singular vectors $\{v_2, \ldots, v_k\}$ of Z_{λ^*} form a basis of the subspace $\mathcal{V} = \text{span}\{\mu_2, \ldots, \mu_k\}$. Let $\mathcal{V}_\perp$ be the perpendicular space of $\mathcal{V}$, and write $\Pi = I - V_{1:(k-1)} V_{1:(k-1)}^T$ for the orthogonal projection onto $\mathcal{V}_\perp$. Since $\Pi\mu_i = 0$ for $i \neq 1$, we have $x_1 = \Pi m = \alpha \Pi \mu_1$.

Now define $b_1, \ldots, b_k$ by $\mu_1 = \sum_{i=1}^k b_i v_i$. In order to prove that the algorithm returns μ_1 correctly, we need to show that $b_i = a_i := c_i/\|x_1\|$. Indeed,

$$c_i := v_1^T A v_i = \sum_{j=1}^k \alpha_j v_1^T \mu_j \mu_j^T v_i = \alpha_1 b_1 b_i,$$

since $v_1^T \mu_j = 0$ for $j \neq 1$. On the other hand, $\|x_1\| = \alpha \|\Pi \mu_1\| = \alpha b_1$, and so $b_i = a_i$, as claimed. Moreover, $\hat{\alpha}_1 = \frac{c_1}{a_1^2} = \alpha_1$, as claimed. ■

Optimization for λ^*: The first step of Algorithm 2 involves finding a smallest λ^* such that $\widehat{Z}'_{\lambda^*} = \widehat{V}\widehat{V}^T(\widehat{A} - \lambda^* \widehat{B})\widehat{V}\widehat{V}^T$ is PSD using line search. Although $\widehat{Z}'_\lambda$ is a $d \times d$ matrix, this step can be performed efficiently as follows. Instead of searching for λ directly for $\widehat{Z}'_\lambda$, we do this for a smaller $k \times k$ matrix $\widehat{V}^T \widehat{Z}'_\lambda \widehat{V} = \widehat{V}^T(\widehat{A} - \lambda^* \widehat{B})\widehat{V}$. This optimization step using line search can be performed in just $O(k^3 \log |\lambda^*|)$ time.

3. Specific Models

In this section we discuss how the search algorithms can be applied in four specific mixture models.

3.1 Gaussian Mixture Model with Spherical Covariance

The model: Besides the mixture parameters $\alpha_1, \ldots, \alpha_k$, the Gaussian mixture model (GMM) has mean parameters $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$ and variance parameters $\sigma_1, \ldots, \sigma_k \in \mathbb{R}$. The conditional densities $g(\cdot; \mu_i, \sigma_i)$ are Gaussian, with mean μ_i and covariance $\sigma_i^2 I_d$. Explicitly,

$$g(x; \mu_i, \sigma_i) = \frac{1}{(2\pi\sigma_i^2)^{d/2}} e^{-\frac{\|x-\mu_i\|^2}{2\sigma_i^2}}.$$

Matrices A and B: We fix a vector $v \in \mathbb{R}^d$, with the assumption that $\langle v, \mu_1 \rangle > \langle v, \mu_i \rangle$ for $i \neq 1$. Recall (from Section 2.1) that $m = \mathbb{E}[x] = \sum_i \alpha_i \mu_i$, $A = \sum_{i=1}^k \alpha_i \mu_i \mu_i^T$, and $B = \sum_{i=1}^k \alpha_i \langle \mu_i, v \rangle \mu_i \mu_i^T$. To compute these quantities, we first define σ^2 to be the $(k+1)$th-largest eigenvalue of the mixture covariance matrix $\mathbb{E}[(x-m)(x-m)^T]$, and let u be a corresponding eigenvector. Then let $\widetilde{m} = \mathbb{E}[x(u^T(x-m))^2]$. Then it follows from moment computations (see Hsu and Kakade (2013)) that:

$$\begin{aligned} A &= \mathbb{E}[xx^T] - \sigma^2 I_d \\ B &= \mathbb{E}[\langle x, v \rangle xx^T] - \widetilde{m}v^T - v\widetilde{m}^T - \langle \widetilde{m}, v \rangle I_d, \end{aligned}$$

Given the samples $\{\hat{x}_i\}$, we can now empirically evaluate these quantities (denoted by $\hat{m}, \hat{A}, \hat{B}$ respectively) by replacing expectations above by the corresponding sample averages; for instance we replace $\mathbb{E}[xx^T]$ by $\widehat{\mathbb{E}}[xx^T] \doteq (1/n) \sum_{j=1}^n \hat{x}_j \hat{x}_j^T$.

Examples of v: Assuming that $\|\mu_1\|^2 > \langle \mu_1, \mu_i \rangle$ for all $i \neq 1$—this will be true, for example, if $\|\mu_i\|$ are all the same—one can find a suitable vector v given a relatively small number of samples from the first mixture component. Specifically, if $\|\mu_1\|^2 \geq \langle \mu_1, \mu_i \rangle + \delta$ and $\|\mu_i\| \leq R$ for all $i \neq 1$ then standard Gaussian tail bounds imply the following: if $v := \ell^{-1} \sum_{j=1}^\ell x_j$ where $\ell = \Omega(R^2 \delta^{-2} \log k)$ and $x_1, \ldots, x_m$ are drawn independently from the distribution $g(\cdot; \mu_1, \sigma_1)$ then with high probability v satisfies $\langle v, \mu_1 \rangle > \langle v, \mu_i \rangle$ for all $i \neq 1$. Here, "high probability" means probability converging to 1 as the hidden constant in $\ell = \Omega(\cdot)$ grows. Note here that the number of tagged samples is nowhere near sufficient to estimate μ_1 by direct averaging; indeed to do so would require the number of samples to grow with the size of the underlying dimension.

Remarks: We note that spectral algorithms which uses the whitening procedure has been proposed before in the context of GMM e.g. Hsu and Kakade (2013). The primary difference between the algorithm in Hsu and Kakade (2013) and Algorithm 1 is that the former, in absence of side information, takes a projection of the third order moment tensor M_3 on a random unit vector to obtain the second matrix, where as our matrix B can be viewed as a projection of M_3 on the side information vector v. The main advantage of projecting onto v is that, when we have reliable side information, this will give a good singular value separation resulting in better empirical performance. The Cancellation algorithm however is distinctly different from both and has not been studied before.

3.2 Latent Dirichlet Allocation

The model: In the LDA model with k topics and a dictionary of size d, the parameters $\mu_1, \ldots, \mu_k \in \Delta_{d-1}$ are the probability distributions corresponding to each topic (Δ_{d-1} denotes the probability simplex $\{y \in \mathbb{R}^d : \sum_i y_i = 1, \min_i y_i \geq 0\}$). The LDA model introduced in Blei et al. (2003) differs slightly from the other models as the mixture distribution cannot be expressed exactly in the parametric form in Section 2. Instead we have a two level hierarchy as follows. Given $\bar{\alpha} = (\alpha_1, \ldots, \alpha_k)$, we first draw a topic distribution θ from the Dirichlet$(\bar{\alpha})$ distribution. Given this $\theta = (\theta_1, \ldots, \theta_k)$ each word in the document is drawn i.i.d. from the distribution $\sum_{i=1}^k \theta_i \mu_i$. However still we can compute the vector m and the matrices A, B as shown below. Then with an appropriate v our algorithms can recover the topic distribution μ_1.

Matrices A and B: Let x_1 denote the random vector with $x_1(w) = 1$ if the first word is w, and 0 otherwise. Similarly define vectors x_2, x_3 corresponding to the second and third word respectively, and let $\alpha_0 = \sum_{i=1}^k \alpha_i$. Then, moment computations under the LDA distribution yields the following expressions for (m, A, B), defined in (1), (2), (3):

$$\begin{aligned} &m = \alpha_0 \mathbb{E}[x_1], \quad A = \alpha_0(\alpha_0+1)\mathbb{E}[x_1 x_2^T] - mm^T \\ &B = \frac{\alpha_0(\alpha_0+1)(\alpha_0+2)}{2} \mathbb{E}[\langle x_3, v \rangle x_1 x_2^T] - \frac{\alpha_0(\alpha_0+1)}{2} \left(\langle m, v \rangle \mathbb{E}[x_1 x_2^T] + \mathbb{E}[\langle x_3, v \rangle x_1 m^T] \right. \\ &\quad \left. + \mathbb{E}[\langle x_3, v \rangle m x_2^T] \right) + \langle m, v \rangle mm^T. \end{aligned}$$

With the given document samples, let $\hat{x}_i$ denote the normalized empirical word frequencies in the document i. Then, $\hat{m} = \frac{\alpha_0}{n} \sum_{i=1}^n \hat{x}_i$, and $\widehat{A}, \widehat{B}$ can be immediately estimated using the above expressions by replacing expectations with sample averages.

Using labeled words to find v: In order to recover the topic distribution μ_1 we now require a vector v which satisfies $\langle \mu_1, v\rangle > \langle \mu_i, v\rangle$ for $i \neq 1$. Now suppose we are given a *labeled word* ℓ such that its occurrence probability in topic 1 is the highest, i.e., $\mu_1(\ell) > \mu_i(\ell)$ for $i \neq 1$ (note that this does not mean ℓ is the most frequent word in topic 1, there may be words with higher occurrence probability in this topic). Then we can simply choose $v = e_\ell$ (the standard basis element with 1 in the ℓ-th coordinate). For most topics of practical interest it is possible to find such labeled words. For example the word "ball" can be a labeled word for topic sport, "party" is a labeled word for topic politics and so on. However, a labeled word is merely indicative of a topic and is not exclusive to a topic (e.g. the word "ball" can occur in other contexts as well). In this sense, the labelled word is quite different from the "anchor word" described in Arora et al. (2013). Note however that anchor words are also labeled words (but *not* vice-versa) since for an anchor word ℓ, $\mu_1(\ell) > 0$ and $\mu_i(\ell) = 0$ for $i \neq 1$.

Using labeled documents to find v: If the different topics are not too similar, then we can estimate a suitable vector v from a small collection of documents that are mostly about the topic of interest. For example, if $\langle \mu_i, \mu_j\rangle \leq \eta\|\mu_i\|\|\mu_j\|$ for all $i \neq j$, and if we observe a total of m words from some collection of documents with $\theta_1 \geq (1+\delta)(1/2+\eta)$ then about $m = \Omega(\delta^{-2}\log k)$ words will suffice to find a suitable vector v.

Remarks: Similar to the case of GMM, a spectral algorithm using whitening procedure to estimate LDA components have been presented before in Anandkumar et al. (2012). Again the main difference with our Whitening algorithm being the fact that in Anandkumar et al. (2012) the second matrix is constructed by taking a random projection of the third order moment tensor *Triples*, and in Algorithm 1 this is constructed as a projection onto v. Empirically this results is a more stable algorithm due to guaranteed singular value separation. The Cancellation algorithm has not been previously studied in LDA model.

3.3 Mixed Regression

The model: In mixed linear regression the mixture samples generated are of the form $y = \langle x, \mu_i\rangle + \xi$, where $x \sim \mathcal{N}(0, I)$ and noise $\xi \sim \mathcal{N}(0, \sigma^2)$. As before, a sample is generated using the i-th linear component μ_i, with probability α_i. We have access to the observations (y, x) but the particular μ_i and ξ are unknown. Hence the conditional density $g(x, y; \mu_i, \sigma)$ is a multivariate Gaussian where $x \sim \mathcal{N}(0, I)$, $y \sim \mathcal{N}(0, \|\mu_i\|^2 + \sigma^2)$, and $\mathrm{Cov}(x, y) = \mu_i$.

Matrices A and B: To compute A and B, we consider the following moments (for more detailed derivations, see Appendix C):

$$M_{1,1} = \mathbb{E}[yx] = \sum_{i=1}^{k} \alpha_i \mu_i$$

$$M_{2,2} = \mathbb{E}[y^2 xx^T] = 2\sum_{i=1}^{k} \alpha_i \mu_i \mu_i^T + \sum_{i=1}^{k} \alpha_i(\sigma^2 + \|\mu_i\|^2) I$$

$$M_{3,1} = \mathbb{E}[y^3 x] = 3\sum_{i=1}^{k} \alpha_i(\sigma^2 + \|\mu_i\|^2)\mu_i$$

$$M_{3,3} = \mathbb{E}[y^3\langle x, v\rangle xx^T] = 6\sum_{i=1}^{k} \alpha_i \langle \mu_i, v\rangle \mu_i \mu_i^T + \left(M_{3,1}v^T + vM_{3,1}^T + \langle M_{3,1}, v\rangle I\right)$$

Let τ^2 be the smallest singular value of the matrix $M_{2,2}$. Then we can compute m, A, B as follows.

$$\begin{aligned} m &= M_{1,1}, \quad A = \frac{1}{2}(M_{2,2} - \tau^2 I) \\ B &= \frac{1}{6}(M_{3,3} - (M_{3,1}v^T + vM_{3,1}^T + \langle M_{3,1}, v\rangle I)) \end{aligned}$$

As in the previous cases with finite samples the estimates $\hat{m}, \widehat{A}, \widehat{B}$ can be computed by taking their empirical expectations e.g., $\widehat{M}_{1,1} = \widehat{\mathbb{E}}[yx] = \frac{1}{n}\sum_{i=1}^{n} \hat{y}_i \hat{x}_i$ and so on, where $(\hat{y}_i, \hat{x}_i)$ denote the i-th sample.

Examples of v: Suppose we are given a few random labeled examples from the first component. Then assuming $\|\mu_1\|^2 > \langle \mu_1, \mu_i\rangle + \delta$, $\|\mu_i\|^2 \leq R$, similar to the GMM case we can estimate a $v := \frac{1}{\ell}\sum_{j=1}^{\ell} \hat{y}_j \hat{x}_j$ using only $\ell = \Omega\left(R^4\delta^{-2}\log k\right)$ labeled samples so that $\langle \mu_1, v\rangle > \langle \mu_i, v\rangle$ holds with high probability.

Remarks: Our construction of the second matrix B is a consequence of some new moment results for the mixed linear regression model. We present these detailed moment derivations in Appendix C.4. This also results in improved sample complexity bounds over previous moment based algorithms (discussed in Section 3.5).

3.4 Subspace Clustering

The model: Besides the mixture parameters $\alpha_1, \ldots, \alpha_k$, the subspace clustering model has parameters $U_1, \ldots, U_k \in \mathbb{R}^{d\times m}$ and $\sigma \in \mathbb{R}$, where the matrices $U_1, \ldots, U_k$ have orthonormal columns. The conditional distribution $g(\cdot; U_i)$ is a standard Gaussian variable supported on the column space of U_i, plus independent Gaussian noise. More precisely, we sample $y \sim \mathcal{N}(0, I_d)$ and set $x = U_i U_i^T y + \xi$, where $\xi \sim \mathcal{N}(0, \sigma^2 I_d)$ is independent of y.

Matrices A and B: The subspace clustering model does not quite fit into the basic method of Section 2; one motivation for presenting it is to show that the basic ideas in Section 2 are more flexible than they first appear. Suppose $v \in \mathbb{R}^d$ satisfies $\|U_1^T v\| > \|U_i^T v\|$

for all $i \neq 1$. We consider

$$\begin{aligned} A &:= \mathbb{E}[xx^T] - \sigma^2 I_d = \sum_{i=1}^{k} \alpha_i U_i U_i^T \\ B &:= \mathbb{E}[\langle x, v\rangle^2 xx^T] - \sigma^2 v^T A v I_d - \sigma^2 \|v\|^2 A - \sigma^4(\|v\|^2 I_d + vv^T) - 2\sigma^2(Avv^T + vv^T A) \\ &= \sum_{i=1}^{k} \alpha_i \|U_i^T v\|^2 U_i U_i^T + 2\sum_{i=1}^{k} \alpha_i U_i U_i^T vv^T U_i U_i^T \end{aligned}$$

and their empirical versions $\hat{A}$ and $\hat{B}$ (the computation giving the claimed formula for B is carried out in Appendix C). Now with these $\hat{A}$ and $\hat{B}$, we can recover the subspace U_1 using Algorithm 3. This algorithm uses the same principle behind the whitening method in Section 2.1.1, the key difference is that here we pick the top m eigenvectors of the whitened B matrix.

Algorithm 3 Subspace clustering algorithm

Input: $\hat{A}, \hat{B}$
Output: $\hat{U}$

1: let $\{\sigma_j, v_j\}$ be the singular values and singular vectors of $\hat{A}$, in non-increasing order
2: let V be the $d \times mk$ matrix whose jth column is v_j
3: let D be the $mk \times mk$ diagonal matrix with $D_{jj} = \sigma_j$
4: let $Y = [u_1, \ldots, u_m]$ be the matrix of m largest eigenvectors of $D^{-1/2}V^T\hat{B}VD^{-1/2}$
5: let $Z = VD^{1/2}Y$
6: let the columns of $\hat{U}$ be the m eigenvectors of the matrix ZZ^T

The following perturbation theorem guarantees that if the side information vector v is substantially more aligned with the subspace spanned by U_1 than it is with any other subspace, and the matrices A, B are estimated within ϵ accuracy, then Algorithm 3 can recover the required subspace with a small error.

Theorem 6 *Suppose that $\|\hat{A} - A\| \leq \epsilon$ and $\|\hat{B} - B\| \leq \epsilon$. Suppose that the side information vector v satisfies $\|U_i v\|^2 \leq (1/3 - \delta)\|U_1 v\|^2$. Then output $\hat{U}$ of Algorithm 3 satisfies*

$$\|\hat{U}\hat{U}^T - U_1 U_1^T\| \leq C\epsilon\alpha_1^{-1}\sigma_1(A)^2\sigma_{mk}(A)^{-2}\delta^{-1}.$$

We prove Theorem 6 in Appendix F. Note that the conditions on v can be satisfied if the spaces U_i satisfy a certain affinity condition and we have a few labelled samples from U_1. Specifically, suppose that $\langle u, w\rangle < (\frac{1}{\sqrt{3}} - \eta)\|u\|\|w\|$ for every $u \in U_1$ and $w \in U_i$, $i \neq 1$. Then any $v \in U_1$ will satisfy the assumption of Theorem 6. Hence, a single labelled sample from U_1 (or several—depending on η—noisy samples) is enough to find a suitable v.

Remarks: To the best of our knowledge Algorithm 3 is the first moment based algorithm for the subspace clustering model. The detailed moment derivations are presented in Appendix C.5. Also our generative model allows samples to be noisy, hence they do not lie exactly on the subspace but close to it. Such a setting has not been considered in most subspace clustering literature.

3.5 Comparison

In this section we compare the theoretical performance of the Whitening and Cancellation algorithms with other algorithms. Both Whitening and Cancellation algorithms require estimating the quantities m, A, B by computing moments from the samples. Therefore the sample complexity primarily depends on how well these quantities concentrate. We compute the specific sample complexities for each model in Appendix G.

For Gaussian mixture model the sample complexity of our algorithm scales as $\tilde{\Omega}(d\epsilon^{-2}\log d)$ similar to moment based algorithm by Hsu and Kakade (2013) and tensor decomposition based algorithm by Anandkumar et al. (2014). In terms of runtime the Whitening algorithm is faster than the tensor decomposition based algorithm by Anandkumar et al. (2014). This can be viewed as follows. The first step in both the algorithms take $O(d^2k)$ time to compute the whitening matrix and in subsequent whitening steps. However, computing the largest eigenvector in Algorithm 1 takes only $O(k^2)$ time, faster than $O(k^5 \log k)$ time required for rank-k tensor power iteration (we also verify this in our experiments in Section 4).

In LDA topic model our algorithms have a sample complexity of $\tilde{\Omega}(\epsilon^{-2}\log d)$, again similar to tensor decomposition based algorithm by Anandkumar et al. (2014), and non-negative matrix factorization (NMF) based algorithm by Arora et al. (2013). The Whitening algorithm again is faster than tensor decomposition as argued for GMM case. The NMF based algorithm using optimization based RecoverKL/RecoverL2 procedures also has a runtime of $O(d^2k)$ similar to our algorithms (in Section 4 again we observe our algorithm to be faster in practice). The spectral topic modeling algorithm in Anandkumar et al. (2012) also has a computation complexity $O(d^2k)$ similar to our algorithms. However, its sample complexity has a high $\Omega(k^5)$ dependence on the number of components. This spectral algorithm also suffer from instability in practice due to the random projection step (as noted in Anandkumar et al. 2014).

In the case of mixed linear regression again our method has a sample complexity of $\tilde{\Omega}(d\epsilon^{-2}\log d)$ similar (upto log factors) to the convex optimization based approach by Chen et al. (2014), alternating minimization based approach by Yi et al. (2014), but better than tensor decomposition based method of Sedghi et al. (2016) which has a sample complexity of $\tilde{\Omega}(d^3\epsilon^{-2})$. However unlike the convex optimization and alternating minimization based techniques our method is also applicable when the number of components $k > 2$. As argued in GMM case the Whitening algorithm is again faster than the tensor algorithm by Sedghi et al. (2016).

Subspace clustering algorithms like greedy subspace clustering by Park et al. (2014), optimization based algorithms by Elhamifar and Vidal (2009), Soltanolkotabi and Candes (2012), requires the samples to exactly lie on a subspace. In contrast our moment based algorithm works even when the samples are noisy and perturbed from the actual subspace. Our subspace clustering algorithm also has a sample complexity of $\tilde{\Omega}(m\epsilon^{-2}\log d)$ which is similar (up to log factors) to greedy subspace clustering algorithm by Park et al. (2014).

We note that it is possible to use approximation methods like randomized svd to further speed up the Whitening, Cancellation and tensor decomposition based algorithms by Anandkumar et al. (2014), however this will result in decreased accuracy in both algorithms. We refer to Huang et al. (2015) for such stochastic optimization, and parallelization techniques used to speed up the tensor algorithms.

In a setting where side information is provided on each of the k components, observe that we can run the Whitening algorithm independently for each of the k components, possibly in parallel. Hence we can recover all k components, without loosing the runtime advantage of the Whitening algorithm. We demonstrate this application on real data set in Section 4.2. In terms of the overall computation time, it can be shown that running the Whitening algorithm for all k components is still faster than the tensor decomposition based algorithm by Anandkumar et al. (2014), when $k = \Omega(n^{\frac{1}{3}} d^{\frac{1}{3}})$.

4. Experiments

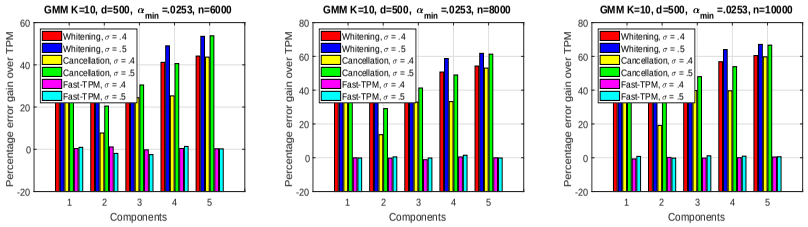
In this section we present the empirical performance of our Whitening, Cancellation, and Subspace clustering algorithms. We consider three of the settings: the Gaussian Mixture Model (GMM), and Latent Dirichlet Allocation (LDA), and Subspace clustering, and validate our algorithms on both real and synthetic data sets.

4.1 Synthetic Data Set

First we compare the sample complexity and runtime of our algorithms with the robust tensor decomposition algorithm by Anandkumar et al. (2014), which is based on tensor power iteration, for learning mixture models (we refer to this as the TPM algorithm). Our second baseline algorithm is a faster heuristic of TPM where we start the tensor power iterations initialized with side information vector v, and recover just the first component. We refer this as the Fast-TPM algorithm. For the Cancellation algorithm we compute the optimum λ for cancellation using two different techniques as follows. First, let $\widehat{Z}'_\lambda = V^T \widehat{Z}_\lambda V$, where V is the matrix of top k singular vectors of $\widehat{A}$. In the first method, we perform a line search over positive λ to find the minimum λ such that $\sigma_k(\widehat{Z}'_\lambda)$ falls below certain threshold. This method works well in GMM case. In a second method we minimize the convex function $\|\widehat{Z}'_\lambda\|_* + \lambda$, subject to $\lambda \geq 0$. This method performs better in the case of LDA. Note that for the Cancellation algorithm after estimating λ, instead of using m and A to find μ_1 we can follow the same steps using $m' = Av$ and B to recover μ_1. Theoretically it has the same performance, however empirically we observe this to work slightly better and we use this version for our experiments. We implement all algorithms for our synthetic data experiments using MATLAB.

Performance metric: We compute the estimation error of parameter μ_1 as $\mathcal{E} = \|\hat{\mu}_1 - \mu_1\|$. In our figures we plot the quantity "percentage relative error gain" which is defined as $G = 100(\mathcal{E}_T - \mathcal{E}_A)/\mathcal{E}_T$, where $\mathcal{E}_T$ is the TPM error and $\mathcal{E}_A$ is the error for Whitening / Cancellation / Fast-TPM algorithm. Note that a positive error gain implies that the TPM error is greater than that of the competing algorithm. In the subspace clustering model we plot similar percentage relative error gain over the baseline k-means algorithm.

Gaussian mixture model: We generate synthetic data sets for GMM with different k, d, α_i, σ, and v. Figure 1 shows the percentage relative error gains of the Whitening, Cancellation, and Fast-TPM algorithms over the TPM algorithm in a GMM with various values of k, d, α_i, σ, and n. The μ_i were generated randomly over the sphere of norm $r = 10$. We define $\alpha_{min} := \min_i \alpha_i$. The side information vector v was chosen as follows. Let $\{v_1, \ldots, v_k\}$ be a orthonormal basis of span$\{\mu_1, \ldots, \mu_k\}$, such that $\{v_2, \ldots, v_k\} \in$ span$\{\mu_2, \ldots, \mu_k\}$. Then we



(a) (b) (c)

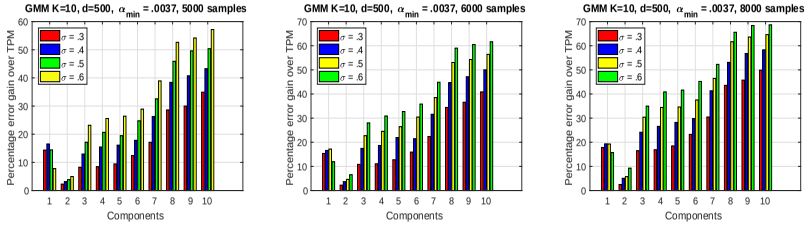
Figure 1: Figure showing the percentage relative error gain by the Whitening, Cancellation, and Fast-TPM algorithm over the TPM algorithm for 5 components of increasing size, in a GMM with $k = 10, d = 500, \sigma \in \{.4, .5\}$, and three different sample complexities (a) $n = 6000$ (b) $n = 8000$ (c) $n = 10000$. Our algorithms shows increasingly better gain over TPM and Fast-TPM as α_i, σ and n increase.

choose $v = \sqrt{\gamma} v_1 + \sqrt{(1-\gamma)/(k-1)} \sum_{i=2}^{k} v_i$ for some $\gamma \in (0, 1)$ such that the condition $\langle \mu_1, v \rangle > \langle \mu_i, v \rangle$ is satisfied. We observe that in all the cases, our algorithms have lower error (positive error gain) than both the tensor algorithms. Moreover, our methods' advantage increases with increasing proportion α_i, increasing sample size n, and increasing variance σ. We also observe that the Fast-TPM algorithm has the same error performance as TPM (error gain close to zero).

Figure 2 gives an example where the Whitening algorithm can successfully recover even rare components. Here we consider a GMM with $k = 10, d = 500$ with the rarest component having probability $\alpha_{min} = .0037$. Again we observe positive relative error gains over TPM algorithm for increasing number of samples n.

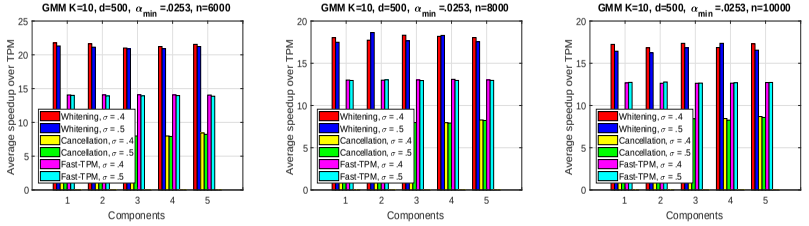
In Figure 3 we plot the speedup of the algorithms over TPM, and observe that the Whitening and Cancellation algorithms are much faster (high speedup) than the TPM algorithm. We also observe that the Fast-TPM algorithm is faster than TPM and Cancellation algorithms, but slower than Whitening algorithm. Note that, while it is also possible to speed up the basic TPM algorithm compared here using techniques such as randomized svd and stochastic tensor gradient descent [Huang et al. 2015], such approximate methods will reduce the overall accuracy. Moreover the randomized svd techniques can also be applied to the search algorithms presented in this paper, to obtain further speedups.

Topic Modeling: We generate a synthetic LDA document corpus according to the model in Blei et al. (2003). The lengths of the documents are generated using a Poission(L) distribution where L is the mean document length. In Figure 4 we plot the percentage relative error gain of the Whitening, Cancellation, and Fast-TPM algorithms over the TPM algorithm. Our side information was a labeled word w satisfying $\mu_1(w) > \mu_i(w)$ for $i \neq 1$. Again we observe positive error gains over the TPM algorithm. Although the Fast-TPM algorithm sometimes perform better than TPM for more frequent topics, the Whitening



(a) (b) (c)

Figure 2: Figure showing the percentage relative error gain of the Whitening algorithm over the TPM algorithm in presence of rare components ($\alpha_{min} = .0037$), for a GMM with $k = 10, d = 500, \sigma \in \{.3, .4, .5, .6\}$, and number of samples (a) $n = 5000$ (b) $n = 6000$ (c) $n = 8000$. The Whitening algorithm recovers even the rarest component with increasing error gain over TPM as the number of samples increase.

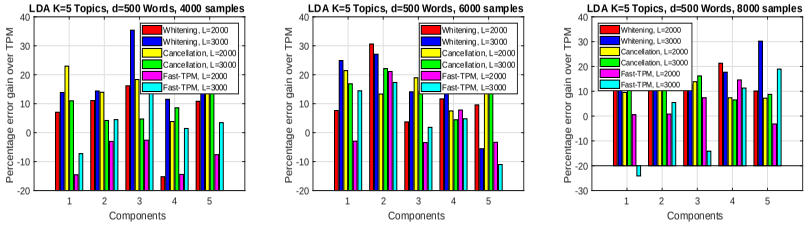


(a) (b) (c)

Figure 3: Figure showing the average speedup of Whitening, Cancellation, and Fast-TPM algorithms over TPM, for 5 components of increasing size, in a GMM with $k = 10, d = 500, \sigma \in \{.4, .5\}$, and three different sample complexities (a) $n = 6000$ (b) $n = 8000$ (c) $n = 10000$. The Whitening algorithm is the fastest.

algorithm still outperforms it. Note that the performance varies across topics since the probability of the labeled word is different for each topic.

Subspace Clustering: We generate synthetic data for the subspace clustering model described in section 3.4 using parameters $d = 500, k = 5, m = 10$, and $\alpha_i \in [.1, .3]$. First we generate $k = 5$ random subspaces with orthonormal basis $\{U_i\}_{i=1}^k$, each of dimension $m = 10$. Then we generate random points on these subspaces, and add white Gaussian perturbations with $\sigma \in \{.1, .2\}$. We choose the side information vector v similar to the sensitivity experiment in GMM, and ensuring $\|U_1^T v\| > \|U_i^T v\|$, for $i \neq 1$. Note that due to the added Gaussian noise, our samples do not lie exactly on the subspaces $\{U_i\}_{i=1}^k$, but close



(a) (b) (c)

Figure 4: Figure showing the percentage relative error gain in each component of the Whitening, Cancellation, and Fast-TPM algorithms over the TPM algorithm in an LDA model with $k = 5, d = 500$, mean document length $L \in \{2000, 3000\}$, and number of documents (a) $n = 4000$ (b) $n = 6000$ (c) $n = 8000$. The Whitening algorithm show an improvement over TPM and Fast-TPM with increasing samples.

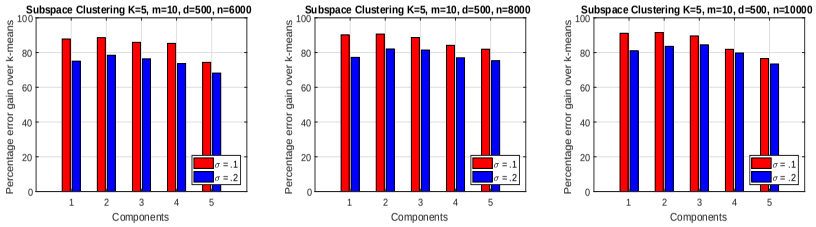
to it. Traditional subspace clustering algorithms, which assume points to lie exactly on the subspace, may not perform well. The TPM algorithm is also not well suited for this model since (a) the required moment tensor will be of 4^{th} order resulting in high computation cost (b) even if mk basis of the tensor are recovered, finding the target subspace will involve a further combinatorial search of $\binom{mk}{m}$ subspaces and finding the one having the strongest projection of v. Therefore we choose the k-means algorithm as our baseline for this model and compare with Algorithm 3. First we compute k clusters using k-means, then we find an m dimensional basis for each cluster using svd, finally we choose the target subspace as the one having the largest projection of v. If $\widehat{U}_1$ is the estimated orthonormal basis for the target subspace U_1, we compute the error as $\mathcal{E} = \|\widehat{U}_1\widehat{U}_1^T - U_1U_1^T\|/\|U_1U_1^T\|$.

Figure 5 shows that Algorithm 3 has a much better error performance over k-means. In the speedup plots in Figure 6 we also observe that our subspace search algorithm is over $4X$ times faster than k-means.

4.2 Real Data Sets

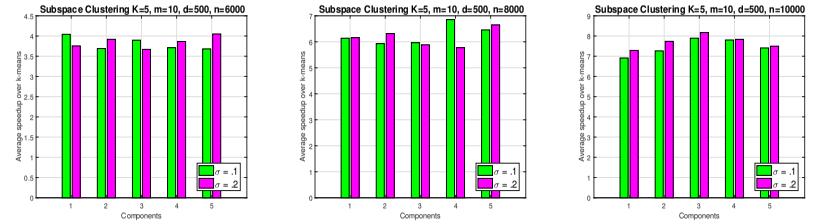
Topic Modeling: In this section we compare the performance of Whitening algorithm with a recent non-negative matrix factorization based topic modeling algorithm by Arora et al. (2013) (we refer this as NMF algorithm), and also the semi-supervised version of this NMF algorithm (we refer to this as SS-NMF). We test on two real large data sets; (a) New York Times news article data set [UCI 2008] (300,000 articles) (b) Yelp data set of business reviews [Yelp 2014] (335,022 reviews). We run both algorithms for $k = 100$ topics. For this experiment we do not consider the TPM algorithm by Anandkumar et al. (2014) since its runtime with $k = 100$ topics becomes extremely large on these data sets.[1]

1. To be more precise, with just $k = 10$ topics, the tensor algorithm takes 908 seconds in NY Times data set, compared to just 188 seconds for the Whitening algorithm (using MATLAB).



(a) (b) (c)

Figure 5: Figure showing the percentage relative error gain by our subspace search algorithm (Algorithm 3) over k-means for 5 components of increasing size, in a subspace clustering model with $k = 5, m = 10, d = 500, \sigma \in \{.1, .2\}$, and three different sample complexities (a) $n = 6000$ (b) $n = 8000$ (c) $n = 10000$. Our algorithm shows much better error performance than k-means.



(a) (b) (c)

Figure 6: Figure showing the average speedup of our subspace search algorithm (Algorithm 3) over k-means, for 5 components of increasing size, in a subspace clustering model with $k = 5, m = 10, d = 500, \sigma \in \{.1, .2\}$, and three different sample complexities (a) $n = 6000$ (b) $n = 8000$ (c) $n = 10000$. Our subspace clustering algorithm shows high speedup over k-means.

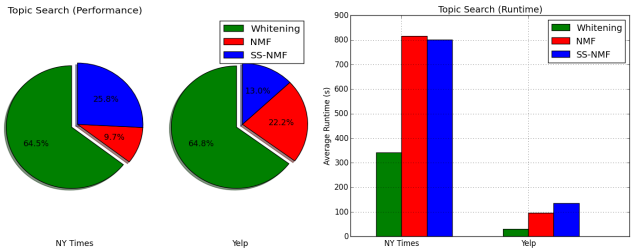
In contrast, the NMF algorithm is known to be faster, and produce topics of comparable quality to more popular variational inference based algorithms [Blei et al. 2003]. The side information for this experiment are chosen as follows. First from the set of topics produced by NMF algorithm we choose a subset of interpretable topics, then we choose labeled words representative of these topics. We test with a set of 62 labeled words for NY Times data set and 54 labeled words for Yelp data set. Note that given labeled word w_l the whitening algorithm produces one topic distribution μ_1, but the NMF algorithm finds k topics. Therefore for NMF algorithm the target topic i is the one which has the highest probability of the labeled word i.e., $\mu_i(w_l)$. For the semi-supervised NMF we first compute

the weighted word-word co-occurrence matrix Q_w where we re-weigh each document by the normalized frequency of the labeled word w_l. Then we apply the NMF algorithm [Arora et al. 2013] on this weighted matrix Q_w. All three algorithms were implemented in Python.

Performance metric: We compare the quality of the topics returned by Whitening, NMF, and SS-NMF algorithms using the pointwise mutual information (PMI) score, known to be a good metric for topic coherence [Newman et al. 2010; Röder et al. 2015]. However in order to also capture the relevance of the estimated topic to the labeled word we compute PMI score for topic i as,

$$PMI(\text{topic i}) = \frac{1}{20} \sum_{w \in \mathcal{T}_{20}^i} \log \frac{p(w_l, w)}{p(w_l)p(w)}$$

where w_l is the labeled word, $\mathcal{T}_{20}^i$ is the set of top 20 words in the i-th topic. The probabilities $p(w_l, w), p(w), p(w_l)$ are computed over a larger data set of English Wikipedia articles to reduce noise [Newman et al. 2011]. For whitening algorithm we choose $\alpha_0 = .01$. Note that other supervised topic modeling algorithms e.g. supervised LDA by Mcauliffe and Blei (2008), labeled LDA by Ramage et al. (2009) require a much stronger notion of side-information than just labeled words, hence we could not compare with them.



(a) (b)

Figure 7: Figure comparing the performance of Whitening, NMF [Arora et al. 2013], and semi-supervised NMF (SS-NMF) algorithms on NY Times and Yelp data sets. (a) Topics estimated by Whitening algorithm have the best PMI score in 40 out of 62 labeled words for NY Times data set, and 35 out of 54 labeled words in Yelp data set. (b) Whitening shows more than 2X speedup over competing algorithm in both data sets.

In Figure 7 (a) we plot the percentage of labeled words for which each algorithm has the best PMI score. Observe that for most labeled words (40 out of 62 labeled words for NY Times data set, and 35 out of 54 labeled words in Yelp data set) the Whitening algorithm estimates topic with better PMI score over NMF and SS-NMF algorithms. The Whitening algorithm is also more than twice as fast as NMF and SS-NMF[2] as shown in Figure 7 (b).

2. For large corpus the NMF algorithm runs much faster than Gibbs sampling and variational inference based algorithms [Arora et al. 2013].

A complete list of topics and PMI scores returned by the algorithms for every labeled word is presented in Tables 2, 3 of Appendix B. Notice that the Whitening algorithm often estimates more coherent topics which are more relevant to the given labeled word than topics produced by the NMF/SS-NMF algorithm. For example in NY Times data set with the labeled word *student* the Whitening algorithm returns top five words in the topic as *student, school, teacher, percent, program*; however those returned by NMF algorithm are *test, school, student, ignore, export*; and those by SS-NMF algorithm are *student, university, shooting, shot, rampage*.

Parallel image segmentation: One method to perform image segmentation is to use GMM clustering. In this experiment we demonstrate how GMM search algorithm can be used to parallelize image segmentation in vision applications. For this we consider the BSDS500 data set introduced in Arbelaez et al. (2011) and choose a subset of 70 images having less than 4 segments in the ground truth. Note that this data set has up to six ground truth segmentation by human users for each image. We randomly choose one pixel from each segment in ground truth as side-information v. We compare our Whitening algorithm with the seeded k-means clustering [Basu et al. 2002] where the centers are initialized by these side-information pixels (we refer to this as s-Kmeans). The Whitening algorithm uses one pixel from the i-th cluster to compute μ_i, in parallel for every i, and then it assigns each pixel to its closest μ_i. The segmentation quality is compared using normalized mutual information (NMI) metric [Manning et al. 2008]. To avoid local minimum in s-Kmeans we consider the maximum NMI over 5 initializations of side-information for each ground truth, and then we compute average NMI over all ground truths for an image.

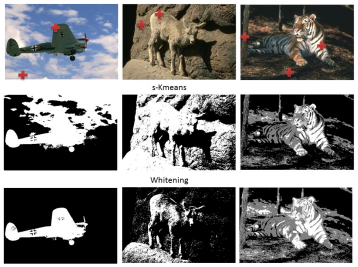


Figure 8: Figure comparing the performance of image segmentation by Whitening (row 3) and s-Kmeans (row 2) algorithms, with images selected from the BSDS500 data set. The side-information pixels are shown in red plus in the original image (row 1). In the segmented images (rows 2, 3) the segments are shown in different shades. Observe that the Whitening algorithm often isolates the foreground segment better than s-Kmeans.

We summarize our result in Table 1. Observe that the Whitening algorithm has a slightly better NMI performance over s-Kmeans in the BSDS test data set and similar performance

Data set	N	N_W	N_K	T_W (s)	T_K (s)	$\overline{NMI}_W$	$\overline{NMI}_K$
BSDS test	30	17	13	6.7	81.5	0.17	0.13
BSDS train	25	12	13	8.2	89.8	0.15	0.15
BSDS val	15	8	7	10.6	117.2	0.11	0.09

Table 1: Table comparing the performance of Whitening and s-Kmeans algorithm on BSDS data set. N is the total number of images, N_W is the number of images where segmentation produced by Whitening has a better NMI than s-Kmeans, and N_K is the number of images where segmentation of s-Kmeans has a better NMI. T_W is the median runtime of Whitening algorithm and T_K is the median runtime of s-Kmeans. $\overline{NMI}_W$ and $\overline{NMI}_K$ are the median NMI scores for the Whitening and s-Kmeans algorithms respectively. Whitening runs much faster than s-Kmeans.

in BSDS train and BSDS val data sets. However the Whitening algorithm runs an order of magnitude faster than s-Kmeans.

5. Conclusion and Discussion

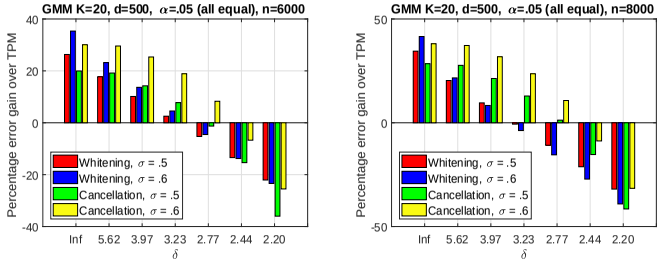
In this paper we developed a new, simple and flexible framework for incorporating side information into mixture model learning. The underlying motivation was to provide a principled way to take into account extra input (e.g. generated by human data analysts etc.). Even for cases where this input is very limited compared to the size/dimensionality of the data, we show meaningful statistical and computational performance improvement over baseline unsupervised and semi-supervised methods. More generally, developing methods which work with very limited human input is a promising research endeavor, in our opinion.

Acknowledgments

We would like to acknowledge support from NSF grants CNS-1320175, 0954059, ARO grants W911NF-15-1-0227, W911NF-14-1-0387, W911NF-16-1-0377, and the US DoT supported D- STOP Tier 1 University Transportation Center. The authors also acknowledge the Texas Advanced Computing Center [TACC 2018] at The University of Texas at Austin for providing HPC resources that have contributed to the research results reported within this paper.

Appendix A. More Experiments for Gaussian Mixture Models

In Figure 9 we show the sensitivity of the Whitening and Cancellation algorithms in GMM with $k = 20, d = 500$, all equal probability components, and two different values of σ and n. Observe that the percentage error gain of the algorithms decreases with decreasing values of $\delta = \min_{i \neq 1} \frac{\langle \mu_1, v \rangle}{\langle \mu_i, v \rangle}$, as we would expect, and it eventually becomes negative when the performance become worse than TPM algorithm. Also here the Cancellation algorithm shows lesser sensitivity, hence better performance compared to the Whitening algorithm.



(a) (b)

Figure 9: Sensitivity plots showing how the percentage relative error gain of the Whitening and Cancellation algorithms over the TPM algorithm decrease with decreasing values of the parameter $\delta = \min_{i \neq 1} \frac{\langle \mu_1, v \rangle}{\langle \mu_i, v \rangle}$, in GMM with $k = 20, d = 500$, all equal probability components, for different values of variance $\sigma \in \{.5, .6\}$, and two different sample complexities (a) $n = 6000$ (b) $n = 8000$.

Appendix B. Complete Results on New York Times and Yelp Data Set

In this section we provide more detailed result of our experiments on NY Times and Yelp data sets. In Tables 2, 3 we show for every labeled word, the top five words in the topics computed by Whtening, NMF, and SS-NMF algorithms along with their corresponding PMI scores.

Table 2: Results of topic search by Whitening and NMF algorithms on NYtimes data set of 300,000 news articles using $K = 100$ topics and 62 labeled words.

NY Times data set							
Label word	Algo	topword-1	topword-2	topword-3	topword-4	topword-5	PMI
passenger	Whitening	flight	security	passenger	airport	hour	0.1424
	NMF	security	government	official	percent	bill	0.0499
	SSNMF	passenger	plane	flight	fire	crash	0.1711
coach	Whitening	coach	season	job	team	head	0.2637
	NMF	team	coach	season	player	jet	0.1740
	SSNMF	coach	arrived	assistant	defenseman	ended	0.1756
art	Whitening	information	question	today	eastern	daily	0.0255
	NMF	art	show	dessert	book	home	0.0769
	SSNMF	art	artist	show	painting	museum	0.1250
campaign	Whitening	campaign	al gore	money	political	republican	0.1530
	NMF	al gore	campaign	george bush	president	bush	0.1608
	SSNMF	nra	florida	article	senator	presidential	0.0926
energy	Whitening	corp	meeting	list	dividend	partial	0.0815
	NMF	corp	meeting	list	group	dividend	0.0570
	SSNMF	partial	energy	dividend	meeting	corp	0.0254
tax	Whitening	tax	cut	taxes	percent	income	0.2126
	NMF	graf	president	bush	mail	information	0.0722
	SSNMF	tax	income	cut	taxes	site	0.2279
chef	Whitening	cup	minutes	food	article	add	0.0227
	NMF	buy	panelist	flavor	thought	product	0.0130
	SSNMF	tobacco	chef	restaurant	pastry	article	0.1495
oil	Whitening	oil	cup	minutes	prices	companies	0.1460
	NMF	oil	million	prices	percent	market	0.0928
	SSNMF	oil	company	listing	largest	brazil	0.0902
court	Whitening	court	case	law	decision	lawyer	0.2288
	NMF	official	court	case	attack	government	0.1285
	SSNMF	chicago	court	decision	ruling	justices	0.1834
election	Whitening	election	ballot	vote	voter	florida	0.2132
	NMF	election	ballot	al gore	bush	vote	0.2155
	SSNMF	gained	election	article	presidential	independence	0.1702
lawyer	Whitening	case	court	lawyer	death	trial	0.1830
	NMF	official	court	case	attack	government	0.1017
	SSNMF	lawyer	rat	legal	client	jokes	0.1314
anthrax	Whitening	mail	official	anthrax	attack	worker	0.0600
	NMF	anthrax	official	mail	worker	letter	0.0156
	SSNMF	anthrax	poverty	cb	show	return	-0.0776
golf	Whitening	tiger wood	shot	round	player	tour	0.1288
	NMF	tiger wood	shot	round	player	play	0.1356
	SSNMF	misstated	master	tee	hit	golf	0.1356
bacteria	Whitening	mail	anthrax	official	test	found	-0.0763
	NMF	anthrax	official	mail	worker	letter	-0.1097
	SSNMF	mas	bacteria	con	una	anos	-0.2420
film	Whitening	film	movie	director	character	actor	0.1906
	NMF	article	misstated	new york	company	million	0.0288
	SSNMF	kiss	film	actress	article	role	0.1295
tourist	Whitening	million	www	percent	building	night	0.0481
	NMF	team	tour	lance armstrong	won	race	-0.0405
	SSNMF	tourist	million	visitor	official	campaign	0.0995
horse	Whitening	race	won	win	run	track	0.1129
	NMF	race	won	horse	win	kentucky derby	0.1338
	SSNMF	horse	truck	road	official	killed	0.0433
republican	Whitening	campaign	george bush	bush	election	republican	0.2449
	NMF	al gore	campaign	george bush	president	bush	0.1868
	SSNMF	republican	democrat	democratic	house	parties	0.1053
computer	Whitening	computer	system	microsoft	program	software	0.1904

Label word	Algo	topword-1	topword-2	topword-3	topword-4	topword-5	PMI
	NMF	company	computer	microsoft	system	companies	0.1533
	SSNMF	computer	chip	mail	program	buy	0.1903
palestinian	Whitening	palestinian	israel	israeli	yasser arafat	peace	0.2189
	NMF	palestinian	israel	official	israeli	yasser arafat	0.1950
	SSNMF	palestinian	reformer	reform	authority	arab	0.1519
movie	Whitening	film	movie	director	character	actor	0.1492
	NMF	film	show	actor	movie	thought	0.0901
	SSNMF	red sox	movie	interview	seattle	host	0.0388
tennis	Whitening	player	play	won	game	women	0.1054
	NMF	game	play	player	point	andre agassi	0.1187
	SSNMF	motif	tennis	season	pros	image	0.1480
fight	Whitening	won	night	fight	win	sport	0.0566
	NMF	fight	mike tyson	lennox lewis	million	round	0.1181
	SSNMF	fight	pound	fighter	beat	boxing	0.1254
music	Whitening	music	song	record	album	band	0.2298
	NMF	music	company	million	companies	napster	0.0812
	SSNMF	music	mp3	customer	digital	online	0.0150
tablespoon	Whitening	cup	minutes	add	oil	tablespoon	0.0608
	NMF	cup	minutes	add	tablespoon	water	0.0431
	SSNMF	coffee	bean	tablespoon	cup	ground	-0.0765
nuclear	Whitening	bush	US	official	system	administration	0.1223
	NMF	official	bush	government	US	nuclear	0.1356
	SSNMF	ibm	nuclear	computer	research	fastest	-0.0253
racing	Whitening	race	car	driver	team	season	0.1443
	NMF	car	race	driver	team	season	0.1319
	SSNMF	sport	file	los angeles	racing	notebook	-0.0640
war	Whitening	military	taliban	war	afghanistan	us	0.0916
	NMF	taliban	official	afghanistan	government	us	0.0796
	SSNMF	russian	war	chechnya	army	veteran	0.1296
quarterback	Whitening	yard	season	game	play	team	0.2389
	NMF	game	team	play	yard	season	0.1773
	SSNMF	effort	quarterback	ucla	heroic	alabama	0.1472
stock	Whitening	stock	market	percent	company	fund	0.1585
	NMF	percent	stock	market	company	companies	0.1338
	SSNMF	stock	market	price	shares	investment	0.0507
ball	Whitening	game	run	yard	play	hit	0.1782
	NMF	run	game	inning	hit	season	0.1361
	SSNMF	ball	hit	run	inning	home	0.1708
patient	Whitening	patient	doctor	care	health	drug	0.2532
	NMF	official	virus	percent	new york	found	0.1003
	SSNMF	patient	study	doctor	article	brain	0.1334
champion	Whitening	won	win	round	shot	tiger wood	0.1029
	NMF	fight	mike tyson	lennox lewis	million	round	0.0955
	SSNMF	olympic	champion	final	meet	medalist	0.1177
business	Whitening	business	company	question	information	companies	0.0887
	NMF	information	eastern	commentary	daily	business	0.0311
	SSNMF	publication	business	send	released	businesses	0.0996
government	Whitening	government	official	country	federal	political	0.1524
	NMF	graf	president	bush	mail	information	0.0767
	SSNMF	program	government	computer	local	newspaper	0.0784
season	Whitening	season	team	game	games	play	0.1799
	NMF	team	game	season	play	games	0.1406

Label word	Algo	topword-1	topword-2	topword-3	topword-4	topword-5	PMI
	SSNMF	season	cotton	fact	simple	variety	0.0626
prison	Whitening	death	case	lawyer	court	trial	0.1333
	NMF	advise	spot	earlier	held	today	-0.0340
	SSNMF	prison	inmates	security	population	bed	0.1472
internet	Whitening	file	spot	internet	read	output	0.0359
	NMF	file	spot	new york	sport	los angeles	0.0228
	SSNMF	wonderful	mail	al gore	george bush	message	0.0766
rain	Whitening	air	part	high	wind	rain	0.1963
	NMF	air	wind	shower	rain	storm	0.1939
	SSNMF	chicago sun times	nominated	rain	east	thought	0.0179
game	Whitening	game	team	play	games	season	0.2000
	NMF	team	game	season	play	games	0.1722
	SSNMF	covering	game	tonight	coverage	celebration	0.0531
voter	Whitening	election	ballot	vote	percent	voter	0.2068
	NMF	election	ballot	al gore	bush	vote	0.1870
	SSNMF	voter	poll	percent	primary	election	0.2067
baseball	Whitening	player	team	season	game	sport	0.1691
	NMF	team	chicago white sox	mariner	season	player	0.1803
	SSNMF	velocity	baseball	air	shot	test	0.0629
student	Whitening	student	school	teacher	percent	program	0.2077
	NMF	test	school	student	ignore	export	0.0729
	SSNMF	student	university	shooting	shot	rampage	0.1396
president	Whitening	president	vice	white house	george bush	executive	0.2116
	NMF	graf	president	bush	mail	information	0.0758
	SSNMF	hedge	president	television	broadway	produced	0.0226
afghan	Whitening	taliban	afghanistan	military	us	war	0.1684
	NMF	taliban	official	afghanistan	government	us	0.1413
	SSNMF	afghan	afghanistan	blanket	friend	country	0.0577
medal	Whitening	team	games	won	women	american	0.1822
	NMF	team	tour	lance arm-strong	won	race	0.0348
	SSNMF	endit	medal	honor	winner	newspaper	0.0786
teacher	Whitening	school	student	teacher	high	program	0.1566
	NMF	test	school	student	ignore	export	0.0388
	SSNMF	teacher	program	pay	school	teaching	0.1499
television	Whitening	show	home	network	television	night	0.1721
	NMF	los angeles daily new	spot	newspaper	new york	show	0.1456
	SSNMF	clinton	home	television	survived	tonight	-0.0090
democratic	Whitening	al gore	campaign	election	political	republican	0.1837
	NMF	al gore	campaign	george bush	president	bush	0.1677
	SSNMF	environmental	democratic	national committee	nominee	fund	0.0813
onion	Whitening	cup	minutes	add	oil	tablespoon	0.1039
	NMF	cup	minutes	add	tablespoon	water	0.1072
	SSNMF	flavor	panelist	ounces	buy	onion	0.1188
campus	Whitening	student	school	college	teacher	program	0.1314
	NMF	game	season	team	play	coach	-0.0595
	SSNMF	campus	operation	aol	building	center	0.0645
car	Whitening	car	driver	race	racing	seat	0.2047
	NMF	car	race	driver	team	season	0.1222

Label word	Algo	topword-1	topword-2	topword-3	topword-4	topword-5	PMI
	SSNMF	car	team	race	driver	winston cup	0.1516
	Whitening	companies	percent	company	business	industry	0.1430
industry	NMF	music	company	million	companies	napster	0.0821
	SSNMF	xxx	show	trade	software	entertainment	0.1161
	Whitening	film	today	system	movie	team	-0.0054
planet	NMF	wire	inadvertently	kill	mandatory	today	-0.0750
	SSNMF	captor	planet	film	kill	astronomer	0.0949
	Whitening	bill	money	member	system	number	0.1257
credit	NMF	bill	tax	bush	member	percent	0.0287
	SSNMF	donation	card	credit	account	voted	0.1382
	Whitening	race	car	driver	won	win	0.1917
race	NMF	car	race	driver	team	season	0.1814
	SSNMF	amazing	race	show	tonight	sit	0.0502
	Whitening	cup	minutes	food	add	oil	0.0499
wine	NMF	wine	wines	percent	company	million	0.0748
	SSNMF	wine	wines	bottle	bottles	age	0.1082
	Whitening	case	death	lawyer	court	trial	0.1952
prosecutor	NMF	official	court	case	attack	government	0.1363
	SSNMF	prosecutor	lawyer	attorney	incorrectly	general	0.1406
	Whitening	team	season	game	player	play	0.1654
team	NMF	team	game	season	play	games	0.1558
	SSNMF	team	qualify	olympic	article	member	0.1530
	Whitening	percent	market	economy	stock	cut	0.1528
economy	NMF	percent	stock	market	company	companies	0.1048
	SSNMF	percent	economy	quarter	rate	recession	0.1452
	Whitening	air	high	part	wind	rain	0.1909
wind	NMF	air	wind	shower	rain	storm	0.1895
	SSNMF	wash	wind	school	winter	white	0.1902
	Whitening	microsoft	computer	system	company	software	0.1981
software	NMF	company	computer	microsoft	system	companies	0.1911
	SSNMF	xxx	software	industry	show	trade	0.1222

Table 3: Results of topic search by Whitening and NMF algorithms on Yelp data set of 335,022 reviews of businesses using $K = 100$ topics and 54 labeled words.

Yelp data set							
Label word	Algo	topword-1	topword-2	topword-3	topword-4	topword-5	PMI
	Whitening	cheese	pizza	time	sandwich	back	0.1842
cheese	NMF	bagel	coffee	bagels	cheese	sandwich	0.1666
	SSNMF	bartender	cheese	tasty	made	server	0.0555
	Whitening	hair	salon	nails	nail	back	0.0678
salon	NMF	hair	absolute	cut	beautiful	salon	-0.0192
	SSNMF	salon	manicure	back	nail	clean	0.0375
	Whitening	mexican	burrito	tacos	salsa	cheese	0.0506
mexican	NMF	mexican	fresh	burrito	tacos	time	0.0389
	SSNMF	exit	mexican	bland	restaurants	world	-0.0720
	Whitening	chicken	chinese	rice	hot	fast	0.0978
chinese	NMF	chicken	chinese	fast	rice	time	0.0717
	SSNMF	chinese	area	type	lot	east	0.0455

Label word	Algo	topword-1	topword-2	topword-3	topword-4	topword-5	PMI
	Whitening	coffee	find	things	tea	starbucks	0.1079
tea	NMF	find	store	things	tea	oil	0.0470
	SSNMF	tea	coffee	starbucks	safeway	ice	0.1787
	Whitening	sushi	roll	happy	rolls	fish	0.0330
sushi	NMF	cooks	fun	hash	browns	reasonable	-0.0441
	SSNMF	2nd	sushi	time	location	amazing	-0.1112
	Whitening	nails	nail	pedicure	salon	time	0.1385
nail	NMF	nails	nail	pedicure	time	salon	0.1316
	SSNMF	nail	nails	grandma	cut	make	0.0658
	Whitening	car	wash	clean	time	job	0.0617
wash	NMF	car	wash	back	time	job	0.0583
	SSNMF	car	wash	feels	clean	time	0.0290
	Whitening	years	business	office	recommend	family	0.0856
insurance	NMF	office	work	walk	time	insurance	0.0189
	SSNMF	insurance	years	business	steve	saved	0.0459
	Whitening	ice	cream	chocolate	cold	wait	0.1739
cream	NMF	ice	cream	school	cone	kids	0.1111
	SSNMF	cream	ice	wait	stone	cold	0.1494
	Whitening	hair	beautiful	absolute	years	salon	0.0749
hair	NMF	hair	absolute	cut	beautiful	salon	0.0507
	SSNMF	beautiful	hair	years	cut	time	0.0532
	Whitening	classes	class	yoga	studio	gym	0.0928
yoga	NMF	yoga	classes	class	studio	time	0.0816
	SSNMF	yoga	practice	dave	feel	amazing	0.0391
	Whitening	tire	tires	oil	car	discount	0.0739
tire	NMF	tire	car	tires	back	time	0.0634
	SSNMF	tire	tires	car	discount	time	0.0274
	Whitening	time	chicken	thai	rice	chinese	-0.0442
vietnamese	NMF	pho	chicken	rice	sauce	back	0.0825
	SSNMF	vietnamese	cake	chinese	back	fresh	-0.0105
	Whitening	donuts	fresh	coffee	donut	chocolate	-0.0349
donuts	NMF	donuts	coffee	donut	store	location	-0.0040
	SSNMF	donuts	donut	chocolate	time	selection	-0.1298
	Whitening	pizza	crust	wings	sauce	cheese	0.0068
crust	NMF	pizza	crust	wings	time	cheese	-0.0503
	SSNMF	min	pizza	crust	hut	pretty	-0.1131
	Whitening	ice	cream	cold	chocolate	flavors	0.1234
ice	NMF	ice	cream	school	cone	kids	0.0718
	SSNMF	ice	cream	wait	stone	cold	0.1312
	Whitening	store	location	big	feel	kids	0.0075
pharmacy	NMF	store	time	location	pharmacy	helpful	0.0049
	SSNMF	pharmacy	customer	clean	safeway	rude	-0.0127
	Whitening	bar	time	beer	wings	drinks	0.0900
beer	NMF	pizza	brick	pretty	bar	box	-0.0190
	SSNMF	beers	beer	operated	hand	locally	0.0817
	Whitening	bike	shop	guys	tires	back	0.0053
bike	NMF	bike	shop	back	bikes	time	0.0525
	SSNMF	bike	time	gun	pretty	store	-0.0293
	Whitening	yogurt	flavors	toppings	frozen	chocolate	0.0659
yogurt	NMF	yogurt	flavors	toppings	frozen	chocolate	0.0420
	SSNMF	yogurt	flavors	back	ice	shop	-0.1370
	Whitening	sushi	chinese	time	fresh	rice	-0.0311
korean	NMF	magazine	market	farmer	farmers	boston	-0.0702

Label word	Algo	topword-1	topword-2	topword-3	topword-4	topword-5	PMI
	SSNMF	korean	chicken	pretty	fried	spicy	0.0376
pizza	Whitening	pizza	crust	wings	time	cheese	0.1491
	NMF	pizza	brick	pretty	bar	box	0.0582
	SSNMF	pizza	ride	brick	long	red	0.0518
coffee	Whitening	coffee	starbucks	donuts	tea	time	0.2728
	NMF	coffee	busy	starbucks	ice	cream	0.2613
	SSNMF	coffee	starbucks	drinks	latte	work	0.0974
sandwich	Whitening	sandwich	subway	sandwiches	bread	time	0.1714
	NMF	sandwich	subway	fresh	bread	location	0.1311
	SSNMF	sandwich	sandwiches	ham	chips	limited	0.0083
pho	Whitening	time	thai	rice	sauce	back	-0.2046
	NMF	pho	chicken	rice	sauce	back	-0.1096
	SSNMF	pho	rice	beef	vietnamese	sauce	-0.0911
gym	Whitening	classes	class	work	gym	yoga	0.1518
	NMF	link	open	isn	working	fast	-0.0304
	SSNMF	gym	fitness	work	open	time	0.1117
park	Whitening	dog	park	dogs	area	kids	0.1099
	NMF	park	dog	time	area	trail	0.1023
	SSNMF	park	dog	dogs	lake	area	0.1303
latte	Whitening	coffee	starbucks	drink	time	make	-0.1617
	NMF	coffee	busy	starbucks	ice	cream	0.0802
	SSNMF	latte	location	work	drink	drinks	-0.0539
trail	Whitening	park	area	phoenix	time	lot	0.1356
	NMF	park	dog	time	area	trail	0.1049
	SSNMF	trail	parking	street	major	easy	0.0267
dentist	Whitening	office	years	dentist	experience	work	0.0734
	NMF	office	dentist	time	work	years	0.1169
	SSNMF	dentist	office	insurance	made	teeth	0.0766
starbucks	Whitening	starbucks	drink	coffee	drinks	times	-0.0972
	NMF	coffee	busy	starbucks	ice	cream	-0.0477
	SSNMF	starbucks	drink	argue	smile	times	-0.1099
taco	Whitening	taco	bell	tacos	fast	sauce	0.0994
	NMF	mexican	fresh	burrito	tacos	time	0.1875
	SSNMF	taco	bell	ghetto	pizza	location	-0.0042
salsa	Whitening	mexican	burrito	tacos	salsa	fresh	0.0887
	NMF	mexican	fresh	burrito	tacos	time	0.0267
	SSNMF	salsa	fresh	tacos	baja	fish	-0.0697
thai	Whitening	thai	rice	chinese	hot	chicken	0.0691
	NMF	thai	chicken	rice	back	sauce	0.1164
	SSNMF	thai	pad	tea	dish	green	0.0275
chocolate	Whitening	yogurt	flavors	chocolate	cream	ice	0.1923
	NMF	gelato	flavors	chocolate	ice	cream	0.1641
	SSNMF	chocolate	caramel	factory	dark	covered	0.1943
bar	Whitening	bar	drinks	night	time	beer	0.0142
	NMF	pizza	brick	pretty	bar	box	-0.0143
	SSNMF	bar	bit	big	seating	beer	-0.0086
noodle	Whitening	chicken	chinese	rice	thai	sauce	0.2423
	NMF	pho	chicken	rice	sauce	back	0.2630
	SSNMF	chicken	noodle	rice	back	sauces	0.0910
burrito	Whitening	burrito	mexican	stars	tacos	salsa	0.1320
	NMF	mexican	fresh	burrito	tacos	time	0.0638
	SSNMF	stars	burrito	green	sauce	mexican	0.0467
salad	Whitening	salad	chicken	fresh	sandwich	bar	0.1780

Label word	Algo	topword-1	topword-2	topword-3	topword-4	topword-5	PMI
	NMF	pizza	brick	pretty	bar	box	-0.0220
	SSNMF	salad	bar	salads	soup	competitors	-0.0123
burger	Whitening	burger	fries	burgers	fast	time	0.1489
	NMF	link	open	isn	working	fast	0.0159
	SSNMF	stale	burger	meat	bite	king	0.0322
hike	Whitening	park	area	time	lot	back	0.0572
	NMF	park	dog	time	area	trail	0.0747
	SSNMF	hike	park	rock	mountain	water	0.1255
pedicure	Whitening	nails	nail	pedicure	job	salon	0.0189
	NMF	nails	nail	pedicure	time	salon	0.0158
	SSNMF	pedicure	job	nail	close	home	-0.0931
fries	Whitening	burger	fries	burgers	fast	cheese	-0.0413
	NMF	cut	wait	time	hair	manager	-0.2616
	SSNMF	fries	grease	dirty	dark	slow	-0.1629
dog	Whitening	dog	dogs	park	pet	hot	0.1501
	NMF	dog	tony	cut	dogs	style	0.0751
	SSNMF	dog	door	tie	made	serve	0.0080
panda	Whitening	chicken	fast	chinese	rice	time	-0.1488
	NMF	chicken	chinese	fast	rice	time	-0.1291
	SSNMF	panda	orange	rice	fried	bad	-0.1327
beans	Whitening	mexican	burrito	chicken	tacos	salsa	-0.0550
	NMF	mexican	fresh	burrito	tacos	time	-0.1419
	SSNMF	trouble	beans	rice	chicken	marinated	-0.1233
subway	Whitening	subway	sandwich	clean	fresh	location	-0.0074
	NMF	sandwich	subway	fresh	bread	location	-0.0445
	SSNMF	subway	location	clean	super	sandwich	-0.0524
car	Whitening	car	wash	back	time	work	0.1064
	NMF	car	wash	back	time	job	0.0874
	SSNMF	visited	car	back	job	weeks	0.0353
cake	Whitening	found	cake	chocolate	shop	yogurt	0.0754
	NMF	back	time	shop	cake	found	0.0099
	SSNMF	cake	wanted	wedding	flavor	perfect	0.0416
steak	Whitening	location	fast	makes	feel	quality	-0.0672
	NMF	prices	selection	quality	family	helpful	-0.1569
	SSNMF	difference	fast	steak	sandwiches	subs	-0.1672
curry	Whitening	thai	chicken	rice	chinese	hot	0.1482
	NMF	thai	chicken	rice	back	sauce	0.1903
	SSNMF	chicken	stew	brown	curry	rice	0.0047
massage	Whitening	massage	back	amazing	years	spa	0.1359
	NMF	massage	time	back	amazing	hour	-0.0035
	SSNMF	massage	arts	experience	amazing	hour	-0.0168
italian	Whitening	sandwich	pizza	time	back	bread	-0.0254
	NMF	gelato	flavors	chocolate	ice	cream	0.0241
	SSNMF	ice	italian	flavors	cream	chocolate	-0.0231

Appendix C. Computation of A, B for Different Models

This section outlines the construction of matrices A, B in various models via different moment computations. First we introduce some notations which we use in Appendices C, D, E, F, and G.

C.1 Notations

For a vector x, $\|x\|$ denotes its ℓ_2 norm. For a matrix X, $\|X\|$ represents the spectral norm of the matrix. We use the notation $\widehat{X}$ or $\widehat{\mathbb{E}}[X]$ to represent the sample estimate of a quantity X, unless mentioned otherwise. For a matrix M let $\sigma_k(M)$ denote the $k-$th largest singular value of M, and $\tilde{\sigma}_k(M)$ denote the $k-$th largest eigenvalue. n represents the number of samples used to obtain the sample estimates. Next, we introduce some basic tensor notations. Let $x, y, z \in \mathbb{R}^d$ be three d dimensional vectors. Then the order-3 tensor $T_3 = x \otimes y \otimes z$ is defined as $T_3(i,j,k) = x(i)y(j)z(k)$, for $i,j,k \in [d]$. Similarly the order-2 tensor $T_2 = x \otimes y$ is equivalent to the matrix outer product $T_2 = xy^T$. Finally let $v \in \mathbb{R}^d$ be another d dimensional vector, I be the d dimensional identity matrix. The tensor contraction $T_3(I,I,v)$ is equal to the order-2 tensor $T_3(I,I,v) = \langle z, v\rangle x \otimes y$, which is again equivalent to the matrix $T_3(I,I,v) = \langle z, v\rangle xy^T$. For order-2 tensors we will use the tensor and matrix notations interchangeably.

C.2 GMM Moments

In this section we prove how the required matrices A, B can be computed in the GMM model. We restate the following useful theorem from Hsu and Kakade (2013) which computes three tensor moments for the GMM model.

Theorem 7 (Hsu and Kakade (2013)) *Consider the GMM model with means $\{\mu_1, \ldots, \mu_k\}$ and corresponding variances $\{\sigma_1^2, \ldots, \sigma_k^2\}$, and α_i denote the proportion of the i-th component in the mixture. Let $\sigma^2 = \sum_{i=1}^k \alpha_i \sigma_i^2$ be the smallest eigenvalue of the covariance matrix $\mathbb{E}[(x - \mathbb{E}[x])(x - \mathbb{E}[x])^T]$ (note that since $\sum \alpha_i \mu_i \mu_i^T$ has rank k, this is the same as the $k+1$th-largest eigenvalue), and u be a unit norm eigenvector corresponding to the eigenvalue σ^2. Define*

$$\begin{aligned} \widetilde{m} &= \mathbb{E}[x(u^T(x - \mathbb{E}[x]))^2], \quad M_2 = \mathbb{E}[x \otimes x] - \sigma^2 I \\ M_3 &= \mathbb{E}[x \otimes x \otimes x] - \sum_{i=1}^d (\widetilde{m} \otimes e_i \otimes e_i + e_i \otimes \widetilde{m} \otimes e_i + e_i \otimes e_i \otimes \widetilde{m}) \end{aligned}$$

where $\{e_1, \ldots, e_d\}$ form standard basis of $\mathbb{R}^d$. Then,

$$\widetilde{m} = \sum_{i=1}^k \alpha_i \sigma_i^2 \mu_i, \quad M_2 = \sum_{i=1}^k \alpha_i \mu_i \otimes \mu_i, \quad M_3 = \sum_{i=1}^k \alpha_i \mu_i \otimes \mu_i \otimes \mu_i.$$

Theorem 8 *In the GMM model define*

$$\begin{aligned} m &= \mathbb{E}[x], \quad A = \mathbb{E}[xx^T] - \sigma^2 I_d \\ B &= \mathbb{E}[\langle x, v\rangle xx^T] - \widetilde{m}v^T - v\widetilde{m}^T - \langle \widetilde{m}, v\rangle I_d \end{aligned}$$

Then, $m = \sum_i \alpha_i \mu_i$, $A = \sum_{i=1}^k \alpha_i \mu_i \mu_i^T$ and $B = \sum_{i=1}^k \alpha_i \langle \mu_i, v\rangle \mu_i \mu_i^T$

Proof The expression for m, A follows directly from Theorem 7 by noting that $A = M_2$ and $\mu_i \otimes \mu_i = \mu_i \mu_i^T$. To compute B consider the tensor contraction $M_3(I,I,v)$, M_3 as in Theorem 7. Then,

$$\begin{aligned} M_3(I,I,v) &= \mathbb{E}[\langle x, v\rangle x \otimes x] - \sum_{i=1}^d (v(i)\widetilde{m} \otimes e_i + v(i) e_i \otimes \widetilde{m} + \langle \widetilde{m}, v\rangle e_i \otimes e_i) \\ &= \mathbb{E}[\langle x, v\rangle xx^T] - \sum_{i=1}^d (v(i)\widetilde{m} e_i^T + v(i) e_i \widetilde{m}^T + \langle \widetilde{m}, v\rangle e_i e_i^T) \\ &= \mathbb{E}[\langle x, v\rangle xx^T] - \widetilde{m}v^T - v\widetilde{m}^T - \langle \widetilde{m}, v\rangle I_d = B \end{aligned}$$

Also from Theorem 7, $M_3(I,I,v) = \sum_{i=1}^k \alpha_i \langle \mu_i, v\rangle \mu_i \otimes \mu_i = \sum_{i=1}^k \alpha_i \langle \mu_i, v\rangle \mu_i \mu_i^T$. Therefore $B = \sum_{i=1}^k \alpha_i \langle \mu_i, v\rangle \mu_i \mu_i^T$. ■

C.3 LDA Moments

In this section we show the m, A, B computation corresponding to the LDA model. Again we restate the following theorem from Anandkumar et al. (2014) which computes the first three tensor moments for LDA distribution.

Theorem 9 (Anandkumar et al. (2014)) *In an LDA model with parameters $\bar{\alpha} = (\alpha_1, \ldots, \alpha_k)$, topic distributions $\mu_1, \ldots, \mu_k$. Let $\alpha_0 = \sum_{i=1}^k \alpha_i$. Define*

$$\begin{aligned} M_1 &= \mathbb{E}[x_1], \quad M_2 = \mathbb{E}[x_1 \otimes x_2] - \frac{\alpha_0}{1+\alpha_0} M_1 \otimes M_1 \\ M_3 &= \mathbb{E}[x_1 \otimes x_2 \otimes x_3] - \frac{\alpha_0}{\alpha_0 + 2} (\mathbb{E}[x_1 \otimes x_2 \otimes M_1] + \mathbb{E}[x_1 \otimes M_1 \otimes x_3] + \mathbb{E}[M_1 \otimes x_2 \otimes x_3]) \\ &\quad + \frac{2\alpha_0^2}{(\alpha_0+1)(\alpha_0+2)} M_1 \otimes M_1 \otimes M_1 \end{aligned}$$

Then,

$$\begin{aligned} M_1 &= \sum_{i=1}^k \frac{\alpha_i}{\alpha_0} \mu_i, \quad M_2 = \sum_{i=1}^k \frac{\alpha_i}{\alpha_0(\alpha_0+1)} \mu_i \otimes \mu_i \\ M_3 &= \sum_{i=1}^k \frac{2\alpha_i}{\alpha_0(\alpha_0+1)(\alpha_0+2)} \mu_i \otimes \mu_i \otimes \mu_i \end{aligned}$$

Theorem 10 *For an LDA model for any $v \in \mathbb{R}^d$ suppose m, A, B be defined as*

$$\begin{aligned} m &= \alpha_0 \mathbb{E}[x_1] \\ A &= \alpha_0(\alpha_0+1)\mathbb{E}[x_1 x_2^T] - mm^T \\ B &= \frac{\alpha_0(\alpha_0+1)(\alpha_0+2)}{2} \mathbb{E}[\langle x_3, v\rangle x_1 x_2^T] - \frac{\alpha_0(\alpha_0+1)}{2} \left(\langle m, v\rangle \mathbb{E}[x_1 x_2^T] + \mathbb{E}[\langle x_3, v\rangle x_1 m^T] \right. \\ &\quad \left. + \mathbb{E}[\langle x_3, v\rangle m x_2^T]\right) + \langle m, v\rangle mm^T. \end{aligned}$$

Then we can express m, A, B as follows.

$$m = \sum_{i=1}^{k} \alpha_i \mu_i, \quad A = \sum_{i=1}^{k} \alpha_i \mu_i \mu_i^T, \quad B = \sum_{i=1}^{k} \alpha_i \langle \mu_i, v \rangle \mu_i \mu_i^T$$

Proof The expressions for m and A follows easily from Theorem 9 since $m = \alpha_0 M_1$ and $A = \alpha_0(\alpha_0+1)M_2$. To show the expression for B consider the tensor contraction $M_3(I, I, v)$, M_3 defined as in Theorem 9. Then we have

$$\begin{aligned} M_3(I,I,v) &= \mathbb{E}[\langle x_3, v\rangle x_1 \otimes x_2] - \frac{\alpha_0}{\alpha_0+2}\left(\mathbb{E}[\langle M_1, v\rangle x_1 \otimes x_2] + \mathbb{E}[\langle x_3, v\rangle x_1 \otimes M_1]\right. \\ &\quad \left. + \mathbb{E}[\langle x_3, v\rangle M_1 \otimes x_2 \otimes x_3]\right) + \frac{2\alpha_0^2}{(\alpha_0+1)(\alpha_0+2)}\langle M_1, v\rangle \otimes M_1 \otimes M_1 \\ &= \frac{2}{\alpha_0(\alpha_0+1)(\alpha_0+2)} B \end{aligned}$$

where we used $x_1 \otimes x_2$ is same as $x_1 x_2^T$ and so on. We also get from Theorem 9 $M_3(I, I, v) = \sum_{i=1}^{k} \frac{2\alpha_i}{\alpha_0(\alpha_0+1)(\alpha_0+2)} \langle \mu_i, v\rangle \mu_i \otimes \mu_i$. Therefore we have

$$B = \frac{\alpha_0(\alpha_0+1)(\alpha_0+2)}{2} M_3(I,I,v) = \sum_{i=1}^{k} \alpha_i \langle \mu_i, v\rangle \mu_i \mu_i^T.$$

■

C.4 Mixed Regression Moments

Recall in mixed regression we have $y = \langle x, \mu_i\rangle + \xi$ where $x \sim \mathcal{N}(0, I)$ and $\xi \sim \mathcal{N}(0, \sigma^2)$. In the following Lemmas we compute the various moments $M_{1,1}, M_{2,2}, M_{3,1}, M_{3,3}$ and show how they are used to compute m, A, B.

Lemma 11 *In mixed linear regression define $M_{1,1} = \mathbb{E}[yx]$, $M_{2,2} = \mathbb{E}[y^2 xx^T]$, $M_{3,1} = \mathbb{E}[y^3 x]$ and $M_{3,3} = \mathbb{E}[y^3 \langle x, v\rangle xx^T]$. Then,*

$$\begin{aligned} M_{1,1} &= \sum_{i=1}^{k} \alpha_i \mu_i \\ M_{2,2} &= 2\sum_{i=1}^{k} \alpha_i \mu_i \mu_i^T + (\sigma^2 + \sum_{i=1}^{k} \alpha_i \|\mu_i\|^2) I \\ M_{3,1} &= 3\sum_{i=1}^{k} \alpha_i (\sigma^2 + \|\mu_i\|^2)\mu_i \\ M_{3,3} &= 6\sum_{i=1}^{k} \alpha_i \langle \mu_i, v\rangle \mu_i \mu_i^T + \left(M_{3,1} v^T + v M_{3,1}^T + \langle M_{3,1}, v\rangle I\right) \end{aligned}$$

Proof

We compute the moments as shown below.

$$M_{1,1} = \mathbb{E}[yx] = \sum_{i=1}^{k} \alpha_i \mathbb{E}[x^T \mu_i x + \xi x] = \sum_{i=1}^{k} \alpha_i \mu_i$$

$$\begin{aligned} M_{2,2} &= \mathbb{E}[y^2 xx^T] = \sum_{i=1}^{k} \alpha_i \mathbb{E}[\langle \mu_i, x\rangle^2 xx^T] + \mathbb{E}[\xi^2]\mathbb{E}[xx^T] \\ &= \sum_{i=1}^{k} \alpha_i \mathbb{E}[\langle \mu_i, x\rangle^2 xx^T] + \sigma^2 I \\ &= \sum_{i=1}^{k} \alpha_i (2\mu_i \mu_i^T + \|\mu_i\|^2 I) + \sigma^2 I \\ &= 2\sum_{i=1}^{k} \alpha_i \mu_i \mu_i^T + \sum_{i=1}^{k} \alpha_i (\sigma^2 + \|\mu_i\|^2) I \end{aligned}$$

Using the fact that all odd moments of normal random variable are zero.

$$\begin{aligned} M_{3,1} &= \mathbb{E}[y^3 x] = \sum_{i=1}^{k} \alpha_i \mathbb{E}[(\langle x, \mu_i\rangle + \xi)^3 x] \\ &= \sum_{i=1}^{k} \alpha_i \mathbb{E}[\langle x, \mu_i\rangle^3 x] + 3\sum_{i=1}^{k} \alpha_i \mathbb{E}[\xi^2]\mathbb{E}[\langle x, \mu_i\rangle x] \\ &= 3\sum_{i=1}^{k} \alpha_i \|\mu_i\|^2 \mu_i + 3\sum_{i=1}^{k} \alpha_i \sigma^2 \mu_i = 3\sum_{i=1}^{k} \alpha_i (\sigma^2 + \|\mu_i\|^2)\mu_i \end{aligned}$$

We use the fact that for even p the moment $\mathbb{E}[z^p] = (p-1)!!$ for a standard normal random variable z and !! denote the double factorial. Next we compute $M_{3,3}$.

$$\begin{aligned} M_{3,3} &= \mathbb{E}[y^3 \langle x, v\rangle xx^T] = \sum_{i=1}^{k} \alpha_i \mathbb{E}[(\langle x, \mu_i\rangle + \xi)^3 \langle x, v\rangle xx^T] \\ &= \sum_{i=1}^{k} \alpha_i \mathbb{E}[\langle x, \mu_i\rangle^3 \langle x, v\rangle xx^T] + 3\sum_{i=1}^{k} \alpha_i \mathbb{E}[\xi^2]\mathbb{E}[\langle x, v\rangle \langle x, \mu_i\rangle xx^T] \\ &= \sum_{i=1}^{k} \alpha_i \mathbb{E}[\langle x, \mu_i\rangle^3 \langle x, v\rangle xx^T] + 3\sigma^2 \sum_{i=1}^{k} \alpha_i \mathbb{E}[\langle x, v\rangle \langle x, \mu_i\rangle xx^T] \end{aligned} \tag{5}$$

Now we compute these individual moments.

$$\mathbb{E}[\langle x, v\rangle \langle x, \mu_i\rangle xx^T] = \mu_i^T v + v\mu_i^T + \langle \mu_i, v\rangle I$$

Using the fact that any odd combination of the variables in x will be zero in expectation. Also,

$$\mathbb{E}[\langle x,\mu_i\rangle^3\langle x,v\rangle xx^T] = 6\langle v,\mu_i\rangle\mu_i\mu_i^T + 3\|\mu_i\|^2[\mu_i^Tv + v\mu_i^T + \langle\mu_i,v\rangle I]$$

Again by using the moments of standard normal variable. This can be verified by considering the (a,b)-th entry of the matrix on the right as a polynomial in $\mu_i(l)$, the l-th component of μ_i, and matching the corresponding coefficients from both sides of the equation.

Combining with equation (5) we get,

$$\begin{aligned}
M_{3,3} &= \sum_{i=1}^k \alpha_i\left[6\langle v,\mu_i\rangle\mu_i\mu_i^T + 3\|\mu_i\|^2(\mu_i^Tv + v\mu_i^T + \langle\mu_i,v\rangle I)\right] \\
&\quad +3\sigma^2\sum_{i=1}^k\alpha_i[\mu_i^Tv + v\mu_i^T + \langle\mu_i,v\rangle I] \\
&= 6\sum_{i=1}^k\alpha_i\langle v,\mu_i\rangle\mu_i\mu_i^T + 3\sum_{i=1}^k\alpha_i(\sigma^2+\|\mu_i\|^2)[\mu_i^Tv + v\mu_i^T + \langle\mu_i,v\rangle I] \\
&= 6\sum_{i=1}^k\alpha_i\langle v,\mu_i\rangle\mu_i\mu_i^T + \left(M_{3,1}v^T + vM_{3,1}^T + \langle M_{3,1},v\rangle I\right)
\end{aligned}$$

■

Theorem 12 *Let m, A, B be defined as*

$$\begin{aligned}
m &= M_{1,1}, \quad A = \frac{1}{2}(M_{2,2} - \tau^2 I), \\
B &= \frac{1}{6}(M_{3,3} - (M_{3,1}v^T + vM_{3,1}^T + \langle M_{3,1},v\rangle I))
\end{aligned}$$

where τ^2 is the smallest singular value of $M_{2,2}$. Then,

$$m = \sum_{i=1}^k\alpha_i\mu_i, \quad A = \sum_{i=1}^k\alpha_i\mu_i\mu_i^T, \quad B = \sum_{i=1}^k\alpha_i\langle\mu_i,v\rangle\mu_i\mu_i^T$$

Proof The proof follows directly from Lemma 11. Note that since μ_i-s are linearly independent the smallest singular vector τ^2 of $M_{2,2}$ is equal to $\sum_{i=1}^k\alpha_i(\sigma^2+\|\mu_i\|^2)$. Then $A = \frac{1}{2}\left(M_{2,2} - \tau^2 I\right) = \sum_{i=1}^k\alpha_i\mu_i\mu_i^T$. Similarly the expression for B holds. ■

C.5 Subspace Clustering Moments

In this section we derive the necessary moments required for subspace clustering. Recall that in the subspace clustering model we have k dimension—m subspaces $U_1,\ldots,U_k \in \mathbb{R}^{d\times m}$ (matrices $U_1,\ldots,U_k$ have orthonormal columns). The data is generated as follows. We sample $y\sim\mathcal{N}(0,I_d)$ and set $x = U_iU_i^Ty + \xi$, where $\xi\sim\mathcal{N}(0,\sigma^2I_d)$ is additive noise.

Theorem 13 *Consider the subspace clustering model. Let M_2, A, B be defined as,*

$$\begin{aligned}
M_2 &:= \mathbb{E}[xx^T], \quad A := M_2 - \sigma^2I_d \\
B &:= \mathbb{E}[\langle x,v\rangle^2xx^T] - \sigma^2(v^TAv)I_d - \sigma^2\|v\|^2A - \sigma^4(\|v\|^2I_d + vv^T) - 2\sigma^2(Avv^T + vv^TA)
\end{aligned}$$

where $\sigma^2 = \sigma_{mk+1}(M_2)$. Then,

$$\begin{aligned}
A &= \sum_{i=1}^k\alpha_iU_iU_i^T \\
B &= \sum_{i=1}^k\alpha_i\|U_i^Tv\|^2U_iU_i^T + 2\sum_{i=1}^k\alpha_iU_iU_i^Tvv^TU_iU_i^T
\end{aligned}$$

Proof First we compute M_2.

$$M_2 = \mathbb{E}(xx^T) = \sum_{i=1}^k\alpha_i\mathbb{E}\left[U_iU_i^Tyy^TU_iU_i^T\right] + \mathbb{E}[\xi\xi^T] = \sum_{i=1}^k\alpha_iU_iU_i^T + \sigma^2I_d$$

Using $\mathbb{E}[yy^T] = I$ as $y\sim\mathcal{N}(0,I)$ and $U_i^TU_i = I$ since the columns are orthogonal. Since $\alpha_i > 0$, the $mk+1$-th singular value of M_2, $\sigma_{mk+1}(M_2) = \sigma^2$. Therefore it follows that,

$$A = M_2 - \sigma^2I_d = \sum_{i=1}^k\alpha_iU_iU_i^T$$

Now we compute the moment $\mathbb{E}[\langle x,v\rangle^2xx^T]$. Given a sample $x = U_iU_i^Ty + \xi$ from the i-th subspace we have,

$$\begin{aligned}
\langle x,v\rangle^2 &= v^TU_iU_i^Tyy^TU_iU_i^Tv + v^T\xi\xi^Tv + 2v^T\xi v^TU_iU_i^Ty \\
xx^T &= U_iU_i^Tyy^TU_iU_i^T + U_iU_i^Ty\xi^T + \xi y^TU_iU_i^T + \xi\xi^T
\end{aligned}$$

Then we can write,

$$\begin{aligned}
&\mathbb{E}[\langle x,v\rangle^2xx^T] \\
&= \sum_{i=1}^k\alpha_i\left(\mathbb{E}[v^TU_iU_i^Tyy^TU_iU_i^TvU_iU_i^Tyy^TU_iU_i^T] + \mathbb{E}[v^TU_iU_i^Tyy^TU_iU_i^Tv]\mathbb{E}[\xi\xi^T]\right. \\
&\quad +\mathbb{E}[v^T\xi\xi^Tv]\mathbb{E}[U_iU_i^Tyy^TU_iU_i^T] + \mathbb{E}[v^T\xi\xi^Tv\xi\xi^T] + 2\mathbb{E}[(v^T\xi v^TU_iU_i^Ty)U_iU_i^Ty\xi^T] \\
&\quad \left.+2\mathbb{E}[(v^T\xi v^TU_iU_i^Ty)\xi y^TU_iU_i^T]\right) \\
&= T_1 + T_2 + T_3 + T_4 + T_5 + T_6 \qquad (6)
\end{aligned}$$

where $T_1, \ldots, T_6$ are as follows. We define $v_i := U_i U_i^T v$, we use the Gaussian moment results $\mathbb{E}[\langle v, z\rangle z] = \sigma^2 v$, and $\mathbb{E}[\langle v, z\rangle^2 z z^T] = \sigma^4(\|v\|^2 I_d + vv^T)$ whenever $z \sim \mathcal{N}(0, \sigma^2 I_d)$.

$$\begin{aligned}
T_1 &= \sum_{i=1}^k \alpha_i \mathbb{E}\left[v^T U_i U_i^T y y^T U_i U_i^T v U_i U_i^T y y^T U_i U_i^T\right] \\
&= \sum_{i=1}^k \alpha_i \mathbb{E}[\langle y, v_i\rangle^2 U_i U_i^T y y^T U_i U_i^T] = \sum_{i=1}^k \alpha_i U_i U_i^T \mathbb{E}[\langle y, v_i\rangle^2 y y^T] U_i U_i^T \\
&= \sum_{i=1}^k \alpha_i U_i U_i^T (\|v_i\|^2 I_d + 2 v_i v_i^T) U_i U_i^T \\
&= \sum_{i=1}^n \alpha_i \|v_i\|^2 U_i U_i^T + 2\sum_{i=1}^k \alpha_i U_i U_i^T v v^T U_i U_i^T \\
&= \sum_{i=1}^k \alpha_i \|U_i^T v\|^2 U_i U_i^T + 2\sum_{i=1}^k \alpha_i U_i U_i^T v v^T U_i U_i^T
\end{aligned}$$

since $\|v_i\| = \|U_i U_i^T v\| = \|U_i^T v\|$.

$$\begin{aligned}
T_2 &= \sum_{i=1}^k \alpha_i \mathbb{E}[v^T U_i U_i^T y y^T U_i U_i^T v]\mathbb{E}[\xi\xi^T] = \sum_{i=1}^k \alpha_i v^T U_i U_i^T v \times \sigma^2 I_d = \sigma^2 (v^T A v) I_d \\
T_3 &= \sum_{i=1}^k \alpha_i \mathbb{E}[v^T \xi\xi^T v]\mathbb{E}[U_i U_i^T y y^T U_i U_i^T] = \sigma^2 \|v\|^2 \sum_{i=1}^k \alpha_i U_i U_i^T = \sigma^2 \|v\|^2 A \\
T_4 &= \sum_{i=1}^k \alpha_i \mathbb{E}[v^T \xi\xi^T v \xi\xi^T] = \mathbb{E}[\langle v, \xi\rangle^2 \xi\xi^T] = \sigma^4(\|v\|^2 I_d + 2vv^T) \\
T_5 &= \sum_{i=1}^k \alpha_i 2\mathbb{E}[(v^T \xi v^T U_i U_i^T y) U_i U_i^T y \xi^T] = 2\sum_{i=1}^k \alpha_i \mathbb{E}[(v^T U_i U_i^T y) U_i U_i^T y]\mathbb{E}[\langle v, \xi\rangle \xi^T] \\
&= 2\sum_{i=1}^k \alpha_i \mathbb{E}[(v^T U_i U_i^T y) U_i U_i^T y] \times \sigma^2 v^T = 2\sigma^2 \sum_{i=1}^k \alpha_i \mathbb{E}[(v^T U_i U_i^T y) U_i U_i^T y v^T] \\
&= 2\sigma^2 \sum_{i=1}^k \alpha_i \mathbb{E}[U_i U_i^T \langle v, y\rangle y v^T] = 2\sigma^2 \sum_{i=1}^k \alpha_i U_i U_i^T v v^T = 2\sigma^2 A v v^T \\
T_6 &= 2\sum_{i=1}^k \alpha_i \mathbb{E}[(v^T \xi v^T U_i U_i^T y)\xi y^T U_i U_i^T] = 2\sum_{i=1}^k \alpha_i \mathbb{E}[\langle v, \xi\rangle \xi]\mathbb{E}[\langle v_i, y\rangle y^T U_i U_i^T] \\
&= 2\sigma^2 \sum_{i=1}^k \alpha_i v v_i^T U_i U_i^T = \sigma^2 \sum_{i=1}^k \alpha_i v v^T U_i U_i^T = \sigma^2 v v^T \sum_{i=1}^k \alpha_i U_i U_i^T = 2\sigma^2 v v^T A
\end{aligned}$$

Therefore,

$$\begin{aligned}
B &= \mathbb{E}[\langle x, v\rangle^2 x x^T] - \sigma^2 (v^T A v) I_d - \sigma^2 \|v\|^2 A - \sigma^4(\|v\|^2 I_d + vv^T) - 2\sigma^2(Avv^T + vv^T A) \\
&= \mathbb{E}[\langle x, v\rangle^2 x x^T] - T_2 - T_3 - T_4 - T_5 - T_6 = T_1 \\
&= \sum_{i=1}^k \alpha_i \|U_i^T v\|^2 U_i U_i^T + 2\sum_{i=1}^k \alpha_i U_i U_i^T v v^T U_i U_i^T
\end{aligned}$$

■

Appendix D. Finite-sample Analysis of the Whitening Method

Suppose that

$$\begin{aligned}
A &= \sum_i \alpha_i \mu_i \mu_i^T \\
B &= \sum_i \beta_i \mu_i \mu_i^T \\
\|A - \hat{A}\| &\le \epsilon \\
\|B - \hat{B}\| &\le \epsilon,
\end{aligned}$$

where σ_k is the kth singular value of A. Let V be the $n \times k$ matrix whose columns are the first k singular vectors of A, and let $\hat{V}$ be the same for $\hat{A}$. Let D be the diagonal matrix of singular values of A, and let $\hat{D}$ be the diagonal matrix of the first k singular values of $\hat{A}$. Then $A = VDV^T$ and $V^T V = \hat{V}^T \hat{V} = I_k$. This entire section is under the assumptions of Theorem 1; in particular, recall that $\epsilon \le \sigma_k(A)/4$.

It will be technically convenient for us to assume that $\|B\| \le \|A\| = \sigma_1(A)$. This assumption holds without loss of generality: if not, simply rescale the side information, setting $v^{\text{new}} = v\frac{\|A\|}{\|B\|}$. This has the effect of rescaling B, so that $\|B^{\text{new}}\| = \|A\|$; define also $\hat{B}^{\text{new}} = \hat{B}\frac{\|A\|}{\|B\|}$. Note that

$$\|B^{\text{new}} - \hat{B}^{\text{new}}\| = \|B - \hat{B}\|\frac{\|A\|}{\|B\|} \le \epsilon$$

under the assumption $\|B - \hat{B}\| \le \epsilon$. Now, the algorithm is homogeneous in $\hat{B}$: it will produce the same output given either $\hat{B}$ or $\hat{B}^{\text{new}}$; hence, it suffices to prove Theorem 1 with v, B, and $\hat{B}$ replaced by their new versions. Since the new versions satisfy $\|B^{\text{new}}\| \le \|A\|$, we may assume this without loss of generality. From now on, we will drop the notation B^{new}, and we will simply prove Theorem 1 under the assumption $\|B\| \le \|A\|$.

Our basic tool is Wedin's theorem:

Theorem 14 *For a matrix A, let $P^A_{\ge s}$ be the orthogonal projection onto the subspace spanned by singular vectors of A with singular value at least s. Let $P^A_{\le s}$ be the orthogonal projection onto the subspace spanned by singular vectors with singular value at most s. Then for any matrices A and B, and for any $s < t$,*

$$\|P^A_{\le s} P^B_{\ge t}\| \le \frac{2\|A - B\|}{t - s}.$$

In applying Wedin's theorem, the following geometric lemma will be useful. In what follows, P_E denotes the orthogonal projection onto E.

Lemma 15 *Let E and F be subspaces of $\mathbb{R}^n$ with $\|P_{E^\perp} P_F\| \leq \delta$. Then $\|P_F v\|^2 \leq \|P_E v\|^2 + 3\delta\|v\|^2$ for every $v \in \mathbb{R}^n$.*

Lemma 16 *If $\epsilon < \sigma_k/4$ then for any $u \in \mathbb{R}^k$,*

$$\sqrt{1 - \frac{16\epsilon^2}{\sigma_k^2}}\|u\| \leq \|\hat{V}^T V u\| \leq \|u\|.$$

By a simple change of variables, if we define

$$O = D^{-1/2}\hat{V}^T V D^{1/2}$$

then O is also an almost-isometry: for every $u \in \mathbb{R}^k$,

$$\sqrt{1 - \frac{16\epsilon^2}{\sigma_k^2}}\|u\| \leq \|Ou\| \leq \|u\|. \tag{7}$$

Proof First, note that $\sigma_k(\hat{A}) \geq \sigma_k(A) - \|A - \hat{A}\| \geq \sigma_k - \epsilon$. If $\epsilon < \sigma_k/4$, we also have $\sigma_{k+1}(\hat{A}) \leq \sigma_{k+1}(A) + \epsilon \leq \sigma_k/4 < \sigma_k - \epsilon$, which implies that $\hat{V}\hat{V}^T = P^{\hat{A}}_{\geq \sigma_k - \epsilon}$.

Let $\hat{W}$ be a $d \times (d-k)$ matrix whose columns form an orthonormal basis for the orthogonal complement of the column span of $\hat{V}$. Note that if $\epsilon < \sigma_k/2$ then the kth singular value of $\hat{A}$ is strictly larger than $\sigma_k/2$ and the $(k+1)$th singular value is at most ϵ. Then $P^{\hat{A}}_{\leq \epsilon} = \hat{W}\hat{W}^T$. By Wedin's theorem,

$$\|\hat{W}\hat{W}^T V V^T\| = \|P^{\hat{A}}_{\leq \epsilon} P^A_{\geq \sigma_k}\| \leq \frac{2\epsilon}{\sigma_k - \epsilon} \leq \frac{4\epsilon}{\sigma_k}$$

Now, $\hat{W}^T$ and V have norm 1, and so it follows that

$$\|\hat{W}^T V\| = \|\hat{W}^T(\hat{W}\hat{W}^T V V^T)V\| \leq \frac{4\epsilon}{\sigma_k}.$$

For any $u \in \mathbb{R}^k$ with $\|u\| = 1$, we have

$$\|\hat{V}^T V u\|^2 = 1 - \|\hat{W}^T V u\|^2 \geq 1 - 16\epsilon^2/\sigma_k^2,$$

from which the claimed lower bound follows. On the other hand, $\|\hat{V}^T V u\| \leq u$ because both $\hat{V}^T$ and V have norm 1. ■

Let $M = D^{-1/2}V^T B V D^{-1/2}$ and $\hat{M} = \hat{D}^{-1/2}\hat{V}^T\hat{B}\hat{V}\hat{D}^{-1/2}$. Then M is the infinite-sample version of A's whitening matrix applied to B, and $\hat{M}$ is the finite-sample analogue. Recall from (7) that $O = D^{-1/2}\hat{V}^T V D^{1/2}$ is an almost-isometry of $\mathbb{R}^k$.

Lemma 17

$$\|OMO^T - \hat{M}\| \leq C\frac{\epsilon\sigma_1}{\sigma_k^2}.$$

Proof The first step is to approximate OMO^T by $D^{-1/2}\hat{V}^T B\hat{V}D^{-1/2}$. To this end, note that

$$OMO^T = D^{-1/2}\hat{V}^T V V^T B V V^T \hat{V} D^{-1/2}.$$

Now, $\hat{V}$ is an isometry of $\mathbb{R}^k$ into $\mathbb{R}^n$; hence,

$$\|\hat{V}^T V V^T - \hat{V}^T\| = \|\hat{V}\hat{V}^T V V^T - \hat{V}\hat{V}^T\| = \|P^{\hat{A}}_{\geq \sigma_k - \epsilon} P^A_{\geq \sigma_k} - P^{\hat{A}}_{\geq \sigma_k - \epsilon}\| = \|P^{\hat{A}}_{\geq \sigma_k - \epsilon} P^A_{\leq 0}\|,$$

where the last equality used the fact that A has rank exactly k, and hence $I - P^A_{\geq \sigma_k} = P^A_{\leq 0}$. Now, Wedin's theorem applied to the computation above implies that

$$\|\hat{V}^T V V^T - \hat{V}^T\| \leq \frac{2\epsilon}{\sigma_k - \epsilon} \leq \frac{4\epsilon}{\sigma_k}$$

(recalling that $\epsilon \leq \sigma_k/4$).

Now, for general matrices $X, Y, \tilde{Y}, Z$ we have

$$\begin{aligned}\|X^T Y^T Z Y X - X^T \tilde{Y}^T Z \tilde{Y} X\| &\leq \|X^T(Y - \tilde{Y})^T Z Y X\| + \|X^T \tilde{Y}^T Z (Y - \tilde{Y}) X\| \\ &\leq \|Y - \tilde{Y}\|\|X\|^2\|Z\|(\|Y\| + \|\tilde{Y}\|).\end{aligned}$$

We apply this with $X = D^{-1/2}$, $Y = \hat{V}$, $\tilde{Y} = \hat{V}VV^T$, and $Z = B$; since $\|D^{-1/2}\| = \sigma_k^{-1/2}$, $\|B\| \leq \sigma_1$, and $\|\hat{V}\|, \|V\|, \|V^T\| = 1$,

$$\|OMO^T - D^{-1/2}\hat{V}^T B\hat{V}D^{-1/2}\| \leq \frac{8\epsilon\sigma_1}{\sigma_k^2}$$

Next, we will replace B by $\hat{B}$ in the above inequality. Since $\|\hat{V}\| = \|\hat{V}^T\| = 1$ and $\|D^{-1/2}\| = \sigma_k^{-1/2}$,

$$\begin{aligned}\|D^{-1/2}\hat{V}^T B\hat{V}D^{-1/2} - D^{-1/2}\hat{V}^T\hat{B}\hat{V}D^{-1/2}\| &= \|D^{-1/2}\hat{V}^T(B - \hat{B})\hat{V}D^{-1/2}\| \\ &\leq \sigma_k^{-1}\|B - \hat{B}\| \leq \frac{\epsilon}{\sigma_k}.\end{aligned}$$

Putting this together with the previous bound yields

$$\|OMO^T - D^{-1/2}\hat{V}^T\hat{B}\hat{V}D^{-1/2}\| \leq \frac{\epsilon}{\sigma_k} + \frac{8\epsilon\sigma_1}{\sigma_k^2} \tag{8}$$

It remains to relate $D^{-1/2}\hat{V}^T\hat{B}\hat{V}D^{-1/2}$ to $\hat{M}$ (which is the same, but with $\hat{D}$ instead of D). Now, Weyl's inequality implies that

$$\|D^{-1/2} - \hat{D}^{-1/2}\| \leq \sigma_k^{-1/2} - (\sigma_k - \epsilon)^{-1/2} \leq \epsilon\sigma_k^{-3/2},$$

where the second inequality follows from a first-order Taylor expansion and the fact that $\epsilon \leq \sigma_k/2$. Hence,

$$\begin{aligned}\|D^{-1/2}\hat{V}^T\hat{B}\hat{V}D^{-1/2} - \hat{M}\| &\leq \|D^{-1/2} - \hat{D}^{-1/2}\|\|\hat{V}^T\hat{B}\hat{V}D^{-1/2}\| \\ &\quad + \|\hat{D}^{-1/2}\hat{V}^T\hat{B}\hat{V}\|\|D^{-1/2} - \hat{D}^{-1/2}\| \\ &\leq 4\epsilon\sigma_1\sigma_k^{-2}.\end{aligned}$$

Combining this with (8) and the triangle inequality, we have

$$\|OMO^T - \hat{M}\| = \frac{\epsilon}{\sigma_k} + 12\frac{\epsilon\sigma_1}{\sigma_k^2} \leq C\frac{\epsilon\sigma_1}{\sigma_k^2}.$$

■

Since O is almost an isometry, it follows that there is an orthogonal matrix $\tilde{O}$ that is close to O (for example, if $UDV^T = O$ is an SVD, let $\tilde{O} = UV^T$). In this way, we may find an orthogonal $\tilde{O}$ such that

$$\|O - \tilde{O}\| \leq 1 - \sqrt{1 - \frac{16\epsilon^2}{\sigma_k^2}} \leq \frac{16\epsilon^2}{\sigma_k^2}.$$

Now let u be the top eigenvector of M and let u_O be the top eigenvector of OMO^T. Then $\tilde{O}u$ is the top eigenvector of $\tilde{O}M\tilde{O}^T$. The triangle inequality implies that

$$\|OMO^T - \tilde{O}M\tilde{O}^T\| \leq 2\|M\|\|O - \tilde{O}\| \leq \frac{32\epsilon^2}{\sigma_k^2}\|M\|.$$

On the other hand, M was assumed to have a spectral gap of $\delta\|M\|$. By Wedin's theorem, it follows that

$$\|u - \tilde{O}^T u_O\| = \|\tilde{O}u - u_O\| \leq \frac{64\epsilon^2}{\delta\sigma_k^2}.$$

Finally, let $\hat{u}$ be the top eigenvector of $\hat{M}$. By Lemma 17 and Wedin's theorem,

$$\|\hat{u} - u_O\| \leq \frac{C\epsilon\sigma_1}{\delta\sigma_k^2}.$$

Then

$$\|Ou - \hat{u}\| \leq \|O - \tilde{O}\| + \|\tilde{O}u - \hat{h}\| \leq C\max\left\{\frac{\epsilon\sigma_1}{\delta\sigma_k^2}, \frac{\epsilon^2}{\delta\sigma_k^2}\right\} \leq \frac{C\epsilon\sigma_1}{\delta\sigma_k^2}, \tag{9}$$

where the last inequality follows because $\epsilon \leq \sigma_k/2 \leq \sigma_1/2$.

Next, we unpack O. Weyl's inequality implies that

$$\|D^{-1/2} - \hat{D}^{-1/2}\| \leq \sigma_k^{-1/2} - (\sigma_k - \epsilon)^{-1/2} \leq \epsilon\sigma_k^{-3/2},$$

where the second inequality follows from a first-order Taylor expansion and the fact that $\epsilon \leq \sigma_k/4$. Hence,

$$\|O - \hat{D}^{-1/2}\hat{V}^T V D^{1/2}\| \leq \|D^{1/2}\|\|D^{-1/2} - \hat{D}^{-1/2}\| \leq \frac{\epsilon\sqrt{\sigma_1}}{\sigma_k^{3/2}}.$$

The right hand side is smaller than $\frac{\epsilon\sigma_1}{\sigma_k^2}$, and so we may plug it into (9) to obtain

$$\|\hat{D}^{-1/2}\hat{V}^T V D^{1/2}u - \hat{u}\| \leq \frac{C\epsilon\sigma_1}{\delta\sigma_k^2}.$$

Finally, (again because $\epsilon \leq \sigma_k/2$), $\|\hat{D}^{-1/2}\| \leq (\sigma_k/2)^{-1/2}$, and so

$$\|VD^{1/2}u - \hat{V}\hat{D}^{1/2}\hat{u}\| \leq \frac{C\epsilon\sigma_1}{\delta\sigma_k^{5/2}}. \tag{10}$$

Setting $w = VD^{1/2}u$ and $\hat{w} = \hat{V}\hat{D}^{1/2}\hat{u}$ and comparing this to the setting of Algorithm 1, (10) shows that the finite-sample algorithm gets almost the same w as the infinite-sample version.

It remains to check the last few lines of Algorithm 1; i.e., to see that we recover the right scaling of w.

Lemma 18 *Let M be a symmetric matrix of rank $k-1$ and let E be the span of its columns. Then $\|w\|\operatorname{dist}(w, E) \geq \sigma_k(M + ww^T)$.*

Proof It suffices to consider the case $\|w\| = 1$ (for a general w, apply the special case of the lemma to $w/\|w\|$ and $M/\|w\|^2$). Let P_E denote the orthogonal projection onto E, and note that $\|w - P_E w\| = \operatorname{dist}(w, E)$ Let $F = \operatorname{span}\{E, w\}$. Since F has dimension k and $y \in F^\perp$ implies $\|(M + ww^T)y\| = 0$, it suffices to find some $y \in F$ such that $\|(M + ww^T)y\| \leq \operatorname{dist}(w, E)\|y\|$. Choose $y = w - P_E w$. Then $My = 0$ and so

$$\|(M + ww^T)y\| = |w^T y| = \|w - P_E w\|^2 = \operatorname{dist}(w, E)\|y\|.$$

■

Lemma 19 *Let E be a subspace and take $w \notin E$. For $x \in \operatorname{span}\{E, w\}$, let $a(x) \in \mathbb{R}$ be the unique solution to $x = aw + e$, $e \in E$. Then $|a(x) - a(y)| \leq \|x - y\|/\operatorname{dist}(w, E)$.*

Proof Given $x, y \in \operatorname{span}\{E, w\}$, we can write $x - y = (a(x) - a(y))w + e$, where $e \in E$. It follows that

$$\begin{aligned}\|x - y\| &= \|(a(x) - a(y))w + e\| \geq \inf_{e \in E}\|(a(x) - a(y))w + e\| \\ &= |a(x) - a(y)|\operatorname{dist}(w, E).\end{aligned}$$

■

Finally, we apply the preceding two lemmas to show that $\hat{\alpha}_1$ is accurate in Algorithm 1. Together with (10) (whose right hand side provides the value of η that we will use), this completes the proof of Theorem 1.

Lemma 20 *Let $m = \sum_i \alpha_i\mu_i$. If $\|\hat{A} - A\| \leq \epsilon$, $\|\hat{m} - m\| \leq \epsilon$ and $\|\hat{w} - \sqrt{\alpha_1}\mu_1\| \leq \eta$ then*

$$|\hat{\alpha}_1 - \alpha_1| \leq \frac{C\sqrt{\alpha_1}|\alpha_1 R + \eta|}{\sigma_k}\left(\eta + R\frac{\epsilon}{\sigma_k} + \epsilon\right),$$

where $R = \max_i \|\mu_i\|$, provided that the right hand side above is at most α_1.

Proof By Wedin's theorem,

$$\|VV^T - \hat{V}\hat{V}^T\| \leq \frac{2\|\hat{A} - A\|}{\sigma_k - \|\hat{A} - A\|} \leq 4\frac{\epsilon}{\sigma_k}$$

if $\epsilon \leq \sigma_k/2$. Hence,

$$\begin{aligned} \|m - \hat{V}\hat{V}^T\hat{m}\| &= \|VV^T m - \hat{V}\hat{V}^T\hat{m}\| \\ &\leq \|(VV^T - \hat{V}\hat{V}^T)m\| + \|\hat{V}\hat{V}^T(m - \hat{m})\| \\ &\leq 4\frac{\epsilon}{\sigma_k}\|m\| + \epsilon. \end{aligned}$$

Now, let $y = \sqrt{\alpha_1}\hat{w} + \hat{V}\hat{V}^T\sum_{i=2}^k \alpha_i\mu_i$. Then

$$\begin{aligned} \|m - y\| &\leq \sqrt{\alpha_1}\|\hat{w} - \sqrt{\alpha_1}\mu_1\| + \left\|\sum_{i=2}^k \alpha_i(\mu_i - \hat{V}\hat{V}^T\mu_i)\right\| \\ &\leq \eta + \max_i \|\mu_i\|\|VV^T - \hat{V}\hat{V}^T\| \\ &\leq \eta + 4\max_i \|\mu_i\|\frac{\epsilon}{\sigma_k}. \end{aligned}$$

Defining $R = \max_i \|\mu_i\|$, we have

$$\|y - \hat{V}\hat{V}^T\hat{m}\| \leq \eta + 8R\frac{\epsilon}{\sigma_k} + \epsilon.$$

Now, let $\hat{E}$ be the span of $\{\hat{V}\hat{D}^{1/2}v : v \in \mathbb{R}^k, v \perp \hat{u}\}$, and note that $\hat{E}$ may also be written as the column space of $\hat{V}\hat{D}^{1/2}(I_k - \hat{u}\hat{u}^T)\hat{D}^{1/2}\hat{V}^T = \hat{V}\hat{D}\hat{V}^T - \hat{w}\hat{w}^T$. Since $\hat{V}\hat{D}^{1/2}$ is injective, $\hat{E}$ has dimension $k-1$ and does not contain $\hat{w} = \hat{V}\hat{D}^{1/2}\hat{u}$. Hence, $y = \sqrt{\alpha_1}\hat{w} + e$ is the unique way to decompose y in $\text{span}\{\hat{w}\} \oplus \hat{E}$. If we define a by the decomposition $\hat{m} = a\hat{w} + e$ then Lemma 19 implies

$$\begin{aligned} |a - \sqrt{\alpha_1}| &\leq \|y - \hat{m}\|/\operatorname{dist}(\hat{w}, \hat{E}) \\ &\leq \frac{1}{\operatorname{dist}(\hat{w}, \hat{E})}\left(\eta + 8R\frac{\epsilon}{\sigma_k} + \epsilon\right). \end{aligned}$$

On the other hand, Lemma 18 applied to $\hat{V}\hat{D}\hat{V}^T - \hat{w}\hat{w}^T$ and $\hat{w}$ implies (because the kth singular value of $\hat{V}\hat{D}\hat{V}^T \geq \sigma_k - \epsilon \geq \sigma_k/2$) that $\|\hat{w}\|\operatorname{dist}(\hat{w}, \hat{E}) \geq \sigma_k/2$. Therefore,

$$|a - \sqrt{\alpha_1}| \leq \frac{2\|\hat{w}\|}{\sigma_k}\left(\eta + 8R\frac{\epsilon}{\sigma_k} + \epsilon\right) \leq \frac{2(\alpha_1\|\mu_1\| + \eta)}{\sigma_k}\left(\eta + 8R\frac{\epsilon}{\sigma_k} + \epsilon\right).$$

Finally, note that $|\hat{\alpha}_1 - \alpha_1| = |a^2 - \alpha_1| = |a - \sqrt{\alpha_1}|(a + \sqrt{\alpha_1})$. We consider two cases: if $a \leq C\sqrt{\alpha_1}$ then $|\hat{\alpha}_1 - \alpha_1| \leq (1 + C)\sqrt{\alpha_1}|a - \sqrt{\alpha_1}|$, which completes the proof. In the other case, we have

$$|\hat{\alpha}_1 - \alpha_1| \sim \hat{\alpha}_1 \leq C\sqrt{\hat{\alpha}_1}|a - \sqrt{\alpha_1}|,$$

which implies that

$$|\hat{\alpha}_1 - \alpha_1| \leq C|a - \sqrt{\alpha_1}|^2$$

for some other constant C. This implies

$$|\hat{\alpha}_1 - \alpha_1| \leq C\left[\frac{(\alpha_1 R + \eta)}{\sigma_k}\left(\eta + R\frac{\epsilon}{\sigma_k} + \epsilon\right)\right]^2 \leq C\sqrt{\alpha_1}\left[\frac{(\alpha_1 R + \eta)}{\sigma_k}\left(\eta + R\frac{\epsilon}{\sigma_k} + \epsilon\right)\right],$$

where the second inequality comes from the assumption that the right hand side in the lemma is bounded by α_1. ■

As we pointed out in Section 2, spectral algorithms similar to Algorithm 1 has been proposed before for GMM [Hsu and Kakade 2013] and LDA [Anandkumar et al. 2012] models, the main difference being how the second matrix (equivalent to B) is constructed. Since the underlying whitening procedure is the same in all these algorithms, the proof approach presented above is similar to those in Hsu and Kakade (2013); Anandkumar et al. (2012). The proofs diverge when computing the perturbation of the second matrix, matrix B in our algorithm, which introduces different dependence on various parameter models in the overall error bound. For example the error bound in Theorem 4.1 of Anandkumar et al. (2012) has a slightly worse dependence on k and σ_k than Theorem 1.

Appendix E. Finite-sample Analysis of the Cancellation Method

In this section we analyze the performance of Algorithm 2 when we have finite sample estimates of the matrices A, B and vector m. For ease of exposition we replaced the quantities $V_{1:(k-1)}, v_i, a_i, c_i$ in Algorithm 2 with the notation representing estimate $\widehat{V}_{1:(k-1)}, \hat{v}_i, \hat{a}_i, \hat{c}_i$ respectively, since these are computed from sample estimates $\widehat{A}, \widehat{B}$. First, we show in Lemma 21 that we can have a good estimate for $\widehat{Z}_{\lambda^*}$ using good estimates for A, B and λ_1.

Lemma 21 *Let $\widehat{Z}_\lambda = \widehat{A} - \lambda\widehat{B}, Z_\lambda = A - \lambda B$. Suppose $\max\{\|\widehat{A} - A\|, \|\widehat{B} - B\|\} < \epsilon$ and $\lambda_1 = 1/w_1$. Then,*

$$\|\widehat{Z}_\lambda - Z_{\lambda_1}\| < \epsilon\left(2 + \frac{1}{w_1}\right) + \epsilon_1\sigma_1(B)$$

when $|\lambda_1 - \lambda| < \epsilon_1 < 1$.

Proof We have,

$$\begin{aligned} \|\widehat{Z}_\lambda - Z_{\lambda_1}\| &\leq \|\widehat{A} - A\| + \|\lambda\widehat{B} - \lambda_1 B\| \\ &< \|\widehat{A} - A\| + \lambda_1\|\widehat{B} - B\| + |\lambda_1 - \lambda|\|\widehat{B}\| \\ &\leq \epsilon + \lambda_1\epsilon + \epsilon_1(\sigma_1(B) + \epsilon) \\ &< \epsilon(1 + 1/w_1 + \epsilon_1) + \epsilon_1\sigma_1(B) < \epsilon\left(2 + \frac{1}{w_1}\right) + \epsilon_1\sigma_1(B) \end{aligned}$$

since $\epsilon_1 < 1$. ■

The following lemma will show that even with noisy estimates of A, B, the estimated λ^* is close to λ_1.

Lemma 22 *Let* $\max\{\|\widehat{A}-A\|, \|\widehat{B}-B\|\} < \epsilon < \sigma_k(A)/2$, *and* $\lambda_1 = 1/w_1 > 0$. *Then,*

$$|\lambda^* - \lambda_1| = O(\epsilon)$$

Proof Define $Z'_\lambda = VV^TAVV^T - \lambda VV^TBVV^T$, V being the $d \times k$ matrix of top k eigenvectors of A. The corresponding empirical estimate $\widehat{Z}'_\lambda = \widehat{V}\widehat{V}^T\widehat{A}\widehat{V}\widehat{V}^T - \lambda\widehat{V}\widehat{V}^T\widehat{B}\widehat{V}\widehat{V}^T$. The main proof idea is the following. We try to find $\lambda_2, \lambda_3 > 0$ such that:

1. $\forall \lambda > \lambda_2$, $\widehat{Z}'_\lambda$ is not PSD.
2. $\forall \lambda < \lambda_3$, $\widehat{Z}'_\lambda$ is PSD.

The above two conditions imply that the optimum λ^* is bounded as $\lambda_3 \le \lambda^* \le \lambda_2$. We then simply bound $\lambda^* - \lambda_1$ as $\lambda_3 - \lambda_1 \le \lambda^* - \lambda_1 \le \lambda_2 - \lambda_1$. We now elaborate the above two steps. First, we bound the perturbation of empirical matrix $\widehat{Z}'_\lambda$ as follows. Using Wedin's theorem we have $\|\widehat{V}\widehat{V}^T - VV^T\| \le \frac{4\epsilon}{\sigma_k(A)}$. Using this and the theorem assumptions we can compute the following bounds.

$$\begin{aligned}
\|\widehat{V}\widehat{V}^T\widehat{A}\widehat{V}\widehat{V}^T - VV^TAVV^T\| &\le 13\epsilon \\
\|\widehat{V}\widehat{V}^T\widehat{B}\widehat{V}\widehat{V}^T - VV^TBVV^T\| &\le \left(1 + \frac{12\sigma_k(B)}{\sigma_k(A)}\right)\epsilon
\end{aligned}$$

Combining, we have

$$\|\widehat{Z}'_\lambda - Z'_\lambda\| \le \|\widehat{V}\widehat{V}^T\widehat{A}\widehat{V}\widehat{V}^T - VV^TAVV^T\| + \lambda\|\widehat{V}\widehat{V}^T\widehat{B}\widehat{V}\widehat{V}^T - VV^TBVV^T\| \le c_1(1+\lambda)\epsilon \quad (11)$$

where $c_1 = \max\{13, 1 + \frac{12\sigma_k(B)}{\sigma_k(A)}\}$.

Step 1: Since matrices A and B share the same column and row space, $VV^TAVV^T = A$, $VV^TBVV^T = B$, and $Z'_\lambda = Z_\lambda = \sum_{i=1}^k (1 - \lambda w_i)\alpha_i\mu_i\mu_i^T$, $w_i = \langle \mu_i, v\rangle$. Recall, $\mathcal{V} = \text{span}\{\mu_2, \ldots, \mu_k\}$ and Π denote the projection onto $\mathcal{V}_\perp$, its perpendicular space. Let $x_1 = \Pi\mu_1/\|\Pi\mu_1\|$, and $x_1 = V\tilde{x}_1$, $\|x_1\| = \|\tilde{x}_1\| = 1$. Consider the eigenvalues of the $k \times k$ Hermitian matrix $V^TZ_\lambda V$. Using variational theorem we can write:

$$\tilde{\sigma}_k(V^TZ_\lambda V) = \min_{x \ne 0, \|x\|=1} x^TV^TZ_\lambda Vx \le \tilde{x}_1^TV^TZ_\lambda V\tilde{x}_1 = x_1^TZ_\lambda x_1 = (1 - \lambda w_1)\alpha_1 a'_1 \quad (12)$$

where $a'_1 = |\langle x_1, \mu_1\rangle|^2 > 0$. Now note that the matrices $Z'_\lambda = VV^TZ_\lambda VV^T$ and $V^TZ_\lambda V$ have the same set of non-zero eigenvalues since V forms an orthonormal basis of the row/column space of Z_λ. Therefore we can write from above,

$$\tilde{\sigma}_k(Z'_\lambda) = \tilde{\sigma}_k(V^TZ_\lambda V) \le (1 - \lambda w_1)\alpha_1 a'_1 \quad (13)$$

For $\lambda = \lambda_1 = 1/w_1$, Z'_{λ_1} is a rank $k-1$ matrix, and for any $\lambda > \lambda_1$, Z'_λ has at least one negative eigenvalue. Consider $\lambda_2 > \lambda_1$ such that Z'_{λ_2} has one negative eigenvalue and $k-1$ positive eigenvalues. Since $\widehat{Z}'_{\lambda_2}, Z'_{\lambda_2}$ are symmetric matrices, using Weyl's inequality we get,

$$\begin{aligned}
\tilde{\sigma}_k(\widehat{Z}'_{\lambda_2}) &\le \tilde{\sigma}_k(Z'_{\lambda_2}) + \|\widehat{Z}'_{\lambda_2} - Z'_{\lambda_2}\| \le \tilde{\sigma}_k(Z'_{\lambda_2}) + c_1(1+\lambda_2)\epsilon \\
&\le (1 - \lambda_2 w_1)\alpha_1 a'_1 + c_1(1+\lambda_2)\epsilon \\
&\le a'_1[(\alpha_1 + \epsilon) - \lambda_2(w_1\alpha_1 - \epsilon)] \quad (14)
\end{aligned}$$

using equations (11), (13), and assuming $a'_1 > c_1$ (else we can simply rescale ϵ). Now for any $\lambda > \lambda_2 = \frac{\alpha_1+\epsilon}{\alpha_1 w_1 - \epsilon}$ we get

$$\tilde{\sigma}_k(\widehat{Z}'_\lambda) \le a'_1[(\alpha_1 + \epsilon) - \lambda(w_1\alpha_1 - \epsilon)] \le a'_1[(\alpha_1 + \epsilon) - \lambda_2(w_1\alpha_1 - \epsilon)] = 0$$

Therefore, when $\lambda > \lambda_2 = \frac{\alpha_1+\epsilon}{\alpha_1 w_1 - \epsilon}$, $\widehat{Z}'_\lambda$ is not PSD. This implies that $\lambda_2 \ge \lambda^*$. Then,

$$\lambda^* - \lambda_1 \le \lambda_2 - \lambda_1 = \frac{\alpha_1 + \epsilon}{\alpha_1 w_1 - \epsilon} - \frac{1}{w_1} = \frac{\epsilon(w_1 + 1)}{(\alpha_1 w_1 - \epsilon)w_1} \quad (15)$$

Step 2: Consider $\lambda_3 < \lambda_1$ such that Z'_{λ_3} is PSD. Then we lower bound $\tilde{\sigma}_k(Z'_{\lambda_3})$ as follows. Let $\tilde{v}_{k,\lambda_3}$ be the $k-$th eigenvector of Z'_{λ_3} having eigenvalue $\tilde{\sigma}_k(Z'_{\lambda_3})$. Then,

$$\begin{aligned}
\tilde{\sigma}_k(Z'_{\lambda_3}) &= \tilde{v}_{k,\lambda_3}^TZ'_{\lambda_3}\tilde{v}_{k,\lambda_3} = \sum_{i=1}^k \alpha_i(1 - \lambda_3 w_i)\tilde{v}_{k,\lambda_3}^T\mu_i\mu_i^T\tilde{v}_{k,\lambda_3} \\
&\ge (1 - \lambda_3 w_1)\sum_{i=1}^k \alpha_i|\langle \tilde{v}_{k,\lambda_3}, \mu_i\rangle|^2 \ge (1 - \lambda_3 w_1)a'_2 \quad (16)
\end{aligned}$$

since $w_1 > w_i$, $i \ne 1$, and where $a'_2 = \inf_{\lambda \ge 0}\sum_{i=1}^k \alpha_i|\langle \tilde{v}_{k,\lambda}, \mu_i\rangle|^2 > 0$. Now using the lower bound of Weyl's inequality,

$$\begin{aligned}
\tilde{\sigma}_k(\widehat{Z}'_{\lambda_3}) &\ge \tilde{\sigma}_k(Z'_{\lambda_3}) - \|\widehat{Z}'_{\lambda_3} - Z'_{\lambda_3}\| \\
&\ge \tilde{\sigma}_k(Z'_{\lambda_3}) - c_1(1 + \lambda_3)\epsilon \\
&\ge (1 - \lambda_3 w_1)a'_2 - c_1(1 + \lambda_3)\epsilon \\
&\ge c_1[(1 - \epsilon) - \lambda_3(w_1 + \epsilon)]
\end{aligned}$$

using equation (16), and assuming $c_1 < a'_2$ (else we can simply rescale ϵ). Then, for any $\lambda < \lambda_3 = \frac{(1-\epsilon)}{(w_1+\epsilon)}$ we have $\tilde{\sigma}_k(\widehat{Z}'_\lambda) > 0$, or $\widehat{Z}'_\lambda$ is PSD. This implies $\lambda^* > \lambda_3$. Therefore,

$$\lambda^* - \lambda_1 \ge \lambda_3 - \lambda_1 = \frac{(1 - \epsilon)}{(w_1 + \epsilon)} - \frac{1}{w_1} = -\frac{(w_1 + 1)\epsilon}{(w_1 + \epsilon)w_1} \quad (17)$$

Combining equations (15), (17) we get,

$$|\lambda^* - \lambda_1| \le c_3\epsilon = O(\epsilon)$$

where $c_3 = \max\left(\frac{(w_1+1)}{(w_1+\epsilon)w_1}, \frac{(w_1+1)}{(\alpha_1 w_1 - \epsilon)w_1}\right)$. ■

In Lemma 22 we assume $w_1 = \langle \mu_1, v\rangle$ is positive. When $w_1 < 0$, we have to modify the line search and find the smallest $\lambda < 0$ such that $\widehat{Z}'_\lambda$ is PSD. However we can still apply similar arguments and prove that as long as the estimates of A, B, are within ϵ in spectral norm, Algorithm 2 can estimate λ^* within an $O(\epsilon)$ accuracy of λ_1. Lemma 21 and 22

together implies that $\|\widehat{Z}_{\lambda^*} - Z_{\lambda_1}\| = O(\epsilon)$ as follows, which will be used to prove Theorem 3. We have,

$$\begin{aligned}\|\widehat{Z}_{\lambda^*} - Z_{\lambda_1}\| &< \epsilon\left(2 + \frac{1}{w_1}\right) + \epsilon_1\sigma_1(B) \\ &\leq \epsilon\left(2 + \frac{1}{w_1}\right) + c_3\epsilon\sigma_1(B) \\ &\leq 3\eta_3\epsilon \end{aligned} \tag{18}$$

where in the last inequality we assume $\epsilon < \alpha_1 w_1/2$, and $\eta_3 = \max\left\{2, \frac{1}{w_1}, c_3\sigma_1(B)\right\}$.

Lemma 23 *Let $\|\hat{m} - m\| < \epsilon$, $\|\widehat{Z}_{\lambda^*} - Z_{\lambda_1}\| < \epsilon_2 < \sigma_{k-1}(Z_{\lambda_1})/2$ for $\lambda_1 = \alpha_1/\beta_1$. $V_{1:(k-1)}$ denote the $d \times (k-1)$ matrix of $k-1$ largest singular vectors of Z_{λ_1} and $\widehat{V}_{1:(k-1)}$ be the $d \times (k-1)$ matrix of $k-1$ largest singular vectors of $\widehat{Z}_{\lambda^*}$. Then,*

$$\begin{aligned}\|\hat{x}_1 - x_1\| &< 2\epsilon + \frac{4\epsilon_2 R}{\sigma_{k-1}(Z_{\lambda_1})} = \epsilon_3 \\ \|\hat{v}_1 - v_1\| &< \frac{2\epsilon_3}{\alpha_1 a_1} = \epsilon_4\end{aligned}$$

where $R = \max_{i\in[k]} \|\mu_i\|$.

Proof Since, $\|\widehat{Z}_{\lambda^*} - Z_{\lambda_1}\| < \epsilon_2 < \sigma_{k-1}(Z_{\lambda_1})/2$, applying Wedin's theorem we get,

$$\|\widehat{V}_{1:(k-1)}\widehat{V}_{1:(k-1)}^T - V_{1:(k-1)}V_{1:(k-1)}^T\| \leq \frac{2\|\widehat{Z}_{\lambda^*} - Z_{\lambda_1}\|}{\sigma_{k-1}(Z_{\lambda_1}) - \|\widehat{Z}_{\lambda^*} - Z_{\lambda_1}\|} \leq \frac{4\epsilon_2}{\sigma_{k-1}(Z_{\lambda_1})} \tag{19}$$

since $\epsilon_2 < \sigma_{k-1}(Z_{\lambda_1})/2$. Now,

$$\begin{aligned}\|\hat{x}_1 - x_1\| &= \|\hat{m} - \widehat{V}_{1:(k-1)}\widehat{V}_{1:(k-1)}^T\hat{m} - m + V_{1:(k-1)}V_{1:(k-1)}^T m\| \\ &\leq \|\hat{m} - m\| + \|(\widehat{V}_{1:(k-1)}\widehat{V}_{1:(k-1)} - V_{1:(k-1)}V_{1:(k-1)}^T)m\| + \|\widehat{V}_{1:(k-1)}\widehat{V}_{1:(k-1)}^T(m - \hat{m})\| \\ &< 2\|m - \hat{m}\| + \frac{4\epsilon_2\|m\|}{\sigma_{k-1}(Z_{\lambda_1})} < 2\epsilon + \frac{4\epsilon_2 R}{\sigma_{k-1}(Z_{\lambda_1})} := \epsilon_3\end{aligned}$$

where we used equation 19 and $\|m\| \leq R$. Recall that $x_1 = \alpha_1 \prod_{\mathcal{V}} \mu_1 = \alpha_1 a_1 v_1$, where $\mathcal{V} = \text{span}\{\mu_2, \ldots, \mu_k\}$ and $a_1 = \langle \mu_1, v_1 \rangle$. To show the second bound,

$$\begin{aligned}\|\hat{v}_1 - v_1\| &= \left\|\frac{\hat{x}_1}{\|\hat{x}_1\|} - \frac{x_1}{\|x_1\|}\right\| \\ &\leq \frac{\|\hat{x}_1 - x_1\|}{\|x_1\|} + \|\hat{x}_1\|\left|\frac{1}{\|x_1\|} - \frac{1}{\|\hat{x}_1\|}\right| \\ &< \frac{\|\hat{x}_1 - x_1\|}{\|x_1\|} + \frac{|\|\hat{x}_1\| - \|x_1\||}{\|x_1\|} \leq 2\frac{\|\hat{x}_1 - x_1\|}{\|x_1\|} \\ &< \frac{2\epsilon_3}{\alpha_1 a_1} := \epsilon_4\end{aligned}$$

■

Lemma 24 *Let $\|\widehat{A} - A\| < \epsilon, \|\hat{v}_1 - v_1\| < \epsilon_4$. Define $d \times k$ matrices $V = [v_1 V_{1:(k-1)}]$ and $\widehat{V} = [\hat{v}_1 \widehat{V}_{1:(k-1)}]$. Then,*

$$\|\widehat{V}\widehat{V}^T\widehat{A}\hat{v}_1 - VV^TAv_1\| < \sigma_1(A)\left(3\epsilon_4 + \frac{4\epsilon}{\sigma_{k-1}(Z_{\lambda_1})}\right) + \epsilon(1 + \epsilon_4)$$

Proof Similar to Lemma 23 we have from Wedin's theorem $\|\widehat{V}_{1:(k-1)}\widehat{V}_{1:(k-1)}^T - V_{1:(k-1)}V_{1:(k-1)}^T\| < \frac{4\epsilon}{\sigma_{k-1}(Z_{\lambda_1})}$. Then we can bound,

$$\begin{aligned}\|\widehat{V}\widehat{V}^T - VV^T\| &\leq \|\hat{v}_1\hat{v}_1^T - v_1v_1^T\| + \|\widehat{V}_{1:(k-1)}\widehat{V}_{1:(k-1)} - V_{1:(k-1)}V_{1:(k-1)}^T\| \\ &< 2\|\hat{v}_1 - v_1\| + \frac{4\epsilon}{\sigma_{k-1}(Z_{\lambda_1})} \\ &< 2\epsilon_4 + \frac{4\epsilon}{\sigma_{k-1}(Z_{\lambda_1})}\end{aligned} \tag{20}$$

Now,

$$\begin{aligned}\|\widehat{V}\widehat{V}^T\widehat{A}\hat{v}_1 - VV^TAv_1\| &\leq \|(\widehat{V}\widehat{V}^T - VV^T)Av_1\| + \|\widehat{V}\widehat{V}^T(A - \widehat{A})v_1\| \\ &\quad + \|\widehat{V}\widehat{V}^T\widehat{A}(v_1 - \hat{v}_1)\| \\ &\leq \|\widehat{V}\widehat{V}^T - VV^T\|\|A\| + \|A - \widehat{A}\| + \|\widehat{A}\|\|v_1 - \hat{v}_1\| \\ &< \sigma_1(A)\left(2\epsilon_4 + \frac{4\epsilon}{\sigma_{k-1}(Z_{\lambda_1})}\right) + \epsilon + (\sigma_1(A) + \epsilon)\epsilon_4\end{aligned}$$

where we use inequality (20), $\|Av_1\| \leq \sigma_1(A)$ as v_1 is unit norm, $\|\widehat{V}\widehat{V}^T\| < 1$ since $\widehat{V}$ is orthonormal, and $\|\widehat{A}\| < \|A\| + \epsilon$. Combining,

$$\|\widehat{V}\widehat{V}^T\widehat{A}\hat{v}_1 - VV^TAv_1\| < \sigma_1(A)\left(3\epsilon_4 + \frac{4\epsilon}{\sigma_{k-1}(Z_{\lambda_1})}\right) + \epsilon(1 + \epsilon_4)$$

■

Lemma 25 *Let $\|\widehat{A} - A\| < \epsilon$, $\|\hat{x}_1 - x_1\| < \epsilon_3 < \frac{\alpha_1 a_1}{2}$, and $\|\hat{v}_1 - v_1\| < \epsilon_4$. Then,*

$$|\hat{a}_1 - a_1| < \frac{\alpha_1 a_1\left(2\sigma_1(A)\epsilon_4 + \epsilon(1 + \epsilon_4)\right) + 2(\sigma_1(A) + \epsilon)\epsilon_3}{\alpha_1^2 a_1^2}$$

Proof We first compute,

$$\begin{aligned}|\hat{v}_1^T\widehat{A}\hat{v}_1 - v_1^TAv_1| &\leq |(v_1^T - \hat{v}_1^T)Av_1| + |\hat{v}_1^T(A - \widehat{A})v_1| + |\hat{v}_1^T\widehat{A}(v_1 - \hat{v}_1)| \\ &\leq \|v_1^T - \hat{v}_1^T\|\sigma_1(A) + \|A - \widehat{A}\| + \sigma_1(\widehat{A})\|v_1 - \hat{v}_1\| \\ &< \sigma_1(A)\epsilon_4 + \epsilon + (\sigma_1(A) + \epsilon)\epsilon_4 = 2\sigma_1(A)\epsilon_4 + \epsilon(1 + \epsilon_4)\end{aligned} \tag{21}$$

using the fact that $v_1, \hat{v}_1$ have unit norms. Now we can bound the error $|\hat{a}_1 - a_1|$ as follows.

$$\begin{aligned}|\hat{a}_1 - a_1| &= \left|\frac{\hat{v}_1^T \widehat{A}\hat{v}_1}{\|\hat{x}_1\|} - \frac{v_1^T A v_1}{\|x_1\|}\right| \\ &\leq \frac{1}{\|x_1\|}|\hat{v}_1^T\widehat{A}\hat{v}_1 - v_1^T A v_1| + |\hat{v}_1^T\widehat{A}\hat{v}_1|\frac{|\|x_1\| - \|\hat{x}_1\||}{\|x_1\|\|\hat{x}_1\|}\end{aligned}$$

From equation (21) and using $|\|x_1\| - \|\hat{x}_1\|| < \|\hat{x}_1 - x_1\| < \epsilon_3, \|x_1\| = \alpha_1 a_1$ we get,

$$\begin{aligned}|\hat{a}_1 - a_1| &< \frac{2\sigma_1(A)\epsilon_4 + \epsilon(1+\epsilon_4)}{\alpha_1 a_1} + \frac{(\sigma_1(A)+\epsilon)\epsilon_3}{\alpha_1 a_1(\alpha_1 a_1 - \epsilon_3)} \\ &< \frac{\alpha_1 a_1\left(2\sigma_1(A)\epsilon_4 + \epsilon(1+\epsilon_4)\right) + 2(\sigma_1(A)+\epsilon)\epsilon_3}{\alpha_1^2 a_1^2}\end{aligned}$$

since $\epsilon_3 < \frac{\alpha_1 a_1}{2}$. ■

Note that from Lemma 23 taking $\frac{2\epsilon_3}{\alpha_1 a_1} = \epsilon_4$ the above bound becomes $|\hat{a}_1 - a_1| < \frac{6\sigma_1(A)\epsilon_3 + \epsilon\alpha_1 a_1 + 4\epsilon\epsilon_3}{\alpha_1^2 a_1^2}$.

E.1 Proof of Theorem 3

We now proof Theorem 3. Assume $\|\widehat{Z}_{\lambda^*} - Z_{\lambda_1}\| \leq \epsilon_2$. Under the assumptions we have using Lemma 23 $\|\hat{x}_1 - x_1\| < \epsilon_3 = 2\epsilon + \frac{4\epsilon_2 R}{\sigma_{k-1}(Z_{\lambda_1})}$, $\|\hat{v}_1 - v_1\| < \epsilon_4 = \frac{2\epsilon_3}{\alpha_1 a_1}$. Also from Lemma 24 we have $\|\widehat{V}\widehat{V}^T\widehat{A}\hat{v}_1 - VV^T A v_1\| < \sigma_1(A)\left(3\epsilon_4 + \frac{4\epsilon}{\sigma_{k-1}(Z_{\lambda_1})}\right) + \epsilon(1+\epsilon_4)$. Using these we compute the first bound as follows.

$$\begin{aligned}\|\hat{\mu}_1 - \mu_1\| &= \left\|\frac{\widehat{V}\widehat{V}^T\widehat{A}\hat{v}_1}{\|\hat{x}_1\|} - \frac{VV^T A v_1}{\|x_1\|}\right\| \\ &\leq \|\widehat{V}\widehat{V}^T\widehat{A}\hat{v}_1\|\left|\frac{1}{\|\hat{x}_1\|} - \frac{1}{\|x_1\|}\right| + \frac{1}{\|x_1\|}\|\widehat{V}\widehat{V}^T\widehat{A}\hat{v}_1 - VV^T A v_1\| \\ &\leq \|\widehat{A}\|\frac{\|\hat{x}_1 - x_1\|}{\|\hat{x}_1\|\|x_1\|} + \frac{1}{\|x_1\|}\|\widehat{V}\widehat{V}^T\widehat{A}\hat{v}_1 - VV^T A v_1\|\end{aligned}$$

Now using bounds from Lemma 23, 24 we get,

$$\begin{aligned}\|\hat{\mu}_1 - \mu_1\| &< \frac{(\sigma_1(A)+\epsilon)\epsilon_3}{\alpha_1 a_1(\alpha_1 a_1 - \epsilon_3)} + \frac{\sigma_1(A)\left(3\epsilon_4 + \frac{4\epsilon}{\sigma_{k-1}(Z_{\lambda_1})}\right) + \epsilon(1+\epsilon_4)}{\alpha_1 a_1} \\ &< \frac{2}{\alpha_1^2 a_1^2}\left[(\sigma_1(A)+\epsilon)\epsilon_3 + \alpha_1 a_1\left((3\sigma_1(A)+\epsilon)\epsilon_4\right.\right. \\ &\quad \left.\left. +\epsilon\left(1 + 4\sigma_1(A)/\sigma_{k-1}(Z_{\lambda_1})\right)\right)\right] \\ &< \frac{2}{\alpha_1^2 a_1^2}\left[(\sigma_1(A)+\epsilon)\epsilon_3 + 2\left(3\sigma_1(A)+\epsilon\right)\epsilon_3\right. \\ &\quad \left. +\alpha_1 a_1\epsilon\left(1 + 4\sigma_1(A)/\sigma_{k-1}(Z_{\lambda_1})\right)\right] \\ &\leq 2\frac{10\sigma_1(A)\epsilon_3 + 5\alpha_1 a_1\epsilon\frac{\sigma_1(A)}{\sigma_{k-1}(Z_{\lambda_1})}}{\alpha_1^2 a_1^2}\end{aligned}$$

assuming $\epsilon_3 \leq \frac{\alpha_1 a_1}{2}$, $\sigma_1(A) \geq \epsilon$, and $\sigma_1(A) > \sigma_{k-1}(Z_{\lambda_1})$. Now expanding ϵ_3 and rearranging terms we have,

$$\begin{aligned}\|\hat{\mu}_1 - \mu_1\| &< \frac{1}{\alpha_1^2 a_1^2}\left(\left(40 + 10\frac{\alpha_1 a_1}{\sigma_{k-1}(Z_{\lambda_1})}\right)\sigma_1(A)\epsilon + 80\frac{\sigma_1(A)R\epsilon_2}{\sigma_{k-1}(Z_{\lambda_1})}\right) \\ &< \frac{80}{\alpha_1^2 a_1^2}\left(\sigma_1(A)\epsilon\left(1 + \frac{\alpha_1 a_1}{\sigma_{k-1}(Z_{\lambda_1})}\right) + \frac{\sigma_1(A)\epsilon_2 R}{\sigma_{k-1}(Z_{\lambda_1})}\right) \qquad (22)\end{aligned}$$

To prove the second bound from Lemma 25 and assuming $\epsilon < \sigma_1(A)$ we have $|\hat{a}_1 - a_1| \leq \frac{10\sigma_1(A)\epsilon_3 + \alpha_1 a_1\epsilon}{\alpha_1^2 a_1^2}$. Then,

$$\begin{aligned}\hat{a}_1(\alpha_1 - \hat{\alpha}_1) &= \hat{a}_1\alpha_1 - \hat{a}_1\hat{\alpha}_1 \\ &= a_1\alpha_1 - \hat{a}_1\hat{\alpha}_1 + \hat{a}_1\alpha_1 - a_1\alpha_1 \\ \hat{a}_1|\alpha_1 - \hat{\alpha}_1| &\leq |a_1\alpha_1 - \hat{a}_1\hat{\alpha}_1| + \alpha_1|\hat{a}_1 - a_1| \\ |\alpha_1 - \hat{\alpha}_1| &\leq \frac{1}{\hat{a}_1}\left(\|x_1 - \hat{x}_1\| + \alpha_1|\hat{a}_1 - a_1|\right) \\ &< \frac{\epsilon_3 + \alpha_1|\hat{a}_1 - a_1|}{a_1 - |\hat{a}_1 - a_1|} \\ &\leq 2\frac{\epsilon_3 + \frac{(10\sigma_1(A)\epsilon_3 + \alpha_1 a_1\epsilon)}{\alpha_1 a_1^2}}{a_1}\end{aligned}$$

using $|\hat{a}_1 - a_1| < \frac{a_1}{2}$. We have,

$$\begin{aligned}|\alpha_1 - \hat{\alpha}_1| &\leq 2\frac{\alpha_1 a_1^2\epsilon_3 + 10\sigma_1(A)\epsilon_3 + \alpha_1 a_1\epsilon}{\alpha_1 a_1^3} \\ &< \frac{2}{\alpha_1 a_1^3}\left(\left(\alpha_1 a_1^2 + 10\sigma_1(A)\right)\left(2\epsilon + 4R\epsilon_2/\sigma_{k-1}(Z_{\lambda_1})\right) + \alpha_1 a_1\epsilon\right) \\ &\leq \frac{4\sigma_1(A)}{\alpha_1 a_1^3}\left(\eta_1\epsilon + \frac{\eta_2 R\epsilon_2}{\sigma_{k-1}(Z_{\lambda_1})}\right) \qquad (23)\end{aligned}$$

where $\eta_1 := \max\{\alpha_1 a_1(2a_1 + 1), 20\}$, and $\eta_2 := \max\{\alpha_1 a_1^2, 10\}$.

Finally using equation (18) we can bound $\|\widehat{Z}_{\lambda^*} - Z_{\lambda_1}\| \leq \epsilon_2 \leq 3\eta_3\epsilon$, where $\eta_3 = \max\left\{1, \frac{1}{w_1}, c_3\sigma_1(B)\right\}$. Using this in equations (22) and (23) proves the theorem.

E.2 Related Lemmas

In this section we prove a supporting lemma for Lemma 5.

Lemma 26 *Let $\{\mu_2, \ldots, \mu_k\}$ be linearly independent. Suppose matrix Z_{λ^*} be expressed as,*

$$Z_{\lambda^*} = \sum_{i=2}^{k}\alpha_i(1 - \lambda^* w_i)\mu_i\mu_i^T = V_{1:(k-1)}\Sigma_{1:(k-1)}V_{1:(k-1)}^T = \sum_{i=2}^{k}\sigma_{i-1}(Z_{\lambda^*})v_i v_i^T, \qquad (24)$$

where $w_i = \langle\mu_i, v\rangle$, $V_{1:(k-1)} = [v_2, \ldots, v_k]$ the matrix of $k-1$ singular vectors, and $\Sigma_{1:(k-1)}$ is a diagonal matrix of singular values of Z_{λ^*}. Then $\{v_2, \ldots, v_k\}$ forms a basis of span$\{\mu_2, \ldots, \mu_k\}$.

Proof Define $\mathcal{V}_{Z_{\lambda^*}}$ as the **column space** of matrix Z_{λ^*}. First observe that from equation (24) each column of Z_{λ^*} can be written as a linear combination of $\{\mu_2, \ldots, \mu_k\}$. Therefore any vector in the column space $\mathcal{V}_{Z_{\lambda^*}}$ can be written as a linear combination of $\{\mu_2, \ldots, \mu_k\}$. this implies,

$$\mathcal{V}_{Z_{\lambda^*}} \subseteq span\{\mu_2, \ldots, \mu_k\} \tag{25}$$

Now any vector $y \in \mathcal{V}_{Z_{\lambda^*}}$ can be written as $y = Z_{\lambda^*}x = \sum_{i=2}^{k} \sigma_{i-1}(Z_{\lambda^*})\langle v_i, x\rangle v_i$ using equation (24). This implies,

$$\mathcal{V}_{Z_{\lambda^*}} \subseteq span\{v_2, \ldots, v_k\} \tag{26}$$

Conversely any vector $s \in span\{v_2, \ldots, v_k\}$ can be written as $s = V_{1:(k-1)}r = Z_{\lambda^*}V_{1:(k-1)}\Sigma^{-1}_{1:(k-1)}r = Z_{\lambda^*}r'$, using equation (24), where $r' = V_{1:(k-1)}\Sigma^{-1}_{1:(k-1)}r$. This implies,

$$span\{v_2, \ldots, v_k\} \subseteq \mathcal{V}_{Z_{\lambda^*}} \tag{27}$$

Therefore combining equations (25),(26),(27) we get,

$$span\{v_2, \ldots, v_k\} = \mathcal{V}_{Z_{\lambda^*}} \subseteq span\{\mu_2, \ldots, \mu_k\} \tag{28}$$

Note that both the vector spaces $span\{v_2, \ldots, v_k\}$ and $span\{\mu_2, \ldots, \mu_k\}$ have rank $k-1$ since $\{v_2, \ldots, v_k\}$ are orthonormal, and $\{\mu_2, \ldots, \mu_k\}$ are linearly independent. Then from this rank constraint and equation (28) we must have:

$$span\{v_2, \ldots, v_k\} = span\{\mu_2, \ldots, \mu_k\}$$

This implies $\{v_2, \ldots, v_k\}$ forms a basis of $span\{\mu_2, \ldots, \mu_k\}$. ■

Appendix F. Subspace Clustering Proofs

In this section we prove Theorem 6 and the necessary lemmas. The main point is the following infinite-sample analysis, which shows that the top m eigenvectors of the whitened matrix B can be used to recover the subspace $\mathcal{U}_1$.

Theorem 27 *Suppose that there is some $\delta > 0$ such that $\|U_i v\|^2 \leq (1/3 - \delta)\|U_1 v\|^2$ for all $i \neq 1$. Let $Y = [u_1, ..., u_m]$ be the matrix of top m eigenvectors of $R = D^{-1/2}V^T BVD^{-1/2}$ and $Z = VD^{1/2}Y$. Let $\mathcal{Z}$ be the subspace spanned by columns of Z. Then,*

1. *$\mathcal{Z} = \mathcal{U}_1$*
2. *$\sigma_m(R) - \sigma_{m+1}(R) \geq 3\delta\|U_1 v\|^2$*

Proof Define $w_i = \|U_i U_i^T v\| = \|U_i^T v\|$, and $\tilde{U}_i := \sqrt{\alpha_i}D^{-1/2}V^T U_i$; note that $\sum_{i=1}^{k} \tilde{U}_i\tilde{U}_i^T$ is the $(km) \times (km)$ identity matrix, which implies that each $\tilde{U}_i$ has orthonormal columns. Consider the whitened B matrix. Using Theorem 13,

$$\begin{aligned} D^{-1/2}V^T BVD^{-1/2} &= \sum_{i=1}^{k} w_i^2 \tilde{U}_i\tilde{U}_i^T + 2\sum_{i=1}^{k} \tilde{U}_i U_i^T vv^T U_i\tilde{U}_i^T \\ &= \sum_{i=1}^{k} w_i^2 \tilde{U}_i\tilde{U}_i^T + 2\sum_{i=1}^{k} \tilde{v}_i\tilde{v}_i^T = \sum_{i=1}^{k}(w_i^2\tilde{U}_i\tilde{U}_i^T + 2\tilde{v}_i\tilde{v}_i^T) \end{aligned}$$

where $\tilde{v}_i = \tilde{U}_i U_i^T v$. Note that $\tilde{v}_i$ are orthogonal to each other and each $\tilde{v}_i$ is in the space $\tilde{\mathcal{U}}_i$, the span of corresponding $\tilde{U}_i$. Moreover, $\|\tilde{v}_i\| = w_i$. Now for each i consider a different orthonormal basis $\tilde{V}_i$ of $\tilde{\mathcal{U}}_i$ such that in this basis the first unit vector is aligned along $\tilde{v}_i$. Define a rotation R_i such that $\tilde{V}_i = \tilde{U}_i R_i$. Then $\tilde{V}_i\tilde{V}_i^T = \tilde{U}_i\tilde{U}_i^T$. Therefore we can write the above equation as

$$R = D^{-1/2}V^T BVD^{-1/2} = \sum_{i=1}^{k} \tilde{V}_i\tilde{D}_i\tilde{V}_i^T \tag{29}$$

where each $\tilde{D}_i$ is a diagonal matrix with one maximum value of $3w_i^2$ and all other values w_i^2, and also the matrices $\tilde{V}_i$ are orthogonal. Under the assumption that $w_i^2 \leq (1/3 - \delta)w_1^2$, it follows that the top m eigenvectors of R are the columns of $\tilde{V}_i$, and that the corresponding eigenvalues are $3w_1^2$ and then w_1^2 repeated $m-1$ times. Therefore we can write $Y = \tilde{U}_i O$, where O is an $m \times m$ orthogonal matrix. Then,

$$Z = VD^{1/2}Y = VD^{1/2}\tilde{U}_i O = \sqrt{\alpha_1}U_1 O$$

This proves the first statement that $\mathcal{Z}$, the span of the columns of Z, is the subspace $\mathcal{U}_1$, the span of columns of U_1. The second statement follows from equation (29) since the maximum value of the $m+1$-th eigenvalue is $3w_i^2$ for some $i \neq 1$. Hence,

$$\sigma_m(R) - \sigma_{m+1}(R) \geq w_1^2 - 3\max_{i \neq 1} w_i^2 \geq 3\delta w_1^2 = 3\delta\|U_1 v\|^2.$$

■

Lemma 28 *Let $\|\hat{A} - A\| < \epsilon < \sigma_{mk}(A)/4$. $A = VDV^T$ and $\hat{A} = \hat{V}\hat{D}\hat{V}^T$ be the eigen decompositions of $A, \hat{A}$. Let $\hat{W} = \hat{V}\hat{D}^{-1/2}$ be the whitening matrix. Then,*

$$\|I_k - (\hat{W}^T A\hat{W})^{-1/2}\| \leq \frac{4\epsilon}{\sigma_{mk}(A)}$$

Proof We prove this along the lines in Hsu and Kakade (2013). The matrix $\hat{W}$ whitens $\hat{A}$ since,

$$\hat{W}^T\hat{A}\hat{W} = \hat{D}^{-1/2}\hat{V}^T\hat{A}\hat{V}\hat{D}^{-1/2} = I_k$$

Also $\epsilon < \sigma_{mk}(A)/2$, hence using Weyl's inequality $\sigma_{mk}(\hat{A}) \geq \sigma_{mk}(A)/2$. This implies

$$\begin{aligned} \|I_k - \hat{W}^T A\hat{W}\| &= \|\hat{W}^T(\hat{A} - A)\hat{W}\| \leq \|\hat{W}\|^2\|\hat{A} - A\| \\ &< \frac{2\epsilon}{\sigma_{mk}(A)} \end{aligned}$$

Therefore all eigenvalues of the matrix $\hat{W}^T A\hat{W}$ lie in the interval $(1 - 2\epsilon/\sigma_{mk}(A), 1 + 2\epsilon/\sigma_{mk}(A))$. This implies the eigenvalues of $(\hat{W}^T A\hat{W})^{-1}$ lie in the interval $(1/(1 + 2\epsilon/\sigma_{mk}(A)), 1/(1 - 2\epsilon/\sigma_{mk}(A)))$. Then,

$$\begin{aligned}(I_k - (\hat{W}^T A\hat{W})^{-1/2})(I_k + (\hat{W}^T A\hat{W})^{-1/2}) &= I_k - (\hat{W}^T A\hat{W})^{-1} \\ I_k - (\hat{W}^T A\hat{W})^{-1/2} &= \left(I_k - (\hat{W}^T A\hat{W})^{-1}\right)(I_k + (\hat{W}^T A\hat{W})^{-1/2})^{-1} \\ \|I_k - (\hat{W}^T A\hat{W})^{-1/2}\| &\leq \|I_k - (\hat{W}^T A\hat{W})^{-1}\| \\ &\leq \frac{1}{1 - 2\epsilon/\sigma_{mk}(A)} - 1 \leq \frac{4\epsilon}{\sigma_{mk}(A)}\end{aligned}$$

■

Lemma 29 (Whitening matrix perturbation) *Assume $\|\hat{A} - A\| < \epsilon < \sigma_{mk}(A)/4$. Let $\hat{W} = \hat{V}\hat{D}^{-1/2}$ be the whitening matrix. Define $W := \hat{W}(\hat{W}^T A\hat{W})^{-1/2}$. Then,*

$$\|\hat{W} - W\| \leq \frac{8\epsilon}{\sigma_{mk}(A)^{3/2}}$$

Proof We note that the matrix W whitens the matrix A, since

$$W^T AW = (\hat{W}^T A\hat{W})^{-1/2}\hat{W}^T A\hat{W}(\hat{W}^T A\hat{W})^{-1/2} = I_k$$

We can bound the perturbation as follows.

$$\begin{aligned}\|\hat{W} - W\| &= \|\hat{W}(I_k - (\hat{W}^T A\hat{W})^{-1/2})\| \\ &\leq \|\hat{W}\|\|I_k - (\hat{W}^T A\hat{W})^{-1/2}\| \\ &\leq \frac{2}{\sqrt{\sigma_{mk}(A)}}\frac{4\epsilon}{\sigma_{mk}(A)} = \frac{8\epsilon}{\sigma_{mk}(A)^{3/2}}\end{aligned}$$

where the last inequality follows from Lemma 28. ■

Lemma 30 *Let $\max\{\|\hat{A} - A\|, \|\hat{B} - B\|\} < \epsilon$, and also let $\epsilon < \min\{\sigma_1(B)/2, \frac{\sigma_{mk}(A)}{16}\}$. $W = \hat{W}(\hat{W}^T A\hat{W})^{-1/2}$ be the whitening matrix. Define $R = W^T BW$ as the whitened B matrix, and $\hat{R} = \hat{W}^T\hat{B}\hat{W}$ is its estimate. Then,*

$$\|\hat{R} - R\| < \frac{51\sigma_1(B)\epsilon}{\sigma_{mk}(A)^2} := \epsilon_1$$

Proof From Lemma 29 we have $\|\hat{W} - W\| \leq \frac{8\epsilon}{\sigma_{mk}(A)^{3/2}} < \|\hat{W}\|/2$. Also we know $\|\hat{W}\| \leq \sqrt{2/\sigma_{mk}(A)}$. We obtain the required bound as follows.

$$\begin{aligned}\|\hat{R} - R\| &= \|\hat{W}^T\hat{B}\hat{W} - W^T BW\| \\ &\leq \|(\hat{W} - W)^T\hat{B}\hat{W}\| + \|W^T(\hat{B} - B)\hat{W}\| + \|W^T B(\hat{W} - W)\| \\ &\leq \frac{3}{2}\|\hat{W} - W\|\|B\|\|\hat{W}\| + \frac{3}{2}\|\hat{W}\|^2\|\hat{B} - B\| + \frac{3}{2}\|\hat{W}^T\|\|B\|\|\hat{W} - W\| \\ &= 3\|\hat{W} - W\|\|B\|\|\hat{W}\| + \frac{3}{2}\|\hat{W}\|^2\|\hat{B} - B\| \\ &< 48\frac{\sigma_1(B)\epsilon}{\sigma_{mk}(A)^2} + \frac{3\epsilon}{\sigma_{mk}(A)} < \frac{51\sigma_1(B)\epsilon}{\sigma_{mk}(A)^2}\end{aligned}$$

■

Lemma 31 *Suppose $Y = [u_1, \ldots, u_m]$ be the matrix of m largest eigenvectors of $R = W^T BW$, and $\hat{Y}$ be that of $\hat{R} = \hat{W}^T\hat{B}\hat{W}$. Let $\hat{Z} = \hat{V}\hat{D}^{1/2}\hat{Y}$. Then,*

$$\|\hat{Z}\hat{Z}^T - ZZ^T\| \leq C_1\frac{\sigma_1(A)\sigma_1(B)\epsilon}{(\sigma_m(R) - \sigma_{m+1}(R))\sigma_{mk}(A)^2}$$

where Z satisfies $Y = W^T Z$, and C_1 is a constant.

Proof First using Wedin's theorem for the matrix A and $\hat{A}$ we get

$$\|\hat{V}\hat{V}^T - VV^T\| < \frac{4\epsilon}{\sigma_{mk}(A)}. \tag{30}$$

From Lemma 30 we have $\|\hat{R} - R\| < \frac{51\sigma_1(B)\epsilon}{\sigma_{mk}(A)^2} = \epsilon_1$. Therefore we can again use Wedin's theorem on the matrices $R, \hat{R}$ to bound the perturbation of the subspace spanned by Y.

$$\begin{aligned}\|\hat{Y}\hat{Y}^T - YY^T\| &\leq \frac{4\|\hat{R} - R\|}{\sigma_m(R) - \sigma_{m+1}(R)} \\ &= \frac{4\epsilon_1}{\sigma_m(R) - \sigma_{m+1}(R)}.\end{aligned} \tag{31}$$

We now bound the following term.

$$\begin{aligned}\|\hat{V}\hat{D}^{1/2}W^T - \hat{V}\hat{V}^T\| &= \|\hat{V}\hat{D}^{1/2}(\hat{W}^T A\hat{W})^{-1/2}\hat{W}^T - \hat{V}\hat{V}^T\| \\ &= \|\hat{V}\hat{D}^{1/2}(\hat{W}^T A\hat{W})^{-1/2}\hat{D}^{-1/2}\hat{V}^T - \hat{V}\hat{V}^T\| \\ &\leq \|\hat{D}^{1/2}(\hat{W}^T A\hat{W})^{-1/2}\hat{D}^{-1/2} - I_k\| \\ &\leq \|\hat{D}^{1/2}\|\|(\hat{W}^T A\hat{W})^{-1/2} - I_k\|\|\hat{D}^{-1/2}\| \\ &\leq \sqrt{\frac{\sigma_1(\hat{A})}{\sigma_{mk}(\hat{A})}}\frac{4\epsilon}{\sigma_{mk}(A)} \leq \frac{8\sigma_1(A)^{1/2}\epsilon}{\sigma_{mk}(A)^{3/2}}\end{aligned} \tag{32}$$

where the second to last inequality follows from Lemma 28. Next we show that $\hat{Z}\hat{Z}^T$ is close to the projection of ZZ^T onto the subspace $\hat{V}\hat{V}^T$.

$$
\begin{aligned}
& \|\hat{Z}\hat{Z}^T - \hat{V}\hat{V}^T ZZ^T \hat{V}\hat{V}^T\| \\
= \; & \|\hat{V}\hat{D}^{1/2}\hat{Y}\hat{Y}^T\hat{D}^{1/2}\hat{V}^T - \hat{V}\hat{V}^T ZZ^T\hat{V}\hat{V}^T\| \\
\leq \; & \|\hat{V}\hat{D}^{1/2}(\hat{Y}\hat{Y}^T - YY^T)\hat{D}^{1/2}\hat{V}^T\| + \|\hat{V}\hat{D}^{1/2}YY^T\hat{D}^{1/2}\hat{V}^T - \hat{V}\hat{V}^T ZZ^T\hat{V}\hat{V}^T\| \\
\leq \; & \sigma_1(\hat{A})\|\hat{Y}\hat{Y}^T - YY^T\| + \|\hat{V}\hat{D}^{1/2}W^T ZZ^T W\hat{D}^{1/2}\hat{V}^T - \hat{V}\hat{V}^T ZZ^T\hat{V}\hat{V}^T\| \qquad (33)
\end{aligned}
$$

We bound the second term as follows. Observe that the matrix $D^{-1/2}V^T$ also whitens the matrix A. Therefore Z can be expressed as $Z = VD^{1/2}U'$ where U' is a matrix with orthonormal columns. This implies $\|ZZ^T\| = \|VD^{1/2}U'U'^T D^{1/2}V^T\| \leq \sigma_1(A)$.

$$
\begin{aligned}
& \|\hat{V}\hat{D}^{1/2}W^T ZZ^T W\hat{D}^{1/2}\hat{V}^T - \hat{V}\hat{V}^T ZZ^T\hat{V}\hat{V}^T\| \\
\leq \; & \|(\hat{V}\hat{D}^{1/2}W^T - \hat{V}\hat{V}^T)ZZ^T W\hat{D}^{1/2}\hat{V}^T\| + \|\hat{V}\hat{V}^T ZZ^T(W\hat{D}^{1/2}\hat{V}^T - \hat{V}\hat{V}^T)\| \\
\leq \; & \|(\hat{V}\hat{D}^{1/2}W^T - \hat{V}\hat{V}^T)ZY^T\hat{D}^{1/2}\hat{V}^T\| + \|ZZ^T\|\|W\hat{D}^{1/2}\hat{V}^T - \hat{V}\hat{V}^T\| \\
\leq \; & \|\hat{V}\hat{D}^{1/2}W^T - \hat{V}\hat{V}^T\|\|Z\|\|\hat{D}^{1/2}\| + \|ZZ^T\|\|W\hat{D}^{1/2}\hat{V}^T - \hat{V}\hat{V}^T\| \\
\leq \; & \frac{8\sigma_1(A)^{1/2}\epsilon}{\sigma_{mk}(A)^{3/2}} \times 2\sigma_1(A) + \sigma_1(A) \times \frac{8\sigma_1(A)^{1/2}\epsilon}{\sigma_{mk}(A)^{3/2}} \\
= \; & 24\frac{\sigma_1(A)^{3/2}\epsilon}{\sigma_{mk}(A)^{3/2}}
\end{aligned}
$$

The second to last step follows from equation 32. Now using the above bound in equation 33 we get,

$$
\begin{aligned}
\|\hat{Z}\hat{Z}^T - \hat{V}\hat{V}^T ZZ^T\hat{V}\hat{V}^T\| &\leq \sigma_1(\hat{A})\|\hat{Y}\hat{Y}^T - YY^T\| + 24\frac{\sigma_1(A)^{3/2}\epsilon}{\sigma_{mk}(A)^{3/2}} \\
&\leq \frac{8\sigma_1(A)\epsilon_1}{\sigma_m(R) - \sigma_{m+1}(R)} + 24\frac{\sigma_1(A)^{3/2}\epsilon}{\sigma_{mk}(A)^{3/2}} \qquad (34)
\end{aligned}
$$

where the last step follows from inequalities (31). We compute the required bound by combining equations (30) and (34) as follows.

$$
\begin{aligned}
\|\hat{Z}\hat{Z}^T - ZZ^T\| &= \|\hat{Z}\hat{Z}^T - VV^T ZZ^T VV^T\| \\
&\leq \|\hat{Z}\hat{Z}^T - \hat{V}\hat{V}^T ZZ^T\hat{V}\hat{V}^T\| + 3\|VV^T - \hat{V}\hat{V}^T\|\|ZZ^T\| \\
&\leq \frac{8\sigma_1(A)\epsilon_1}{\sigma_m(R) - \sigma_{m+1}(R)} + 24\frac{\sigma_1(A)^{3/2}\epsilon}{\sigma_{mk}(A)^{3/2}} + \frac{12\sigma_1(A)\epsilon}{\sigma_{mk}(A)} \\
&\leq C_1\frac{\sigma_1(A)\sigma_1(B)\epsilon}{(\sigma_m(R) - \sigma_{m+1}(R))\sigma_{mk}(A)^2}
\end{aligned}
$$

where C_1 is a constant. ■

F.1 Proof of Theorem 6

The proof follows from Theorem 27 and Lemma 31. Note that the matrix Z has all singular values equal to $\sqrt{\alpha_1}$, therefore ZZ^T has singular values α_1. Under the affinity condition from Theorem 27, we have

$$\sigma_m(R) - \sigma_{m+1}(R) \geq 3\delta\|U_1 v\|^2$$

Combining with Lemma 31 we get

$$\|\hat{Z}\hat{Z}^T - ZZ^T\| \leq \frac{C_2\sigma_1(A)\sigma_1(B)\epsilon}{\delta\|U_1 v\|^2\sigma_{mk}(A)^2}$$

where C_2 is a constant. Finally applying Wedin's theorem for the matrices $\hat{Z}\hat{Z}^T$ and ZZ^T, we have

$$\|\hat{U}\hat{U}^T - U_1 U_1^T\| \leq \frac{C_3\sigma_1(A)\sigma_1(B)\epsilon}{\alpha_1\delta\|U_1 v\|^2\sigma_{mk}(A)^2} \leq \frac{C\sigma_1(A)^2\epsilon}{\alpha_1\delta\sigma_{mk}(A)^2}$$

where $C_3 = 4C_2$.

Appendix G. Sample Complexity Analysis

Since the basic application of our method requires the estimation of certain covariance matrices, we need to show that one can estimate these matrices. There is a large literature on estimating covariance matrices, but for simplicity we will only focus on the simplest estimator: the sample covariance matrix. By well-known matrix concentration inequalities, one can show that the sample covariance matrix will be close to the covariance matrix with high probability if the sample size is large enough:

Theorem 32 *Tropp (2015) Let $A_1, \ldots, A_n$ be i.i.d. symmetric random $d \times d$ matrices. If $\|A_1\| \leq L$ a.s. then*

$$\Pr\left(\left\|\frac{1}{n}\sum_{i=1}^n A_i - \mathbb{E}A_i\right\| \geq t\right) \leq 8d\exp\left(-\frac{nt^2}{L^2}\right).$$

G.1 Truncation

Unfortunately, the matrices we will be dealing with do not usually have almost sure bounds on their norm. Here, we develop some straightforward truncation arguments in order to adapt Theorem 32.

Theorem 33 *Suppose that $A_1, \ldots, A_n$ are i.i.d. symmetric random $d \times d$ matrices satisfying the tail bound*

$$\Pr(\|A_1\| \geq t) \leq Ce^{-ct^\alpha}$$

for some $\alpha > 0$. Then for any $\epsilon, \delta > 0$, if $n \geq \tilde{\Omega}_\alpha(\epsilon^{-2}\log(d/\delta))$ then

$$\Pr(\|\hat{\mathbb{E}}A - \mathbb{E}A\| \geq \epsilon) \leq \delta,$$

where $\tilde{\Omega}_\alpha(k)$ means $C(\alpha)\Omega(k\log^{C(\alpha)}k)$.

Proof Fix $L > 0$ (to be determined later) and define the random matrix B_i by $B_i = A_i 1_{\{\|A_i\| \le L\}}$. Then Theorem 32 applies to B_i: if $n \ge \Omega(L^2\epsilon^{-2}\log(d/\delta))$ then

$$\Pr(\|\hat{\mathbb{E}}B - \mathbb{E}B\| \ge \epsilon) \le \delta.$$

To compare this with the similar quantity involving A, we will consider $\hat{\mathbb{E}}(A-B)$ and $\mathbb{E}(A-B)$ separately.

First, note that $\Pr(A_i \ne B_i) = \Pr(\|A\| \ge L) \le C\exp(-cL^\alpha)$. If $L = \Omega(\log^{1/\alpha}(n/\delta))$ then $\Pr(A_i \ne B_i) \le \delta/n$. By a union bound,

$$\Pr(\hat{\mathbb{E}}A \ne \hat{\mathbb{E}}B) \le \delta. \tag{35}$$

Now we fix $L = C'\log^{1/\alpha}(n/(\delta \vee \epsilon))$ and we consider $\|\mathbb{E}(A-B)\|$. By the triangle inequality,

$$\|\mathbb{E}(A-B)\| = \|\mathbb{E}A1_{\{\|A\|\ge L\}}\| \le \mathbb{E}\|A\|1_{\{\|A\|\ge L\}}.$$

On the other hand, we can bound

$$\mathbb{E}\|A\|1_{\{\|A\|\ge L\}} = \int_L^\infty \Pr(\|A\| \ge t)\,dt \le C\int_L^\infty e^{-ct^\alpha}\,dt.$$

With the change of variables $t = u^{1/\alpha}$, we have

$$\mathbb{E}\|A\|1_{\{\|A\|\ge L\}} \le \frac{1}{\alpha}\int_{L^\alpha}^\infty u^{1/\alpha}e^{-cu}\,du.$$

Now, if $u \ge C''\frac{1}{\alpha}\log\frac{1}{\alpha}$ for large enough C'' then $u^{1/\alpha}e^{-cu} \le e^{-cu/2}$. Hence, if $L^\alpha \ge C''\frac{1}{\alpha}\log\frac{1}{\alpha}$ then

$$\mathbb{E}\|A\|1_{\{\|A\|\ge L\}} \le \frac{1}{\alpha}\int_{L^\alpha}^\infty e^{-cu/2}\,du \le C(\alpha)e^{-cL^\alpha/2} \le C(\alpha)\epsilon$$

where the last inequality holds if the constant C' in the definition of L is large enough compared to c. On the other hand, if $L^\alpha < C''\frac{1}{\alpha}\log\frac{1}{\alpha}$ then we must have $\epsilon > c(\alpha)$ for some $c(\alpha) > 0$. In this case, $\mathbb{E}\|A\|1_{\{\|A\|\ge L\}} \le C \le C(\alpha)\epsilon$ trivially. To summarize, in every case we have

$$\|\mathbb{E}(A-B)\| \le C(\alpha)\epsilon.$$

Putting this together with (35), we have that if $n \ge \Omega(L^2\epsilon^{-2}\log(d/\delta))$ then with probability at least $1-2\delta$,

$$\begin{aligned}\|\hat{\mathbb{E}}A - \mathbb{E}A\| &\le \|\hat{\mathbb{E}}B - \mathbb{E}B\| + \|\hat{\mathbb{E}}A - \hat{\mathbb{E}}B\| + \|\mathbb{E}A - \mathbb{E}B\| \\ &\le (1 + C(\alpha))\epsilon.\end{aligned}$$

Finally, recalling that $L = \mathrm{polylog}(n, 1/\epsilon, 1/\delta)$ (with the polynomial depending on α), we see that $n = \tilde{\Omega}_\alpha(\epsilon^{-2}\log(d/\delta))$ suffices. Finally, we can absorb the constant $C(\alpha)$ into ϵ. ■

We will now show how Theorem 33 bounds the error in estimating the various matrices that we had to estimate for the various different models we considered. Essentially, we will repeatedly use the observation that if z is a standard Gaussian variable then $z^{2/\alpha}$ has a tail that decays like e^{-ct^α}. In other words, moments of Gaussians will naturally lead to a condition that the one assumed in Theorem 33.

G.2 Gaussian Mixture Model

For the following theorem, we revert to the notation of the Gaussian mixture model.

Theorem 34 *Fix $\epsilon, \delta > 0$. Let $\hat{A} = \hat{\mathbb{E}}[xx^T]$ and $\hat{B} = \hat{\mathbb{E}}[\langle x, v\rangle xx^T]$, where $\hat{\mathbb{E}}$ is taken with n i.i.d. samples. If $n \ge \tilde{\Omega}(d\epsilon^{-2}\log(d/\delta))$ then with probability at least $1-\delta$, $\|\hat{\mathbb{E}}A - \mathbb{E}A\| \le \epsilon$ and $\|\hat{\mathbb{E}}B - \mathbb{E}B\| \le \epsilon$.*

Proof To estimate A, first note that $\|xx^T\| = \|x\|^2$. Now, $\mathbb{E}\|x\|^2 \le R^2 + d\sigma^2$, where $R = \max_i \|\mu_i\|$, and also $\Pr(\|x\|^2 \ge \mathbb{E}\|x\|^2 + t\sqrt{d}) \le Ce^{-ct}$. Hence, we may apply Theorem 33 with $A_i = x_i x_i^T/\sqrt{d}$ and $\alpha = 1$; this yields the claimed bound on $\|\hat{\mathbb{E}}A - \mathbb{E}A\|$.

To estimate B, note that $\|\langle x, v\rangle^2 xx^T\| = \langle x, v\rangle^2\|x\|^2$. Now, the triangle inequailty implies that $\langle x, v\rangle^2\|x\|^2$ is stochastically dominated by

$$4R^4 + 4\mathbb{E}[\langle z, v\rangle^2\|z\|^2] = 4R^4 + 4\mathbb{E}[z_1^2\|z\|^2],$$

where z is a standard (i.e., centered) Gaussian vector. Then $\mathbb{E}[z_1^2\|z\|^2] = 2 + d$, and $z_1^2\|z\|^2$ has tails of order $e^{-ct^{1/2}}$; that is it satisfies the assumptions of Theorem 33 with $\alpha = 1/2$. Applying Theorem 33 with $A_i = \langle x_i, v\rangle^2 x_i x_i^T/\sqrt{d}$ then yields the claimed bound on $\|\hat{\mathbb{E}}B - \mathbb{E}B\|$. ■

G.3 LDA Topic Model

For the following theorem, we revert to the notation of the LDA topic model, where d is the size of the dictionary.

Theorem 35 *Fix $\epsilon, \delta > 0$. Let $\hat{A} = \hat{\mathbb{E}}[x_1 x_2^T]$ and $\hat{B} = \hat{E}[\langle x_3, v\rangle x_1 x_2^T]$, where $\hat{\mathbb{E}}$ is taken with n i.i.d. samples. If $n \ge \Omega(\epsilon^{-2}\log(d/\delta))$ then with probability at least $1-\delta$, $\|\hat{A} - \mathbb{E}A\| \le \epsilon$ and $\|\hat{B} - \mathbb{E}B\| \le \epsilon$.*

Proof We can apply Theorem 32 directly, since $\|x_1 x_2^T\| \le 1$ and $\langle x_3, v\rangle x_1 x_2^T \le 1$. ■

G.4 Mixed Regression

For the following theorem, we revert to the notation of the mixed regression model.

Theorem 36 *Fix $\epsilon, \delta > 0$. Let $\hat{A} = \hat{\mathbb{E}}[y^2 xx^T]$ and $\hat{B} = \hat{\mathbb{E}}[y^3\langle x, v\rangle xx^T]$, where $\hat{\mathbb{E}}$ is taken with n i.i.d. samples. Let $R = \max_i \|\mu_i\|$. If $n \ge \tilde{\Omega}((R^2 + \sigma^2)\epsilon^{-2}d\log(d/\delta))$ then with probability at least $1-\delta$, $\|\hat{A} - \mathbb{E}A\| \le \epsilon$ and $\|\hat{B} - \mathbb{E}B\| \le \epsilon$.*

Proof Recalling that in cluster i we have $y = \langle x, \mu_i\rangle + \xi$, we have

$$\|y^2 xx^T\| \le 2\langle x, \mu_i\rangle^2\|x\|^2 + 2\xi^2\|x\|^2.$$

Hence, $\mathbb{E}\|y^2 xx^T\| \le 2R^2(2+d) + \sigma^2 d$, with tails that decay at the rate $e^{-ct^{1/2}}$. Applying Theorem 33 implies the claimed bounds for A. The case of B is analogous, except that since it involves sixth moments the tails will decay at the rate $e^{-ct^{1/3}}$; this only effects the poly-logarithmic terms hidden in the $\tilde{\Omega}$ notation. ■

G.5 Subspace Clustering

For the following theorem, we revert to the notation of the subspace clustering model. We assume for simplicity that σ is known, since if it isn't then it can be easily and accurately learnt.

Theorem 37 *Fix $\epsilon, \delta > 0$. Let $\hat{A} = \hat{\mathbb{E}}[xx^T] - \sigma^2 I_d$ and*

$$\hat{B} = \hat{\mathbb{E}}[\langle x, v\rangle^2 xx^T] - \sigma^2(v^T\hat{A}v)I_d - \sigma^2\|v\|^2\hat{A} - \sigma^4(\|v\|^2 I_d + vv^T) - 2\sigma^2(\hat{A}vv^T + vv^T\hat{A})$$

where $\hat{\mathbb{E}}$ is taken with respect to n i.i.d. samples. If $n \geq \tilde{\Omega}(\epsilon^{-2}(1+\sigma^2)\|v\|^2 m\log(d/\delta))$ then with probability at least $1-\delta$, $\|\hat{A} - A\| \leq \epsilon$ and $\|\hat{B} - B\| \leq \epsilon$.

Proof Since x/σ is an m-dimensional Gaussian vector, $\|x\|^2/(\sigma^2 m)$ is concentrated around its mean (1) with tails of order e^{-ct}. In other words, Theorem 33 (with $\alpha = 1$) implies our claim for A. The claim for B is analogous, except that since it involves fourth moments, the tails will decay at the rate $e^{-ct^{1/2}}$. ■

References

Anima Anandkumar, Dean P Foster, Daniel J Hsu, Sham M Kakade, and Yi-Kai Liu. A spectral algorithm for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, pages 917–925, 2012.

Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *The Journal of Machine Learning Research*, 15(1):2773–2832, 2014.

Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5): 898–916, May 2011.

Sanjeev Arora, Rong Ge, Yonatan Halpern, David M Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu. A practical algorithm for topic modeling with provable guarantees. In *Proceedings of ICML-2013*, pages 280–288, 2013.

Sugato Basu, Arindam Banerjee, and Raymond Mooney. Semi-supervised clustering by seeding. In *Proceedings of 19th International Conference on Machine Learning (ICML-2002*, 2002.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3:993–1022, 2003.

Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors. *Semi-supervised learning*. MIT press Cambridge, 2006.

Yudong Chen, Xinyang Yi, and Constantine Caramanis. A convex formulation for mixed regression with two components: Minimax optimal rates. In *COLT*, pages 560–604, 2014.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

Ehsan Elhamifar and René Vidal. Sparse subspace clustering. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2790–2797. IEEE, 2009.

Moritz Hardt and Eric Price. Tight bounds for learning a mixture of two gaussians. In *Proceedings of ACM on Symposium on Theory of Computing, STOC*, pages 753–760, 2015.

Daniel Hsu and Sham M Kakade. Learning mixtures of spherical gaussians: moment methods and spectral decompositions. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 11–20. ACM, 2013.

Furong Huang, UN Niranjan, Mohammad Umar Hakeem, and Animashree Anandkumar. Online tensor methods for learning latent variable models. *Journal of Machine Learning Research*, 16:2797–2835, 2015.

Pirkko Kuusela and Daniel Ocone. Learning with side information: Pac learning bounds. *Journal of Computer and System Sciences*, 68(3):521–545, 2004.

Yue Lu and Chengxiang Zhai. Opinion integration through semi-supervised topic modeling. In *Proceedings of the 17th International Conference on World Wide Web*, pages 121–130. ACM, 2008.

Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

Jon D Mcauliffe and David M Blei. Supervised topic models. In *Advances in neural information processing systems*, pages 121–128, 2008.

Ankur Moitra and Gregory Valiant. Settling the polynomial learnability of mixtures of gaussians. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 93–102. IEEE, 2010.

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conf. of the North American Chapter of the Association for Computational Linguistics*, pages 100–108. Association for Computational Linguistics, 2010.

David Newman, Edwin V Bonilla, and Wray Buntine. Improving topic coherence with regularized topic models. In *Advances in neural information processing systems*, pages 496–504, 2011.

Dohyung Park, Constantine Caramanis, and Sujay Sanghavi. Greedy subspace clustering. In *Advances in Neural Information Processing Systems*, pages 2753–2761, 2014.

Karl Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, pages 71–110, 1894.

Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proc. of the 2009 Conf. on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 248–256. Association for Computational Linguistics, 2009.

Richard A Redner and Homer F Walker. Mixture densities, maximum likelihood and the em algorithm. *SIAM review*, 26(2):195–239, 1984.

M. Röder, A. Both, and A. Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408. ACM, 2015.

Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. In *Proceedings of the 20th conference on Uncertainty in Artificial Intelligence*, pages 487–494, 2004.

Hanie Sedghi, Majid Janzamin, and Anima Anandkumar. Provable tensor methods for learning mixtures of generalized linear models. In *Proceedings of International Conference on Artificial Intelligence and Statistics, AISTATS 2016*, pages 1223–1231, 2016.

Mahdi Soltanolkotabi and Emmanuel J Candes. A geometric analysis of subspace clustering with outliers. *The Annals of Statistics*, pages 2195–2238, 2012.

TACC. Texas advanced computing center, 2018. `http://www.tacc.utexas.edu`.

Joel Tropp. An introduction to matrix concentration inequalities. *arXiv preprint arXiv:1501.01571*, 2015.

UCI. NY Times dataset, 2008. `http://mlr.cs.umass.edu/ml/machine-learning-databases/`.

Eric P Xing, Michael I Jordan, Stuart Russell, and Andrew Y Ng. Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems*, pages 505–512, 2002.

Tianbao Yang, Rong Jin, and Anil K Jain. Learning from noisy side information by generalized maximum entropy model. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 1199–1206, 2010.

Yelp. Yelp dataset, 2014. `http://www.yelp.com/dataset_challenge/`.

Xinyang Yi, Constantine Caramanis, and Sujay Sanghavi. Alternating minimization for mixed linear regression. In *Proceedings of International Conference on Machine Learning, ICML 2014*, pages 613–621, 2014.

Journal of Machine Learning Research 18 (2018) 1-42 Submitted 3/16; Revised 5/17; Published 4/18

An ℓ_∞ Eigenvector Perturbation Bound and Its Application to Robust Covariance Estimation

Jianqing Fan JQFAN@PRINCETON.EDU
Weichen Wang NICKWEICHWANG@GMAIL.COM
Yiqiao Zhong YIQIAOZ@PRINCETON.EDU
Department of Operations Research and Financial Engineering
Princeton University
Princeton, NJ 08544, USA

Editor: Nicolai Meinshausen

Abstract

In statistics and machine learning, we are interested in the eigenvectors (or singular vectors) of certain matrices (e.g. covariance matrices, data matrices, etc). However, those matrices are usually perturbed by noises or statistical errors, either from random sampling or structural patterns. The Davis-Kahan $\sin\theta$ theorem is often used to bound the difference between the eigenvectors of a matrix A and those of a perturbed matrix $\widetilde{A} = A + E$, in terms of ℓ_2 norm. In this paper, we prove that when A is a low-rank and incoherent matrix, the ℓ_∞ norm perturbation bound of singular vectors (or eigenvectors in the symmetric case) is smaller by a factor of $\sqrt{d_1}$ or $\sqrt{d_2}$ for left and right vectors, where d_1 and d_2 are the matrix dimensions. The power of this new perturbation result is shown in robust covariance estimation, particularly when random variables have heavy tails. There, we propose new robust covariance estimators and establish their asymptotic properties using the newly developed perturbation bound. Our theoretical results are verified through extensive numerical experiments.

Keywords: Matrix perturbation theory, Incoherence, Low-rank matrices, Sparsity, Approximate factor model

1. Introduction

The perturbation of matrix eigenvectors (or singular vectors) has been well studied in matrix perturbation theory (Wedin, 1972; Stewart, 1990). The best known result of eigenvector perturbation is the classic Davis-Kahan theorem (Davis and Kahan, 1970). It originally emerged as a powerful tool in numerical analysis, but soon found its widespread use in other fields, such as statistics and machine learning. Its popularity continues to surge in recent years, which is largely attributed to the omnipresent data analysis, where it is a common practice, for example, to employ PCA (Jolliffe, 2002) for dimension reduction, feature extraction, and data visualization.

The eigenvectors of matrices are closely related to the underlying structure in a variety of problems. For instance, principal components often capture most information of data and extract the latent factors that drive the correlation structure of the data (Bartholomew et al., 2011); in classical multidimensional scaling (MDS), the centered squared distance matrix encodes the coordinates of data points embedded in a low dimensional subspace (Borg and Groenen, 2005); and in clustering and network analysis, spectral algorithms are used to reveal clusters and community structure (Ng et al., 2002; Rohe et al., 2011). In those problems, the low dimensional structure that we want to recover, is often 'perturbed' by observation uncertainty or statistical errors. Besides, there might be a sparse pattern corrupting the low dimensional structure, as in approximate factor models (Chamberlain et al., 1983; Stock and Watson, 2002) and robust PCA (De La Torre and Black, 2003; Candès et al., 2011).

A general way to study these problems is to consider

$$\widetilde{A} = A + S + N, \tag{1}$$

where A is a low rank matrix, S is a sparse matrix, and N is a random matrix regarded as random noise or estimation error, all of which have the same size $d_1 \times d_2$. Usually A is regarded as the 'signal' matrix we are primarily interested in, S is some sparse contamination whose effect we want to separate from A, and N is the noise (or estimation error in covariance matrix estimation).

The decomposition (1) forms the core of a flourishing literature on robust PCA (Chandrasekaran et al., 2011; Candès et al., 2011), structured covariance estimation (Fan et al., 2008, 2013), multivariate regression (Yuan et al., 2007) and so on. Among these works, a standard condition on A is matrix incoherence (Candès et al., 2011). Let the singular value decomposition be

$$A = U\Sigma V^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T, \tag{2}$$

where r is the rank of A, the singular values are $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$, and the matrices $U = [u_1, \ldots, u_r] \in \mathbb{R}^{d_1 \times r}$, $V = [v_1, \ldots, v_r] \in \mathbb{R}^{d_2 \times r}$ consist of the singular vectors. The coherences $\mu(U), \mu(V)$ are defined as

$$\mu(U) = \frac{d_1}{r} \max_i \sum_{j=1}^{r} U_{ij}^2, \qquad \mu(V) = \frac{d_2}{r} \max_i \sum_{j=1}^{r} V_{ij}^2, \tag{3}$$

where U_{ij} and V_{ij} are the (i,j) entry of U and V, respectively. It is usually expected that $\mu_0 := \max\{\mu(U), \mu(V)\}$ is not too large, which means the singular vectors u_i and v_i are incoherent with the standard basis. This incoherence condition (3) is necessary for us to separate the sparse component S from the low rank component A; otherwise A and S are not identifiable. Note that we do not need any incoherence condition on UV^T, which is different from Candès et al. (2011) and is arguably unnecessary (Chen, 2015).

Now we denote the eigengap $\gamma_0 = \min\{\sigma_i - \sigma_{i+1} : i = 1, \ldots, r\}$ where $\sigma_{r+1} := 0$ for notational convenience. Also we let $E = S + N$, and view it as a perturbation matrix to the matrix A in (1). To quantify the perturbation, we define a rescaled measure as $\tau_0 := \max\{\sqrt{d_2/d_1}\|E\|_1, \sqrt{d_1/d_2}\|E\|_\infty\}$, where

$$\|E\|_1 = \max_j \sum_{i=1}^{d_1} |E_{ij}|, \quad \|E\|_\infty = \max_i \sum_{j=1}^{d_2} |E_{ij}|, \tag{4}$$

which are commonly used norms gauging sparsity (Bickel and Levina, 2008). They are also operator norms in suitable spaces (see Section 2). The rescaled norms $\sqrt{d_2/d_1}\|E\|_1$ and $\sqrt{d_1/d_2}\|E\|_\infty$ are comparable to the spectral norm $\|E\|_2 := \max_{\|u\|_2=1}\|Eu\|_2$ in many cases; for example, when E is an all-one matrix, $\sqrt{d_2/d_1}\|E\|_1 = \sqrt{d_1/d_2}\|E\|_\infty = \|E\|_2$.

Suppose the perturbed matrix $\widetilde{A}$ also has the singular value decomposition:

$$\widetilde{A} = \sum_{i=1}^{d_1\wedge d_2} \widetilde{\sigma}_i \widetilde{u}_i \widetilde{v}_i^T, \tag{5}$$

where $\widetilde{\sigma}_i$ are nonnegative and in the decreasing order, and the notation $\wedge$ means $a \wedge b = \min\{a, b\}$. Denote $\widetilde{U} = [\widetilde{u}_1, \ldots, \widetilde{u}_r], V = [\widetilde{v}_1, \ldots, \widetilde{v}_r]$, which are counterparts of top r singular vectors of A.

We will present an ℓ_∞ matrix perturbation result that bounds $\|\widetilde{u}_i - u_i\|_\infty$ and $\|\widetilde{v}_i - v_i\|_\infty$ up to sign.[1] This result is different from ℓ_2 bounds, Frobenius-norm bounds, or the $\sin\Theta$ bounds, as the ℓ_∞ norm is not orthogonal invariant. The following theorem is a simplified version of our main results in Section 2.

Theorem 1 *Let $\widetilde{A} = A + E$ and suppose the singular decomposition in (2) and (5). Denote $\gamma_0 = \min\{\sigma_i - \sigma_{i+1} : i = 1, \ldots, r\}$ where $\sigma_{r+1} := 0$. Then there exists $C(r, \mu_0) = O(r^4\mu_0^2)$ such that, if $\gamma_0 > C(r, \mu_0)\tau_0$, up to sign,*

$$\max_{1\le i\le r}\|\widetilde{u}_i - u_i\|_\infty \le C(r,\mu_0)\frac{\tau_0}{\gamma_0\sqrt{d_1}} \quad \text{and} \quad \max_{1\le i\le r}\|\widetilde{v}_i - v_i\|_\infty \le C(r,\mu_0)\frac{\tau_0}{\gamma_0\sqrt{d_2}}, \tag{6}$$

where $\mu_0 = \max\{\mu(U), \mu(V)\}$ is the coherence given after (3) and $\tau_0 := \max\{\sqrt{d_2/d_1}\|E\|_1, \sqrt{d_1/d_2}\|E\|_\infty\}$.

When A is symmetric, $\tau_0 = \|E\|_\infty$ and the condition on the eigengap is simply $\gamma_0 > C(r, \mu_0)\|E\|_\infty$. The incoherence condition naturally holds for a variety of applications, where the low rank structure emerges as a consequence of a few factors driving the data matrix. For example, in Fama-French factor models, the excess returns in a stock market are driven by a few common factors (Fama and French, 1993); in collaborative filtering, the ratings of users are mostly determined by a few common preferences (Rennie and Srebro, 2005); in video surveillance, A is associated with the stationary background across image frames (Oliver et al., 2000). We will have a detailed discussion in Section 2.3.

The eigenvector perturbation was studied by Davis and Kahan (1970), where Hermitian matrices were considered, and the results were extended by Wedin (1972) to general rectangular matrices. To compare our result with these classical results, assuming $\gamma_0 \ge 2\|E\|_2$, a combination of Wedin's theorem and Mirsky's inequality (Mirsky, 1960) (the counterpart of Weyl's inequality for singular values) implies

$$\max_{1\le k\le r}\left\{\|v_k - \widetilde{v}_k\|_2 \vee \|u_k - \widetilde{u}_k\|_2\right\} \le \frac{2\sqrt{2}\|E\|_2}{\gamma_0}. \tag{7}$$

where $a \vee b := \max\{a, b\}$.

1. 'Up to sign' means we can appropriately choose an eigenvector or singular vector u to be either u or $-u$ in the bounds. This is becuase eigenvectors and singular vectors are not unique.

Yu et al. (2015) also proved a similar bound as in (7), and that result is more convenient to use. If we are interested in the ℓ_∞ bound but naively use the trivial inequality $\|x\|_\infty \le \|x\|_2$, we would have a suboptimal bound $O(\|E\|_2/\gamma_0)$ in many situations, especially in cases where $\|E\|_2$ is comparable to $\|E\|_\infty$. Compared with (6), the bound is worse by a factor of $\sqrt{d_1}$ for u_k and $\sqrt{d_2}$ for v_k. In other words, converting the ℓ_2 bound from Davis-Kahan theorem directly to the ℓ_∞ bound does not give a sharp result in general, in the presence of incoherent and low rank structure of A. Actually, assuming $\|E\|_2$ is comparable with $\|E\|_\infty$, for square matrices, our ℓ_∞ bound (6) matches the ℓ_2 bound (7) in terms of dimensions d_1 and d_2. This is because $\|x\|_2 \le \sqrt{n}\,\|x\|_\infty$ for any $x \in \mathbb{R}^n$, so we expect to gain a factor $\sqrt{d_1}$ or $\sqrt{d_2}$ in those ℓ_∞ bounds. The intuition is that, when A has an incoherent and low-rank structure, the perturbation of singular vectors is not concentrated on a few coordinates.

To understand how matrix incoherence helps, let us consider a simple example with no matrix incoherence, in which (7) is tight up to a constant. Let $A = d(1, 0, \ldots, 0)^T(1, 0, \ldots, 0)$ be a d-dimensional square matrix, and $E = d(0, 1/2, 0, \ldots, 0)^T(1, 0, \ldots, 0)$ of the same size. It is apparent that $\gamma_0 = d, \tau_0 = d/2$, and that $v_1 = (1, 0, \ldots, 0)^T, \widetilde{v}_1 = (2/\sqrt{5}, 1/\sqrt{5}, 0, \ldots, 0)^T$ up to sign. Clearly, the perturbation $\|\widetilde{v}_1 - v_1\|_\infty$ is not vanishing as d tends to infinity in this example, and thus, there is no hope of a strong upper bound as in (6) without the incoherence condition.

The reason that the factor $\sqrt{d_1}$ or $\sqrt{d_2}$ comes into play in (7) is that, the error $u_k - \widetilde{u}_k$ (and similarly for v_k) spreads out evenly in d_1 (or d_2) coordinates, so that the ℓ_∞ error is far smaller than the ℓ_2 error. This, of course, hinges on the incoherence condition, which in essence precludes eigenvectors from aligning with any coordinate.

Our result is very different from the sparse PCA literature, in which it is usually assumed that the leading eigenvectors are sparse. In Johnstone and Lu (2009), it is proved that there is a threshold for p/n (the ratio between the dimension and the sample size), above which PCA performs poorly, in the sense that $\langle\widetilde{v}_1, v_1\rangle$ is approximately 0. This means that the principal component computed from the sample covariance matrix reveals nothing about the true eigenvector. In order to mitigate this issue, in Johnstone and Lu (2009) and subsequent papers (Vu and Lei, 2013; Ma, 2013; Berthet and Rigollet, 2013), sparse leading eigenvectors are assumed. However, our result is different, in the sense that we require a stronger eigengap condition $\gamma_0 > C(r, \mu_0)\|E\|_\infty$ (i.e. stronger signal), whereas in Johnstone and Lu (2009), the eigengap of the leading eigenvectors is a constant times $\|E\|_2$. This explains why it is plausible to have a strong uniform eigenvector perturbation bound in this paper.

We will illustrate the power of this perturbation result using robust covariance estimation as one application. In the approximate factor model, the true covariance matrix admits a decomposition into a low rank part A and a sparse part S. Such models have been widely applied in finance, economics, genomics, and health to explore correlation structure.

However, in many studies, especially financial and genomics applications, it is well known that the observations exhibit heavy tails (Gupta et al., 2013). This problem can be resolved with the aid of recent results of concentration bounds in robust estimation (Catoni, 2012; Hsu and Sabato, 2014; Fan et al., 2017a), which produces the estimation error N in (1) with an optimal entry-wise bound. It nicely fits our perturbation results, and we can tackle it easily by following the ideas in Fan et al. (2013).

Here are a few notations in this paper. For a generic d_1 by d_2 matrix, the matrix max-norm is denoted as $\|M\|_{\max} = \max_{i,j} |M_{ij}|$. The matrix operator norm induced by vector ℓ_p norm is $\|M\|_p = \sup_{\|x\|_p=1} \|Mx\|_p$ for $1 \le p \le \infty$. In particular, $\|M\|_1 = \max_j \sum_{i=1}^{d_1} |M_{ij}|$; $\|M\|_\infty = \max_i \sum_{j=1}^{d_2} |M_{ij}|$; and $\|\cdot\|$ denotes the spectral norm, or the matrix 2-norm $\|\cdot\|_2$ for simplicity. We use $\sigma_j(M)$ to denote the j^{th} largest singular value. For a symmetric matrix M, denote $\lambda_j(M)$ as its j^{th} largest eigenvalue. If M is a positive definite matrix, then $M^{1/2}$ is the square root of M, and $M^{-1/2}$ is the square root of M^{-1}.

2. The ℓ_∞ perturbation results

2.1 Symmetric matrices

First, we study ℓ_∞ perturbation for symmetric matrices (so $d_1 = d_2$). The approach we study symmetric matrices will be useful to analyze asymmetric matrices, because we can always augment a $d_1 \times d_2$ rectangular matrix into a $(d_1+d_2)\times(d_1+d_2)$ symmetric matrix, and transfer the study of singular vectors to the eigenvectors of the augmented matrix. This augmentation is called *Hermitian dilation*. (Tropp, 2012; Paulsen, 2002)

Suppose that $A \in \mathbb{R}^{d\times d}$ is an d-dimensional symmetric matrix. The perturbation matrix $E \in \mathbb{R}^{d\times d}$ is also d-dimensional and symmetric. Let the perturbed matrix be $\widetilde{A} := A + E$. Suppose the spectral decomposition of A is given by

$$A = [V, V_\perp] \begin{pmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{pmatrix} [V, V_\perp]^T = \sum_{i=1}^{r} \lambda_i v_i v_i^T + \sum_{i>r} \lambda_i v_i v_i^T, \tag{8}$$

where $\Lambda_1 = \mathrm{diag}\{\lambda_1, \ldots, \lambda_r\}$, $\Lambda_2 = \mathrm{diag}\{\lambda_{r+1}, \ldots, \lambda_n\}$, and where $|\lambda_1| \ge |\lambda_2| \ge \ldots \ge |\lambda_n|$. Note the best rank-$r$ approximation of A under the Frobenius norm is $A_r := \sum_{i\le r} \lambda_i v_i v_i^T$.[2] Analogously, the spectral decomposition of $\widetilde{A}$ is

$$\widetilde{A} = \sum_{i=1}^{r} \widetilde{\lambda}_i \widetilde{v}_i \widetilde{v}_i^T + \sum_{i>r} \widetilde{\lambda}_i \widetilde{v}_i \widetilde{v}_i^T,$$

and write $\widetilde{V} = [\widetilde{v}_1, \ldots, \widetilde{v}_r] \in \mathbb{R}^{d\times r}$, where $|\widetilde{\lambda}_1| \ge |\widetilde{\lambda}_2| \ge \ldots \ge |\widetilde{\lambda}_n|$. Recall that $\|E\|_\infty$ given by (4) is an operator norm in the ℓ_∞ space, in the sense that $\|E\|_\infty = \sup_{\|u\|_\infty \le 1} \|Eu\|_\infty$. This norm is the natural counterpart of the spectral norm $\|E\|_2 := \sup_{\|u\|_2\le 1} \|Eu\|_2$.

We will use notations $O(\cdot)$ and $\Omega(\cdot)$ to hide absolute constants.[3] The next theorem bounds the perturbation of eigenspaces up to a rotation.

Theorem 2 *Suppose $|\lambda_r| - \varepsilon = \Omega(r^3\mu^2\|E\|_\infty)$, where $\varepsilon = \|A - A_r\|_\infty$, which is the approximation error measured under the matrix ∞-norm and $\mu = \mu(V)$ is the coherence of V defined in (3). Then, there exists an orthogonal matrix $R \in \mathbb{R}^{r\times r}$ such that*

$$\|\widetilde{V}R - V\|_{\max} = O\left(\frac{r^{5/2}\mu^2\|E\|_\infty}{(|\lambda_r| - \varepsilon)\sqrt{d}}\right).$$

This result involves an unspecified rotation R, due to the possible presence of multiplicity of eigenvalues. In the case where $\lambda_1 = \cdots = \lambda_r > 0$, the individual eigenvectors of V are only identifiable up to rotation. However, assuming an eigengap (similar to Davis-Kahan theorem), we are able to bound the perturbation of individual eigenvectors (up to sign).

Theorem 3 *Assume the conditions in Theorem 2. In addition, suppose δ satisfies $\delta > \|E\|_2$, and for any $i \in [r]$, the interval $[\lambda_i - \delta, \lambda_i + \delta]$ does not contain any eigenvalues of A other than λ_i. Then, up to sign,*

$$\max_{i\in[r]} \|\widetilde{v}_i - v_i\|_\infty = \|\widetilde{V} - V\|_{\max} = O\left(\frac{r^4\mu^2\|E\|_\infty}{(|\lambda_r| - \varepsilon)\sqrt{d}} + \frac{r^{3/2}\mu^{1/2}\|E\|_2}{\delta\sqrt{d}}\right).$$

To understand the above two theorems, let us consider the case where A has exactly rank r (i.e., $\varepsilon = 0$), and r and μ are not large (say, bounded by a constant). Theorem 2 gives a uniform entrywise bound $O(\|E\|_\infty/|\lambda_r|\sqrt{d})$ on the eigenvector perturbation. As a comparison, the Davis–Kahan $\sin\Theta$ theorem (Davis and Kahan, 1970) gives a bound $O(\|E\|_2/|\lambda_r|)$ on $\|\widetilde{V}R - V\|_2$ with suitably chosen rotation R.[4] This is an order of $\sqrt{d}$ larger than the bound given in Theorem 2 when $\|E\|_\infty$ is of the same order as $\|E\|_2$. Thus, in scenarios where $\|E\|_2$ is comparable to $\|E\|_\infty$, this is a refinement of Davis-Kahan theorem, because the max-norm bound in Theorem 2 provides an entry-wise control of perturbation. Although $\|E\|_\infty \ge \|E\|_2$,[5] there are many settings where the two quantities are comparable; for example, if E has a submatrix whose entries are identical and has zero entries otherwise, then $\|E\|_\infty = \|E\|_2$.

Theorem 3 provides the perturbation of individual eigenvectors, under a usual eigengap assumption. When r and μ are not large, we incur an additional term $O(\|E\|_2/\delta\sqrt{d})$ in the bound. This is understandable, since $\|\widetilde{v}_i - v_i\|_2$ is typically $O(\|E\|_2/\delta)$.

When the rank of A is not exactly r, we require that $|\lambda_r|$ is larger than the approximation error $\|A - A_r\|_\infty$. It is important to state that this assumption is more restricted than the eigengap assumption in the Davis-Kahan theorem, since $\|A - A_r\|_\infty \ge \|A - A_r\|_2 = |\lambda_{r+1}|$. However, different from the matrix max-norm, the spectral norm $\|\cdot\|_2$ only depends on the eigenvalues of a matrix, so it is natural to expect ℓ_2 perturbation bounds that only involve λ_r and λ_{r+1}. It is not clear whether we should expect an ℓ_∞ bound that involves λ_{r+1} instead of ε. More discussions can be found in Section 5.

We do not pursue the optimal bound in terms of r and $\mu(V)$ in this paper, as the two quantities are not large in many applications, and the current proof is already complicated.

2. This is a consequence of Wielandt-Hoffman theorem.
3. We write $a = O(b)$ if there is a constant $C > 0$ such that $a < Cb$; and $a = \Omega(b)$ if there is a constant $C' > 0$ such that $a > C'b$.
4. To see how the Davis-Kahan $\sin\Theta$ theorem relates to this form, we can use the identity $\|\sin\Theta(\widetilde{V}, V)\|_2 = \|\widetilde{V}\widetilde{V}^T - VV^T\|_2$ (Stewart, 1990), and the (easily verifiable) inequality $2\min_R \|\widetilde{V}R - V\|_2 \ge \|\widetilde{V}\widetilde{V}^T - VV^T\|_2 \ge \min_R \|\widetilde{V}R - V\|_2$ where R is an orthogonal matrix.
5. Since $\|E\|_1\|E\|_\infty \ge \|E\|_2^2$ (Stewart, 1990), the inequality follows from $\|E\|_1 = \|E\|_\infty$ by symmetry.

2.2 Rectangular matrices

Now we establish ℓ_∞ perturbation bounds for general rectangular matrices. The results here are more general than those in Section 1, and in particular, we allow the matrix A to be of approximate low rank. Suppose that both A and E are $d_1 \times d_2$ matrices, and $\widetilde{A} := A + E$. The rank of A is at most $d_1 \wedge d_2$ (where $a \wedge b = \min\{a, b\}$). Suppose an integer r satisfies $r \le \mathrm{rank}(A)$. Let the singular value decomposition of A be

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T + \sum_{i=r+1}^{d_1 \wedge d_2} \sigma_i u_i v_i^T,$$

where the singular values are ordered as $\sigma_1 \ge \sigma_2 \ge \ldots \ge \sigma_{d_1 \wedge d_2} \ge 0$, and the unit vectors $u_1, \ldots, u_{d_1 \wedge d_2}$ (or unit vectors $v_1, \ldots, v_{d_1 \wedge d_2}$) are orthogonal to each other. We denote $U = [u_1, \ldots, u_r] \in \mathbb{R}^{d_1 \times r}$ and $V = [v_1, \ldots, v_r] \in \mathbb{R}^{d_2 \times r}$. Analogously, the singular value decomposition of $\widetilde{A}$ is

$$\widetilde{A} = \sum_{i=1}^{r} \widetilde{\sigma}_i \widetilde{u}_i \widetilde{v}_i^T + \sum_{i=r+1}^{d_1 \wedge d_2} \widetilde{\sigma}_i \widetilde{u}_i \widetilde{v}_i^T,$$

where $\widetilde{\sigma}_1 \ge \ldots \ge \widetilde{\sigma}_{d_1 \wedge d_2}$. Similarly, columns of $\widetilde{U} = [\widetilde{u}_1, \ldots, \widetilde{u}_r] \in \mathbb{R}^{d_1 \times r}$ and $\widetilde{V} = [\widetilde{v}_1, \ldots, \widetilde{v}_r] \in \mathbb{R}^{d_2 \times r}$ are orthonormal.

Define $\mu_0 = \max\{\mu(V), \mu(U)\}$, where $\mu(U)$ (resp. $\mu(V)$) is the coherence of U (resp. V). This μ_0 will appear in the statement of our results, as it controls both the structure of left and right singular spaces. When, specially, A is a symmetric matrix, the spectral decomposition of A is also the singular value decomposition (up to sign), and thus μ_0 coincides with μ defined in Section 2.1.

Recall the definition of matrix ∞-norm and 1-norm of a rectangular matrix (4). Similar to the matrix ∞-norm, $\|\cdot\|_1$ is an operator norm in the ℓ_1 space. An obvious relationship between matrix ∞-norm and 1-norm is $\|E\|_\infty = \|E^T\|_1$. Note that the matrix ∞-norm and 1-norm have different number of summands in their definitions, so we are motivated to consider $\tau_0 := \max\{\sqrt{d_1/d_2}\|E\|_\infty, \sqrt{d_2/d_1}\|E\|_1\}$ to balance the dimensions d_1 and d_2.

Let $A_r = \sum_{i \le r} \sigma_i u_i v_i^T$ be the best rank-r approximation of A under the Frobenius norm, and let $\varepsilon_0 = \sqrt{d_1/d_2}\|A - A_r\|_\infty \vee \sqrt{d_2/d_1}\|A - A_r\|_1$, which also balances the two dimensions. Note that in the special case where A is symmetric, this approximation error ε_0 is identical to ε defined in Section 2.1. The next theorem bounds the perturbation of singular spaces.

Theorem 4 *Suppose that $\delta_0 - \varepsilon_0 = \Omega(r^3 \mu_0^2 \tau_0)$. Then, there exists orthogonal matrices $R_U, R_V \in \mathbb{R}^{r \times r}$ such that,*

$$\|\widetilde{U} R_U - U\|_{\max} = O\Big(\frac{r^{5/2}\mu_0^2\tau_0}{(\sigma_r - \epsilon_0)\sqrt{d_1}}\Big), \qquad \|\widetilde{V} R_V - V\|_{\max} = O\Big(\frac{r^{5/2}\mu_0^2\tau_0}{(\sigma_r - \epsilon_0)\sqrt{d_2}}\Big).$$

Similar to Theorem 3, under an assumption of gaps between singular values, the next theorem bounds the perturbation of individual singular vectors.

Theorem 5 *Suppose the same assumptions in Theorem 4 hold. In addition, suppose δ_0 satisfies $\delta_0 > \|E\|_2$, and for any $i \in [r]$, the interval $[\sigma_i - \delta_0, \sigma_i + \delta_0]$ does not contain any eigenvalues of A other than σ_i. Then, up to sign,*

$$\max_{i \in [r]} \|\widetilde{u}_i - u_i\|_\infty = O\Big(\frac{r^4\mu_0^2\tau_0}{(\sigma_r - \varepsilon_0)\sqrt{d_1}} + \frac{r^{3/2}\mu_0^{1/2}\|E\|_2}{\delta_0\sqrt{d_1}}\Big), \tag{9}$$

$$\max_{i \in [r]} \|\widetilde{v}_i - v_i\|_\infty = O\Big(\frac{r^4\mu_0^2\tau_0}{(\sigma_r - \varepsilon_0)\sqrt{d_2}} + \frac{r^{3/2}\mu_0^{1/2}\|E\|_2}{\delta_0\sqrt{d_2}}\Big). \tag{10}$$

As mentioned in the beginning of this section, we will use dilation to augment all $d_1 \times d_2$ matrices into symmetric ones with size $d_1 + d_2$. In order to balance the possibly different scales of d_1 and d_2, we consider a weighted max-norm. This idea will be further illustrated in Section 5.

2.3 Examples: which matrices have such structure?

In many problems, low-rank structure naturally arises due to the impact of pervasive latent factors that influence most observed data. Since observations are imperfect, the low-rank structure is often 'perturbed' by an additional sparse structure, gross errors, measurement noises, or the idiosyncratic components that can not be captured by the latent factors. We give some motivating examples with such structure.

Panel data in stock markets. Consider the excess returns from a stock market over a period of time. The driving factors in the market are reflected in the covariance matrix as a low rank component A. The residual covariance of the idiosyncratic components is often modeled by a sparse component S. Statistical analysis including PCA is usually conducted based on the estimated covariance matrix $\widetilde{A} = \widehat{\Sigma}$, which is perturbed from the true covariance $\Sigma = A + S$ by the estimation error N (Stock and Watson, 2002; Fan et al., 2013). In Section 3.1, we will develop a robust estimation method in the presence of heavy-tailed return data.

Video surveillance. In image processing and computer vision, it is often desired to separate moving objects from static background before further modeling and analysis (Oliver et al., 2000; Hu et al., 2004). The static background corresponds to the low rank component A in the data matrix, which is a collection of video frames, each consisting of many pixels represented as a long vector in the data matrix. Moving objects and noise correspond to the sparse matrix S and noise matrix N. Since the background is global information and reflected by many pixels of a frame, it is natural for the incoherence condition to hold.

Wireless sensor network localization. In wireless sensor networks, we are usually interested in determining the location of sensor nodes with unknown position based on a few (noisy) measurements between neighboring nodes (Doherty et al., 2001; Biswas and Ye, 2004). Let $\mathbb{X}$ be an r by n matrix such that each column x_i gives the coordinates of each node in a plane ($r = 2$) or a space ($r = 3$). Assume the center of the sensors has been relocated at origin. Then the low rank matrix $A = \mathbb{X}^T\mathbb{X}$, encoding the true distance information, has to satisfy distance constraints given by the measurements. The noisy distance matrix $\widetilde{A}$ after centering, equals to the sum of A and a matrix N consisting of measurement

errors. Suppose that each node is a random point uniformly distributed in a rectangular region. It is not difficult to see that with high probability, the top r eigenvalues of $\mathbb{X}^T\mathbb{X}$ and their eigengap scales with the number of sensors n and the leading eigenvectors have a bounded coherence.

In our theorems, we require that the coherence μ is not too large. This is a natural structural condition associated with the low rank matrices. Consider the following very simple example: if the eigenvectors $v_1, \ldots, v_r$ of the low rank matrix A are uniform unit vectors in a sphere, then with high probability, $\max_i \|v_i\|_\infty = O(\sqrt{\log n})$, which implies $\mu = O(\log n)$. An intuitive way to understand the incoherence structure is that no coordinates of v_1 (or $v_2, \ldots v_r$) are dominant. In other words, the eigenvectors are not concentrated on a few coordinates.

In all our examples, the incoherence structure is natural. The factor model satisfies such structure, which will be discussed in Section 3. In the video surveillance example, ideally, when the images are static, A is a rank one matrix $x\mathbf{1}^T$. Since usually a majority of pixels (coordinates of x) help to display an image, the vector x often has dense coordinates with comparable magnitude, so A also has an incoherence structure in this example. Similarly, in the sensor localization example, the coordinates of all sensor nodes are comparable in magnitude, so the low rank matrix A formed by $\mathbb{X}^T\mathbb{X}$ also has the desired incoherence structure.

2.4 Other perturbation results

Although the eigenvector perturbation theory is well studied in numerical analysis, there is a renewed interest among statistics and machine learning communities recently, due to the wide applicability of PCA and other eigenvector-based methods. In Cai and Zhang (2016); Yu et al. (2015), they obtained variants or improvements of Davis-Kahan theorem (or Wedin's theorem), which are user-friendly in the statistical contexts. These results assume the perturbation is deterministic, which is the same as Davis-Kahan theorem and Wedin's theorem. In general, these results are sharp, even when the perturbation is random, as evidenced by the BBP transition (Baik et al., 2005).

However, these classical results can be suboptimal, when the perturbation is random and the smallest eigenvalue gap $\lambda_1 - \lambda_2$ does not capture particular spectrum structure. For example, Vu (2011); O'Rourke et al. (2013) showed that with high probability, there are bounds sharper than the Wedin's theorem, when the signal matrix is low-rank and satisfies certain eigenvalue conditions.

In this paper, our perturbation results are deterministic, thus the bound can be suboptimal when the perturbation is random with certain structure (e.g. the difference between sample covariance and population one for i.i.d. samples). However, the advantage of a deterministic result is that it is applicable to any random perturbation. This is especially useful when we cannot make strong random assumptions on the perturbation (e.g., the perturbation is an unknown sparse matrix). In Section 3, we will see examples of this type.

3. Application to robust covariance estimation

We will study the problem of robust estimation of covariance matrices and show the strength of our perturbation result. Throughout this section, we assume both rank r and the coherence $\mu(V)$ are bounded by a constant, though this assumption can be relaxed. We will use C to represent a generic constant, and its value may change from line to line.

3.1 PCA in spiked covariance model

To initiate our discussions, we first consider sub-Gaussian random variables. Let $X = (X_1, \ldots, X_d)$ be a random d-dimensional vector with mean zero and covariance matrix

$$\Sigma = \sum_{i=1}^{r} \lambda_i v_i v_i^T + \sigma^2 I_d := \Sigma_1 + \Sigma_2, \qquad (\lambda_1 \geq \ldots \geq \lambda_r > 0), \tag{11}$$

and $\mathbb{X}$ be an n by d matrix, whose rows are independently sampled from the same distribution. This is the spiked covariance model that has received intensive study in recent years. Let the empirical covariance matrix be $\widehat{\Sigma} = \mathbb{X}^T\mathbb{X}/n$. Viewing the empirical covariance matrix as its population version plus an estimation error, we have the decomposition

$$\widehat{\Sigma} = \Sigma_1 + \Sigma_2 + \Big(\frac{1}{n}\mathbb{X}^T\mathbb{X} - \Sigma\Big),$$

which is a special case of the general decomposition in (1). Here, Σ_2 is the sparse component, and the estimation error $\mathbb{X}^T\mathbb{X}/n - \Sigma$ is the noise component. Note that $v_1, \ldots, v_r$ are just the top r leading eigenvectors of Σ and we write $V = [v_1, \ldots, v_r]$. Assume the top r eigenvectors of $\widehat{\Sigma}$ are denoted by $\widehat{v}_1, \ldots, \widehat{v}_r$. We want to find an ℓ_∞ bound on the estimation error $\widehat{v}_i - v_i$ for all $i \in [r]$.

When the dimension d is comparable to or larger than n, it has been shown by Johnstone and Lu (2009) that the leading empirical eigenvector $\widehat{v}_1$ is not a consistent estimate of the true eigenvector v_1, unless we assume larger eigenvalues. Indeed, we will impose more stringent conditions on λ_i's in order to obtain good ℓ_∞ bounds.

Assuming the coherence $\mu(V)$ is bounded, we can easily see $\mathrm{Var}(X_j) \leq \sigma^2 + C\lambda_1/d$ for some constant C. It follows from the standard concentration result (e.g., Vershynin (2010)) that if rows of $\mathbb{X}$ contains i.i.d sub-Gaussian vectors and $\log d = O(n)$, then with probability greater than $1 - d^{-1}$,

$$\|\frac{1}{n}\mathbb{X}^T\mathbb{X} - \Sigma\|_{\max} \leq C\big(\sigma^2 + \frac{\lambda_1}{d}\big)\sqrt{\frac{\log d}{n}}. \tag{12}$$

To apply Theorem 3, we treat Σ_1 as A and $\widehat{\Sigma} - \Sigma_1$ as E. If the conditions in Theorem 3 are satisfied, we will obtain

$$\max_{1\leq k\leq r} \|\widehat{v}_k - v_k\|_\infty = O(\|E\|_\infty/(\lambda_r\sqrt{d}) + \|E\|_2/(\delta\sqrt{d})). \tag{13}$$

Note there are simple bounds on $\|E\|_\infty$ and $\|E\|_2$:

$$\|E\|_2 \leq \|E\|_\infty \leq \sigma^2 + d\,\|\frac{1}{n}\mathbb{X}^T\mathbb{X} - \Sigma\|_{\max} \leq C\Big\{1 + \big(d\sigma^2 + \lambda_1\big)\sqrt{\frac{\log d}{n}}\Big\}.$$

By assuming a strong uniform eigengap, the conditions in Theorem 3 are satisfied, and the bound in (13) can be simplified. Define the uniform eigengap as

$$\gamma = \min\{\lambda_i - \lambda_{i+1} : 1 \le i \le r\}, \qquad \lambda_{r+1} := 0.$$

Note that $\gamma \le \min\{\lambda_r, \delta\}$, so if $\gamma > C(1 + \big(d\sigma^2 + \lambda_1\big)\sqrt{\log d/n})$, we have

$$\max_{1\le k\le r} \|\widehat{v}_k - v_k\|_\infty = O_P\Big(\frac{\|E\|_\infty}{\gamma\sqrt{d}}\Big) = O_P\Big(\frac{1 + \big(d\sigma^2 + \lambda_1\big)\sqrt{\log d/n}}{\gamma\sqrt{d}}\Big),$$

In particular, when $\lambda_1 \asymp \gamma$ and $\gamma \gg \max\{1, \sigma^2 d\sqrt{\log d/n}\}$, we have

$$\max_{1\le k\le r} \|\widehat{v}_k - v_k\|_\infty = o_P\Big(\frac{1}{\sqrt{d}}\Big).$$

The above analysis pertains to the structure of sample covariance matrix. In the following subsections, we will estimate the covariance matrix using more complicated robust procedure. Our perturbation theorems in Section 2 provide a fast and clean approach to obtain new results.

3.2 PCA for robust covariance estimation

The usefulness of Theorem 3 is more pronounced when the random variables are heavy-tailed. Consider again the covariance matrix Σ with structure (11). Instead of assuming sub-Gaussian distribution, we assume there exists a constant $C > 0$ such that $\max_{j\le d} EX_j^4 < C$, i.e. the fourth moments of the random variables are uniformly bounded.

Unlike sub-Gaussian variables, there is no concentration bound similar to (12) for the empirical covariance matrix. Fortunately, thanks to recent advances in robust statistics (e.g., Catoni (2012)), robust estimate of Σ with guaranteed concentration property becomes possible. We shall use the method proposed in Fan et al. (2017a). Motivated by the classical M-estimator of Huber (1964), Fan et al. (2017a) proposed a robust estimator for each element of $\widehat{\Sigma}$, by solving a Huber loss based minimization problem

$$\widehat{\Sigma}_{ij} = \mathrm{argmin}_\mu \sum_{t=1}^n l_\alpha(X_{ti}X_{tj} - \mu), \tag{14}$$

where l_α is the Huber loss defined as

$$l_\alpha(x) = \begin{cases} 2\alpha|x| - \alpha^2, & |x| \ge \alpha, \\ x^2, & |x| \le \alpha. \end{cases}$$

The parameter α is suggested to be $\alpha = \sqrt{nv^2/\log(\epsilon^{-1})}$ for $\epsilon \in (0,1)$, where v is assumed to satisfy $v \ge \max_{ij}\sqrt{\mathrm{Var}(X_iX_j)}$. If $\log(\epsilon^{-1}) \le n/8$, Fan et al. (2017a) showed

$$P\Big(|\widehat{\Sigma}_{ij} - \Sigma_{ij}| \le 4v\sqrt{\frac{\log(\epsilon^{-1})}{n}}\Big) \ge 1 - 2\epsilon.$$

From this result, the next proposition is immediate by taking $\epsilon = d^{-3}$.

Proposition 1 *Suppose that there is a constant C with $\max_{j\le d} EX_j^4 < C$. Then with probability greater than $1 - d^{-1}(1 + d^{-1})$, the robust estimate of covariance matrix with $\alpha = \sqrt{3nv^2\log(d)}$ satisfies*

$$\|\widehat{\Sigma} - \Sigma\|_{\max} \le 4v\sqrt{\frac{3\log d}{n}},$$

where v is a pre-determined parameter assumed to be no less than $\max_{ij}\sqrt{Var(X_iX_j)}$.

This result relaxes the sub-Gaussianity assumption by robustifying the covariance estimate. It is apparent that the ℓ_∞ bound in the previous section is still valid in this case. To be more specific, suppose $\mu(V)$ is bounded by a constant. Then, (13) holds for the PCA based on the robust covariance estimation. When $\lambda_1 \asymp \gamma$ and $\gamma \gg \max\{1, \sigma^2 d\sqrt{\log d/n}\}$, we again have

$$\max_{1\le k\le r} \|\widehat{v}_k - v_k\|_\infty = O_P\Big(\frac{1 + \big(d\sigma^2 + \lambda_1\big)\sqrt{\log d/n}}{\gamma\sqrt{d}}\Big) = o_P\Big(\frac{1}{\sqrt{d}}\Big).$$

Note that an entrywise estimation error $o_p(1/\sqrt{d})$ necessarily implies consistency of the estimated eigenvectors, since we can easily convert an ℓ_∞ result into an ℓ_2 result. The minimum signal strength (or magnitude of leading eigenvalues) for such consistency is shown to be $\sigma^2 d/n$ under the sub-Gaussian assumption (Wang and Fan, 2017).

If the goal is simply to prove consistency of $\widehat{v}_k$, the strategy of using our ℓ_∞ perturbation bounds is not optimal. However, there are also merits: our result is nonasymptotic; it holds for more general distributions (beyond sub-Gaussian distributions); and its entrywise bound gives stronger guarantee. Moreover, the ℓ_∞ perturbation bounds provide greater flexibility for analysis, since it is straightforward to adapt analysis to problems with more complicated structure. For example, the above discussion can be easily extended to a general Σ_2 with bounded $\|\Sigma_2\|_\infty$ rather than a diagonal matrix.

3.3 Robust covariance estimation via factor models

In this subsection, we will apply Theorem 3 to robust large covariance matrix estimation for approximate factor models in econometrics. With this theorem, we are able to extend the data distribution in factor analysis beyond exponentially decayed distributions considered by Fan et al. (2013), to include heavy-tailed distributions.

Suppose the observation y_{it}, say, the excess return at day t for stock i, admits a decomposition

$$y_{it} = b_i^T f_t + u_{it}, \qquad i \le d, t \le n, \tag{15}$$

where $b_i \in \mathbb{R}^r$ is the unknown but fixed loading vector, $f_t \in \mathbb{R}^r$ denotes the unobserved factor vector at time t, and u_{it}'s represent the idiosyncratic noises. Let $y_t = (y_{1t}, \ldots, y_{dt})^T$ and $u_t = (u_{1t}, \ldots, u_{dt})^T$ so that $y_t = Bf_t + u_t$, where $B = (b_1, \ldots, b_d)^T \in \mathbb{R}^{d\times r}$. Suppose that f_t and u_t are uncorrelated and centered random vectors, with bounded fourth moments, i.e., the fourth moments of all entries of f_t and u_t are bounded by some constant. We assume $\{f_t, u_t\}$ are independent for t, although it is possible to allow for weak temporal

dependence as in Fan et al. (2013). From (15), we can decompose $\Sigma = \mathrm{Cov}(y_t)$ into a low rank component and a residual component:

$$\Sigma = BB^T + \Sigma_u, \tag{16}$$

where $\Sigma_u := \mathrm{Cov}(u_t)$. To circumvent the identifiability issue common in latent variable models, here we also assume, without loss of generality, $\mathrm{Cov}(f_t) = I_r$ and that B^TB is a diagonal matrix, since rotating B will not affect the above decomposition (16).

We will need two major assumptions for our analysis: (1) the factors are *pervasive* in the sense of Definition 2, and (2) there is a constant $C > 0$ such that $\|\Sigma_u^{-1}\|_2, \|\Sigma_u\|_2 \le C$, which are standard assumptions in the factor model literature. The pervasive assumption is reasonable in financial applications, since the factors have impacts on a large fraction of the outcomes (Chamberlain et al., 1983; Bai, 2003). If the factor loadings $\{b_i\}_{i=1}^d$ are regarded as random realizations from a bounded random vector, the assumption holds (Fan et al., 2013).

Definition 2 *In the factor model (15), the factors are called pervasive if there is a constant $C > 0$ such that $\|B\|_{\max} \le C$ and the eigenvalues of the r by r matrix B^TB/d are distinct and bounded away from zero and infinity.*

Let $\{\lambda_i, v_i\}_{i=1}^r$ be the top r eigenvalues and eigenvectors of Σ, and similarly, $\{\overline{\lambda}_i, \overline{v}_i\}_{i=1}^r$ for BB^T. In the following proposition, we show that pervasiveness is naturally connected to the incoherence structure. This connects the econometrics and machine learning literature and provide a good interpretation on the concept of the incoherence. Its proof can be found in the appendix.

Proposition 3 *Suppose there exists a constant $C > 0$ such that $\|\Sigma_u\| \le C$. The factors f_t are pervasive if and only if the coherence $\mu(V)$ for $V = (v_1, \ldots, v_r) \in \mathbb{R}^{d\times r}$ is bounded by some constant, and $\lambda_i = \lambda_i(\Sigma) \asymp d$ for $i \le r$ so that $\min_{1\le i\ne j\le r} |\lambda_i - \lambda_j|/\lambda_j > 0$.*

Our goal is to obtain a good covariance matrix estimator by exploiting the structure (16). Our strategy is to use a generalization of the principal orthogonal complement thresholding (POET) method proposed in Fan et al. (2013). The generic POET procedure encompasses three steps:

(1) Given three pilot estimators $\widehat{\Sigma}, \widehat{\Lambda} = \mathrm{diag}(\widehat{\lambda}_1, \ldots, \widehat{\lambda}_r), \widehat{V} = (\widehat{v}_1, \ldots, \widehat{v}_r)$ respectively for true covariance Σ, leading eigenvalues $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_r)$ and leading eigenvectors $V = (v_1, \ldots, v_r)$, compute the principal orthogonal complement $\widehat{\Sigma}_u$:

$$\widehat{\Sigma}_u = \widehat{\Sigma} - \widehat{V}\widehat{\Lambda}\widehat{V}^T. \tag{17}$$

(2) Apply the correlation thresholding to $\widehat{\Sigma}_u$ to obtain thresholded estimate $\widehat{\Sigma}_u^\top$ defined as follows:

$$\widehat{\Sigma}_{u,ij}^\top = \begin{cases} \widehat{\Sigma}_{u,ij}, & i = j \\ s_{ij}(\widehat{\Sigma}_{u,ij}) I(|\widehat{\Sigma}_{u,ij}| \ge \tau_{ij}), & i \ne j \end{cases}, \tag{18}$$

where $s_{ij}(\cdot)$ is the generalized shrinkage function (Antoniadis and Fan, 2001; Rothman et al., 2009) and $\tau_{ij} = \tau(\hat{\sigma}_{u,ii}\hat{\sigma}_{u,jj})^{1/2}$ is an entry-dependent threshold. τ will be determined later in Theorem 6. This step exploits the sparsity of Σ_u.

(3) Construct the final estimator $\widehat{\Sigma}^\top = \widehat{V}\widehat{\Lambda}\widehat{V}^T + \widehat{\Sigma}_u^\top$.

The key feature in the above procedure lies in the flexibility of choosing the pilot estimators in the first step. We will choose $\widehat{\Sigma}$ according to data generating distribution. Typically we can use $\hat{\lambda}_i, \hat{v}_i$ for $i \le r$ as the eigenvalues/vectors of $\widehat{\Sigma}$. However, $\hat{\Lambda}$ and $\hat{V}$ in general do not have to come from the spectral information of $\widehat{\Sigma}$ and can be obtained separately via different methods.

To guide the selection of proper pilot estimators, Fan et al. (2017+) provided a high level sufficient condition for this simple procedure to be effective, and its performance is gauged, in part, through the sparsity level of Σ_u, defined as $m_d := \max_{i\le d} \sum_{j\le d} |\Sigma_{u,ij}|^q$. When $q = 0$, m_d corresponds to the maximum number of nonzero elements in each row of Σ_u. For completeness, we present the theorem given by Fan et al. (2017+) in the following.

Theorem 6 *Let $w_n = \sqrt{\log d/n} + 1/\sqrt{d}$. Suppose there exists $C > 0$ such that $\|\Sigma_u^{-1}\|, \|\Sigma_u\| \le C$ and we have pilot estimators $\widehat{\Sigma}, \widehat{\Lambda}, \widehat{V}$ satisfying*

$$\|\widehat{\Sigma} - \Sigma\|_{\max} = O(\sqrt{\log d/n}), \tag{19}$$

$$|\widehat{\lambda}_i/\lambda_i - 1| = O(\sqrt{\log d/n}), \tag{20}$$

$$\|\widehat{v}_i - v_i\|_\infty = O(w_n/\sqrt{d}). \tag{21}$$

Under the pervasiveness condition of the factor model (15), with $\tau \asymp w_n$, if $m_d w_n^{1-q} = o(1)$, the following rates of convergence hold with the generic POET procedure:

$$\|\widehat{\Sigma}_u^\top - \Sigma_u\|_2 = O\Big(m_d w_n^{1-q}\Big) = \|(\widehat{\Sigma}_u^\top)^{-1} - {\Sigma_u}^{-1}\|_2, \tag{22}$$

and

$$\begin{aligned} \|\widehat{\Sigma}^\top - \Sigma\|_{\max} &= O\Big(w_n\Big), \\ \|\widehat{\Sigma}^\top - \Sigma\|_\Sigma &= O\Big(\frac{\sqrt{d}\log d}{n} + m_d w_n^{1-q}\Big), \\ \|(\widehat{\Sigma}^\top)^{-1} - \Sigma^{-1}\|_2 &= O\Big(m_d w_n^{1-q}\Big), \end{aligned} \tag{23}$$

where $\|A\|_\Sigma = d^{-1/2}\|\Sigma^{-1/2}A\Sigma^{-1/2}\|_F$ is the relative Frobenius norm.

We remark that the additional term $1/\sqrt{d}$ in w_n, is due to the estimation of unobservable factors and is negligible when the dimensional d is high. The optimality of the above rates of convergence is discussed in details in Fan et al. (2017+). Theorem 6 reveals a profound deterministic connection between the estimation error bound of the pilot estimators with the rate of convergences of the POET output estimators. Notice that the eigenvector estimation error is under the ℓ_∞ norm, for which our ℓ_∞ perturbation bounds will prove to be useful.

In this subsection, since we assume only bounded fourth moments, we choose $\widehat{\Sigma}$ to be the robust estimate of covariance matrix Σ defined in (14). We now invoke our ℓ_∞ bounds to show that the spectrum properties (eigenvalues and eigenvectors) are stable to perturbation.

Let us decompose $\widehat{\Sigma}$ into a form such that Theorem 3 can be invoked:

$$\widehat{\Sigma} = \sum_{i=1}^{r} \overline{\lambda}_i \overline{v}_i \overline{v}_i^T + \Sigma_u + (\widehat{\Sigma} - \Sigma),$$

where $\widehat{\Sigma}$ is viewed as $\widetilde{A}$, the low-rank part $\sum_{i=1}^r \overline{\lambda}_i \overline{v}_i \overline{v}_i^T$, which is also BB^T, is viewed as A, and the remaining terms are treated as E. The following results follow immediately.

Proposition 4 *Assume that there is a constant $C > 0$ such that $\|\Sigma_u\| \le C$. If the factors are pervasive, then with probability greater than $1 - d^{-1}$, we have (19) – (21) hold with $\widehat{\lambda}_i, \widehat{v}_i$ as the leading eigenvalues/vectors of $\widehat{\Sigma}$ for $i \le r$. In addition, (22) and (23) hold.*

The inequality (19) follows directly from Proposition 1 under the assumption of bounded fourth moments. It is also easily verifiable that (20), (21) follow from (19) by Weyl's inequality and Theorem 3 (noting that $\|\Sigma_u\|_\infty \le \sqrt{d}\|\Sigma_u\|$). See Section 3.2 for more details.

Note that in the case of sub-Gaussian variables, sample covariance matrix and its leading eigenvalues/vectors will also serve the same purpose due to (12) and Theorem 3 as discussed in Section 3.1.

We have seen that the ℓ_∞ perturbation bounds are useful in robust covariance estimation, and particularly, they resolve a theoretical difficulty in the generic POET procedure for factor model based covariance matrix estimation.

4. Simulations

4.1 Simulation: the perturbation result

In this subsection, we implement numerical simulations to verify the perturbation bound in Theorem 3. We will show that the error behaves in the same way as indicated by our theoretical bound.

In the experiments, we let the matrix size d run from 200 to 2000 by an increment of 200. We fix the rank of A to be 3 ($r = 3$). To generate an incoherence low rank matrix, we sample a $d \times d$ random matrix with iid standard normal variables, perform singular value decomposition, and extract the first r right singular vectors $v_1, v_2, \ldots, v_r$. Let $V = (v_1, \ldots, v_r)$ and $D = \mathrm{diag}(r\gamma, (r-1)\gamma, \ldots, \gamma)$ where γ as before represents the eigengap. Then, we set $A = VDV^T$. By orthogonal invariance, v_i is uniformly distributed on the unit sphere $\mathbb{S}^{d-1}$. It is not hard to see that with probability $1 - O(d^{-1})$, the coherence of V $\mu(V) = O(\sqrt{\log d})$.

We consider two types of sparse perturbation matrices E: (a) construct a $d \times d$ matrix E_0 by randomly selecting s entries for each row, and sampling a uniform number in $[0, L]$ for each entry, and then symmetrize the perturbation matrix by setting $E = (E_0 + E_0^T)/2$; (b) pick $\rho \in (0, 1), L' > 0$, and let $E_{ij} = L'\rho^{|i-j|}$. Note that in (b) we have $\|E\|_\infty \le 2L'/(1-\rho)$, and thus we can choose suitable L' and ρ to control the ℓ_∞ norm of E. This covariance structure is common in cases where correlations between random variables depend on their "distance" $|i - j|$, which usually arises from autoregressive models.

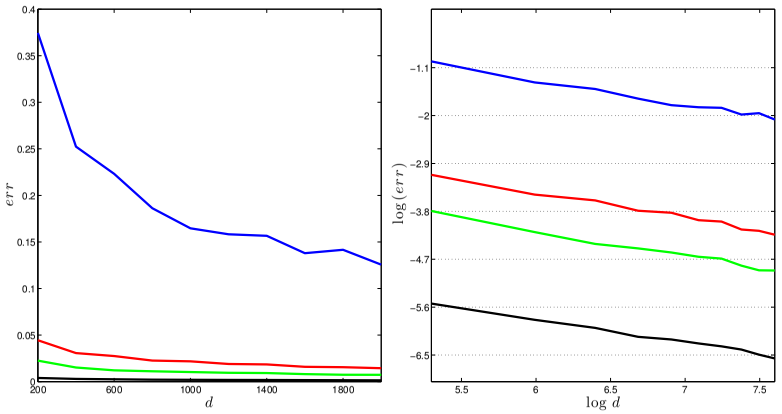


Figure 1: *The left plot shows the perturbation error of eigenvectors against matrix size d ranging from* 200 *to* 2000*, with different eigengap γ. The right plot shows* $\log(err)$ *against* $\log(d)$*. The slope is around* -0.5*. Blue lines represent $\gamma = 10$; red lines $\gamma = 50$; green lines $\gamma = 100$; and black lines $\gamma = 500$. We report the largest error over* 100 *runs.*

The perturbation of eigenvectors is measured by the element-wise error:

$$err := \max_{1 \le i \le r} \min_{\eta_i \in \{\pm 1\}} \|\eta_i \widetilde{v}_i - v_i\|_\infty,$$

where $\{\widetilde{v}_i\}_{i=1}^r$ are the eigenvectors of $\widetilde{A} = A + E$ in the descending order.

To investigate how the error depends on γ and d, we generate E according to mechanism (a) with $s = 10, L = 3$, and run simulations in different parameter configurations: (1) let the matrix size d range from 200 to 2000, and choose the eigengap γ in $\{10, 50, 100, 500\}$ (Figure 1); (2) fix the product $\gamma\sqrt{d}$ to be one of $\{2000, 3000, 4000, 5000\}$, and let the matrix size d run from 200 to 2000 (Figure 2).

To find how the errors behave for E generated from different methods, we run simulations as in (1) but generate E differently. We construct E through mechanism (a) with $L = 10, s = 3$ and $L = 0.6, s = 50$, and also through mechanism (b) with $L' = 1.5, \rho = 0.9$ and $L' = 7.5, \rho = 0.5$ (Figure 3). The parameters are chosen such that $\|E\|_\infty$ is about 30.

In Figure 1 – 3, we report the largest error based on 100 runs. Figure 1 shows that the error decreases as d increases (the left plot); and moreover, the logarithm of the error is linear in $\log(d)$, with a slope -0.5, that is, $err \propto 1/\sqrt{d}$ (the right plot). We can take the eigengap γ into consideration and characterize the relationship in a more refined way. In Figure 2, it is clear that err almost falls on the same horizontal line for different configurations of d and γ, with $\gamma\sqrt{d}$ fixed. The right panel clearly indicates that $err \times \gamma\sqrt{d}$ is a constant, and therefore $err \propto 1/(\gamma\sqrt{d})$. In Figure 3, we find that the errors behave almost the

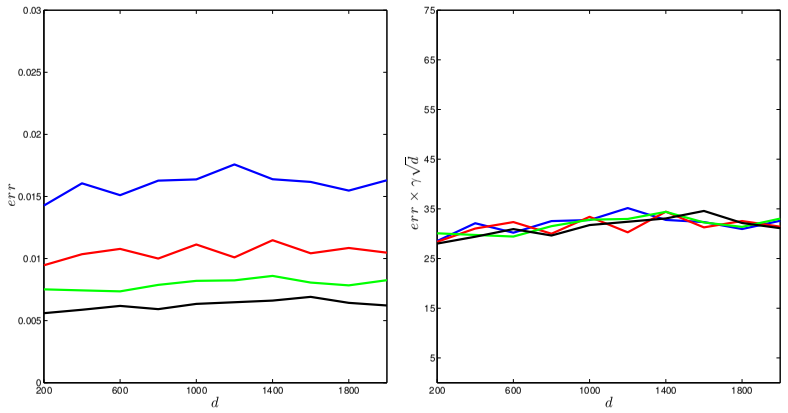


Figure 2: *The left plot shows the perturbation error of eigenvectors against matrix size* d *ranging from* 200 *to* 2000*, when* $\gamma\sqrt{d}$ *is kept fixed, with different values. The right plot shows the error multiplied by* $\gamma\sqrt{d}$ *against* d*. Blue lines represent* $\gamma\sqrt{d} = 2000$*; red lines* $\gamma\sqrt{d} = 3000$*; green lines* $\gamma\sqrt{d} = 4000$*; and black lines* $\gamma\sqrt{d} = 5000$*. We report the largest error over* 100 *runs.*

same regardless of how E is generated. These simulation results provide stark evidence supporting the ℓ_∞ perturbation bound in Theorem 3.

4.2 Simulation: robust covariance esitmation

We consider the performance of the generic POET procedure in robust covariance estimation in this subsection. Note that the procedure is flexible in employing any pilot estimators $\widehat{\Sigma}, \widehat{\Lambda}, \widehat{V}$ satisfying the conditions (19) – (21) respectively.

We implemented the robust procedure with four different initial trios: (1) the sample covariance $\widehat{\Sigma}^S$ with its leading r eigenvalues and eigenvectors as $\widehat{\Lambda}^S$ and $\widehat{V}^S$; (2) the Huber's robust estimator $\widehat{\Sigma}^R$ given in (14) and its top r eigen-structure estimators $\widehat{\Lambda}^R$ and $\widehat{V}^R$; (3) the marginal Kendall's tau estimator $\widehat{\Sigma}^K$ with its corresponding $\widehat{\Lambda}^K$ and $\widehat{V}^K$; (4) lastly, we use the spatial Kendall's tau estimator to estimate the leading eigenvectors instead of the marginal Kendall' tau, so $\widehat{V}^K$ in (3) is replaced with $\widetilde{V}^K$. We need to briefly review the two types of Kendall's tau estimators here, and specifically give the formula for $\widehat{\Sigma}^K$ and $\widetilde{V}^K$.

Kendall's tau correlation coefficient, for estimating pairwise comovement correlation, is defined as

$$\hat{\tau}_{jk} := \frac{2}{n(n-1)} \sum_{t<t'} \mathrm{sgn}((y_{tj} - y_{t'j})(y_{tk} - y_{t'k})) \,. \tag{24}$$

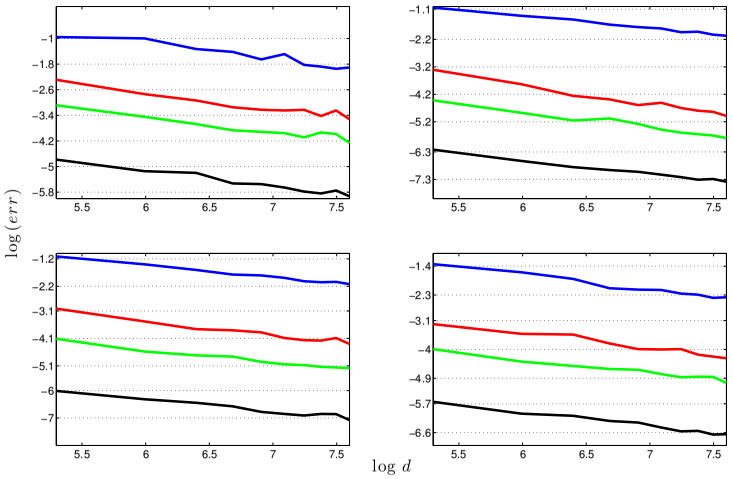


Figure 3: *These plots show* $\log(err)$ *aginst* $\log(d)$*, with matrix size* d *ranging from* 200 *to* 2000 *and different eigengap* γ*. The perturbation* E *is generated from different ways. Top left:* $L = 10, s = 3$*; top right:* $L = 0.6, s = 50$*; bottom left:* $L' = 1.5, \rho = 0.9$*; bottom right:* $L' = 7.5, \rho = 0.5$*. The slopes are around* -0.5*. Blue lines represent* $\gamma = 10$*; red lines* $\gamma = 50$*; green lines* $\gamma = 100$*; and black lines* $\gamma = 500$*. We report the largest error over* 100 *runs.*

Its population expectation is related to the Pearson correlation via the transform $r_{jk} = \sin\left(\frac{\pi}{2} E[\hat{\tau}_{jk}]\right)$ for elliptical distributions (which are far too restrictive for high-dimensional applications). Then $\hat{r}_{jk} = \sin\left(\frac{\pi}{2}\hat{\tau}_{jk}\right)$ is a valid estimation for the Pearson correlation r_{jk}. Letting $\widehat{R} = (\hat{r}_{jk})$ and $\widehat{D} = \mathrm{diag}(\sqrt{\widehat{\Sigma}^R_{11}}, \ldots, \sqrt{\widehat{\Sigma}^R_{dd}})$ containing the robustly estimated standard deviations, we define the marginal Kendall's tau estimator as

$$\widehat{\Sigma}^K = \widehat{D}\,\widehat{R}\,\widehat{D}\,. \tag{25}$$

In the above construction of $\widehat{D}$, we still use the robust variance estimates from $\widehat{\Sigma}^R$.

The spatial Kendall's tau estimator is a second-order U-statstic, defined as

$$\widetilde{\Sigma}^K := \frac{2}{n(n-1)} \sum_{t<t'} \frac{(y_t - y_{t'})(y_t - y_{t'})^T}{\|y_t - y_{t'}\|_2^2} \,. \tag{26}$$

Then $\widetilde{V}^S$ is constructed by the top r eigenvectors of $\widetilde{\Sigma}^K$. It has been shown by Fan et al. (2017+) that under elliptical distribution, $\widehat{\Sigma}^K$ and its top r eigenvalues $\widehat{\Lambda}^K$ satisfy (19) and (20) while $\widetilde{V}^S$ suffices to conclude (21). Hence Method (4) indeed provides good initial estimators if data are from elliptical distribution. However, since $\widehat{\Sigma}^K$ attains (19) for elliptical distribution, by similar argument for deriving Proposition 4 based on our ℓ_∞ pertubation bound, $\widehat{V}^K$ consisting of the leading eigenvectors of $\widehat{\Sigma}^K$ is also valid for the generic POET procedure. For more details about the two types of Kendall's tau, we refer the readers to Fang et al. (1990); Choi and Marden (1998); Han and Liu (2014); Fan et al. (2017+) and references therein.

In summary, Method (1) is designed for the case of sub-Gaussian data; Method (3) and (4) work under the situation of elliptical distribution; while Method (2) is proposed in this paper for the general heavy-tailed case with bounded fourth moments without further distributional shape constraints.

We simulated n samples of $(f_t^T, u_t^T)^T$ from two settings: (a) a multivariate t-distribution with covariance matrix $\mathrm{diag}\{I_r, 5I_d\}$ and various degrees of freedom ($\nu = 3$ for very heavy tail, $\nu = 5$ for medium heavy tail and $\nu = \infty$ for Gaussian tail), which is one example of the elliptical distribution (Fang et al., 1990); (b) an element-wise iid one-dimensional t distribution with the same covariance matrix and degrees of freedom $\nu = 3, 5$ and ∞, which is a non-elliptical heavy-tailed distribution.

Each row of coefficient matrix B is independently sampled from a standard normal distribution, so that with high probability, the pervasiveness condition holds with $\|B\|_{\max} = O(\sqrt{\log d})$. The data is then generated by $y_t = Bf_t + u_t$ and the true population covariance matrix is $\Sigma = BB^T + 5I_d$.

For d running from 200 to 900 and $n = d/2$, we calculated errors of the four robust estimators in different norms. The tuning for α in minimization (14) is discussed more throughly in Fan et al. (2017b). For the thresholding parameter, we used $\tau = 2\sqrt{\log d/n}$. The estimation errors are gauged in the following norms: $\|\widehat{\Sigma}_u^\top - \Sigma_u\|$, $\|(\widehat{\Sigma}^\top)^{-1} - \Sigma^{-1}\|$ and $\|\widehat{\Sigma}^\top - \Sigma\|_\Sigma$ as shown in Theorem 6. The two different settings are separately plotted in Figures 4 and 5. The estimation errors of applying sample covariance matrix $\widehat{\Sigma}^S$ in Method (1) are used as the baseline for comparison. For example, if relative Frobenius norm is used to measure performance, $\|(\widehat{\Sigma}^\top)^{(k)} - \Sigma\|_\Sigma / \|(\widehat{\Sigma}^\top)^{(1)} - \Sigma\|_\Sigma$ will be depicted for $k = 2, 3, 4$, where $(\widehat{\Sigma}^\top)^{(k)}$ are generic POET estimators based on Method (k). Therefore if the ratio curve moves below 1, the method is better than naive sample estimator (Fan et al., 2013) and vice versa. The more it gets below 1, the more robust the procedure is against heavy-tailed randomness.

The first setting (Figure 4) represents a heavy-tailed elliptical distribution, where we expect Methods (2), (3), (4) all outperform the POET estimator based on the sample covariance, i.e. Method (1), especially in the presence of extremely heavy tails (solid lines for $\nu = 3$). As expected, all three curves under various measures show error ratios visibly smaller than 1. On the other hand, if data are indeed Gaussian (dotted line for $\nu = \infty$), Method (1) has better behavior under most measures (error ratios are greater than 1). Nevertheless, our robust Method (2) still performs comparably well with Method (1), whereas the median error ratios for the two Kendall's tau methods are much worse. In addition, the IQR (interquartile range) plots reveal that Method (2) is indeed more stable than two Kendall's tau Methods (3) and (4). It is also noteworthy that Method (4), which

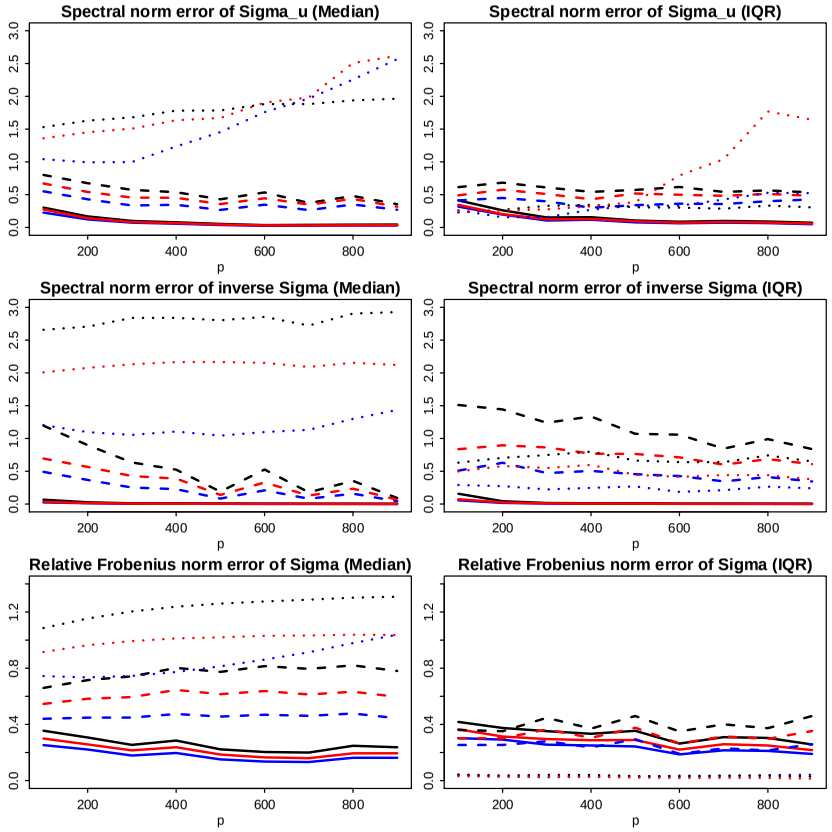


Figure 4: *Error ratios of robust estimates against varying dimension. Blue lines represent errors of Method (2) over Method (1) under different norms; black lines errors of Method (3) over Method (1); red lines errors of Method (4) over Method (1).* (f_t^T, u_t^T) *is generated by multivariate t-distribution with df* $= 3$ *(solid),* 5 *(dashed) and* ∞ *(dotted). The median errors and their IQR's (interquartile range) over* 100 *simulations are reported.*

leverages the advantage of spatial Kendall's tau, performs more robustly than Method (3), which solely base its estimation of the eigen-structure on marginal Kendall's tau.

The second setting (Figure 5) provides an example of non-elliptical distributed data. We can see that the performance of the general robust Method (2) dominates the other three methods, which verifies the benefit of robust estimation for a general heavy-tailed distribu-

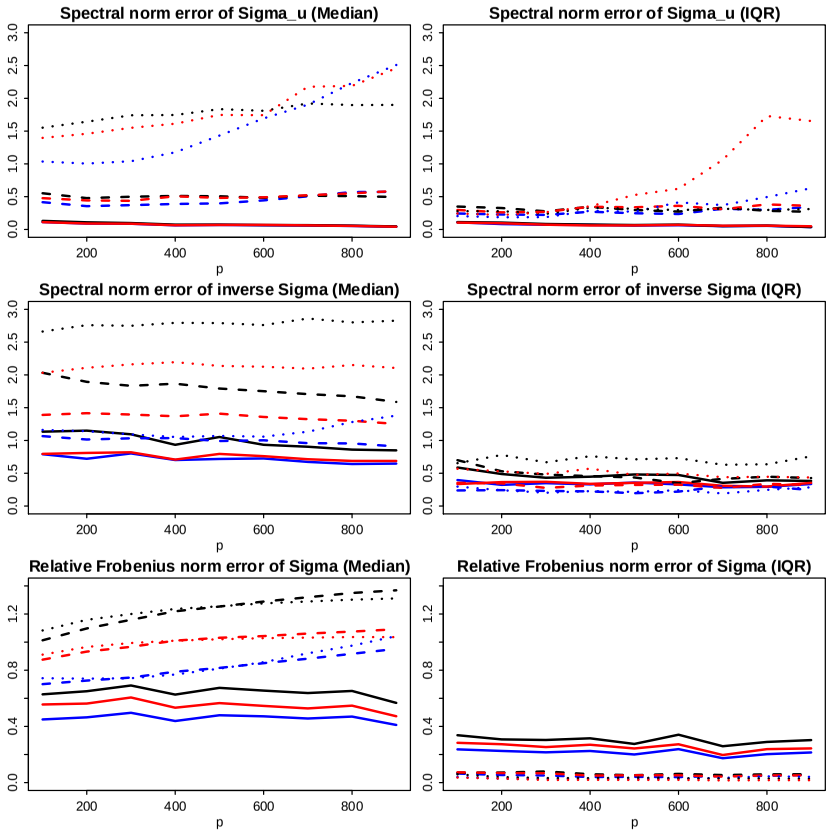


Figure 5: *Error ratios of robust estimates against varying dimension. Blue lines represent errors of Method (2) over Method (1) under different norms; black lines errors of Method (3) over Method (1); red lines errors of Method (4) over Method (1).* (f_t^T, u_t^T) *is generated by element-wise iid t-distribution with df* $= 3$ *(solid),* 5 *(dashed) and* ∞ *(dotted). The median errors and their IQR's (interquartile range) over* 100 *simulations are reported.*

tion. Note that Kendall's tau methods do not apply to distributions outside the elliptical family, excluding even the element-wise iid t distribution in this setting. Nonetheless, even in the first setting where the data are indeed elliptical, with proper tuning, the proposed robust method can still outperform Kendall's tau by a clear margin.

5. Proof organization of main theorems

5.1 Symmetric Case

For shorthand, we write $\tau = \|E\|_\infty$, and $\kappa = \sqrt{d}\,\|EV\|_{\max}$. An obvious bound for κ is $\kappa \le \sqrt{r\mu}\,\tau$ (by Cauchy-Schwarz inequality). We will use these notations throughout this subsection.

Recall the spectral decomposition of A in (8). Expressing E in terms of the column vectors of V and $V_\perp$, which form an orthogonal basis in $\mathbb{R}^n$, we write

$$[V, V_\perp]^T E [V, V_\perp] =: \begin{pmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{pmatrix}. \tag{27}$$

Note that $E_{12} = E_{21}^T$ since E is symmetric. Conceptually, the perturbation results in a rotation of $[V, V_\perp]$, and we write a candidate orthogonal basis as follows:

$$\overline{V} := (V + V_\perp Q)(I_r + Q^T Q)^{-1/2}, \qquad \overline{V}_\perp := (V_\perp - V Q^T)(I_{d-r} + Q Q^T)^{-1/2}, \tag{28}$$

where $Q \in \mathbb{R}^{(d-r)\times r}$ is to be determined. It is straightforward to check that $[\overline{V}, \overline{V}_\perp]$ is an orthogonal matrix. We will choose Q in a way such that $(\overline{V}, \overline{V}_\perp)^T \widetilde{A} (\overline{V}, \overline{V}_\perp)$ is a block diagonal matrix, i.e., $\overline{V}_\perp^T \widetilde{A} \overline{V} = 0$. Substituting (28) and simplifying the equation, we obtain

$$Q(\Lambda_1 + E_{11}) - (\Lambda_2 + E_{22})Q = E_{21} - Q E_{12} Q. \tag{29}$$

The approach of studying perturbation through a quadratic equation is known. See Stewart (1990) for example. Yet, to the best of our knowledge, existing results study perturbation under orthogonal-invariant norms (or unitary-invariant norms in the complex case), which includes a family of matrix operator norms and Frobenius norm, but excludes the matrix max-norm. The advantages of orthogonal-invariant norms are pronounced: such norms of a symmetric matrix only depend on its eigenvalues regardless of eigenvectors; moreover, with suitable normalization they are *consistent* in the sense $\|AB\| \le \|A\| \cdot \|B\|$. See Stewart (1990) for a clear exposition.

The max-norm, however, does not possess these important properties. An imminent issue is that it is not clear how to relate Q to $V_\perp Q$, which will appear in (29) after expanding E according to (27), and which we want to control. Our approach here is to study $\overline{Q} := V_\perp Q$ directly through a transformed quadratic equation, obtained by left multiplying $V_\perp$ to (29). Denote $H = V_\perp E_{21}, \overline{Q} = V_\perp Q, \overline{L}_1 = \Lambda_1 + E_{11}, \overline{L}_2 = V_\perp(\Lambda_2 + E_{22})V_\perp^T$. If we can find an appropriate matrix $\overline{Q}$ with $\overline{Q} = V_\perp Q$, and it satisfies the quadratic equation

$$\overline{Q}\,\overline{L}_1 - \overline{L}_2 \overline{Q} = H - \overline{Q} H^T \overline{Q}, \tag{30}$$

then Q also satisfies the quadratic equation (29). This is because multiplying both sides of (30) by $V_\perp^T$ yields (29), and thus any solution $\overline{Q}$ to (30) with the form $\overline{Q} = V_\perp Q$ must result in a solution Q to (29).

Once we have such $\overline{Q}$ (or equivalently Q), then $(\overline{V}, \overline{V}_\perp)^T \widetilde{A} (\overline{V}, \overline{V}_\perp)$ is a block diagonal matrix, and the span of column vectors of $\overline{V}$ is a candidate space of the span of first r

eigenvectors, namely span$\{\widetilde{v}_1, \ldots, \widetilde{v}_r\}$. We will verify the two spaces are identical in Lemma 7. Before stating that lemma, we first provide bounds on $\|\overline{Q}\|_{\max}$ and $\|\overline{V} - V\|_{\max}$.

Lemma 5 *Suppose $|\lambda_r| - \varepsilon > 4r\mu(\tau + 2r\kappa)$. Then, there exists a matrix $Q \in \mathbb{R}^{(d-r)\times r}$ such that $\overline{Q} = V_\perp Q \in \mathbb{R}^{d\times r}$ is a solution to the quadratic equation (30), and $\overline{Q}$ satisfies $\|\overline{Q}\|_{\max} \le \omega/\sqrt{d}$. Moreover, if $r\omega < 1/2$, the matrix $\overline{V}$ defined in (28) satisfies*

$$\|\overline{V} - V\|_{\max} \le 2\sqrt{\mu}\,\omega r/\sqrt{d}\,. \tag{31}$$

Here, ω is defined as $\omega = 8(1 + r\mu)\kappa/(|\lambda_r| - \varepsilon)$.

The second claim of the lemma (i.e., the bound (31)) is relatively easy to prove once the first claim (i.e., the bound on $\|\overline{Q}\|_{\max}$) is proved. To understand this, note that we can rewrite $\overline{V}$ as $\overline{V} = (V + \overline{Q})(I_r + \overline{Q}^T\overline{Q})^{-1/2}$, and $\|\overline{Q}^T\overline{Q}\|_{\max}$ can be controlled by a trivial inequality $\|\overline{Q}^T\overline{Q}\|_{\max} \le d\|\overline{Q}\|_{\max}^2 \le w^2$. To prove the first claim, we construct a sequence of matrices through recursion that converges to the fixed point $\overline{Q}$, which is a solution to the quadratic equation (30). For all iterates of matrices, we prove a uniform max-norm bound, which leads to a max-bound on $\|\overline{Q}\|_{\max}$ by continuity. To be specific, we initialize $\overline{Q}^0 = 0$, and given $\overline{Q}^t$, we solve a linear equation:

$$\overline{Q}\,\overline{L}_1 - \overline{L}_2\overline{Q} = H - \overline{Q}^t H^T \overline{Q}^t, \tag{32}$$

and the solution is defined as $\overline{Q}^{t+1}$. Under some conditions, the iterate $\overline{Q}^t$ converges to a limit $\overline{Q}$, which is a solution to (30). The next general lemma captures this idea. It follows from Stewart (1990) with minor adaptations.

Lemma 6 *Let T be a bounded linear operator on a Banach space $\mathcal{B}$ equipped with a norm $\|\cdot\|$. Assume that T has a bounded inverse, and define $\beta = \|T^{-1}\|^{-1}$. Let $\varphi : \mathcal{B} \to \mathcal{B}$ be a map that satisfies*

$$\|\varphi(x)\| \le \eta\|x\|^2, \qquad \text{and} \qquad \|\varphi(x) - \varphi(y)\| \le 2\eta\max\{\|x\|, \|y\|\}\|x - y\| \tag{33}$$

for some $\eta \ge 0$. Suppose that $\mathcal{B}_0$ is a closed subspace of $\mathcal{B}$ such that $T^{-1}(\mathcal{B}_0) \subseteq \mathcal{B}_0$ and $\varphi(\mathcal{B}_0) \subseteq \mathcal{B}_0$. Suppose $y \in \mathcal{B}_0$ that satisfies $4\eta\|y\| < \beta^2$. Then, the sequence initialized with $x_0 = 0$ and iterated through

$$x_{k+1} = T^{-1}(y + \varphi(x_k)), \quad k \ge 0 \tag{34}$$

converges to a solution $x^\star$ to $Tx = y + \varphi(x)$. Moreover, we have $x^\star \subseteq \mathcal{B}_0$, and $\|x^\star\| \le 2\|y\|/\beta$.

To apply this lemma to the equation (30), we view $\mathcal{B}$ as the space of matrices $\mathbb{R}^{d\times r}$ with the max-norm $\|\cdot\|_{\max}$, and $\mathcal{B}_0$ as the subspace of matrices of the form $V_\perp Q$ where $Q \in \mathbb{R}^{(d-r)\times r}$. The linear operator T is set to be the $T(\overline{Q}) = \overline{Q}\,\overline{L}_1 - \overline{L}_2\overline{Q}$, and the map φ is set to be the quadratic function $\varphi(\overline{Q}) = -\overline{Q}H^T\overline{Q}$. Roughly speaking, under the assumption of Lemma 6, the nonlinear effect caused by φ is weak compared with the linear operator T. Therefore, it is crucial to show T is invertible, i.e. to give a good lower bound on $\|T^{-1}\|_{\max}^{-1} = \inf_{\|\overline{Q}\|_{\max}=1}\|T(\overline{Q})\|_{\max}$. Since the norm is not orthogonal-invariant, a subtle

issue arises when A is not of exact low rank, which will be discussed at the end of the subsection.

If there is no perturbation (i.e., $E = 0$), all the iterates $\overline{Q}^t$ are simply 0, so $\overline{V}$ is identical to V. If the perturbation is not too large, the next lemma shows that the column vectors of $\overline{V}$ span the same space as span$\{\widetilde{v}_1, \ldots, \widetilde{v}_r\}$.

In other words, with a suitable orthogonal matrix R, the columns of $\overline{V}R$ are $\widetilde{v}_1, \ldots, \widetilde{v}_r$.

Lemma 7 *Suppose $|\lambda_r| - \varepsilon > \max\{3\tau, 64(1 + r\mu)r^{3/2}\mu^{1/2}\kappa\}$. Then, there exists an orthogonal matrix $R \in \mathbb{R}^{r\times r}$ such that the column vectors of $\overline{V}R$ are $\widetilde{v}_1, \ldots, \widetilde{v}_r$.*

Proof of Theorem 2 It is easy to check that under the assumption of Theorem 2, the conditions required in Lemma 5 and Lemma 7 are satisfied. Hence, the two lemmas imply Theorem 2. ■

To study the perturbation of individual eigenvectors, we assume, in addition to the condition on $|\lambda_r|$, that $\lambda_1, \ldots, \lambda_r$ satisfy a uniform gap, (namely $\delta > \|E\|_2$). This additional assumption is necessary, because otherwise, the perturbation may lead to a change of relative order of eigenvalues, and we may be unable to match eigenvectors from the order of eigenvalues. Suppose $R \in \mathbb{R}^{r\times r}$ is an orthogonal matrix such that $\overline{V}R$ are eigenvectors of $\widetilde{A}$. Now, under the assumption of Theorem 2, the column vectors of $\widetilde{V}$ and $\overline{V}R$ are identical up to sign, so we can rewrite the difference $\widetilde{V} - V$ as

$$\widetilde{V} - V = \overline{V}(R - I_r) + (\overline{V} - V). \tag{35}$$

We already provided a bound on $\|\overline{V} - V\|_{\max}$ in Lemma 5. By the triangular inequality, we can derive a bound on $\|\overline{V}\|_{\max}$. If we can prove a bound on $\|R - I_r\|_{\max}$, it will finally leads to a bound on $\|\widetilde{V} - V\|_{\max}$. In order to do so, we use the Davis-Kahan theorem to obtain an bound on $\langle\widetilde{v}_i, v_i\rangle$ for all $i \in [r]$. This will lead to a max-norm bound on $R - I_r$ (with the price of potentially increasing the bound by a factor of r). The details about the proof of Theorem 3 are in the appendix.

We remark that the conditions on $|\lambda_r| - \epsilon$ in Theorem 2 and Theorem 3 are only useful in cases where $|\lambda_r| > \|A - A_r\|_\infty$. Ideally, we would like to have results with assumptions only involving λ_r and λ_{r+1}, like in the Davis-Kahan theorem. Unfortunately, unlike orthogonal-invariant norms that only depend on the eigenvalues of a matrix, the max-norm $\|\cdot\|_{\max}$ is not orthogonal-invariant, and thus it also depends on the eigenvectors of a matrix. For this reason, it is not clear whether we could obtain a lower bound on $\|T^{-1}\|_{\max}^{-1}$ using only the eigenvalues λ_r and λ_{r+1} so that Lemma 6 could be applied. The analysis appears to be difficult if we do not have a bound on $\|T^{-1}\|_{\max}^{-1}$, considering that even in the analysis of linear equations, we need invertibility and a well-controlled condition number.

5.2 Asymmetric case

Let A^d, E^d be $d_1 + d_2$ square matrices defined as

$$A^d := \begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix}, \qquad E^d := \begin{pmatrix} 0 & E \\ E^T & 0 \end{pmatrix}.$$

Also denote $\widetilde{A}^d := A^d + E^d$. This augmentation of an asymmetric matrix into a symmetric one is called Hermitian dilation. Here the superscript d means the Hermitian dilation. We also use this notation to denote quantities corresponding to A^d and $\widetilde{A}^d$.

An important observation is that

$$\begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} u_i \\ \pm v_i \end{pmatrix} = \pm \sigma_i \begin{pmatrix} u_i \\ \pm v_i \end{pmatrix}.$$

From this identity, we know that A^d have nonzero eigenvalues $\pm\sigma_i$ where $1 \le i \le \mathrm{rank}(A)$, and its corresponding eigenvectors are $(u_i^T, \pm v_i^T)^T$. For a given r, we stack these (normalized) eigenvectors with indices $i \in [r]$ into a matrix $V^d \in \mathbb{R}^{(d_1+d_2)\times 2r}$:

$$V^d := \frac{1}{\sqrt{2}} \begin{pmatrix} U & U \\ V & -V \end{pmatrix}.$$

Through the augmented matrices, we can transfer eigenvector results for symmetric matrices to singular vectors of asymmetric matrices. However, we cannot directly invoke the results proved for symmetric matrices, due to an issue about the coherence of V^d: when d_1 and d_2 are not comparable, the coherence $\mu(V^d)$ can be very large even when $\mu(V)$ and $\mu(U)$ are bounded. To understand this, consider the case where $r = 1$, $d_1 \gg d_2$, and all entries of U are $O(1/\sqrt{d_1})$, and all entries of V are $O(1/\sqrt{d_2})$. Then, the coherences $\mu(U)$ and $\mu(V)$ are $O(1)$, but $\mu(V^d) = O((d_1+d_2)/d_2) \gg 1$.

This unpleasant issue about the coherence, nevertheless, can be tackled if we consider a different matrix norm. In order to deal with the different scales of d_1 and d_2, we define the weighted max-norm for any matrix M with $d_1 + d_2$ rows as follows:

$$\|M\|_w := \left\| \begin{pmatrix} \sqrt{d_1} I_{d_1} & 0 \\ 0 & \sqrt{d_2} I_{d_2} \end{pmatrix} M \right\|_{\max}. \tag{36}$$

In other words, we rescale the top d_1 rows of M by a factor of $\sqrt{d_1}$, and rescale the bottom d_2 rows by $\sqrt{d_2}$. This weighted norm serves to balance the potential different scales of d_1 and d_2.

The proofs of theorems in Section 2.2 will be almost the same with those in the symmetric case, with the major difference being the new matrix norm. Because the derivation is slightly repetitive, we will provide concise proofs in the appendix . Similar to the decomposition in (2.1),

$$A^d = \begin{pmatrix} 0 & A_r \\ A_r^T & 0 \end{pmatrix} + \begin{pmatrix} 0 & A - A_r \\ A^T - A_r^T & 0 \end{pmatrix} =: A_r^d + (A^d - A_r^d),$$

where A_r^d is has rank $2r$. Equivalently,

$$A_r^d = \sum_{i=1}^r \sigma_i (u_i^T, v_i^T)^T (u_i^T, v_i^T) - \sum_{i=1}^r \sigma_i (u_i^T, -v_i^T)^T (u_i^T, -v_i^T).$$

Analogously, we define notations in (28)–(30) and use d in the superscript to signify that they are augmented through Hermitian dilation. It is worthwhile to note that $\Lambda_1^d = \mathrm{diag}\{\sigma_1, \ldots, \sigma_r, -\sigma_r, \ldots, -\sigma_1\}$, and that $\min\{|\pm\sigma_i| : i \in [r]\} = \sigma_r$ (a similar quantity as $|\lambda_r|$). Recall $\mu_0 = \mu(U) \vee \mu(V)$, $\tau_0 = \sqrt{d_1/d_2}\|E\|_\infty \vee \sqrt{d_2/d_1}\|E\|_1$ and $\varepsilon_0 = \sqrt{d_1/d_2}\|A - A_r\|_\infty \vee \sqrt{d_2/d_1}\|A - A_r\|_1$. In the proof, we will also use $\kappa_0 = \max\{\sqrt{d_1}\,\|EV\|_{\max}, \sqrt{d_2}\,\|E^T U\|_{\max}\}$, which is a quantity similar to κ.

The next key lemma, which is parallel to Lemma 5, provides a bound on the solution $\overline{Q}^d$ to the quadratic equation

$$\overline{Q}^d \overline{L}_1^d - \overline{L}_2^d \overline{Q}^d = H^d - \overline{Q}^d (H^d)^T \overline{Q}^d. \tag{37}$$

Lemma 8 *Suppose $\sigma_r - \varepsilon_0 > 16 r\mu_0(\tau_0 + r\kappa_0)$. Then, there exists a matrix $Q^d \in \mathbb{R}^{(d_1+d_2-2r)\times 2r}$ such that $\overline{Q}^d = V_\perp^d Q^d \in \mathbb{R}^{(d_1+d_2)\times 2r}$ is a solution to the quadratic equation (37), and $\overline{Q}^d$ satisfies $\|\overline{Q}^d\|_w \le \omega_0$. Moreover, if $r\omega_0 < 1/2$, the matrix $\overline{V}^d$ defined in (28) satisfies*

$$\|\overline{V}^d - V^d\|_w \le 6\sqrt{\mu_0}\, r\omega_0. \tag{38}$$

Here, ω_0 is defined as $\omega_0 = 8(1 + r\mu_0)\kappa_0/3(\sigma_r - \varepsilon_0)$.

In this lemma, the bound (38) bears a similar form to (31): if we consider the max-norm, the first d_1 rows of $\overline{V}^d - V^d$ correspond to the left singular vectors u_i's, and they scale with $1/\sqrt{d_1}$; and the last d_2 rows correspond to the right singular vectors v_i's, which scale with $1/\sqrt{d_2}$. Clearly, the weighted max-norm $\|\cdot\|_w$ indeed helps to balance the two dimensions.

The rest of the proofs can be found in the appendix.

Acknowledgments

We would like to acknowledge support for this project from the following grants: NSF grants DMS-1406266, DMS-1662139, DMS-1712591 and NIH grant R01-GM072611-13.

A. Proofs for Section 2.1

Denote the column span of a matrix M by $\mathrm{span}(M)$. Suppose two matrices $M_1, M_2 \in \mathbb{R}^{n\times m}$ ($m \le n$) have orthonormal column vectors. It is known that (Stewart, 1990)

$$d(M_1, M_2) := \|M_1 M_1^T - M_2 M_2^T\|_2 = \|\sin\Theta(M_1, M_2)\|_2. \tag{39}$$

where $\Theta(M_1, M_2)$ are the canonical angles between $\mathrm{span}(M_1)$ and $\mathrm{span}(M_2)$. Recall the notations defined in (27), and also recall $\kappa = \sqrt{d}\|EV\|_{\max}$, $\Lambda_1 = \mathrm{diag}\{\lambda_1, \ldots, \lambda_r\}$, $\Lambda_2 = \mathrm{diag}\{\lambda_{r+1}, \ldots, \lambda_n\}$, $\overline{L}_1 = \Lambda_1 + E_{11}$, $\overline{L}_2 = V_\perp(\Lambda_2 + E_{22})V_\perp^T$ and $H = V_\perp E_{21}$. The first lemma bounds $\|H\|_{\max}$.

Lemma 9 *We have the following bound on $\|H\|_{\max}$:*

$$\|H\|_{\max} \le (1 + r\mu)\kappa/\sqrt{d}.$$

Proof Using the definition $E_{21} = V_\perp^T EV$ in (27), we can write $H = V_\perp V_\perp^T EV$. Since the columns of V and $V_\perp$ form an orthogonal basis in $\mathbb{R}^d$, clearly

$$VV^T + V_\perp V_\perp^T = I_d\,. \tag{40}$$

By Cauchy-Schwarz inequality and the definition of μ, for any $i, j \in [d]$,

$$|(VV^T)_{ij}| = \sum_{k=1}^r |V_{ik}V_{jk}| \le \big(\sum_{k=1}^r V_{ik}^2\big)^{1/2} \cdot \big(\sum_{k=1}^r V_{jk}^2\big)^{1/2} \le \frac{r\mu}{d}\,.$$

Using the identity (40) and the above inequality, we derive

$$\begin{aligned}\|H\|_{\max} &\le \|EV\|_{\max} + \|VV^T EV\|_{\max} \\ &\le (1 + d\|VV^T\|_{\max})\,\|EV\|_{\max} \le (1 + r\mu)\|EV\|_{\max}\,,\end{aligned}$$

which completes the proof. ∎

Lemma 10 *If $|\lambda_r| > \kappa r\sqrt{\mu}$, then $\overline{L}_1$ is an invertible matrix. Furthermore,*

$$\inf_{\|Q_0\|_{\max}=1} \|Q_0\overline{L}_1 - \overline{L}_2 Q_0\|_{\max} \ge |\lambda_r| - 3r\mu(\tau + r\kappa) - \varepsilon\,, \tag{41}$$

where Q_0 is an $d \times r$ matrix.

Proof Let Q_0 be any $d \times r$ matrix with $\|Q_0\|_{\max} = 1$. Note

$$Q_0\overline{L}_1 - \overline{L}_2 Q_0 = Q_0\Lambda_1 + Q_0E_{11} - \overline{L}_2 Q_0.$$

We will derive upper bounds on Q_0E_{11} and $\overline{L}_2Q_0$, and a lower bound on $Q_0\Lambda_1$. Since $E_{11} = V^TEV$ by definition, we expand Q_0E_{11} and use a trivial inequality to derive

$$\|Q_0E_{11}\|_{\max} \le d\,\|Q_0V^T\|_{\max}\|EV\|_{\max}\,. \tag{42}$$

By Cauchy-Schwarz inequality and the definition of μ in (3), for $i, j \in [d]$,

$$|(Q_0V^T)_{ij}| \le \sum_{k=1}^r |(Q_0)_{ik}V_{jk}| \le \big(\sum_{k=1}^r (Q_0)_{ik}^2\big)^{1/2}\big(\sum_{k=1}^r V_{jk}^2\big)^{1/2} \le \sqrt{r}\cdot\sqrt{\frac{r\mu}{d}}\,,$$

Substituting $\|EV\|_{\max} = \kappa/\sqrt{d}$ into (42), we obtain an upper bound:

$$\|Q_0E_{11}\|_{\max} \le \kappa r\sqrt{\mu}\,. \tag{43}$$

To bound $\overline{L}_2Q_0 = (V_\perp E_{22}V_\perp^T + (A - A_r))Q_0$, we use the identity (40) and write

$$V_\perp E_{22}V_\perp^T Q_0 = V_\perp V_\perp^T EV_\perp V_\perp^T Q_0 = (I_d - VV^T)E(I_d - VV^T)Q_0\,.$$

Using two trivial inequalities $\|EQ_0\|_{\max} \le \|E\|_\infty\|Q_0\|_{\max} = \|E\|_\infty$ and $\|V^TQ_0\|_{\max} \le \|V^T\|_\infty\|Q_0\|_{\max} \le \sqrt{d}$, we have

$$\begin{aligned}\|E(I_d - VV^T)Q_0\|_{\max} &\le \|EQ_0\|_{\max} + r\|EV\|_{\max}\|V^TQ_0\|_{\max} \\ &\le \|E\|_\infty + r\sqrt{d}\,\|EV\|_{\max} = \tau + r\kappa\,.\end{aligned}$$

In the proof of Lemma 9, we showed $\|VV^T\|_{\max} \le r\mu/d$. Thus,

$$\|V_\perp E_{22}V_\perp^TQ_0\|_{\max} \le (1 + d\,\|VV^T\|_{\max})\cdot\|E(I_d - VV^T)Q_0\|_{\max} \le (1 + r\mu)(\tau + r\kappa)\,.$$

Moreover, $\|(A - A_r)Q_0\|_{\max} \le \|A - A_r\|_\infty\|Q_0\|_{\max} = \varepsilon$. Combining the two bounds,

$$\|\overline{L}_2Q_0\|_{\max} \le (1 + r\mu)(\tau + r\kappa) + \varepsilon. \tag{44}$$

It is straightforward to obtain a lower bound on $\|Q_0\Lambda_1\|_{\max}$: since there is an entry of Q_0, say $(Q_0)_{ij}$, that has an absolute value of 1, we have

$$\|Q_0\Lambda_1\|_{\max} \ge |(Q_0)_{ij}\lambda_j| \ge |\lambda_r|. \tag{45}$$

To show $\overline{L}_1$ is invertible, we use (42) and (45) to obtain

$$\|Q_0\overline{L}_1\|_{\max} \ge \|Q_0\Lambda_1\|_{\max} - \|Q_0E_{11}\|_{\max} \ge |\lambda_r| - \kappa r\sqrt{\mu}\,.$$

When $|\lambda_r| - \kappa r\sqrt{\mu} > 0$, $\overline{L}_1$ must have full rank, because otherwise we can choose an appropriate Q_0 in the null space of $\overline{L}_1^T$ so that $Q_0\overline{L}_1 = 0$, which is a contradiction. To prove the second claim of the lemma, we combine the lower bound (45) with upper bounds (43) and (44) to derive

$$\begin{aligned}\|Q_0\overline{L}_1 - \overline{L}_2Q_0\|_{\max} &\ge \|Q_0L_1\|_{\max} - \|Q_0E_{11}\|_{\max} - \|\overline{L}_2Q_0\|_{\max} \\ &\ge |\lambda_r| - \kappa r\sqrt{\mu} - (1 + r\mu)(\tau + r\kappa) - \varepsilon \\ &\ge |\lambda_r| - 3r\mu(\tau + r\kappa) - \varepsilon\,,\end{aligned}$$

which is exactly the desired inequality. ∎

Next we prove Lemma 6. This lemma follows from Stewart (1990), with minor changes that involves $\mathcal{B}_0$. We provide a proof for the sake of completeness.

Proof of Lemma 6 Let us write $\alpha = \|y\|$ for shorthand and recall $\beta = \|T^{-1}\|^{-1}$. As the first step, we show that the sequence $\{x_k\}_{k=0}^{\infty}$ is bounded. By construction in (34), we bound $\|x_{k+1}\|$ using $\|x_k\|$:

$$\|x_{k+1}\| \le \|T^{-1}\|(\|y\| + \|\varphi(x_k)\|) \le \frac{\alpha}{\beta} + \frac{\eta}{\beta}\|x_k\|^2.$$

We use this inequality to derive an upper bound on $\{x_k\}$ for all k. We define $\xi_0 = 0$ and

$$\xi_{k+1} = \frac{\alpha}{\beta} + \frac{\eta}{\beta}\xi_k^2, \quad k \ge 0,$$

then clearly $\|x_k\| \le \xi_k$ (which can be shown by induction). It is easy to check (by induction) that the sequence $\{\xi_k\}_{k=1}^{\infty}$ is increasing. Moreover, since $4\alpha\eta < \beta^2$, the quadratic function

$$\phi(\xi) = \frac{\alpha}{\beta} + \frac{\eta}{\beta}\xi^2,$$

has two fixed points (namely solutions to $\phi(\xi) = \xi$), and the smaller one satisfies

$$\xi_\star = \frac{2\alpha}{\beta + \sqrt{\beta^2 - 4\eta\alpha}} < \frac{2\alpha}{\beta}.$$

If $\xi_k < \xi_\star$, then $\xi_{k+1} = \phi(\xi_k) \le \phi(\xi_\star) = \xi_\star$. Thus, by induction, all ξ_k are bounded by $\xi_\star$. This implies $\|x_k\| \le \xi_\star < 2\alpha/\beta$. The next step is to show that the sequence $\{x_k\}$ converges. Using the recursive definition (34) again, we derive

$$\begin{aligned}\|x_{k+1} - x_k\| &\le \|T^{-1}\|\|\varphi(x_k) - \varphi(x_{k-1})\| \\ &\le 2\beta^{-1}\eta \max\{\|x_k\|, \|x_{k-1}\|\}\|x_k - x_{k-1}\| \\ &\le \frac{4\alpha\eta}{\beta^2}\|x_k - x_{k-1}\|.\end{aligned}$$

Since $4\alpha\eta/\beta^2 < 1$, the sequence $\{x_k\}_{k=0}^{\infty}$ is a Cauchy sequence, and convergence is secured. Let $x^\star \in \mathcal{B}$ be the limit. It is clear by assumption that $x_k \in \mathcal{B}_0$ implies $x_{k+1} \in \mathcal{B}_0$, so $x^\star \in \mathcal{B}_0$ and $\|x^\star\| \le 2\alpha/\beta$ by continuity.

The final step is to show $x^\star$ is a solution to $Tx = y + \phi(x)$. Because $\{x_k\}_{k=0}^{\infty}$ is bounded and ϕ satisfies (33), the sequence $\{\phi(x_k)\}_{k=0}^{\infty}$ converges to $\phi(x^\star)$ by continuity and compactness. The linear operator T is also continuous, so we can take limits on both sides of $Tx_{k+1} = y + \phi(x_k)$, we conclude that $x^\star$ is a solution to $Tx = y + \phi(x)$. ■

With all the preparations, we are now ready to present the key lemma. As discussed in Section 5, we set

$$\mathcal{B}_0 := \{\overline{Q} \in \mathbb{R}^{d\times r} : \overline{Q} = V_\perp Q \text{ for some } Q \in \mathbb{R}^{(d-r)\times r}\}.$$

which is a subspace of $\mathcal{B} = \mathbb{R}^{d\times r}$. Consider the matrix max-norm $\|\cdot\|_{\max}$ in $\mathcal{B}$.

Lemma 11 *Suppose $|\lambda_r| - \varepsilon > 4r\mu(\tau + 2r\kappa)$. Then there exists a solution $\overline{Q} \in \mathcal{B}_0$ to the equation (30) with*

$$\|\overline{Q}\|_{\max} \le \frac{8(1 + r\mu)\kappa}{(|\lambda_r| - \varepsilon)\sqrt{d}}.$$

Proof We will invoke Lemma 6 and apply it to the quadratic equation (30). To do so, we first check the conditions required in Lemma 6.

Let the linear operator $\mathcal{T}$ be $\mathcal{T}\overline{Q} = \overline{Q}\overline{L}_1 - \overline{L}_2\overline{Q}$. By Lemma 10, $\mathcal{T}$ has a bounded inverse, and $\beta := \|\mathcal{T}^{-1}\|_{\max}^{-1}$ is bounded from below:

$$\beta \ge |\lambda_r| - 3r\mu(\tau + r\kappa) - \varepsilon. \tag{46}$$

Let us define φ by $\varphi(\overline{Q}) = \overline{Q}H^T\overline{Q}$. To check the inequalities in (33), observe that

$$\|\varphi(\overline{Q})\|_{\max} \le rd\|\overline{Q}\|_{\max}\|H\|_{\max}\|\overline{Q}\|_{\max} \le (1 + r\mu)\kappa r\sqrt{d}\,\|\overline{Q}\|_{\max}^2$$

where we used Lemma 9. We also observe

$$\begin{aligned}\|\varphi(\overline{Q}_1) - \varphi(\overline{Q}_2)\|_{\max} &= \|\overline{Q}_1 H^T(\overline{Q}_1 - \overline{Q}_2) + (\overline{Q}_1 - \overline{Q}_2)H^T\overline{Q}_2\|_{\max} \\ &\le rd\|\overline{Q}_1\|_{\max}\|H\|_{\max}\|\overline{Q}_1 - \overline{Q}_2\|_{\max} + rd\|\overline{Q}_1 - \overline{Q}_2\|_{\max}\|H\|_{\max}\|\overline{Q}_2\|_{\max} \\ &\le 2(1 + r\mu)\kappa r\sqrt{d}\,\max\{\|\overline{Q}_1\|_{\max}, \|\overline{Q}_2\|_{\max}\}\|\overline{Q}_1 - \overline{Q}_2\|_{\max}.\end{aligned}$$

Thus, if we set $\eta = (1 + r\mu)\kappa r\sqrt{d}$, then inequalities required in (33) are satisfied. For any $\overline{Q}$ with $\overline{Q} = V_\perp Q \in \mathcal{B}_0$, obviously $\varphi(\overline{Q}) = V_\perp QH^T\overline{Q} \in \mathcal{B}_0$. To show $\mathcal{T}^{-1}(\overline{Q}) \in \mathcal{B}_0$, let $Q_0 = \mathcal{T}^{-1}(\overline{Q})$ and observe that

$$Q_0\overline{L}_1 - \overline{L}_2 Q_0 = \overline{Q} \in \mathcal{B}_0.$$

By definition, we know $\overline{L}_2 Q_0 = V_\perp(E_{22} + \Lambda_2)V_\perp^T Q_0 \in \mathcal{B}_0$, so we deduce $Q_0\overline{L}_1 \in \mathcal{B}_0$. Our assumption implies $|\lambda_r| > \kappa r\sqrt{\mu}$, so by Lemma 10, the matrix $\overline{L}_1$ is invertible, and thus $Q_0 \in \mathcal{B}_0$. The last condition we check is $4\eta\|H\|_{\max} < \beta^2$. By Lemma 9 and (46), this is true if

$$4(1 + r\mu)^2\kappa^2 r < \big[|\lambda_r| - 3r\mu(\tau + r\kappa) - \varepsilon\big]^2.$$

The above inequality holds when $|\lambda_r| > 4r\mu(V)(\tau + 2r\kappa) + \varepsilon$. Under this condition, we have, by Lemma 6,

$$\|\overline{Q}\|_{\max} \le \frac{2(1 + r\mu)\kappa}{\big(|\lambda_r| - 3r\mu(\tau + r\kappa) - \varepsilon\big)\sqrt{d}} \le \frac{8(1 + r\mu)\kappa}{(|\lambda_r| - \varepsilon)\sqrt{d}},$$

where, the second inequality is due to $3r\mu(\tau + r\kappa) \le 3(|\lambda_r| - \varepsilon)/4$. ■

The next lemma is a consequence of Lemma 11. We define, as in Lemma 5, that $\omega = 8(1 + r\mu)\kappa/(|\lambda_r| - \varepsilon)$.

Lemma 12 *If $r\omega^2 < 1/2$, then*

$$\|(I_r + \overline{Q}^T\overline{Q})^{-1/2} - I_r\|_{\max} \le r\omega^2, \qquad \|(I_r + \overline{Q}^T\overline{Q})^{-1/2}\|_{\max} \le \frac{3}{2}.$$

Proof By the triangular inequality, the second inequality is immediate from the first one. To prove the first inequality, suppose the spectral decomposition of $\overline{Q}^T\overline{Q}$ is $\overline{Q}^T\overline{Q} = \overline{U}\,\overline{\Sigma}\,\overline{U}^T$, where $\overline{\Sigma} = \mathrm{diag}\{\overline{\lambda}_1, \ldots, \overline{\lambda}_r\}$ where $\overline{\lambda}_1 \ge \ldots \ge \overline{\lambda}_r$, and $\overline{U} = [\overline{u}_1, \overline{u}_2, \ldots, \overline{u}_r]$ where $\overline{u}_1, \ldots, \overline{u}_r$ are orthonormal vectors in $\mathbb{R}^r$. Since $\overline{Q}^T\overline{Q}$ has nonnegative eigenvalues, we have $\overline{\lambda}_r \ge 0$. Using these notations, we can rewrite the matrix as

$$(I_r + \overline{Q}^T\overline{Q})^{-1/2} - I_r = \sum_{i=1}^{r} \big((1+\overline{\lambda}_i)^{-1/2} - 1\big)\overline{u}_i\overline{u}_i^T\,.$$

Note that $\overline{\lambda}_1 \le \|\overline{Q}^T\overline{Q}\| \le rd\|\overline{Q}\|_{\max}^2 \le r\omega^2$, which implies $\overline{\lambda}_1 < 1/2$. It is easy to check that $1+|x| \ge (1+x)^{-1/2} \ge 1-|x|$ whenever $|x| < 1/2$. From this fact, we know $|(1+\overline{\lambda}_i)^{-1/2} - 1| \le \overline{\lambda}_i \le r\omega^2$. Using Cauchy-Schwarz inequality, we deduce that for any $j, k \in [d]$,

$$\begin{aligned}
\big|[(I_r + \overline{Q}^T\overline{Q})^{-1/2} - I_r]_{jk}\big| &\le \sum_{i=1}^{r} \big|(1+\overline{\lambda}_i)^{-1/2} - 1\big| \cdot |\overline{U}_{ji}\overline{U}_{ki}| \\
&\le r\omega^2 \cdot \Big(\sum_{i=1}^{r} \overline{U}_{ji}^2\Big)^{1/2}\Big(\sum_{i=1}^{r} \overline{U}_{ki}^2\Big)^{1/2} \\
&\le r\omega^2.
\end{aligned}$$

This leads to the desired max-norm bound. ■

Proof of Lemma 5 The first claim of the lemma (the existence of $\overline{Q}$ and its max-norm bound) follows directly from Lemma 11. To prove the second claim, we split $\overline{V} - V$ into two parts:

$$\begin{aligned}
\overline{V} - V &= V\Big((I_r + Q^TQ)^{-1/2} - I_r\Big) + V_\perp Q(I_r + Q^TQ)^{-1/2} \\
&= V\Big((I_r + \overline{Q}^T\overline{Q})^{-1/2} - I_r\Big) + \overline{Q}(I_r + \overline{Q}^T\overline{Q})^{-1/2}, \qquad (47)
\end{aligned}$$

where we used identity $V_\perp^T V_\perp = I_{d-r}$. Note that $r\omega < 1/2$ implies $r\omega^2 = (r\omega)^2/r < 1/(4r) < 1/2$. Thus, we can use Lemma 12 and derive

$$\Big\|V\Big((I_r + \overline{Q}^T\overline{Q})^{-1/2} - I_r\Big)\Big\|_{\max} \le \sqrt{\frac{r^2\mu}{d}}\|(I_r + \overline{Q}^T\overline{Q})^{-1/2} - I_r\|_{\max} \le \sqrt{\frac{\mu}{d}}\, r^2\omega^2.$$

where we used Cauchy-Schwarz inequality. Using the above inequality and the bound on $\|Q\|_{\max}$ (namely, the first claim in the lemma),

$$\begin{aligned}
\|\overline{V} - V\|_{\max} &\le \sqrt{\frac{\mu}{d}}\, r^2\omega^2 + r\|\overline{Q}\|_{\max}\|(I_r + \overline{Q}^T\overline{Q})^{-1/2}\|_{\max} \\
&\le (\sqrt{\mu}\,\omega^2 r^2 + 3\omega r/2)/\sqrt{d}.
\end{aligned}$$

Simplifying the bound using $r\omega \le 1/2$ and a trivial bound $\mu \ge 1$, we obtain (31). ■

Proof of Lemma 7 Using the identity in (39), it follows from Davis-Kahan $\sin\Theta$ theorem (Davis and Kahan, 1970) and Weyl's inequality that

$$d(\widetilde{V}, V) \le \frac{\|E\|_2}{\delta_r - \|E\|_2},$$

when $\delta_r > \|E\|_2$, where $\delta_r = |\lambda_r| - |\lambda_{r+1}|$. Since $\lambda_{r+1} \le \|A - A_r\|_2 \le \varepsilon$ and $\|E\|_2 \le \tau$, the condition $|\lambda_r| - \varepsilon > 3\tau$ implies $\delta_r > 3\|E\|_2$. Hence, we have $d(\widetilde{V}, V) < 1/2$. Moreover,

$$\begin{aligned}
d(\overline{V}, V) &= \|\overline{V}\overline{V}^T - VV^T\|_2 \le \|\overline{V}(\overline{V} - V)^T\|_2 + \|(\overline{V} - V)V^T\|_2 \\
&\le 2\|\overline{V} - V\|_2 \le 2\sqrt{rd}\,\|\overline{V} - V\|_{\max} \\
&\le 4r^{3/2}\sqrt{\mu}\,\omega,
\end{aligned}$$

where we used a trivial inequality $\|M\|_2 \le \|M\|_F \le \sqrt{rd}\,\|M\|_{\max}$ for any $M \in \mathbb{R}^{d\times r}$. Under the condition $|\lambda_r| - \epsilon > 64(1+r\mu)r^{3/2}\mu^{1/2}\kappa$, it is easy to check that $4r^{3/2}\sqrt{\mu}\,\omega \le 1/2$. Thus, we obtain $d(\overline{V}, V) < 1/2$. By the triangular inequality,

$$d(\widetilde{V}, \overline{V}) \le d(\widetilde{V}, V) + d(\overline{V}, V) < 1.$$

Since $(\overline{V}, \overline{V}_\perp)^T\widetilde{A}(\overline{V}, \overline{V}_\perp)$ is a block diagonal matrix, $\mathrm{span}(\overline{V})$ is the same as the subspace spanned by r eigenvectors of $\widetilde{A}$. We claim that $\mathrm{span}(\overline{V}) = \mathrm{span}(\widetilde{v}_1, \ldots, \widetilde{v}_r)$. Otherwise, there exists an eigenvector $u \in \mathrm{span}(\overline{V})$ whose associated eigenvalue is distinct from $\widetilde{\lambda}_1, \ldots, \widetilde{\lambda}_r$ (since $\delta_r > 3\|E\|_2$), and thus u is orthogonal to $\widetilde{v}_1, \ldots, \widetilde{v}_r$. Therefore,

$$\|(\widetilde{V}\widetilde{V}^T - \overline{V}\overline{V}^T)u\|_2 = \|\overline{V}\overline{V}^T u\|_2 = \|u\|_2.$$

This implies $d(\widetilde{V}, \overline{V}) \ge 1$, which is a contradiction. ■

Proof of Theorem 3 We split $\widetilde{V} - V$ into two parts—see (35). In the following, we first obtain a bound on $\|R - I_r\|_{\max}$, which then results in a bound on $\|\widetilde{V} - V\|_{\max}$.

Under the assumption of the theorem, $r\omega < 1/2$, so

$$\|\overline{V}\|_{\max} \le \|\overline{V} - V\|_{\max} + \|V\|_{\max} \le (2\sqrt{\mu}\,r\omega)/\sqrt{d} + \sqrt{r\mu}/\sqrt{d} \le 2\sqrt{r\mu/d}\,. \qquad (48)$$

To bound $\|R - I_r\|_{\max}$, we rewrite R as $R = \overline{V}^T \overline{V} R = \overline{V}^T \widetilde{V}$. Expand $\overline{V}$ according to (28),

$$R = (I_r + \overline{Q}^T \overline{Q})^{-1/2} (V + \overline{Q})^T \widetilde{V} \,.$$

Let us make a few observations: (a) $\|\overline{Q}^T \widetilde{V}\|_{\max} \le \sqrt{d} \|\overline{Q}\|_{\max} \le \omega$ by Cauchy-Schwarz inequality; (b) $\|\widetilde{V}^T V\|_{\max} \le 1$ by Cauchy-Schwarz inequality again; and (c) $\|(I_r + \overline{Q}^T \overline{Q})^{-1/2} - I_r\|_{\max} \le r\omega^2$ by Lemma 12. Using these inequalities, we have

$$\begin{aligned} \|R - (V + \overline{Q})^T \widetilde{V}\|_{\max} &\le r \|(I_r + \overline{Q}^T \overline{Q})^{-1/2} - I_r\|_{\max} \, \|(V + \overline{Q})^T \widetilde{V}\|_{\max} \\ &\le r^2 \omega^2 (1 + \omega) \,. \end{aligned} \tag{49}$$

Furthermore, by Davis-Kahn $\sin\Theta$ theorem (Davis and Kahan, 1970) and Weyl's inequality, for any $i \in [r]$,

$$\sin\theta(v_i, \widetilde{v}_i) = \sqrt{1 - \langle v_i, \widetilde{v}_i \rangle^2} \le \frac{\|E\|_2}{\delta - \|E\|_2} \,. \tag{50}$$

when $\delta > \|E\|_2$ (δ is defined in Theorem 3). This leads to the bound $\sin\theta(v_i, \widetilde{v}_i) \le 2\|E\|_2/\delta$ (which is a simplified bound). This is because when $\delta \ge 2\|E\|_2$, the bound is implied by (50); when $\delta < 2\|E\|_2$, the bound trivially follows from $\sin\theta(v_i, \widetilde{v}_i) \le 1$. We obtain, up to sign, for $i \le r$,

$$\sqrt{1 - \langle v_i, \widetilde{v}_i \rangle} \le \sqrt{1 - \langle v_i, \widetilde{v}_i \rangle^2} \le \frac{2\|E\|_2}{\delta} \,. \tag{51}$$

In other words, each diagonal entry of $I_r - V^T \widetilde{V}$, namely $1 - \langle v_i, \widetilde{v}_i \rangle$, is bounded by $4\|E\|_2^2/\delta^2$. Since $\{\widetilde{v}_i\}_{i=1}^r$ are orthonormal vectors, we have $1 - \langle v_i, \widetilde{v}_i \rangle^2 \ge \sum_{i' \ne i} \langle v_i, \widetilde{v}_{i'} \rangle^2 \ge \langle v_i, \widetilde{v}_j \rangle^2$ for any $i \ne j$, which leads to bounds on off-diagonal entries of $V^T \widetilde{V} - I_r$. We will combine the two bounds. Note that when $\delta \ge 2\|E\|_2$,

$$\|V^T \widetilde{V} - I_r\|_{\max} \le \max\Big\{ \frac{4\|E\|_2^2}{\delta^2}, \frac{2\|E\|_2}{\delta} \Big\} = \frac{2\|E\|_2}{\delta};$$

and when $\delta < 2\|E\|_2$, $\|V^T \widetilde{V} - I_r\|_{\max}$ is trivially bounded by 1 (up to sign), which is trivially bounded by $2\|E\|_2/\delta$. In either case, we deduce

$$\|V^T \widetilde{V} - I_r\|_{\max} \le \frac{2\|E\|_2}{\delta}. \tag{52}$$

Using the bounds in (49) and (52) and $\|\overline{Q}^T \widetilde{V}\|_{\max} \le \omega$, we obtain

$$\begin{aligned} \|R - I_r\|_{\max} &\le \|R - (V + \overline{Q})^T \widetilde{V}\|_{\max} + \|V^T \widetilde{V} - I_r\|_{\max} + \|\overline{Q}^T \widetilde{V}\|_{\max} \\ &\le r^2 \omega^2 (1 + \omega) + \frac{2\|E\|_2}{\delta} + \omega \,. \end{aligned}$$

We use the inequality $r\omega < 1/2$ to simplify the above bound:

$$\begin{aligned} \|R - I_r\|_{\max} &\le r^2 \omega^2 (1 + \omega) + \omega + 2\|E\|_2/\delta \le (\frac{1}{2} + \frac{1}{4} + 1) r\omega + 2\|E\|_2/\delta \\ &\le 2r\omega + 2\|E\|_2/\delta \,. \end{aligned} \tag{53}$$

We are now ready to bound $\|\widetilde{V} - V\|_{\max}$. In (35), we use the bounds (48), (53), (31) to obtain

$$\begin{aligned} \|\widetilde{V} - V\|_{\max} &= \|\overline{V}(R - I_r) + (\overline{V} - V)\|_{\max} \le r\|\overline{V}\|_{\max} \|R - I_r\|_{\max} + \|\overline{V} - V\|_{\max} \\ &\le 2r\sqrt{r\mu/d}\,(2r\omega + 2\|E\|_2/\delta) + 2r\sqrt{\mu}\,\omega/\sqrt{d} \\ &\le \frac{(4r^{5/2}\mu^{1/2} + 2r\mu^{1/2})\omega}{\sqrt{d}} + \frac{4r^{3/2}\mu^{1/2}\|E\|_2}{\delta\sqrt{d}} \\ &\le \frac{48(1 + r\mu) r^{5/2} \mu^{1/2} \kappa}{(|\lambda_r| - \varepsilon)\sqrt{d}} + \frac{4r^{3/2}\mu^{1/2}\|E\|_2}{\delta\sqrt{d}}. \end{aligned}$$

Using a trivial inequality $\kappa \le \sqrt{r\mu}\,\tau$, the above bound leads to

$$\|\widetilde{V} - V\|_{\max} = O\Big(\frac{r^4 \mu^2 \tau}{(|\lambda_r| - \varepsilon)\sqrt{d}} + \frac{r^{3/2}\mu^{1/2}\|E\|_2}{\delta\sqrt{d}} \Big).$$

■

B. Proofs for Section 2.2

Recall the definitions of μ_0, τ_0, κ_0 and ε_0 in Section 5.2. Similar to the symmetric case, we will use the following easily verifiable inequalities.

$$\kappa_0 \le \sqrt{r\mu_0}\,\tau_0, \qquad \|E\|_2 \le \Big(\sqrt{d_1/d_2}\,\|E\|_\infty \cdot \sqrt{d_2/d_1}\,\|E\|_1 \Big)^{1/2} \le \tau_0. \tag{54}$$

Lemma 13 *Parallel to Lemma 9, we have*

$$\|H^d\|_w \le (1 + r\mu_0)\kappa_0 \,,$$

where $\kappa_0 = \sqrt{d_1}\,\|EV\|_{\max} \vee \sqrt{d_2}\|E^T U\|_{\max}$ *as defined.*

Proof Recall $H^d = V_\perp^d (V_\perp^d)^T E^d V^d = E^d V^d - V^d (V^d)^T E^d V^d$. Note $V^d (V^d)^T = \mathrm{diag}(UU^T, VV^T)$ and $\|UU^T\|_{\max} \le r\mu(U)/d_1$, $\|VV^T\|_{\max} \le r\mu(V)/d_2$. Thus,

$$\begin{aligned} \|H^d\|_w &\le \|E^d V^d\|_w + \|V^d (V^d)^T E^d V^d\|_w \\ &\le (1 + d_1 \|UU^T\|_{\max} \vee d_2 \|VV^T\|_{\max}) \, \|E^d V^d\|_w \le (1 + r\mu_0)\kappa_0 \,. \end{aligned}$$

■

Lemma 14 *Parallel to Lemma 10, if $\sigma_r > 2\kappa_0 r\sqrt{\mu_0}$, then $\overline{L}_1^d$ is a non-degenerate matrix. Furthermore, we have the following bound*

$$\inf_{\|Q_0^d\|_w=1} \|Q_0^d \overline{L}_1^d - \overline{L}_2^d Q_0^d\|_w \geq \sigma_r - 4r\mu_0(\tau_0 + r\kappa_0) - \varepsilon_0\,, \tag{55}$$

where $Q_0^d \in \mathbb{R}^{(d_1+d_2)\times 2r}$.

Proof Following similar derivations with Lemma 10, we have $\|Q_0^d E_{11}^d\|_w \leq 2\kappa_0 r\sqrt{\mu_0}$, and for any matrix $Q_0^d \in \mathbb{R}^{(d_1+d_2)\times 2r}$ with $\|Q_0^d\|_w = 1$,

$$\|\overline{L}_2^d Q_0^d\|_w = \|(A^d - A_r^d)Q_0^d + V_\perp^d (V_\perp^d)^T E^d V_\perp^d (V_\perp^d)^T Q_0^d\|_w \leq \varepsilon_0 + (1 + r\mu_0)(\tau_0 + r\kappa_0).$$

This can be checked by expressing Q_0^d as a block matrix and expand the matrix multiplication. In particular, one can verify that (i) $\|(A^d - A_r^d)Q_0^d\|_w \leq \varepsilon_0$; (ii) For any matrix M with $d_1 + d_2$ rows, $\|V^d(V^d)^T M\|_w \leq r\mu_0\|M\|_w$; (iii) $\|E^d Q_0^d\|_w \leq \tau_0$; (iv) $\|E^d V^d (V^d)^T Q_0^d\|_w \leq r\kappa_0$. Moreover, $\|Q_0^d \Lambda_1^d\|_w \geq \sigma_r \|Q_0^d\|_w \geq \sigma_r$. Thus,

$$\inf_{\|Q_0^d\|_w=1} \|Q_0^d \overline{L}_1^d - \overline{L}_2^d Q_0^d\|_w \geq \sigma_r - 4r\mu_0(\tau_0 + r\kappa_0) - \varepsilon_0\,,$$

which is the desired inequality in the lemma. In addition, $\overline{L}_1^d$ is non-degenerate if $\sigma_r > 2\kappa_0 r\sqrt{\mu_0} > 0$. ■

Lemma 15 *Parallel to Lemma 11, there is a solution $\overline{Q}^d \in \mathcal{B}_0$ to the system (37) such that if $\sigma_r - \varepsilon_0 > 16r\mu_0(\tau_0 + r\kappa_0)$, then*

$$\|\overline{Q}^d\|_w \leq \frac{8(1 + r\mu_0)\kappa_0}{3(\sigma_r - \varepsilon_0)}\,.$$

Proof We again invoke Lemma 6. Let $\mathcal{B}$ be the space $\mathbb{R}^{(d_1+d_2)\times 2r}$ equipped with the weighted max-norm $\|\cdot\|_w$. We also define $\mathcal{B}_0$ as a subspace of $\mathcal{B}$ consisting of matrices of the form $V_\perp^d Q^d$ where Q^d has size $(d_1 + d_2 - 2r) \times 2r$. Let the linear operator $\mathcal{T}^d$ be $\mathcal{T}^d \overline{Q}^d := \overline{Q}^d \overline{L}_1^d - \overline{L}_2^d \overline{Q}^d$. First notice from Lemma 14, $\mathcal{T}^d$ is a linear operator with bounded inverse, i.e., $\beta := \|(\mathcal{T}^d)^{-1}\|_w^{-1}$ is bounded from below by

$$\beta \geq \sigma_r - 4r\mu_0(\tau_0 + r\kappa_0) - \varepsilon_0\,.$$

Let φ be a map given by $\varphi(\overline{Q}^d) = \overline{Q}^d (H^d)^T \overline{Q}^d$. Note that $H^d \in \mathcal{B}$. Using the (easily verifiable) inequality

$$\|M_1 M_2^T M_3\|_w \leq 2r\|M_1\|_w \|M_2^T M_3\|_{\max} \leq 4r\|M_1\|_w \|M_2\|_w \|M_3\|_w \quad \forall\, M_1, M_2, M_3 \in \mathcal{B}, \tag{56}$$

we derive, by the bound on $\|H^d\|_w$ (Lemma 13), that

$$\|\varphi(\overline{Q}^d)\|_w \leq 4r\|H^d\|_w \|\overline{Q}^d\|_w^2 \leq 4r(1 + r\mu_0)\kappa_0 \|\overline{Q}^d\|_w^2\,.$$

Moreover, using the inequality (56) and the bound on $\|H^d\|_w$ (Lemma 13),

$$\begin{aligned}\|\varphi(\overline{Q}_1^d) - \varphi(\overline{Q}_2^d)\|_w &\leq 4r\|\overline{Q}_1^d\|_w \|H^d\|_w \|\overline{Q}_1^d - \overline{Q}_2^d\|_w + 4r\|\overline{Q}_1^d - \overline{Q}_2^d\|_w \|H^d\|_w \|\overline{Q}_2^d\|_w \\ &\leq 8r(1 + r\mu_0)\kappa_0 \max\{\|\overline{Q}_1^d\|_w, \|\overline{Q}_2^d\|_w\} \|\overline{Q}_1^d - \overline{Q}_2^d\|_w.\end{aligned}$$

Thus, we can choose $\eta = 4r(1 + r\mu_0)\kappa_0$, and the condition (33) in Lemma 6 is satisfied. To ensure $4\eta\|H^d\|_w < \beta^2$, it suffices to require (again by Lemma 13),

$$16r(1 + r\mu_0)^2 \kappa_0^2 < \big[\sigma_r - 4r\mu_0(\tau_0 + r\kappa_0) - \varepsilon_0\big]^2.$$

It is easily checkable that the above inequality holds when $\sigma_r - \varepsilon_0 > 16r\mu_0(\tau_0 + r\kappa_0)$. Under this condition, by Lemma 6,

$$\|\overline{Q}^d\|_w \leq \frac{2\|H^d\|_w}{\beta} \leq \frac{2(1 + r\mu_0)\kappa_0}{\sigma_r - 4r\mu(\tau_0 + r\kappa_0) - \varepsilon_0} \leq \frac{2(1 + r\mu_0)\kappa_0}{\sigma_r - \varepsilon_0 - (\sigma_r - \varepsilon_0)/4} \leq \frac{8(1 + r\mu_0)\kappa_0}{3(\sigma_r - \varepsilon_0)}\,,$$

which completes the proof. ■

Proof of Lemma 8 The first claim of the lemma (existence of $\overline{Q}^d$ and its max-norm bound) follows from Lemma 15. To prove the second claim, we split $\overline{V}^d - V^d$ into two parts:

$$\overline{V}^d - V^d = V^d \Big((I_{2r} + (\overline{Q}^d)^T \overline{Q}^d)^{-1/2} - I_{2r}\Big) + \overline{Q}^d (I_{2r} + (\overline{Q}^d)^T \overline{Q}^d)^{-1/2}, \tag{57}$$

Note $\kappa_0 \leq \tau_0\sqrt{r\mu_0}$ (see (54)). It can be checked that the condition $\sigma_r - \varepsilon_0 > 16r\mu_0(\tau_0 + r\kappa_0)$ implies $r\omega_0 < 1/3$. Since $\|(\overline{Q}^d)^T \overline{Q}^d\|_{\max} \leq 2\omega_0^2$ and $\|(\overline{Q}^d)^T \overline{Q}^d\|_2 \leq 2r\|(\overline{Q}^d)^T \overline{Q}^d\|_{\max} \leq 4r\omega_0^2 < 1/2$, similar to Lemma 12, we have

$$\begin{aligned}&\|(I_{2r} + (\overline{Q}^d)^T \overline{Q}^d)^{-1/2} - I_{2r}\|_{\max} \leq 4r\omega_0^2, \\ &\|(I_{2r} + (\overline{Q}^d)^T \overline{Q}^d)^{-1/2}\|_{\max} \leq 3/2.\end{aligned} \tag{58}$$

This yields

$$\begin{aligned}\|\overline{V}^d - V^d\|_w &= \|V^d \Big((I_{2r} + (\overline{Q}^d)^T \overline{Q}^d)^{-1/2} - I_{2r}\Big)\|_w + \|\overline{Q}^d (I_{2r} + (\overline{Q}^d)^T \overline{Q}^d)^{-1/2}\|_w \\ &\leq \sqrt{2r^2\mu_0}\, 4r\omega_0^2 + 2r \cdot 3/2 \cdot \|\overline{Q}^d\|_w \leq 8\sqrt{\mu_0}\,\omega_0^2 r^2 + 3\,\omega_0 r \qquad (59) \\ &\leq 8\sqrt{\mu_0}\,\omega_0 r/3 + 3\sqrt{\mu_0}\,\omega_0 r \leq 6\sqrt{\mu_0}\, r\omega_0. \qquad (60)\end{aligned}$$

■

Lemma 16 *Suppose $\sigma_r - \varepsilon_0 > \max\{16r\mu_0(\tau_0 + r\kappa_0), 64r^{3/2}\mu_0^{1/2}(1 + r\mu_0)\kappa_0\}$. Then, there exists an orthogonal matrix $R_V \in \mathbb{R}^{r\times r}$ (or R_U) such that the column vectors of $\overline{V}R_V$ (and $\overline{U}R_U$) are the top r right (and left) singular vectors of $\widetilde{A}$.*

Proof Similar to the proof of Lemma 7, we will prove $d(\overline{V},V) < 1/2$ and $d(\widetilde{V},V) \le 1/2$, which would then imply that $\overline{V}$ and $\widetilde{V}$ are the same only up to an orthogonal transformation. The same is true for $\overline{U}$ and $\widetilde{U}$, and we will leave out its proof.

By Weyl's inequality for singular values (also known as Mirsky's theorem (Mirsky, 1960)), for any i, $|\overline{\sigma}_i - \sigma_i| \le \|E\|$. By Wedin's perturbation bounds for singular vectors (Wedin, 1972),

$$d(\widetilde{V},V) \le \frac{\|E\|_2}{\sigma_r - \|E\|_2}.$$

Note that $\|E\|_2 \le \tau_0$ (see (54)) Under the assumption in the lemma, clearly $\sigma_r - \varepsilon_0 > 3\tau_0$, and we have $d(\widetilde{V},V) \le 1/2$. Moreover, by Lemma 8, we have $\|\overline{V}^d - V^d\|_w \le 6r\omega_0\sqrt{\mu_0}$. Note that each column vector of V^d and $\overline{V}^d$ are (d_1+d_2)-dimensional. Looking at the last d_2 dimensions, we have $\|\overline{V} - V\|_{\max} \le 6r\omega_0\sqrt{\mu_0/d_1}$.

$$d(\overline{V},V) \le 2\|\overline{V} - V\| \le 2\sqrt{rd_1}\|\overline{V} - V\|_{\max} \le 12r^{3/2}\mu_0^{1/2}\omega_0.$$

Under the assumption of the lemma, $d(\overline{V},V) \le 1/2$. Therefore, we deduce $d(\widetilde{V},\overline{V}) = 0$, and conclude that there exists an orthogonal matrix $R_V \in \mathbb{R}^{r\times r}$ such that $\widetilde{V} = \overline{V}R_V$.

■

Proof of Theorem 4 Lemma 8, together with Lemma 16, implies Theorem 4. ■

Proof of Theorem 5 Similar to the proof of Theorem 3, we first split the difference $\widetilde{V}^d - V^d$:

$$\widetilde{V}^d - V^d = \overline{V}^d(R^d - I_{2r}) + (\overline{V}^d - V^d). \tag{61}$$

To bound the first term, note that under our assumption, $r\omega_0 < 1/3$ (derived in the proof of Lemma 8), it is easy to check $\|\overline{V}^d\|_w \le 3\sqrt{r\mu_0}$. We rewrite the matrix R^d as

$$R^d = (I_{2r} + (\overline{Q}^d)^T\overline{Q}^d)^{-1/2}(V^d + \overline{Q}^d)^T\widetilde{V}^d.$$

Notice that $\|(\overline{Q}^d)^T\widetilde{V}^d\|_{\max} \le \sqrt{2}\|\overline{Q}^d\|_w \le \sqrt{2}\,\omega_0$, $\|(\widetilde{V}^d)^TV^d\|_{\max} \le 1$ and

$$\begin{aligned}\|R^d - (V^d + \overline{Q}^d)^T\widetilde{V}^d\|_{\max} &\le 2r\|(I_{2r} + (\overline{Q}^d)^T\overline{Q}^d)^{-1/2} - I_{2r}\|_{\max}\,\|(V^d + \overline{Q}^d)^T\widetilde{V}^d\|_{\max}\\ &\le 8r^2\omega_0^2(1+\sqrt{2}\,\omega_0).\end{aligned}$$

where we used (58). Following the same derivations as in the proof of Theorem 3, and using the (easily verifiable) fact $\|E^d\|_2 = \|E\|_2$, we can bound $\|(V^d)^T\widetilde{V}^d - I_{2r}\|_{\max}$ by $2\|E\|_2/\delta_0$. Thus, using $r\omega_0 \le 1/3$, under $\delta_0 > 2\|E\|_2$, we have

$$\begin{aligned}\|R^d - I_{2r}\|_{\max} &\le \|R^d - (V^d + \overline{Q}^d)^T\widetilde{V}^d\|_{\max} + \|(V^d)^T\widetilde{V}^d - I_{2r}\|_{\max} + \|(\overline{Q}^d)^T\widetilde{V}^d\|_{\max}\\ &\le 8r^2\omega_0^2(1+\sqrt{2}\,\omega_0) + \frac{2\|E\|_2}{\delta_0} + \sqrt{2}\,\omega_0 < 4\sqrt{2}\,r\omega_0 + \frac{2\|E\|_2}{\delta_0}.\end{aligned} \tag{62}$$

Finally, in order to bound $\|\widetilde{V}^d - V^d\|_w$, we use (60), (61) and (62), and derive

$$\begin{aligned}\|\widetilde{V}^d - V^d\|_w &= \|\overline{V}^d(R^d - I_{2r}) + (\overline{V}^d - V^d)\|_w \le 2r\|\overline{V}^d\|_w\|R^d - I_{2r}\|_{\max} + \|\overline{V}^d - V^d\|_w\\ &\le 6r\sqrt{r\mu_0}\Big(4\sqrt{2}\,r\omega_0 + \frac{2\|E\|_2}{\delta_0}\Big) + 6r\sqrt{\mu_0}\,\omega_0 \le \Big(40r^{5/2}\,\omega_0 + 12r^{3/2}\,\frac{\|E\|_2}{\delta_0}\Big)\cdot\sqrt{\mu_0}\\ &\le \frac{107r^{5/2}\mu_0^{1/2}(1+r\mu_0)\kappa_0}{\sigma_r - \varepsilon_0} + \frac{12r^{3/2}\mu_0^{1/2}\|E\|_2}{\delta_0}\\ &= O\Big(\frac{r^4\mu_0^2\tau_0}{\sigma_r - \varepsilon_0} + \frac{r^{3/2}\mu_0^{1/2}\|E\|_2}{\delta_0}\Big).\end{aligned}$$

This completes the proof. ■

C. Proofs for Section 3

Proof of Proposition 3 Note first by Weyl's inequality, $|\lambda_i - \overline{\lambda}_i| \le \|\Sigma_u\| \le C$. So this implies that $\overline{\lambda}_i = \lambda_i(B^TB) \asymp d$ if and only if $\lambda_i = \lambda_i(\Sigma) \asymp d$ for $i \le r$. And furthermore the eigenvalues of B^TB/d are distinct if and only if $\min_{1\le i\ne j\le r}|\lambda_i(\Sigma) - \lambda_j(\Sigma)|/\lambda_j(\Sigma) > 0$.

To prove the equivalency of bounded $\|B\|_{\max}$ and bounded coherence. We first prove the necessary condition. Again from Weyl's inequality, $\lambda_i(\Sigma) \le C$ for $i \ge r+1$. If $\mu(V)$ is bounded, Σ_{ii} must also be bounded, since $\Sigma_{ii} \le \sum_{j=1}^r v_{ij}^2\lambda_j(\Sigma) + \lambda_{r+1}(\Sigma) \le C(\mu(V)+1)$. Therefore $\|b_i\|^2 \le \|b_i\|^2 + (\Sigma_u)_{ii} = \Sigma_{ii}$ implies $\|B\|_{\max}$ is bounded. Namely, the factors are pervasive.

On the contrary, if pervasiveness holds, we need to prove that $\mu(V)$ is bounded. Let $B = (\widetilde{b}_1,\ldots,\widetilde{b}_r)$. Obviously $\overline{\lambda}_i = \|\widetilde{b}_i\|^2 \asymp d$ and $\overline{v}_i = \widetilde{b}_i/\|\widetilde{b}_i\|$. Without loss of generality, assume $\overline{\lambda}_i$'s are decreasing. So $\|\overline{v}_i\|_\infty \le \|B\|_{\max}/\|\widetilde{b}_i\| \le C/\sqrt{d}$ and $\mu(\overline{V}) \le C$ where $\overline{V} = (\overline{v}_1,\ldots,\overline{v}_r)$. By Theorem 3,

$$\|\overline{v}_i - v_i\|_\infty \le C\frac{\|\Sigma_u\|_\infty}{\overline{\gamma}\sqrt{d}},$$

where $\overline{\gamma} = \min\{\overline{\lambda}_i - \overline{\lambda}_{i+1} : 1 \le i \le r\} \asymp d$ with the convention $\overline{\lambda}_{r+1} = 0$. Hence, we have $\|v_i\|_\infty \le C/\sqrt{d}$, which implies bounded coherence $\mu(V)$. ■

References

Anestis Antoniadis and Jianqing Fan. Regularization of wavelet approximations. *Journal of the American Statistical Association*, 96(455), 2001.

Jushan Bai. Inferential theory for factor models of large dimensions. *Econometrica*, 71(1): 135–171, 2003.

Jinho Baik, Gérard Ben Arous, and Sandrine Péché. Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices. *Annals of Probability*, pages 1643–

1697, 2005.

David J Bartholomew, Martin Knott, and Irini Moustaki. *Latent variable models and factor analysis: A unified approach*, volume 904. John Wiley & Sons, 2011.

Quentin Berthet and Philippe Rigollet. Optimal detection of sparse principal components in high dimension. *The Annals of Statistics*, 41(4):1780–1815, 2013.

Peter J Bickel and Elizaveta Levina. Covariance regularization by thresholding. *The Annals of Statistics*, pages 2577–2604, 2008.

Pratik Biswas and Yinyu Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 46–54. ACM, 2004.

Ingwer Borg and Patrick JF Groenen. *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005.

T Tony Cai and Anru Zhang. Rate-optimal perturbation bounds for singular subspaces with applications to high-dimensional statistics. *arXiv preprint arXiv:1605.00353*, 2016.

Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

Olivier Catoni. Challenging the empirical mean and empirical variance: a deviation study. In *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, volume 48, pages 1148–1185. Institut Henri Poincaré, 2012.

Gary Chamberlain, Michael Rothschild, et al. Arbitrage, factor structure, and mean-variance analysis on large asset markets. *Econometrica*, 51(5):1281–1304, 1983.

Venkat Chandrasekaran, Sujay Sanghavi, Pablo A Parrilo, and Alan S Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2): 572–596, 2011.

Yudong Chen. Incoherence-optimal matrix completion. *IEEE Transactions on Information Theory*, 61(5):2909–2923, 2015.

Kyungmee Choi and J Marden. A multivariate version of kendall's τ. *Journal of Nonparametric Statistics*, 9(3):261–293, 1998.

Chandler Davis and William Morton Kahan. The rotation of eigenvectors by a perturbation. III. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

Fernando De La Torre and Michael J Black. A framework for robust subspace learning. *International Journal of Computer Vision*, 54(1-3):117–142, 2003.

Lance Doherty, Kristofer SJ Pister, and Laurent El Ghaoui. Convex position estimation in wireless sensor networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1655–1663. IEEE, 2001.

Eugene F Fama and Kenneth R French. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33(1):3–56, 1993.

Jianqing Fan, Yingying Fan, and Jinchi Lv. High dimensional covariance matrix estimation using a factor model. *Journal of Econometrics*, 147(1):186–197, 2008.

Jianqing Fan, Yuan Liao, and Martina Mincheva. Large covariance estimation by thresholding principal orthogonal complements. *Journal of the Royal Statistical Society: Series B*, 75:1–44, 2013.

Jianqing Fan, Quefeng Li, and Yuyan Wang. Robust estimation of high-dimensional mean regression. *Journal of Royal Statistical Society, Series B*, 79:247–265, 2017a.

Jianqing Fan, Weichen Wang, and Yiqiao Zhong. Robust covariance estimation for approximate factor models. *Journal of Econometrics*, 2017b.

Jianqing Fan, Han Liu, and Weichen Wang. Large covariance estimation through elliptical factor models. *The Annals of Statitistics*, page to appear, 2017+.

Kai-Tai Fang, Samuel Kotz, and Kai Wang Ng. *Symmetric multivariate and related distributions*. CRC Press, 1990.

Arjun K Gupta, Tamas Varga, and Taras Bodnar. *Elliptically contoured models in statistics and portfolio theory*. Springer, 2013.

Fang Han and Han Liu. Scale-invariant sparse PCA on high-dimensional meta-elliptical data. *Journal of the American Statistical Association*, 109(505):275–287, 2014.

Daniel Hsu and Sivan Sabato. Heavy-tailed regression with a generalized median-of-means. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 37–45, 2014.

Weiming Hu, Tieniu Tan, Liang Wang, and Steve Maybank. A survey on visual surveillance of object motion and behaviors. *Systems, Man, and Cybernetics, Part C: IEEE Transactions on Applications and Reviews*, 34(3):334–352, 2004.

Peter J Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

Iain M. Johnstone and Arthur Yu Lu. On consistency and sparsity for principal components analysis in high dimensions. *Journal of the American Statistical Association*, 104(486): 682–693, 2009. doi: 10.1198/jasa.2009.0121.

Ian Jolliffe. *Principal Component Analysis*. Wiley Online Library, 2002.

Zongming Ma. Sparse principal component analysis and iterative thresholding. *The Annals of Statistics*, 41(2):772–801, 2013.

Leon Mirsky. Symmetric gauge functions and unitarily invariant norms. *The quarterly journal of mathematics*, 11:50–59, 1960.

Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.

Nuria M Oliver, Barbara Rosario, and Alex P Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, 2000.

Sean O'Rourke, Van Vu, and Ke Wang. Random perturbation of low rank matrices: Improving classical bounds. *arXiv preprint arXiv:1311.2657*, 2013.

Vern Paulsen. *Completely bounded maps and operator algebras*, volume 78. Cambridge University Press, 2002.

Jasson DM Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on Machine learning*, pages 713–719. ACM, 2005.

Karl Rohe, Sourav Chatterjee, and Bin Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, pages 1878–1915, 2011.

Adam J Rothman, Elizaveta Levina, and Ji Zhu. Generalized thresholding of large covariance matrices. *Journal of the American Statistical Association*, 104(485):177–186, 2009.

Gilbert W Stewart. Matrix perturbation theory. 1990.

James H Stock and Mark W Watson. Forecasting using principal components from a large number of predictors. *Journal of the American Statistical Association*, 97(460):1167–1179, 2002.

Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.

Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

Van Vu. Singular vectors under random perturbation. *Random Structures & Algorithms*, 39(4):526–538, 2011.

Vincent Q Vu and Jing Lei. Minimax sparse principal subspace estimation in high dimensions. *The Annals of Statistics*, 41(6):2905–2947, 2013.

Weichen Wang and Jianqing Fan. Asymptotics of empirical eigenstructure for high dimensional spiked covariance. *The Annals of Statitistics*, 45(3):1342–1374, 2017.

Per-Åke Wedin. Perturbation bounds in connection with singular value decomposition. *BIT Numerical Mathematics*, 12(1):99–111, 1972.

Yi Yu, Tengyao Wang, and Richard J Samworth. A useful variant of the Davis–Kahan theorem for statisticians. *Biometrika*, 102:315–323, 2015.

Ming Yuan, Ali Ekici, Zhaosong Lu, and Renato Monteiro. Dimension reduction and coefficient estimation in multivariate linear regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(3):329–346, 2007.

Journal of Machine Learning Research 18 (2018) 1-42 Submitted 6/16; Revised 5/17; Published 4/18

A Tight Bound of Hard Thresholding

Jie Shen JS2007@RUTGERS.EDU
Department of Computer Science
Rutgers University
Piscataway, NJ 08854, USA

Ping Li PINGLI98@GMAIL.COM
Baidu Research
Bellevue, WA 98004, USA

Editor: Sujay Sanghavi

Abstract

This paper is concerned with the hard thresholding operator which sets all but the k largest absolute elements of a vector to zero. We establish a *tight* bound to quantitatively characterize the deviation of the thresholded solution from a given signal. Our theoretical result is universal in the sense that it holds for all choices of parameters, and the underlying analysis depends only on fundamental arguments in mathematical optimization. We discuss the implications for two domains:

Compressed Sensing. On account of the crucial estimate, we bridge the connection between the restricted isometry property (RIP) and the sparsity parameter for a vast volume of hard thresholding based algorithms, which renders an improvement on the RIP condition especially when the true sparsity is unknown. This suggests that in essence, many more kinds of sensing matrices or fewer measurements are admissible for the data acquisition procedure.

Machine Learning. In terms of large-scale machine learning, a significant yet challenging problem is learning accurate sparse models in an efficient manner. In stark contrast to prior work that attempted the ℓ_1-relaxation for promoting sparsity, we present a novel stochastic algorithm which performs hard thresholding in each iteration, hence ensuring such parsimonious solutions. Equipped with the developed bound, we prove the *global linear convergence* for a number of prevalent statistical models under mild assumptions, even though the problem turns out to be non-convex.

Keywords: sparsity, hard thresholding, compressed sensing, stochastic optimization

1. Introduction

Over the last two decades, pursuing sparse representations has emerged as a fundamental technique throughout bioinformatics (Olshausen and Field, 1997), statistics (Tibshirani, 1996; Efron et al., 2004), signal processing (Chen et al., 1998; Donoho et al., 2006; Donoho, 2006; Candès and Wakin, 2008) and mathematical science (Chandrasekaran et al., 2012), to name just a few. In order to obtain a sparse solution, a plethora of practical algorithms have been presented, among which two prominent examples are greedy pursuit and convex relaxation (Tropp and Wright, 2010). For instance, as one of the earliest greedy algorithms, orthogonal matching pursuit (OMP) (Pati et al., 1993) repeatedly picks a coordinate as the potential support of a solution. While OMP may fail for some deterministic sensing matrices, Tropp (2004); Tropp and Gilbert (2007) showed that it recovers the true signal with high probability when using random matrices such as Gaussian. Inspired by the success of OMP, the two concurrent work of compressive sampling matching pursuit (CoSaMP) (Needell and Tropp, 2009) and subspace pursuit (SP) (Dai and Milenkovic, 2009) made improvement by selecting multiple coordinates followed by a pruning step in each iteration, and the recovery condition was framed under the restricted isometry property (RIP) (Candès and Tao, 2005). Interestingly, the more careful selection strategy of CoSaMP and SP leads to an optimal sample complexity. The iterative hard thresholding (IHT) algorithm (Daubechies et al., 2004; Blumensath and Davies, 2008, 2009) gradually refines the iterates by gradient descent along with truncation. Foucart (2011) then developed a concise algorithm termed hard thresholding pursuit (HTP), which combined the idea of CoSaMP and IHT, and showed that HTP is superior to both in terms of the RIP condition. Jain et al. (2011) proposed an interesting variant of the HTP algorithm and obtained a sharper RIP result. Recently, Bahmani et al. (2013) and Yuan et al. (2018) respectively extended CoSaMP and HTP to general objective functions, for which a global convergence was established.

Since the sparsity constraint counts the number of non-zero components which renders the problem non-convex, the ℓ_1-norm was suggested as a convex relaxation dating back to basis pursuit (Chen et al., 1998; Donoho and Tsaig, 2008) and Lasso (Tibshirani, 1996). The difference is that Lasso looks for an ℓ_1-norm constrained solution that minimizes the residual while the principle of basis pursuit is to find a signal with minimal ℓ_1-norm that fits the observation data. Candès and Tao (2005) carried out a detailed analysis on the recovery performance of basis pursuit. Another popular estimator in the high-dimensional statistics is the Dantzig selector (Candès and Tao, 2007) which, instead of constraining the residual of the linear model, penalizes the maximum magnitude of the gradient. From a computational perspective, both basis pursuit and Dantzig selector can be solved by linear programming, while Lasso is formulated as a quadratic problem. Interestingly, under the RIP condition or the uniform uncertainty assumption (Candès et al., 2006), a series of work showed that exact recovery by convex programs is possible as soon as the observation noise vanishes (Candès and Tao, 2005; Candès, 2008; Wainwright, 2009; Cai et al., 2010; Foucart, 2012).

In this paper, we are interested in the hard thresholding (HT) operator underlying a large body of the developed algorithms in compressed sensing (e.g., IHT, CoSaMP, SP), machine learning (Yuan and Zhang, 2013), and statistics (Ma, 2013). Our motivation is two-fold. From a high level, compared to the convex programs, these HT-based algorithms are always orders of magnitude computationally more efficient, hence more practical for large-scale problems (Tropp and Wright, 2010). Nevertheless, they usually require a more stringent condition to guarantee the success. This naturally raises an interesting question of whether we can derive milder conditions for HT-based algorithms to achieve the best of the two worlds. For practitioners, to address the huge volume of data, a popular strategy in machine learning is to appeal to stochastic algorithms that sequentially update the solution. However, as many researchers observed (Langford et al., 2009; Duchi and Singer, 2009; Xiao, 2010), it is hard for the ℓ_1-based stochastic algorithms to preserve the sparse structure of the solution as the batch solvers do. This immediately poses the question of whether we are able to apply the principal idea of hard thresholding to stochastic algorithms while still ensuring a fast convergence.

To elaborate the problem more precisely, let us first turn to some basic properties of hard thresholding along with simple yet illustrative cases. For a general vector $\boldsymbol{b} \in \mathbb{R}^d$, the hard thresholded signal $\mathcal{H}_k(\boldsymbol{b})$ is formed by setting all but the largest (in magnitude) k elements of $\boldsymbol{b}$ to zero. Ties are broken lexicographically. Hence, the hard thresholded signal $\mathcal{H}_k(\boldsymbol{b})$ is always k-sparse, i.e., the number of non-zero components does not exceed k. Moreover, the resultant signal $\mathcal{H}_k(\boldsymbol{b})$ is a best

k-sparse approximation to $\boldsymbol{b}$ in terms of any ℓ_p norm ($p \geq 1$). That is, for any k-sparse vector $\boldsymbol{x}$

$$\|\mathcal{H}_k(\boldsymbol{b}) - \boldsymbol{b}\|_p \leq \|\boldsymbol{x} - \boldsymbol{b}\|_p.$$

In view of the above inequality, a broadly used bound in the literature for the deviation of the thresholded signal is as follows:

$$\|\mathcal{H}_k(\boldsymbol{b}) - \boldsymbol{x}\|_2 \leq 2\|\boldsymbol{b} - \boldsymbol{x}\|_2. \tag{1.1}$$

To gain intuition on the utility of (1.1) and to spell out the importance of offering a tight bound for it, let us consider the compressed sensing problem as an example for which we aim to recover the true sparse signal $\boldsymbol{x}$ from its linear measurements. Here, $\boldsymbol{b}$ is a good but dense approximation to $\boldsymbol{x}$ obtained by, e.g., full gradient descent. Then (1.1) justifies that in order to obtain a structured (i.e., sparse) approximation by hard thresholding, the distance of the iterate to the true signal $\boldsymbol{x}$ is upper bounded by a multiple of 2 to the one before. For comparison, it is worth mentioning that ℓ_1-based convex algorithms usually utilize the soft thresholding operator which enjoys the non-expansiveness property (Defazio et al., 2014), i.e., the iterate becomes closer to the optimum after projection. This salient feature might partially attribute to the wide range of applications of the ℓ_1-regularized formulations. Hence, to derive comparable performance guarantee, tightening the bound (1.1) is crucial in that it controls how much deviation the hard thresholding operator induces. This turns out to be more demanding for stochastic gradient methods, where the proxy $\boldsymbol{b}$ itself is affected by the randomness of sample realization. In other words, since $\boldsymbol{b}$ does not minimize the objective function (it only optimizes the objective in expectation), the deviation (1.1) makes it more challenging to analyze the convergence behavior. As an example, Nguyen et al. (2014) proposed a stochastic solver for general sparsity-constrained programs but suffered a non-vanishing optimization error due to randomness. This indicates that to mitigate the randomness barrier, we have to seek a better bound to control the precision of the thresholded solution and the variance.

1.1 Summary of Contributions

In this work, we make three contributions:

1. We examine the tightness of (1.1) that has been used for a decade in the literature and show that the equality therein will never be attained. We then improve this bound and quantitatively characterize that the deviation is inversely proportional to the value of $\sqrt{k}$. Our bound is tight, in the sense that the equality we build can be attained for specific signals, hence cannot be improved if no additional information is available. Our bound is universal in the sense that it holds for all choices of k-sparse signals $\boldsymbol{x}$ and for general signals $\boldsymbol{b}$.

2. Owing to the tight estimate, we demonstrate how the RIP (or RIP-like) condition assumed by a wide range of hard thresholding based algorithms can be relaxed. In the context of compressed sensing, it means that in essence, many more kinds of sensing matrices or fewer measurements can be utilized for data acquisition. For machine learning, it suggests that existing algorithms are capable of handling more difficult statistical models.

3. Finally, we present an computationally efficient algorithm that applies hard thresholding in large-scale setting and we prove its linear convergence to a global optimum up to the statistical precision of the problem. We also prove that with sufficient samples, our algorithm identifies the true parameter for prevalent statistical models. Returning to (1.1), our analysis shows that only when the deviation is controlled below the multiple of 1.15 can such an algorithm succeed. This immediately implies that the conventional bound (1.1) is not applicable in the challenging scenario.

1.2 Notation

Before delivering the algorithm and main theoretical results, let us instate several pieces of notation that are involved throughout the paper. We use bold lowercase letters, e.g., $\boldsymbol{v}$, to denote a vector (either column or row) and its ith element is denoted by v_i. The ℓ_2-norm of a vector $\boldsymbol{v}$ is denoted by $\|\boldsymbol{v}\|_2$. The support set of $\boldsymbol{v}$, i.e., indices of non-zeros, is denoted by $\operatorname{supp}(\boldsymbol{v})$ whose cardinality is written as $|\operatorname{supp}(\boldsymbol{v})|$ or $\|\boldsymbol{v}\|_0$. We write bold capital letters such as $\boldsymbol{M}$ for matrices and its (i,j)-th entry is denoted by m_{ij}. The capital upright letter C and its subscript variants (e.g., $\mathrm{C}_0, \mathrm{C}_1$) are reserved for absolute constants whose values may change from appearance to appearance.

For an integer $d > 0$, suppose that Ω is a subset of $\{1,\ 2,\ \ldots,\ d\}$. Then for a general vector $\boldsymbol{v} \in \mathbb{R}^d$, we define $\mathcal{P}_\Omega(\cdot)$ as the orthogonal projection onto the support set Ω which retains elements contained in Ω and sets others to zero. That is,

$$\left(\mathcal{P}_\Omega(\boldsymbol{v})\right)_i = \begin{cases} v_i, & \text{if } i \in \Omega, \\ 0, & \text{otherwise.} \end{cases}$$

In particular, let Γ be the support set indexing the k largest absolute components of $\boldsymbol{v}$. In this way, the hard thresholding operator is given by

$$\mathcal{H}_k(\boldsymbol{v}) = \mathcal{P}_\Gamma(\boldsymbol{v}).$$

We will also use the orthogonal projection of a vector $\boldsymbol{v}$ onto an ℓ_2-ball with radius ω. That is,

$$\Pi_\omega(\boldsymbol{v}) = \frac{\boldsymbol{v}}{\max\{1, \|\boldsymbol{v}\|_2/\omega\}}.$$

1.3 Roadmap

We present the key tight bound for hard thresholding in Section 2, along with a justification why the conventional bound (1.1) is not tight. We then discuss the implications of the developed tight bound to compressed sensing and machine learning in Section 3, which shows that the RIP or RIP-like condition can be improved for a number of popular algorithms. Thanks to our new estimation, Section 4 develops a novel stochastic algorithm which applies hard thresholding to large-scale problems and establishes the global linear convergence. A comprehensive empirical study on the tasks of sparse recovery and binary classification is carried out in Section 5. Finally, We conclude the paper in Section 6 and all the proofs are deferred to the appendix.

2. The Key Bound

We argue that the conventional bound (1.1) is not tight, in the sense that the equality therein can hardly be attained. To see this, recall how the bound was derived for a k-sparse signal $\boldsymbol{x}$ and a general one $\boldsymbol{b}$:

$$\|\mathcal{H}_k(\boldsymbol{b}) - \boldsymbol{x}\|_2 = \|\mathcal{H}_k(\boldsymbol{b}) - \boldsymbol{b} + \boldsymbol{b} - \boldsymbol{x}\|_2 \overset{\xi}{\leq} \|\mathcal{H}_k(\boldsymbol{b}) - \boldsymbol{b}\|_2 + \|\boldsymbol{b} - \boldsymbol{x}\|_2 \leq 2\|\boldsymbol{b} - \boldsymbol{x}\|_2,$$

where the last inequality holds because $\mathcal{H}_k(\boldsymbol{b})$ is a best k-sparse approximation to $\boldsymbol{b}$. The major issue occurs in ξ. Though it is the well-known triangle inequality and the equality could be attained if there is no restriction on the signals $\boldsymbol{x}$ and $\boldsymbol{b}$, we remind here that the signal $\boldsymbol{x}$ does have a specific structure – it is k-sparse. Note that in order to fulfill the equality in ξ, we must have $\mathcal{H}_k(\boldsymbol{b}) - \boldsymbol{b} = \gamma(\boldsymbol{b} - \boldsymbol{x})$ for some $\gamma \geq 0$, that is,

$$\mathcal{H}_k(\boldsymbol{b}) = (\gamma+1)\boldsymbol{b} - \gamma\boldsymbol{x}. \tag{2.1}$$

One may verify that the above equality holds *if and only if*

$$\boldsymbol{x} = \boldsymbol{b} = \mathcal{H}_k(\boldsymbol{b}). \tag{2.2}$$

To see this, let Ω be the support set of $\mathcal{H}_k(\boldsymbol{b})$ and $\overline{\Omega}$ be the complement. Let $\boldsymbol{b}_1 = \mathcal{P}_\Omega(\boldsymbol{b}) = \mathcal{H}_k(\boldsymbol{b})$ and $\boldsymbol{b}_2 = \mathcal{P}_{\overline{\Omega}}(\boldsymbol{b})$. Likewise, we define $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ as the components of $\boldsymbol{x}$ supported on Ω and $\overline{\Omega}$ respectively. Hence, (2.1) indicates $\boldsymbol{x}_1 = \boldsymbol{b}_1$ and $\boldsymbol{x}_2 = (1+\gamma^{-1})\boldsymbol{b}_2$ where we assume $\gamma > 0$ since $\gamma = 0$ immediately implies $\mathcal{H}_k(\boldsymbol{b}) = \boldsymbol{b}$ and hence the equality of (1.1) does not hold. If $\|\boldsymbol{b}_1\|_0 < k$, then we have $\boldsymbol{x}_2 = \boldsymbol{b}_2 = \boldsymbol{0}$ since $\boldsymbol{b}_1$ contains the k largest absolute elements of $\boldsymbol{b}$. Otherwise, the fact that $\|\boldsymbol{x}\|_0 \leq k$ and $\boldsymbol{x}_1 = \boldsymbol{b}_1$ implies $\boldsymbol{x}_2 = \boldsymbol{0}$, and hence $\boldsymbol{b}_2$. Therefore, we obtain (2.2).

When (2.2) happens, however, we in reality have $\|\mathcal{H}_k(\boldsymbol{b}) - \boldsymbol{x}\|_2 = \|\boldsymbol{b} - \boldsymbol{x}\|_2 = 0$. In other words, the factor of 2 in (1.1) can essentially be replaced with an *arbitrary constant*! In this sense, we conclude that the bound (1.1) is not tight. Our new estimate for hard thresholding is as follows:

Theorem 1 (Tight Bound for Hard Thresholding) *Let $\boldsymbol{b} \in \mathbb{R}^d$ be an arbitrary vector and $\boldsymbol{x} \in \mathbb{R}^d$ be any K-sparse signal. For any $k \geq K$, we have the following bound:*

$$\|\mathcal{H}_k(\boldsymbol{b}) - \boldsymbol{x}\|_2 \leq \sqrt{\nu}\,\|\boldsymbol{b} - \boldsymbol{x}\|_2, \quad \nu = 1 + \frac{\rho + \sqrt{(4+\rho)\,\rho}}{2}, \quad \rho = \frac{\min\{K, d-k\}}{k - K + \min\{K, d-k\}}.$$

In particular, our bound is tight in the sense that there exist specific vectors of $\boldsymbol{b}$ and $\boldsymbol{x}$ such that the equality holds.

Remark 2 (Maximum of ν) *In contrast to the constant bound* (1.1)*, our result asserts that the deviation resulting from hard thresholding is inversely proportional to $\sqrt{k}$ (when $K \leq d-k$) in a universal manner. When k tends to d, ρ is given by $(d-k)/(d-K)$ which is still decreasing with respect to k. Thus, the maximum value of ρ equals one. Even in this case, we find that $\sqrt{\nu_{\max}} = \sqrt{1 + \frac{\sqrt{5}+1}{2}} = \frac{\sqrt{5}+1}{2} \approx 1.618$.*

Remark 3 *Though for some batch algorithms such as IHT and CoSaMP, the constant bound* (1.1) *suffices to establish the convergence due to specific conditions, we show in Section 4 that it cannot ensure the global convergence for stochastic algorithms.*

Remark 4 *When $\boldsymbol{x}$ is not exactly K-sparse, we still can bound the error by $\|\mathcal{H}_k(\boldsymbol{b}) - \boldsymbol{x}\|_2 \leq \|\mathcal{H}_k(\boldsymbol{b}) - \mathcal{H}_k(\boldsymbol{x})\|_2 + \|\mathcal{H}_k(\boldsymbol{x}) - \boldsymbol{x}\|_2$. Thus, without loss of generality, we assumed that the signal $\boldsymbol{x}$ is K-sparse.*

Proof (Sketch) Our bound follows from fully exploring the sparsity pattern of the signals and from fundamental arguments in optimization. Denote

$$\boldsymbol{w} := \mathcal{H}_k(\boldsymbol{b}).$$

Let Ω be the support set of $\boldsymbol{w}$ and let $\overline{\Omega}$ be its complement. We immediately have $\mathcal{P}_\Omega(\boldsymbol{b}) = \boldsymbol{w}$. Let Ω' be the support set of $\boldsymbol{x}$. Define

$$\boldsymbol{b}_1 = \mathcal{P}_{\Omega\setminus\Omega'}(\boldsymbol{b}), \quad \boldsymbol{b}_2 = \mathcal{P}_{\Omega\cap\Omega'}(\boldsymbol{b}), \quad \boldsymbol{b}_3 = \mathcal{P}_{\overline{\Omega}\setminus\Omega'}(\boldsymbol{b}), \quad \boldsymbol{b}_4 = \mathcal{P}_{\overline{\Omega}\cap\Omega'}(\boldsymbol{b}).$$

Likewise, we define $\boldsymbol{x}_i$ and $\boldsymbol{w}_i$ for $1 \leq i \leq 4$. Due to the construction, we have $\boldsymbol{w}_1 = \boldsymbol{b}_1, \boldsymbol{w}_2 = \boldsymbol{b}_2, \boldsymbol{w}_3 = \boldsymbol{w}_4 = \boldsymbol{x}_1 = \boldsymbol{x}_3 = \boldsymbol{0}$. Our goal is to estimate the maximum value of $\|\boldsymbol{w} - \boldsymbol{x}\|_2^2 / \|\boldsymbol{b} - \boldsymbol{x}\|_2^2$. It is easy to show that when attaining the maximum, $\|\boldsymbol{b}_3\|_2$ must be zero. Denote

$$\gamma := \frac{\|\boldsymbol{w} - \boldsymbol{x}\|_2^2}{\|\boldsymbol{b} - \boldsymbol{x}\|_2^2} = \frac{\|\boldsymbol{b}_1\|_2^2 + \|\boldsymbol{b}_2 - \boldsymbol{x}_2\|_2^2 + \|\boldsymbol{x}_4\|_2^2}{\|\boldsymbol{b}_1\|_2^2 + \|\boldsymbol{b}_2 - \boldsymbol{x}_2\|_2^2 + \|\boldsymbol{b}_4 - \boldsymbol{x}_4\|_2^2}. \tag{2.3}$$

Note that the variables here only involve $\boldsymbol{x}$ and $\boldsymbol{b}$. Arranging the equation we obtain

$$(\gamma-1)\,\|\boldsymbol{b}_2 - \boldsymbol{x}_2\|_2^2 + \gamma\,\|\boldsymbol{b}_4 - \boldsymbol{x}_4\|_2^2 - \|\boldsymbol{x}_4\|_2^2 + (\gamma-1)\,\|\boldsymbol{b}_1\|_2^2 = 0. \tag{2.4}$$

It is evident that for specific choices of $\boldsymbol{b}$ and $\boldsymbol{x}$, we have $\gamma = 1$. Since we are interested in the maximum of γ, we assume $\gamma > 1$ below. Fixing $\boldsymbol{b}$, we can view the left-hand side of the above equation as a function of $\boldsymbol{x}$. One may verify that the function has a positive definite Hessian matrix and thus it attains the minimum at stationary point given by

$$\boldsymbol{x}_2^* = \boldsymbol{b}_2, \quad \boldsymbol{x}_4^* = \frac{\gamma}{\gamma-1}\boldsymbol{b}_4. \tag{2.5}$$

On the other hand, (2.4) implies that the minimum function value should not be greater than zero. Plugging the stationary point back gives

$$\|\boldsymbol{b}_1\|_2^2\gamma^2 - (2\,\|\boldsymbol{b}_1\|_2^2 + \|\boldsymbol{b}_4\|_2^2)\gamma + \|\boldsymbol{b}_1\|_2^2 \leq 0.$$

Solving the above inequality with respect to γ, we obtain

$$\gamma \leq 1 + \left(2\,\|\boldsymbol{b}_1\|_2^2\right)^{-1}\left(\|\boldsymbol{b}_4\|_2^2 + \sqrt{\left(4\,\|\boldsymbol{b}_1\|_2^2 + \|\boldsymbol{b}_4\|_2^2\right)\|\boldsymbol{b}_4\|_2^2}\right). \tag{2.6}$$

To derive an upper bound that is uniform over the choice of $\boldsymbol{b}$, we recall that $\boldsymbol{b}_1$ contains the largest absolute elements of $\boldsymbol{b}$ while $\boldsymbol{b}_4$ has smaller values. In particular, the average in $\boldsymbol{b}_1$ is larger than that in $\boldsymbol{b}_4$, which gives

$$\|\boldsymbol{b}_4\|_2^2/\|\boldsymbol{b}_4\|_0 \leq \|\boldsymbol{b}_1\|_2^2/\|\boldsymbol{b}_1\|_0.$$

Note that $\|\boldsymbol{b}_1\|_0 = k - \|\boldsymbol{b}_2\|_0 = k - (K - \|\boldsymbol{b}_4\|_0)$. Hence, combining with the fact that $0 \leq \|\boldsymbol{b}_4\|_0 \leq \min\{K, d-k\}$ and optimizing over $\|\boldsymbol{b}_4\|_0$ in the above inequality gives

$$\|\boldsymbol{b}_4\|_2^2 \leq \frac{\min\{K, d-k\}}{k - K + \min\{K, d-k\}}\,\|\boldsymbol{b}_1\|_2^2. \tag{2.7}$$

Finally, we arrive at a uniform upper bound

$$\gamma \leq 1 + \frac{\rho + \sqrt{(4+\rho)\,\rho}}{2}, \quad \rho = \frac{\min\{K, d-k\}}{k - K + \min\{K, d-k\}}.$$

See Appendix B for the full proof. ■

Remark 5 (Tightness) *We construct proper vectors $\boldsymbol{b}$ and $\boldsymbol{x}$ to establish the tightness of our bound by a backward induction. Note that γ equals ν if and only if $\|\boldsymbol{b}_4\|_2^2 = \rho\,\|\boldsymbol{b}_1\|_2^2$. Hence, we pick*

$$\|\boldsymbol{b}_4\|_2^2 = \rho\,\|\boldsymbol{b}_1\|_2^2, \quad \boldsymbol{x}_2 = \boldsymbol{b}_2, \quad \boldsymbol{x}_4 = \frac{\nu}{\nu-1}\boldsymbol{b}_4, \tag{2.8}$$

where $\boldsymbol{x}_2$ and $\boldsymbol{x}_4$ are actually chosen as the stationary point as in (2.5)*. We note that the quantity of ν only depends on d, k and K, not on the components of $\boldsymbol{b}$ or $\boldsymbol{x}$. Plugging the above back to* (2.3) *justifies $\gamma = \nu$.*

It remains to show that our choices in (2.8) *do not violate the definition of $\boldsymbol{b}_i$'s, i.e., we need to ensure that the elements in $\boldsymbol{b}_1$ or $\boldsymbol{b}_2$ are equal to or greater than those in $\boldsymbol{b}_3$ or $\boldsymbol{b}_4$. Note that there is no such constraint for the K-sparse vector $\boldsymbol{x}$. Let us consider the case $K < d-k$ and $\|\boldsymbol{b}_4\|_0 = K$, so that $\|\boldsymbol{b}_1\|_0 = k$ and $\rho = K/k$. Thus, the first equality of* (2.8) *holds as soon as all the entries of $\boldsymbol{b}$ have same magnitude. The fact $\|\boldsymbol{b}_4\|_0 = K$ also implies Ω' is a subset of $\overline{\Omega}$ due to the definition of $\boldsymbol{b}_4$ and the sparsity of $\boldsymbol{x}$, hence we have $\boldsymbol{x}_2 = \boldsymbol{0} = \boldsymbol{b}_2$. Finally, picking $\boldsymbol{x}_4$ as we did in* (2.8) *completes the reasoning since it does not violate the sparsity constraint on $\boldsymbol{x}$.*

As we pointed out and just verified, the bound given by Theorem 1 is tight. However, if there is additional information for the signals, a better bound can be established. For instance, let us further assume that the signal $\boldsymbol{b}$ is r-sparse. If $r \le k$, then $\boldsymbol{b}_4$ is a zero vector and (2.6) reads as $\gamma \le 1$. Otherwise, we have $\|\boldsymbol{b}_4\|_0 \le \min\{K, r-k\}$ and (2.7) is improved to

$$\|\boldsymbol{b}_4\|_2^2 \le \frac{\min\{K, r-k\}}{k - K + \min\{K, r-k\}}\,\|\boldsymbol{b}_1\|_2^2\,.$$

Henceforth, we can show that the parameter ρ is given by

$$\rho = \frac{\min\{K, r-k\}}{k - K + \min\{K, r-k\}}.$$

Note that the fact $r \le d$ implies that the above is a tighter bound than the one in Theorem 1.

We would also like to mention that in Lemma 1 of Jain et al. (2014), a closely related bound was established:

$$\|\mathcal{H}_k(\boldsymbol{b}) - \boldsymbol{b}\|_2 \le \sqrt{\frac{d-k}{d-K}}\,\|\boldsymbol{b} - \boldsymbol{x}\|_2\,. \tag{2.9}$$

One may use this nice result to show that

$$\|\mathcal{H}_k(\boldsymbol{b}) - \boldsymbol{x}\|_2 \le \|\mathcal{H}_k(\boldsymbol{b}) - \boldsymbol{b}\|_2 + \|\boldsymbol{b} - \boldsymbol{x}\|_2 \le \left(1 + \sqrt{\frac{d-k}{d-K}}\right)\|\boldsymbol{b} - \boldsymbol{x}\|_2\,, \tag{2.10}$$

which also improves on (1.1) provided $k > K$. However, one shortcoming of (2.10) is that the factor depends on the dimension. For comparison, we recall that in the regime $K \le d-k$, our bound is free of the dimension. This turns out to be a salient feature to integrate hard thresholding into stochastic methods, and we will comment on it more in Section 4.

3. Implications to Compressed Sensing

In this section, we investigate the implications of Theorem 1 for compressed sensing and signal processing. Since most of the HT-based algorithms utilize the deviation bound (1.1) to derive the convergence condition, they can be improved by our new bound. We exemplify the power of our theorem on two popular algorithms: IHT (Blumensath and Davies, 2009) and CoSaMP (Needell and Tropp, 2009). We note that our analysis also applies to their extensions such as Bahmani et al. (2013). To be clear, the purpose of this section is not dedicated to improving the best RIP condition for which recovery is possible by any methods (either convex or non-convex). Rather, we focus on two broadly used greedy algorithms and illustrate how our bound improves on previous results.

We proceed with a brief review of the problem setting in compressed sensing. Compressed sensing algorithms aim to recover the true K-sparse signal $\boldsymbol{x}^* \in \mathbb{R}^d$ from a set of its (perhaps noisy) measurements

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^* + \boldsymbol{\varepsilon}, \tag{3.1}$$

where $\boldsymbol{\varepsilon} \in \mathbb{R}^d$ is some observation noise and $\boldsymbol{A}$ is a known $n \times d$ sensing matrix with $n \ll d$, hence the name compressive sampling. In general, the model is not identifiable since it is an underdetermined system. Yet, the prior knowledge that $\boldsymbol{x}^*$ is sparse radically changes the premise. That is, if the geometry of the sparse signal is preserved under the action of the sampling matrix $\boldsymbol{A}$ for a restricted set of directions, then it is possible to invert the sampling process. Such a novel idea was quantified as the kth restricted isometry property of $\boldsymbol{A}$ by Candès and Tao (2005), which requires that there exists a constant $\delta \ge 0$, such that for all k-sparse signals $\boldsymbol{x}$

$$(1-\delta)\,\|\boldsymbol{x}\|_2^2 \le \|\boldsymbol{A}\boldsymbol{x}\|_2^2 \le (1+\delta)\,\|\boldsymbol{x}\|_2^2\,. \tag{3.2}$$

The kth restricted isometry constant (RIC) δ_k is then defined as the smallest one that satisfies the above inequalities. Note that $\delta_{2k} < 1$ is the minimum requirement for distinguishing all k-sparse signals from the measurements. This is because for two arbitrary k-sparse vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ and their respective measurements $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$, the RIP condition reads as

$$(1-\delta_{2k})\,\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2 \le \|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_2^2 \le (1+\delta_{2k})\,\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2\,,$$

for which $\delta_{2k} < 1$ guarantees that $\boldsymbol{x}_1 \ne \boldsymbol{x}_2$ implies $\boldsymbol{y}_1 \ne \boldsymbol{y}_2$. To date, there are three quintessential examples known to exhibit a profound restricted isometry behavior as long as the number of measurements is large enough: Gaussian matrices (optimal RIP, i.e., very small δ_k), partial Fourier matrices (fast computation) and Bernoulli ensembles (low memory footprint). Notably, it was shown in recent work that random matrices with a heavy-tailed distribution also satisfy the RIP with overwhelming probability (Adamczak et al., 2011; Li et al., 2014).

Equipped with the standard RIP condition, many efficient algorithms have been developed. A partial list includes ℓ_1-norm based convex programs, IHT, CoSaMP, SP and regularized OMP (Needell and Vershynin, 2010), along with much interesting work devoted to improving or sharpening the RIP condition (Wang and Shim, 2012; Mo and Shen, 2012; Cai and Zhang, 2013; Mo, 2015). To see why relaxing RIP is of central interest, note that the standard result (Baraniuk et al., 2008) asserts that the RIP condition $\delta_k \le \delta$ holds with high probability over the draw of $\boldsymbol{A}$ provided

$$n \ge \mathrm{C}_0\delta^{-2}k\log(d/k). \tag{3.3}$$

Hence, a slight relaxation of the condition $\delta_k \le \delta$ may dramatically decrease the number of measurements. That being said, since the constant C_0 above is unknown, in general one cannot tell

the precise sample size for greedy algorithms. Estimating the constant is actually the theme of phase transition (Donoho and Tanner, 2010; Donoho et al., 2013). While precise phase transition for ℓ_1-based convex programs has been well understood (Wainwright, 2009), an analogous result for greedy algorithms remains an open problem. Notably, in Blanchard and Tanner (2015), phase transition for IHT/CoSaMP was derived using the constant bound (1.1). We believe that our tight bound shall sharpen these results and we leave it as our future work. In the present paper, we focus on the ubiquitous RIP condition. In the language of RIP, we establish improved results.

3.1 Iterative Hard Thresholding

The IHT algorithm recovers the underlying K-sparse signal $\boldsymbol{x}^*$ by iteratively performing a full gradient descent on the least-squares loss followed by a hard thresholding step. That is, IHT starts with an arbitrary point $\boldsymbol{x}^0$ and at the t-th iteration, it updates the new solution as follows:

$$\boldsymbol{x}^t = \mathcal{H}_k\left(\boldsymbol{x}^{t-1} + \boldsymbol{A}^\top(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^{t-1})\right). \tag{3.4}$$

Note that Blumensath and Davies (2009) used the parameter $k = K$. However, in practice one may only know to an upper bound on the true sparsity K. Thus, we consider the projection sparsity k as a parameter that depends on K. To establish the global convergence with a geometric rate of 0.5, Blumensath and Davies (2009) applied the bound (1.1) and assumed the RIP condition

$$\delta_{2k+K} \le 0.18. \tag{3.5}$$

As we have shown, (1.1) is actually not tight and hence, their results, especially the RIP condition can be improved by Theorem 1.

Theorem 6 *Consider the model* (3.1) *and the IHT algorithm* (3.4)*. Pick $k \ge K$ and let $\{\boldsymbol{x}^t\}_{t\ge1}$ be the iterates produced by IHT. Then, under the RIP condition $\delta_{2k+K} \le 1/\sqrt{8\nu}$, for all $t \ge 1$*

$$\left\|\boldsymbol{x}^t - \boldsymbol{x}^*\right\|_2 \le 0.5^t \left\|\boldsymbol{x}^0 - \boldsymbol{x}^*\right\|_2 + \mathrm{C}\left\|\boldsymbol{\varepsilon}\right\|_2,$$

where ν is given by Theorem 1.

Let us first study the vanilla case $k = K$. Blumensath and Davies (2009) required $\delta_{3K} \le 0.18$ whereas our analysis shows $\delta_{3K} \le 0.22$ suffices. Note that even a little relaxation on RIP is challenging and may require several pages of mathematical induction (Candès, 2008; Cai et al., 2010; Foucart, 2012). In contrast, our improvement comes from a direct application of Theorem 1 which only modifies several lines of the original proof in Blumensath and Davies (2009). See Appendix C for details. In view of (3.3), we find that the necessary number of measurements for IHT is dramatically reduced with a factor of 0.67 by our new theorem in that the minimum requirement of n is inversely proportional to the square of δ_{2k+K}.

Another important consequence of the theorem is a characterization on the RIP condition and the sparsity parameter, which, to the best of our knowledge, has not been studied in the literature. In Blumensath and Davies (2009), when gradually tuning k larger than K, it always requires $\delta_{2k+K} \le 0.18$. Note that due to the monotonicity of RIC, i.e., $\delta_r \le \delta_{r'}$ if $r \le r'$, the condition turns out to be more and more stringent. Compared to their result, since ν is inversely proportional to $\sqrt{k}$, Theorem 6 is powerful especially when k becomes larger. For example, suppose $k = 20K$. In this

case, Theorem 6 justifies that IHT admits the linear convergence as soon as $\delta_{41K} \le 0.32$ whereas Blumensath and Davies (2009) requires $\delta_{41K} \le 0.18$. Such a property is appealing in practice, in that among various real-world applications, the true sparsity is indeed unknown and we would like to estimate a conservative upper bound on it.

On the other hand, for a given sensing matrix, there does exist a fundamental limit for the maximum choice of k. To be more precise, the condition in Theorem 6 together with the probabilistic argument (3.3) require

$$1/\sqrt{8\nu} \ge \delta_{2k+K}, \quad \mathrm{C}_1\nu(2k+K)\log\left(d/(2k+K)\right) \le n.$$

Although it could be very interesting to derive a quantitative characterization for the maximum value of k, we argue that it is perhaps intractable owing to two aspects: First, it is known that one has to enumerate all the combinations of the $2k+K$ columns of $\boldsymbol{A}$ to compute the restricted isometry constant δ_{2k+K} (Bah and Tanner, 2010, 2014). This suggests that it is NP-hard to estimate the largest admissible value of k. Also, there is no analytic solution of the stationary point for the left-hand side of the second inequality.

3.2 Compressive Sampling Matching Pursuit

The CoSaMP algorithm proposed by Needell and Tropp (2009) is one of the most efficient algorithms for sparse recovery. Let $F(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2$. CoSaMP starts from an arbitrary initial point $\boldsymbol{x}^0$ and proceeds as follows:

$$\begin{aligned}
\Omega^t &= \mathrm{supp}\left(\nabla F(\boldsymbol{x}^{t-1}),\ k\right) \cup \mathrm{supp}\left(\boldsymbol{x}^{t-1}\right),\\
\boldsymbol{b}^t &= \arg\min_{\boldsymbol{x}}\ F(\boldsymbol{x}),\ \text{s.t. supp}\left(\boldsymbol{x}\right) \subset \Omega^t,\\
\boldsymbol{x}^t &= \mathcal{H}_k\left(\boldsymbol{b}^t\right).
\end{aligned}$$

Compared to IHT which performs hard thresholding after gradient update, CoSaMP prunes the gradient at the beginning of each iteration, followed by solving a least-squares program restricted on a small support set. In particular, in the last step, CoSaMP applies hard thresholding to form a k-sparse iterate for future updates. The analysis of CoSaMP consists of bounding the estimation error in each step. Owing to Theorem 1, we advance the theoretical result of CoSaMP by improving the error bound for its last step, and hence the RIP condition.

Theorem 7 *Consider the model* (3.1) *and the CoSaMP algorithm. Pick $k \ge K$ and let $\{\boldsymbol{x}^t\}_{t\ge1}$ be the iterates produced by CoSaMP. Then, under the RIP condition*

$$\delta_{3k+K} \le \frac{\left(\sqrt{32\nu+49}-9\right)^{1/2}}{4\sqrt{\nu-1}},$$

it holds that for all $t \ge 1$

$$\left\|\boldsymbol{x}^t - \boldsymbol{x}^*\right\|_2 \le 0.5^t \left\|\boldsymbol{x}^0 - \boldsymbol{x}^*\right\|_2 + \mathrm{C}\left\|\boldsymbol{\varepsilon}\right\|_2,$$

where ν is given by Theorem 1.

Roughly speaking, the bound is still inversely proportional to $\sqrt{\nu}$. Hence, it is monotonically increasing with respect to k, indicating our theorem is more effective for a large quantity of k. In fact, for the CoSaMP algorithm, our bound above is superior to the best known result even when $k = K$. To see this, we have the RIP condition $\delta_{4K} \leq 0.31$. In comparison, Needell and Tropp (2009) derived a bound $\delta_{4K} \leq 0.1$ and Foucart and Rauhut (2013, Theorem 6.27) improved it to $\delta_{4K} < 0.29$ for a geometric rate of 0.5. We notice that for binary sparse vectors, Jain et al. (2014) presented a different proof technique and obtained the RIP condition $\delta_{4K} \leq 0.35$ for CoSaMP.

4. Hard Thresholding in Large-Scale Optimization

Now we move on to the machine learning setting where our focus is pursuing an optimal sparse solution that minimizes a given objective function based on a set of training samples $Z_1^n := \{Z_i\}_{i=1}^n$. Different from compressed sensing, we usually have sufficient samples which means n can be very large. Therefore, the computational complexity is of primary interest. Formally, we are interested in optimizing the following program:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} F(\boldsymbol{x}; Z_1^n) = \frac{1}{n}\sum_{i=1}^n f(\boldsymbol{x}; Z_i), \quad \text{s.t. } \|\boldsymbol{x}\|_0 \leq K,\ \|\boldsymbol{x}\|_2 \leq \omega. \tag{4.1}$$

The global optimum of the above problem is denoted by $\boldsymbol{x}_{\rm opt}$. We note that the objective function is presumed to be decomposable with respect to the samples. This is quite a mild condition and most of the popular machine learning models fulfill it. Typical examples include (but not limited to) the sparse linear regression and sparse logistic regression:

- Sparse Linear Regression: For all $1 \leq i \leq n$, we have $Z_i = (\boldsymbol{a}_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$ and the loss function $F(\boldsymbol{x}; Z_1^n) = \frac{1}{2n}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2$ is the least-squares and can be explained by $f(\boldsymbol{x}; Z_i) = \frac{1}{2}\|\boldsymbol{a}_i \cdot \boldsymbol{x} - y_i\|_2^2$.
- Sparse Logistic Regression: For all $1 \leq i \leq n$, we have $Z_i = (\boldsymbol{a}_i, y_i) \in \mathbb{R}^d \times \{+1, -1\}$ and the negative log-likelihood is penalized, i.e., $F(\boldsymbol{x}; Z_1^n) = \frac{1}{n}\sum_{i=1}^n \log\left(1 + \exp\left(-y_i \boldsymbol{a}_i \cdot \boldsymbol{x}\right)\right)$ for which $f(\boldsymbol{x}; Z_i) = \log\left(1 + \exp\left(-y_i \boldsymbol{a}_i \cdot \boldsymbol{x}\right)\right)$.

To ease notation, we will often write $F(\boldsymbol{x}; Z_1^n)$ as $F(\boldsymbol{x})$ and $f(\boldsymbol{x}; Z_i)$ as $f_i(\boldsymbol{x})$ for $i = 1, 2, \cdots, n$. It is worth mentioning that the objective function $F(\boldsymbol{x})$ is allowed to be non-convex. Hence, in order to ensure the existence of a global optimum, a natural option is to impose an ℓ_p-norm ($p \geq 1$) constraint (Loh and Wainwright, 2012, 2015). Here we choose the ℓ_2-norm constraint owing to its fast projection. Previous work, e.g., Agarwal et al. (2012) prefers the computationally less efficient ℓ_1-norm to promote sparsity and to guarantee the existence of optimum. In our problem, yet, we already have imposed the hard sparsity constraint so the ℓ_2-norm constraint is a better fit.

The major contribution of this section is a computationally efficient algorithm termed hard thresholded stochastic variance reduced gradient method (HT-SVRG) to optimize (4.1), tackling one of the most important problems in large-scale machine learning: producing sparse solutions by stochastic methods. We emphasize that the formulation (4.1) is in stark contrast to the ℓ_1-regularized programs considered by previous stochastic solvers such as Prox-SVRG (Xiao and Zhang, 2014) and SAGA (Defazio et al., 2014). We target here a stochastic algorithm for the *non-convex* problem that is less exploited in the literature. From a theoretical perspective, (4.1) is more difficult to analyze but it always produces sparse solutions, whereas performance guarantees for convex programs

are fruitful but one cannot characterize the sparsity of the obtained solution (usually the solution is not sparse). When we appeal to stochastic algorithms to solve the convex programs, the ℓ_1-norm formulation becomes much less effective in terms of sparsification, naturally owing to the randomness. See Langford et al. (2009); Xiao (2010); Duchi and Singer (2009) for more detailed discussion on the issue. We also remark that existing work such as Yuan et al. (2018); Bahmani et al. (2013); Jain et al. (2014) investigated the sparsity-constrained problem (4.1) in a batch scenario, which is not practical for large-scale learning problems. The perhaps most related work to our new algorithm is Nguyen et al. (2014). Nonetheless, the optimization error therein does not vanish for noisy statistical models.

Our main result shows that for prevalent statistical models, our algorithm is able to recover the true parameter with a linear rate. Readers should distinguish the optimal solution $\boldsymbol{x}_{\rm opt}$ and the true parameter. For instance, consider the model (3.1). Minimizing (4.1) does not amount to recovering $\boldsymbol{x}^*$ if there is observation noise. In fact, the convergence to $\boldsymbol{x}_{\rm opt}$ is only guaranteed to an accuracy reflected by the *statistical precision* of the problem, i.e., $\|\boldsymbol{x}^* - \boldsymbol{x}_{\rm opt}\|_2$, which is the best one can hope for any statistical model (Agarwal et al., 2012). We find that the global convergence is attributed to both the tight bound and the variance reduction technique to be introduced below, and examining the necessity of them is an interesting future work.

Algorithm 1 Hard Thresholded Stochastic Variance Reduced Gradient Method (HT-SVRG)

Require: Training samples $\{Z_i\}_{i=1}^n$, maximum stage count S, sparsity parameter k, update frequency m, learning rate η, radius ω, initial solution $\widetilde{\boldsymbol{x}}^0$.
Ensure: Optimal solution $\widetilde{\boldsymbol{x}}^S$.

```
for s = 1 to S do
    Set x̃ = x̃^{s-1}, μ̃ = (1/n) Σ_{i=1}^n ∇f_i(x̃), x^0 = x̃.
    for t = 1 to m do
        Uniformly pick i_t ∈ {1, 2, ··· , n} and update the solution
            b^t = x^{t-1} − η (∇f_{i_t}(x^{t-1}) − ∇f_{i_t}(x̃) + μ̃) ,
            r^t = H_k (b^t) ,
            x^t = Π_ω(r^t).
    end for
    Uniformly choose j^s ∈ {0, 1, ··· , m − 1} and set x̃^s = x^{j^s}.
end for
```

4.1 Algorithm

Our algorithm (Algorithm 1) applies the framework of Johnson and Zhang (2013), where the primary idea is to leverage past gradients for the current update for the sake of variance reduction – a technique that has a long history in statistics (Owen and Zhou, 2000). To guarantee that each iterate is k-sparse, it then invokes the hard thresholding operation. Note that the orthogonal projection for $\boldsymbol{r}^t$ will not change the support set, and hence $\boldsymbol{x}^t$ is still k-sparse. Also note that our sparsity constraint in (4.1) reads as $\|\boldsymbol{x}\|_0 \leq K$. What we will show below is that when the parameter k is properly chosen (which depends on K), we obtain a globally convergent sequence of iterates.

The most challenging part on establishing the global convergence comes from the hard thresholding operation $\mathcal{H}_k\left(\boldsymbol{r}^t\right)$. Note that it is $\boldsymbol{b}^t$ that reduces the objective value in expectation. If $\boldsymbol{b}^t$ is not k-sparse (usually it is dense), $\boldsymbol{x}^t$ is not equal to $\boldsymbol{b}^t$ so it does not decrease the objective function. In addition, compared with the convex proximal operator (Defazio et al., 2014) which enjoys the non-expansiveness of the distance to the optimum, the hard thresholding step can enlarge the distance up to a multiple of 2 if using the bound (1.1). What makes it a more serious issue is that these inaccurate iterates $\boldsymbol{x}^t$ will be used for future updates, and hence the error might be progressively propagated at an exponential rate.

Our key idea is to first bound the curvature of the function from below and above to establish RIP-like condition, which, combined with Theorem 1, downscales the deviation resulting from hard thresholding. Note that ν is always greater than one (see Theorem 1), hence the curvature bound is necessary. Due to variance reduction, we show that the optimization error vanishes when restricted on a small set of directions as soon as we have sufficient samples. Moreover, with hard thresholding we are able to control the error per iteration and to obtain near-optimal sample complexity.

4.2 Deterministic Analysis

We will first establish a general theorem that characterizes the progress of HT-SVRG for approximating an arbitrary K-sparse signal $\widehat{\boldsymbol{x}}$. Then we will discuss how to properly choose the hyper-parameters of the algorithm. Finally we move on to specify $\widehat{\boldsymbol{x}}$ to develop convergence results for a global optimum of (4.1) and for a true parameter (e.g., $\boldsymbol{x}^*$ of the compressed sensing problem).

4.2.1 ASSUMPTION

Our analysis depends on two properties of the curvature of the objective function that have been standard in the literature. Readers may refer to Bickel et al. (2009); Negahban et al. (2009); Jain et al. (2014) for a detailed description.

Definition 8 (Restricted Strong Convexity) *A differentiable function $g:\ \mathbb{R}^d \to \mathbb{R}$ is said to satisfy the property of restricted strong convexity (RSC) with parameter $\alpha_r > 0$, if for all vectors $\boldsymbol{x}$, $\boldsymbol{x}' \in \mathbb{R}^d$ with $\|\boldsymbol{x}-\boldsymbol{x}'\|_0 \le r$, it holds that*

$$g(\boldsymbol{x}') - g(\boldsymbol{x}) - \left\langle \nabla g(\boldsymbol{x}), \boldsymbol{x}'-\boldsymbol{x}\right\rangle \ge \frac{\alpha_r}{2}\left\|\boldsymbol{x}'-\boldsymbol{x}\right\|_2^2.$$

Definition 9 (Restricted Smoothness) *A differentiable function $g:\ \mathbb{R}^d \to \mathbb{R}$ is said to satisfy the property of restricted smoothness (RSS) with parameter $L_r > 0$, if for all vectors $\boldsymbol{x}$, $\boldsymbol{x}' \in \mathbb{R}^d$ with $\|\boldsymbol{x}-\boldsymbol{x}'\|_0 \le r$, it holds that*

$$\left\|\nabla g(\boldsymbol{x}') - \nabla g(\boldsymbol{x})\right\|_2 \le L_r\left\|\boldsymbol{x}'-\boldsymbol{x}\right\|_2.$$

With these definitions, we assume the following:

$(A1)$ $F(\boldsymbol{x})$ satisfies the RSC condition with parameter α_{k+K}.

$(A2)$ For all $1 \le i \le n$, $f_i(\boldsymbol{x})$ satisfies the RSS condition with parameter L_{3k+K}.

Here, we recall that K was first introduced in (4.1) and the parameter k was used in our algorithm. Compared to the convex algorithms such as SAG (Roux et al., 2012), SVRG (Johnson and Zhang, 2013) and SAGA (Defazio et al., 2014) that assume strong convexity and smoothness everywhere, we only assume these in a restricted sense. This is more practical especially in the high dimensional regime where the Hessian matrix could be degenerate (Agarwal et al., 2012). We also stress that the RSS condition is imposed on each $f_i(\boldsymbol{x})$, whereas prior work requires it for $F(\boldsymbol{x})$ which is milder than ours (Negahban et al., 2009).

4.2.2 UPPER BOUND OF PROGRESS

For brevity, let us denote

$$L := L_{3k+K}, \quad \alpha := \alpha_{k+K}, \quad c := L/\alpha,$$

where we call the quantity c as the condition number of the problem. It is also crucial to measure the ℓ_2-norm of the gradient restricted on sparse directions, and we write

$$\left\|\nabla_{3k+K}F(\boldsymbol{x})\right\|_2 := \max_{\Omega}\left\{\left\|\mathcal{P}_{\Omega}\left(\nabla F(\boldsymbol{x})\right)\right\|_2:\ |\Omega| \le 3k+K\right\}.$$

Note that for convex programs, the above evaluated at a global optimum is zero. As will be clear, $\left\|\nabla_{3k+K}F(\boldsymbol{x})\right\|_2$ reflects how close the iterates returned by HT-SVRG can be to the point $\boldsymbol{x}$. For prevalent statistical models, it vanishes when there are sufficient samples. Related to this quantity, our analysis also involves

$$Q(\boldsymbol{x}) := \left(16\nu\eta^2 L\omega m + \frac{2\omega}{\alpha}\right)\left\|\nabla_{3k+K}F(\boldsymbol{x})\right\|_2 + 4\nu\eta^2 m\left\|\nabla_{3k+K}F(\boldsymbol{x})\right\|_2^2,$$

where we recall that ν is the expansiveness factor given by Theorem 1, η and m are used in the algorithm and ω is a universal constant that upper bounds the ℓ_2-norm of the signal we hope to estimate. Virtually, with an appropriate parameter setting, $Q(\boldsymbol{x})$ scales as $\left\|\nabla_{3k+K}F(\boldsymbol{x})\right\|_2$ which will be clarified. For a particular stage s, we denote $\mathcal{I}^s := \{i_1, i_2, \cdots, i_m\}$, i.e., the samples randomly chosen for updating the solution.

Theorem 10 *Consider Algorithm 1 and a K-sparse signal $\widehat{\boldsymbol{x}}$ of interest. Assume $(A1)$ and $(A2)$. Pick the step size $0 < \eta < 1/(4L)$. If $\nu < 4L/(4L-\alpha)$, then it holds that*

$$\mathbb{E}\left[F(\widetilde{\boldsymbol{x}}^s) - F(\widehat{\boldsymbol{x}})\right] \le \beta^s\left[F(\widetilde{\boldsymbol{x}}^0) - F(\widehat{\boldsymbol{x}})\right] + \tau(\widehat{\boldsymbol{x}}),$$

where the expectation is taken over $\{\mathcal{I}^1, j^1, \mathcal{I}^2, j^2, \cdots, \mathcal{I}^s, j^s\}$ and $0 < \beta < 1$ provided that m is large enough. In particular, for $1/(1-\eta\alpha) < \nu < 4L/(4L-\alpha)$, we have

$$\beta = \beta_1 := \frac{1}{\left(2\nu\eta\alpha - 2\nu\eta^2\alpha L - \nu + 1\right)m} + \frac{2\nu\eta^2\alpha L}{2\nu\eta\alpha - 2\nu\eta^2\alpha L - \nu + 1},$$
$$\tau(\widehat{\boldsymbol{x}}) = \tau_1(\widehat{\boldsymbol{x}}) := \frac{\alpha Q(\widehat{\boldsymbol{x}})}{2(2\nu\eta\alpha - 2\nu\eta^2\alpha L - \nu + 1)(1-\beta_1)m}.$$

For $\nu \le 1/(1-\eta\alpha)$, we have

$$\beta = \beta_2 := \frac{1}{\nu\eta\alpha(1-2\eta L)m} + \frac{2\eta L}{1-2\eta L}, \quad \tau(\widehat{\boldsymbol{x}}) = \tau_2(\widehat{\boldsymbol{x}}) := \frac{Q(\widehat{\boldsymbol{x}})}{2\nu\eta\alpha(1-2\eta L)(1-\beta_2)m}.$$

The proof can be found in Appendix D.1.

Remark 11 *For the theorem to hold,* $\sqrt{\nu} < \sqrt{4L/(4L-\alpha)} \leq \sqrt{4/3} \approx 1.15$ *due to* $L \geq \alpha$. *Hence, the conventional bound* (1.1) *is not applicable. In contrast, Theorem 1 asserts that this condition can be fulfilled by tuning* k *slightly larger than* K.

Remark 12 *With the conditions on* η *and* ν, *the coefficient* β *is always less than one provided that* m *is sufficiently large.*

Remark 13 *The theorem does* not *assert convergence to an arbitrary sparse vector* $\widehat{\boldsymbol{x}}$. *This is because* $F(\widetilde{\boldsymbol{x}}^s) - F(\widehat{\boldsymbol{x}})$ *might be less than zero. However, specifying* $\widehat{\boldsymbol{x}}$ *does give convergence results, as to be elaborated later.*

4.2.3 HYPER-PARAMETER SETTING

Before moving on to the convergence guarantee, let us discuss the minimum requirement on the hyper-parameters k, m and η, and determine how to choose them to simplify Theorem 10.

For the sake of success of HT-SVRG, we require $\nu < 4c/(4c-1)$, which implies $\rho < 1/(16c^2 - 4c)$. Recall that ρ is given in Theorem 1. In general, we are interested in the regime $K \leq k \ll d$. Hence, we have $\rho = K/k$ and the minimum requirement for the sparsity parameter is

$$k > (16c^2 - 4c)K. \tag{4.2}$$

To our knowledge, the idea of relaxed sparsity was first introduced in Zhang (2011) for OMP and in Jain et al. (2014) for projected gradient descent. However, the relaxed sparsity here emerges in a different way in that HT-SVRG is a stochastic algorithm, and their proof technique cannot be used.

We also contrast our tight bound to the inequality (2.10) that is obtained by combining the triangle inequality and Lemma 1 of Jain et al. (2014). Following our proof pipeline, (2.10) gives

$$k \geq \left(1 - \left(\sqrt{4c(4c-1)^{-1}} - 1\right)^2\right) d + \left(\sqrt{4c(4c-1)^{-1}} - 1\right)^2 K$$

which grows with the dimension d, whereas using Theorem 1 the sparsity parameter k depends only on the desired sparsity K. In this regard, we conclude that for the stochastic case, our bound is vital.

Another component of the algorithm is the update frequency m. Intuitively, HT-SVRG performs m number of stochastic gradient update followed by a full gradient evaluation, in order to mitigate the variance. In this light, m should not be too small. Otherwise, the algorithm reduces to the full gradient method which is not computationally efficient. On the other spectrum, a large m leads to a slow convergence that is reflected in the convergence coefficient β. To quantitatively analyze how m should be selected, let us consider the case $\nu \leq 1/(1-\eta\alpha)$ for example. The case $1/(1-\eta\alpha) < \nu < 4L/(4L-\alpha)$ follows in a similar way. In order to ensure $\beta_2 < 1$, we must have $m > 1/\left(\nu\eta\alpha(1-4\eta L)\right)$. In particular, picking

$$\eta = \frac{\eta'}{L}, \quad \eta' \in (0, 1/4), \tag{4.3}$$

we find that the update frequency m has to satisfy

$$m > \frac{c}{\nu\eta'(1-\eta')}, \tag{4.4}$$

which is of the same order as in the convex case (Johnson and Zhang, 2013) when $\eta' = \Theta(1)$. Note that the way we choose the learning rate $\eta = \eta'/L$ is also a common practice in convex optimization (Nesterov, 2004).

With (4.2), (4.3) and (4.4) in mind, we provide detailed choices of the hyper-parameters. Due to $0 < \eta < 1/(4L)$, β_1 is monotonically increasing with respect to ν. By Theorem 1, we know that ν is decreasing with respect to k. Thus, a larger quantity of k results in a smaller value of β_1, and hence a faster rate. Interestingly, for β_2 we discover that the smaller the k is, the faster the algorithm concentrates. Hence, we have the following consequence:

Proposition 14 *Fix* η *and* m. *Then the optimal choice of* ν *in Theorem 10 is* $\nu = 1/(1-\eta\alpha)$ *in the sense that the convergence coefficient* β *attains the minimum.*

In light of the proposition, in the sections to follow, we will only consider the setting $\nu = 1/(1-\eta\alpha)$. But we emphasize that our analysis and results essentially apply to any $\nu \leq 4L/(4L-\alpha)$.

Now let

$$\eta = \frac{1}{8L}, \quad m = 4(8c-1), \quad k = 8c(8c-1)K. \tag{4.5}$$

This gives

$$\beta = \frac{2}{3}, \quad \tau(\widehat{\boldsymbol{x}}) = \frac{5\omega}{\alpha}\left\|\nabla_{3k+K}F(\widehat{\boldsymbol{x}})\right\|_2 + \frac{1}{\alpha L}\left\|\nabla_{3k+K}F(\widehat{\boldsymbol{x}})\right\|_2^2. \tag{4.6}$$

4.2.4 GLOBAL LINEAR CONVERGENCE

We are in the position to state the global linear convergence to an optimum of the sparsity-constrained optimization program (4.1).

Corollary 15 *Assume* $(A1)$ *and* $(A2)$. *Consider the HT-SVRG algorithm with hyper-parameters given in* (4.5). *Then the sequence* $\{\widetilde{\boldsymbol{x}}^s\}_{s\geq 1}$ *converges linearly to a global optimum* $\boldsymbol{x}_{\rm opt}$ *of* (4.1)

$$\begin{aligned}\mathbb{E}\left[F(\widetilde{\boldsymbol{x}}^s) - F(\boldsymbol{x}_{\rm opt})\right] \leq &\left(\frac{2}{3}\right)^s \left[F(\widetilde{\boldsymbol{x}}^0) - F(\boldsymbol{x}_{\rm opt})\right] \\ &+ \frac{5\omega}{\alpha}\left\|\nabla_{3k+K}F(\boldsymbol{x}_{\rm opt})\right\|_2 + \frac{1}{\alpha L}\left\|\nabla_{3k+K}F(\boldsymbol{x}_{\rm opt})\right\|_2^2.\end{aligned}$$

Proof This is a direct consequence of Theorem 10. ■

Whenever $\nabla_{3k+K}F(\boldsymbol{x}_{\rm opt}) = \boldsymbol{0}$, the corollary reads as

$$\mathbb{E}\left[F(\widetilde{\boldsymbol{x}}^s) - F(\boldsymbol{x}_{\rm opt})\right] \leq \left(\frac{2}{3}\right)^s \left[F(\widetilde{\boldsymbol{x}}^0) - F(\boldsymbol{x}_{\rm opt})\right].$$

It implies that if one is solving a convex problem without the sparsity constraint but the optimal solution happens to be sparse, it is safe to perform hard thresholding without loss of optimality. We exemplify such behavior with another algorithm SAGA (Defazio et al., 2014) in Appendix E. In the noiseless compressed sensing setting where $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^*$, the corollary guarantees that HT-SVRG exactly recovers the underlying true signal $\boldsymbol{x}^*$ when $F(\boldsymbol{x})$ is chosen as the least-squares loss in that $\boldsymbol{x}_{\rm opt} = \boldsymbol{x}^*$ and $\nabla F(\boldsymbol{x}^*) = \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{y}) = \boldsymbol{0}$.

On the other side, the RSC property implies that

$$\left\|\widetilde{\boldsymbol{x}}^s - \widehat{\boldsymbol{x}}\right\|_2 \leq \sqrt{\frac{2\max\{F(\widetilde{\boldsymbol{x}}^s) - F(\widehat{\boldsymbol{x}}), 0\}}{\alpha}} + \frac{2\left\|\nabla_{k+K}F(\widehat{\boldsymbol{x}})\right\|_2}{\alpha}.$$

The proof is straightforward and can be found in Lemma 14 of Shen and Li (2017a). Now we specify $\widehat{\boldsymbol{x}}$ as the true parameter of some statistical model, for instance, $\boldsymbol{x}^*$ in (3.1). It is hence possible to establish recovery guarantee of $\boldsymbol{x}^*$, which is known as the problem of parameter estimation.

Corollary 16 *Assume* $(A1)$ *and* $(A2)$. *Let* L' *be the RSS parameter of* $F(\boldsymbol{x})$ *at the sparsity level* $3k+K$. *Consider the HT-SVRG algorithm with hyper-parameters given in* (4.5). *Then the sequence* $\{\widetilde{\boldsymbol{x}}^s\}_{s\geq 1}$ *recovers a* K*-sparse signal* $\boldsymbol{x}^*$ *with a geometric rate*

$$\mathbb{E}\left[\left\|\widetilde{\boldsymbol{x}}^s-\boldsymbol{x}^*\right\|_2\right] \leq \sqrt{\frac{2L'}{\alpha}}\cdot\left(\frac{2}{3}\right)^{\frac{s}{2}}\left\|\widetilde{\boldsymbol{x}}^0-\boldsymbol{x}^*\right\|_2+\sqrt{\frac{10\omega}{\alpha^2}\left\|\nabla_{3k+K}F(\boldsymbol{x}^*)\right\|_2} + \left(\sqrt{\frac{2}{\alpha^3}}+\frac{3}{\alpha}\right)\left\|\nabla_{3k+K}F(\boldsymbol{x}^*)\right\|_2.$$

The proof can be found in Appendix D.2.

Remark 17 *The RSS parameter* L' *of* $F(\boldsymbol{x})$ *always ranges in* $[\alpha, L]$, *which is simply by definition.*

4.2.5 COMPUTATIONAL COMPLEXITY

We compare the computational complexity of HT-SVRG to that of projected gradient descent (PGD) studied in Jain et al. (2014), which is a batch counterpart to HT-SVRG. First, we remark that the analysis of PGD is based on the smoothness parameter L' of $F(\boldsymbol{x})$ at sparsity level $2k+K$. We write $c'=L'/\alpha$. To achieve a given accuracy $\epsilon>0$, PGD requires $\mathcal{O}\left(c'\log(1/\epsilon)\right)$ iterations. Hence the total computational complexity is $\mathcal{O}\left(nc'd\log(1/\epsilon)\right)$. For HT-SVRG, in view of Corollary 15, the convergence coefficient is a constant. Hence, HT-SVRG needs $\mathcal{O}\left(\log(1/\epsilon)\right)$ iterations where we note that the error term $\left\|\nabla_{3k+K}F(\boldsymbol{x}^*)\right\|_2$ can be made as small as ϵ with sufficient samples (to be clarified in the sequel). In each stage, HT-SVRG computes a full gradient $\widetilde{\boldsymbol{\mu}}$ followed by m times stochastic updates. Therefore, the total complexity of HT-SVRG is given by $\mathcal{O}\left((n+c)d\log(1/\epsilon)\right)$ by noting the fact $m=\mathcal{O}\left(c\right)$. In the scenario $c<n(c'-1)$, HT-SVRG significantly improves on PGD in terms of time cost.

4.3 Statistical Results

The last ingredient of our theorem is the term $\tau(\widehat{\boldsymbol{x}})$ which measures how close the iterates could be to a given sparse signal $\widehat{\boldsymbol{x}}$. With appropriate hyper-parameter settings, the quantity relies exclusively on $\left\|\nabla_{3k+K}F(\widehat{\boldsymbol{x}})\right\|_2$, as suggested by (4.6). Thereby, this section is dedicated to characterizing $\left\|\nabla_{3k+K}F(\widehat{\boldsymbol{x}})\right\|_2$. We will also give examples for which HT-SVRG is computationally more efficient than PGD. For the purpose of a concrete result, we study two problems: sparse linear regression and sparse logistic regression. These are two of the most popular statistical models in the literature and have found a variety of applications in machine learning and statistics (Raskutti et al., 2011). Notably, it is known that similar statistical results can be built for low-rank matrix regression, sparse precision matrix estimation, as suggested in Negahban et al. (2009); Agarwal et al. (2012).

4.3.1 SPARSE LINEAR REGRESSION

For sparse linear regression, the observation model is given by

$$\boldsymbol{y}=\boldsymbol{A}\boldsymbol{x}^*+\boldsymbol{\varepsilon},\quad \left\|\boldsymbol{x}^*\right\|_0\leq K,\ \left\|\boldsymbol{x}^*\right\|_2\leq\omega, \tag{4.7}$$

where $\boldsymbol{A}\in\mathbb{R}^{n\times d}$ is the design matrix, $\boldsymbol{y}\in\mathbb{R}^n$ is the response, $\boldsymbol{\varepsilon}\in\mathbb{R}^n$ is some noise, and $\boldsymbol{x}^*$ is the K-sparse true parameter we hope to estimate from the knowledge of $\boldsymbol{A}$ and $\boldsymbol{y}$. Note that when we have the additional constraint $n\ll d$, the model above is exactly that of compressed sensing (3.1).

In order to (approximately) estimate the parameter, a natural approach is to optimize the following non-convex program:

$$\min_{\boldsymbol{x}}\ F(\boldsymbol{x}):=\frac{1}{2n}\sum_{i=1}^{n}\left\|y_i-\boldsymbol{a}_i\cdot\boldsymbol{x}\right\|_2^2,\quad \text{s.t. } \left\|\boldsymbol{x}\right\|_0\leq K,\ \left\|\boldsymbol{x}\right\|_2\leq\omega. \tag{4.8}$$

For our analysis, we assume the following on the design matrix and the noise:

$(A3)$ $\boldsymbol{a}_1,\boldsymbol{a}_2,\ldots,\boldsymbol{a}_n$ are independent and identically distributed (i.i.d.) Gaussian random vectors $N(\boldsymbol{0},\boldsymbol{\Sigma})$. All the diagonal elements of $\boldsymbol{\Sigma}$ satisfy $\Sigma_{jj}\leq 1$. The noise $\boldsymbol{\varepsilon}$ is independent of $\boldsymbol{A}$ and its entries are i.i.d. Gaussian random variables $N(0,\sigma^2)$.

Proposition 18 *Consider the sparse linear regression model* (4.7) *and the program* (4.8). *Assume* $(A3)$. *Then for a sparsity level* r,

- *with probability at least* $1-\exp(-\mathrm{C}_0 n)$,

$$\alpha_r=\lambda_{\min}(\boldsymbol{\Sigma})-\mathrm{C}_1\frac{r\log d}{n},\quad L_r'=\lambda_{\max}(\boldsymbol{\Sigma})+\mathrm{C}_2\frac{r\log d}{n};$$

- *with probability at least* $1-\mathrm{C}_3 r/d$

$$L_r=\mathrm{C}_4 r\log d;$$

- *and with probability at least* $1-\mathrm{C}_5/d$

$$\left\|\nabla_r F(\boldsymbol{x}^*)\right\|_2\leq\mathrm{C}_6\sigma\sqrt{\frac{r\log d}{n}},\quad \left\|\nabla_r F(\boldsymbol{x}_{\mathrm{opt}})\right\|_2\leq L_r'\left\|\boldsymbol{x}_{\mathrm{opt}}-\boldsymbol{x}^*\right\|_2+\mathrm{C}_6\sigma\sqrt{\frac{r\log d}{n}}.$$

Above, $\lambda_{\min}(\boldsymbol{\Sigma})$ *and* $\lambda_{\max}(\boldsymbol{\Sigma})$ *are the minimum and maximum singular values of* $\boldsymbol{\Sigma}$ *respectively.*

We recall that α_r and L_r are involved in our assumptions $(A1)$ and $(A2)$, and L_r' is the RSS parameter of $F(\boldsymbol{x})$. The estimation for α_r, L_r' and $\left\|\nabla_r F(\boldsymbol{x}^*)\right\|_2$ follows from standard results in the literature (Raskutti et al., 2011), while that for L_r follows from Proposition E.1 in Bellec et al. (2016) by noting the fact that bounding L_r amounts to estimating $\max_i\left\|\mathcal{H}_r(\boldsymbol{a}_i)\right\|_2^2$. In order to estimate $\left\|\nabla_r F(\boldsymbol{x}_{\mathrm{opt}})\right\|_2$, notice that

$$\begin{aligned}\left\|\nabla_r F(\boldsymbol{x}_{\mathrm{opt}})\right\|_2 &\leq \left\|\nabla_r F(\boldsymbol{x}_{\mathrm{opt}})-\nabla_r F(\boldsymbol{x}^*)\right\|_2+\left\|\nabla_r F(\boldsymbol{x}^*)\right\|_2\\ &\leq \left\|\nabla F(\boldsymbol{x}_{\mathrm{opt}})-\nabla F(\boldsymbol{x}^*)\right\|_2+\left\|\nabla_r F(\boldsymbol{x}^*)\right\|_2\\ &\leq L_r'\left\|\boldsymbol{x}_{\mathrm{opt}}-\boldsymbol{x}^*\right\|_2+\left\|\nabla_r F(\boldsymbol{x}^*)\right\|_2,\end{aligned}$$

where we use the definition of RSS in the last inequality.

Now we let $r=3k+K=\mathrm{const}\cdot c^2K$ and get $\alpha=\lambda_{\min}(\boldsymbol{\Sigma})-\mathrm{C}_1\frac{c^2K\log d}{n}$, $L=\mathrm{C}_4c^2K\log d$. Suppose that $\lambda_{\min}(\boldsymbol{\Sigma})=2\mathrm{C}_4(K\log d)^2$ and $n=q\cdot\frac{\mathrm{C}_1}{\mathrm{C}_4}K\log d$ with $q\geq 1$. Then our assumptions $(A1)$ and $(A2)$ are met with high probability with

$$\alpha=\mathrm{C}_4(K\log d)^2,\ L=\mathrm{C}_4(K\log d)^3,\ \text{and } c=K\log d.$$

For Corollary 15, as far as

$$s \geq \mathrm{C}_7 \log\left(\frac{F(\widetilde{\boldsymbol{x}}^0) - F(\boldsymbol{x}_{\mathrm{opt}})}{\epsilon}\right),\ n = \mathrm{C}_7\left(\omega\sigma\right)^2 \epsilon^{-2} K \log d,$$

we have

$$\mathbb{E}\left[F(\widetilde{\boldsymbol{x}}^s) - F(\boldsymbol{x}_{\mathrm{opt}})\right] \leq \epsilon + \frac{\lambda_{\max}(\boldsymbol{\Sigma})}{\lambda_{\min}(\boldsymbol{\Sigma})}\left\|\boldsymbol{x}_{\mathrm{opt}} - \boldsymbol{x}^*\right\|_2 + \left(\frac{\lambda_{\max}(\boldsymbol{\Sigma})}{\lambda_{\min}(\boldsymbol{\Sigma})}\left\|\boldsymbol{x}_{\mathrm{opt}} - \boldsymbol{x}^*\right\|_2\right)^2$$

for some accuracy parameter $\epsilon > 0$. This suggests that it is possible for HT-SVRG to approximate a global optimum of (4.1) up to $\left\|\boldsymbol{x}_{\mathrm{opt}} - \boldsymbol{x}^*\right\|_2$, namely the statistical precision of the problem.

Returning to Corollary 16, to guarantee that

$$\mathbb{E}\left[\left\|\widetilde{\boldsymbol{x}}^s - \boldsymbol{x}^*\right\|_2\right] \leq \epsilon,$$

it suffices to pick

$$s \geq \mathrm{C}_8 \log(\omega\sqrt{c'}/\epsilon), \quad n = \mathrm{C}_8(\omega\sigma)^2\epsilon^{-4} K \log d.$$

Finally, we compare the computational cost to PGD. It is not hard to see that under the same situation $\lambda_{\min}(\boldsymbol{\Sigma}) = 2\mathrm{C}_4(K\log d)^2$ and $n = \frac{\mathrm{C}_1}{\mathrm{C}_4} K \log d$,

$$L' = \mathrm{C}_4(K\log d)^3,\ c' = K\log d,\ \text{provided that } \lambda_{\max}(\boldsymbol{\Sigma}) = \mathrm{C}_4(K\log d)^3 - \frac{\mathrm{C}_2\mathrm{C}_4}{\mathrm{C}_1}(K\log d)^2.$$

Thus $c < n(c'-1)$, i.e., HT-SVRG is more efficient than PGD. It is also possible to consider other regimes of the covariance matrix and the sample size, though we do not pursue it here.

4.3.2 SPARSE LOGISTIC REGRESSION

For sparse logistic regression, the observation model is given by

$$\Pr(y_i \mid \boldsymbol{a}_i;\, \boldsymbol{x}^*) = \frac{1}{1+\exp(-y_i\boldsymbol{a}_i\cdot\boldsymbol{x}^*)}, \quad \|\boldsymbol{x}^*\|_0 \leq K,\ \|\boldsymbol{x}\|_2 \leq \omega,\ \forall\, 1 \leq i \leq n, \tag{4.9}$$

where y_i is either 0 or 1. It then learns the parameter by minimizing the negative log-likelihood:

$$\min_{\boldsymbol{x}}\ F(\boldsymbol{x}) \coloneqq \frac{1}{n}\sum_{i=1}^{n}\log\left(1+\exp(-y_i\boldsymbol{a}_i\cdot\boldsymbol{x})\right), \quad \text{s.t. } \|\boldsymbol{x}\|_0 \leq K,\ \|\boldsymbol{x}\|_2 \leq \omega. \tag{4.10}$$

There is a large body of work showing that the statistical property is rather analogous to that of linear regression. See, for example, Negahban et al. (2009). In fact, the statistical results apply to generalized linear models as well.

4.4 A Concurrent Work

After we posted the first version Shen and Li (2016) on arXiv, Li et al. (2016) made their work public where a similar algorithm to HT-SVRG was presented. Their theoretical analysis applies to convex objective functions while we allow the function $F(\boldsymbol{x})$ to be non-convex. We also fully characterize the convergence behavior of the algorithm by showing the trade-off between the sparsity parameter k and the convergence coefficient β (Proposition 14).

5. Experiments

In this section, we present a comprehensive empirical study for the proposed HT-SVRG algorithm on two tasks: sparse recovery (compressed sensing) and image classification. The experiments on sparse recovery is dedicated to verifying the theoretical results we presented, and we visualize the classification models learned by HT-SVRG to demonstrate the practical efficacy.

5.1 Sparse Recovery

To understand the practical behavior of our algorithm as well as to justify the theoretical analysis, we perform experiments on synthetic data. The experimental settings are as follows:

- **Data Generation.** The data dimension d is fixed as 256 and we generate an $n \times d$ Gaussian random sensing matrix $\boldsymbol{A}$ whose entries are i.i.d. with zero mean and variance $1/n$. Then 1000 K-sparse signals $\boldsymbol{x}^*$ are independently generated, where the support of each signal is uniformly chosen. That is, we run our algorithm and the baselines for 1000 trials. The measurements $\boldsymbol{y}$ for each signal $\boldsymbol{x}^*$ is obtained by $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^*$ which is noise free. In this way, we are able to study the convergence rate by plotting the logarithm of the objective value since the optimal objective value is known to be zero.

- **Baselines.** We mainly compare with two closely related algorithms: IHT and PGD. Both of them compute the full gradient of the least-squares loss followed by hard thresholding. Yet, PGD is more general, in the sense that it allows the sparsity parameter k to be larger than the true sparsity K ($k = K$ for IHT) and also considers a flexible step size η ($\eta = 1$ for IHT). Hence, PGD can be viewed as a batch counterpart to our method HT-SVRG.

- **Evaluation Metric.** We say a signal $\boldsymbol{x}^*$ is successfully recovered by a solution $\boldsymbol{x}$ if

$$\frac{\|\boldsymbol{x} - \boldsymbol{x}^*\|_2}{\|\boldsymbol{x}^*\|_2} < 10^{-3}.$$

  In this way, we can compute the percentage of success over the 1000 trials for each algorithm.

- **Hyper-Parameters.** If not specified, we use $m = 3n$, $k = 9K$, and $S = 10000$ for HT-SVRG. We also use the heuristic step size $\eta = 2/\mathrm{svds}(\boldsymbol{A}\boldsymbol{A}^\top)$ for HT-SVRG and PGD, where $\mathrm{svds}(\boldsymbol{A}\boldsymbol{A}^\top)$ returns the largest singular value of the matrix $\boldsymbol{A}\boldsymbol{A}^\top$. Since for each stage, HT-SVRG computes the full gradient for $(2m/n+1)$ times, we run the IHT and PGD for $(2m/n+1)S$ iterations for fair comparison, i.e., all of the algorithms have the same number of full gradient evaluations.

5.1.1 PHASE TRANSITION

Our first simulation aims at offering a big picture on the recovery performance. To this end, we vary the number of measurements n from 1 to 256, roughly with a step size 8. We also study the performance with respect to the true sparsity parameter K, which ranges from 1 to 26, roughly with step size 2. The results are illustrated in Figure 1, where a brighter block means a higher percentage of success and the brightest ones indicate exact sparse recovery. It is apparent that PGD and HT-SVRG require fewer measurements for an accurate recovery than IHT, possibly due to the flexibility in choosing the sparsity parameter and the step size. We also observe that as a stochastic algorithm,

HT-SVRG performs comparably to PGD. This suggests that HT-SVRG is an appealing solution to large-scale sparse learning problems in that HT-SVRG is computationally more efficient.

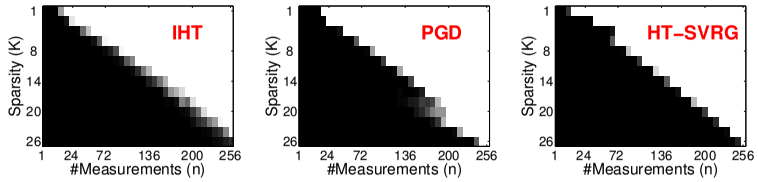


Figure 1: **Percentage of successful recovery under various sparsity and sample size.** The values range from 0 to 100, where a brighter color means a higher percentage of success (the brightest blocks correspond to the value of 100). PGD admits a higher percentage of recovery compared to IHT because it flexibly chooses the step size and sparsity parameter. As a stochastic variant, HT-SVRG performs comparably to the batch counterpart PGD.

In Figure 2, we exemplify some of the results obtained from HT-SVRG by plotting two kinds of curves: the success of percentage against the sample size n and that against the signal sparsity K. In this way, one can examine the detailed values and can determine the minimum sample size for a particular sparsity. For instance, the left panel tells that to ensure that 80% percents of the 16-sparse signals are recovered, we have to collect 175 measurements. We can also learn from the right panel that using 232 measurements, any signal whose sparsity is 22 or less can be reliably recovered.

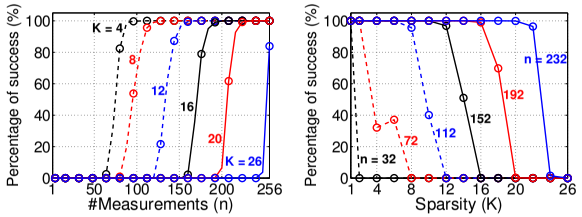


Figure 2: **Percentage of success of HT-SVRG against the number of measurements (left) and the sparsity (right).**

Based on the results in Figure 1 and Figure 2, we have an approximate estimation on the minimum requirement of the sample size which ensures accurate (or exact) recovery. Now we are to investigate how many measurements are needed to guarantee a success percentage of 95% and 99%. To this end, for each signal sparsity K, we look for the number of measurements n_0 from Figure 1 where 90 percents of success are achieved. Then we carefully enlarge n_0 with step size 1 and run the algorithms. The empirical results are recorded in Figure 3, where the circle markers represent the empirical results with different colors indicating different algorithms, e.g., red circle for empirical observation of HT-SVRG. Then we fit these empirical results by linear regression, which are

plotted as solid or dashed lines. For example, the green line is a fitted model for IHT. We find that n is almost linear with K. Especially, the curve of HT-SVRG is nearly on top of that of PGD, which again verifies HT-SVRG is an attractive alternative to the batch method.

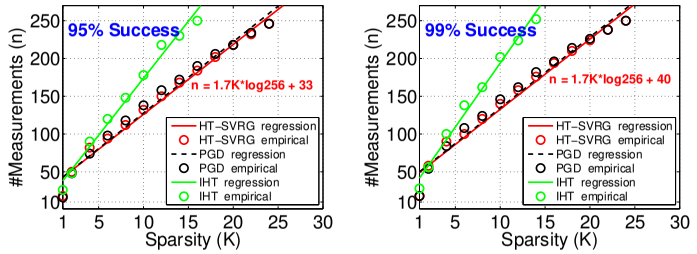


Figure 3: **Minimum number of measurements to achieve 95% and 99% percentage of success.** Red equation indicates the linear regression of HT-SVRG. The markers and curves for HT-SVRG are almost on top of PGD, which again justifies that HT-SVRG is an appealing stochastic alternative to the batch method PGD.

5.1.2 INFLUENCE OF HYPER-PARAMETERS

Next, we turn to investigate the influence of the hyper-parameters, i.e., the sparsity parameter k, update frequency m and step size η on the convergence behavior of HT-SVRG. We set the true sparsity $K = 4$ and collect 100 measurements for each groundtruth signal, i.e., $n = 100$. Note that the standard setting we employed is $k = 9K = 36$, $m = 3n = 300$ and $\eta = 2/\mathrm{svds}(\boldsymbol{A}\boldsymbol{A}^\top) \approx 0.3$. Each time we vary one of these parameters while fixing the other two, and the results are plotted in Figure 4. We point out that although the convergence result (Theorem 10) is deterministic, the vanishing optimization error (Proposition 18) is guaranteed under a probabilistic argument. Hence, it is possible that for a specific configuration of parameters, 97% of the signals are exactly recovered but HT-SVRG fails on the remaining, as we have observed in, e.g., Figure 2. Clearly, we are not supposed to average all the results to examine the convergence rate. For our purpose, we set a threshold 95%, that is, we average over the success trials if more than 95% percents of the signals are exactly recovered. Otherwise, we say that the set of parameters cannot ensure convergence and we average over these failure signals which will give an illustration of divergence.

The left panel of Figure 4 verifies the condition that k has to be larger than K, while the second panel shows the update frequency m can be reasonably small in the price of a slow convergence rate. Finally, the empirical study demonstrates that our heuristic choice $\eta = 0.3$ works well, and when $\eta > 3$, the objective value exceeds 10^{120} within 3 stages (which cannot be depicted in the figure). For very small step sizes, we plot the convergence curve by gradually enlarging the update frequency m in Figure 5. The empirical results agree with Theorem 10 that for any $0 < \eta < 1/(4L)$, HT-SVRG converges as soon as m is large enough.

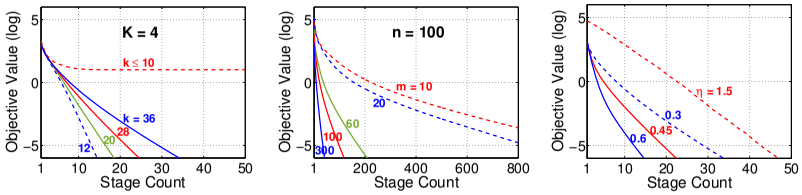


Figure 4: **Convergence of HT-SVRG with different parameters.** We have 100 measurements for the 256-dimensional signal where only 4 elements are non-zero. The standard setting is $k = 36$, $m = 300$ and $\eta = 0.3$. **Left:** If the sparsity parameter k is not large enough, HT-SVRG will not recover the signal. **Middle:** A small m leads to a frequent full gradient evaluation and hence slow convergence. **Right:** We observe divergence when $\eta \geq 3$.

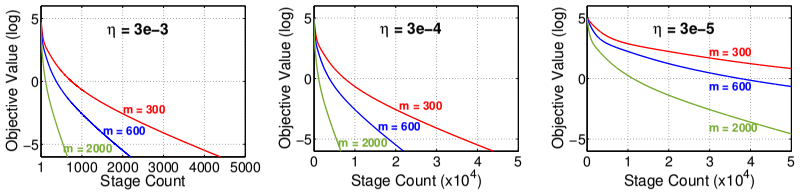


Figure 5: **Convergence behavior under small step size.** We observe that as long as we pick a sufficiently large value for m, HT-SVRG always converges. This is not surprising since our theorem guarantees for any $\eta < 1/(4L)$, HT-SVRG will converge if m is large enough. Also note that the geometric convergence rate is observed after certain iterations, e.g., for $\eta = 3 \times 10^{-5}$, the log(error) decreases linearly after 20 thousands iterations.

5.2 Classification

In addition to the application of sparse recovery, we illustrated that HT-SVRG can deal with binary classification by minimizing the sparse logistic regression problem (4.10). Here, we study the performance on a realistic image dataset MNIST[1], consisting of 60 thousands training samples and 10 thousands samples for testing. There is one digit on each image of size 28-by-28, hence totally 10 classes. Some of the images are shown in Figure 6.

The update frequency m is fixed as $m = 3n$. We compute the heuristic step size η as in the previous section, i.e., $\eta = 2/\mathrm{svds}(\boldsymbol{A}\boldsymbol{A}^\top) \approx 10^{-3}$. Since for the real-world dataset, the true sparsity is actually unknown, we tune the sparsity parameter k and study the performance of the algorithm.

First, we visualize five pair-wise models learned by HT-SVRG in Figure 7, where each row is associated with a binary classification task indicated by the two digits at the leading of the row, and the subsequent red-blue figures are used to illustrate the learned models under different sparsity parameter. For example, the third colorful figure depicted on the second row corresponds to recognizing a digit is "1" or "7" with the sparsity $k = 30$. In particular, for each pair, we label the small digit as positive and the large one as negative, and the blue and red pixels are the weights with positive and negative values respectively. Apparently, the models we learned are discriminative.

1. http://yann.lecun.com/exdb/mnist/

Figure 6: **Sample images in the MNIST database.**

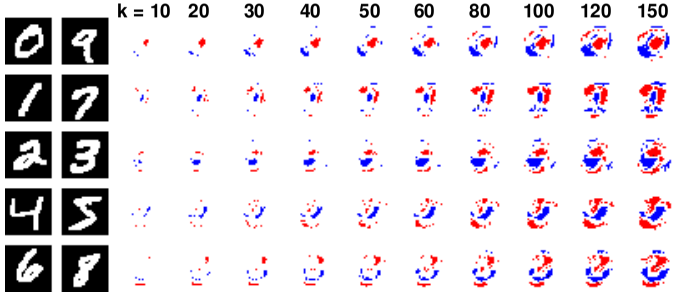


Figure 7: **Visualization of the models.** We visualize 5 models learned by HT-SVRG under different choices of sparsity shown on the top of each column. Note that the feature dimension is 784. From the top row to the bottom row, we illustrate the models of "0 vs 9", "1 vs 7", "2 vs 3", "4 vs 5" and "6 vs 8", where for each pair, we label the small digit as positive and the large one as negative. The red color represents negative weights while the blue pixels correspond with positive weights.

We also quantitatively show the convergence and prediction accuracy curves in Figure 8. Note that here, the y-axis is the objective value $F(\widetilde{\boldsymbol{x}}^s)$ rather than $\log(F(\widetilde{\boldsymbol{x}}^s) - F(\boldsymbol{x}_{\mathrm{opt}}))$, due to the fact that computing the exact optimum of (4.10) is NP-hard. Generally speaking, HT-SVRG converges quite fast and usually attains the minimum of objective value within 20 stages. It is not surprising to see that choosing a large quantity for the sparsity leads to a better (lower) objective value. However,

in practice a small assignment for the sparsity, e.g., $k = 70$ facilitates an efficient computation while still suffices to ensure fast convergence and accurate prediction.

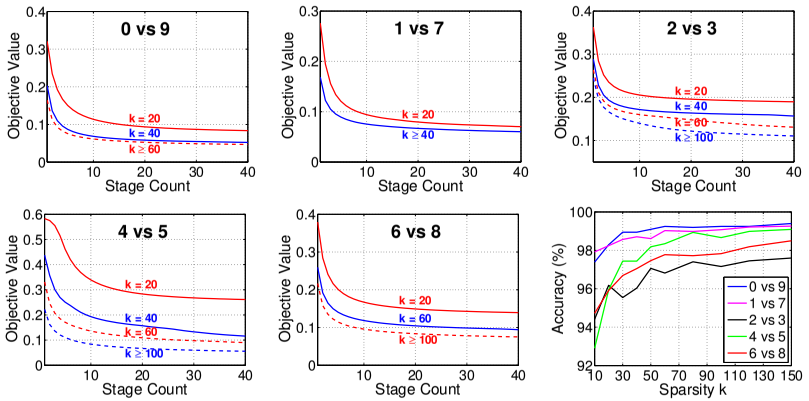


Figure 8: **Quantitative results on convergence and accuracy.** The first 5 figures demonstrate the convergence behavior of HT-SVRG for each binary classification task, where curves with different colors represent the objective value against number of stages under different sparsity k. Generally speaking, HT-SVRG converges within 20 stages which is a very fast rate. The last figure reflects the classification accuracy against the sparsity for all 5 classification tasks, where we find that for a moderate choice, e.g., $k = 70$, it already guarantees an accurate prediction (we recall the dimension is 784).

6. Conclusion and Open Problems

In this paper, we have provided a tight bound on the deviation resulting from the hard thresholding operator, which underlies a vast volume of algorithms developed for sparsity-constrained problems. Our derived bound is universal over all choices of parameters and we have proved that it cannot be improved without further information on the signals. We have discussed the implications of our result to the community of compressed sensing and machine learning, and have demonstrated that the theoretical results of a number of popular algorithms in the literature can be advanced. In addition, we have devised a novel algorithm which tackles the problem of sparse learning in large-scale setting. We have elaborated that our algorithm is guaranteed to produce global optimal solution for prevalent statistical models only when it is equipped with the tight bound, hence justifying that the conventional bound is not applicable in the challenging scenario.

There are several interesting open problems. The first question to ask is whether one can establish sharp RIP condition or sharp phase transition for hard thresholding based algorithms such as IHT and CoSaMP with the tight bound. Moreover, compared to the hard thresholded SGD method (Nguyen et al., 2014), HT-SVRG admits a vanishing optimization error. This poses a

question of whether we are able to provably show the necessity of variance reduction for such a sparsity-constrained problem.

Acknowledgments

We would like to thank Jing Wang for insightful discussion since the early stage of the work. We also thank Martin Slawski for helpful discussion on the statistical precision of the problem, and thank Jian Wang for bringing the paper Nguyen et al. (2014) into our attention. We appreciate Huan Xu's high level comments on the work. Finally, we thank the anonymous reviewers for a careful check on our proof and for the encouraging comments. The work was partially funded by NSF-Bigdata-1419210 and NSF-III-1360971.

Appendix A. Technical Lemmas

We present some useful lemmas that will be invoked by subsequent analysis. The following is a characterization of the co-coercivity of the objective function $F(\boldsymbol{x})$. A similar result was obtained in Nguyen et al. (2014) but we present a refined analysis which is essential for our purpose.

Lemma 19 *For a given support set Ω, assume that the continuous function $F(\boldsymbol{x})$ is $L_{|\Omega|}$-RSS and is α_K-RSC for some sparsity level K. Then, for all vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$ with $|\mathrm{supp}\,(\boldsymbol{x}-\boldsymbol{x}')\cup\Omega| \leq K$,*

$$\left\|\nabla_\Omega F(\boldsymbol{x}') - \nabla_\Omega F(\boldsymbol{x})\right\|_2^2 \leq 2L_{|\Omega|}\big(F(\boldsymbol{x}') - F(\boldsymbol{x}) - \left\langle \nabla F(\boldsymbol{x}), \boldsymbol{x}'-\boldsymbol{x}\right\rangle\big).$$

Proof We define an auxiliary function

$$G(\boldsymbol{w}) := F(\boldsymbol{w}) - \langle \nabla F(\boldsymbol{x}), \boldsymbol{w}\rangle .$$

For all vectors $\boldsymbol{w}$ and $\boldsymbol{w}'$, we have

$$\left\|\nabla G(\boldsymbol{w}) - \nabla G(\boldsymbol{w}')\right\|_2 = \left\|\nabla F(\boldsymbol{w}) - \nabla F(\boldsymbol{w}')\right\|_2 \leq L_{|\mathrm{supp}(\boldsymbol{w}-\boldsymbol{w}')|}\left\|\boldsymbol{w}-\boldsymbol{w}'\right\|_2,$$

which is equivalent to

$$G(\boldsymbol{w}) - G(\boldsymbol{w}') - \left\langle \nabla G(\boldsymbol{w}'), \boldsymbol{w}-\boldsymbol{w}'\right\rangle \leq \frac{L_r}{2}\left\|\boldsymbol{w}-\boldsymbol{w}'\right\|_2^2, \tag{A.1}$$

where $r := |\mathrm{supp}\,(\boldsymbol{w}-\boldsymbol{w}')|$. On the other hand, due to the RSC property of $F(\boldsymbol{x})$, we obtain

$$G(\boldsymbol{w}) - G(\boldsymbol{x}) = F(\boldsymbol{w}) - F(\boldsymbol{x}) - \langle \nabla F(\boldsymbol{x}), \boldsymbol{w}-\boldsymbol{x}\rangle \geq \frac{\alpha_{|\mathrm{supp}(\boldsymbol{w}-\boldsymbol{x})|}}{2}\left\|\boldsymbol{w}-\boldsymbol{x}\right\|_2^2 \geq 0,$$

provided that $|\mathrm{supp}\,(\boldsymbol{w}-\boldsymbol{x})| \leq K$. For the given support set Ω, we pick $\boldsymbol{w} = \boldsymbol{x}' - \frac{1}{L_{|\Omega|}}\nabla_\Omega G(\boldsymbol{x}')$. Clearly, for such a choice of $\boldsymbol{w}$, we have $\mathrm{supp}\,(\boldsymbol{w}-\boldsymbol{x}) = \mathrm{supp}\,(\boldsymbol{x}-\boldsymbol{x}')\cup\Omega$. Hence, by assuming that $|\mathrm{supp}\,(\boldsymbol{x}-\boldsymbol{x}')\cup\Omega|$ is not larger than K, we get

$$\begin{aligned} G(\boldsymbol{x}) &\leq G\left(\boldsymbol{x}' - \frac{1}{L_{|\Omega|}}\nabla_\Omega G(\boldsymbol{x}')\right) \\ &\leq G(\boldsymbol{x}') + \left\langle \nabla G(\boldsymbol{x}'), -\frac{1}{L_{|\Omega|}}\nabla_\Omega G(\boldsymbol{x}')\right\rangle + \frac{1}{2L_{|\Omega|}}\left\|\nabla_\Omega G(\boldsymbol{x}')\right\|_2^2 \\ &= G(\boldsymbol{x}') - \frac{1}{2L_{|\Omega|}}\left\|\nabla_\Omega G(\boldsymbol{x}')\right\|_2^2, \end{aligned}$$

where the second inequality follows from (A.1). Now expanding $\nabla_\Omega G(\boldsymbol{x}')$ and rearranging the terms gives the desired result. ■

Lemma 20 *Consider the HT-SVRG algorithm for a fixed stage s. Let $\widehat{\boldsymbol{x}}$ be the target sparse vector. Let Ω be a support set such that $\operatorname{supp}\left(\boldsymbol{x}^{t-1}\right) \cup \operatorname{supp}\left(\widetilde{\boldsymbol{x}}\right) \cup \operatorname{supp}\left(\widehat{\boldsymbol{x}}\right) \subseteq \Omega$. Put $r = |\Omega|$. Assume (A2). For all $1 \leq t \leq m$, denote $\boldsymbol{v}^t = \nabla f_{i_t}(\boldsymbol{x}^{t-1}) - \nabla f_{i_t}(\widetilde{\boldsymbol{x}}) + \widetilde{\boldsymbol{\mu}}$. Then we have the following:*

$$\begin{aligned}\mathbb{E}_{i_t|\boldsymbol{x}^{t-1}}\left[\left\|\mathcal{P}_\Omega\left(\boldsymbol{v}^t\right)\right\|_2^2\right] \leq\ & 4L_r\left[F(\boldsymbol{x}^{t-1}) - F(\widehat{\boldsymbol{x}})\right] + 4L_r\left[F(\widetilde{\boldsymbol{x}}) - F(\widehat{\boldsymbol{x}})\right] \\ & - 4L_r\left\langle \nabla F(\widehat{\boldsymbol{x}}), \boldsymbol{x}^{t-1} + \widetilde{\boldsymbol{x}} - 2\widehat{\boldsymbol{x}}\right\rangle + 4\left\|\mathcal{P}_\Omega\left(\nabla F(\widehat{\boldsymbol{x}})\right)\right\|_2^2.\end{aligned}$$

Proof We have

$$\begin{aligned}\left\|\mathcal{P}_\Omega\left(\boldsymbol{v}^t\right)\right\|_2^2 =\ & \left\|\mathcal{P}_\Omega\left(\nabla f_{i_t}(\boldsymbol{x}^{t-1}) - \nabla f_{i_t}(\widetilde{\boldsymbol{x}}) + \widetilde{\boldsymbol{\mu}}\right)\right\|_2^2 \\ \leq\ & 2\left\|\mathcal{P}_\Omega\left(\nabla f_{i_t}(\boldsymbol{x}^{t-1}) - \nabla f_{i_t}(\widehat{\boldsymbol{x}})\right)\right\|_2^2 + 2\left\|\mathcal{P}_\Omega\left(\nabla f_{i_t}(\widetilde{\boldsymbol{x}}) - \nabla f_{i_t}(\widehat{\boldsymbol{x}}) - \widetilde{\boldsymbol{\mu}}\right)\right\|_2^2 \\ =\ & 2\left\|\mathcal{P}_\Omega\left(\nabla f_{i_t}(\boldsymbol{x}^{t-1}) - \nabla f_{i_t}(\widehat{\boldsymbol{x}})\right)\right\|_2^2 + 2\left\|\mathcal{P}_\Omega\left(\nabla f_{i_t}(\widetilde{\boldsymbol{x}}) - \nabla f_{i_t}(\widehat{\boldsymbol{x}})\right)\right\|_2^2 \\ & + 2\left\|\mathcal{P}_\Omega\left(\widetilde{\boldsymbol{\mu}}\right)\right\|_2^2 - 4\left\langle \mathcal{P}_\Omega\left(\nabla f_{i_t}(\widetilde{\boldsymbol{x}}) - \nabla f_{i_t}(\widehat{\boldsymbol{x}})\right), \mathcal{P}_\Omega\left(\widetilde{\boldsymbol{\mu}}\right)\right\rangle \\ \overset{\xi_1}{=}\ & 2\left\|\mathcal{P}_\Omega\left(\nabla f_{i_t}(\boldsymbol{x}^{t-1}) - \nabla f_{i_t}(\widehat{\boldsymbol{x}})\right)\right\|_2^2 + 2\left\|\mathcal{P}_\Omega\left(\nabla f_{i_t}(\widetilde{\boldsymbol{x}}) - \nabla f_{i_t}(\widehat{\boldsymbol{x}})\right)\right\|_2^2 \\ & + 2\left\|\mathcal{P}_\Omega\left(\widetilde{\boldsymbol{\mu}}\right)\right\|_2^2 - 4\left\langle \nabla f_{i_t}(\widetilde{\boldsymbol{x}}) - \nabla f_{i_t}(\widehat{\boldsymbol{x}}), \mathcal{P}_\Omega\left(\widetilde{\boldsymbol{\mu}}\right)\right\rangle \\ \overset{\xi_2}{\leq}\ & 4L_r\left[f_{i_t}(\boldsymbol{x}^{t-1}) - f_{i_t}(\widehat{\boldsymbol{x}}) - \left\langle \nabla f_{i_t}(\widehat{\boldsymbol{x}}), \boldsymbol{x}^{t-1} - \widehat{\boldsymbol{x}}\right\rangle\right] \\ & + 4L_r\left[f_{i_t}(\widetilde{\boldsymbol{x}}) - f_{i_t}(\widehat{\boldsymbol{x}}) - \langle \nabla f_{i_t}(\widehat{\boldsymbol{x}}), \widetilde{\boldsymbol{x}} - \widehat{\boldsymbol{x}}\rangle\right] \\ & + 2\left\|\mathcal{P}_\Omega\left(\widetilde{\boldsymbol{\mu}}\right)\right\|_2^2 - 4\left\langle \nabla f_{i_t}(\widetilde{\boldsymbol{x}}) - \nabla f_{i_t}(\widehat{\boldsymbol{x}}), \mathcal{P}_\Omega\left(\widetilde{\boldsymbol{\mu}}\right)\right\rangle,\end{aligned}$$

where ξ_1 is by algebra, ξ_2 applies Lemma 19 and the fact that $|\Omega| = r$.

Taking the conditional expectation, we obtain the following:

$$\begin{aligned}& \mathbb{E}_{i_t|\boldsymbol{x}^{t-1}}\left[\left\|\mathcal{P}_\Omega\left(\boldsymbol{v}^t\right)\right\|_2^2\right] \\ \leq\ & 4L_r\left[F(\boldsymbol{x}^{t-1}) - F(\widehat{\boldsymbol{x}})\right] + 4L_r\left[F(\widetilde{\boldsymbol{x}}) - F(\widehat{\boldsymbol{x}})\right] \\ & - 4L_r\left\langle \nabla F(\widehat{\boldsymbol{x}}), \boldsymbol{x}^{t-1} + \widetilde{\boldsymbol{x}} - 2\widehat{\boldsymbol{x}}\right\rangle + 2\left\langle 2\mathcal{P}_\Omega\left(\nabla F(\widehat{\boldsymbol{x}})\right) - \mathcal{P}_\Omega\left(\widetilde{\boldsymbol{\mu}}\right), \mathcal{P}_\Omega\left(\widetilde{\boldsymbol{\mu}}\right)\right\rangle \\ =\ & 4L_r\left[F(\boldsymbol{x}^{t-1}) - F(\widehat{\boldsymbol{x}})\right] + 4L_r\left[F(\widetilde{\boldsymbol{x}}) - F(\widehat{\boldsymbol{x}})\right] \\ & - 4L_r\left\langle \nabla F(\widehat{\boldsymbol{x}}), \boldsymbol{x}^{t-1} + \widetilde{\boldsymbol{x}} - 2\widehat{\boldsymbol{x}}\right\rangle + \left\|2\mathcal{P}_\Omega\left(\nabla F(\widehat{\boldsymbol{x}})\right)\right\|_2^2 \\ & - \left\|2\mathcal{P}_\Omega\left(\nabla F(\widehat{\boldsymbol{x}})\right) - \mathcal{P}_\Omega\left(\widetilde{\boldsymbol{\mu}}\right)\right\|_2^2 - \left\|\mathcal{P}_\Omega\left(\widetilde{\boldsymbol{\mu}}\right)\right\|_2^2 \\ \leq\ & 4L_r\left[F(\boldsymbol{x}^{t-1}) - F(\widehat{\boldsymbol{x}})\right] + 4L_r\left[F(\widetilde{\boldsymbol{x}}) - F(\widehat{\boldsymbol{x}})\right] \\ & - 4L_r\left\langle \nabla F(\widehat{\boldsymbol{x}}), \boldsymbol{x}^{t-1} + \widetilde{\boldsymbol{x}} - 2\widehat{\boldsymbol{x}}\right\rangle + 4\left\|\mathcal{P}_\Omega\left(\nabla F(\widehat{\boldsymbol{x}})\right)\right\|_2^2.\end{aligned}$$

The proof is complete. ■

Corollary 21 *Assume the same conditions as in Lemma 20. If $\nabla F(\widehat{\boldsymbol{x}}) = 0$, we have*

$$\mathbb{E}_{i_t|\boldsymbol{x}^{t-1}}\left[\left\|\mathcal{P}_\Omega\left(\boldsymbol{v}^t\right)\right\|_2^2\right] \leq 4L_r\left[F(\boldsymbol{x}^{t-1}) + F(\widetilde{\boldsymbol{x}}) - 2F(\widehat{\boldsymbol{x}})\right].$$

Appendix B. Proofs for Section 2

B.1 Proof of Theorem 1

Proof The result is true for the trivial case that $\boldsymbol{b}$ is a zero vector. In the following, we assume that $\boldsymbol{b}$ is not a zero vector. Denote

$$\boldsymbol{w} := \mathcal{H}_k\left(\boldsymbol{b}\right).$$

Let Ω be the support set of $\boldsymbol{w}$ and let $\overline{\Omega}$ be its complement. We immediately have $\mathcal{P}_\Omega\left(\boldsymbol{b}\right) = \boldsymbol{w}$.

Let Ω' be the support set of $\boldsymbol{x}$. For the sake of simplicity, let us split the vector $\boldsymbol{b}$ as follows:

$$\begin{aligned}\boldsymbol{b}_1 = \mathcal{P}_{\Omega\setminus\Omega'}\left(\boldsymbol{b}\right), &\quad \boldsymbol{b}_2 = \mathcal{P}_{\Omega\cap\Omega'}\left(\boldsymbol{b}\right), \\ \boldsymbol{b}_3 = \mathcal{P}_{\overline{\Omega}\setminus\Omega'}\left(\boldsymbol{b}\right), &\quad \boldsymbol{b}_4 = \mathcal{P}_{\overline{\Omega}\cap\Omega'}\left(\boldsymbol{b}\right).\end{aligned}$$

Likewise, we denote

$$\begin{aligned}&\boldsymbol{w}_1 = \mathcal{P}_{\Omega\setminus\Omega'}\left(\boldsymbol{w}\right), \quad \boldsymbol{w}_2 = \mathcal{P}_{\Omega\cap\Omega'}\left(\boldsymbol{w}\right), \quad \boldsymbol{w}_3 = \mathcal{P}_{\overline{\Omega}\setminus\Omega'}(\boldsymbol{w}) = \boldsymbol{0}, \quad \boldsymbol{w}_4 = \mathcal{P}_{\overline{\Omega}\cap\Omega'}\left(\boldsymbol{w}\right) = \boldsymbol{0}, \\ &\boldsymbol{x}_1 = \mathcal{P}_{\Omega\setminus\Omega'}\left(\boldsymbol{x}\right) = \boldsymbol{0}, \quad \boldsymbol{x}_2 = \mathcal{P}_{\Omega\cap\Omega'}\left(\boldsymbol{x}\right), \quad \boldsymbol{x}_3 = \mathcal{P}_{\overline{\Omega}\setminus\Omega'}(\boldsymbol{x}) = \boldsymbol{0}, \quad \boldsymbol{x}_4 = \mathcal{P}_{\overline{\Omega}\cap\Omega'}\left(\boldsymbol{x}\right).\end{aligned}$$

Due to the hard thresholding, we have

$$\boldsymbol{w}_1 = \boldsymbol{b}_1, \quad \boldsymbol{w}_2 = \boldsymbol{b}_2.$$

In this way, by simple algebra we have

$$\begin{aligned}\|\boldsymbol{w} - \boldsymbol{x}\|_2^2 &= \|\boldsymbol{b}_1\|_2^2 + \|\boldsymbol{b}_2 - \boldsymbol{x}_2\|_2^2 + \|\boldsymbol{x}_4\|_2^2, \\ \|\boldsymbol{b} - \boldsymbol{x}\|_2^2 &= \|\boldsymbol{b}_1\|_2^2 + \|\boldsymbol{b}_2 - \boldsymbol{x}_2\|_2^2 + \|\boldsymbol{b}_3\|_2^2 + \|\boldsymbol{b}_4 - \boldsymbol{x}_4\|_2^2.\end{aligned}$$

Our goal is to estimate the maximum of $\|\boldsymbol{w} - \boldsymbol{x}\|_2^2 / \|\boldsymbol{b} - \boldsymbol{x}\|_2^2$. It is easy to show that when attaining the maximum value, $\|\boldsymbol{b}_3\|_2$ must be zero since otherwise one may decrease this term to make the objective larger. Hence, maximizing $\|\boldsymbol{w} - \boldsymbol{x}\|_2^2 / \|\boldsymbol{b} - \boldsymbol{x}\|_2^2$ amounts to estimating the upper bound of the following over all choices of $\boldsymbol{x}$ and $\boldsymbol{b}$:

$$\gamma := \frac{\|\boldsymbol{b}_1\|_2^2 + \|\boldsymbol{b}_2 - \boldsymbol{x}_2\|_2^2 + \|\boldsymbol{x}_4\|_2^2}{\|\boldsymbol{b}_1\|_2^2 + \|\boldsymbol{b}_2 - \boldsymbol{x}_2\|_2^2 + \|\boldsymbol{b}_4 - \boldsymbol{x}_4\|_2^2}. \tag{B.1}$$

Firstly, we consider the case of $\|\boldsymbol{b}_1\|_2 = 0$, which means $\Omega = \Omega'$ implying $\gamma = 1$. In the following, we consider $\|\boldsymbol{b}_1\|_2 \neq 0$. In particular, we consider $\gamma > 1$ since we are interested in the maximum value of γ.

Arranging (B.1) we obtain

$$(\gamma - 1)\|\boldsymbol{b}_2 - \boldsymbol{x}_2\|_2^2 + \gamma\|\boldsymbol{b}_4 - \boldsymbol{x}_4\|_2^2 - \|\boldsymbol{x}_4\|_2^2 + (\gamma - 1)\|\boldsymbol{b}_1\|_2^2 = 0. \tag{B.2}$$

Let us fix $\boldsymbol{b}$ and define the function

$$G(\boldsymbol{x}_2, \boldsymbol{x}_4) = (\gamma - 1)\|\boldsymbol{b}_2 - \boldsymbol{x}_2\|_2^2 + \gamma\|\boldsymbol{b}_4 - \boldsymbol{x}_4\|_2^2 - \|\boldsymbol{x}_4\|_2^2 + (\gamma - 1)\|\boldsymbol{b}_1\|_2^2.$$

Thus, (B.2) indicates that $G(\boldsymbol{x}_2, \boldsymbol{x}_4)$ can attain the objective value of zero. Note that $G(\boldsymbol{x}_2, \boldsymbol{x}_4)$ is a quadratic function and its gradient and Hessian matrix can be computed as follows:

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{x}_2} G(\boldsymbol{x}_2, \boldsymbol{x}_4) &= 2(\gamma - 1)(\boldsymbol{x}_2 - \boldsymbol{b}_2),\\ \frac{\partial}{\partial \boldsymbol{x}_4} G(\boldsymbol{x}_2, \boldsymbol{x}_4) &= 2\gamma(\boldsymbol{x}_4 - \boldsymbol{b}_4) - 2\boldsymbol{x}_4,\\ \nabla^2 G(\boldsymbol{x}_2, \boldsymbol{x}_4) &= 2(\gamma - 1)\boldsymbol{I},\end{aligned}$$

where $\boldsymbol{I}$ is the identity matrix. Since the Hessian matrix is positive definite, $G(\boldsymbol{x}_2, \boldsymbol{x}_4)$ attains the global minimum at the stationary point, which is given by

$$\boldsymbol{x}_2^* = \boldsymbol{b}_2, \quad \boldsymbol{x}_4^* = \frac{\gamma}{\gamma - 1}\boldsymbol{b}_4,$$

resulting in the minimum objective value

$$G(\boldsymbol{x}_2^*, \boldsymbol{x}_4^*) = \frac{\gamma}{1-\gamma}\left\|\boldsymbol{b}_4\right\|_2^2 + (\gamma - 1)\left\|\boldsymbol{b}_1\right\|_2^2.$$

In order to guarantee the feasible set of (B.2) is non-empty, we require that

$$G(\boldsymbol{x}_2^*, \boldsymbol{x}_4^*) \le 0,$$

implying

$$\left\|\boldsymbol{b}_1\right\|_2^2\gamma^2 - (2\left\|\boldsymbol{b}_1\right\|_2^2 + \left\|\boldsymbol{b}_4\right\|_2^2)\gamma + \left\|\boldsymbol{b}_1\right\|_2^2 \le 0.$$

Solving the above inequality with respect to γ, we obtain

$$\gamma \le 1 + \frac{\left\|\boldsymbol{b}_4\right\|_2^2 + \sqrt{\left(4\left\|\boldsymbol{b}_1\right\|_2^2 + \left\|\boldsymbol{b}_4\right\|_2^2\right)\left\|\boldsymbol{b}_4\right\|_2^2}}{2\left\|\boldsymbol{b}_1\right\|_2^2}. \tag{B.3}$$

To derive an upper bound that is uniform over the choice of $\boldsymbol{b}$, we recall that $\boldsymbol{b}_1$ contains the largest absolute elements of $\boldsymbol{b}$ while $\boldsymbol{b}_4$ has smaller values. In particular, the averaged value of $\boldsymbol{b}_4$ is no greater than that of $\boldsymbol{b}_1$ in magnitude, i.e.,

$$\frac{\left\|\boldsymbol{b}_4\right\|_2^2}{\left\|\boldsymbol{b}_4\right\|_0} \le \frac{\left\|\boldsymbol{b}_1\right\|_2^2}{\left\|\boldsymbol{b}_1\right\|_0}.$$

Note that $\left\|\boldsymbol{b}_1\right\|_0 = k - \left\|\boldsymbol{b}_2\right\|_0 = k - (K - \left\|\boldsymbol{b}_4\right\|_0)$. Hence, combining with the fact that $0 \le \left\|\boldsymbol{b}_4\right\|_0 \le \min\{K, d-k\}$ and optimizing over $\left\|\boldsymbol{b}_4\right\|_0$ gives

$$\left\|\boldsymbol{b}_4\right\|_2^2 \le \frac{\min\{K, d-k\}}{k - K + \min\{K, d-k\}}\left\|\boldsymbol{b}_1\right\|_2^2.$$

Plugging back to (B.3), we finally obtain

$$\gamma \le 1 + \frac{\rho + \sqrt{(4+\rho)\,\rho}}{2}, \quad \rho = \frac{\min\{K, d-k\}}{k - K + \min\{K, d-k\}}.$$

The proof is complete. ■

Appendix C. Proofs for Section 3

C.1 Proof of Theorem 6

We follow the proof pipeline of Blumensath and Davies (2009) and only remark the difference of our proof and theirs, i.e., where Theorem 1 applies. In case of possible confusion due to notation, we follow the symbols in Blumensath and Davies. One may refer to that article for a complete proof.

The first difference occurs in Eq. (22) of Blumensath and Davies (2009), where they reached

$$\text{(Old)} \quad \left\|\boldsymbol{x}^s - \boldsymbol{x}^{[n+1]}\right\|_2 \le 2\left\|\boldsymbol{x}^s_{B^{n+1}} - \boldsymbol{a}^{[n+1]}_{B^{n+1}}\right\|_2,$$

while Theorem 1 gives

$$\text{(New)} \quad \left\|\boldsymbol{x}^s - \boldsymbol{x}^{[n+1]}\right\|_2 \le \sqrt{\nu}\left\|\boldsymbol{x}^s_{B^{n+1}} - \boldsymbol{a}^{[n+1]}_{B^{n+1}}\right\|_2.$$

Combining this new inequality and Eq. (23) therein, we obtain

$$\left\|\boldsymbol{x}^s - \boldsymbol{x}^{[n+1]}\right\|_2 \le \sqrt{\nu}\left\|(\boldsymbol{I} - \boldsymbol{\Phi}^\top_{B^{n+1}}\boldsymbol{\Phi}_{B^{n+1}})\boldsymbol{r}^{[n]}_{B^{n+1}}\right\|_2 + \sqrt{\nu}\left\|(\boldsymbol{\Phi}^\top_{B^{n+1}}\boldsymbol{\Phi}_{B^{n+1}\setminus B^{n+1}})\boldsymbol{r}^{[n]}_{B^{n+1}\setminus B^{n+1}}\right\|_2.$$

By noting the fact that $\left|B^n \cup B^{n+1}\right| \le 2s + s^*$ where s^* denotes the sparsity of the global optimum and following their reasoning of Eq. (24) and (25), we have a new bound for Eq. (26):

$$\text{(New)} \quad \left\|\boldsymbol{r}^{[n+1]}\right\|_2 \le \sqrt{2\nu}\delta_{2s+s^*}\left\|\boldsymbol{r}^{[n]}\right\|_2 + \sqrt{(1+\delta_{s+s^*})\nu}\left\|\boldsymbol{e}\right\|_2.$$

Now our result follows by setting the coefficient of $\left\|\boldsymbol{r}^{[n]}\right\|_2$ to 0.5. Note that specifying $\nu = 4$ gives the result of Blumensath and Davies (2009).

C.2 Proof of Theorem 7

We follow the proof technique of Theorem 6.27 in Foucart and Rauhut (2013) which gives the best known RIP condition for the CoSaMP algorithm to date. Since most of the reasoning is similar, we only point out the difference of our proof and theirs, i.e., where Theorem 1 applies. In case of confusion by notation, we follow the symbols used in Foucart and Rauhut (2013). The reader may refer to that book for a complete proof.

The first difference is in Eq. (6.49) of Foucart and Rauhut (2013). Note that to derive this inequality, Foucart and Rauhut invoked the conventional bound (1.1), which gives

$$\text{(Old)} \quad \left\|\boldsymbol{x}_S - \boldsymbol{x}^{n+1}\right\|_2^2 \le \left\|(\boldsymbol{x}_S - \boldsymbol{u}^{n+1})_{\overline{U^{n+1}}}\right\|_2^2 + 4\left\|(\boldsymbol{x}_S - \boldsymbol{u}^{n+1})_{U^{n+1}}\right\|_2^2,$$

while utilizing Theorem 1 gives

$$\text{(New)} \quad \left\|\boldsymbol{x}_S - \boldsymbol{x}^{n+1}\right\|_2^2 \le \left\|(\boldsymbol{x}_S - \boldsymbol{u}^{n+1})_{\overline{U^{n+1}}}\right\|_2^2 + \nu\left\|(\boldsymbol{x}_S - \boldsymbol{u}^{n+1})_{U^{n+1}}\right\|_2^2.$$

Combining this new inequality with Eq. (6.50) and Eq. (6.51) therein, we obtain

$$\begin{aligned}\left\|\boldsymbol{x}_S - \boldsymbol{x}^{n+1}\right\|_2 \leq \sqrt{2}\delta_{3s+s^*}\sqrt{\frac{1+(\nu-1)\delta_{3s+s^*}^2}{1-\delta_{3s+s^*}^2}}\left\|\boldsymbol{x}^n - \boldsymbol{x}_S\right\|_2 \\ + \sqrt{2}\delta_{3s+s^*}\sqrt{\frac{1+(\nu-1)\delta_{3s+s^*}^2}{1-\delta_{3s+s^*}^2}}\left\|(\boldsymbol{A}^*\boldsymbol{e}')_{(S\cup S^n)\Delta T^{n+1}}\right\|_2 \\ + \frac{2}{1-\delta_{3s+s^*}}\left\|(\boldsymbol{A}^*\boldsymbol{e}')_{U^{n+1}}\right\|_2,\end{aligned}$$

where s^* denotes the sparsity of the optimum. Our new bound follows by setting the coefficient of $\|\boldsymbol{x}^n - \boldsymbol{x}_S\|_2$ to 0.5 and solving the resultant equation. Note that setting $\nu = 4$ gives the old bound of Foucart and Rauhut.

Appendix D. Proofs for Section 4

D.1 Proof of Theorem 10

Proof Fix a stage s. Let us denote

$$\boldsymbol{v}^t = \nabla f_{i_t}(\boldsymbol{x}^{t-1}) - \nabla f_{i_t}(\widetilde{\boldsymbol{x}}) + \widetilde{\boldsymbol{\mu}},$$

so that

$$\boldsymbol{b}^t = \boldsymbol{x}^{t-1} - \eta\boldsymbol{v}^t.$$

By specifying $\Omega = \operatorname{supp}\left(\boldsymbol{x}^{t-1}\right) \cup \operatorname{supp}\left(\boldsymbol{x}^t\right) \cup \operatorname{supp}\left(\widetilde{\boldsymbol{x}}\right) \cup \operatorname{supp}\left(\widehat{\boldsymbol{x}}\right)$, it follows that

$$\boldsymbol{r}^t = \mathcal{H}_k\left(\boldsymbol{b}^t\right) = \mathcal{H}_k\left(\mathcal{P}_\Omega\left(\boldsymbol{b}^t\right)\right).$$

Thus, the Euclidean distance of $\boldsymbol{x}^t$ and $\widehat{\boldsymbol{x}}$ can be bounded as follows:

$$\left\|\boldsymbol{x}^t - \widehat{\boldsymbol{x}}\right\|_2^2 \leq \left\|\boldsymbol{r}^t - \widehat{\boldsymbol{x}}\right\|_2^2 = \left\|\mathcal{H}_k\left(\mathcal{P}_\Omega\left(\boldsymbol{b}^t\right)\right) - \widehat{\boldsymbol{x}}\right\|_2^2 \leq \nu\left\|\mathcal{P}_\Omega\left(\boldsymbol{b}^t\right) - \widehat{\boldsymbol{x}}\right\|_2^2, \tag{D.1}$$

where the first inequality holds because $\boldsymbol{x}^t = \Pi_\omega(\boldsymbol{r}^t)$ and $\|\widehat{\boldsymbol{x}}\|_2 \leq \omega$. We also have

$$\begin{aligned}\left\|\mathcal{P}_\Omega\left(\boldsymbol{b}^t\right) - \widehat{\boldsymbol{x}}\right\|_2^2 &= \left\|\boldsymbol{x}^{t-1} - \widehat{\boldsymbol{x}} - \eta\mathcal{P}_\Omega\left(\boldsymbol{v}^t\right)\right\|_2^2 \\ &= \left\|\boldsymbol{x}^{t-1} - \widehat{\boldsymbol{x}}\right\|_2^2 + \eta^2\left\|\mathcal{P}_\Omega\left(\boldsymbol{v}^t\right)\right\|_2^2 - 2\eta\left\langle\boldsymbol{x}^{t-1} - \widehat{\boldsymbol{x}}, \boldsymbol{v}^t\right\rangle,\end{aligned}$$

where the second equality uses the fact that $\left\langle\boldsymbol{x}^{t-1} - \widehat{\boldsymbol{x}}, \mathcal{P}_\Omega\left(\boldsymbol{v}^t\right)\right\rangle = \left\langle\boldsymbol{x}^{t-1} - \widehat{\boldsymbol{x}}, \boldsymbol{v}^t\right\rangle$. The first term will be preserved for mathematical induction. The third term is easy to manipulate thanks to the unbiasedness of $\boldsymbol{v}^t$. For the second term, we use Lemma 20 to bound it. Put them together,

conditioning on $\boldsymbol{x}^{t-1}$ and taking the expectation over i_t for (D.1), we have

$$\begin{aligned}&\mathbb{E}_{i_t|\boldsymbol{x}^{t-1}}\left[\left\|\boldsymbol{x}^t - \widehat{\boldsymbol{x}}\right\|_2^2\right] \\ \overset{\xi_1}{\leq}\ &\nu\left\|\boldsymbol{x}^{t-1} - \widehat{\boldsymbol{x}}\right\|_2^2 + 4\nu\eta^2 L\left[F(\boldsymbol{x}^{t-1}) - F(\widehat{\boldsymbol{x}}) + F(\widetilde{\boldsymbol{x}}) - F(\widehat{\boldsymbol{x}})\right] - 2\nu\eta\left\langle\boldsymbol{x}^{t-1} - \widehat{\boldsymbol{x}}, \nabla F(\boldsymbol{x}^{t-1})\right\rangle \\ &- 4\nu\eta^2 L\left\langle\nabla F(\widehat{\boldsymbol{x}}), \boldsymbol{x}^{t-1} + \widetilde{\boldsymbol{x}} - 2\widehat{\boldsymbol{x}}\right\rangle + 4\nu\eta^2\left\|\mathcal{P}_\Omega\left(\nabla F(\widehat{\boldsymbol{x}})\right)\right\|_2^2 \\ \overset{\xi_2}{\leq}\ &\nu(1-\eta\alpha)\left\|\boldsymbol{x}^{t-1} - \widehat{\boldsymbol{x}}\right\|_2^2 - 2\nu\eta(1-2\eta L)\left[F(\boldsymbol{x}^{t-1}) - F(\widehat{\boldsymbol{x}})\right] + 4\nu\eta^2 L\left[F(\widetilde{\boldsymbol{x}}) - F(\widehat{\boldsymbol{x}})\right] \\ &+ 4\nu\eta^2 L\left\|\mathcal{P}_\Omega\left(\nabla F(\widehat{\boldsymbol{x}})\right)\right\|_2 \cdot \left\|\boldsymbol{x}^{t-1} + \widetilde{\boldsymbol{x}} - 2\widehat{\boldsymbol{x}}\right\|_2 + 4\nu\eta^2\left\|\mathcal{P}_\Omega\left(\nabla F(\widehat{\boldsymbol{x}})\right)\right\|_2^2 \\ \leq\ &\nu(1-\eta\alpha)\left\|\boldsymbol{x}^{t-1} - \widehat{\boldsymbol{x}}\right\|_2^2 - 2\nu\eta(1-2\eta L)\left[F(\boldsymbol{x}^{t-1}) - F(\widehat{\boldsymbol{x}})\right] \\ &+ 4\nu\eta^2 L\left[F(\widetilde{\boldsymbol{x}}) - F(\widehat{\boldsymbol{x}})\right] + 4\nu\eta^2 Q'(4L\omega + Q')\end{aligned}$$

where ξ_1 applies Lemma 20, ξ_2 applies Assumption $(A1)$ and we write $Q' := \|\nabla_{3k+K}F(\widehat{\boldsymbol{x}})\|_2$ for brevity.

Now summing over the inequalities over $t = 1, 2, \cdots, m$, conditioning on $\widetilde{\boldsymbol{x}}$ and taking the expectation with respect to $\mathcal{I}^s = \{i_1, i_2, \cdots, i_m\}$, we have

$$\begin{aligned}&\mathbb{E}_{\mathcal{I}^s|\widetilde{\boldsymbol{x}}}\left[\left\|\boldsymbol{x}^m - \widehat{\boldsymbol{x}}\right\|_2^2\right] \\ \leq\ &\left[\nu(1-\eta\alpha) - 1\right]\mathbb{E}_{\mathcal{I}^s|\widetilde{\boldsymbol{x}}}\sum_{t=1}^m\left\|\boldsymbol{x}^{t-1} - \widehat{\boldsymbol{x}}\right\|_2^2 + \left\|\boldsymbol{x}^0 - \widehat{\boldsymbol{x}}\right\|_2^2 + 4\nu\eta^2 Q'(4L\omega + Q')m \\ &- 2\nu\eta(1-2\eta L)\mathbb{E}_{\mathcal{I}^s|\widetilde{\boldsymbol{x}}}\sum_{t=1}^m\left[F(\boldsymbol{x}^{t-1}) - F(\widehat{\boldsymbol{x}})\right] + 4\nu\eta^2 Lm\left[F(\widetilde{\boldsymbol{x}}) - F(\widehat{\boldsymbol{x}})\right] \\ =\ &\left[\nu(1-\eta\alpha) - 1\right]m\mathbb{E}_{\mathcal{I}^s,j^s|\widetilde{\boldsymbol{x}}}\left\|\widetilde{\boldsymbol{x}}^s - \widehat{\boldsymbol{x}}\right\|_2^2 + \left\|\widetilde{\boldsymbol{x}} - \widehat{\boldsymbol{x}}\right\|_2^2 + 4\nu\eta^2 Q'(4L\omega + Q')m \\ &- 2\nu\eta(1-2\eta L)m\mathbb{E}_{\mathcal{I}^s,j^s|\widetilde{\boldsymbol{x}}}\left[F(\widetilde{\boldsymbol{x}}^s) - F(\widehat{\boldsymbol{x}})\right] + 4\nu\eta^2 Lm\left[F(\widetilde{\boldsymbol{x}}) - F(\widehat{\boldsymbol{x}})\right] \\ \leq\ &\left[\nu(1-\eta\alpha) - 1\right]m\mathbb{E}_{\mathcal{I}^s,j^s|\widetilde{\boldsymbol{x}}}\left\|\widetilde{\boldsymbol{x}}^s - \widehat{\boldsymbol{x}}\right\|_2^2 + \left(\frac{2}{\alpha} + 4\nu\eta^2 Lm\right)\left[F(\widetilde{\boldsymbol{x}}) - F(\widehat{\boldsymbol{x}})\right] \\ &- 2\nu\eta(1-2\eta L)m\mathbb{E}_{\mathcal{I}^s,j^s|\widetilde{\boldsymbol{x}}}\left[F(\widetilde{\boldsymbol{x}}^s) - F(\widehat{\boldsymbol{x}})\right] + 4\nu\eta^2 Q'(4L\omega + Q')m + 2Q'\omega/\alpha,\end{aligned} \tag{D.2}$$

where we recall that j^s is the randomly chosen index used to determine $\widetilde{\boldsymbol{x}}^s$ (see Algorithm 1). The last inequality holds due to the RSC condition and $\left\|\boldsymbol{x}^t\right\|_2 \leq \omega$. For brevity, we write

$$Q := 4\nu\eta^2 Q'(4L\omega + Q')m + 2Q'\omega/\alpha, \quad Q' = \|\nabla_{3k+K}F(\widehat{\boldsymbol{x}})\|_2.$$

Based on (D.2), we discuss two cases to examine the convergence of the algorithm.

Case 1. $\nu(1-\eta\alpha) \leq 1$. This immediately results in

$$\begin{aligned}&\mathbb{E}_{\mathcal{I}^s|\widetilde{\boldsymbol{x}}}\left[\left\|\boldsymbol{x}^m - \widehat{\boldsymbol{x}}\right\|_2^2\right] \\ \leq\ &\left(\frac{2}{\alpha} + 4\nu\eta^2 Lm\right)\left[F(\widetilde{\boldsymbol{x}}) - F(\widehat{\boldsymbol{x}})\right] - 2\nu\eta(1-2\eta L)m\,\mathbb{E}_{\mathcal{I}^s,j^s|\widetilde{\boldsymbol{x}}}\left[F(\widetilde{\boldsymbol{x}}^s) - F(\widehat{\boldsymbol{x}})\right] + Q,\end{aligned}$$

which implies

$$\nu\eta(1-2\eta L)m\mathbb{E}_{\mathcal{I}^s,j^s|\widetilde{\boldsymbol{x}}}\left[F(\widetilde{\boldsymbol{x}}^s)-F(\widehat{\boldsymbol{x}})\right]\leq\left(\frac{1}{\alpha}+2\nu\eta^2Lm\right)\left[F(\widetilde{\boldsymbol{x}})-F(\widehat{\boldsymbol{x}})\right]+\frac{Q}{2}.$$

Pick η such that

$$1-2\eta L>0, \tag{D.3}$$

we obtain

$$\mathbb{E}_{\mathcal{I}^s,j^s|\widetilde{\boldsymbol{x}}}\left[F(\widetilde{\boldsymbol{x}}^s)-F(\widehat{\boldsymbol{x}})\right]\leq\left(\frac{1}{\nu\eta\alpha(1-2\eta L)m}+\frac{2\eta L}{1-2\eta L}\right)\left[F(\widetilde{\boldsymbol{x}})-F(\widehat{\boldsymbol{x}})\right]+\frac{Q}{2\nu\eta\alpha(1-2\eta L)m}.$$

To guarantee the convergence, we must impose

$$\frac{2\eta L}{1-2\eta L}<1. \tag{D.4}$$

Putting (D.3), (D.4) and $\nu(1-\eta\alpha)\leq 1$ together gives

$$\eta<\frac{1}{4L},\quad \nu\leq\frac{1}{1-\eta\alpha}. \tag{D.5}$$

The convergence coefficient here is

$$\beta=\frac{1}{\nu\eta\alpha(1-2\eta L)m}+\frac{2\eta L}{1-2\eta L}. \tag{D.6}$$

Thus, we have

$$\mathbb{E}\left[F(\widetilde{\boldsymbol{x}}^s)-F(\widehat{\boldsymbol{x}})\right]\leq\beta^s\left[F(\widetilde{\boldsymbol{x}}^0)-F(\widehat{\boldsymbol{x}})\right]+\frac{Q}{2\nu\eta\alpha(1-2\eta L)(1-\beta)m},$$

where the expectation is taken over $\{\mathcal{I}^1,j^1,\mathcal{I}^2,j^2,\cdots,\mathcal{I}^s,j^s\}$.

Case 2. $\nu(1-\eta\alpha)>1$. In this case, (D.2) implies

$$\begin{aligned}\mathbb{E}_{\mathcal{I}^s|\widetilde{\boldsymbol{x}}}\left[\|\boldsymbol{x}^m-\widehat{\boldsymbol{x}}\|_2^2\right]\leq&\left(\frac{2}{\alpha}+4\nu\eta^2Lm\right)\left[F(\widetilde{\boldsymbol{x}})-F(\widehat{\boldsymbol{x}})\right]+Q\\&+\left(\frac{2}{\alpha}\left[\nu(1-\eta\alpha)-1\right]m-2\nu\eta(1-2\eta L)m\right)\mathbb{E}_{\mathcal{I}^s,j^s|\widetilde{\boldsymbol{x}}}\left[F(\widetilde{\boldsymbol{x}}^s)-F(\widehat{\boldsymbol{x}})\right].\end{aligned}$$

Rearranging the terms gives

$$\left(2\nu\eta\alpha-2\nu\eta^2\alpha L-\nu+1\right)m\,\mathbb{E}_{\mathcal{I}^s,j^s|\widetilde{\boldsymbol{x}}}\left[F(\widetilde{\boldsymbol{x}}^s)-F(\widehat{\boldsymbol{x}})\right]\leq\left(1+2\nu\eta^2\alpha Lm\right)\left[F(\widetilde{\boldsymbol{x}})-F(\widehat{\boldsymbol{x}})\right]+\frac{\alpha Q}{2}.$$

To ensure the convergence, the minimum requirements are

$$\begin{aligned}&2\nu\eta\alpha-2\nu\eta^2\alpha L-\nu+1>0,\\&2\nu\eta\alpha-2\nu\eta^2\alpha L-\nu+1>2\nu\eta^2\alpha L.\end{aligned}$$

That is,

$$4\nu\alpha L\eta^2-2\nu\alpha\eta+\nu-1<0.$$

We need to guarantee the feasible set of the above inequality is non-empty for the positive variable η. Thus, we require

$$4\nu^2\alpha^2-4\times 4\nu\alpha L(\nu-1)>0,$$

which is equivalent to

$$\nu<\frac{4L}{4L-\alpha}.$$

Combining it with $\nu(1-\eta\alpha)>1$ gives

$$\frac{1}{1-\eta\alpha}<\nu<\frac{4L}{4L-\alpha}.$$

To ensure the above feasible set is non-empty, we impose

$$\frac{1}{1-\eta\alpha}<\frac{4L}{4L-\alpha},$$

so that

$$0<\eta<\frac{1}{4L},\quad \frac{1}{1-\eta\alpha}<\nu<\frac{4L}{4L-\alpha}. \tag{D.7}$$

The convergence coefficient for this case is

$$\beta=\frac{1}{\left(2\nu\eta\alpha-2\nu\eta^2\alpha L-\nu+1\right)m}+\frac{2\nu\eta^2\alpha L}{2\nu\eta\alpha-2\nu\eta^2\alpha L-\nu+1}. \tag{D.8}$$

Thus,

$$\mathbb{E}\left[F(\widetilde{\boldsymbol{x}}^s)-F(\widehat{\boldsymbol{x}})\right]\leq\beta^s\left[F(\widetilde{\boldsymbol{x}}^0)-F(\widehat{\boldsymbol{x}})\right]+\frac{\alpha Q}{2(2\nu\eta\alpha-2\nu\eta^2\alpha L-\nu+1)(1-\beta)m}.$$

By combining (D.5) and (D.7), the minimum requirement for η and ν is

$$0<\eta<\frac{1}{4L},\quad \nu<\frac{4L}{4L-\alpha}.$$

The proof is complete. ■

D.2 Proof of Corollary 16

Proof By noting the concavity of the square root function, we have

$$\begin{aligned}\mathbb{E}\left[\sqrt{\max\{F(\widetilde{\boldsymbol{x}}^s)-F(\widehat{\boldsymbol{x}}),0\}}\right]&\leq\sqrt{\mathbb{E}\left[\max\{F(\widetilde{\boldsymbol{x}}^s)-F(\widehat{\boldsymbol{x}}),0\}\right]}\\&\leq\sqrt{(2/3)^s\max\{F(\widetilde{\boldsymbol{x}}^0)-F(\widehat{\boldsymbol{x}}),0\}+\tau(\widehat{\boldsymbol{x}})}.\end{aligned}$$

Suppose that $F(\boldsymbol{x})$ satisfies RSS with parameter $L' \in [\alpha, L]$. It follows that

$$F(\widetilde{\boldsymbol{x}}^0) - F(\widehat{\boldsymbol{x}}) \leq \left\langle \nabla F(\widehat{\boldsymbol{x}}), \widetilde{\boldsymbol{x}}^0 - \widehat{\boldsymbol{x}} \right\rangle + \frac{L'}{2} \left\| \widetilde{\boldsymbol{x}}^0 - \widehat{\boldsymbol{x}} \right\|_2^2 \leq \frac{1}{2L'} \left\| \nabla_{k+K} F(\widehat{\boldsymbol{x}}) \right\|_2^2 + L' \left\| \widetilde{\boldsymbol{x}}^0 - \widehat{\boldsymbol{x}} \right\|_2^2.$$

Recall that

$$\tau(\widehat{\boldsymbol{x}}) = \frac{5\omega}{\alpha} \left\| \nabla_{3k+K} F(\widehat{\boldsymbol{x}}) \right\|_2 + \frac{1}{\alpha L} \left\| \nabla_{3k+K} F(\widehat{\boldsymbol{x}}) \right\|_2^2.$$

Hence using $\sqrt{a+b+c+d} \leq \sqrt{a} + \sqrt{b} + \sqrt{c} + \sqrt{d}$ gives

$$\begin{aligned} \mathbb{E}\left[\sqrt{\max\{F(\widetilde{\boldsymbol{x}}^s) - F(\widehat{\boldsymbol{x}}), 0\}} \right] \leq \sqrt{L'} \left(\frac{2}{3} \right)^{\frac{s}{2}} \left\| \widetilde{\boldsymbol{x}}^0 - \widehat{\boldsymbol{x}} \right\|_2 + \sqrt{\frac{5\omega}{\alpha} \left\| \nabla_{3k+K} F(\widehat{\boldsymbol{x}}) \right\|_2} \\ + \left(\frac{1}{\alpha} + \sqrt{\frac{1}{2\alpha}} \right) \left\| \nabla_{3k+K} F(\widehat{\boldsymbol{x}}) \right\|_2. \end{aligned}$$

Finally, the RSC property immediately suggests that (see, e.g., Lemma 20 in Shen and Li (2017b))

$$\begin{aligned} \mathbb{E}\left[\left\| \widetilde{\boldsymbol{x}}^s - \widehat{\boldsymbol{x}} \right\|_2 \right] &\leq \sqrt{\frac{2}{\alpha}} \mathbb{E}\left[\sqrt{\max\{F(\widetilde{\boldsymbol{x}}^s) - F(\widehat{\boldsymbol{x}}), 0\}} \right] + \frac{2 \left\| \nabla_{k+K} F(\widehat{\boldsymbol{x}}) \right\|_2}{\alpha} \\ &\leq \sqrt{\frac{2L'}{\alpha}} \cdot \left(\frac{2}{3} \right)^{\frac{s}{2}} \left\| \widetilde{\boldsymbol{x}}^0 - \widehat{\boldsymbol{x}} \right\|_2 + \sqrt{\frac{10\omega}{\alpha^2} \left\| \nabla_{3k+K} F(\widehat{\boldsymbol{x}}) \right\|_2} \\ &\quad + \left(\sqrt{\frac{2}{\alpha^3}} + \frac{3}{\alpha} \right) \left\| \nabla_{3k+K} F(\widehat{\boldsymbol{x}}) \right\|_2. \end{aligned}$$

The proof is complete. ■

Appendix E. HT-SAGA

We demonstrate that the hard thresholding step can be integrated into SAGA (Defazio et al., 2014) as shown in Algorithm 2. Note that the only difference of Algorithm 2 and the one proposed in Defazio et al. (2014) is that we perform hard thresholding rather than proximal operator. Hence, our algorithm guarantees k-sparse solution.

Theorem 22 *Assume the same conditions as in Defazio et al. (2014). Further assume the optimum of* (4.1) *without the sparsity constraint happens to be k-sparse. Then, the sequence of the solutions produced by Algorithm 2 converges to the optimum with geometric rate for some properly chosen sparsity parameter k.*

Proof Define the Lyapunov function Z as follows:

$$Z^t := Z(\boldsymbol{x}^t, \{\boldsymbol{\phi}_i^t\}) = \frac{1}{n} \sum_{i=1}^n f_i(\boldsymbol{\phi}_i^t) - F(\widehat{\boldsymbol{x}}) - \frac{1}{n} \sum_{i=1}^n \left\langle \nabla f_i(\widehat{\boldsymbol{x}}), \boldsymbol{\phi}_i^t - \widehat{\boldsymbol{x}} \right\rangle + c \left\| \boldsymbol{x}^t - \widehat{\boldsymbol{x}} \right\|_2^2.$$

Algorithm 2 SAGA with Hard Thresholding (HT-SAGA)

Require: The current iterate $\boldsymbol{x}^t$ and of each $\nabla f_i(\boldsymbol{\phi}_i^t)$ at the end of iteration t, the step size η.
Ensure: The new iterate.
1: Pick $j \in \{1, 2, \cdots, n\}$ uniformly at random.
2: Take $\boldsymbol{\phi}_j^{t+1} = \boldsymbol{x}^t$ and store $\nabla f_j(\boldsymbol{\phi}_j^{t+1})$ in the table. All other entries in the table remain unchanged.
3: Update the new iterate $\boldsymbol{x}^{t+1}$ as follows:

$$\begin{aligned} \boldsymbol{b}^{t+1} &= \boldsymbol{x}^t - \eta \left[\nabla f_j(\boldsymbol{\phi}_j^{t+1}) - \nabla f_j(\boldsymbol{\phi}_j^t) + \frac{1}{n} \sum_{i=1}^n \nabla f_i(\boldsymbol{\phi}_i^t) \right], \\ \boldsymbol{x}^{t+1} &= \mathcal{H}_k\left(\boldsymbol{b}^{t+1} \right). \end{aligned}$$

We examine Z^{t+1}. We have

$$\begin{aligned} \mathbb{E}\left[\frac{1}{n} \sum_i f_i(\boldsymbol{\phi}_i^{t+1}) \right] &= \frac{1}{n} F(\boldsymbol{x}^t) + \left(1 - \frac{1}{n} \right) \frac{1}{n} \sum_i f_i(\boldsymbol{\phi}_i^t), \\ \mathbb{E}\left[-\frac{1}{n} \sum_i \left\langle \nabla f_i(\widehat{\boldsymbol{x}}), \boldsymbol{\phi}_i^{t+1} - \widehat{\boldsymbol{x}} \right\rangle \right] &= -\frac{1}{n} \left\langle \nabla F(\widehat{\boldsymbol{x}}), \boldsymbol{x}^t - \widehat{\boldsymbol{x}} \right\rangle \\ &\quad - \left(1 - \frac{1}{n} \right) \frac{1}{n} \sum_i \left\langle \nabla f_i(\widehat{\boldsymbol{x}}), \boldsymbol{\phi}_i^t - \widehat{\boldsymbol{x}} \right\rangle. \end{aligned}$$

Also,

$$c \left\| \boldsymbol{x}^{t+1} - \widehat{\boldsymbol{x}} \right\|_2^2 \leq c\nu \left\| \boldsymbol{b}^{t+1} - \widehat{\boldsymbol{x}} \right\|_2^2 = c\nu \left\| \boldsymbol{b}^{t+1} - \widehat{\boldsymbol{x}} + \eta \nabla F(\widehat{\boldsymbol{x}}) \right\|_2^2.$$

For the first term, we have

$$\begin{aligned} &c\nu \, \mathbb{E} \left\| \boldsymbol{b}^{t+1} - \widehat{\boldsymbol{x}} + \eta \nabla F(\widehat{\boldsymbol{x}}) \right\|_2^2 \\ \leq\ & c\nu(1 - \eta\alpha) \left\| \boldsymbol{x}^t - \widehat{\boldsymbol{x}} \right\|_2^2 + c\nu \left((1+\mu)\eta^2 - \frac{\eta}{L} \right) \mathbb{E} \left\| \nabla f_j(\boldsymbol{x}^t) - \nabla f_j(\widehat{\boldsymbol{x}}) \right\|_2^2 \\ &- \frac{2c\nu\eta(L - \alpha)}{L} \left[F(\boldsymbol{x}^t) - F(\widehat{\boldsymbol{x}}) - \left\langle \nabla F(\widehat{\boldsymbol{x}}), \boldsymbol{x}^t - \widehat{\boldsymbol{x}} \right\rangle \right] - c\nu\eta^2\mu \left\| \nabla F(\boldsymbol{x}^t) - \nabla F(\widehat{\boldsymbol{x}}) \right\|_2^2 \\ &+ 2c\nu(1 + \mu^{-1})\eta^2 L \left[\frac{1}{n} \sum_i f_i(\boldsymbol{\phi}_i^t) - F(\widehat{\boldsymbol{x}}) - \frac{1}{n} \sum_i \left\langle \nabla f_i(\widehat{\boldsymbol{x}}), \boldsymbol{\phi}_i^t - \widehat{\boldsymbol{x}} \right\rangle \right]. \end{aligned}$$

Therefore,

$$\begin{aligned} &\mathbb{E}[Z^{t+1}] - Z^t \\ \leq\ & -\frac{1}{\kappa} Z^t + \left(\frac{1}{n} - \frac{2c\nu\eta(L - \alpha)}{L} - 2c\nu\eta^2\alpha\mu \right) \left[F(\boldsymbol{x}^t) - F(\widehat{\boldsymbol{x}}) - \left\langle \nabla F(\widehat{\boldsymbol{x}}), \boldsymbol{x}^t - \widehat{\boldsymbol{x}} \right\rangle \right] \\ &+ \left(\frac{1}{\kappa} + 2c\nu(1 + \mu^{-1})\eta^2 L - \frac{1}{n} \right) \left[\frac{1}{n} \sum_i f_i(\boldsymbol{\phi}_i^t) - F(\widehat{\boldsymbol{x}}) - \frac{1}{n} \sum_i \left\langle \nabla f_i(\widehat{\boldsymbol{x}}), \boldsymbol{\phi}_i^t - \widehat{\boldsymbol{x}} \right\rangle \right] \\ &+ \left(\frac{c}{\kappa} - c\nu\eta\alpha \right) \left\| \boldsymbol{x}^t - \widehat{\boldsymbol{x}} \right\|_2^2 + \left((1+\mu)\eta - \frac{1}{L} \right) c\nu\eta \, \mathbb{E} \left\| \nabla f_j(\boldsymbol{x}^t) - \nabla f_j(\widehat{\boldsymbol{x}}) \right\|_2^2. \end{aligned}$$

In order to guarantee the convergence, we choose proper values for η, c, κ, μ and ν such that the terms in round brackets are non-positive. That is, we require

$$\begin{aligned}\frac{c}{\kappa} - c\nu\eta\alpha &\leq 0,\\ (1+\mu)\eta - \frac{1}{L} &\leq 0,\\ \frac{1}{n} - \frac{2c\nu\eta(L-\alpha)}{L} - 2c\nu\eta^2\alpha\mu &\leq 0,\\ \frac{1}{\kappa} + 2c\nu(1+\mu^{-1})\eta^2 L - \frac{1}{n} &\leq 0.\end{aligned}$$

Pick

$$\begin{aligned}\eta &= \frac{1}{2(\alpha n + L)},\\ \mu &= \frac{2\alpha n + L}{L},\\ \kappa &= \frac{1}{\nu\eta\alpha},\end{aligned}$$

we fulfill the first two inequalities. Pick

$$c = \frac{1}{2\eta(1-\eta\alpha)n}.$$

Then by the last two equalities, we require

$$1 - \eta\alpha \leq \nu \leq \frac{(1-\eta\alpha)L}{\eta\alpha(1-\eta\alpha)Ln + 1}.$$

On the other hand, by Theorem 1, we have

$$\nu > 1.$$

Thus, we require

$$1 < \nu \leq \frac{(1-\eta\alpha)L}{\eta\alpha(1-\eta\alpha)Ln + 1},$$

By algebra, the above inequalities has non-empty feasible set provided that

$$(6\alpha^2 - 8\alpha^2 L)n^2 + (14\alpha L - \alpha - 16\alpha L^2)n + 8L^2(1-L) < 0.$$

Due to $\alpha \leq L$, we know

$$n \geq \frac{14L + \sqrt{224L^3 + 1}}{2\alpha(8L-6)}$$

suffices where we assume $L > 3/4$. Picking

$$\nu = \frac{(1-\eta\alpha)L}{\eta\alpha(1-\eta\alpha)Ln + 1}$$

completes the proof. ■

References

Radosaw Adamczak, Alexander E. Litvak, Alain Pajor, and Nicole Tomczak-Jaegermann. Restricted isometry property of matrices with independent columns and neighborly polytopes by random sampling. *Constructive Approximation*, 34(1):61–88, 2011.

Alekh Agarwal, Sahand Negahban, and Martin J. Wainwright. Fast global convergence of gradient methods for high-dimensional statistical recovery. *The Annals of Statistics*, 40(5):2452–2482, 2012.

Bubacarr Bah and Jared Tanner. Improved bounds on restricted isometry constants for gaussian matrices. *SIAM Journal on Matrix Analysis Applications*, 31(5):2882–2898, 2010.

Bubacarr Bah and Jared Tanner. Bounds of restricted isometry constants in extreme asymptotics: Formulae for Gaussian matrices. *Linear Algebra and its Applications*, 441:88–109, 2014.

Sohail Bahmani, Bhiksha Raj, and Petros T. Boufounos. Greedy sparsity-constrained optimization. *Journal of Machine Learning Research*, 14(1):807–841, 2013.

Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael B. Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008.

Pierre C. Bellec, Guillaume Lecué, and Alexandre B. Tsybakov. Slope meets lasso: improved oracle bounds and optimality. *CoRR*, abs/1605.08651, 2016.

Peter J. Bickel, Ya'acov Ritov, and Alexandre B. Tsybakov. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, pages 1705–1732, 2009.

Jeffrey D. Blanchard and Jared Tanner. Performance comparisons of greedy algorithms in compressed sensing. *Numerical Linear Algebra with Applications*, 22(2):254–282, 2015.

Thomas Blumensath and Mike E. Davies. Iterative thresholding for sparse approximations. *Journal of Fourier Analysis and Applications*, 14(5-6):629–654, 2008.

Thomas Blumensath and Mike E. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265–274, 2009.

Tony T. Cai and Anru Zhang. Sharp RIP bound for sparse signal and low-rank matrix recovery. *Applied and Computational Harmonic Analysis*, 35(1):74–93, 2013.

Tony T. Cai, Lie Wang, and Guangwu Xu. New bounds for restricted isometry constants. *IEEE Transactions on Information Theory*, 56(9):4388–4394, 2010.

Emmanuel J. Candès. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, 346(9):589–592, 2008.

Emmanuel J. Candès and Terence Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

Emmanuel J. Candès and Terence Tao. The Dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2313–2351, 2007.

Emmanuel J. Candès and Michael B. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008.

Emmanuel J. Candès, Justin K. Romberg, and Terence Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

Venkat Chandrasekaran, Benjamin Recht, Pablo A. Parrilo, and Alan S. Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12(6):805–849, 2012.

Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

Wei Dai and Olgica Milenkovic. Subspace pursuit for compressive sensing signal reconstruction. *IEEE Transactions on Information Theory*, 55(5):2230–2249, 2009.

Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004.

Aaron Defazio, Francis R. Bach, and Simon Lacoste-Julien. SAGA: a fast incremental gradient method with support for non-strongly convex composite objectives. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems*, pages 1646–1654, 2014.

David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

David L. Donoho and Jared Tanner. Precise undersampling theorems. *Proceedings of the IEEE*, 98 (6):913–924, 2010.

David L. Donoho and Yaakov Tsaig. Fast solution of ℓ_1-norm minimization problems when the solution may be sparse. *IEEE Transactions on Information Theory*, 54(11):4789–4812, 2008.

David L. Donoho, Michael Elad, and Vladimir N. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory*, 52(1): 6–18, 2006.

David L. Donoho, Iain Johnstone, and Andrea Montanari. Accurate prediction of phase transitions in compressed sensing via a connection to minimax denoising. *IEEE Transactions on Information Theory*, 59(6):3396–3433, 2013.

John C. Duchi and Yoram Singer. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, 10:2899–2934, 2009.

Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

Simon Foucart. Hard thresholding pursuit: An algorithm for compressive sensing. *SIAM Journal on Numerical Analysis*, 49(6):2543–2563, 2011.

Simon Foucart. Sparse recovery algorithms: Sufficient conditions in terms of restricted isometry constants. In *Approximation Theory XIII: San Antonio 2010*, pages 65–77. Springer, New York, NY, 2012.

Simon Foucart and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. Applied and Numerical Harmonic Analysis. Birkhäuser, 2013.

Prateek Jain, Ambuj Tewari, and Inderjit S. Dhillon. Orthogonal matching pursuit with replacement. In *Proceedings of the 25th Annual Conference on Neural Information Processing Systems*, pages 1215–1223, 2011.

Prateek Jain, Ambuj Tewari, and Purushottam Kar. On iterative hard thresholding methods for high-dimensional M-estimation. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems*, pages 685–693, 2014.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems*, pages 315–323, 2013.

John Langford, Lihong Li, and Tong Zhang. Sparse online learning via truncated gradient. *Journal of Machine Learning Research*, 10:777–801, 2009.

Ping Li, Cun-Hui Zhang, and Tong Zhang. Compressed counting meets compressed sensing. In *Proceedings of The 27th Conference on Learning Theory*, pages 1058–1077, 2014.

Xingguo Li, Tuo Zhao, Raman Arora, Han Liu, and Jarvis Haupt. Stochastic variance reduced optimization for nonconvex sparse learning. *CoRR*, abs/1605.02711, 2016.

Po-Ling Loh and Martin J. Wainwright. High-dimensional regression with noisy and missing data: Provable guarantees with non-convexity. *The Annals of Statistics*, 40(3):1637–1664, 2012.

Po-Ling Loh and Martin J. Wainwright. Regularized m-estimators with nonconvexity: statistical and algorithmic theory for local optima. *Journal of Machine Learning Research*, 16:559–616, 2015.

Zongming Ma. Sparse principal component analysis and iterative thresholding. *The Annals of Statistics*, 41(2):772–801, 2013.

Qun Mo. A sharp restricted isometry constant bound of orthogonal matching pursuit. *CoRR*, abs/1501.01708, 2015.

Qun Mo and Yi Shen. A remark on the restricted isometry property in orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 58(6):3654–3656, 2012.

Deanna Needell and Joel A. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.

Deanna Needell and Roman Vershynin. Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit. *IEEE Journal of Selected Topics in Signal Processing*, 4(2):310–316, 2010.

Sahand Negahban, Pradeep Ravikumar, Martin J. Wainwright, and Bin Yu. A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems*, pages 1348–1356, 2009.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87 of *Applied Optimization*. Springer US, 2004.

Nam H. Nguyen, Deanna Needell, and Tina Woolf. Linear convergence of stochastic iterative greedy algorithms with sparse constraints. *CoRR*, abs/1407.0088, 2014.

Bruno A. Olshausen and David J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision research*, 37(23):3311–3325, 1997.

Art Owen and Yi Zhou. Safe and effective importance sampling. *Journal of the American Statistical Association*, 95(449):135–143, 2000.

Yagyensh C. Pati, Ramin Rezaiifar, and Perinkulam S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Conference Record of The 27th Asilomar Conference on Signals, Systems and Computers*, pages 40–44, 1993.

Garvesh Raskutti, Martin J. Wainwright, and Bin Yu. Minimax rates of estimation for high-dimensional linear regression over ℓ_q-balls. *IEEE Transactions on Information Theory*, 57(10): 6976–6994, 2011.

Nicolas Le Roux, Mark W. Schmidt, and Francis R. Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems*, pages 2672–2680, 2012.

Jie Shen and Ping Li. A tight bound of hard thresholding. *CoRR*, abs/1605.01656, 2016.

Jie Shen and Ping Li. On the iteration complexity of support recovery via hard thresholding pursuit. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3115–3124, 2017a.

Jie Shen and Ping Li. Partial hard thresholding: Towards A principled analysis of support recovery. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems*, pages 3127–3137, 2017b.

Robert Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Joel A. Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004.

Joel A. Tropp and Anna C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007.

Joel A. Tropp and Stephen J. Wright. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 98(6):948–958, 2010.

Martin J. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (Lasso). *IEEE Transactions on Information Theory*, 55(5): 2183–2202, 2009.

Jian Wang and Byonghyo Shim. On the recovery limit of sparse signals using orthogonal matching pursuit. *IEEE Transactions on Signal Processing*, 60(9):4973–4976, 2012.

Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11:2543–2596, 2010.

Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

Xiao-Tong Yuan and Tong Zhang. Truncated power method for sparse eigenvalue problems. *Journal of Machine Learning Research*, 14(1):899–925, 2013.

Xiao-Tong Yuan, Ping Li, and Tong Zhang. Gradient hard thresholding pursuit. *Journal of Machine Learning Research*, 18(166):1–43, 2018.

Tong Zhang. Sparse recovery with orthogonal matching pursuit under RIP. *IEEE Transactions on Information Theory*, 57(9):6215–6221, 2011.

Journal of Machine Learning Research 18 (2018) 1-48 Submitted 9/16; Revised 4/17; Published 4/18

Estimation of Graphical Models through Structured Norm Minimization

Davoud Ataee Tarzanagh TARZANAGH@UFL.EDU
Department of Mathematics
UF Informatics Institute
University of Florida
Gainesville, FL 32611-8105, USA

George Michailidis GMICHAIL@UFL.EDU
Department of Statistics
UF Informatics Institute
University of Florida
Gainesville, FL 32611-8545, USA

Editor: Bert Huang

Abstract

Estimation of Markov Random Field and covariance models from high-dimensional data represents a canonical problem that has received a lot of attention in the literature. A key assumption, widely employed, is that of *sparsity* of the underlying model. In this paper, we study the problem of estimating such models exhibiting a more intricate structure comprising simultaneously of *sparse, structured sparse* and *dense* components. Such structures naturally arise in several scientific fields, including molecular biology, finance and political science. We introduce a general framework based on a novel structured norm that enables us to estimate such complex structures from high-dimensional data. The resulting optimization problem is convex and we introduce a linearized multi-block alternating direction method of multipliers (ADMM) algorithm to solve it efficiently. We illustrate the superior performance of the proposed framework on a number of synthetic data sets generated from both random and structured networks. Further, we apply the method to a number of real data sets and discuss the results.

Keywords: Markov Random Fields, Gaussian covariance graph model, structured sparse norm, regularization, alternating direction method of multipliers (ADMM), convergence.

1. Introduction

There is a substantial body of literature on methods for estimating network structures from high-dimensional data, motivated by important biomedical and social science applications; see Barabási and Albert (1999); Liljeros et al. (2001); Robins et al. (2007); Guo et al. (2011a); Danaher et al. (2014); Friedman et al. (2008); Tan et al. (2014); Guo et al. (2015). Two powerful formalisms have been employed for this task, the Markov Random Field (MRF) model and the Gaussian covariance graph model (GCGM). The former captures statistical conditional dependence relationships amongst random variables that correspond to the network nodes, while the latter to marginal associations. Since in most applications the number of model parameters to be estimated far exceeds the available sample size, the assumption of sparsity is made and imposed through regularization. An ℓ_1 penalty on the parameters encoding the network edges is the most common choice; see Friedman et al. (2008); Karoui (2008); Cai and Liu (2011); Xue et al. (2012), which can also be interpreted from the Bayesian perspective as using an independent double-exponential prior distribution on each edge parameter. Consequently, this approach encourages sparse uniform network structures that may not be the most suitable choice for many real world applications, which in turn have *hub* nodes or *dense subgraphs*. As argued in Barabási and Albert (1999); Liljeros et al. (2001); Newman (2001); Li et al. (2005); Fortunato (2010); Newman (2012) many networks exhibit different structures at different scales. An example includes a densely connected subgraph, also known as a *community* in the social networks literature. Such structures in social interaction networks may correspond to groups of people sharing common interests or being co-located (Traud et al., 2011; Newman and Girvan, 2004), while in biological systems to groups of proteins responsible for regulating or synthesizing chemical products (Guimera and Amaral, 2005; Lewis et al., 2010; see, Figure 3 for an example). Hence, in many applications, simple sparsity or alternatively, a dense structure fails to capture salient features of the true underlying mechanism that gave rise to the available data.

In this paper, we introduce a framework based on a novel structured sparse norm that allows us to recover such complex structures. Specifically, we consider Markov Random Field and covariance models where the parameter of interest, Θ can be expressed as the superposition of sparse, structured sparse and dense components as follows:

$$\Theta = \underbrace{Z_1 + Z_1^\top}_{\text{sparse part}} + \underbrace{Z_2 + Z_2^\top + \cdots + Z_n + Z_n^\top}_{\text{structured sparse part}} + \underbrace{E}_{\text{dense part}}, \tag{1}$$

where Z_1 is a sparse matrix, $Z_2, \ldots, Z_n$ are the set of $n-1$ structured sparse matrices (see, Figure 3 for an example of such structured matrices), and E is a dense matrix having possibly very many small, non-zero entries. As shown in Figure 3, the elements of Z_1 represent edges between non-structured nodes, and the non-zero parts of structured matrices $Z_2, \ldots, Z_n$ correspond to densely connected subgraphs (communities).

We elaborate more on the decomposition proposed above. We start by discussing on the sparse and structured sparse component and then elaborate on the dense component. Traditional sparse (lasso Tibshirani, 1996; Friedman et al., 2008) and group sparse (group lasso Yuan and Lin, 2007; Jacob et al., 2009; Obozinski et al., 2011) are tailor-made to estimate and recover sparse and structured sparse model structures, respectively. However, these methods can not accommodate different structures, unless users specify a *priori* the structure of interest (e.g. hub nodes and sparse components), thus severely limiting their application scope. On the other hand, the general framework introduced, is capable of estimating from high-dimensional data, *groups with overlaps*, *hubs* and *dense subgraphs*, with the size and location of such structures *not known a priori*.

Next, we discuss the role of the dense component E. In many applications, the data generation mechanism may correspond to a true sparse or structured sparse structure, "corrupted" by a dense component comprising of possible many small entries. A simple example of such a generating mechanism in linear models would have the regression coefficient being sparse with a few large entries and a more dense component having possibly many small, nonzero entries. In such instances, a pure sparse model formulation may not perform particularly well due to the presence of the dense component and may require very careful

tuning to recover the sparse component of interest. This line of reasoning is also adopted in Chernozhukov et al. (2017). Note however, that the model may also be used in settings where there is a significant dense component; however, as discussed in Chernozhukov et al. (2017) recovery of the individual component is not guaranteed. Hence, in this work we adopt the viewpoint that E represents a small "perturbation" of the sparse+structured sparse structure. To achieve these goals, it leverages a new structured norm that is used as the regularization term of the corresponding objective function.

The resulting optimization problem is solved through a multi-block ADMM algorithm. A key technical innovation is the development of a linearized ADMM algorithm that avoids introducing auxiliary variables which is a common strategy in the literature. We establish the global convergence of the proposed algorithm and illustrate its efficiency through numerical experimentation. The algorithm takes advantage of the special structure of the problem formulation and thus is suitable for large instances of the problem. To the best of our knowledge, this is the first work that gives global convergence guarantees for linearized multi-block ADMM with Gauss-Seidel updates, which is of interest in its own accord.

The remainder of the paper is organized as follows: In Section 2, we present the new structured norm used as the regularization term in the objective function of the Markov Random Field, covariance graph, regression and vector auto-regression models. In Section 3, we introduce an efficient multi-block ADMM algorithm to solve the problem, and provide the convergence analysis of the algorithm. In Section 4, we illustrate the proposed framework on a number of synthetic and real data sets, while some concluding remarks are drawn in Section 5.

2. A General Framework for Learning under Structured Sparsity

We start by introducing key definitions and associated notation.

2.1 Symmetric Structured Overlap Norm

Let X be an $m \times p$ data matrix, Θ be a $p \times p$ symmetric matrix containing the parameters of interest of the statistical loss function $\mathcal{G}(X, \Theta)$. The most popular assumption used in the literature is that Θ is *sparse* and can be successfully recovered from high-dimensional data by solving the following optimization problem

$$\underset{\Theta \in \mathcal{S}}{\text{minimize}} \quad \mathcal{G}(X, \Theta) + \lambda \big\|\Theta\big\|_1, \tag{2}$$

where $\mathcal{S}$ is some set depending on the loss function; λ is s a non-negative regularization constant; and $\|.\|_1$ denotes the ℓ_1 norm or the sum of the absolute values of the matrix elements.

To explicitly model different structures in the parameter Θ, we introduce the following *symmetric structured overlap norm (SSON)*:

Definition 1 ***(Symmetric Structured Overlap Norm)****. Let Θ be a $p \times p$ symmetric matrix containing the model parameters of interest. The symmetric structured overlap norm for a set of partitioned matrices $Z_1, \ldots, Z_n$ is given by,*

$$\begin{aligned}
\underset{Z_1,\ldots,Z_n,\ E}{\text{minimize}} \quad \Omega(\Theta, Z_1, \ldots, Z_n, E) \ &:= \ \lambda_1 \|Z_1 - \text{diag}(Z_1)\|_1 \\
&+ \ \sum_{i=2}^{n} \hat{\lambda}_i \|Z_i - \text{diag}(Z_i)\|_1 + \lambda_i \sum_{j=1}^{l_i} \|(Z_i - \text{diag}(Z_i))_j\|_F \\
&+ \ \frac{\lambda_e}{2} \|E\|_F^2, \\
\Theta \ &= \ \sum_{i=1}^{n} \big(Z_i + Z_i^\top\big) + E,
\end{aligned} \tag{3}$$

where $\{\lambda_i\}_{i=1}^n$ and $\{\hat{\lambda}_i\}_{i=2}^n$ are nonnegative regularization constants; l_i is the number of blocks of the partitioned matrix Z_i; $(Z_i - \text{diag}(Z_i))_j$ is the jth block of the partitioned matrix Z_i; E is an unstructured noise matrix; $\|.\|_1$ denotes the ℓ_1 norm or the sum of the absolute values of the matrix elements; and $\|.\|_F$ the Frobenius norm.

We note that the overlap norm defined by Mohan et al. (2012); Tan et al. (2014) encourages the recovery of matrices that can be expressed as a union of few rows and the corresponding columns (i.e. hub nodes). However, SSON represents a new symmetric and significantly more general variant of the overlap norm that promotes matrices that can be expressed as the sum of symmetric structured matrices. Moreover, unlike the previous group sparsity and the latent group lasso discussed in Yuan and Lin (2007); Jacob et al. (2009); Obozinski et al. (2011) that require users to specify structures of interest *a priori*, the SSON achieves a similar objective in an *agnostic manner*, relying only on how well such structures fit the observed data.

In many applications, such as regression models, we are interested in modeling different structures in a parameter vector θ. In these cases, we have the following definition as a special case of SSON:

Definition 2 *Let θ be a $p \times 1$ vector containing the model parameters of interest. The structured overlap norm for a set of partitioned vectors $z_1, \ldots, z_n$ is given by,*

$$\begin{aligned}
\underset{z_1,\ldots,z_n,\ e}{\text{minimize}} \quad \omega(\theta, z_1, \ldots, z_n, e) \ &:= \ \lambda_1 \|z_1\|_1 + \sum_{i=2}^{n} \hat{\lambda}_i \|z_i\|_1 + \lambda_i \sum_{j=1}^{l_i} \|z_{ij}\|_2 + \frac{\lambda_e}{2} \|e\|_2^2, \\
\theta \ &= \ z_1 + z_2 + \cdots + z_n + e,
\end{aligned} \tag{4}$$

where $\{\lambda_i\}_{i=1}^n$ and $\{\hat{\lambda}_i\}_{i=2}^n$ are nonnegative regularization constants; l_i is the number of blocks of the partitioned vector z_i; z_{i_j} is the jth block of the partitioned vector z_i (see, Figure 1); e is an unstructured noise vector; $\|.\|_1$ denotes the ℓ_1 norm or the sum of the absolute values of the vector elements; and $\|.\|_2$ the two norm.

Remark 3 *In the formulation of the problem, λ_1, $\{\hat{\lambda}_2, \ldots, \hat{\lambda}_n, \lambda_2, \ldots, \lambda_n\}$, and λ_e are tuning parameters corresponding to the sparse component Z_1, the structured components $\{Z_2, \ldots Z_n\}$ and the dense (noisy) component E, respectively. While the nonzero components may be clustered into groups, the nonzero groups may also be sparse. The latter can be achieved by (3) when $\{\hat{\lambda}_2, \ldots, \hat{\lambda}_n\}$ are positive constants.*

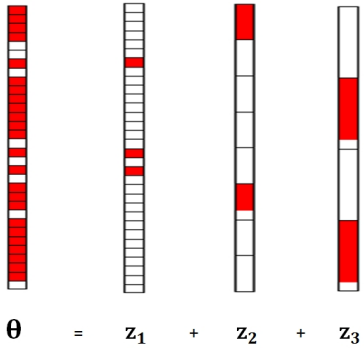


Figure 1: Decomposition of a vector θ into partitioned vectors z_1, z_2 and z_3, where z_1 is sparse, z_2 and z_3 are structured sparse vectors. White and red elements are zero and non-zero in the model parameter vector θ, respectively.

Remark 4 *The SSON admits the lasso (Tibshirani, 1996), the group lasso with overlaps (Jacob et al., 2009; Obozinski et al., 2011) and the ridge shrinkage (Hoerl and Kennard, 1970) methods as three extreme cases, by respectively setting* $\{\hat{\lambda}_2, \ldots, \hat{\lambda}_n, \lambda_2, \ldots, \lambda_n, \lambda_e\} \to \infty$, $\{\lambda_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_n, \lambda_e\} \to \infty$, *and* $\{\lambda_1, \ldots, \lambda_n, \hat{\lambda}_2, \ldots, \hat{\lambda}_n\} \to \infty$[1].

Note that SSON is rather different from the sparse group lasso, which also uses a combination of ℓ_1 and ℓ_G penalization, where $\|.\|_G$ is the group lasso norm. The sparse group lasso penalty is $\bar{\omega}(\theta) = \lambda_1\|\theta\|_1 + \lambda_2\|\theta\|_G$, and thus the includes lasso and group lasso as extreme cases corresponding to $\lambda_2 = 0$ and $\lambda_1 = 0$, respectively. However, $\bar{\omega}(\theta)$ does not split θ into a sparse and a group sparse part and will produce a sparse solution as long as $\lambda_1 > 0$. Hence, the sparse group lasso method can be thought of as a sparsity-based method with additional shrinkage by $\|\theta\|_G$. The group sparsity processes data very differently from SSON and consequently has very different prediction risk behavior. The same argument illustrates the advantages of the proposed SSON penalty over the well-known elastic net penalty. The elastic net is a combination of lasso and ridge penalties (Zou and Hastie, 2005). However, the elastic net does not split θ into a sparse and a dense component. Our results show that SSON tends to perform no worse than, and often performs significantly better than ridge, lasso, group lasso or elastic net with penalty levels chosen by cross-validation.

Remark 5 *In order to encourage different structures in the parameter matrix* Θ*, we consider the Frobenius norm of blocks of partitioned matrices, which leads to recovery of dense subgraphs. Other values for the norm of such blocks are also possible; e.g. the* ℓ_∞ *norm.*

Remark 6 *The matrix* E *is an important component of the SSON framework.*

It enables to develop a convergent multi-block ADMM *to solve the problem of estimating a structured Markov Random Field or covariance model. Note that in general, a direct extension of ADMM to multi-block convex minimization problems is* not *necessarily convergent even without linearization of the corresponding subproblems as shown in Chen et al. (2016).*

From a performance standpoint, our results show that adding a ridge penalty term $\frac{\lambda_e}{2}\|E\|_F^2$ *to the structured norm is provably effective in correctly identifying the underlying structures in the presence of noisy data (Zou and Hastie, 2005; Chernozhukov et al., 2017) (see, Figure 2 for an example of decomposition* (1) *in the presense of noise for covariance matrix estimation.)*

1. For example, with $\lambda_e \to \infty$, we set $\frac{\lambda_e}{2}\|E\|_F^2 = 0$ when $E = 0$, so the problem is well-defined.

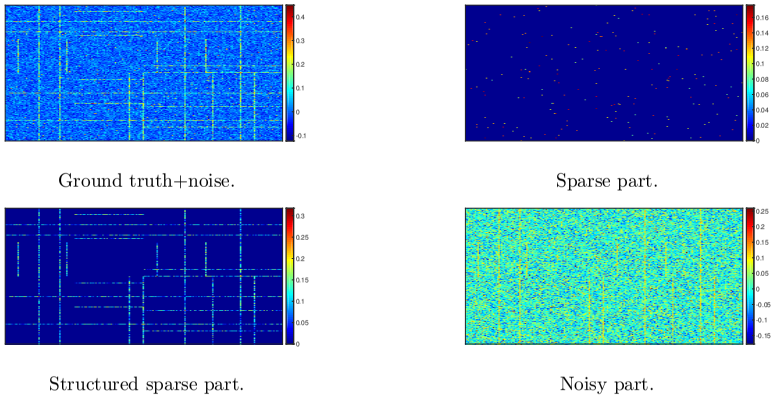


Figure 2: Heat map of the covariance matrix Θ_3 decomposed into sparse and structured sparse parts in the presence of noise, estimated by SSON using problem (11).

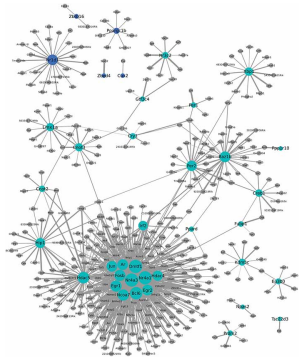
Next, we discuss the use of the SSON as a regularizer for maximum likelihood estimation of the following popular statistical models: (i) members of the Markov Random Field family including the Gaussian graphical model, the Gaussian graphical model with latent variables and the binary Ising model, (ii) the Gaussian covariance graph model and (iii) the classical regression and the vector auto-regression models. For the sake of completeness, we provide a complete, but succinct description of the corresponding models and the proposed regularization.

2.2 Structured Gaussian Graphical Models

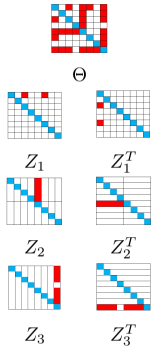
Let X be a data matrix consisting of p-dimensional samples from a zero mean Gaussian distribution,

$$x_1, \ldots, x_m \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma).$$

In order to obtain a sparse and interpretable estimate of the precision matrix Σ^{-1} that captures conditional dependence relationships, many authors have considered the well-known



(a) Postpartum NAC Gene Network (Zhao et al., 2014).



(b) Examples of partitioned matrices for the underlying network in (a).

Figure 3: The figure illustrates that block partitions through structured matrices could be set based on a desire for interpretability of the resulting estimated network structure. Panel (a) shows example of structured gene network, while panel (b) provides decomposition into structured matrices for the network in (a). Blue elements are diagonal ones, white elements are zero and red elements are non-zero in the model parameter matrix Θ. The structured penalty function (3) is then applied to each block for matrices $\{Z_i\}_{i=1}^n$.

graphical lasso problem (Friedman et al., 2008; Rothman et al., 2008) in the form of (2) with loss function

$$\mathcal{G}_1(X, \Theta_1) := \text{trace}(\hat{\Sigma}\Theta_1) - \log\det\Theta_1, \qquad \Theta_1 \in \mathcal{S}, \tag{5}$$

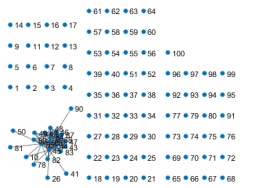
where $\hat{\Sigma}$ is the empirical covariance matrix of X; Θ_1 is the estimate of the precision matrix Σ^{-1}; and $\mathcal{S}$ is the set of $p \times p$ symmetric positive definite matrices.

As is well known, the norm penalty in (2) encourages zeros (sparsity) in the solution. However, as previously argued, many biological and social network applications exhibit more complex structures than mere sparsity. Using the proposed SSON, we define the following objective function for the problem at hand:

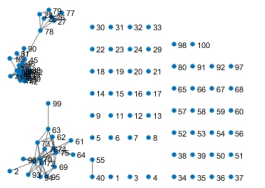
$$\begin{aligned} \underset{\Theta_1, Z_1, \ldots, Z_n \in \mathcal{S},\ E}{\text{minimize}} \quad & \mathcal{G}_1(X, \Theta_1) + \Omega(\Theta_1, Z_1, \ldots, Z_n, E), \\ \Theta_1 \quad = \quad & \sum_{i=1}^{n} \left(Z_i + Z_i^\top\right) + E, \end{aligned} \tag{6}$$

where Θ_1 is the model parameter matrix and $\Omega(\Theta_1, Z_1, \ldots, Z_n, E)$ the corresponding SSON defined in (3).

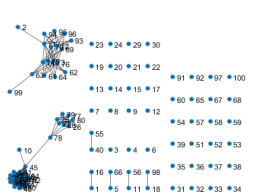
Formulation (6) allows us to obtain more accurate and compact network estimates than conventional methods whenever the network exhibits different structures. Moreover, our formulation does not require *a priori* knowledge of the underlying network structure (i.e. which nodes in the network form densely connected subgraphs (see, Figure 4)).



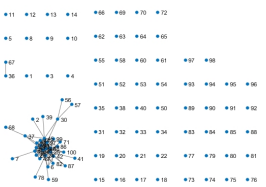
(a) Graphical lasso regularization.



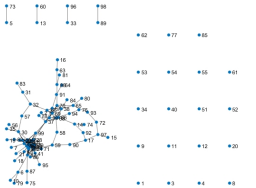
(b) SSON based regularization.



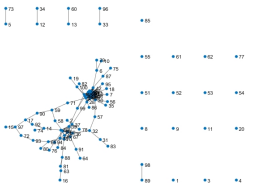
(c) Ground truth.



(d) Graphical lasso regularization.



(e) SSON based regularization.



(f) Ground truth.

Figure 4: Estimates from the SSON based regularization on two examples of Gaussian graphical models comprising of $p = 100$ nodes, using in (4b) three structured matrices and in (4e) four structured matrices.

In Figure 4, the performance of our proposed approach is illustrated on two simulated data sets exhibiting different structures (sub-figures (4c) and (4f)); it can be seen that the proposed SSON based graphical lasso (sub-figures (4b) and (4e)) can recover the network structure much better than the popular graphical lasso based estimator (Friedman et al., 2008) (subfigures (4a) and (4d)).

2.3 Structured Ising Model

Another popular graphical model, suitable for binary or categorical data, is the Ising one (Ising, 1925). It is assumed that observations $x_1, \ldots, x_m$ are independent and identically distributed from

$$f(x, \Theta_2) = \frac{1}{\mathbb{W}(\Theta_2)} \exp\Big(\sum_{j=1}^{p} \theta_{jj} x_j + \sum_{1 \le j < j' \le p} \theta_{jj'} x_j x_{j'}\Big), \tag{7}$$

where $\mathbb{W}(\Theta_2)$ is the partition function, which ensures that the density sums to one. Here, Θ_2 is a $p \times p$ symmetric matrix that specifies the network structure: $\theta_{jj'} = 0$ implies that the jth and j'th variables are conditionally independent given the remaining ones.

Several papers proposing estimation procedures for this model have been published. Lee et al. (2007) considered maximizing an ℓ_1-penalized log-likelihood for this model. Due to the difficulty in computing the log-likelihood with the expensive partition function, several authors have considered alternative approaches. For instance, Ravikumar et al. (2011)

proposed a neighborhood selection approach. The latter proposal involves solving p logistic regressions separately (one for each node in the network), which leads to an estimated parameter matrix that is in general not symmetric. In contrast, several authors considered maximizing an ℓ_1-penalized pseudo-likelihood with a symmetric constraint on Θ_2 (Guo et al., 2011a,b). Under the model (7), the log-pseudo-likelihood for m observations takes the form

$$\mathcal{G}_2(X,\Theta_2) := \sum_{j=1}^{p}\sum_{j'=1}^{p}\theta_{jj'}(X^TX)_{jj'} - \sum_{i=1}^{m}\sum_{j=1}^{p}\log\Big(1+\exp\big(\theta_{jj}+\sum_{j'\neq j}\theta_{jj'}x_{ij'}\big)\Big), \tag{8}$$

We propose instead to impose the SSON on Θ_2 in (8) in order to estimate a binary network with different structures. This leads to the following optimization problem

$$\begin{aligned}\underset{\Theta_2,Z_1,\ldots,Z_n\in\mathcal{S},\ E}{\text{minimize}} \quad & \mathcal{G}_2(X,\Theta_2)+\Omega(\Theta_2,Z_1,\ldots,Z_n,E),\\ \Theta_2 \;&=\; \sum_{i=1}^{n}\big(Z_i+Z_i^\top\big)+E,\end{aligned} \tag{9}$$

where Θ_2 is the model parameter matrix and $\Omega(\Theta_2,Z_1,\ldots,Z_n,E)$ the corresponding SSON defined in (3).

An interesting connection can be drawn between our technique and the Ising block model discussed in Berthet et al. (2016), which is a perturbation of the mean field approximation of the Ising model known as the Curie-Weiss model: the sites are partitioned into two blocks of equal size and the interaction between those within the same block is stronger than across blocks, to account for more order within each block. However, one can easily seen that the Ising block model is a special case of (9).

2.4 Structured Gaussian Covariance Graphical Models

Next, we consider estimation of a covariance matrix under the assumption that

$$x_1,\ldots,x_m \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,\Sigma).$$

This is of interest because the sparsity pattern of Σ specifies the structure of the marginal independence graph (Drton and Richardson, 2002, 2008).

Let Θ_3 be a $p\times p$ symmetric matrix containing the parameters of interest. Setting the loss function $\mathcal{G}_3(X,\Theta_3) := \frac{1}{2}\|\Theta_3-\hat{\Sigma}\|_F^2$, Xue et al. (2012) proposed to estimate the positive definite covariance matrix, $\hat{\Sigma}$ by solving

$$\underset{\Theta_3\in\mathcal{S}}{\text{minimize}} \quad \mathcal{G}_3(X,\Theta_3)+\lambda\|\Theta_3\|_1, \tag{10}$$

where $\hat{\Sigma}$ is the empirical covariance matrix, $\mathcal{S}=\{\Theta_3:\Theta_3\succeq\varepsilon I \text{ and } \Theta_3=\Theta_3^T\}$, and ε is a small positive constant. We extend (10) to accommodate structures of the covariance graph by imposing the SSON on Θ_3. This results in the following optimization problem

$$\begin{aligned}\underset{\Theta_3,Z_1,\ldots,Z_n\in\mathcal{S},\ E}{\text{minimize}} \quad & \mathcal{G}_3(X,\Theta_3)+\Omega(\Theta_3,Z_1,\ldots,Z_n,E),\\ \Theta_3 \;&=\; \sum_{i=1}^{n}\big(Z_i+Z_i^\top\big)+E.\end{aligned} \tag{11}$$

where Θ_3 is the model parameter matrix and $\Omega(\Theta_3,Z_1,\ldots,Z_n,E)$ the corresponding SSON defined in (3).

2.5 Structured Gaussian Graphical Models with latent variables

In many applications throughout science and engineering, it is often the case that some relevant variables are not observed. For the Gaussian Graphical model, Chandrasekaran et al. (2010) proposed a convex optimization problem to estimate it in the presence of latent variables. Let Θ_4 be a $p\times p$ symmetric matrix containing the parameters of interest. Setting $\mathcal{G}_4(X,\Theta_4) := \langle\Theta_4,\Sigma_O\rangle - \log\det\Theta_4$, their objective function is given by

$$\begin{aligned}\underset{\Theta_4,Z_1,Z_2\in\mathcal{S}}{\text{minimize}} \quad & \mathcal{G}_4(X,\Theta_4)+\alpha\|Z_1\|_1+\beta\,\text{trace}(Z_{n+1})+\mathbb{1}_{Z_{n+1}\succeq 0},\\ \Theta_4 \;&=\; Z_1-Z_{n+1},\end{aligned} \tag{12}$$

where Σ_O is the sample covariance matrix of the observed variables; α and β are positive constants; and the indicator function $\mathbb{1}_{Z_{n+1}\succeq 0}$ is defined as

$$\mathbb{1}_{Z_{n+1}\succeq 0} := \begin{cases}0, & \text{if } Z_{n+1}\succeq 0,\\ +\infty, & \text{otherwise.}\end{cases}$$

This convex optimization problem aims to estimate an inverse covariance matrix that can be decomposed into a sparse matrix Z_1 minus a low-rank matrix Z_{n+1} based on high-dimensional data.

Next, we extend the SSON to solve the latent variable graphical model selection. Problem (12) can be rewritten in the following equivalent form by introducing new variables $\{Z_i\}_{i=1}^n$:

$$\begin{aligned}\underset{\Theta_4,Z_1,\ldots,Z_n\in\mathcal{S},\ E}{\text{minimize}} \quad & \mathcal{G}_4(X,\Theta_4)+\Omega(\Theta_4,Z_1,\ldots,Z_n,E)+\lambda_{n+1}\,\text{trace}(Z_{n+1})+\mathbb{1}_{Z_{n+1}\succeq 0},\\ \Theta_4 \;&=\; \sum_{i=1}^{n}\big(Z_i+Z_i^\top\big)-Z_{n+1}+E,\end{aligned} \tag{13}$$

where Θ_4 is the model parameter matrix and $\Omega(\Theta_4,Z_1,\ldots,Z_n,E)$ the corresponding SSON defined in (3).

2.6 Structured Linear Regression and Vector Auto-Regression

The proposed SSON is also applicable to structured regression problems. Although this is not the main focus on this paper, nevertheless, we include a brief discussion, especially for lag selection in vector autoregressive models that are of prime interest in the analysis of high-dimensional time series data. The canonical formulation of the regularized regression problem is given by:

$$\min_{\theta\in\mathbb{R}^p}\|y-X\theta\|_2+\lambda\Psi(\theta). \tag{14}$$

where $\{(y_i,x_i)\}_{i=1}^m$, $y_i\in\mathbb{R}$, $x_i\in\mathbb{R}^p$, with $y=[y_1,\ldots,y_m]^\top$ being the response variable and $X=[x_1^\top,\ldots x_m^\top]$ a set of p-predictors that are assumed to be independently and identically distributed (i.i.d.); $\lambda>0$ is a regularization parameter and $\Psi(\theta)$ is a suitable norm. Specific

choices of $\Psi(.)$ lead to popular regularizers including the lasso -$\Psi(\theta) = \|\theta\|_1$- and the group lasso.

We propose instead to impose the SSON on θ in (14) in order to solve structured regression problems. Problem (14) can be rewritten in the following form by introducing new variables $\{z_i\}_{i=1}^n$ and e:

$$\begin{aligned} \underset{\theta, z_1, .., z_n, e}{\text{minimize}} \quad & \mathcal{G}(X, \theta) + \omega(\theta, z_1, \ldots, z_n, e), \\ \theta_4 \quad = \quad & z_1 + z_2 + \cdots + z_n + e, \end{aligned} \tag{15}$$

where $\mathcal{G}(X, \theta) = \|y - X\theta\|_2$; θ is the model parameter vector and $\omega(\theta, z_1, \ldots, z_n, e)$ the corresponding structured norm defined in (4).

Problem (15) can equivalently be thought of as a generalization of subspace clustering (Elhamifar and Vidal, 2009). Indeed, in order to segment the data into their respective subspaces, we need to compute an affinity vector θ that encodes the pairwise affinities between data vectors.

An interesting application of the SSON for multivariate regression problems is on structured estimation of *vector autoregression* (VAR) models (Lütkepohl, 2005), a popular model for economic and financial time series data (Tsay, 2005), dynamical systems (Ljung, 1998) and more recently brain function connectivity (Valdés-Sosa et al., 2005). The model captures both temporal and cross-dependencies between stationary time series. Formally, let $\{x_1, \ldots, x_m\}$ be a p-dimensional time series set of observations that evolve over time according to a lag-d model:

$$x_{t+1} = \sum_{k=1}^{d} \Theta_k^\top x_{t-k} + \epsilon_t, \qquad \epsilon_1, \ldots, \epsilon_{m-1} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma), \qquad t = 1, \ldots, m-1,$$

where $\{\Theta\}_{k=1}^d \in \mathbb{R}^{p \times p}$ are *transition matrices* for different lags, and $\{\epsilon_1, \ldots, \epsilon_{m-1}\}$ independent multivariate Gaussian *white noise* processes. The VAR process is assumed to be stable and stationary (bounded spectral density), while the noise covariance matrix Σ is assumed to be positive definite with bounded largest eigenvalue (Basu and Michailidis, 2015).

Given m observations $\{x_1, x_2, \ldots, x_m\}$ from a stationary VAR process, the lag-m VAR can be written is given by

$$\underbrace{\begin{bmatrix} x_m \\ x_{m-1} \\ \vdots \\ x_2 \end{bmatrix}}_{\mathbf{Y}} = \underbrace{\begin{bmatrix} {x_{m-1}}^\top \\ {x_{m-2}}^\top \\ \vdots \\ {x_1}^\top \end{bmatrix}}_{\mathbf{X}} \Theta + \underbrace{\begin{bmatrix} {\epsilon_{m-1}}^\top \\ {\epsilon_{m-2}}^\top \\ \vdots \\ {\epsilon_1}^\top \end{bmatrix}}_{\varepsilon}. \tag{16}$$

It can be seen that to estimate Θ one can solve the following least squares problem

$$\min_{\Theta \in \mathbb{R}^{p \times p}} \|Y - X\Theta\|_F. \tag{17}$$

However, as the number of component time series increases, the number of parameters to be estimated grows as dp^2 ; hence, structural assumptions are imposed to estimate them from

limited sample size. A popular choice is the lasso (Basu and Michailidis, 2015), that leads to sparse estimates. However, it does not incorporate the notion of lag selection, which could lead to certain spurious coefficients coming from further lags in the past. To address this problem, Basu et al. (2015) proposed a thresholded lasso estimate. However, our SSON can be used for lag selection, that guarantees that more recent lags are favored over further in the past ones.

Let Θ_5 be a $mp \times mp$ symmetric matrix containing the parameters of interest for all m lages of the problem. Setting the loss function $\mathcal{G}_5(X, \Theta_5) := \|Y - X\Theta_5\|$, we propose to estimate the transition matrix, Θ by solving the following optimization problem:

$$\begin{aligned} \min_{\Theta_5, Z_1, .., Z_n, E \in \mathbb{R}^{p \times p}} \quad & \mathcal{G}_5(X, \Theta_5) + \Omega(\Theta_5, Z_1, \ldots, Z_n, E), \\ \Theta_5 \quad = \quad & \sum_{i=1}^{n} \left(Z_i + Z_i^\top\right) + E, \end{aligned} \tag{18}$$

where Θ_5 is the estimate of the covariance matrix and $\Omega(\Theta_5, Z_1, \ldots, Z_n, E)$ the corresponding SSON defined in (3).

3. Multi-Block ADMM for Estimating Structured Network Models

Objective functions (6), (9), (11), (13), (15), and (18) involve separable convex functions, while the constraint is simply linear, and therefore they are suitable for ADMM based algorithms. We next introduce a linearized multi-block ADMM algorithm to solve these problems and establish its global convergence properties.

The alternating direction method of multipliers (ADMM) is widely used in solving structured convex optimization problems due to its superior performance in practice; see Scheinberg et al. (2010); Boyd et al. (2011); Hong and Luo (2017); Lin et al. (2015, 2016); Sun et al. (2015); Davis and Yin (2015); Hajinezhad and Hong (2015); Hajinezhad et al. (2016). On the theoretical side, Chen et al. (2016) provided a counterexample showing that the ADMM may *fail to converge* when the number of blocks exceeds *two*. Hence, many authors reformulate the problem of estimating a Markov Random Field model to a two block ADMM algorithm by grouping the variables and introducing auxiliary variables (Ma et al., 2013; Mohan et al., 2012; Tan et al., 2014). However, in the context of large-scale optimization problems, the grouping ADMM method becomes expensive due to its high memory requirements. Moreover, despite lack of convergence guarantees under standard convexity assumptions, it has been observed by many researchers that the unmodified multi-block ADMMs with Gauss-Seidel updates often outperform all its modified versions in practice (Wang et al., 2013; Sun et al., 2015; Davis and Yin, 2015).

Next, we present a *convergent multi-block* ADMM with Gauss-Seidel updates to solve convex problems (6), (9), (11), (13), and (18). The ADMM is constructed for an augmented Lagrangian function defined by

$$\begin{aligned} \mathcal{L}_\gamma(\Theta, Z_1, \ldots, Z_n, E; \Lambda) \quad = \quad & \mathcal{G}(X, \Theta) + f_1(Z_1) + \cdots + f_n(Z_n) + \frac{\lambda_e}{2}\|E\|_F^2 \\ - \quad & \langle \Lambda, \Theta - \sum_{i=1}^{n} Z_i + Z_i^\top - E \rangle + \frac{\gamma}{2}\|\Theta - \sum_{i=1}^{n} Z_i + Z_i^\top - E\|_F^2, \end{aligned} \tag{19}$$

where Λ is the Lagrange multiplier, γ a penalty parameter, $\mathcal{G}(X,\Theta)$ the loss function of interest and

$$\begin{aligned} f_1(Z_1) &:= \lambda_1\|Z_1 - \mathrm{diag}(Z_1)\|_1, \\ f_i(Z_i) &:= \hat{\lambda}_i\|Z_i - \mathrm{diag}(Z_i)\|_1 + \lambda_i \sum_{j=1}^{l_i} \|(Z_i - \mathrm{diag}(Z_i))_j\|_F, \quad i = 2,\ldots,n. \end{aligned} \tag{20}$$

In a typical iteration of the ADMM for solving (19), the following updates are implemented:

$$\Theta^{k+1} = \underset{\Theta}{\mathrm{argmin}} \quad \mathcal{G}(X,\Theta) + \frac{\gamma}{2}\|\Theta - B_0\|_F^2, \tag{21}$$

$$Z_i^{k+1} = \underset{Z_i}{\mathrm{argmin}} \quad f_i(Z_i) + \frac{\gamma}{2}\|Z_i + Z_i^\top - B_i\|_F^2, \qquad i = 1,\ldots n, \tag{22}$$

$$E^{k+1} = \underset{E}{\mathrm{argmin}} \quad f_e(E) + \frac{\gamma}{2}\|E - B_{n+1}\|_F^2, \tag{23}$$

$$\Lambda^{k+1} = \Lambda^k - \gamma(\Theta^{k+1} - \sum_{i=1}^{n} Z_i^{k+1} + {Z_i^{k+1}}^\top - E^{k+1}). \tag{24}$$

where

$$\begin{aligned} B_0 &= \sum_{i=1}^{n} Z_i^k + {Z_i^k}^\top + E^k + \frac{1}{\gamma}\Lambda^k, \\ B_1 &= \Theta^{k+1} - (\sum_{i=2}^{n} Z_i^k + {Z_i^k}^\top + E^k + \frac{1}{\gamma}\Lambda^k), \\ B_i &= \Theta^{k+1} - (\sum_{j=1}^{i-1} Z_j^{k+1} + {Z_j^{k+1}}^\top \\ &\quad + \sum_{j=i+1}^{n} Z_j^k + {Z_j^k}^\top + E^k + \frac{1}{\gamma}\Lambda^k), \qquad i = 2,\ldots n-1, \\ B_n &= \Theta^{k+1} - (\sum_{i=1}^{n-1} Z_i^{k+1} + {Z_i^{k+1}}^\top + E^k + \frac{1}{\gamma}\Lambda^k), \\ B_{n+1} &= \Theta^{k+1} - (\sum_{i=1}^{n} Z_i^{k+1} + {Z_i^{k+1}}^\top + \frac{1}{\gamma}\Lambda^k). \end{aligned} \tag{25}$$

To avoid introducing auxiliary variables and still solve subproblems (22) efficiently, we propose to approximate the subproblems (22) by linearizing the quadratic term of its objective function (see also Bolte et al., 2014; Lin et al., 2011; Yang and Yuan, 2013). With this linearization, the resulting approximation to (22) is then simple enough to have a closed-form solution. More specifically, letting $H_i(Z_i) = \frac{\gamma}{2}\|Z_i + Z_i^\top - B_i\|_F^2$, we define the following majorant function of $H_i(Z_i)$ at point Z_i^k,

$$H_i(Z_i) \le \gamma\Big(\frac{1}{2}\|Z_i^k + {Z_i^k}^\top - B_i\|_F^2 + \langle \nabla H_i(Z_i^k), Z_i - Z_i^k\rangle + \frac{\varrho}{2}\|Z_i - Z_i^k\|_F^2\Big), \tag{26}$$

where ϱ is a proximal parameter, and

$$\nabla H_i(Z_i^k) := 2(Z_i^k + {Z_i^k}^\top) - (B_i + B_i^\top), \tag{27}$$

Plugging (26) into (22), with simple algebraic manipulations, we obtain:

$$Z_i^{k+1} = \underset{Z_i}{\mathrm{argmin}} \quad f_i(Z_i) + \frac{\varrho\gamma}{2}\|Z_i - C_i\|_F^2, \qquad i = 1,\ldots n, \tag{28}$$

where $C_i = Z_i^k - \frac{1}{\varrho}\nabla H_i(Z_i^k)$.

The next result establishes the sufficient decrease property of the objective function given in (22), after a proximal map step computed in (28).

Lemma 7 *(Sufficient decrease property). Let $\varrho > \frac{L_{H_i}}{\gamma}$, where L_{H_i} is a Lipschitz constant of the gradient $\nabla H_i(Z_i)$ and γ is a penalty parameter defined in* (19)*. Then, we have*

$$f_i(Z_i^{k+1}) + H_i(Z_i^{k+1}) \le f_i(Z_i^k) + H_i(Z_i^k) - \frac{(\varrho\gamma - L_{H_i})}{2}\|Z_i^{k+1} - Z_i^k\|_F^2, \qquad i = 1,\ldots,n,$$

where $Z_i^{k+1} \in \mathbb{R}^{n\times n}$ defined by (28)*.*

Proof. The proof of this Lemma follows along similar lines to the proof of Lemma 3.2 in Bolte et al. (2014).

It is well known that (28) has a closed-form solution that is given by the shrinkage operation (Boyd et al., 2011):

$$\begin{aligned} Z_1^{k+1} &= \mathrm{Shrink}\Big(C_1, \frac{\lambda_1}{\varrho\gamma}\Big), \\ Z_{i_j}^{k+1} &= \max\big(1 - \frac{\lambda_i}{\varrho\gamma\|\mathrm{Shrink}(C_{i_j}, \frac{\hat{\lambda}_i}{\varrho\gamma})\|_F}, 0\big)\cdot \mathrm{Shrink}(C_{i_j}, \frac{\hat{\lambda}_i}{\varrho\gamma}), \quad \substack{i=2,\ldots n,\\ j=1,\ldots,l_i,} \end{aligned} \tag{29}$$

where $\mathrm{Shrink}(\cdot,\cdot)$ in (29) denotes the soft-thresholds operator, applied element-wise to a matrix A (Boyd et al., 2011):

$$\mathrm{Shrink}(A_{ij}, b) := \mathrm{sign}(A_{ij})\max\big(|A_{ij}| - b, 0\big) \qquad \substack{i=1,\ldots p,\\ j=1,\ldots,p,}.$$

Remark 8 *Note that in the case of solving problem* (13)*, one needs to add another block function $f_{n+1}(Z_{n+1}) := \lambda_{n+1}\,\mathrm{trace}(Z_{n+1}) + \mathbb{1}_{Z_{n+1}\succeq 0}$ to the augmented Lagrangian function* (19) *and update $\{C_i\}_{i=1}^n$. In this case, the proximal mapping of f_{n+1} is*

$$prox(f_{n+1}, \gamma, Z_{n+1}) := \underset{Z_{n+1}}{\mathrm{argmin}}\, f_{n+1}(Z_{n+1}) + \frac{\gamma}{2}\|Z_{n+1} - C_{n+1}\|_F^2, \tag{30}$$

where $C_{n+1} = \Theta^{k+1} - (\sum_{i=1}^n Z_i^{k+1} + {Z_i^{k+1}}^\top + E^k + \frac{1}{\gamma}\Lambda^k)$. It is easy to verify that (30) *has a closed-form solution given by*

$$Z_{n+1} = U\ \max(D - \frac{\lambda_{n+1}}{\gamma}, 0)\ U^T,$$

where UDU^T is the eigenvalue decomposition of C_{n+1} (see, Chandrasekaran et al., 2010; Ma et al., 2013 for more details).

The discussions above suggest that the following unmodified ADMM for solving (19) gives rise to an efficient algorithm.

Algorithm 1 Multi-Block ADMM Algorithm for Solving (19).

1: **Initialize** The parameters:

(a) Primal variables Θ, Z_1, ..., Z_n, E, to the $p \times p$ identity matrix.

(b) Dual variable Λ to the $p \times p$ zero matrix.

(c) Constants $\varrho, \lambda_e, \tau > 0$, and $\gamma \geq \sqrt{2}\lambda_e$.

(d) Nonnegative regularization constants $\lambda_1, \ldots, \lambda_n$, $\hat{\lambda}_2, \ldots, \hat{\lambda}_n$.

2: **Iterate** Until the stopping criterion $\|\Theta^k - \Theta^{k-1}\|_F^2 / \|\Theta^{k-1}\|_F \leq \tau$ is met:

(a) Update Θ:

$$\Theta^{k+1} = \underset{\Theta \in \mathcal{S}}{\operatorname{argmin}} \ \mathcal{G}(X, \Theta) + \frac{\gamma}{2}\|\Theta - B_0\|_F^2,$$

where B_0 is defined in (25).

(b) Update Z_i:

i. $Z_1^{k+1} = \mathrm{Shrink}\Big(C_1, \frac{\lambda_1}{\varrho\gamma}\Big)$,

ii. $Z_{ij}^{k+1} = \max\big(1 - \dfrac{\lambda_i}{\varrho\gamma\|\mathrm{Shrink}(C_{ij}, \frac{\hat{\lambda}_i}{\varrho\gamma})\|_F}, 0\big) \cdot \mathrm{Shrink}(C_{ij}, \frac{\hat{\lambda}_i}{\varrho\gamma}), \quad \substack{i=2,\ldots n,\\ j=1,\ldots l_i,}$

where C_i is defined in (28).

(c) Update E:

$$E^{k+1} = \underset{E}{\operatorname{argmin}} \ \tfrac{\lambda_e}{2}\|E\|_F^2 + \tfrac{\gamma}{2}\|E - B_{n+1}\|_F^2$$

where B_{n+1} is defined in (25).

(d) Update Λ:

$$\Lambda^{k+1} = \Lambda^k - \gamma(\Theta^{k+1} - \textstyle\sum_{i=1}^n Z_i^{k+1} + Z_i^{k+1^\top} - E^{k+1})$$

Remark 9 *The complexity of Algorithm 1 is of the same order as the graphical lasso (Friedman et al., 2008), the method in Tan et al. (2014) for hub node discovery and the algorithm used for estimation of sparse covariance matrices introduced by Xue et al. (2012). Indeed, one can easily see that with any set of structured matrices $\{Z_i\}_{i=1}^n$, the complexity of Algorithm 1 is equal to $O(p^3)$, which is the complexity of the eigen-decomposition for updating Θ in step 2(a).*

Since both the objective function and constraints of (19) become separable after using the linearization technique introduced in (26), the problem can be decomposed into $n+2$ smaller subproblems; the latter can be solved in a parallel and distributed manner with a small modification in Algorithm 1. Indeed, we can apply a Jacobian ADMM to solve (19)

with the following updates,

$$\begin{aligned}
\Theta^{k+1} &= \underset{\Theta}{\operatorname{argmin}} \quad \mathcal{G}(X, \Theta) + \frac{\gamma}{2}\|\Theta - B_0\|_F^2, \\
Z_i^{k+1} &= \underset{Z_i}{\operatorname{argmin}} \quad f_i(Z_i) + \frac{\varrho\gamma}{2}\|Z_i - C_i\|_F^2, \qquad i = 1, \ldots n, \\
E^{k+1} &= \underset{E}{\operatorname{argmin}} \quad f_e(E) + \frac{\gamma}{2}\|E - B_{n+1}\|_F^2, \\
\Lambda^{k+1} &= \Lambda^k - \gamma(\Theta^{k+1} - \sum_{i=1}^n Z_i^{k+1} + Z_i^{k+1^\top} - E^{k+1}).
\end{aligned} \tag{31}$$

where C_i is defined in (28) with

$$\begin{aligned}
B_0 &= \sum_{i=1}^n Z_i^k + Z_i^{k^\top} + E^k + \frac{1}{\gamma}\Lambda^k, \\
B_i &= \Theta^k - (\sum_{j=1}^{i-1} Z_j^k + Z_j^{k^\top} \\
&\quad + \sum_{j=i+1}^{n} Z_j^k + Z_j^{k^\top} + E^k + \frac{1}{\gamma}\Lambda^k), \qquad i = 2, \ldots n-1, \\
B_n &= \Theta^k - (\sum_{i=1}^{n-1} Z_i^k + Z_i^{k^\top} + E^k + \frac{1}{\gamma}\Lambda^k), \\
B_{n+1} &= \Theta^k - (\sum_{i=1}^{n} Z_i^k + Z_i^{k^\top} + \frac{1}{\gamma}\Lambda^k).
\end{aligned} \tag{32}$$

Intuitively, the performance of the Jacobian ADMM should be worse than the Gauss-Seidel version, because the latter always uses the latest information of the primal variables in the updates. We refer to Liu et al. (2015); Lin et al. (2015) for a detailed discussion on the convergence analysis of the Jacobian ADMM and its variants. On the positive side, we obtain a parallelizable version of the multi-block ADMM algorithm.

3.1 Convergence analysis

The next result establishes the global convergence of the standard multi-block ADMM for solving SSON based statistical learning problems, by using the Kurdyka- Lojasiewicz (KL) property of the objective function in (19).

Theorem 10 *The sequence $U^k := (\Theta^k, Z_1^k, \ldots, Z_n^k, E^k, \Lambda^k)$ generated by Algorithm 1 from any starting point converges to a stationary point of the problem given in* (19).

Proof. A detailed exposition is given in Appendix B.

4. Experimental Results

In this section, we present numerical results for Algorithm 1 (henceforth called SSONA), on both synthetic and real data sets. The results are organized in the following three sub-sections: in Section 4.1, we present numerical results on synthetic data comparing the performance of SSONA to that of grouping variables ADMM and also for assessing the accuracy in recovering a multi-layered structure in Markov Random Field and covariance graph models that constitute the prime focus in this paper. In Section 4.2 we use the proposed SSONA for feature selection in classification problems involving two real data sets in order to calibrate SSON performance with respect to an independent validation set. Finally, in Section 4.3, we analyze using SSONA on some other interesting real data sets from the social and biological sciences.

4.1 Experimental results for the SSON algorithm on graphical models based on synthetic data

Next, we evaluate the performance of SSONA on ten synthetic graphical model problems, comprising of $p = 100$, 500 and 1000 variables. The underlying network structure corresponds to an Erdős-Rényi model graph, a nearest neighbor graph and a scale-free random graph, respectively. The CONTEST [1] package is used to generate the synthetic graphs, and the UGM [2] package to implement Gibbs sampling for estimating the Ising Model. Based on the generated graph topologies, we consider the following settings for generating synthetic data sets:

I. **Gaussian graphical models**:
For a given number of variables p, we first create a symmetric matrix $E \in \mathbb{R}^{p\times p}$ by using CONTEST in a MATLAB environment. Given matrix E, we set Σ^{-1} equal to $E + (0.1 - \bar{\Lambda}_{\min}(E))\ I$, where $\bar{\Lambda}_{\min}(E)$ is the smallest eigenvalue of E and I denotes the identity matrix. We then draw $N = 5p$ i.i.d. vectors $x_1, \ldots, x_m$ from the gasserian distribution $\mathcal{N}(0, \Sigma)$ by using the *mvnrnd* function in MATLAB, and then compute a sample covariance matrix of the variables.

II. **Gaussian graphical models with latent variables:**
For a given number of variables p, we first create a matrix $\Sigma^{-1} \in \mathbb{R}^{(p+r)\times(p+r)}$ by using CONTEST as described in I. We then choose the sub-matrix $\Theta_O = \Sigma^{-1}(1:p, 1:p)$ as the ground truth matrix of the matrix Θ_4 and chose

$$\begin{aligned}\Theta_U &= \Sigma^{-1}(1:p,\ p+1:p+r)\big(\Sigma^{-1}(p+1:p+r,\ p+r:p+r)\big)^{-1} \\ &\quad \Sigma^{-1}(p+1:p+r,\ 1:p)\end{aligned}$$

as the ground truth matrix of the low rank matrix U. We then draw $N = 5p$ i.i.d. vectors $x_1, \ldots, x_m$ from the Gaussian distribution $\mathcal{N}(0, (\Theta_O - \Theta_U)^{-1})$, and compute the sample covariance matrix of the variables Σ_O.

III. **The Binary Network**:
To generate the parameter matrix Σ, we create an adjacency matrix as in Setup I by using CONTEST. Then, each of $N = 5p$ observations is generated through Gibbs sampling. We take the first 100000 iterations as our burn-in period, and then collect observations, so that they are nearly independent.

1. CONTEST is available at http://www.mathstat.strath.ac.uk/outreach/contest/
2. UGM is available at http://www.di.ens.fr/ mschmidt/Software/UGM.html

We compare SSONA to the following competing methods:

- **CovSel**, designed to estimate a sparse Gaussian graphical model (Friedman et al., 2008);
- **HGL**, focusing on learning a Gaussian graphical model having hub nodes (Tan et al., 2014);
- **PGADM**, designed to learn a Gaussian graphical model with some latent nodes (Ma et al., 2013);
- **Pseudo-Exact**, designed to learn a binary Ising graphical model (Höfling and Tibshirani, 2009);
- **glasso-SF**, Learning Scale Free Networks by reweighted ℓ_1 Regularization (Liu and Ihler, 2011);
- **GADMM**, A two block ADMM method with grouping variables.

All the algorithms have been implemented in the MATLAB R2015b environment on a PC with a 1.8 GHz processor and 6GB RAM memory. Further, all the algorithms are being terminated either when

$$\frac{\|\Theta^k - \Theta^{k-1}\|_F^2}{\|\Theta^{k-1}\|_F^2} \leq \tau, \qquad \tau = 1e-5,$$

or the number of iterations and CPU times exceed 1,000 and 10 minutes, respectively.

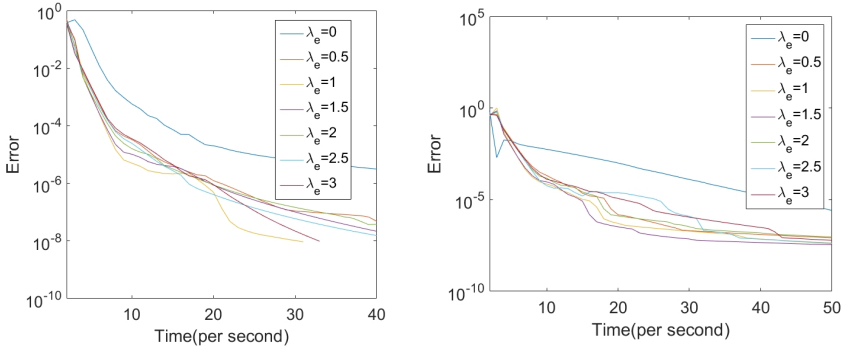
We found that in practice the computation cost for SSONA increases with the size of structured matrices. Therefore, we use a limited memory version of SSONA in our experimental results to obtain good accuracy. Block sizes in Figure 3 could be set based on a desire for interpretability of the resulting estimates. In this section, we choose four structured matrices with blocks of size

$$\begin{aligned}(Z_2)_j &= [1, \frac{p}{2}], \qquad j = 1\ldots, l_2,\\ (Z_3)_j &= [1, \frac{p}{5}], \qquad j = 1\ldots, l_3,\\ (Z_4)_j &= [1, \frac{p}{10}], \qquad j = 1\ldots, l_4,\\ (Z_5)_j &= [1, \frac{p}{20}], \qquad j = 1\ldots, l_5,\end{aligned}$$

where l_i is determined based on size of the adjacency matrix, p (see, Figure 3).

The penalty parameters λ_e and $\{\lambda_i\}_{i=1}^n$ play an important rule for the convex decomposition to be successful. We learn them through numerical experimentation (see Figures 5 and 6) and set them respectively to

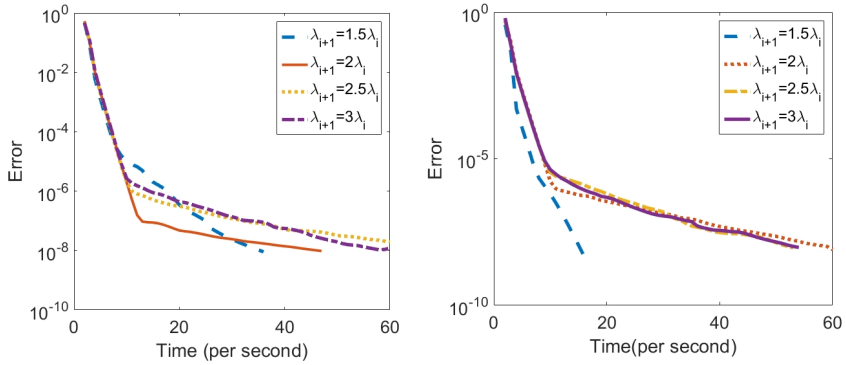
$$\varrho = 4, \quad \lambda_e = 1, \quad \lambda_1, \lambda_2 = 0.5\lambda_e, \quad \hat{\lambda}_i = 0.25\lambda_e, \quad \text{and} \quad \lambda_{i+1} = 2\lambda_i \quad \text{for} \quad i = 2, \ldots, n.$$



(a) (b)

Figure 5: Learning turning parameter λ_e for two covariance estimation problems. Comparison of the absolute errors produced by the algorithms based on CPU time for different choices of λ_e.

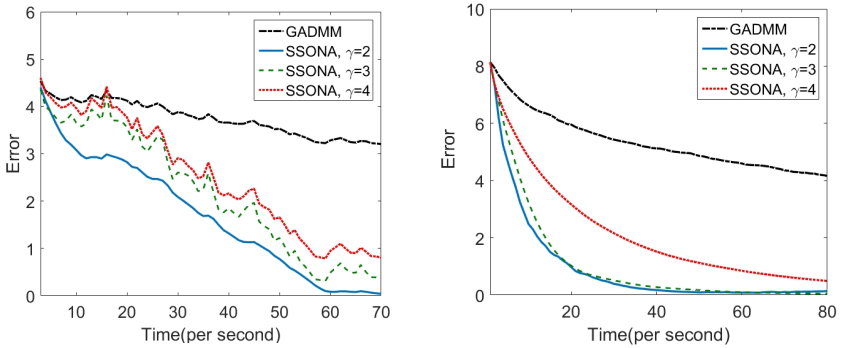
It can be seen from Figure 5 that with the addition of the ridge penalty term $\frac{\lambda_e}{2}\|E\|_F^2$ the algorithm clearly outperforms its unmodified counterpart in terms of CPU time for any fixed number of iterations. Indeed, when the model becomes more dense, SSONA is more effective to recover the network structure.



(a) (b)

Figure 6: Learning turning parameter λ_i for $i = 2, \ldots, n$ for two covariance estimation problems for different choices of λ_i for $i = 2, \ldots, n$.

Next, we conduct experiments to assess the performance of the developed multi-block ADMM algorithm (SSONA) vis-a-vis the GADMM for solving two covariance graph estimation problems of dimension 1000 in the presence of noise. Figure 7 depicts the absolute error of the objective function for different choices of the regularization parameter γ of the augmented Lagrangian and that of the dense noisy component λ_e; note that the latter is key for the convergence of the proposed algorithm.



(a) (b)

Figure 7: Comparison of the absolute errors produced by the algorithms based on CPU time for different choices of γ.

We define the following two performance measures, as proposed in Tan et al. (2014):

- Number of correctly estimated edges, n_e :

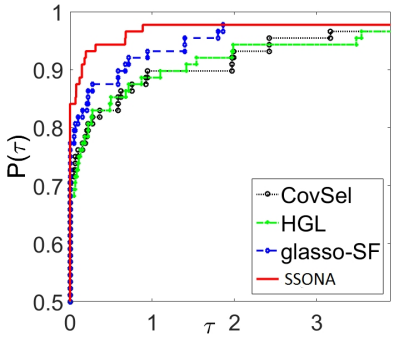
$$\sum_{j<j'} \left(1_{\{|\hat{\Theta}|>1e-4 \text{ and } |\Theta_{jj'}|\neq 0\}}\right).$$

- Sum of squared errors, s_e:

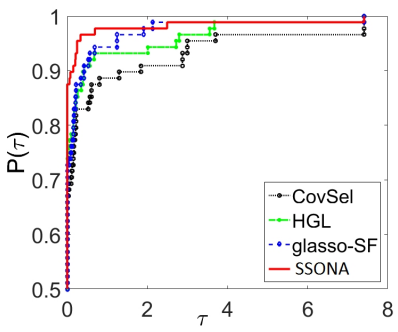
$$\sum_{j<j'} \left(|\hat{\Theta}_{jj'} - \Theta_{jj'}|\right)^2.$$

The experiment is repeated ten times and the average number of correctly estimated edges, n_e and sum of squared errors, s_e are considered for comparison. We have used the *performance profile*, as proposed in Dolan and Moré (2002), to display the efficiency of the algorithms considered, in terms of n_e and s_e. As stated in Dolan and Moré (2002), this profile provides a wealth of information such as solver efficiency, robustness and probability of success in compact form and eliminates the influence of a small number of problems on the evaluating process and the sensitivity of results associated with the ranking of solvers. Indeed, the performance profile plots the fraction of problem instances for which any given method is within a factor of the best solver. The horizontal axis of the figure gives the percentage of the test problems for which a method is efficient, while the vertical axis gives

the percentage of the test problems that were successfully solved by each method (robustness). The performance profiles of the considered algorithms in log2 scale are depicted in Figures 8,9 and 10.

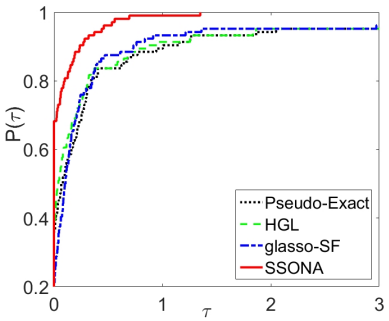


(a) Performance based on n_e.

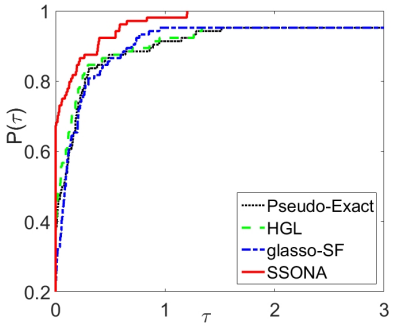


(b) Performance based on s_e.

Figure 8: Performance profiles of CovSel, HGL, glasso-SF and SSONA



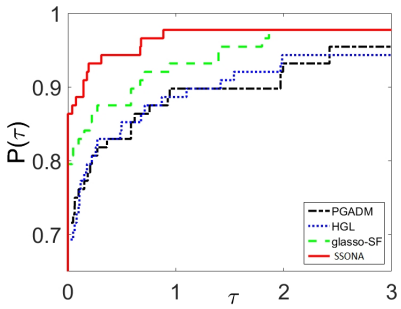
(a) Performance based on n_e.



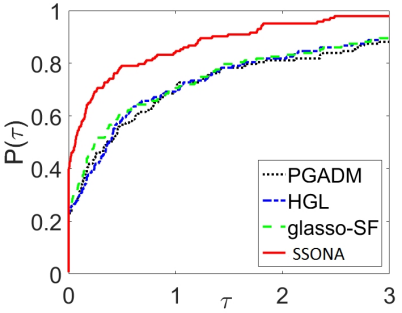
(b) Performance based on s_e.

Figure 9: Performance profiles of Pseudo-Exact, HGL, glasso-SF and SSONA.

Figures 8,9 and 10 show the performance profiles of the considered algorithms for estimation of graphical models in terms of number of correctly estimated edges and sum of squared errors, respectively. The left and right panel are drawn in terms of n_e and s_e, respectively. The results in these figures clearly demonstrate the superior performance of the proposed method, since it solves all test problems without exhibiting any failure. Moreover, the SSONA algorithm is the best algorithm among the considered ones, as it solves more than 80 % of the test problems achieving the maximum number of correctly estimated edge



(a) Performance based on n_e.



(b) Performance based on s_e.

Figure 10: Performance profiles of PGADM, HGL, glasso-SF and SSONA.

n_e and minimum value of estimation loss s_e. Further, the performance index of SSONA grows up rapidly in comparison with the other considered algorithms. The latter implies that whenever SSONA is not the best algorithm, its performance index is close to the index of the best one.

4.1.1 Experiments on structured graphical models

In this section, we present numerical results on structured graphical models to demonstrate the efficiency of SSONA. We compare the behavior of SSONA for a fixed value of $p = 100$ with a lasso version of our algorithm. Results provided in Figures 11, 12, 13 and 14 indicate the efficiency of algorithm 1 on structured graphical models. These results also show how the structure of the network returned by the two algorithms changes with growing m (note that λ_i and $\hat{\lambda}_i$ are kept fixed for each value of m). It can be easily seen from these figures (comparing Row I and II) that SSONA is less sensitive to the number of samples and shows a better approximation of the network structure even for small sample sizes.

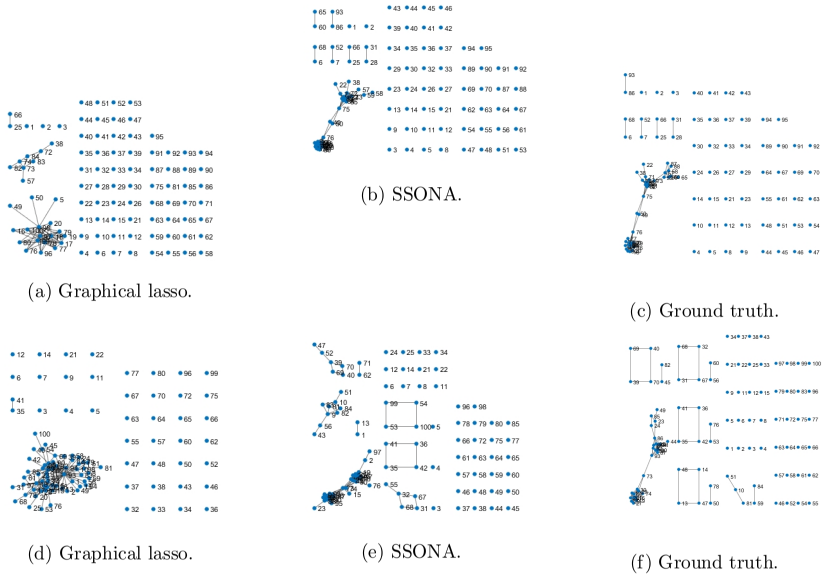
4.2 Classification and clustering accuracy based on SSONA

In this section, we evaluate the efficiency of SSONA on real data sets in recovering complex structured sparsity patterns and subsequently evaluate them on a classification task. The two data sets deal with applications in cancer genomic and document classification.

4.2.1 SSONA for Gene Selection Task

Classification with a sparsity constraint has become a standard tool in applications involving Omics data, due to the large number of available features and the small number of samples. The data set under study considers gene expression profiles of lung cancer tumors. Specifically, the data[2] consist of gene expression profiles of 12,626 genes for 197 lung tissue

2. http://www.broadinstitute.org/cgibin/cancer/publications/view/87.



(a) Graphical lasso. (b) SSONA. (c) Ground truth.

(d) Graphical lasso. (e) SSONA. (f) Ground truth.

Figure 11: Simulation for the Gaussian graphical model. Row I: Results for $p = 100$ and $m = 200$. Row II: Results for $p = 100$ and $m = 100$.

samples, with 139 adenocarcinomas(AD), 21 squamous cell carcinomas(SQ), 20 carcinoids (COID) and 17 normal lung tissue (NL). To distinguish lung adenocarcinomas from the normal lung tissues, we consider the diagnosis of lung cancer as a binary classification problem. Let the 17 normal lung comprise the positive class and the 139 lung adenocarcinomas the negative class. Following the workflow in Monti et al. (2003), we reserve the 1000 most significant genes after a preprocessing step. In the numerical experiment, we compare group lasso (Yuan and Lin, 2007), group lasso with overlap (Obozinski et al., 2011) and SSONA according to the following two criteria: average classification accuracy and gene selection performance. The experiment is repeated ten times and the average accuracy and performance are depicted in Table 1.

(a) Graphical lasso. (b) SSONA. (c) Ground truth.

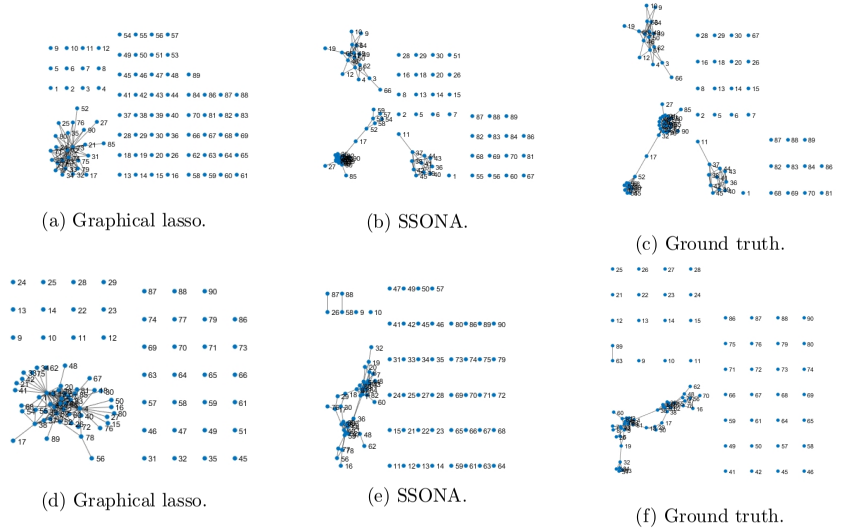
(d) Graphical lasso. (e) SSONA. (f) Ground truth.

Figure 12: Simulation for the Covariance graph model. Row I: Results for $p = 100$ and $m = 200$. Row II: Results for $p = 100$ and $m = 100$.

Method	**Average classification accuracy**	**Average number of genes selected**
Group lasso (Yuan and Lin, 2007)	0.815(0.046)	69.11(3.23)
Group lasso with overlap (Obozinski et al., 2011)	0.834(0.035)	57.30(2.71)
SSONA (4 structured matrices)	0.807(0.028)	61.44(2.80)
SSONA (6 structured matrices)	0.839(0.022)	56.111(2.100)

Table 1: Experimental results on lung cancer data over 10 replications (the standard deviations are reported in parentheses).

As is shown in Table 1, SSONA achieves higher classification accuracy than the group lasso and lower classification accuracy than the latent group lasso, although the performance of all three methods is very similar and within the variability induced by the replicates. However, our SSON based lasso does not require a priori knowledge of group structures, which is a prerequisite for the other two methods. One can easily improve the the classification accuracy and gene selection performance of SSONA by adding more structured matrices.



(a) Graphical lasso. (b) SSONA. (c) Ground truth.

(d) Graphical lasso. (e) SSONA. (f) Ground truth.

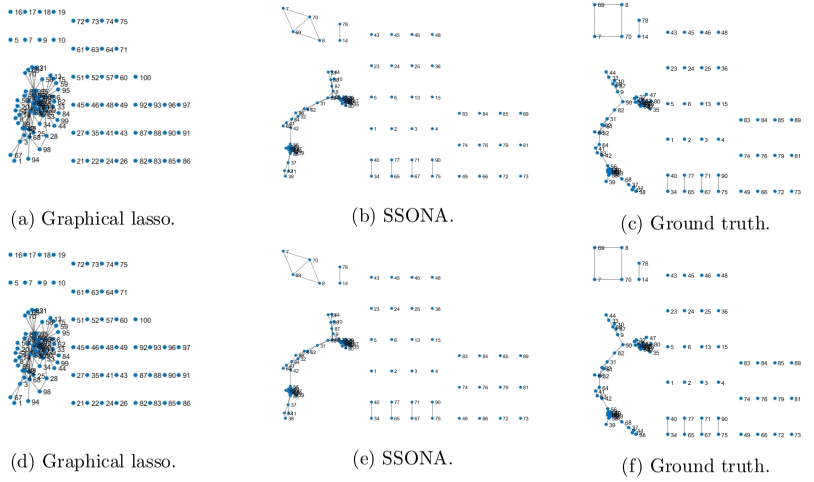
Figure 13: Simulation for the Gaussian graphical model with 10 latent variables. Row I: Results for $p = 100$ and $m = 200$. Row II: Results for $p = 100$ and $m = 100$.

In our experiments, SSONA selects the least number of genes and achieves the smallest standard deviation of average number of genes *without any priori knowledge*. Due to the different number of randomly selected genes, the average number of gene sometimes will be a non-integer.

4.2.2 SSONA for Document Classification Task

The next example involves a data set [3] containing 1427 documents with a corpus of size 17785 words. We randomly partition the data into 999 training, 214 validation and 214 test examples, corresponding to a 70/15/15 split (Rao et al., 2016). We first train a Latent Dirichlet Allocation based topics model (Blei et al., 2003) to assign the words to 100 "topics". These correspond to our groups, and since a single word can be assigned to multiple topics, the groups overlap. We then train a lasso logistic model using as outcome variable indicating whether the document discusses atheism or not , together with an overlapping group lasso and a SSON based lasso model where the tuning parameters are selected based on cross validation. Table 2 shows that the variants of the SSON yield almost the same misclassification rate compared to the other two methods, while it does not require a priori knowledge of group structures.

3. `http://qwone.com/jason/ 20Newsgroups/`



(a) Graphical lasso. (b) SSONA. (c) Ground truth.

(d) Graphical lasso. (e) SSONA. (f) Ground truth.

Figure 14: Simulation for the binary Ising Markov random field. Row I: Results for $p = 100$ and $m = 200$. Row II: Results for $p = 100$ and $m = 100$.

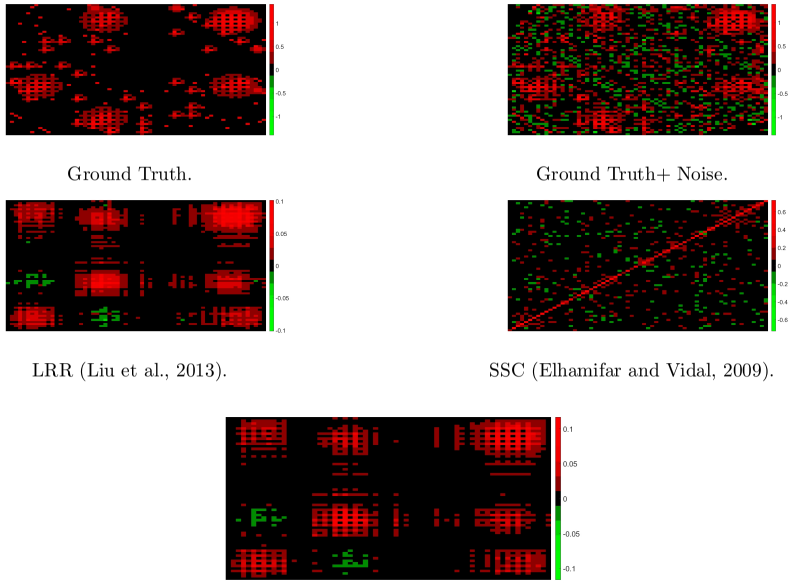
Method	**Misclassification Rate**
Group lasso (Yuan and Lin, 2007)	0.445
Group lasso with overlap (Obozinski et al., 2011)	0.390
SSONA (5 structured matrices)	0.435
SSONA (6 structured matrices)	0.421
SSONA (7 structured matrices)	0.401

Table 2: Misclassification rate on the test set for document classification.

4.2.3 SSONA for structured subspace clustering

Our last example focuses on data clustering. The data come from multiple low-dimensional linear or affine subspaces embedded in a high-dimensional space. Our method is based on (11), wherein each point in a union of subspaces has a representation with respect to a dictionary formed by all other data points. In general, finding such a representation is NP hard. We apply our subspace clustering algorithm to a structured data in the presence of noise. The segmentation of the data is obtained by applying SSONA to the adjacency matrix built from the data. Our method can handle noise and missing data and is effective to detect the clusters.

Figure 15 shows that our approach significantly outperforms state-of-the-art methods.



Ground Truth.

Ground Truth+ Noise.

LRR (Liu et al., 2013).

SSC (Elhamifar and Vidal, 2009).

SSONA.

Figure 15: Heatmap of different algorithms for detecting clusters in data.

4.3 Application to real data sets

Next, we use the SSON framework to analyze three data sets from molecular and social science domains. Although there is no known ground truth, the proposed framework recovers interesting patterns and highly interpretable structures.

Analysis of connectivity in the financial sector. We applied the SSON methodology to analyze connectivity in the financial sector. We use monthly stock returns data from August, 2001 to July, 2016 for three financial sectors, namely banks (BA), primary broker/dealers (PB), and insurance companies (INS). The data are obtained from the University of Chicago's Center for Research in Security Prices database (CRSP).

Our final sample covers 75 different institutions spanning a 16-year period. Figure 16 shows the mean (in %) of monthly stock returns across different sectors in each 3-year long rolling windows. As expected, the average returns are significantly lower during the financial 2007-2009 crisis period, compared to any other period in our sample. Indeed, looking across the sectors, all three sectors experienced diminished performance during the 2007-2009 crisis. Further, the almost linear ramp-up following 2009 clearly captures the recovery of financial stocks and the broader market.

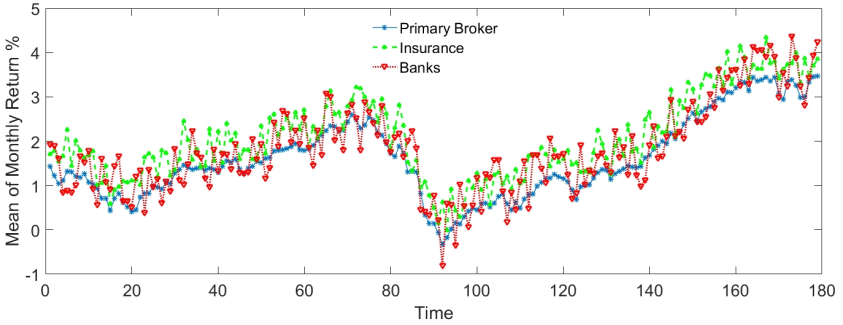
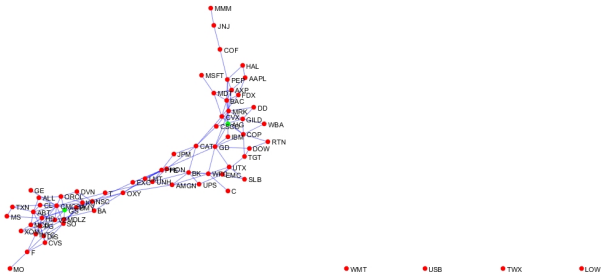


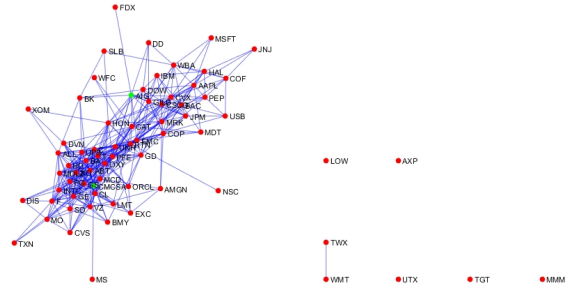
Figure 16: Average monthly return of firms in the three sectors- Bank, primary broker-dealer and insurance firms, in different 3-year rolling windows during 180 months. The figure shows diminished performance during the 2007-2009 crisis (time step : 80-100) and also clearly captures the strong recovery of stock performance starting in 2009.

Next, we estimate a measure of network connectivity for a sample of the 71 components of the SP100 index that were present during the entire 2001-16 period under consideration. Figure 17 depicts the network estimates of the transition (lead-lag) matrices using straight lasso VAR and SSONA based VAR for the January 2007 to Oct 2009 period. It can be seen that the lasso VAR estimates produce a more highly connected network, while the SSONA ones identify two more connected components. Both methods highlight the key role played by AIG and GS (Goldman Sachs), but the SSONA based network indicates that one dense connected component is centered around the former, while the other dense connected component around the latter. In summary, both methods capture the main connectivity patterns during the crisis period, but SSONA provides a more nuanced picture.

US House voting data set. We applied SSONA to describe the relationships amongst House Representatives in the U.S. Congress during the 2005-2006 period (109th Congress). The variables correspond to the 435 representatives, and the observations to the 1210 votes that the House deliberated and voted on during that period, which include bills, resolutions, motions, debates and roll call votes. The assumption of our model is that bills are i.i.d. sample from the same underlying Ising model. The votes are recorded as "yes" (encoded as "1") and "no" (encoded as "0"). Missing observations were replaced with the majority vote of the House members party on that particular vote. Following Guo et al. (2015), we used a bootstrap procedure with the proposed SSONA estimator to evaluate the confidence of the estimated edges. Specifically, we estimated the network for multiple bootstrap samples of the same size, and only retained the edges that appeared more that ω percent of the time. The goal of the analysis is to understand the type of relationships that existed among the House members in the 109th Congress. In particular, we wish to identify and interpret the presence of densely connected components, as well of sparse components. The heatmap of



(a) SSONA VAR



(b) Lasso VAR

Figure 17: Networks estimate by SSONA and Lasso VAR during crisis period of Jan 2007 to Oct 2009.

the adjacency matrix of the estimated network by using SSONA is depicted in Figure 18. It can be easily seen that there exist densely connected components in the network, a fact that the glasso algorithm (Friedman et al., 2008) fails to recover (see, Figure 19).

The network representation of subgraphs, with a cut-off value of 0.6, is given in Figures 20, 21 and 22. We only plot the edges associated with the subgraphs to enhance the visual reading of densely correlated areas. An interesting result of applying SSONA on this data set is the clear separation between members of the Democratic and Republican parties, as expected (see, Figures 20, 21 and 22). Moreover, voting relationships within the two parties exhibit a clustering structure, which a closer inspection of the votes and subsequent analysis showed was mainly driven by the position of the House member on the ideological/political spectrum.

Other interesting patterns emerging from the analysis is that SSONA recovers members of opposite parties as a sparse component in each subgraph (see, Figures 20, 21 and 22). For instance, Figure 21 shows that Republican members such as Simpson, Kirk and Hyde are sparsely connected in a clustered group of Democratic members. This is possibly due to

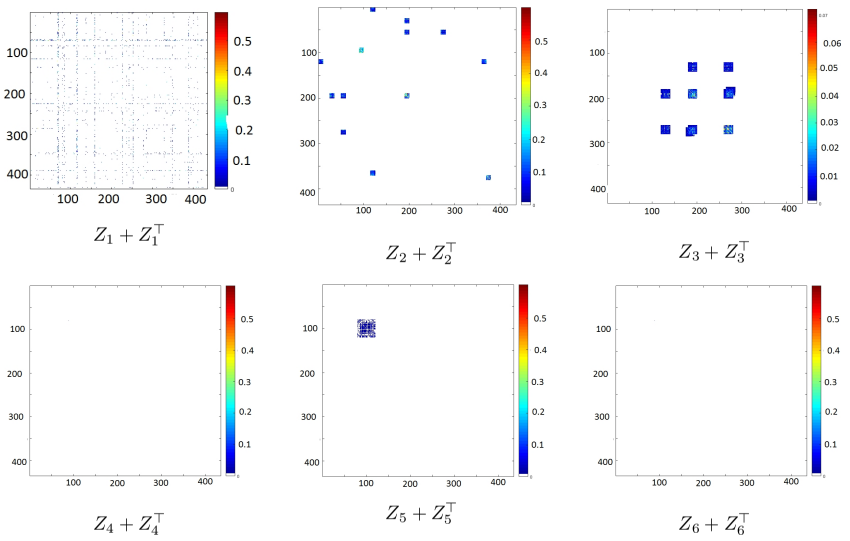


Figure 18: Heatmap of the structured precision matrix Θ decomposed into $Z_1 + Z_1^\top + \cdots + Z_6 + Z_6^\top$ in the House voting data, estimated by SSONA.

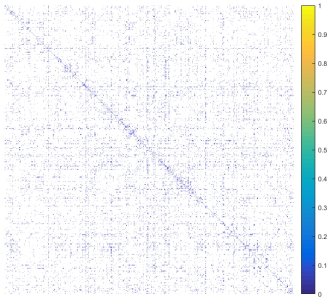


Figure 19: Heatmap of the inverse covariance matrix in the voting record of the U.S. House of Representatives, estimated by the graphical lasso method (Friedman et al., 2008).

the overall centrist record of Kirk and alignment of Hyde and Simpson on selected issues. Similarly, Figure 21 indicates that Democratic members Bishop, Hastings and Meek are approximately sparsely connected to a subgraph of Republican members. Bishop from Georgia has compiled a fairly conservative voting record. The same conclusion can be derived from Figure 21. Indeed, Figures 20, 21 and 22 reveals that there are strong positive

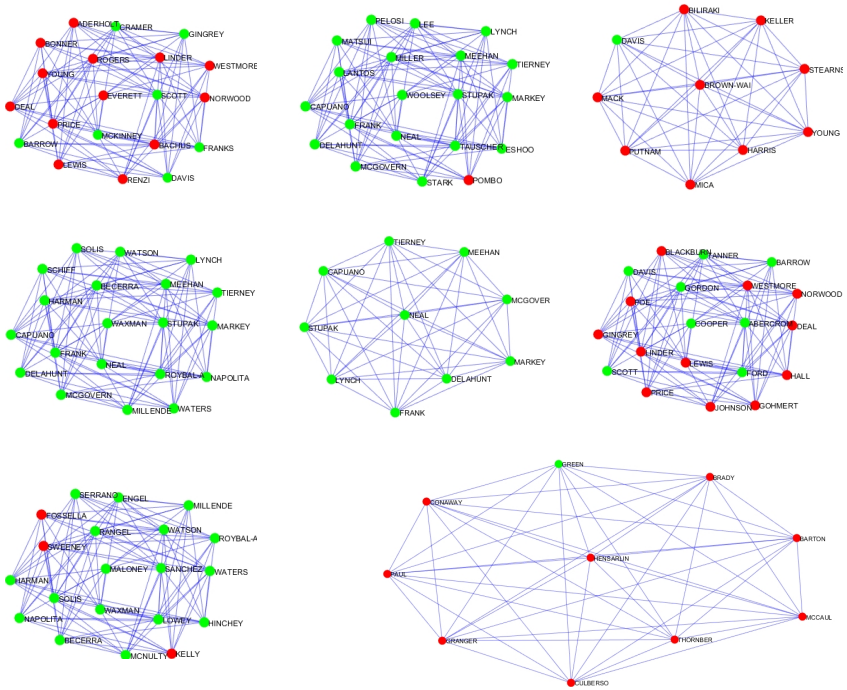


Figure 20: Dense subgraphs identified by SSONA for the House voting data with an inclusion cutoff value of 0.6. Subfigures correspond to a densely connected area in Figure 18 for the symmetric structured matrix $Z_2 + Z_2^\top$. The nodes represent House members, with red and green colored nodes corresponding to Republicans and Democrats, respectively. A blue line corresponds to an edge between two nodes.

associations between members of the same party and negative associations between members of opposite parties. Obviously, at the higher cutoff value the dependence structure between members of opposite parties becomes sparser.

Other patterns of interest include a strong dependence between members of two opposite parties in selected subgraphs when the members come from the same state, as is the case for New York state members Jerrold Nadler (D), Anthony D. Weiner (D), Ed Towns (D), Major Owens (D), Nydia Velzquez (D), Vito Fossella (R), Carolyn B. Maloney (D), Charles B. Rangel (D), Jos Serrano (D), Eliot L. Engel (D), Nita Lowey (D), Sue W. Kelly (R), John E. Sweeney (R), Michael R. McNulty (D), Maurice Hinchey (D), John M. McHugh (R), Sherwood Boehlert (R), Jim Walsh (R), Tom Reynolds (R), Brian Higgins (D) -see

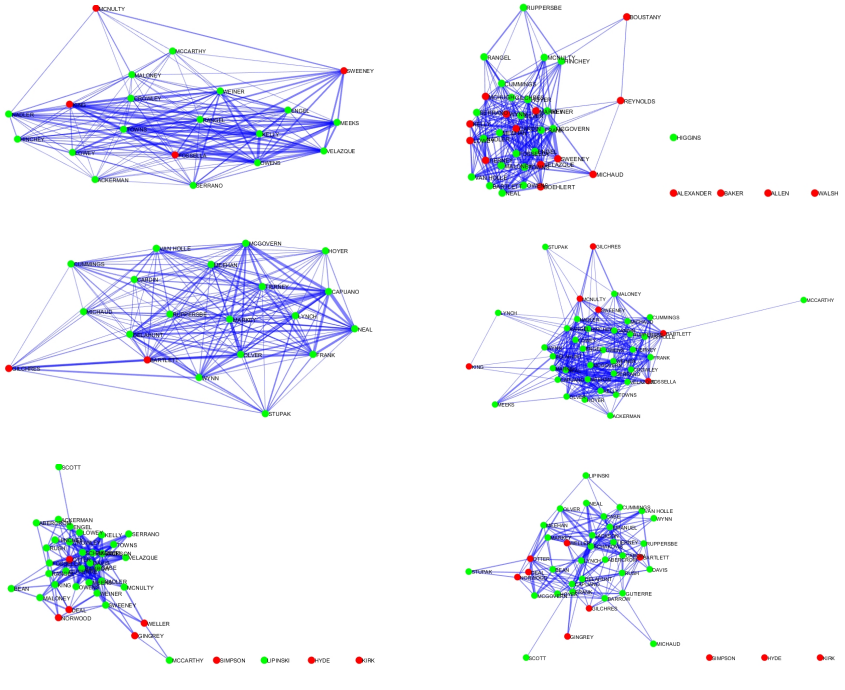


Figure 21: Dense subgraphs identified by SSONA for the House voting data with an inclusion cutoff value of 0.6. Subfigures correspond to a densely connected area in Figure 18 for the symmetric structured matrix $Z_3 + Z_3^\top$. The nodes represent House members, with red and green colored nodes corresponding to Republicans and Democrats, respectively. A blue line corresponds to an edge between two nodes.

Figure 21. However, in this instance, there is also a cluster of positive associations between Democrats.

In summary, SSONA provides deeper insights into relationships between House members, going beyond the obvious separation into two parties, according to their voting record.

Analysis of a breast cancer data set. We applied SSONA to a data set containing 800 gene expression measurements from large epithelial cells obtained from 255 patients with breast cancer. The goal is to capture regulatory interactions amongst the genes, as well as to identify genes that tend to have interactions with other genes in a group and hence act as master regulators, thus providing insights into the molecular circuitry of the disease. Figure 23 depicts the heat map of the estimated adjacency matrix for the breast cancer

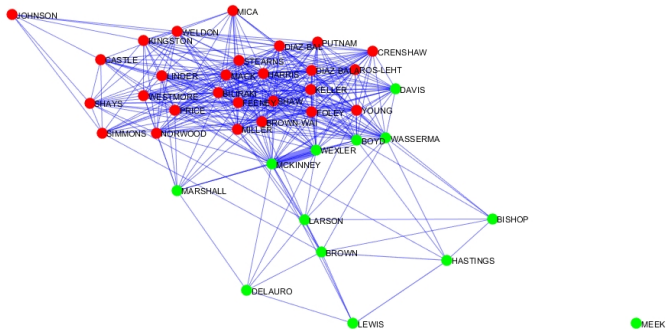


Figure 22: Dense subgraph identified by SSONA for the House voting data with an inclusion cutoff value of 0.6. Subfigures corresponds to a densely connected area in Figure 18 for the symmetric structured matrix $Z_5 + Z_5^\top$. The nodes represent House members, with red and blue node colors corresponding to Republicans and Democrats, respectively. A blue line corresponds to an edge between two nodes.

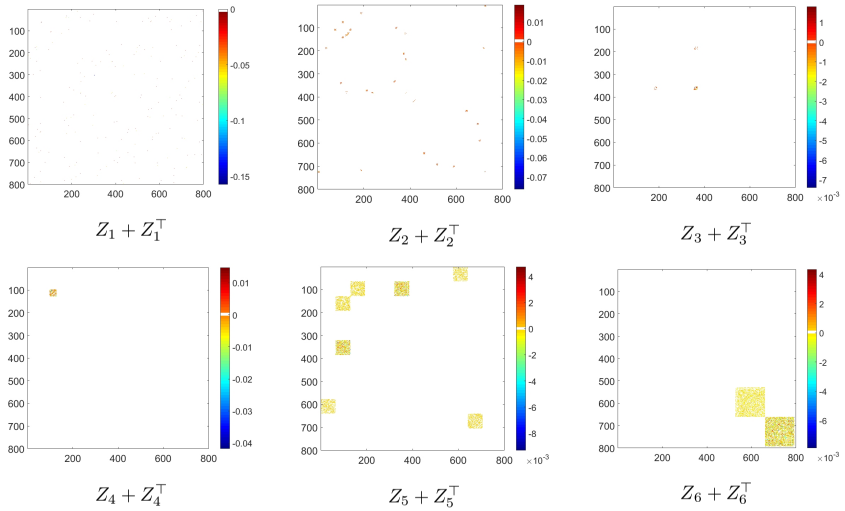


Figure 23: Heat map of the structured precision matrix Θ decomposed into $Z_1 + Z_1^\top + \cdots + Z_6 + Z_6^\top$ in the breast cancer data set, estimated by SSONA.

data set. As it is clear in Figure 23, $Z_2 + Z_2^\top, \ldots, Z_5 + Z_5^\top$ and $Z_6 + Z_6^\top$ show that selected

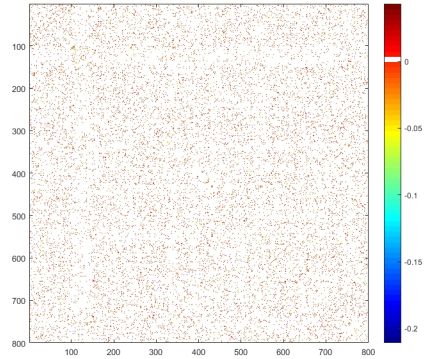


Figure 24: Heatmap of the inverse covariance matrix in the breast cancer data sets, estimated from graphical lasso (Friedman et al., 2008).

genes are densely connected, which is not the case when employing the the graphical lasso algorithm (see, Figure 24). Therefore, SSONA can provide an intuitive explanation of the relationships among the genes in the breast cancer data set (see, Figure 25 and 26 for two examples). These genes connectivity in the tumor samples may indicate a relationship that is common to an important subset of cancers. Many other genes belong to this network, each indicating a potentially interesting interaction in cancer biology. We omit the full list of densely connected genes in our estimated network and provide a complete list in the on-line supplementary materials available in the first author's homepage.

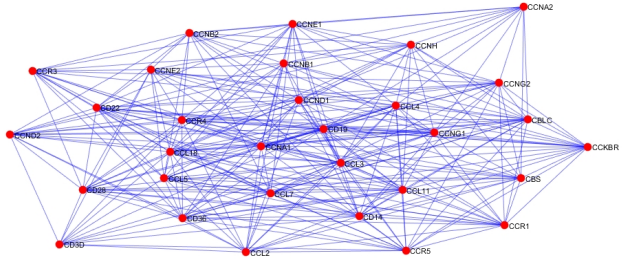


Figure 25: Network layout of grouped genes identified by SSONA for the breast cancer data set. Subfigure corresponds to a densely connected component in Figure 23 for the structured matrix $Z_4 + Z_4^\top$.

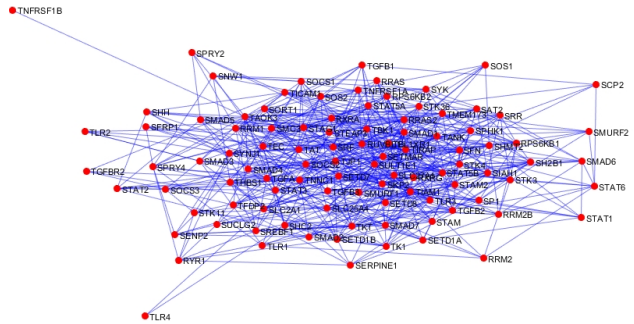


Figure 26: Network layout of grouped genes identified by SSONA for the breast cancer data set. Subfigure corresponds to a densely connected component in Figure 23 for the structured matrix $Z_6 + Z_6^\top$.

5. Conclusion

In this paper, a new structured norm minimization method for solving multi-structure graphical model selection problems is proposed. Using the proposed SSON, we can efficiently and accurately recover the underlying network structure. Our method utilizes a class of sparse structured norms in order to achieve higher order accuracy in approximating the decomposition of the parameter matrix in Markov Random Field and Gaussian Covariance Graph models. We also provide a brief discussion of its application to regression and classification problems. Further, we introduce a linearized multi-block ADMM algorithm to solve the resulting optimization problem. The global convergence of the algorithm is established without any upper bound on the penalty parameter. We applied the proposed methodology to a number of real and synthetic data sets that establish its overall usefulness and superior performance to competing methods in the literature.

Acknowledgments

The authors would like to thank the Editor and three anonymous referees for many constructive comments and suggestions that improved significantly the structure and readability of the paper. This work was supported in part by NSF grants DMS-1545277, DMS-1632730, NIH grant 1R01-GM1140201A1 and by the UF Informatics Institute.

Appendix A. Update for Θ

In each iteration of Algorithm 1 the update for Θ depends on the form of the loss function $g(\Theta)$. We consider the following cases to update Θ:

1. The update for Θ_1 in Algorithm 1 (step 2(a)) can be obtained by minimizing

$$\text{trace}(\hat{\Sigma}\Theta_1) - \log\det\Theta_1 + \frac{\gamma}{2}\|\Theta_1 - (\sum_{i=1}^{n} Z_i^k + {Z_i^k}^\top + E^k + \frac{1}{\gamma}\Lambda^k)\|_F^2,$$

with respect to Θ_1 (note that the constraint $\Theta_1 \in \mathcal{S}$ in (6) is treated as an implicit constraint, due to the domain of definition of the log det function). This can be shown to have the solution

$$\Theta_1 = \frac{1}{2}U\Big(D + \sqrt{D^2 + \frac{4}{\gamma}I}\Big)U^T,$$

where UDU^T stands for the eigen-decomposition of $\sum_{i=1}^n Z_i^k + {Z_i^k}^\top + E^k + \frac{1}{\gamma}\Lambda^k - \frac{1}{\gamma}\hat{\Sigma}$.

2. Update for Θ_2 in Step 2(a) of Algorithm 1 leads to the following optimization problem

$$\begin{aligned}\underset{\Theta_3\in\mathcal{S}}{\text{minimize}}\ \Phi(\Theta_2) \quad &= \quad \sum_{j=1}^{p}\sum_{j'=1}^{p}\theta_{jj'}(X^TX)_{jj'} - \sum_{i=1}^{m}\sum_{j=1}^{p}\log\Big(1+\exp[\theta_{jj}+\sum_{j'\neq j}\theta_{jj'}x_{ij'}]\Big)\\ &+ \quad \frac{\gamma}{2}\|\Theta_2 - (\sum_{i=1}^{n} Z_i^k + {Z_i^k}^\top + E^k + \frac{1}{\gamma}\Lambda^k)\|_F^2. \qquad (33)\end{aligned}$$

We use a novel non-monotone version of the Barzilai-Borwein method (Barzilai and Borwein, 1988; Raydan, 1997; Fletcher, 2005; Ataee Tarzanagh et al., 2014) to solve (33). The details are given in Algorithm 2.

Algorithm 2 Non-monotone Barzilai Borwein Method for solving (33)

Initialize The parameters:

(a) $\Theta^0 = I$, $\Theta^1 = 2\Theta^0$, $\alpha^1 = 1$ and $t^0 = 10$.

(b) A positive sequence $\{\eta^t\}$ satisfying $\sum_{k=1}^{\infty}\eta^t = \eta < \infty$.

(c) Constants $\sigma > 0$, $\epsilon > 0$, and $\nu \in (0,1)$.

Iterate Until the stopping criterion $\dfrac{\|\Theta^t - \Theta^{t-1}\|_F^2}{\|\Theta^{t-1}\|_F^2} \leq \epsilon$ is met:

1. $\mathbb{G}^t = -\alpha^t\nabla\Phi(\Theta^t)$.
2. Set $\rho = 1$.
3. **If** $t > t^0$, **then**
   While $\|\Phi(\Theta^t + \rho^t\mathbb{G}^t)\|_F \leq \Phi(\Theta^t) + \eta^t - \sigma\rho^2{\alpha^t}^2\|\mathbb{G}^t\|_F^2$, **do**
   Set $\rho = \nu\rho$;
   EndWhile
   EndIf
4. Define $\rho^t = \rho$ and $\Theta^{t+1} = \Theta^t + \rho^t\mathbb{G}^t$.
5. Define $\alpha^{t+1} = \dfrac{\text{trace}\Big((\Theta^t - \Theta^{t+1})^T(\Theta^t - \Theta^{t+1})\Big)}{\text{trace}\Big((\nabla\Phi(\Theta^t) - \nabla\Phi(\Theta^{t+1}))^T(\Theta^t - \Theta^{t+1})\Big)}$

3. To update Θ_3 in step 2(a), using (11), we have that

$$\begin{aligned}\underset{\Theta_3}{\text{minimize}}\ \frac{1}{2}\|\Theta_3-\hat{\Sigma}\|_F^2 \quad &+ \quad \frac{\gamma}{2}\|\Theta_3-\Big(\sum_{i=1}^{n}Z_i^k+Z_i^{k^\top}+E^k+\frac{1}{\gamma}\Lambda^k\Big)\|_F^2\\ &= \quad \Big(\frac{1}{1+\gamma}(\hat{\Sigma}+\gamma(\sum_{i=1}^{n}Z_i^k+Z_i^{k^\top}+E^k)+\Lambda^k)\Big)_+\end{aligned}$$

where $V_+ = U_\dagger D_+ U_\dagger$ such that

$$UDU=\begin{pmatrix}U_\dagger & U_\ddagger\end{pmatrix}\begin{pmatrix}D_+ & 0\\ 0 & D_-\end{pmatrix}\begin{pmatrix}U_\dagger\\ U_\ddagger\end{pmatrix},$$

is the eigen-decomposition of the matrix V , and D_+ and D_- are the nonnegative and negative eigenvalues of V.

Appendix B. Convergence Analysis

Before establishing the main result on global convergence of the proposed ADMM algorithm, we provide the necessary definitions used in the proofs (for more details see Bolte et al. (2014)):

Definition 11 *(Kurdyka- Lojasiewicz property).*
The function f is said to have the Kurdyka- Lojasiewicz (K-L) property at point Z_0, if there exist $c_1>0$, $c_2>0$ and $\phi\in\Gamma_{c_2}$ such that for all

$$Z\in B(Z_0,c_1)\cap\{Z: f(Z_0)<f(Z)<f(Z_0)+c_2\},$$

the following inequality holds

$$\phi'\big(f(Z)-f(Z_0)\big)dist\big(0,\partial f(Z)\big)\geq 1,$$

where Γ_{c_2} stands for the class of functions $\phi:[0,c_2)\to\mathbb{R}^+$ with the properties:

(i) ϕ is continuous on $[0,c_2)$;

(ii) ϕ is smooth concave on $(0,c_2)$;

(iii) $\phi(0)=0$, $\nabla\phi(s)>0$, $\forall\ s\in(0,c_2)$.

Definition 12 *(Semi-algebraic sets and functions).*

(i) A subset $C\in\mathbb{R}^{n\times n}$ is semi-algebraic, if there exists a finite number of real polynomial functions h_{ij}, $s_{ij}:\mathbb{R}^{n\times n}\to\mathbb{R}$ such that

$$C=\cup_{i=1}^{\bar{p}}\cap_{j=1}^{\bar{q}}\{Z\in\mathbb{R}^{n\times n}:\ g_{ij}(Z)=0\quad and\quad s_{ij}(Z)<0\}.$$

(ii) A function $h:\mathbb{R}^{n\times n}\to(-\infty,+\infty]$ is called semi-algebraic, if its graph

$$\mathbb{G}(h):=\{(Z,y)\in\mathbb{R}^{n\times n+1}: h(Z)=y\},$$

is a semi-algebraic set in $R^{n\times n+1}$.

Definition 13 *(Sub-analytic sets and functions).*

(i) A subset $C\in\mathbb{R}^{n\times n}$ is sub-analytic, if there exists a finite number of real analytic functions h_{ij}, $s_{ij}:\mathbb{R}^{n\times n}\to\mathbb{R}$ such that

$$C=\cup_{i=1}^{\bar{p}}\cap_{j=1}^{\bar{q}}\{Z\in\mathbb{R}^{d}: g_{ij}(Z)=0\quad and\quad s_{ij}(Z)<0\}.$$

(ii) A function h: $R^{n\times n}\to(-\infty,+\infty]$ is called sub-analytic, if its graph

$$\mathbb{G}(h):=\{(Z,y)\in\mathbb{R}^{n\times n+1}: h(Z)=y\}$$

is a sub-analytic set in $R^{n\times n+1}$.

It can be easily seen that both real analytic and semi-algebraic functions are sub-analytic. In general, the sum of two sub-analytic functions is not necessarily sub-analytic. However, it is easy to show that for two sub-analytic functions, if at least one function maps bounded sets to bounded sets, then their sum is also sub-analytic (Bolte et al., 2014).

Remark 14 *Each f_i in (19) is a convex semi-algebraic function (see, example 5.3 in (Bolte et al., 2014)), while the loss function $\mathcal{G}$ in (6), (9), (11), (13), and (18) is sub-analytic (even analytic). Since each function f_i maps bounded sets to bounded sets, we can conclude that the augmented Lagrangian function*

$$\begin{aligned}\mathcal{L}_\gamma(\Theta,Z_1,\ldots,Z_n,E;\Lambda) \quad &= \quad \mathcal{G}(X,\Theta)+f_1(Z_1)+\cdots+f_n(Z_n)+f_e(E)\\ &- \quad \langle\Lambda,\Theta-\sum_{i=1}^{n}Z_i+Z_i^\top-E\rangle\\ &+ \quad \frac{\gamma}{2}\|\Theta-\sum_{i=1}^{n}Z_i+Z_i^\top-E\|_F^2,\end{aligned}$$

which is the summation of sub-analytic functions is itself sub-analytic. All sub-analytic functions which are continuous over their domain satisfy a K-L inequality, as well as some, but not all, convex functions (see Bolte et al., 2014 for details and a counterexample). Therefore, the augmented Lagrangian function $\mathcal{L}_\gamma$ satisfies the K-L property.

Next, we establish a series of lemmas used in the proof of Theorem 10.

Lemma 15 *Let $U^k:=(\Theta^k,Z_1^k,\ldots,Z_n^k,E^k;\Lambda^k)$ be a sequence generated by Algorithm 1, then there exists a positive constant ϑ such that*

$$\begin{aligned}\mathcal{L}_\gamma(U^{k+1}) \quad &\leq \quad \mathcal{L}_\gamma(U^k)-\frac{\vartheta}{2}\Big(\|\Theta^k-\Theta^{k+1}\|_F\\ &+ \quad \sum_{i=1}^{n}\|Z_i^k-Z_i^{k+1}\|_F+\|E^k-E^{k+1}\|_F+\|\Lambda^k-\Lambda^{k+1}\|_F\Big).\end{aligned}\tag{34}$$

Proof. Using the first-order optimality conditions for (21) and the convexity of $\mathcal{G}(X,\Theta)$, we obtain

$$\begin{aligned}
0 &= \langle \Theta^k - \Theta^{k+1}, \nabla\mathcal{G}(X,\Theta^{k+1}) - \Lambda^k + \gamma(\Theta^{k+1} - \sum_{i=1}^n Z_i^k + Z_i^{k\top} - E^k)\rangle \\
&\leq \mathcal{G}(X,\Theta^k) - \mathcal{G}(X,\Theta^{k+1}) - \langle \Theta^k - \Theta^{k+1}, \Lambda^k\rangle \\
&+ \gamma\langle \Theta^k - \Theta^{k+1}, \Theta^{k+1} - \sum_{i=1}^n Z_i^k + Z_i^{k\top} - E^k\rangle \\
&= \mathcal{G}(X,\Theta^k) - \langle \Theta^k, \Lambda^k\rangle + \frac{\gamma}{2}\sum_{i=1}^n \|\Theta^k - \sum_{i=1}^n Z_i^k + Z_i^{k\top} - E^k\|_F^2 - \frac{\gamma}{2}\|\Theta^k - \Theta^{k+1}\|_F^2 \\
&- \Big(\mathcal{G}(X,\Theta^{k+1}) - \langle \Theta^{k+1}, \Lambda^k\rangle + \frac{\gamma}{2}\|\Theta^{k+1} - \sum_{i=1}^n Z_i^k + Z_i^{k\top} - E^k\|_F^2\Big) \\
&= \mathcal{L}_\gamma(U^k) - \mathcal{L}_\gamma(\Theta^{k+1}, Z_1^k, \ldots, Z_n^k, E^k; \Lambda^k) - \frac{\gamma}{2}\|\Theta^k - \Theta^{k+1}\|_F^2, \qquad (35)
\end{aligned}$$

where the second equality follows from the fact that

$$(u_1-u_2)^T(u_3-u_1) = \frac{1}{2}\Big(\|u_2-u_3\|_F^2 - \|u_1-u_2\|_F^2 - \|u_1-u_3\|_F^2\Big).$$

Using (22), (23) and Lemma 7, we have that

$$\begin{aligned}
\mathcal{L}_\gamma(\Theta^{k+1}, Z_1^k, Z_2^k, \ldots, E^k; \Lambda^k) &- \mathcal{L}_\gamma(\Theta^{k+1}, Z_1^{k+1}, Z_2^k, \ldots, E^k; \Lambda^k) \\
&- \frac{(\gamma\varrho - L_{H_1})}{2}\|Z_1^k - Z_1^{k+1}\|_F^2 \\
&\geq 0, \\
\mathcal{L}_\gamma(\Theta^{k+1}, \ldots, Z_{i-1}^{k+1}, Z_i^k, \ldots, E^k; \Lambda^k) &- \mathcal{L}_\gamma(\Theta^{k+1}, \ldots, Z_i^{k+1}, Z_{i+1}^k, \ldots, E^k; \Lambda^k) \\
&- \frac{(\gamma\varrho - L_{H_i})}{2}\|Z_i^k - Z_i^{k+1}\|_F^2 \\
&\geq 0, \qquad i = 2, \ldots, n, \qquad (36)
\end{aligned}$$

where L_{H_i} is a Lipschitz constant of the gradient $\nabla H_i(Z_i)$, and $\varrho \geq \frac{L_{H_i}}{\gamma}$, $(i = 1, \ldots, n)$ is a proximal parameter.

Following the same steps as (35), we have that

$$\begin{aligned}
\mathcal{L}_\gamma(\Theta^k, Z_1^{k+1}, \ldots, Z_n^{k+1}, E^k; \Lambda^k) &- \mathcal{L}_\gamma(\Theta^{k+1}, Z_1^{k+1}, \ldots, Z_n^{k+1}, E^{k+1}; \Lambda^k) \\
&- \frac{\gamma}{2}\|E^k - E^{k+1}\|_F^2 \\
&\geq 0, \qquad (37)
\end{aligned}$$

and

$$\begin{aligned}
\mathcal{L}_\gamma(\Theta^{k+1}, Z_1^{k+1}, \ldots, Z_n^{k+1}, E^{k+1}; \Lambda^k) &- \mathcal{L}_\gamma(\Theta^{k+1}, Z_1^{k+1}, \ldots, Z_n^{k+1}, E^{k+1}; \Lambda^{k+1}) \\
&- \frac{\lambda_e^2}{\gamma}\|E^k - E^{k+1}\|_F^2 \\
&\geq 0. \qquad (38)
\end{aligned}$$

Let

$$\hat{\gamma} := \max(\gamma\varrho - L_{H_1}, \ldots, \gamma\varrho - L_{H_n}), \quad \bar{\gamma} := \frac{\gamma^2 - 2\lambda_e^2}{\gamma(1+\lambda_e^2)}, \quad \vartheta := \max(\hat{\gamma}, \bar{\gamma}, \gamma).$$

Then, using (35)– (38), and $\gamma \geq \sqrt{2}\lambda_e$, we have

$$\begin{aligned}
&\mathcal{L}_\gamma(U^k) - \mathcal{L}_\gamma(U^{k+1}) \geq \frac{\gamma}{2}\|\Theta^k - \Theta^{k+1}\|_F^2 \\
&+ \frac{\hat{\gamma}}{2}\sum_{i=1}^n \|Z_i^k - Z_i^{k+1}\|_F^2 + \frac{\gamma^2 - 2\lambda_e^2}{2\gamma}\|E^k - E^{k+1}\|_F^2, \\
&= \frac{\gamma}{2}\|\Theta^k - \Theta^{k+1}\|_F^2 + \frac{\hat{\gamma}}{2}\sum_{i=1}^n \|Z_i^k - Z_i^{k+1}\|_F^2 + \frac{\bar{\gamma}}{2}\|E^k - E^{k+1}\|_F^2 + \frac{\lambda_e^2\bar{\gamma}}{2}\|E^k - E^{k+1}\|_F^2, \\
&= \frac{\gamma}{2}\|\Theta^k - \Theta^{k+1}\|_F^2 + \frac{\hat{\gamma}}{2}\sum_{i=1}^n \|Z_i^k - Z_i^{k+1}\|_F^2 + \frac{\bar{\gamma}}{2}\Big(\|E^k - E^{k+1}\|_F^2 + \|\Lambda^k - \Lambda^{k+1}\|_F^2\Big), \\
&\geq \frac{\vartheta}{2}\Big(\|\Theta^k - \Theta^{k+1}\|_F^2 + \sum_{i=1}^n \|Z_i^k - Z_i^{k+1}\|_F^2 + \|E^k - E^{k+1}\|_F^2 + \|\Lambda^k - \Lambda^{k+1}\|_F^2\Big).
\end{aligned}$$

□

Lemma 16 *Let $U^k = (\Theta^k, Z_1^k, \ldots X_n^k, E^k, \Lambda^k)$ be a sequence generated by Algorithm 1. Then, there exists a subsequence U^{k_s} of $\{U^k\}$, such that*

$$\lim_{s\to\infty} \mathcal{G}(X,\Theta^{k_s}) = g(\Theta^*), \quad \lim_{s\to\infty} f_i(Z_i^{k_s}) = f_i(Z_i^*), \quad \lim_{s\to\infty} f_e(E_i^{k_s}) = f_e(E_i^*),$$

where

$$\lim_{s\to\infty} U^{k_s} = (\Theta^*, Z_1^*, \ldots, Z_n^*, E^*, \Lambda^*).$$

Proof. Let $\Upsilon^{k+1} = \Theta^{k+1} - \sum_{i=1}^n Z_i^{k+1} + Z_i^{k+1\top} - E^{k+1}$. Using the quadratic function $f_e(E) = \frac{\lambda_e}{2}\|E\|_F^2$, we have that

$$\begin{aligned}
f_e(E^{k+1} - \Upsilon^{k+1}) &= \frac{\lambda_e}{2}\|E^{k+1} - \Upsilon^{k+1}\|_F^2 \\
&= \frac{\lambda_e}{2}\|E^{k+1}\|^2 - \lambda_e\langle E^{k+1}, \Upsilon^{k+1}\rangle + \frac{\lambda_e}{2}\|\Upsilon^{k+1}\|_F^2. \qquad (39)
\end{aligned}$$

Using (39) and the fact that each function f_i is lower bounded, there exists $\underline{\mathcal{L}}$, such that

$$\begin{aligned}
\mathcal{L}_\gamma(U^{k+1}) &= \mathcal{G}(X,\Theta^{k+1}) + f_1(Z_1^{k+1}) + \ldots f_n(Z_n^{k+1}) + \frac{\lambda_e}{2}\|E^{k+1} - \Upsilon^{k+1}\|_F^2 \\
&+ \frac{\gamma - \lambda_e}{2}\|\Upsilon^{k+1}\|_F^2 \geq \underline{g} + \underline{f_1} + \cdots + \underline{f_n} \geq \underline{\mathcal{L}}, \qquad (40)
\end{aligned}$$

since $\mathcal{G}(X,\Theta^{k+1})$ and $f_i(Z_i^{k+1})(i = 1, \ldots, n)$ are all lower bounded.

Now, using Lemma 15, we have that

$$\frac{\vartheta}{2}\sum_{k=0}^{K}\Big(\|\Theta^k-\Theta^{k+1}\|_F^2+\sum_{i=1}^{n}\|Z_i^k-Z_i^{k+1}\|_F^2 \;+\; \|E^k-E^{k+1}\|_F^2+\|\Lambda^k-\Lambda^{k+1}\|_F^2\Big) \leq \mathcal{L}_\gamma(U^0)-\underline{\mathcal{L}}. \tag{41}$$

Lemma 15 together with (41) shows that $\mathcal{L}_\gamma(U^k)$ converges to $\mathcal{L}_\gamma(U^*)$. Note that (41) and the coerciveness of $\mathcal{G}(X,\Theta)$ and f_i $(i=1,\ldots,n)$ imply that $\{(\Theta^k,Z_1^k,\ldots,Z_n^k)\}$ is a bounded sequence. This together with the updating formula of Λ^{k+1} and (41) yield the boundedness of E^{k+1}. Moreover, the fact that $\Lambda^k=-\lambda_e E^k$, gives the boundedness of Λ^k, which implies that the entire sequence $\{U^k\}$ is a bounded one. Therefore, there exists a subsequence

$$U^{k_s}=(\Theta^{k_s},Z_1^{k_s},\ldots,Z_n^{k_s},E^{k_s};\Lambda^{k_s}),\qquad s=0,1,\ldots$$

such that $U^{k_s}\to U^*$ as $s\to\infty$.

Now, using the fact that $\mathcal{G}(X,\Theta)$, $f_i(Z_i)$ $(i=1,\ldots,n)$ and $f_e(E)$ are continuous functions, we have that

$$\lim_{s\to\infty}\mathcal{G}(X,\Theta^{k_s})=g(\Theta^*),\quad \lim_{s\to\infty}f_i(Z_i^{k_q})=f_i(Z_i^*),\quad \lim_{s\to\infty}f_e(E_i^{k_q})=f_e(E_i^*).$$

□

Lemma 17 *Algorithm 1 either stops at a stationary point of the problem* (19) *or generates an infinite sequence* $\{U^k\}$*, so that any limit point of* $\{U^k\}$ *is a critical point of* $\mathcal{L}_\gamma(U^k)$ (19)*.*

Proof. From the definition of the augmented Lagrangian function in (19), we have that

$$\begin{aligned}
&\nabla\mathcal{G}(X,\Theta^{k+1})-\Lambda^{k+1}+\gamma\Upsilon^{k+1}=\nabla_\Theta\mathcal{L}_\gamma(U^{k+1}),\\
&\partial f_i(Z_i^{k+1})-\Lambda^{k+1}-{\Lambda^{k+1}}^\top-\gamma(\Upsilon^{k+1}+{\Upsilon^{k+1}}^\top)\in\partial_{Z_i}\mathcal{L}_\gamma(U^{k+1}),\quad i=1,\ldots,n,\\
&\lambda_e E^{k+1}+\Lambda^{k+1}-\gamma\Upsilon^{k+1}=\nabla_E\mathcal{L}_\gamma(U^{k+1}),\\
&\gamma\Upsilon^{k+1}=-\nabla_\Lambda\mathcal{L}_\gamma(U^{k+1}),
\end{aligned}\tag{42}$$

where $\Upsilon^{k+1}=\Theta^{k+1}-\sum_{i=1}^n Z_i^{k+1}+{Z_i^{k+1}}^\top-E^{k+1}$.

Moreover, the updating formula of Λ^{k+1}, (20) and (28) yields that

$$\begin{aligned}
\nabla\mathcal{G}(X,\Theta^{k+1})-\Lambda^{k+1} &= \gamma\Big(\Theta^{k+1}-\Theta^k\\
&+\sum_{i=1}^n Z_i^k-Z_i^{k+1}+(Z_i^k-Z_i^{k+1})^\top+E^k-E^{k+1}\Big)\\
\partial f_1(Z_1^{k+1})-\Lambda^{k+1}-{\Lambda^{k+1}}^\top &= \gamma\varrho(Z_1^k-Z_1^{k+1})+\gamma\Big(\Theta^{k+1}-\Theta^k\\
&+(\Theta^{k+1}-\Theta^k)^\top+\sum_{i=1}^n Z_i^k-Z_i^{k+1}\\
&+(Z_i^k-Z_i^{k+1})^\top+E^k-E^{k+1}+(E^k-E^{k+1})^\top\Big)
\end{aligned}\tag{43}$$

$$\begin{aligned}
\partial f_i(Z_i^{k+1})-\Lambda^{k+1}-{\Lambda^{k+1}}^\top &= \gamma\varrho(Z_i^k-Z_i^{k+1})\\
&+\gamma\Big(\Theta^{k+1}-\Theta^k+(\Theta^{k+1}-\Theta^k)^\top\\
&+\sum_{j=i}^n Z_i^k-Z_i^{k+1}+(Z_i^k-Z_i^{k+1})^\top\\
&+E^k-E^{k+1}+(E^k-E^{k+1})^\top\Big)\qquad i=2,\ldots,n,\\
\lambda_e E^{k+1}+\Lambda^{k+1} &= 0.
\end{aligned}\tag{44}$$

Combining (42), (43), and the updating formula of Λ^{k+1}, we have that

$$(\hbar_\Theta^{k+1},\hbar_1^{k+1},\ldots,\hbar_n^{k+1},\hbar_E^{k+1},\hbar_\Lambda^{k+1})\in\partial\mathcal{L}_\gamma(U^{k+1}),\tag{45}$$

where

$$\begin{aligned}
\hbar_\Theta^{k+1} &:= \Lambda^k-\Lambda^{k+1}+\gamma\Big(\Theta^{k+1}-\Theta^k+\sum_{i=1}^n Z_i^k-Z_i^{k+1}+(Z_i^k-Z_i^{k+1})^\top+E^k-E^{k+1}\Big)\\
\hbar_{Z_1}^{k+1} &:= \Lambda^k-\Lambda^{k+1}+(\Lambda^k-\Lambda^{k+1})^\top+\gamma\varrho(Z_1^k-Z_1^{k+1})\\
&+\gamma\Big(\Theta^{k+1}-\Theta^k+(\Theta^{k+1}-\Theta^k)^\top+\sum_{i=1}^n Z_i^k-Z_i^{k+1}+(Z_i^k-Z_i^{k+1})^\top\\
&+E^k-E^{k+1}+(E^k-E^{k+1})^\top\Big)\\
\hbar_{Z_i}^{k+1} &:= \Lambda^k-\Lambda^{k+1}+(\Lambda^k-\Lambda^{k+1})^\top+\gamma\varrho(Z_i^k-Z_i^{k+1})\\
&+\gamma\Big(\Theta^{k+1}-\Theta^k+(\Theta^{k+1}-\Theta^k)^\top+\sum_{j=i}^n Z_i^k-Z_i^{k+1}+(Z_i^k-Z_i^{k+1})^\top\\
&+E^k-E^{k+1}+(E^k-E^{k+1})^\top\Big),\qquad i=2,\ldots,n,\\
\hbar_E^{k+1} &:= \Lambda^k-\Lambda^{k+1},\\
\hbar_\Lambda^{k+1} &:= \frac{1}{\gamma}(\Lambda^{k+1}-\Lambda^k),
\end{aligned}\tag{46}$$

Now, using (41), we obtain that

$$\lim_{k\to\infty}(\|h_\Theta^{k+1}\|_F, \|h_{Z_1}^{k+1}\|_F, \ldots, \|h_{Z_n}^{k+1}\|_F, \|h_E^{k+1}\|_F; \|R_\Lambda^{k+1}\|_F) = (0, \ldots, 0). \tag{47}$$

Suppose that Algorithm 1 does not stop at a stationary point. Using Lemma 16, there exists a subsequence U^{k_s}, such that $U^{k_s} \to U^*$ as $s \to \infty$. Using (45) and (47), we conclude that $(0, \ldots, 0) \in \partial\mathcal{L}_\gamma(U^*)$. □

Proof of Theorem 10. Lemmas 16 and 17 imply that $\{U^k\}$ is a bounded sequence and the set of limit points of $\{U^k\}$ starting from U^0 is non-empty, respectively. Moreover, Lemma 5 and Remark 5 of (Bolte et al., 2014) imply that the set of limit points of $\{U^k\}$ starting from U^0 is compact. The remainder of the proof of this Theorem follows along similar lines to the proof of Theorem 1 in (Bolte et al., 2014), by utilizing the K-L property of the problem (19) (see, Remark 14). □

References

D Ataee Tarzanagh, M Reza Peyghami, and H Mesgarani. A new nonmonotone trust region method for unconstrained optimization equipped by an efficient adaptive radius. *Optimization Methods and Software*, 29(4):819–836, 2014.

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

Jonathan Barzilai and Jonathan M Borwein. Two-point step size gradient methods. *IMA journal of numerical analysis*, 8(1):141–148, 1988.

Sumanta Basu and George Michailidis. Regularized estimation in sparse high-dimensional time series models. *The Annals of Statistics*, 43(4):1535–1567, 2015.

Sumanta Basu, Ali Shojaie, and George Michailidis. Network granger causality with inherent grouping structure. *The Journal of Machine Learning Research*, 16(1):417–453, 2015.

Quentin Berthet, Philippe Rigollet, and Piyush Srivastava. Exact recovery in the Ising block model. *arXiv:1612.03880*, 2016.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2): 459–494, 2014.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

Tony Cai and Weidong Liu. Adaptive thresholding for sparse covariance matrix estimation. *Journal of the American Statistical Association*, 106(494):672–684, 2011.

Venkat Chandrasekaran, Pablo A Parrilo, and Alan S Willsky. Latent variable graphical model selection via convex optimization. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pages 1610–1613. IEEE, 2010.

Caihua Chen, Bingsheng He, Yinyu Ye, and Xiaoming Yuan. The direct extension of admm for multi-block convex minimization problems is not necessarily convergent. *Mathematical Programming*, 155(1-2):57–79, 2016.

Victor Chernozhukov, Christian Hansen, and Yuan Liao. A lava attack on the recovery of sums of dense and sparse signals. *The Annals of Statistics*, 45(1):39–76, 2017.

Patrick Danaher, Pei Wang, and Daniela M Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(2):373–397, 2014.

Damek Davis and Wotao Yin. A three-operator splitting scheme and its optimization applications. *Set-Valued and Variational Analysis*, pages 1–30, 2015.

Elizabeth D Dolan and Jorge J Moré. Benchmarking optimization software with performance profiles. *Mathematical programming*, 91(2):201–213, 2002.

Mathias Drton and Thomas S Richardson. A new algorithm for maximum likelihood estimation in gaussian graphical models for marginal independence. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pages 184–191. Morgan Kaufmann Publishers Inc., 2002.

Mathias Drton and Thomas S Richardson. Graphical methods for efficient likelihood inference in gaussian covariance models. *Journal of Machine Learning Research*, 9(May): 893–914, 2008.

Ehsan Elhamifar and René Vidal. Sparse subspace clustering. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2790–2797. IEEE, 2009.

Roger Fletcher. On the barzilai-borwein method. *Optimization and control with applications*, pages 235–256, 2005.

Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Roger Guimera and Luis A Nunes Amaral. Functional cartography of complex metabolic networks. *Nature*, 433(7028):895–900, 2005.

Jian Guo, Elizaveta Levina, George Michailidis, and Ji Zhu. Joint estimation of multiple graphical models. *Biometrika*, 98(1):1–15, 2011a.

Jian Guo, Elizaveta Levina, George Michailidis, and Ji Zhu. Asymptotic properties of the joint neighborhood selection method for estimating categorical markov networks. *arXiv preprint math.PR/0000000*, 2011b.

Jian Guo, Jie Cheng, Elizaveta Levina, George Michailidis, and Ji Zhu. Estimating heterogeneous graphical models for discrete data with an application to roll call voting. *The Annals of Applied Statistics*, 9(2):821, 2015.

Davood Hajinezhad and Mingyi Hong. Nonconvex alternating direction method of multipliers for distributed sparse principal component analysis. In *Signal and Information Processing (GlobalSIP), 2015 IEEE Global Conference on*, pages 255–259. IEEE, 2015.

Davood Hajinezhad, Mingyi Hong, Tuo Zhao, and Zhaoran Wang. Nestt: A nonconvex primal-dual splitting method for distributed and stochastic optimization. In *Advances in Neural Information Processing Systems*, pages 3215–3223, 2016.

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Holger Höfling and Robert Tibshirani. Estimation of sparse binary pairwise markov networks using pseudo-likelihoods. *Journal of Machine Learning Research*, 10(Apr):883–906, 2009.

Mingyi Hong and Zhi-Quan Luo. On the linear convergence of the alternating direction method of multipliers. *Mathematical Programming*, 162(1-2):165–199, 2017.

Ernst Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik A Hadrons and Nuclei*, 31(1):253–258, 1925.

Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. Group lasso with overlap and graph lasso. In *Proceedings of the 26th annual international conference on machine learning*, pages 433–440. ACM, 2009.

Noureddine El Karoui. Operator norm consistent estimation of large-dimensional sparse covariance matrices. *The Annals of Statistics*, pages 2717–2756, 2008.

Su-In Lee, Varun Ganapathi, and Daphne Koller. Efficient structure learning of Markov networks using ℓ_1-regularization. In *Advances in neural Information processing systems*, pages 817–824, 2007.

Anna CF Lewis, Charlotte M Deane, Mason A Porter, and Nick S Jones. The function of communities in protein interaction networks at multiple scales. *BMC systems biology*, 4 (1):100, 2010.

Lun Li, David Alderson, John C Doyle, and Walter Willinger. Towards a theory of scale-free graphs: Definition, properties, and implications. *Internet Mathematics*, 2(4):431–523, 2005.

Fredrik Liljeros, Christofer R Edling, Luis A Nunes Amaral, H Eugene Stanley, and Yvonne Åberg. The web of human sexual contacts. *Nature*, 411(6840):907–908, 2001.

Tianyi Lin, Shiqian Ma, and Shuzhong Zhang. On the global linear convergence of the ADMM with multiblock variables. *SIAM Journal on Optimization*, 25(3):1478–1497, 2015.

Tianyi Lin, Shiqian Ma, and Shuzhong Zhang. Iteration complexity analysis of multi-block ADMM for a family of convex minimization without strong convexity. *Journal of Scientific Computing*, 69(1):52–81, 2016.

Zhouchen Lin, Risheng Liu, and Zhixun Su. Linearized alternating direction method with adaptive penalty for low-rank representation. In *Advances in neural information processing systems*, pages 612–620, 2011.

Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):171–184, 2013.

Qiang Liu and Alexander Ihler. Learning scale free networks by reweighted l1 regularization. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 40–48, 2011.

Risheng Liu, Zhouchen Lin, and Zhixun Su. Linearized alternating direction method with parallel splitting and adaptive penalty for separable convex programs in machine learning. *Machine Learning*, 99(2), 2015.

Lennart Ljung. System identification. In *Signal analysis and prediction*, pages 163–173. Springer, 1998.

Helmut Lütkepohl. *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005.

Shiqian Ma, Lingzhou Xue, and Hui Zou. Alternating direction methods for latent variable gaussian graphical model selection. *Neural computation*, 25(8):2172–2198, 2013.

Karthik Mohan, Mike Chung, Seungyeop Han, Daniela Witten, Su-In Lee, and Maryam Fazel. Structured learning of Gaussian graphical models. In *Advances in neural information processing systems*, pages 620–628, 2012.

Stefano Monti, Pablo Tamayo, Jill Mesirov, and Todd Golub. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine learning*, 52(1):91–118, 2003.

Mark EJ Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98(2):404–409, 2001.

Mark EJ Newman. Communities, modules and large-scale structure in networks. *Nature Physics*, 8(1):25–31, 2012.

Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

Guillaume Obozinski, Laurent Jacob, and Jean-Philippe Vert. Group lasso with overlaps: the latent group lasso approach. *arXiv preprint arXiv:1110.0413*, 2011.

Nikhil Rao, Robert Nowak, Christopher Cox, and Timothy Rogers. Classification with the sparse group lasso. *IEEE Transactions on Signal Processing*, 64(2):448–463, 2016.

Pradeep Ravikumar, Martin J Wainwright, Garvesh Raskutti, and Bin Yu. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

Marcos Raydan. The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM Journal on Optimization*, 7(1):26–33, 1997.

Garry Robins, Pip Pattison, Yuval Kalish, and Dean Lusher. An introduction to exponential random graph $p*$ models for social networks. *Social networks*, 29(2):173–191, 2007.

Adam J Rothman, Peter J Bickel, Elizaveta Levina, and Ji Zhu. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515, 2008.

Katya Scheinberg, Shiqian Ma, and Donald Goldfarb. Sparse inverse covariance selection via alternating linearization methods. In *Advances in Neural Information Processing Systems*, pages 2101–2109, 2010.

Defeng Sun, Kim-Chuan Toh, and Liuqin Yang. A convergent 3-block semiproximal alternating direction method of multipliers for conic programming with 4-type constraints. *SIAM journal on Optimization*, 25(2):882–915, 2015.

Kean Ming Tan, Palma London, Karthik Mohan, Su-In Lee, Maryam Fazel, and Daniela M Witten. Learning graphical models with hubs. *Journal of Machine Learning Research*, 15(1):3297–3331, 2014.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

Amanda L Traud, Eric D Kelsic, Peter J Mucha, and Mason A Porter. Comparing community structure to characteristics in online collegiate social networks. *SIAM review*, 53(3): 526–543, 2011.

Ruey S Tsay. *Analysis of financial time series*, volume 543. John Wiley & Sons, 2005.

Pedro A Valdés-Sosa, Jose M Sánchez-Bornot, Agustín Lage-Castellanos, Mayrim Vega-Hernández, Jorge Bosch-Bayard, Lester Melie-García, and Erick Canales-Rodríguez. Estimating brain functional connectivity with sparse multivariate autoregression. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 360(1457): 969–981, 2005.

Xiangfeng Wang, Mingyi Hong, Shiqian Ma, and Zhi-Quan Luo. Solving multiple-block separable convex minimization problems using two-block alternating direction method of multipliers. *arXiv preprint arXiv:1308.5294*, 2013.

Lingzhou Xue, Shiqian Ma, and Hui Zou. Positive-definite 1-penalized estimation of large covariance matrices. *Journal of the American Statistical Association*, 107(500):1480–1491, 2012.

Junfeng Yang and Xiaoming Yuan. Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization. *Mathematics of computation*, 82(281): 301–329, 2013.

Ming Yuan and Yi Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

Changjiu Zhao, Brian Earl Eisinger, Terri M Driessen, and Stephen C Gammie. Addiction and reward-related genes show altered expression in the postpartum nucleus accumbens. *Frontiers in behavioral neuroscience*, 8, 2014.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

Journal of Machine Learning Research 18 (2018) 1–71 Submitted 8/16; Revised 6/17; Published 5/18

Sparse Exchangeable Graphs and Their Limits via Graphon Processes

Christian Borgs BORGS@MICROSOFT.COM
Microsoft Research
One Memorial Drive
Cambridge, MA 02142, USA

Jennifer T. Chayes JCHAYES@MICROSOFT.COM
Microsoft Research
One Memorial Drive
Cambridge, MA 02142, USA

Henry Cohn COHN@MICROSOFT.COM
Microsoft Research
One Memorial Drive
Cambridge, MA 02142, USA

Nina Holden NINAH@MATH.MIT.EDU
Department of Mathematics
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

Editor: Edoardo M. Airoldi

Abstract

In a recent paper, Caron and Fox suggest a probabilistic model for sparse graphs which are exchangeable when associating each vertex with a time parameter in $\mathbb{R}_+$. Here we show that by generalizing the classical definition of graphons as functions over probability spaces to functions over σ-finite measure spaces, we can model a large family of exchangeable graphs, including the Caron-Fox graphs and the traditional exchangeable dense graphs as special cases. Explicitly, modelling the underlying space of features by a σ-finite measure space $(S, \mathcal{S}, \mu)$ and the connection probabilities by an integrable function $W\colon S \times S \to [0,1]$, we construct a random family $(G_t)_{t\geq 0}$ of growing graphs such that the vertices of G_t are given by a Poisson point process on S with intensity $t\mu$, with two points x, y of the point process connected with probability $W(x,y)$. We call such a random family a *graphon process*. We prove that a graphon process has convergent subgraph frequencies (with possibly infinite limits) and that, in the natural extension of the cut metric to our setting, the sequence converges to the generating graphon. We also show that the underlying graphon is identifiable only as an equivalence class over graphons with cut distance zero. More generally, we study metric convergence for arbitrary (not necessarily random) sequences of graphs, and show that a sequence of graphs has a convergent subsequence if and only if it has a subsequence satisfying a property we call *uniform regularity of tails*. Finally, we prove that every graphon is equivalent to a graphon on $\mathbb{R}_+$ equipped with Lebesgue measure.

Keywords: graphons, graph convergence, sparse graph convergence, modelling of sparse networks, exchangeable graph models

1. Introduction

The theory of graphons has provided a powerful tool for sampling and studying convergence properties of sequences of dense graphs. Graphons characterize limiting properties of dense graph sequences, such as properties arising in combinatorial optimization and statistical physics. Furthermore, sequences of dense graphs sampled from a (possibly random) graphon are characterized by a natural notion of exchangeability via the Aldous-Hoover theorem. This paper presents an analogous theory for sparse graphs.

In the past few years, graphons have been used as non-parametric extensions of stochastic block models, to model and learn large networks. There have been several rigorous papers on the subject of consistent estimation using graphons (see, for example, papers by Bickel and Chen, 2009, Bickel, Chen, and Levina, 2011, Rohe, Chatterjee, and Yu, 2011, Choi, Wolfe, and Airoldi, 2012, Wolfe and Olhede, 2013, Gao, Lu, and Zhou, 2015, Chatterjee, 2015, Klopp, Tsybakov, and Verzelen, 2017, and Borgs, Chayes, Cohn, and Ganguly, 2015, as well as references therein), and graphons have also been used to estimate real-world networks, such as Facebook and LinkedIn (E. M. Airoldi, private communication, 2015). This makes it especially useful to have graphon models for sparse networks with unbounded degrees, which are the appropriate description of many large real-world networks.

In the classical theory of graphons as studied by, for example, Borgs, Chayes, Lovász, Sós, and Vesztergombi (2006), Lovász and Szegedy (2006), Borgs, Chayes, Lovász, Sós, and Vesztergombi (2008), Bollobás and Riordan (2009), Borgs, Chayes, and Lovász (2010), and Janson (2013), a graphon is a symmetric $[0,1]$-valued function defined on a probability space. In our generalized theory we let the underlying measure space of the graphon be a σ-finite measure space; i.e., we allow the space to have infinite total measure. More precisely, given a σ-finite measure space $\mathscr{S} = (S, \mathcal{S}, \mu)$ we define a graphon to be a pair $\mathcal{W} = (W, \mathscr{S})$, where $W\colon S \times S \to \mathbb{R}$ is a symmetric integrable function, with the special case when W is $[0,1]$-valued being most relevant for the random graphs studied in the current paper. We present a random graph model associated with these generalized graphons which has a number of properties making it appropriate for modelling sparse networks, and we present a new theory for convergence of graphs in which our generalized graphons arise naturally as limits of sparse graphs.

Given a $[0,1]$-valued graphon $\mathcal{W} = (W, \mathscr{S})$ with $\mathscr{S} = (S, \mathcal{S}, \mu)$ a σ-finite measure space, we will now define a random process which generalizes the classical notion of $\mathcal{W}$-random graphs, introduced in the statistics literature (Hoff, Raftery, and Handcock, 2002) under the name latent position graphs, in the context of graph limits (Lovász and Szegedy, 2006) as $\mathcal{W}$-random graphs, and in the context of extensions of the classical random graph theory (Bollobás, Janson, and Riordan, 2007) as inhomogeneous random graphs. Recall that in the classical setting where $\mathcal{W}$ is defined on a probability space, $\mathcal{W}$-random graphs are generated by first choosing n points $x_1, \dots, x_n$ i.i.d. from the probability distribution μ over the feature space S, and then connecting the vertices i and j with probability $W(x_i, x_j)$. Here, inspired by Caron and Fox (2014), we generalize this to arbitrary σ-finite measure spaces by first considering a Poisson point process[1] Γ_t with intensity $t\mu$ on S for any fixed $t > 0$, and

1. We will make this construction more precise in Section 2.4; in particular, we will explain that we may associate Γ_t with a collection of random variables $x_i \in S$. The same result holds for the Poisson point process Γ considered in the next paragraph.

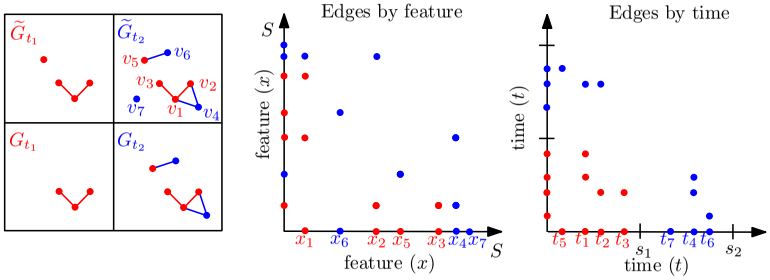


Figure 1: This figure illustrates how we can generate a graphon process $(G_t)_{t\geq 0}$ from a graphon $\mathcal{W} = (W, \mathscr{S})$, where $\mathscr{S} = (S, \mathcal{S}, \mu)$ is a σ-finite measure space. The two coordinate axes on the middle figure represent our feature space S, where the red (resp. blue) dots on the axes represent vertices born during $[0, s_1]$ (resp. $(s_1, s_2]$) for $0 < s_1 < s_2$, and the red (resp. blue) dots in the interior of the first quadrant represent edges in G_t for $t \geq s_1$ (resp. $t \geq s_2$). The graph G_t is an induced subgraph of a graph $\widetilde{G}_t$ with infinitely many vertices in the case $\mu(S) = \infty$, such that G_t is obtained from $\widetilde{G}_t$ by removing isolated vertices. At time $t \geq 0$ the marginal law of the features of $V(\widetilde{G}_t)$ is a Poisson point process on S with intensity $t\mu$. Two distinct vertices with features x and x', respectively, are connected to each other by an undirected edge with probability $W(x, x')$. The coordinate axes on the right figure represent time $\mathbb{R}_+$. We get the graph G_t by considering the edges restricted to $[0, t]^2$. Note that the coordinate axes in the right figure and the graphs $\widetilde{G}_t$ in the left figure are slightly inaccurate if we assume $\mu(S) = \infty$, since in this case there are infinitely many isolated vertices in $\widetilde{G}_t$ for each $t > 0$. We have chosen to label the vertices by the order in which they appear in G_t, where ties are resolved by considering the time the vertices were born, i.e., by considering the time they appeared in $\widetilde{G}_t$.

then connecting two points x_i, x_j in Γ_t with probability $W(x_i, x_j)$. As explained in the next paragraph, this leads to a family of graphs $(\widetilde{G}_t)_{t\geq 0}$ such that the graphs $\widetilde{G}_t$ have almost surely at most countably infinitely many vertices and (assuming appropriate integrability conditions on W, e.g., $W \in L^1$) a finite number of edges. Removing all isolated vertices from $\widetilde{G}_t$, we obtain a family of graphs $(G_t)_{t\geq 0}$ that are almost surely finite. We refer to the families $(\widetilde{G}_t)_{t\geq 0}$ and $(G_t)_{t\geq 0}$ as *graphon processes*; when it is necessary to distinguish the two, we call them graphon processes with or without isolated vertices, respectively.

To interpret the graphon process $(G_t)_{t\geq 0}$ as a family of growing graphs we will need to couple the graphs G_t for different times $t \geq 0$. To this end, we consider a Poisson point process Γ on $\mathbb{R}_+ \times S$ (with $\mathbb{R}_+ := [0, \infty)$ being equipped with the Borel σ-algebra and Lebesgue measure). Each point $v = (t, x)$ of Γ corresponds to a vertex of an infinite graph $\widetilde{G}$, where the coordinate t is interpreted as the time the vertex is born and the coordinate

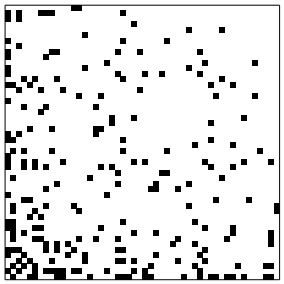
x describes a feature of the vertex. Two distinct vertices $v = (t, x)$ and $v' = (t', x')$ are connected by an undirected edge with probability $W(x, x')$, independently for each possible pair of distinct vertices. For each fixed time $t \geq 0$ define a graph G_t by considering the induced subgraph of $\widetilde{G}$ corresponding to vertices which are born at time t or earlier, where we do not include vertices which would be isolated in G_t. See Figure 1 for an illustration. The family of growing graphs $(G_t)_{t\geq 0}$ just described includes classical dense $\mathcal{W}$-random graphs (up to isolated vertices) and the sparse graphs studied by Caron and Fox (2014) and Herlau, Schmidt, and Mørup (2016) as special cases, and is (except for minor technical differences) identical to the family of random graphs studied by Veitch and Roy (2015), a paper which was written in parallel with our paper; see our remark at the end of this introduction.

The graphon process $(\widetilde{G}_t)_{t\geq 0}$ satisfies a natural notion of exchangeability. Roughly speaking, in our setting this means that the features of newly born vertices are homogeneous in time. More precisely, it can be defined as joint exchangeability of a random measure in $\mathbb{R}^2_+$, where the two coordinates correspond to time, and each edge of the graph corresponds to a point mass. We will prove that graphon processes as defined above, with W integrable and possibly random, are characterized by exchangeability of the random measure in $\mathbb{R}^2_+$ along with a certain regularity condition we call *uniform regularity of tails*. See Proposition 26 in Section 2.4. This result is an analogue in the setting of possibly sparse graphs satisfying the aforementioned regularity condition of the Aldous-Hoover theorem (Aldous, 1981; Hoover, 1979), which characterizes $\mathcal{W}$-random graphs over probability spaces as graphs that are invariant in law under permutation of their vertices.

The graphon processes defined above also have a number of other properties making them particularly natural to model sparse graphs or networks. They are suitable for modelling networks which grow over time since no additional rescaling parameters (like the explicitly given density dependence on the number of vertices specified by Bollobás and Riordan, 2009, and Borgs, Chayes, Cohn, and Zhao, 2014a) are necessary; all information about the random graph model is encoded by the graphon alone. The graphs are *projective* in the sense that if $s < t$ the graph G_s is an induced subgraph of G_t. Finally, a closely related family of weighted graphs is proven by Caron and Fox (2014) to have power law degree distribution for certain $\mathcal{W}$, and our graphon processes are expected to behave similarly. The graphon processes studied in this paper have a different qualitative behavior than the sparse $\mathcal{W}$-random graphs studied by Bollobás and Riordan (2009) and Borgs, Chayes, Cohn, and Zhao (2014a,b) (see Figure 2), with the only overlap of the two theories occurring when the graphs are dense. If the sparsity of the graphs is caused by the degrees of the vertices being scaled down approximately uniformly over time, then the model studied by Bollobás and Riordan (2009) and Borgs, Chayes, Cohn, and Zhao (2014a,b) is most natural. If the sparsity is caused by later vertices typically having lower connectivity probabilities than earlier vertices, then the model presented in this paper is most natural. The sampling method we will use in our forthcoming paper (Borgs, Chayes, Cohn, and Holden, 2017) generalizes both of these methods.

To compare different models, and to discuss notions of convergence, we introduce the following natural generalization of the cut metric for graphons on probability spaces to our setting. For two graphons $\mathcal{W}_1 = (W_1, \mathscr{S}_1)$ and $\mathcal{W}_2 = (W_2, \mathscr{S}_2)$, this metric is easiest to define when the two graphons are defined over the same space. However, for applications we

rescaled graphon on probability space



graphon with non-compact support

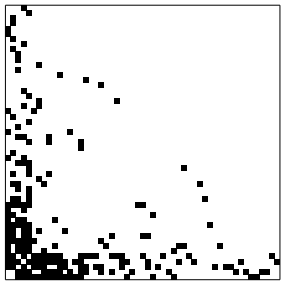


Figure 2: The adjacency matrices of graphs sampled as described by Borgs, Chayes, Cohn, and Zhao (2014a) (left) and in this paper (right), where we used the graphon $\mathcal{W}_1 = (W_1, [0,1])$ (left) and the graphon $\mathcal{W}_2 = (W_2, \mathbb{R}_+)$ (right), with $W_1(x_1, x_2) = x_1^{-1/2} x_2^{-1/2}$ for $x_1, x_2 \in [0,1]$ and $W_2(x_1, x_2) = \min(0.8, 7\min(1, x_1^{-2})\min(1, x_2^{-2}))$ for $x_1, x_2 \in \mathbb{R}_+$. Black (resp. white) indicates that there is (resp. is not) an edge. We rescaled the height of the graphon by $\rho := 1/40$ on the left figure. As described by Borgs, Chayes, Cohn, and Zhao (2014a,b) the type of each vertex is sampled independently and uniformly from $[0,1]$, and each pair of vertices is connected with probability $\min(\rho W_1, 1)$. In the right figure the vertices were sampled by a Poisson point process on $\mathbb{R}_+$ of intensity $t = 4$, and two vertices were connected independently with a probability given by W_2; see Section 2.4 and the main text of this introduction. The two graphs have very different qualitative properties. In the left graph most vertices have a degree close to the average degree, where the average degree depends on our scaling factor ρ. In the right graph the edges are distributed more inhomogeneously: most of the edges are contained in induced subgraphs of constant density, and the sparsity is caused by a large number of vertices with very low degree.

want to compare graphons over different spaces, say two Borel spaces $\mathscr{S}_1$ and $\mathscr{S}_2$. Assuming that both Borel spaces have infinite total measure, the cut distance between $\mathcal{W}_1$ and $\mathcal{W}_2$ can then be defined as

$$\delta_\square(\mathcal{W}_1, \mathcal{W}_2) = \inf_{\psi_1,\psi_2} \sup_{U,V\subseteq\mathbb{R}_+^2} \left| \int_{U\times V} \left(W_1^{\psi_1} - W_2^{\psi_2}\right) d\mu\, d\mu \right|, \tag{1}$$

where we take the infimum over measure-preserving maps $\psi_j \colon \mathbb{R}_+ \to S_j$ for $j = 1, 2$, $W_j^{\psi_j}(x, y) := W_j(\psi_j(x), \psi_j(y))$ for $x, y \in \mathbb{R}_+$, and the supremum is over measurable sets $U, V \subseteq \mathbb{R}_+^2$. (See Definition 5 below for the definition of the cut distance for graphons over general spaces, including the case where one or both spaces have finite total mass.) We call two graphons *equivalent* if they have cut distance zero. As we will see, two graphons are equivalent if and only if the random families $(G_t)_{t\geq 0}$ generated from these graphons have the same distribution; see Theorem 27 below.

To compare graphs and graphons, we embed a graph on n vertices into the set of step functions over $[0,1]^2$ in the usual way by decomposing $[0,1]$ into adjacent intervals $I_1, \ldots, I_n$ of lengths $1/n$, and define a step function W^G as the function which is equal to 1 on $I_i \times I_j$ if i and j are connected in G, and equal to 0 otherwise. Extending W^G to a function on $\mathbb{R}_+^2$ by setting it to zero outside of $[0,1]^2$, we can then compare graphs to graphons on measure spaces of infinite mass, and in particular we get a notion of convergence in metric of a sequence of graphs $(G_n)_{n\in\mathbb{N}}$ to a graphon $\mathcal{W}$.

In the classical theory of graph convergence, such a sequence will converge to the zero graphon whenever the sequence is sparse.[2] We resolve this difficulty by rescaling the input arguments of the step function W^G so as to get a "stretched graphon" $\mathcal{W}^{G,s} = (W^{G,s}, \mathbb{R}_+)$ satisfying $\|W^{G,s}\|_1 = 1$. Equivalently, we may interpret $\mathcal{W}^{G,s}$ as a graphon where the measure of the underlying measure space is rescaled. See Figure 3 for an illustration, which also compares the rescaling in the current paper with the rescaling considered by Borgs, Chayes, Cohn, and Zhao (2014a). We say that $(G_n)_{n\in\mathbb{N}}$ converges to a graphon $\mathcal{W}$ (with L^1 norm equal to 1) for the stretched cut metric if $\lim_{n\to\infty} \delta_\square(\mathcal{W}^{G_n,s}, \mathcal{W}) = 0$. Graphons on σ-finite measure spaces of infinite total measure may therefore be considered as limiting objects for sequences of sparse graphs, similarly as graphons on probability spaces are considered limits of dense graphs. We prove that graphon processes converge to the generating graphon in the stretched cut metric; see Proposition 28 in Section 2.4. We will also consider another family of random sparse graphs associated with a graphon $\mathcal{W}$ over a σ-finite measure space, and prove that these graphs are also converging for the stretched cut metric.

Particular random graph models of special interest arise by considering certain classes of graphons $\mathcal{W}$. Caron and Fox (2014) consider graphons on the form $W(x_1, x_2) = 1 - \exp(-f(x_1)f(x_2))$ (with a slightly different definition on the diagonal, since they also allow for self-edges) for certain decreasing functions $f \colon \mathbb{R}_+ \to \mathbb{R}_+$. In this model x represents a sociability parameter of each vertex. A multi-edge version of this model allows for an alternative sampling procedure to the one we present above (Caron and Fox, 2014, Section 3). Herlau, Schmidt, and Mørup (2016) introduced a generalization of the model of Caron and Fox (2014) to graphs with block structure. In this model each node is associated to a

2. Here, as usual, a sequence of simple graphs is considered sparse if the number of edges divided by the square of the number of vertices goes to zero.

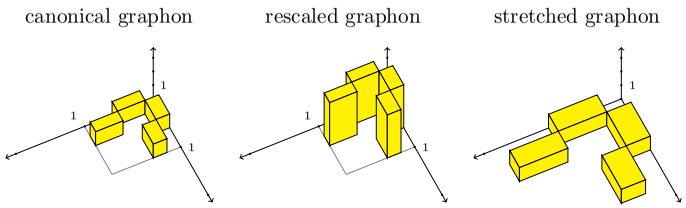


Figure 3: The figure shows three graphons associated with the same simple graph G on five vertices. In the classical theory of graphons all simple sparse graphs converge to the zero graphon. We may prevent this by renormalizing the graphons, either by rescaling the height of the graphon (middle) or by stretching the domain on which it is defined (right). The first approach was chosen by Bollobás and Riordan (2009) and Borgs, Chayes, Cohn, and Zhao (2014a,b), and the second approach is chosen in this paper. In our forthcoming paper (Borgs, Chayes, Cohn, and Holden, 2017) we choose a combined approach, where the renormalization depends on the observed graph.

type from a finite index set $[K] := \{1, \dots, K\}$ for some $K \in \mathbb{N}$, in addition to its sociability parameter, such that the probability of two nodes connecting depends both on their type and their sociability. More generally we can obtain sparse graphs with block structure by considering integrable functions $W_{k_1,k_2} \colon \mathbb{R}_+^2 \to [0,1]$ for $k_1, k_2 \in \{1, \dots, K\}$, and defining $S := [K] \times \mathbb{R}_+$ and $W((k_1, x_1), (k_2, x_2)) := W_{k_1,k_2}(x_1, x_2)$. As compared to the block model of Herlau, Schmidt, and Mørup (2016), this allows for a more complex interaction within and between the blocks. An alternative generalization of the stochastic block model to our setting is to consider infinitely many disjoint intervals $I_k \subset \mathbb{R}_+$ for $k \in \mathbb{N}$, and define $W := \sum_{k_1,k_2 \in \mathbb{N}} p_{k_1,k_2} \mathbf{1}_{I_{k_1} \times I_{k_2}}$ for constants $p_{k_1,k_2} \in [0,1]$. For the block model of Herlau, Schmidt, and Mørup (2016) and our first generalization above (with $S := [K] \times \mathbb{R}_+$), the degree distribution of the vertices within each block will typically be strongly inhomogeneous; by contrast, in our second generalization above (with infinitely many blocks), all vertices within the same block have the same connectivity probabilities, and hence the degree distribution will be more homogeneous.

We can also model sparse graphs with mixed membership structure within our framework. In this case we let $\widetilde{S} \subset [0,1]^K$ be the standard $(K-1)$-simplex, and define $S := \widetilde{S} \times \mathbb{R}_+$. For a vertex with feature $(\widetilde{x}, x) \in \widetilde{S} \times \mathbb{R}_+$ the first coordinate $\widetilde{x} = (\widetilde{x}_1, \dots, \widetilde{x}_K)$ is a vector such that $\widetilde{x}_j$ for $j \in [K]$ describes the proportion of time the vertex is part of community $j \in [K]$, and the second coordinate x describes the role of the vertex within the community; for example, x could be a sociability parameter. For each $k_1, k_2 \in [K]$ let $\mathcal{W}_{k_1,k_2} = (W_{k_1,k_2}, \mathbb{R}_+)$ be a graphon describing the interactions between the communities k_1 and k_2. We define our mixed membership graphon $\mathcal{W} = (W, \mathscr{S})$ by

$$W\big((\widetilde{x}^1, x^1), (\widetilde{x}^2, x^2)\big) := \sum_{k_1,k_2 \in [K]} \widetilde{x}^1_{k_1} \widetilde{x}^2_{k_2} W_{k_1,k_2}(x^1, x^2).$$

Alternatively, we could define $S := \widetilde{S} \times \mathbb{R}_+^K$, which would provide a model where, for example, the sociability of a node varies depending on which community it is part of.

In the classical setting of dense graphs, many papers only consider graphons defined on the unit square, instead of graphons on more general probability spaces. This is justified by the fact that every graphon with a probability space as base space is equivalent to a graphon with base space $[0,1]$. The analogue in our setting would be graphons over $\mathbb{R}_+$ equipped with the Lebesgue measure. As the examples in the preceding paragraphs illustrate, for certain random graph models it is more natural to consider another underlying measure space. For example, each coordinate in some higher-dimensional space may correspond to a particular feature of the vertices, and changing the base space can disrupt certain properties of the graphon, such as smoothness conditions. For this reason we consider graphons defined on general σ-finite measure spaces in this paper. However, we will prove that every graphon is equivalent to a graphon on $\mathbb{R}_+$ equipped with the Borel σ-algebra and Lebesgue measure, in the sense that their cut distance is zero; see Proposition 10 in Section 2.2. As stated before, our results then imply that they correspond to the same random graph model.

The set of $[0,1]$-valued graphons on probability spaces is compact for the cut metric. For the possibly unbounded graphons studied by Borgs, Chayes, Cohn, and Zhao (2014a), which are real-valued and defined on probability spaces, compactness holds if we consider closed subsets of the space of graphons which are *uniformly upper regular* (see Section 2.3 for the definition). In our setting, where we look at graphons over spaces of possibly infinite measure, the analogous regularity condition is *uniform regularity of tails* if we restrict ourselves to, say, $[0,1]$-valued graphons. In particular our results imply that a sequence of simple graphs with uniformly regular tails is subsequentially convergent, and conversely, that every convergent sequence of simple graphs has uniformly regular tails. See Theorem 15 in Section 2.3 and the two corollaries following this theorem.

In the setting of dense graphs, convergence for the cut metric is equivalent to left convergence, meaning that subgraph densities converge. This equivalence does not hold in our setting, or for the unbounded graphons studied by Borgs, Chayes, Cohn, and Zhao (2014a,b); its failure is characteristic of sparse graphs, because deleting even a tiny fraction of the edges in a sparse graph can radically change the densities of larger subgraphs (see the discussion by Borgs, Chayes, Cohn, and Zhao, 2014a, Section 2.9). However, randomly sampled graphs do satisfy a notion of left convergence; see Proposition 30 in Section 2.5.

As previously mentioned, in our forthcoming paper (Borgs, Chayes, Cohn, and Holden, 2017) we will generalize and unify the theories and models presented by Bollobás and Riordan (2009), Borgs, Chayes, Cohn, and Zhao (2014a,b), Caron and Fox (2014), Herlau, Schmidt, and Mørup (2016), and Veitch and Roy (2015). Along with the introduction of a generalized model for sampling graphs and an alternative (and weaker) cut metric, we will prove a number of convergence properties of these graphs. Since the graphs in this paper are obtained as a special case of the graphs in our forthcoming paper, the mentioned convergence results also hold in our setting.

In Section 2 we will state the main results of this paper, which will be proved in the subsequent appendices. In Appendix A we prove that the cut metric $\delta_\square$ is well defined. In Appendix B we prove that any graphon is equivalent to a graphon with underlying measure space $\mathbb{R}_+$. We also prove that under certain conditions on the underlying measure space we may define the cut metric $\delta_\square$ in a number of equivalent ways. In Appendix C, we deal with

some technicalities regarding graph-valued processes. In Appendix D we prove that certain random graph models derived from a graphon $\mathcal{W}$, including the graphon processes defined above, give graphs converging to $\mathcal{W}$ for the cut metric. We also prove that two graphons are equivalent (i.e., they have cut distance zero) iff the corresponding graphon processes are equal in law. In Appendix E we prove that uniform regularity of tails is sufficient to guarantee subsequential metric convergence for a sequence of graphs; conversely, we prove that every convergent sequence of graphs with non-negative edge weights has uniformly regular tails. In Appendix F we prove some basic properties of sequences of graphs which are metric convergent, for example that metric convergence implies unbounded average degree if the number of edges diverge and the graph does not have too many isolated vertices; see Proposition 22 below. We also compare the notion of metric graph convergence in this paper to the one studied by Borgs, Chayes, Cohn, and Zhao (2014a). In Appendix G we prove with reference to the Kallenberg theorem for jointly exchangeable measures that graphon processes for integrable W are uniquely characterized as exchangeable graph processes satisfying uniform tail regularity. We also describe more general families of graphs that may be obtained from the Kallenberg representation theorem if this regularity condition is not imposed. Finally, in Appendix H we prove our results on left convergence of graphon processes.

Remark 1 *After writing a first draft of this work, but a little over a month before completing the paper, we became aware of parallel, independent work by Veitch and Roy (2015), who introduce a closely related model for exchangeable sparse graphs and interpret it with reference to the Kallenberg theorem for exchangeable measures. The random graph model studied by Veitch and Roy (2015) is (up to minor differences) the same as the graphon processes introduced in the current paper. Aside from both introducing this model, the results of the two papers are essentially disjoint. While Veitch and Roy (2015) focus on particular properties of the graphs in a graphon process (in particular, the expected number of edges and vertices, the degree distribution, and the existence of a giant component under certain assumptions on $\mathcal{W}$), our focus is graph convergence, the cut metric, and the question of when two different graphons lead to the same graphon process.*

See also the subsequent paper by Janson (2016) expanding on the results of our paper, characterizing in particular when two graphons are equivalent, and proving additional compactness results for graphons over σ-finite spaces.

2. Definitions and Main Results

We will work mainly with simple graphs, but we will allow the graphs to have weighted vertices and edges for some of our definitions and results. We denote the vertex set of a graph G by $V(G)$ and the edge set of G by $E(G)$. The sets $V(G)$ and $E(G)$ may be infinite, but we require them to be countable. If G is weighted, with edge weights $\beta_{ij}(G)$ and vertex weights $\alpha_i(G)$, we require the vertex weights to be non-negative, and we often (but not always) require that $\|\beta(G)\|_1 := \sum_{i,j\in V(G)} \alpha_i(G)\alpha_j(G)|\beta_{ij}(G)| < \infty$ (note that $\|\beta(G)\|_1$ is defined in such a way that for an unweighted graph, it is equal to $2|E(G)|$, as opposed to the density, which is ill-defined if $|V(G)| = \infty$). We define the edge density of a finite simple graph G to be $\rho(G) := 2|E(G)|/|V(G)|^2$. Letting $\mathbb{N} = \{1, 2, \dots\}$ denote the positive integers, a sequence $(G_n)_{n\in\mathbb{N}}$ of simple, finite graphs will be called sparse if $\rho(G_n) \to 0$ as $n \to \infty$,

and dense if $\liminf_{n\to\infty} \rho(G_n) > 0$. When we consider graph-valued stochastic processes $(G_n)_{n\in\mathbb{N}}$ or $(G_t)_{t\geq 0}$ of simple graphs, we will assume each vertex is labeled by a distinct number in $\mathbb{N}$, so we can view $V(G)$ as a subset of $\mathbb{N}$ and $E(G)$ as a subset of $\mathbb{N}\times\mathbb{N}$. The labels allow us to keep track of individual vertices in the graph over time. In Section 2.4 we define a topology and σ-algebra on the set of such graphs.

2.1 Measure-theoretic Preliminaries

We start by recalling several notions from measure theory.

For two measure spaces $\mathscr{S} = (S, \mathcal{S}, \mu)$ and $\mathscr{S}' = (S', \mathcal{S}', \mu')$, a measurable map $\phi\colon S \to S'$ is called *measure-preserving* if for every $A \in \mathcal{S}'$ we have $\mu(\phi^{-1}(A)) = \mu'(A)$. Two measure spaces $(S, \mathcal{S}, \mu)$ and $(S', \mathcal{S}', \mu')$ are called *isomorphic* if there exists a bimeasurable, bijective, and measure-preserving map $\phi\colon S \to S'$. A *Borel measure space* is defined as a measure space that is isomorphic to a Borel subset of a complete separable metric space equipped with a Borel measure.

Throughout most of this paper, we consider *σ-finite measure spaces*, i.e., spaces $\mathscr{S} = (S, \mathcal{S}, \mu)$ such that S can be written as a countable union of sets $A_i \in \mathcal{S}$ with $\mu(A_i) < \infty$. Recall that a set $A \in \mathcal{S}$ is an *atom* if $\mu(A) > 0$ and if every measurable $B \subseteq A$ satisfies either $\mu(B) = 0$ or $\mu(B) = \mu(A)$. The measure space $\mathscr{S}$ is atomless if it has no atoms. Every atomless σ-finite Borel space of infinite measure is isomorphic to $(\mathbb{R}_+, \mathcal{B}, \lambda)$, where $\mathcal{B}$ is the Borel σ-algebra and λ is Lebesgue measure; for the convenience of the reader, we prove this as Lemma 33 below.

We also need the notion of a coupling, a concept well known for probability spaces: if $(S_i, \mathcal{S}_i, \mu_i)$ is a measure space for $i = 1, 2$ and $\mu_1(S_1) = \mu_2(S_2) \in (0, \infty]$, we say that μ is a *coupling* of μ_1 and μ_2 if μ is a measure on $(S_1 \times S_2, \mathcal{S}_1 \times \mathcal{S}_2)$ with marginals μ_1 and μ_2, i.e., if $\mu(U \times S_2) = \mu_1(U)$ for all $U \in \mathcal{S}_1$ and $\mu(S_1 \times U) = \mu_2(U)$ for all $U \in \mathcal{S}_2$. Note that this definition of coupling is closely related to the definition of coupling of probability measures, which applies when $\mu_1(S_1) = \mu_2(S_2) = 1$. For probability spaces, it is easy to see that every pair of measures has a coupling (for example, the product space of the two probability spaces). We prove the existence of a coupling for σ-finite measure spaces in Appendix A, where this fact is stated as part of a more general lemma, Lemma 34.

Finally, we say that a measure space $\widetilde{\mathscr{S}} = (\widetilde{S}, \widetilde{\mathcal{S}}, \widetilde{\mu})$ *extends* a measure space $\mathscr{S} = (S, \mathcal{S}, \mu)$ if $S \in \widetilde{\mathcal{S}}$, $\mathcal{S} = \{A \cap S : A \in \widetilde{\mathcal{S}}\}$, and $\mu(A) = \widetilde{\mu}(A)$ for all $A \in \mathcal{S}$. We say that $\mathscr{S}$ is a *restriction* of $\widetilde{\mathscr{S}}$, or, if S is specified, *the restriction* of $\widetilde{\mathscr{S}}$ *to* S.

2.2 Graphons and Cut Metric

We will work with the following definition of a graphon.

Definition 2 *A* graphon *is a pair $\mathcal{W} = (W, \mathscr{S})$, where $\mathscr{S} = (S, \mathcal{S}, \mu)$ is a σ-finite measure space satisfying $\mu(S) > 0$ and W is a symmetric real-valued function $W \in L^1(S \times S)$ that is measurable with respect to the product σ-algebra $\mathcal{S} \times \mathcal{S}$ and integrable with respect to $\mu \times \mu$. We say that $\mathcal{W}$ is a* graphon over $\mathscr{S}$.

Remark 3 *Most literature on graphons defines a graphon to be the function W instead of the pair $(W, \mathscr{S})$. We have chosen the above definition since the underlying measure space*

will play an important role. Much literature on graphons requires W to take values in $[0,1]$, and some of our results will also be restricted to this case. The major difference between the above definition and the definition of a graphon in the existing literature, however, is that we allow the graphon to be defined on a measure space of possibly infinite measure, instead of a probability space.[3]

Remark 4 *One may relax the integrability condition for W in the above definition such that the corresponding random graph model (as defined in Definition 25 below) still gives graphs with finitely many vertices and edges for each bounded time. This more general definition is used by Veitch and Roy (2015). We work with the above definition since the majority of the analysis in this paper is related to convergence properties and graph limits, and our definition of the cut metric is most natural for integrable graphons. An exception is the notion of subgraph density convergence in the corresponding random graph model, which we discuss in the more general setting of not necessarily integrable graphons; see Remark 31 below.*

We will mainly study simple graphs in the current paper, in particular, graphs which do not have self-edges. However, the theory can be generalized in a straightforward way to graphs with self-edges, in which case we would also impose an integrability condition for W along its diagonal.

If $\mathcal{W} = (W, (S, \mathcal{B}, \lambda))$, where S is a Borel subset of $\mathbb{R}$, $\mathcal{B}$ is the Borel σ-algebra, and λ is Lebesgue measure, we write $\mathcal{W} = (W, S)$ to simplify notation. For example, we write $\mathcal{W} = (W, \mathbb{R}_+)$ instead of $\mathcal{W} = (W, (\mathbb{R}_+, \mathcal{B}, \lambda))$.

For any measure space $\mathscr{S} = (S, \mathcal{S}, \mu)$ and integrable function $W\colon S \times S \to \mathbb{R}$, define the *cut norm* of W over $\mathscr{S}$ by

$$\|W\|_{\square,S,\mu} := \sup_{U,V\in\mathcal{S}} \left| \int_{U\times V} W(x,y)\, d\mu(x)\, d\mu(y) \right|.$$

If S and/or μ is clear from the context we may write $\|\cdot\|_\square$ or $\|\cdot\|_{\square,\mu}$ to simplify notation.

Given a graphon $\widetilde{\mathcal{W}} = (\widetilde{W}, \widetilde{\mathscr{S}})$ with $\widetilde{\mathscr{S}} = (\widetilde{S}, \widetilde{\mathcal{S}}, \widetilde{\mu})$ and a set $S \in \widetilde{\mathcal{S}}$, we say that $\mathcal{W} = (W, \mathscr{S})$ is *the restriction of $\widetilde{\mathcal{W}}$ to S* if $\mathscr{S}$ is the restriction of $\widetilde{\mathscr{S}}$ to S and $\widetilde{W}|_{S\times S} = W$. We say that $\widetilde{\mathcal{W}} = (\widetilde{W}, \widetilde{\mathscr{S}})$ is *the trivial extension of $\mathcal{W}$ to $\widetilde{\mathscr{S}}$* if $\mathcal{W} = (W, \mathscr{S})$ is the restriction of $\widetilde{\mathcal{W}}$ to S and $\operatorname{supp}(\widetilde{W}) \subseteq S \times S$. For measure spaces $\mathscr{S} = (S, \mathcal{S}, \mu)$ and $\mathscr{S}' = (S', \mathcal{S}', \mu')$, a graphon $\mathcal{W} = (W, \mathscr{S})$, and a measurable map $\phi\colon S' \to S$, we define the graphon $\mathcal{W}^\phi = (W^\phi, \mathscr{S}')$ by $W^\phi(x_1, x_2) := W(\phi(x_1), \phi(x_2))$ for $x_1, x_2 \in S'$. We say that $\mathcal{W}^\phi$ (resp. W^ϕ) is a *pullback* of $\mathcal{W}$ (resp. W) onto $\mathscr{S}'$. Finally, let $\|\cdot\|_1$ denote the L^1 norm.

Definition 5 *For $i = 1, 2$, let $\mathcal{W}_i = (W_i, \mathscr{S}_i)$ with $\mathscr{S}_i = (S_i, \mathcal{S}_i, \mu_i)$ be a graphon.*

3. The term "graphon" was coined by Borgs, Chayes, Lovász, Sós, and Vesztergombi (2008), but the use of this concept in combinatorics goes back to at least Frieze and Kannan (1999), who considered a version of the regularity lemma for functions over $[0,1]^2$. As a limit object for convergent graph sequences it was introduced by Lovász and Szegedy (2006), where it was called a W-function, and graphons over general probability spaces were first studied by Borgs, Chayes, and Lovász (2010) and Janson (2013).

(i) If $\mu_1(S_1) = \mu_2(S_2) \in (0, \infty]$, the cut metric $\delta_\square$ *and* invariant L^1 metric δ_1 *are defined by*

$$\begin{aligned} \delta_\square(\mathcal{W}_1, \mathcal{W}_2) &:= \inf_\mu \|W_1^{\pi_1} - W_2^{\pi_2}\|_{\square, S_1\times S_2, \mu} \quad \text{and} \\ \delta_1(\mathcal{W}_1, \mathcal{W}_2) &:= \inf_\mu \|W_1^{\pi_1} - W_2^{\pi_2}\|_{1, S_1\times S_2, \mu}, \end{aligned} \tag{2}$$

where $\pi_i\colon S_1 \times S_2 \to S_i$ denotes projection for $i = 1, 2$, and we take the infimum over all couplings μ of μ_1 and μ_2.

(ii) If $\mu_1(S_1) \neq \mu_2(S_2)$, let $\widetilde{\mathscr{S}}_i = (\widetilde{S}_i, \widetilde{\mathcal{S}}_i, \widetilde{\mu}_i)$ be a σ-finite measure space extending $\mathscr{S}_i$ for $i = 1, 2$ such that $\widetilde{\mu}_1(\widetilde{S}_1) = \widetilde{\mu}_2(\widetilde{S}_2) \in (0, \infty]$. Let $\widetilde{\mathcal{W}}_i = (\widetilde{W}_i, \widetilde{\mathscr{S}}_i)$ be the trivial extension of $\mathcal{W}_i$ to $\widetilde{\mathscr{S}}_i$, and define

$$\delta_\square(\mathcal{W}_1, \mathcal{W}_2) := \delta_\square(\widetilde{\mathcal{W}}_1, \widetilde{\mathcal{W}}_2) \quad \text{and} \quad \delta_1(\mathcal{W}_1, \mathcal{W}_2) := \delta_1(\widetilde{\mathcal{W}}_1, \widetilde{\mathcal{W}}_2).$$

(iii) We call two graphons $\mathcal{W}_1$ and $\mathcal{W}_2$ equivalent *if $\delta_\square(\mathcal{W}_1, \mathcal{W}_2) = 0$.*

The following proposition will be proved in Appendix A. Recall that a pseudometric on a set S is a function from $S \times S$ to $\mathbb{R}_+$ which satisfies all the requirements of a metric, except that the distance between two different points might be zero.

Proposition 6 *The metrics $\delta_\square$ and δ_1 given in Definition 5 are well defined; in other words, under the assumptions of (i) there exists at least one coupling μ, and under the assumptions of (ii) the definitions of $\delta_\square(\mathcal{W}_1, \mathcal{W}_2)$ and $\delta_1(\mathcal{W}_1, \mathcal{W}_2)$ do not depend on the choice of extensions $\widetilde{\mathscr{S}}_1, \widetilde{\mathscr{S}}_2$. Furthermore, $\delta_\square$ and δ_1 are pseudometrics on the space of graphons.*

An important input to the proof of the proposition (Lemma 42 in Appendix A) is that the $\delta_\square$ (resp. δ_1) distance between two graphons over spaces of equal measure, as defined in Definition 5(i), is invariant under trivial extensions. The lemma is proved by first showing that it holds for step functions (where the proof more or less boils down to an explicit calculation) and then using the fact that every graphon can be approximated by a step function.

We will see in Proposition 48 in Appendix B that under additional assumptions on the underlying measure spaces $\mathscr{S}_1$ and $\mathscr{S}_2$ the cut metric can be defined equivalently in a number of other ways, giving, in particular, the equivalence of the definitions (1) and (2) in the case of two Borel spaces of infinite mass. Similar results hold for the metric δ_1; see Remark 49.

While the two metrics $\delta_\square$ and δ_1 are not equivalent, a fact which is already well known from the theory of graph convergence for dense graphs, it turns out that the statement that two graphons have distance zero in the cut metric is equivalent to the same statement in the invariant L^1 metric. This is the content of our next proposition.

Proposition 7 *Let $\mathcal{W}_1$ and $\mathcal{W}_2$ be graphons. Then $\delta_\square(\mathcal{W}_1, \mathcal{W}_2) = 0$ if and only if $\delta_1(\mathcal{W}_1, \mathcal{W}_2) = 0$.*

The proposition will be proved in Appendix B. (We will actually prove a generalization of this proposition involving an invariant version of the L^p metric; see Proposition 50.) The proof proceeds by first showing (Proposition 51) that if $\delta_\square(\mathcal{W}_1, \mathcal{W}_2) = 0$ for graphons $\mathcal{W}_i = (W_i, \mathscr{S}_i)$ with $\mathscr{S}_i = (S_i, \mathcal{S}_i, \mu_i)$ for $i = 1, 2$, then there exists a particular measure μ on $S_1 \times S_2$ such that $\|W_1^{\pi_1} - W_2^{\pi_2}\|_{\square,\mu} = 0$. Under certain conditions we may assume that μ is a coupling measure, in which case it follows that the infimum in the definition of $\delta_\square$ is a minimum; see Proposition 8 below.

To state our next proposition we define a coupling between two graphons $\mathcal{W}_i = (W_i, \mathscr{S}_i)$ with $\mathscr{S}_i = (S_i, \mathcal{S}_i, \mu_i)$ for $i = 1, 2$ as a pair of graphons $\widetilde{\mathcal{W}}_i$ over a space of the form $\mathscr{S} = (S_1 \times S_2, \mathcal{S}_1 \times \mathcal{S}_2, \mu)$, where μ is a coupling of μ_1 and μ_2 and $\widetilde{\mathcal{W}}_i = W_i^{\pi_i}$, and where as before, π_i denotes the projection from $S_1 \times S_2$ onto S_i for $i = 1, 2$.

Proposition 8 *Let $\mathcal{W}_i$ be graphons over σ-finite Borel spaces $\mathscr{S}_i = (S_i, \mathcal{S}_i, \mu_i)$, and let $\widetilde{S}_i = \{x \in S_i \colon \int |W_i(x, y)| d\mu_i(y) > 0\}$, for $i = 1, 2$. If $\delta_\square(\mathcal{W}_1, \mathcal{W}_2) = 0$, then the restrictions of $\mathcal{W}_1$ and $\mathcal{W}_2$ to $\widetilde{S}_1$ and $\widetilde{S}_2$ can be coupled in such a way that they are equal a.e.*

The proposition will be proved in Appendix B. Note that Janson (2016, Theorem 5.3) independently proved a similar result, building on a previous version of the present paper which did not yet contain Proposition 8. His result states that if the cut distance between two graphons over σ-finite Borel spaces is zero, then there are trivial extensions of these graphons such that the extensions can be coupled so as to be equal almost everywhere. It is easy to see that our result implies his, but we believe that with a little more work, it should be possible to deduce ours from his as well.

Remark 9 *Note that the classical theory of graphons on probability spaces appears as a special case of the above definitions by taking $\mathscr{S}$ to be a probability space. Our definition of the cut metric $\delta_\square$ is equivalent to the standard definition for graphons on probability spaces; see, for example, papers by Borgs, Chayes, Lovász, Sós, and Vesztergombi (2008) and Janson (2013). Note that $\delta_\square$ is not a true metric, only a pseudometric, but we call it a metric to be consistent with existing literature on graphons. However, it is a metric on the set of equivalence classes as derived from the equivalence relation in Definition 5 (iii).*

We work with graphons defined on general σ-finite measure spaces, rather than graphons on $\mathbb{R}_+$, since particular underlying spaces are more natural to consider for certain random graphs or networks. However, the following proposition shows that every graphon is equivalent to a graphon over $\mathbb{R}_+$.

Proposition 10 *For each graphon $\mathcal{W} = (W, \mathscr{S})$ there exists a graphon $\mathcal{W}' = (W', \mathbb{R}_+)$ such that $\delta_\square(\mathcal{W}, \mathcal{W}') = 0$.*

The proof of the proposition follows a similar strategy as the proof of the analogous result for probability spaces by Borgs, Chayes, and Lovász (2010, Theorem 3.2) and Janson (2013, Theorem 7.1), and will be given in Appendix B. The proof uses in particular the result that an atomless σ-finite Borel space is isomorphic to an interval equipped with Lebesgue measure (Lemma 33).

2.3 Graph Convergence

To define graph convergence in the cut metric, one traditionally (Borgs, Chayes, Lovász, Sós, and Vesztergombi, 2006; Lovász and Szegedy, 2006; Borgs, Chayes, Lovász, Sós, and Vesztergombi, 2008) embeds the set of graphs into the set of graphons via the following map. Given any finite weighted graph G we define the *canonical graphon* $\mathcal{W}^G = (W^G, [0, 1])$ as follows. Let $v_1, \dots, v_n$ be an ordering of the vertices of G. For any $v_i \in V(G)$ let $\alpha_i > 0$ denote the weight of v_i, for any $(v_i, v_j) \in E(G)$ let $\beta_{ij} \in \mathbb{R}$ denote the weight of the edge (v_i, v_j), and for $(v_i, v_j) \notin V(G)$ define $\beta_{ij} = 0$. By rescaling the vertex weights if necessary we assume without loss of generality that $\sum_{i=1}^{|V(G)|} \alpha_i = 1$. If G is simple all vertices have weight $|V(G)|^{-1}$, and we define $\beta_{ij} := \mathbf{1}_{(v_i, v_j) \in E(G)}$. Let $I_1, \dots, I_n$ be a partition of $[0, 1]$ into adjacent intervals of lengths $\alpha_1, \dots, \alpha_n$ (say the first one closed, and all others half open), and finally define W^G by

$$W^G(x_1, x_2) = \beta_{ij} \quad \text{if} \quad x_1 \in I_i \text{ and } x_2 \in I_j.$$

Note that W^G depends on the ordering of the vertices, but that different orderings give graphons with cut distance zero. We define a sequence of weighted, finite graphs G_n to be *sparse*[4] if $\|W^{G_n}\|_1 \to 0$ as $n \to \infty$. Note that this generalizes the definition we gave in the very beginning of Section 2 for simple graphs.

A sequence $(G_n)_{n \in \mathbb{N}}$ of graphs is then defined to be *convergent in metric* if $\mathcal{W}^{G_n}$ is a Cauchy sequence in the metric $\delta_\square$, and it is said to be *convergent to a graphon* $\mathcal{W}$ if $\delta_\square(\mathcal{W}^{G_n}, \mathcal{W}) \to 0$. Equivalently, one can define convergence of $(G_n)_{n \in \mathbb{N}}$ by identifying a weighted graph G with the graphon $(\beta(G), \mathscr{S}_G)$, where $\mathscr{S}_G$ consists of the vertex set $V(G)$ equipped with the probability measure given by the weights α_i (or the uniform measure if G has no vertex weights), and $\beta(G)$ is the function that maps $(i, j) \in V(G) \times V(G)$ to $\beta_{ij}(G)$.

In the classical theory of graph convergence a sequence of sparse graphs converges to the trivial graphon with $W \equiv 0$. This follows immediately from the fact that $\delta_\square(\mathcal{W}^{G_n}, 0) \le \|W^{G_n}\|_1 \to 0$ for sparse graphs. To address this problem, Bollobás and Riordan (2009) and Borgs, Chayes, Cohn, and Zhao (2014a) considered the sequence of reweighted graphons $(\mathcal{W}^{G_n, r})_{n \in \mathbb{N}}$, where $\mathcal{W}^{G,r} := (W^{G,r}, [0, 1])$ with $W^{G,r} := \frac{1}{\|W^G\|_1} W^G$ for any graph G, and defined $(G_n)_{n \in \mathbb{N}}$ to be convergent iff $(\mathcal{W}^{G_n, r})_{n \in \mathbb{N}}$ is convergent. The theory developed in the current paper considers a different rescaling, namely a rescaling of the *arguments* of the function W^G, which, as explained after Definition 11 below, is equivalent to *rescaling the measure* of the underlying measurable space.

We define the *stretched canonical graphon* $\mathcal{W}^{G,s}$ to be identical to $\mathcal{W}^G$ except that we "stretch" the function W^G to a function $W^{G,s}$ such that $\|W^{G,s}\|_1 = 1$. More precisely, $\mathcal{W}^{G,s} := (W^{G,s}, \mathbb{R}_+)$, where

$$W^{G,s}(x_1, x_2) := \begin{cases} W^G\left(\|W^G\|_1^{1/2} x_1, \|W^G\|_1^{1/2} x_2\right) & \text{if } 0 \le x_1, x_2 \le \|W^G\|_1^{-1/2}, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

4. Note that in the case of weighted graphs there are multiple natural definitions of what it means for a sequence of graphs to be sparse or dense. Instead of considering the L^1 norm as in our definition, one may for example consider the fraction of edges with non-zero weight, either weighted by the vertex weights or not. In the current paper we do not define what it means for a sequence of weighted graphs to be dense, since it is not immediate which definition is most natural, and since the focus of this paper is sparse graphs.

Note that in the case of a simple graph G, each node in $V(G)$ corresponds to an interval of length $1/|V(G)|$ in the canonical graphon $\mathcal{W}^G$, while it corresponds to an interval of length $1/\sqrt{2|E(G)|}$ in the stretched canonical graphon.

It will sometimes be convenient to define stretched canonical graphons for graphs with infinitely many vertices (but finitely[5] many edges). Our definition of W^G makes no sense for simple graphs with infinitely many vertices, because they cannot all be crammed into the unit interval. Instead, given a finite or countably infinite graph G with vertex weights $(\alpha_i)_{i\in V(G)}$ which do not necessarily sum to 1 (and may even sum to ∞), we define a graphon $\widetilde{\mathcal{W}}^G = (\widetilde{W}^G, \mathbb{R}_+)$ by setting $\widetilde{W}^G(x,y) = \beta_{ij}(G)$ if $(x,y) \in I_i \times I_j$, and $\widetilde{W}^G(x,y) = 0$ if there exist no such pair $(i,j) \in V(G)\times V(G)$, with I_i being the interval $[a_{i-1}, a_i)$ where we assume the vertices of G have been labeled $1, 2, \ldots$, and $a_i = \sum_{1\le k\le i} \alpha_k$ for $i = 0, 1, \ldots$. The stretched canonical graphon will then be defined as the graphon $\mathcal{W}^{G,s} := (W^{G,s}, \mathbb{R}_+)$ with

$$W^{G,s}(x_1,x_2) := \widetilde{W}^G\left(\|\widetilde{W}^G\|_1^{1/2} x_1, \|\widetilde{W}^G\|_1^{1/2} x_2\right),$$

a definition which can easily be seen to be equivalent to the previous one if G is a finite graph.

Alternatively, one can define a stretched graphon G^s as a graphon over $V(G)$ equipped with the measure $\widehat{\mu}_G$, where

$$\widehat{\mu}_G(A) = \frac{1}{\sqrt{\|\beta(G)\|_1}} \sum_{i\in A} \alpha_i$$

for any $A \subseteq V(G)$. In the case where $\sum_i \alpha_i < \infty$, this graphon is obtained from the graphon representing G by rescaling the probability measure

$$\mu_G(A) = \frac{1}{\sum_{i\in V(G)} \alpha_i} \sum_{i\in A} \alpha_i$$

to the measure $\widehat{\mu}_G$, while the function $\beta(G)\colon V(G)\times V(G) \to \mathbb{R}$ with $(i,j) \mapsto \beta_{ij}(G)$ is left untouched.

Note that any graphon with underlying measure space $\mathbb{R}_+$ can be "stretched" in the same way as W^G; in other words, given any graphon $\mathcal{W} = (W, \mathbb{R}_+)$ we may define a graphon $(W^\phi, \mathbb{R}_+)$, where $\phi\colon \mathbb{R}_+ \to \mathbb{R}_+$ is defined to be the linear map such that $\|W^\phi\|_1 = 1$, except when $\|W\|_1 = 0$, in which case we define the stretched graphon to be 0. But for graphons over general measure spaces, this rescaling is ill-defined. Instead, we consider a different, but related, notion of rescaling, by rescaling the measure of the underlying space, a notion which is the direct generalization of our definition of the stretched graphon G^s.

Definition 11 *(i) For two graphons $\mathcal{W}_i = (W_i, \mathscr{S}_i)$ with $\mathscr{S}_i = (S_i, \mathcal{S}_i, \mu_i)$ for $i = 1, 2$, define the* stretched cut metric $\delta^s_\square$ *by*

$$\delta^s_\square(\mathcal{W}_1, \mathcal{W}_2) := \delta_\square(\widehat{\mathcal{W}}_1, \widehat{\mathcal{W}}_2),$$

where $\widehat{\mathcal{W}}_i := (W_i, \widehat{\mathscr{S}_i})$ with $\widehat{\mathscr{S}_i} := (S_i, \mathcal{S}_i, \widehat{\mu}_i)$ and $\widehat{\mu}_i := \|W_i\|_1^{-1/2}\mu_i$. (In the particular case where $\|W_i\|_1 = 0$, we define $\widehat{\mathcal{W}}_i := (0, \mathscr{S}_i)$.) Identifying G with the graphon $\widetilde{\mathcal{W}}^G$ introduced above, this also defines the stretched distance between two graphs, or a graph and a graphon.

(ii) A sequence of graphs $(G_n)_{n\in\mathbb{N}}$ or graphons $(\mathcal{W}_n)_{n\in\mathbb{N}}$ is called convergent in the stretched cut metric *if they form a Cauchy sequence for this metric; they are called* convergent to a graphon $\mathcal{W}$ for the stretched cut metric *if $\delta^s_\square(G_n, \mathcal{W}) \to 0$ or $\delta^s_\square(\mathcal{W}_n, \mathcal{W}) \to 0$, respectively.*

5. More generally, in the setting of weighted graphs, we can allow for infinitely many edges as long as $\|\beta(G)\|_1 < \infty$.

Note that for the case of graphons over $\mathbb{R}_+$, the above notion of convergence is equivalent to the one involving the stretched graphons $\mathcal{W}_i^s = (W_i^s, \mathbb{R}_+)$ of $\mathcal{W}_i$ defined by

$$W_i^s(x_1,x_2) := W_i\left(\|W_i\|_1^{1/2} x_1, \|W_i\|_1^{1/2} x_2\right).$$

To see this, just note that by the obvious coupling between λ and $\widehat{\mu}_i$, where in this case $\widehat{\mu}_i$ is a constant multiple of Lebesgue measure, we have $\delta_\square(\mathcal{W}_i^s, \widehat{\mathcal{W}}_i) = 0$, and hence $\delta^s_\square(\mathcal{W}_1, \mathcal{W}_2) = \delta_\square(\mathcal{W}_1^s, \mathcal{W}_2^s)$. As a consequence, we have in particular that $\delta^s_\square(G, G') = \delta_\square(G^s, (G')^s) = \delta_\square(\mathcal{W}^{G,s}, \mathcal{W}^{G',s})$ for any two graphs G and G'. Note also that the stretched cut metric does not distinguish two graphs obtained from each other by deleting isolated vertices, in the sense that

$$\delta^s_\square(G, G') = 0 \tag{3}$$

whenever G is obtained from G' by removing a set of vertices that have degree 0 in G'.

The following basic example illustrates the difference between the notions of convergence in the classical theory of graphons, the approach for sparse graphs taken by Bollobás and Riordan (2009) and Borgs, Chayes, Cohn, and Zhao (2014a), and the approach of the current paper. Proposition 20 below makes this comparison more general.

Example 12 *Let $\alpha \in (0,1)$. For any $n \in \mathbb{N}$ let G_n be an Erdős-Rényi graph on n vertices with parameter $n^{\alpha-1}$; i.e., each two vertices of the graph are connected independently with probability $n^{\alpha-1}$. Let $\widetilde{G}_n$ be a simple graph on n vertices, such that $\lfloor n^{(1+\alpha)/2}\rfloor$ vertices form a complete subgraph, and $n - \lfloor n^{(1+\alpha)/2}\rfloor$ vertices are isolated. Both graph sequences are sparse, and hence their canonical graphons converge to the trivial graphon for which $W \equiv 0$, i.e., $\delta_\square(\mathcal{W}^{G_n}, 0), \delta_\square(\mathcal{W}^{\widetilde{G}_n}, 0) \to 0$, where we let 0 denote the mentioned trivial graphon. The sequence $(G_n)_{n\in\mathbb{N}}$ converges to $\mathcal{W}_1 := (\mathbf{1}_{[0,1]^2}, [0,1])$ with the notion of convergence introduced by Bollobás and Riordan (2009) and Borgs, Chayes, Cohn, and Zhao (2014a), but does not converge for $\delta^s_\square$. The sequence $(\widetilde{G}_n)_{n\in\mathbb{N}}$ converges to $\mathcal{W}_1$ for the stretched cut metric, i.e., $\delta^s_\square(\widetilde{G}_n, \mathcal{W}_1) = \delta_\square(\mathcal{W}^{\widetilde{G}_n,s}, \mathcal{W}_1) \to 0$, but it does not converge with the notion of convergence studied by Bollobás and Riordan (2009) and Borgs, Chayes, Cohn, and Zhao (2014a).*

The sequence $(\widetilde{G}_n)_{n\in\mathbb{N}}$ defined above illustrates one of our motivations to introduce the stretched cut metric. One might argue that this sequence of graphs should converge to the same limit as a sequence of complete graphs; however, earlier theories for graph convergence are too sensitive to isolated vertices or vertices with very low degree to accommodate this.

The space of all $[0,1]$-valued graphons over $[0,1]$ is compact under the cut metric (Lovász and Szegedy, 2007). This implies that every sequence of simple graphs is subsequentially

convergent to some graphon under $\delta_\square$, when we identify a graph G with its canonical graphon $\mathcal{W}^G$. Our generalized definition of a graphon, along with the introduction of the stretched canonical graphon $\mathcal{W}^{G,s}$ and the stretched cut metric $\delta^s_\square$, raises the question of whether a similar result holds in this setting. We will see in Theorem 15 and Corollary 17 below that the answer is yes, provided we restrict ourselves to uniformly bounded graphons and impose a suitable regularity condition; see Definition 13. The sequence $(G_n)_{n\in\mathbb{N}}$ in Example 12 illustrates that we may not have subsequential convergence when this regularity condition is not satisfied.

Definition 13 *Let $\widetilde{\mathscr{W}}$ be a set of uniformly bounded graphons. We say that $\widetilde{\mathscr{W}}$ has* uniformly regular tails *if for every $\varepsilon > 0$ we can find an $M > 0$ such that for every $\mathcal{W} = (W, \mathscr{S}) \in \widetilde{\mathscr{W}}$ with $\mathscr{S} = (S, \mathcal{S}, \mu)$, there exists $U \in \mathcal{S}$ such that $\|W - W\mathbf{1}_{U\times U}\|_1 < \varepsilon$ and $\mu(U) \le M$. A set $\mathcal{G}$ of graphs has* uniformly regular tails *if $\|\beta(G)\|_1 < \infty$ for all $G \in \mathcal{G}$ and the corresponding set of stretched canonical graphons $\{\mathcal{W}^{G,s} : G \in \mathcal{G}\}$ has uniformly regular tails.*

Remark 14 *It is immediate from the definition that a set of simple graphs $\mathcal{G}$ has uniformly regular tails if and only if for each $\varepsilon > 0$ we can find $M > 0$ such that the following holds. For all $G \in \mathcal{G}$, assuming the vertices of G are labeled by degree (from largest to smallest) with ties resolved in an arbitrary way,*

$$\sum_{i\le\lceil M\sqrt{|E(G)|}\rceil} \deg(i; G) \le \varepsilon|E(G)|.$$

In Lemma 59 in Appendix F we will prove that for a set of graphs with uniformly regular tails we may assume the sets U in the above definition correspond to sets of vertices. Note that if a collection $\widetilde{\mathscr{W}}$ of graphons has uniformly regular tails, then every collection of graphons which can be derived from $\widetilde{\mathscr{W}}$ by adding a finite number of the graphons to $\widetilde{\mathscr{W}}$ will still have uniformly regular tails. In other words, if $\widetilde{\mathscr{W}}, M, \varepsilon$ are such that the conditions of Definition 13 are satisfied for all but finitely many graphons in $\widetilde{\mathscr{W}}$, then the collection $\widetilde{\mathscr{W}}$ has uniformly regular tails.

The following theorem shows that a necessary and sufficient condition for subsequential convergence is the existence of a subsequence with uniformly regular tails.

Theorem 15 *Every sequence $(\mathcal{W}_n)_{n\in\mathbb{N}}$ of uniformly bounded graphons with uniformly regular tails converges subsequentially to some graphon $\mathcal{W}$ for the cut metric $\delta_\square$. Moreover, if $\mathcal{W}_n$ is non-negative then every $\delta_\square$-Cauchy sequence of uniformly bounded, non-negative graphons has uniformly regular tails.*

The proof of the theorem will be given in Appendix E. The most challenging part of the proof is to show that uniform regularity of tails implies subsequential convergence. We prove in Lemma 58 that the property of having uniformly regular tails is invariant under certain operations, which allows us to prove subsequential convergence similarly as in the setting of dense graphs, i.e., by approximating the graphons by step functions and using a martingale convergence theorem.

Two immediate corollaries of Theorem 15 are the following results.

Corollary 16 *The set of all $[0,1]$-valued graphons is complete for the cut metric $\delta_\square$, and hence also for $\delta^s_\square$.*

Corollary 17 *Let $(G_n)_{n\in\mathbb{N}}$ be a sequence of finite graphs with non-negative, uniformly bounded edge weights such that $|E(G_n)| < \infty$ for each $n \in \mathbb{N}$. Then the following hold:*

(i) If $(G_n)_{n\in\mathbb{N}}$ has uniformly regular tails, then $(G_n)_{n\in\mathbb{N}}$ has a subsequence that converges to some graphon $\mathcal{W}$ in the stretched cut metric.

(ii) If $(G_n)_{n\in\mathbb{N}}$ is a $\delta^s_\square$-Cauchy sequence, then it has uniformly regular tails.

(iii) If $(G_n)_{n\in\mathbb{N}}$ is a $\delta^s_\square$-Cauchy sequence, then it converges to some graphon $\mathcal{W}$ in the stretched cut metric.

The former of the above corollaries makes two assumptions: (i) the graphons are uniformly bounded, and (ii) the graphons are non-negative. We remark that both of these conditions are necessary.

Remark 18 *The set of all $\mathbb{R}_+$-valued graphons is not complete for the cut metric $\delta_\square$; see for example the argument of Borgs, Chayes, Cohn, and Zhao (2014a, Proposition 2.12(b)) for a counterexample. The set of all $[-1,1]$-valued graphons is also not complete, as the following example suggested to us by Svante Janson illustrates. For each $n \in \mathbb{N}$ let $\mathcal{V}_n = (V_n, \mathbb{R}_+)$ be a $\{-1,1\}$-valued graphon supported in $[n-1,n]^2$ satisfying $\|V_n\|_\square < 2^{-n}$ and $\|V_n\|_1 = 1$, by defining $\mathcal{V}_n$ to be an appropriately rescaled version of a graphon for a sufficiently large Erdős-Rényi random graph with edge density $1/2$. Define $\mathcal{W}_n = (W_n, \mathbb{R}_+)$ by $W_n := \sum_{k=1}^n V_k$, and assume there is a graphon $\mathcal{W} = (W, \mathbb{R}_+)$ such that $\lim_{n\to\infty} \delta_\square(\mathcal{W}, \mathcal{W}_n) = 0$. Then we can find a sequence of measure-preserving transformations $(\phi_n)_{n\in\mathbb{N}}$ with $\phi_n\colon \mathbb{R}_+ \to \mathbb{R}_+$, such that $\lim_{n\to\infty} \|W^{\phi_n} - W_n\|_\square = 0$. This implies that $\lim_{n\to\infty} \|W^{\phi_n}\mathbf{1}_{[k-1,k]^2} - V_k\|_\square = 0$ for each $k \in \mathbb{N}$. Since V_k is a graphon associated with an Erdős-Rényi random graph it is a step graphon. For any intervals $I, J \subseteq \mathbb{R}_+$ such that $V_k|_{I\times J} = 1$ or $V_k|_{I\times J} = -1$ we have $\lim_{n\to\infty} \|(W^{\phi_n}\mathbf{1}_{[k-1,k]^2} - V_k)\mathbf{1}_{I\times J}\|_\square = 0$, so since W takes values in $[-1,1]$ we have $\lim_{n\to\infty} \|W^{\phi_n}\|_{L^1(I\times J)} = \|V_k\|_{L^1(I\times J)}$. Since $\|V_k\|_{L^1([k-1,k]^2)} = 1$ this implies that $\lim_{n\to\infty} \|W^{\phi_n}\|_{L^1([k-1,k]^2)} = 1$. We have obtained a contradiction to the assumption that $\mathcal{W}$ is a graphon, since for each $n \in \mathbb{N}$ we have $\|W\|_1 \ge \sum_{k=1}^\infty \|W^{\phi_n}\|_{L^1([k-1,k]^2)}$.*

Remark 19 *For comparison, Lovász and Szegedy (2007, Theorem 5.1) proved that $[0,1]$-valued graphons on the probability space $[0,1]$ (and hence any probability space) form a compact metric space under $\delta_\square$. Compactness fails in our setting, because convergence requires uniformly regular tails, but completeness still holds.*

Our next result compares the theory of graph convergence developed by Borgs, Chayes, Cohn, and Zhao (2014a,b) with the theory developed in this paper. First we will define the *rescaled cut metric* $\delta^r_\square$. A sequence of graphs is convergent in the sense considered by Borgs, Chayes, Cohn, and Zhao (2014a,b) iff it converges for this metric. For two graphons $\mathcal{W}_1 = (W_1, \mathscr{S}_1)$ and $\mathcal{W}_2 = (W_2, \mathscr{S}_2)$, where $\mathscr{S}_1$ and $\mathscr{S}_2$ are measure spaces of the same total measure, define $\widetilde{W}_1 := \|W_1\|_1^{-1} W_1$, $\widetilde{W}_2 := \|W_2\|_1^{-1} W_2$, and

$$\delta^r_\square(\mathcal{W}_1, \mathcal{W}_2) := \inf_\mu \|\widetilde{W}_1^{\pi_1} - \widetilde{W}_2^{\pi_2}\|_{\square, S_1\times S_2, \mu},$$

where we take the infimum over all measures μ on $S_1 \times S_2$ with marginals μ_1 and μ_2, respectively. For any graphs G and G' we let $\mathcal{W}^G$ and $\mathcal{W}^{G'}$, respectively, denote the canonical graphons associated with G and G', and for any graphon $\mathcal{W}$ we define

$$\delta^r_\square(G,\mathcal{W}) := \delta^r_\square(\mathcal{W}^G,\mathcal{W}), \qquad \delta^r_\square(G,G') := \delta^r_\square(\mathcal{W}^G,\mathcal{W}^{G'}).$$

For the notion of convergence studied by Borgs, Chayes, Cohn, and Zhao (2014a), *uniform upper regularity* plays a similar role to that of regularity of tails in the current paper. More precisely, subsequential uniform upper regularity for a sequence of graphs or graphons defined over a probability space is equivalent to subsequential convergence to a graphon for the metric $\delta^r_\square$ (Borgs, Chayes, Cohn, and Zhao, 2014a, Appendix C). The primary conceptual difference is that the analogue of Corollary 16 does not hold in the theory studied by Borgs, Chayes, Cohn, and Zhao (2014a).

We will now define what it means for a sequence of graphs or graphons to be uniformly upper regular. A *partition* of a measurable space $(S,\mathcal{S})$ is a finite collection $\mathcal{P}$ of disjoint elements of $\mathcal{S}$ with union S. For any graphon $\mathcal{W} = (W,\mathscr{S})$ with $\mathscr{S} = (S,\mathcal{S},\mu)$ and a partition $\mathcal{P}$ of $(S,\mathcal{S})$ into parts of nonzero measure, define $\mathcal{W}_\mathcal{P}$ by averaging W over the partitions. More precisely, if $\mathcal{P} = \{I_i : i = 1,\dots,m\}$ for some $m \in \mathbb{N}$, define $\mathcal{W}_\mathcal{P} := ((W)_\mathcal{P},\mathscr{S})$, where

$$(W_\mathcal{P})(x_1,x_2) := \frac{1}{\mu(I_i)\mu(I_j)} \int_{I_i\times I_j} W(x_1',x_2')\,dx_1'\,dx_2' \qquad \text{if } (x_1,x_2) \in I_i \times I_j.$$

A sequence $(\mathcal{W}_n)_{n\in\mathbb{N}}$ of graphons $\mathcal{W}_n = (W_n,\mathscr{S}_n)$ over probability spaces $\mathscr{S}_n = (S_n,\mathcal{S}_n,\mu_n)$ is *uniformly upper regular* if there exists a function $K\colon (0,\infty) \to (0,\infty)$ and a sequence $\{\eta_n\}_{n\in\mathbb{N}}$ of positive real numbers converging to zero, such that for every $\varepsilon > 0$, $n \in \mathbb{N}$, and partition $\mathcal{P}$ of S_n such that the μ_n-measure of each part is at least η_n, we have

$$\|(W_n)_\mathcal{P}\mathbf{1}_{|(W_n)_\mathcal{P}|\ge K(\varepsilon)}\|_1 \le \varepsilon.$$

For any graph G define the *rescaled canonical graphon* $\mathcal{W}^{G,r} = (W^{G,r},[0,1])$ of G to be equal to the canonical graphon $\mathcal{W}^G$ of G, except that we rescale the graphon such that $\|W^{G,r}\|_1 = 1$. More precisely, we define $\mathcal{W}^{G,r} := (W^{G,r},[0,1])$ with $W^{G,r} := \|W^G\|_1^{-1}W^G$. We say that a sequence of graphs $(G_n)_{n\in\mathbb{N}}$ is uniformly upper regular if $(\mathcal{W}^{G_n,r})_{n\in\mathbb{N}}$ is uniformly upper regular, where we only consider partitions $\mathcal{P}$ corresponding to partitions of $V(G_n)$, and we require every vertex of G_n to have weight less than a fraction η_n of the total weight of $V(G_n)$.

The following proposition, which will be proved in Appendix F, illustrates the very different nature of the sparse graphs studied by Borgs, Chayes, Cohn, and Zhao (2014a,b) and the graphs studied in this paper.

Proposition 20 *Let $(G_n)_{n\in\mathbb{N}}$ be a sequence of simple graphs satisfying $|V(G_n)| < \infty$ for each $n \in \mathbb{N}$.*

(i) If $(G_n)_{n\in\mathbb{N}}$ is sparse it cannot both be uniformly upper regular and have uniformly regular tails; hence it cannot converge for both metrics $\delta^s_\square$ and $\delta^r_\square$ if it is sparse.

(ii) Assume $(G_n)_{n\in\mathbb{N}}$ is dense and has convergent edge density. Then $(G_n)_{n\in\mathbb{N}}$ is a Cauchy sequence for $\delta^s_\square$ iff it is a Cauchy sequence for $\delta^r_\square$. If we do not assume convergence of the edge density, being a Cauchy sequence for $\delta^r_\square$ (resp. $\delta^s_\square$) does not imply being a Cauchy sequence for $\delta^s_\square$ (resp. $\delta^r_\square$).

Many natural properties of graphons are continuous under the cut metric, for example certain properties related to the degrees of the vertices. For graphons defined on probability spaces it was shown by Borgs, Chayes, Cohn, and Ganguly (2015, Section 2.6) that the appropriately normalized degree distribution is continuous under the cut metric. A similar result holds in our setting, but the normalization is slightly different: instead of the proportion of vertices whose degrees are at least λ times the average degree, we will consider a normalization in terms of the square root of the number of edges. Given a graph G and vertex $v \in V(G)$, let $d_G(v)$ denote the degree of v, and given a graphon $\mathcal{W} = (W,(S,\mathcal{S},\mu))$, define the analogous function $D_W\colon S \to \mathbb{R}$ by

$$D_W(x) = \int_S W(x,y)\,d\mu(y).$$

The following proposition is an immediate consequence of Lemma 45 in Appendix A, which compares the functions D_{W_1} and D_{W_2} for graphons that are close in the cut metric.

Proposition 21 *Let $\mathcal{W}_n = (W_n,(S_n,\mathcal{S}_n,\mu_n))$ be a sequence of graphons that converge to a graphon $\mathcal{W} = (W,(S,\mathcal{S},\mu))$ in the cut metric $\delta_\square$, and let $\lambda > 0$ be a point where the function $\lambda \mapsto \mu(\{D_W > \lambda\})$ is continuous. Then $\mu_n(\{D_{W_n} > \lambda\}) \to \mu(\{D_W > \lambda\})$. In particular,*

$$\frac{1}{\sqrt{2|E(G_n)|}}\left|\left\{v \in V(G_n)\colon d_{G_n}(v) > \lambda\sqrt{2|E(G_n)|}\right\}\right| \to \mu(\{D_{W^s} > \lambda\})$$

whenever G_n is a sequence of finite simple graphs that converge to a graphon $\mathcal{W}^s$ in the stretched cut metric and $\mu(\{D_{W^s} > \lambda\})$ is continuous at λ.

Our final result in this section, which will be proved in Appendix F, is that graphs which converge for the stretched cut metric have unbounded average degree under certain assumptions, a result which also holds for graphs that converge under the rescaled cut metric (Borgs, Chayes, Cohn, and Zhao, 2014a, Proposition C.15).

Proposition 22 *Let $(G_n)_{n\in\mathbb{N}}$ be a sequence of finite simple graphs such that the number of isolated vertices in G_n is $o(|E(G_n)|)$ and such that $\lim_{n\to\infty}|E(G_n)| = \infty$. If there is a graphon $\mathcal{W}$ such that $\lim_{n\to\infty}\delta^s_\square(G_n,\mathcal{W}) = 0$, then $(G_n)_{n\in\mathbb{N}}$ has unbounded average degree.*

The proof of the proposition proceeds by showing that graphs with bounded average degree and a divergent number of edges cannot have uniformly regular tails.

2.4 Random Graph Models

In this section we will present two random graph models associated with a given $[0,1]$-valued graphon $\mathcal{W} = (W,\mathscr{S})$ with $\mathscr{S} = (S,\mathcal{S},\mu)$.

Before defining these models, we introduce some notation. In particular, we will introduce the notion of a graph process, defined as a stochastic process taking values in the set of labeled graphs with finitely many edges and countably many vertices, equipped with a suitable σ-algebra. Explicitly, consider a family of graphs $\mathcal{G} = (G_t)_{t\ge 0}$, where the vertices have labels in $\mathbb{N}$. Let $\mathbb{G}$ denote the set of simple graphs with finitely many edges and countably many vertices, such that the vertices have distinct labels in $\mathbb{N}$. Observe that a

graph in this space can be identified with an element of $\{0,1\}^{\mathbb{N}\cup\binom{\mathbb{N}}{2}}$. We equip $\{0,1\}^{\mathbb{N}\cup\binom{\mathbb{N}}{2}}$ with the product topology and $\mathbb{G}$ with the subspace topology $\mathbb{T}$. Recall that a stochastic process is càdlàg if it is right-continuous with a left limit at every point. Observe that the topological space $(\mathbb{G},\mathbb{T})$ is Hausdorff, which implies that a convergent sequence of graphs has a unique limit. The σ-algebra on $\mathbb{G}$ is the Borel σ-algebra induced by $\mathbb{T}$.

Definition 23 *A* graph process *is a càdlàg stochastic process $\mathcal{G} = (G_t)_{t\geq 0}$ taking values in the space of graphs $\mathbb{G}$ equipped with the topology $\mathbb{T}$ defined above. The process is called* projective *if for all $s < t$, G_s is an induced subgraph of G_t.*

We now define the graphon process already described informally in the introduction. Sample a Poisson random measure $\mathcal{V}$ on $\mathbb{R}_+ \times S$ with intensity given by $\lambda \times \mu$ (see the book of Çınlar, 2011, Chapter VI, Theorem 2.15), and identify $\mathcal{V}$ with the collection of points (t,x) at which $\mathcal{V}$ has a point mass.[6] Let $\widetilde{G}$ be a graph with vertex set $\mathcal{V}$, such that for each pair of vertices $v_1 = (t_1, x_1)$ and $v_2 = (t_2, x_2)$ with $v_1 \neq v_2$, there is an edge between v_1 and v_2 with probability $W(x_1, x_2)$, independently for any two v_1, v_2. Note that $\widetilde{G}$ is a graph with countably infinitely many vertices, and that the set of edges is also countably infinite except if W is equal to 0 almost everywhere. For each $t \geq 0$ let $\widetilde{G}_t$ be the induced subgraph of $\widetilde{G}$ consisting only of the vertices (t', x) for which $t' \leq t$. Finally define G_t to be the induced subgraph of $\widetilde{G}_t$ consisting only of the vertices having degree at least one. While $\widetilde{G}_t$ is a graph on infinitely many vertices if $\mu(S) = \infty$, it has finitely many edges almost surely, and thus G_t is a graph with finitely many vertices. We view the graphs G_t and $\widetilde{G}_t$ as elements of $\mathbb{G}$ by enumerating the points of $\mathcal{V}$ in an arbitrary but fixed way.

When $\mu(S) < \infty$ the set of graphs $\{\widetilde{G}_t : t \geq 0\}$ considered above is identical in law to a sequence of $\mathcal{W}$-random graphs as defined by Lovász and Szegedy (2006) for graphons over $[0,1]$ and, for example, by Bollobás, Janson, and Riordan (2007) for graphons over general probability spaces. More precisely, defining a stopping time t_n as the first time when $|V(\widetilde{G}_t)| = n$ and relabeling the vertices in $V(\widetilde{G}_{t_n})$ by labels in $[n]$, we have that the sequence $\{\widetilde{G}_{t_n} : n \in \mathbb{N}\}$ has the same distribution as the sequence of random graphs generated from $\mathcal{W}$, except for the fact that μ should be replaced by the probability measure $\widetilde{\mu} = \frac{1}{\mu(S)}\mu$, a fact which follows immediately from the observation that a Poisson process with intensity $t\mu$ conditioned on having n points is just a distribution of n points chosen i.i.d. from the distribution $\widetilde{\mu}$. In the case when $\mu(S) = \infty$ it is primarily the graphs G_t (rather than $\widetilde{G}_t$) which are of interest for applications, since the graphs $\widetilde{G}_t$ have infinitely many (isolated) vertices. But from a mathematical point of view, both turn out to be useful.

Definition 24 *Two graph processes $(G_t^1)_{t\geq 0}$ and $(G_t^2)_{t\geq 0}$ are said to be* equal up to relabeling of the vertices *if there is a bijection $\phi\colon \bigcup_{t\geq 0} V(G_t^1) \to \bigcup_{t\geq 0} V(G_t^2)$ such that $\phi(G_t^1) = G_t^2$ for all $t \geq 0$, where $\phi(G_t^1)$ is the graph whose vertex and edge sets are $\{\phi(i)\}_{i\in V(G_t)}$ and $\{\phi(i)\phi(j)\}_{ij\in E(G_t)}$, respectively. Two graph processes $(G_t^1)_{t\geq 0}$ and $(G_t^2)_{t\geq 0}$ are said to be* equal in law up to relabeling of the vertices *if they can be coupled in such a way that a.s., the two families are equal up to relabeling of the vertices.*

6. We see that this collection of points exists by observing that for any measurable set $A \subset \mathbb{R}_+ \times S$ of finite measure, we may sample $\{(t,x) \in \mathcal{V} \cap A\}$ by first sampling the total number of points $N_A \in \mathbb{N} \cup \{0\}$ in the set (which is a Poisson random variable with parameter $\mu(A)$), and then sampling N_A points independently at random from A using the measure $\mu|_A$ renormalized to be a probability measure. Note that our Poisson random measure is not necessarily a random counting measure as defined for example by Çınlar (2011), since in general, not all singletons (t,x) are measurable, unless we assume that the singletons $\{x\}$ in S are measurable.

Note that in order for the notion of "equal in law up to relabeling of the vertices" to be well defined, one needs to show that the event that two graph processes $(G_t)_{t\geq 0}$ and $(\widetilde{G}_t)_{t\geq 0}$ are equal up to relabeling is measurable. The proof of this fact is somewhat technical and will be given in Appendix C.

Definition 25 *Let $\mathcal{W} = (W, \mathscr{S})$ be a $[0,1]$-valued graphon. Define $\widetilde{\mathcal{G}}(\mathcal{W}) = (\widetilde{G}_t(\mathcal{W}))_{t\geq 0}$ (resp. $\mathcal{G}(\mathcal{W}) = (G_t(\mathcal{W}))_{t\geq 0}$) to be a random family of graphs with the same law as the graphs $(\widetilde{G}_t)_{t\geq 0}$ (resp. $(G_t)_{t\geq 0}$) defined above.*

(i) A random family of simple graphs is called a graphon process without isolated vertices *generated by $\mathcal{W}$ if it has the same law as $\mathcal{G}(\mathcal{W})$ up to relabeling of the vertices, and it is called a* graphon process with isolated vertices *generated by $\mathcal{W}$ if it has the same law as $\widetilde{\mathcal{G}}(\mathcal{W})$ up to relabeling of the vertices.*

(ii) A random family $\widetilde{\mathcal{G}} = (\widetilde{G}_t)_{t\geq 0}$ of simple graphs is called a graphon process *if there exists a graphon $\mathcal{W}$ such that after removal of all isolated vertices, $\widetilde{\mathcal{G}}$ has the same law as $\mathcal{G}(\mathcal{W})$ up to relabeling of the vertices.*

If $\mathcal{G} = (G_t)_{t\geq 0}$ is a graphon process, then we refer to G_t as the graphon process at time t.

Given a graphon $\mathcal{W} = (W, \mathscr{S})$ one can define multiple other natural random graph models; see below. However, the graph models of Definition 25 have one property which sets them apart from these models: exchangeability. To formulate this, we first recall that a random measure ξ in the first quadrant $\mathbb{R}_+^2$ is jointly exchangeable iff for every $h > 0$, permutation σ of $\mathbb{N}$, and $i, j \in \mathbb{N}$,

$$\xi(I_i \times I_j) \stackrel{d}{=} \xi(I_{\sigma(i)} \times I_{\sigma(j)}), \qquad \text{where } I_k := [h(k-1), hk].$$

Here $\stackrel{d}{=}$ means equality in distribution, and, as usual, a random measure on $\mathbb{R}_+^2$ is a measure drawn from some probability distribution over the set of all Borel measures on $\mathbb{R}_+^2$, equipped with the minimal σ-algebra for which the functions $\mu \mapsto \mu(B)$ are measurable for all Borel sets B.

To relate this notion of exchangeability to a property of a graphon process, we will assign a random measure $\xi(\mathcal{G})$ to an arbitrary projective graph process $\mathcal{G} = (G_t)_{t\geq 0}$. Defining the birth time t_v of a vertex $v \in V(\mathcal{G})$ as the infimum over all times t such that $v \in V(G_t)$, we define a random measure $\xi = \xi(\mathcal{G})$ on $\mathbb{R}_+^2$ by

$$\xi(\mathcal{G}) := \sum_{(u,v)\in E(\mathcal{G})} \delta_{(t_u, t_v)}, \tag{4}$$

where each edge $(u,v) = (v,u)$ is counted twice so that the measure is symmetric. If $\mathcal{G}$ is a graphon process with isolated vertices, i.e., $\mathcal{G} = \widetilde{\mathcal{G}}(\mathcal{W})$ for some graphon $\mathcal{W}$, it is easy to see

that at any given time, at most one vertex is born, and that at time $t = 0$, G_t is empty. In other words,

$$V(G_0) = \emptyset \quad \text{and} \quad |V(G_t)\backslash V(G_{t^-})| \leq 1 \text{ for all } t > 0. \tag{5}$$

It is not that hard to check that the measure ξ is jointly exchangeable if $\mathcal{G}$ is a graphon process with isolated vertices[7] generated from some graphon $\mathcal{W}$. But it turns out that the converse is true as well, provided the sequence has uniformly regular tails. The following theorem will be proved in Appendix G, and as with Caron and Fox (2014) we will rely on the Kallenberg theorem for jointly exchangeable measures (Kallenberg, 2005, Theorem 9.24) for this description. Veitch and Roy (2015) have independently formulated and proved a similar theorem, except that their version does not include integrability of the graphon or uniform tail regularity of the sequence of random graphs.

Before stating our theorem, we note that given a locally finite symmetric measure ξ that is a countable sum of off-diagonal, distinct atoms of weight one in the interior of $\mathbb{R}^2_+$, we can always find a projective family of simple graphs G_t obeying the condition (5) and the other assumptions we make above, and that up to vertices which stay isolated for all times, this family of graphs is uniquely determined by ξ up to relabeling of the vertices. Any projective family of countable simple graphs $\mathcal{G}$ with finitely many edges at any given time can be transformed into one obeying the condition (5) (by letting the vertices appear in the graph G_t exactly at the time they were born and merging vertices born at the same time, and then labeling vertices by their birth time), provided the measure $\xi(\mathcal{G})$ has only point masses of weight one, and has no points on the diagonal and the coordinate axes.

Theorem 26 *Let $\widetilde{\mathcal{G}} = (\widetilde{G}_t)_{t\geq 0}$ be a projective family of random simple graphs which satisfy (5), and define $\xi = \xi(\widetilde{\mathcal{G}})$ by (4). Then the following two conditions are equivalent:*

(i) The measure ξ is a jointly exchangeable random measure and $(\widetilde{G}_t)_{t\geq 0}$ has uniformly regular tails.

(ii) There is a $\mathbb{R}_+$-valued random variable α such that $\mathcal{W}_\alpha = (W_\alpha, \mathbb{R}_+)$ is a $[0,1]$-valued graphon almost surely, and such that conditioned on α, $(\widetilde{G}_t)_{t\geq 0}$ (modulo vertices that are isolated for all $t \geq 0$) has the law of $\widetilde{\mathcal{G}}(\mathcal{W}_\alpha)$ up to relabeling of the vertices.

Recall that we called two graphons equivalent if their distance in the cut metric $\delta_\square$ is zero. The following theorem shows that this notion of equivalence is the same as equivalence of the graphon process generated from two graphons, in the sense that the resulting random graphs have the same distribution. Note that in (ii) we only identify the law of $\widetilde{G}_t$ up to vertices that are isolated for all times; it is clear that if we extend the underlying measure space $\mathscr{S}$ and extend $\mathcal{W}$ trivially to this measure space, the resulting graphon is equivalent to $\mathcal{W}$ and the law of the graphs G_t remains unchanged, while the law of $\widetilde{G}_t$ might change due to additional isolated vertices.

7. This is one of the instances in which the family $\widetilde{\mathcal{G}}(\mathcal{W})$ is more useful that the family $\mathcal{G}(\mathcal{W})$: the latter only contains information about when a vertex first appeared in an edge in $\widetilde{\mathcal{G}}(\mathcal{W})$, and not information about when it was born.

Theorem 27 *For $i = 1, 2$ let $\mathcal{W}_i = (W_i, \mathscr{S}_i)$ be $[0,1]$-valued graphons, and let $(\widetilde{G}^i_t)_{t\geq 0}$ and $(G^i_t)_{t\geq 0}$ be the graphon processes generated from $\mathcal{W}_i$ with and without, respectively, isolated vertices. Then the following statements are equivalent:*

(i) $\delta_\square(\mathcal{W}_1, \mathcal{W}_2) = 0$.

(ii) After removing all vertices which are isolated for all times, $(\widetilde{G}^1_t)_{t\geq 0}$ and $(\widetilde{G}^2_t)_{t\geq 0}$ are equal in law up to relabeling of the vertices.

(iii) $(G^1_t)_{t\geq 0}$ and $(G^2_t)_{t\geq 0}$ are equal in law up to relabeling of the vertices.

The theorem will be proved in Appendix D. We show that (i) implies (ii) and (iii) by using Proposition 51, which says that the infimum in the definition of $\delta_\square$ is attained under certain assumptions on the underlying graphons. We show that (ii) or (iii) imply (i) by using Theorem 28(i).

As indicated before, in addition to the graphon processes defined above, there are several other natural random graph models generated from a graphon $\mathcal{W}$. Consider a sequence of probability measures $(\mu_n)_{n\in\mathbb{N}}$ on $(S, \mathcal{S})$, and construct a sequence of random graphs G_n as follows. Start with a single vertex $(1, x_1)$ with x_1 sampled from μ_1. In step n, sample x_n from μ_n, independently from all vertices and edges sampled so far, and for each $i = 1, \dots k$, add an edge between (i, x_i) and (n, x_n) with probability $W(x_i, x_n)$, again independently for each i (and independently of all vertices and edges chosen before). Alternatively, sample an infinite sequence of independent features $x_1, x_2, \dots$ distributed according to $\mu_1, \mu_2, \dots$, and let G be the graph on infinitely many vertices with vertex set identified with $\{(n, x_n) : n \in \mathbb{N}\}$, such that for any two $n_1, n_2 \in \mathbb{N}$ there is an edge between (n_1, x_{n_1}) and (n_2, x_{n_2}) independently with probability $W(x_{n_1}, x_{n_2})$. For each $n \in \mathbb{N}$ let G_n be the induced subgraph of G consisting only of the vertices (k, x_k) for which $k \leq n$.

It was proven by Borgs, Chayes, Lovász, Sós, and Vesztergombi (2008) that dense $\mathcal{W}$-random graphs generated from graphons on probability spaces converge to $\mathcal{W}$. The following theorem generalizes this to graphon processes, as well as for the alternative model defined in terms of a suitable sequence of measures μ_n.

Theorem 28 *Let $\mathcal{W} = (W, \mathscr{S})$ with $\mathscr{S} = (S, \mathcal{S}, \mu)$ be a $[0,1]$-valued graphon. Then the following hold:*

(i) Almost surely $\lim_{t\to\infty} \delta^s_\square(\mathcal{W}, \widetilde{G}_t(\mathcal{W})) = 0$ and $\lim_{t\to\infty} \delta^s_\square(\mathcal{W}, G_t(\mathcal{W})) = 0$.

(ii) Let $(G_n)_{n\in N}$ be the sequence of simple graphs generated from $\mathcal{W}$ with arrival probabilities $\mu_n := \mu(S_n)^{-1}\mu|_{S_n}$ as described above, where we assume $\bigcup_{n\in\mathbb{N}} S_n = S$, $S_n \subseteq S_{n+1}$, and $0 < \mu(S_n) < \infty$ for all $n \in \mathbb{N}$, and W is not equal to 0 almost everywhere. Then a.s.-$\lim_{n\to\infty} \delta^s_\square(\mathcal{W}, G_n) = 0$ if and only if $\sum_{n=1}^\infty \mu(S_n)^{-1} = \infty$.

We will prove the theorem in Appendix D. Part (i) of the theorem is proved by observing that for any set $A \subseteq S$ of finite measure, the induced subgraph of $\widetilde{G}_t$ consisting of the vertices with feature in A has the law of a graph generated from a graphon over a probability space. This implies that we can use convergence results for dense graphs to conclude the proof. In

our proof of part (ii) we first show that the condition on S_n is necessary for convergence, by showing that otherwise $E(G_n)$ is empty for all $n \in \mathbb{N}$ with positive probability. We show that the condition on S_n is sufficient by constructing a coupling of $(G_n)_{n\in\mathbb{N}}$ and a graphon process $(\widetilde{G}_t)_{t\geq 0}$.

2.5 Left Convergence

Left convergence is a notion of convergence where we consider subgraph counts of small test graphs. Existing literature defines left convergence both for dense graphs (Lovász and Szegedy, 2006) and for bounded degree graphs (Borgs, Chayes, Kahn, and Lovász, 2013), with a different renormalization factor to adjust for the difference in edge density. We will operate with a definition of subgraph density with an intermediary renormalization factor, to take into account that our graphon process satisfies $\omega(|V(G_t)|) = |E(G_t)| = O(|V(G_t)|^2)$. For dense graphs our definition of left convergence coincides with the standard definition in the theory of dense graphs.

For a simple graph F and a simple graph G define $\hom(F,G)$ to be the number of adjacency preserving maps $\phi\colon V(F) \to V(G)$, i.e., maps ϕ such that if $(v_1,v_2) \in E(F)$, then $(\phi(v_1),\phi(v_2)) \in E(G)$, and define $\mathrm{inj}(F,G)$ be the number of such maps that are injective.

Define the *rescaled homomorphism density* $h(F,G)$ and the *rescaled injective homomorphism density* $h_{\mathrm{inj}}(F,G)$ of F in G by

$$h(F,G) := \frac{\hom(F,G)}{(2|E(G)|)^{|V(F)|/2}} \quad \text{and} \quad h_{\mathrm{inj}}(F,G) := \frac{\mathrm{inj}(F,G)}{(2|E(G)|)^{|V(F)|/2}}.$$

For any $[0,1]$-valued graphon $\mathcal{W} = (W,\mathscr{S})$ we define the rescaled homomorphism density of F in $\mathcal{W}$ by

$$h(F,\mathcal{W}) := \|W\|_1^{-|V(F)|/2} \int_{S^{|V(F)|}} \prod_{(i,j)\in E(F)} W(x_i,x_j)\, dx_1 \dots dx_{|V(F)|}.$$

Note that in general, $h(F,\mathcal{W})$ need not be finite. Take, for example, $\mathcal{W} = (W,\mathbb{R}_+)$ to be a graphon of the form

$$W(x,y) = \begin{cases} 1 & \text{if } y \leq f(x), \text{ and} \\ 0 & \text{otherwise,} \end{cases}$$

where

$$f(x) = \begin{cases} x^{-1/2} & \text{if } 0 \leq x \leq 1, \text{ and} \\ x^{-2} & \text{if } x \geq 1. \end{cases}$$

Let $D_W(x) = \int_{\mathbb{R}_+} W(x,y)\, dy$. Then D_W is in $L^1(\mathbb{R}_+)$, but not in $L^k(\mathbb{R}_+)$ for any $k \geq 2$. Thus if F is a star with $k \geq 2$ leaves, then $h(F,\mathcal{W}) := \|W\|_1^{-(k+1)/2} \int_{\mathbb{R}_+} D_W^k(x)dx = \infty$. Proposition 30(ii) below, whose proof is based on Lemma 62 in Appendix H, gives one criterion which guarantees that $h(F,\mathcal{W}) < \infty$ for all simple connected graphs F.

Definition 29 *A sequence $(G_n)_{n\in\mathbb{N}}$ is left convergent if its edge density is converging, and if for every simple connected graph F with at least two vertices, the limit $\lim_{n\to\infty} h(F,G_n)$ exists and is finite. Left convergence is defined similarly for a continuous-time family of graphs $(G_t)_{t\geq 0}$.*

For dense graphs left convergence is equivalent to metric convergence (Borgs, Chayes, Lovász, Sós, and Vesztergombi, 2008). This equivalence does not hold for our graphs, but convergence of subgraph densities (possibly with an infinite limit) does hold for graphon processes.

Proposition 30 *(i) If $\mathcal{W} = (W,\mathscr{S})$ is a $[0,1]$-valued graphon and $(G_t)_{t>0}$ is a graphon process, then for every simple connected graph F with at least two vertices,*

$$\lim_{t\to\infty} h_{\mathrm{inj}}(F,G_t) = h(F,\mathcal{W}) \in [0,\infty]$$

almost surely.

(ii) In the setting of (i), if $D_W(x) := \int_S W(x,x')\, d\mu(x')$ is in L^p for all $p \in [1,\infty)$, then $h(F,\mathcal{W}) < \infty$ for every simple connected graph F with at least two vertices and

$$\lim_{t\to\infty} h(F,G_t) = \lim_{t\to\infty} h_{\mathrm{inj}}(F,G_t) = h(F,\mathcal{W})$$

almost surely, so in particular $(G_t)_{t>0}$ is left convergent.

(iii) Assume $(G_n)_{n\in\mathbb{N}}$ is a sequence of simple graphs with bounded degree such that

$$\lim_{n\to\infty} |V(G_n)| = \infty$$

and $E(G_n) \neq \emptyset$ for all sufficiently large n. Then $(G_n)_{n\in\mathbb{N}}$ is trivially left convergent, and $\lim_{n\to\infty} h(F,G_n) = 0$ for every connected F for which $|V(F)| \geq 3$.

(iv) Left convergence does not imply convergence for $\delta^{\mathrm{s}}_{\square}$, and convergence for $\delta^{\mathrm{s}}_{\square}$ does not imply left convergence.

The proposition will be proved in Appendix H. Part (i) is immediate from Proposition 56, which is proved using martingale convergence and that $\mathrm{inj}(F,G_{-t})$ appropriately normalized evolves as a backwards martingale. Part (ii) is proved by using that $h(F,G_t)$ and $h_{\mathrm{inj}}(F,G_t)$ are not too different under certain assumptions on the underlying graphon. Part (iii) is proved by bounding $\hom(F,G_n)$ from above, and part (iv) is proved by constructing explicit counterexamples.

Remark 31 *While we stated the above proposition for graphons, i.e., for the case when $W \in L^1$, the main input used in the proof, Proposition 56 below, does not require an integrable W, but just the measurability of the function $W\colon S \times S \to [0,1]$.*

Acknowledgments

We thank Svante Janson for his careful reading of an earlier version of the paper and for his numerous suggestions, and we thank Edoardo Airoldi for helpful discussions. Holden was supported by an internship at Microsoft Research New England and by a doctoral research fellowship from the Norwegian Research Council.

Appendix A. Cut Metric and Invariant L^p Metric

The main goal of this appendix is to prove Proposition 6, which says that $\delta_\square$ and δ_1 are well defined and pseudometrics. In the course of our proof, we will actually generalize this proposition, and show that it can be extended to the invariant L^p metric δ_p, provided the two graphons are non-negative and in L^p.

We start by defining the distance $\delta_p(\mathcal{W}_1,\mathcal{W}_2)$ for two such graphons $\mathcal{W}_1 = (W_1,\mathscr{S}_1)$ and $\mathcal{W}_2 = (W_1,\mathscr{S}_2)$ over two spaces $\mathscr{S}_1 = (S_1,\mathcal{S}_1,\mu_i)$ and $\mathscr{S}_2 = (S_2,\mathcal{S}_2,\mu_2)$ of equal total measure, in which case we set

$$\delta_p(\mathcal{W}_1,\mathcal{W}_2) := \inf_\mu \|W_1^{\pi_1} - W_2^{\pi_2}\|_{p,S_1\times S_2,\mu},$$

where, as before, π_1 and π_2 are the projections from $S_1\times S_2$ to S_1 and S_2, respectively, and the infimum is over all couplings μ of μ_1 and μ_2. If $\mu_1(S_1)\neq\mu_2(S_2)$ we define $\delta_p(\mathcal{W}_1,\mathcal{W}_2)$ by trivially extending $\mathcal{W}_1$ and $\mathcal{W}_2$ to two graphons $\widetilde{\mathcal{W}}_1$ and $\widetilde{\mathcal{W}}_2$, respectively, over measure spaces of equal measure, and defining $\delta_p(\mathcal{W}_1,\mathcal{W}_2) := \delta_p(\widetilde{\mathcal{W}}_1,\widetilde{\mathcal{W}}_2)$, just as in Definition 5 (ii).

Proposition 32 *For $i=1,2$, let $\mathcal{W}_i=(W_i,\mathscr{S}_i)$ be non-negative graphons over $\mathscr{S}_i = (S_i,\mathcal{S}_i,\mu_i)$ with $W_i\in L^p(S_i\times S_i)$ for some $p\in(1,\infty)$. Then $\delta_p(\mathcal{W}_1,\mathcal{W}_2)$ is well defined. In particular, $\delta_p(\mathcal{W}_1,\mathcal{W}_2)$ does not depend on the choice of extensions $\widetilde{\mathcal{W}}_1$ and $\widetilde{\mathcal{W}}_2$. Furthermore, δ_p is a pseudometric on the space of non-negative graphons in L^p.*

We will prove Proposition 32 at the same time as Proposition 6. We will also establish an estimate (Lemma 44) saying that two graphons are close in the cut metric if we obtain one from the other by slightly modifying the measure of the underlying measure space. Finally we state and prove a lemma, Lemma 45, that immediately implies Proposition 21.

The following lemma will be used in the proof of Propositions 10 and 48. The analogous result for probability spaces can for example be found in a paper by Janson (2013, Theorem A.7), and the extension to σ-finite measure spaces is straightforward.

Lemma 33 *Let $\mathscr{S}=(S,\mathcal{S},\mu)$ be an atomless σ-finite Borel space. Then $\mathscr{S}$ is isomorphic to $([0,\mu(S)),\mathcal{B},\lambda)$, where $\mathcal{B}$ is the Borel σ-algebra and λ is Lebesgue measure.*

Proof For $\mu(S)<\infty$, this holds because every atomless Borel probability space is isomorphic to $[0,1]$ equipped with the Borel σ-algebra and Lebesgue measure (Janson, 2013, Theorem A.7). For $\mu(S)=\infty$ we use that by the hypotheses of σ-finiteness there exist disjoint sets $S_k\in\mathcal{S}$ for $k\in\mathbb{N}$ such that $S=\bigcup_{k=1}^\infty S_k$ and $\mu(S_k)<\infty$ for all $k\in\mathbb{N}$. For each $k\in\mathbb{N}$ we can find isomorphisms $\phi\colon [0,\mu(S_k)]\to[0,\mu(S_k))$ and $\widetilde{\phi}\colon S_k\to[0,\mu(S_k)]$. It follows by considering the composed map $\phi\circ\widetilde{\phi}$ that S_k is isomorphic to $[0,\mu(S_k))$. The lemma follows by constructing an isomorphism from S to $\mathbb{R}_+$ where each set S_k is mapped onto an half-open interval of length $\mu(S_k)$. ∎

The first statement of Proposition 6, i.e., the existence of a coupling, follows directly from the following more general result.

Lemma 34 *For $k=1,2$ let $\mathscr{S}_k=(S_k,\mathcal{S}_k,\mu_k)$ be a σ-finite measure space such that $\mu_1(S_1)=\mu_2(S_2)\in(0,\infty]$. Let $D_k\in\mathcal{S}_k$, and let $\widetilde{\mu}$ be a measure on the product space $D_1\times D_2$, where D_k is equipped with the induced σ-algebra from S_k. Assume the marginals $\widetilde{\mu}_1$ and $\widetilde{\mu}_2$ of $\widetilde{\mu}$ are bounded above by $\mu_1|_{D_1}$ and $\mu_2|_{D_2}$, respectively, and that either $D_1=D_2=\emptyset$ or $\mu_k(S_k\setminus D_k)=\infty$ for $k=1,2$. Then there exists a coupling μ of $\mathscr{S}_1$ and $\mathscr{S}_2$, such that $\mu|_{D_1\times D_2}=\widetilde{\mu}$.*

Proof First we consider the case when $D_1=D_2=\emptyset$. If $\mu_1(S_1)=\mu_2(S_2)<\infty$ we define μ to be proportional to the product measure of μ_1 and μ_2. Explicitly, for $A\in\mathcal{S}_1$ and $B\in\mathcal{S}_2$, we set $\mu(A\times B)=\mu_1(A)\mu_2(B)/\mu_1(S_1)$. This clearly gives $\mu(S_1\times B)=\mu_2(B)$ and $\mu(A\times S_2)=\mu_1(A)$, as required.

If $\mu_1(S_1)=\mu_2(S_2)=\infty$, we consider partitions of S_1 and S_2 into disjoint sets of finite measure, with $S_k=\bigcup_{i\geq1}A_i^k$ for $k=1,2$. Let $I_1,I_2,\dots$ and $J_1,J_2,\dots$ be decompositions of $[0,\infty)$ into adjacent intervals of lengths $\mu_1(A_1^1),\mu_1(A_2^1),\dots$ and $\mu_2(A_1^2),\mu_2(A_2^2),\dots$, respectively. We then define a measure μ on $(S_1\times S_2,\mathcal{S}_1\times\mathcal{S}_2)$ by

$$\mu(A\times B)=\sum_{i,j\geq1}\frac{\lambda(I_i\cap J_j)}{\lambda(I_i)\lambda(J_j)}\mu_1(A\cap A_i^1)\mu_2(B\cap A_j^2),\quad\text{for } A\in\mathcal{S}_1, B\in\mathcal{S}_2.$$

As a weighted sum of product measures, μ is a measure, and inserting $A=S_1$ or $B=S_2$, one easily verifies that μ has marginals μ_1 and μ_2. This completes the proof of the lemma in the case that $D_1=D_2=\emptyset$.

Now we consider the general case. Decomposing D_1 and D_2 into disjoint sets of finite mass with respect to μ_1 and μ_2, $D_k=\bigcup_{i\geq1}D_i^k$ with $\mu_k(D_i^k)<\infty$, we define measures $\widehat{\mu}_k^{(\ell)}$ on S_k for $k,\ell=1,2$ by

$$\begin{aligned}
\widehat{\mu}_1^{(1)}(A)&=\frac{1}{2}\mu_1(A\cap(S_1\setminus D_1))\text{ for all }A\in\mathcal{S}_1,\\
\widehat{\mu}_2^{(1)}(B)&=\frac{1}{2}\mu_2(B\cap(S_2\setminus D_2))+\sum_{i\geq1}\Big[\mu_2(B\cap D_i^2)-\widetilde{\mu}_2(B\cap D_i^2)\Big]\text{ for all }B\in\mathcal{S}_2,\\
\widehat{\mu}_1^{(2)}(A)&=\frac{1}{2}\mu_1(A\cap(S_1\setminus D_1))+\sum_{i\geq1}\Big[\mu_1(A\cap D_i^1)-\widetilde{\mu}_1(A\cap D_i^1)\Big]\text{ for all }A\in\mathcal{S}_1,\text{ and}\\
\widehat{\mu}_2^{(2)}(S_1)&=\frac{1}{2}\mu_2(B\cap(S_2\setminus D_2))\text{ for all }B\in\mathcal{S}_2.
\end{aligned}$$

Note that $\widehat{\mu}_1^{(\ell)}(S_1)=\widehat{\mu}_2^{(\ell)}(S_2)=\infty$ for $\ell=1,2$ by our assumption $\mu_k(S_k\setminus D_k)=\infty$ for $k=1,2$. By the result for the case $D_1=D_2=\emptyset$, for $\ell=1,2$, we can find couplings $\widehat{\mu}^{(\ell)}$ of $\widehat{\mu}_1^{(\ell)}$ and $\widehat{\mu}_2^{(\ell)}$ on $S_1\times S_2$. Extending the measure $\widetilde{\mu}$ to a measure on $S_1\times S_2$ by assigning measure 0 to all sets which have an empty intersection with $D_1\times D_2$, the measure $\mu:=\widehat{\mu}^{(1)}+\widehat{\mu}^{(2)}+\widetilde{\mu}$ has the appropriate marginals. To see that $\mu|_{D_1\times D_2}=\widetilde{\mu}$, we note that $\widehat{\mu}^{(1)}(D_1\times S_2)=\widehat{\mu}_1^{(1)}(D_1)=0$ and $\widehat{\mu}^{(2)}(S_1\times D_2)=\widehat{\mu}_2^{(2)}(D_2)=0$, implying in particular that $\widehat{\mu}^{(1)}(D_1\times D_2)=\widehat{\mu}^{(2)}(D_1\times D_2)=0$. ∎

Corollary 35 *For $k = 1, 2$ let $\mathscr{S}_k = (S_k, \mathcal{S}_k, \mu_k)$ be a σ-finite measure space such that $\mu_1(S_1) = \mu_2(S_2) \in (0, \infty]$, and let μ be a coupling of μ_1 and μ_2. Let $D_k \in \mathcal{S}_k$ be such that $\mu(D_1 \times (S_2 \setminus D_2)) = \mu((S_1 \setminus D_1) \times D_2) \in (0, \infty]$. Then there exists a coupling $\widetilde{\mu}$ of μ_1 and μ_2 such that $\widetilde{\mu}$ is supported on $(D_1 \times D_2) \cup ((S_1 \setminus D_1) \times (S_2 \setminus D_2))$ and $\widetilde{\mu} \geq \mu$ on $(D_1 \times D_2) \cup ((S_1 \setminus D_1) \times (S_2 \setminus D_2))$.*

Proof Let μ' be the restriction of μ to $(D_1 \times D_2) \cup ((S_1 \setminus D_1) \times (S_2 \setminus D_2))$, let μ'_1 and μ'_2 be its marginals, and let $\delta_i = \mu_i - \mu'_i$. Then $\delta_1(D_1) = \mu(D_1 \times (S_2 \setminus D_2))$ and $\delta_2(D_2) = \mu((S_1 \setminus D_1) \times D_2) = \delta_1(D_1)$ by the hypotheses of the corollary. In a similar way, $\delta_1(S_1 \setminus D_1) = \mu((S_1 \setminus D_1) \times D_2) = \delta_2(S_2 \setminus D_2)$. With the help of the previous lemma, and considering the domains $D_1 \times D_2$ and $(S_1 \setminus D_1) \times (S_2 \setminus D_2)$ separately, we then construct a coupling δ of δ_1 and δ_2 that has support on $(D_1 \times D_2) \cup ((S_1 \setminus D_1) \times (S_2 \setminus D_2))$. Setting $\tilde{\mu} = \mu' + \delta$ we obtain the statement of the corollary. ∎

The following basic lemma will be used multiple times throughout this appendix. The analogous result for probability spaces can be found for example in a paper by Janson (2013, Lemma 6.4).

Lemma 36 *Let $p \geq 1$, let $\mathscr{S}_i = (S_i, \mathcal{S}_i, \mu_i)$ for $i = 1, 2$ be such that $\mu_1(S_1) = \mu_2(S_2)$, and let $\mathcal{W}_1 = (W_1, \mathscr{S}_1)$, $\mathcal{W}'_1 = (W'_1, \mathscr{S}_1)$, and $\mathcal{W}_2 = (W_2, \mathscr{S}_2)$ be graphons in L^p. Defining $\delta_\square$ and δ_p as in Definition 5(i), we have*

$$\delta_\square(\mathcal{W}_1, \mathcal{W}_2) \leq \delta_\square(\mathcal{W}'_1, \mathcal{W}_2) + \|W_1 - W'_1\|_\square \leq \delta_\square(\mathcal{W}'_1, \mathcal{W}_2) + \|W_1 - W'_1\|_1$$

and

$$\delta_p(\mathcal{W}_1, \mathcal{W}_2) \leq \delta_p(\mathcal{W}'_1, \mathcal{W}_2) + \|W_1 - W'_1\|_p.$$

Proof The second bound on $\delta_\square(\mathcal{W}_1, \mathcal{W}_2)$ is immediate, so the rest of the proof will consist of proving the first bound on $\delta_\square(\mathcal{W}_1, \mathcal{W}_2)$ as well as the bound on $\delta_p(\mathcal{W}_1, \mathcal{W}_2)$. Let μ be a measure on $(S_1 \times S_2, \mathcal{S}_1 \times \mathcal{S}_2)$ with marginals μ_1 and μ_2, respectively, and let $\pi_i \colon S_1 \times S_2 \to S_i$ denote projections for $i = 1, 2$. Since $\|\cdot\|_\square$ clearly satisfies the triangle inequality,

$$\begin{aligned}\delta_\square(\mathcal{W}_1, \mathcal{W}_2) &\leq \|W_1^{\pi_1} - W_2^{\pi_2}\|_{\square, S_1 \times S_2, \mu} \\ &\leq \|(W'_1)^{\pi_1} - W_2^{\pi_2}\|_{\square, S_1 \times S_2, \mu} + \|(W'_1)^{\pi_1} - W_1^{\pi_1}\|_{\square, S_1 \times S_2, \mu} \\ &= \|(W'_1)^{\pi_1} - W_2^{\pi_2}\|_{\square, S_1 \times S_2, \mu} + \|W'_1 - W_1\|_{\square, S_1, \mu_1}.\end{aligned}$$

The desired result follows by taking an infimum over all couplings. The bound on $\delta_p(\mathcal{W}_1, \mathcal{W}_2)$ follows in the same way from the triangle inequality for $\|\cdot\|_p$. ∎

Remark 37 *We state the above lemma only for the case when $\mu_1(S_1) = \mu_2(S_2)$, since we have not yet proved that $\delta_\square$ and δ_p are well defined otherwise. However, once we have proved this, it is a direct consequence of Definition 5(ii) that the above lemma also holds when $\mu_1(S_1) \neq \mu_2(S_2)$.*

Definition 38 *Let $(S, \mathcal{S})$ be a measurable space, and consider a function $W \colon S \times S \to \mathbb{R}$. Then W is a* step function *if there are some $n \in \mathbb{N}$, disjoint sets $A_i \in \mathcal{S}$ satisfying $\mu(A_i) < \infty$ for $i \in \{1, \dots, n\}$, and constants $a_{i,j} \in \mathbb{R}$ for $i, j \in \{1, \dots, n\}$ such that*

$$W = \sum_{i,j \in \{1,\dots,n\}} a_{i,j} \mathbf{1}_{A_i \times A_j}.$$

Note that in order for W to be a step function it is not sufficient that it is simple, i.e., that it attains a finite number of values; the sets on which the function is constant are required to be product sets. The set of step functions is dense in L^1; hence Lemma 36 implies that every graphon can be approximated arbitrarily closely by a step function for the $\delta_\square$ metric.

Lemma 39 *Let $p \geq 1$, let $\mathcal{W}_1 = (W_1, \mathscr{S}_1)$ and $\mathcal{W}_2 = (W_2, \mathscr{S}_2)$ be graphons, and let $S_1 = \bigcup_{i \in I} A_i$ and $S_2 = \bigcup_{k \in J} B_k$ for finite index sets I and J such that $A_i \cap A_{i'} = \emptyset$ for $i \neq i'$ and $B_j \cap B_{j'} = \emptyset$ for $j \neq j'$. Suppose W_1 and W_2 are step functions of the form*

$$W_1 = \sum_{i,i' \in I} a_{i,i'} \mathbf{1}_{A_i \times A_{i'}} \quad \text{and} \quad W_2 = \sum_{j,j' \in J} b_{j,j'} \mathbf{1}_{B_j \times B_{j'}},$$

where $a_{i,i'}$ and $b_{j,j'}$ are constants in $\mathbb{R}$. Let μ and μ' be two coupling measures on $S_1 \times S_2$, such that $\mu(A_i \times B_k) = \mu'(A_i \times B_k)$ for all $(i, k) \in I \times J$. Then $\|W_1^{\pi_1} - W_2^{\pi_2}\|_{\square,\mu} = \|W_1^{\pi_1} - W_2^{\pi_2}\|_{\square,\mu'}$ and $\|W_1^{\pi_1} - W_2^{\pi_2}\|_{p,\mu} = \|W_1^{\pi_1} - W_2^{\pi_2}\|_{p,\mu'}$.

Proof For all $U, V \subseteq S_1 \times S_2$,

$$\int_{U \times V} \left(W_1^{\pi_1} - W_2^{\pi_2}\right) d\mu \, d\mu = \sum_{i,i',j,j'} \mu(U \cap (A_i \times B_j)) \mu(V \cap (A_{i'} \times B_{j'}))(a_{i,i'} - b_{j,j'}). \quad (6)$$

From the form of this expression and the definition of $\|\cdot\|_\square$ it is clear that we may assume there are sets $U', V' \subseteq S_1 \times S_2$ such that for all i, j,

$$A_i \times B_j \subseteq U' \quad \text{or} \quad (A_i \times B_j) \cap U' = \emptyset$$

and

$$A_i \times B_j \subseteq V' \quad \text{or} \quad (A_i \times B_j) \cap V' = \emptyset,$$

and such that

$$\|W_1^{\pi_1} - W_2^{\pi_2}\|_{\square,\mu} = \int_{U' \times V'} \left(W_1^{\pi_1} - W_2^{\pi_2}\right) d\mu \, d\mu.$$

Hence it follows from (6) that $\|W_1^{\pi_1} - W_2^{\pi_2}\|_{\square,\mu} = \|W_1^{\pi_1} - W_2^{\pi_2}\|_{\square,\mu'}$ if $\mu(A_i \times B_j) = \mu'(A_i \times B_j)$ for all i, j. The proof for the L^p metric follows from the fact that

$$\|W_1^{\pi_1} - W_2^{\pi_2}\|_{p,\mu}^p = \sum_{i,i',j,j'} |a_{i,i'} - b_{j,j'}|^p \mu(A_i \times B_j) \mu(A_{i'} \times B_{j'}).$$

∎

Corollary 40 *Let $p \geq 1$ and for $k = 1, 2$, let $\mathcal{W}_k = (W_k, \mathscr{S}_k)$ with $\mathscr{S}_k = (S_k, \mathcal{S}_k, \mu_k)$ be graphons in L^p. For $k = 1, 2$, let $\widetilde{\mathscr{S}_k} = (\widetilde{S}_k, \widetilde{\mathcal{S}}_k, \widetilde{\mu}_k)$ and $\widehat{\mathscr{S}_k} = (\widehat{S}_k, \widehat{\mathcal{S}}_k, \widehat{\mu}_k)$ be extensions of $\mathscr{S}_k$ with $\widetilde{\mu}_1(\widetilde{S}_1) = \widetilde{\mu}_2(\widetilde{S}_2) \in (0, \infty]$ and $\widehat{\mu}_1(\widehat{S}_1) = \widehat{\mu}_2(\widehat{S}_2) \in (0, \infty]$, and let $\widetilde{W}_k$ and $\widehat{W}_k$ be the trivial extensions of W_k to $\widetilde{\mathscr{S}_k}$ and $\widehat{\mathscr{S}_k}$. Let $\widetilde{\mu}$ and $\widehat{\mu}$ be couplings of $\widetilde{\mu}_1$ and $\widetilde{\mu}_2$, and $\widehat{\mu}_1$ and $\widehat{\mu}_2$, respectively. If $\widetilde{\mu}$ and $\widehat{\mu}$ agree on $S_1 \times S_2$, then $\|\widetilde{W}_1^{\pi_1} - \widetilde{W}_2^{\pi_2}\|_{\square,\widetilde{\mu}} = \|\widehat{W}_1^{\pi_1} - \widehat{W}_2^{\pi_2}\|_{\square,\widehat{\mu}}$ and $\|\widetilde{W}_1^{\pi_1} - \widetilde{W}_2^{\pi_2}\|_{p,\widetilde{\mu}} = \|\widehat{W}_1^{\pi_1} - \widehat{W}_2^{\pi_2}\|_{p,\widehat{\mu}}$.*

Proof By Lemma 36 and the fact that step functions are dense in L^1 and in L^p, it is sufficient to prove the corollary for step functions. The corollary then follows from Lemma 39 by observing that for two sets $A \in \mathcal{S}_1$ and $B \in \mathcal{S}_2$ with finite measure $\mu_1(A)$ and $\mu_2(B)$, the $\widetilde{\mu}$ measure of sets of the form $A \times (\widetilde{S}_2 \setminus S_2)$ and $(\widetilde{S}_1 \setminus S_1) \times B$ can be expressed as $\mu_1(A) - \widetilde{\mu}(A \times S_2)$ and $\mu_2(B) - \widetilde{\mu}(S_1 \times B)$, respectively, implying that $\|\widetilde{W}_1^{\pi_1} - \widetilde{W}_2^{\pi_2}\|_{\square,\widetilde{\mu}}$ and $\|\widetilde{W}_1^{\pi_1} - \widetilde{W}_2^{\pi_2}\|_{p,\widetilde{\mu}}$ depend only on the restriction of $\widetilde{\mu}$ to $S_1 \times S_2$, and similarly for $\|\widehat{W}_1^{\pi_1} - \widehat{W}_2^{\pi_2}\|_{\square,\widehat{\mu}}$ and $\|\widehat{W}_1^{\pi_1} - \widehat{W}_2^{\pi_2}\|_{p,\widehat{\mu}}$. ■

Lemma 39 is also used in the proof of the triangle inequality in the following lemma. The proof follows the same strategy as the proof by Janson (2013, Lemma 6.5) for the case of probability spaces.

Lemma 41 *Let $p \geq 1$. For $i = 1, 2, 3$ let $\mathcal{W}_i = (W_i, \mathscr{S}_i)$ with $\mathscr{S}_i = (S_i, \mathcal{S}_i, \mu_i)$ be a graphon in L^p, such that $\mu_1(S_1) = \mu_2(S_2) = \mu_3(S_3) \in (0, \infty]$. Defining $\delta_\square$ and δ_p as in Definition 5(i), we have*

$$\delta_\square(\mathcal{W}_1, \mathcal{W}_3) \leq \delta_\square(\mathcal{W}_1, \mathcal{W}_2) + \delta_\square(\mathcal{W}_2, \mathcal{W}_3) \quad \text{and} \quad \delta_p(\mathcal{W}_1, \mathcal{W}_3) \leq \delta_p(\mathcal{W}_1, \mathcal{W}_2) + \delta_p(\mathcal{W}_2, \mathcal{W}_3).$$

Proof By Lemma 36 and since step functions are dense in L^1, we may assume that W_i is a step function for $i = 1, 2, 3$. Let $\mathscr{S}_1 = \bigcup_{j=1}^\infty A_j$ (resp. $\mathscr{S}_2 = \bigcup_{j=1}^\infty B_j$, $\mathscr{S}_3 = \bigcup_{j=1}^\infty C_j$) be such that $W_1|_{A_j \times A_k}$ (resp. $W_2|_{B_j \times B_k}$, $W_3|_{C_j \times C_k}$) is constant for all $j, k \in \mathbb{N}$, and assume without loss of generality that $\mu_1(A_j), \mu_2(B_j), \mu_3(C_j) \in (0, \infty)$ for all $j \in \mathbb{N}$. Throughout the proof we abuse notation slightly and let π_i denote projection onto S_i from any space which is a product of S_i and another space; for example, π_1 denotes projection onto S_1 from $S_1 \times S_2 \times S_3$, $S_1 \times S_2$, and $S_1 \times S_3$.

Let $\varepsilon > 0$, and let μ' (resp. μ'') be a coupling measure on $S_1 \times S_2$ (resp. $S_2 \times S_3$) such that

$$\|W_1^{\pi_1} - W_2^{\pi_2}\|_{\square,\mu'} < \delta_\square(\mathcal{W}_1, \mathcal{W}_2) + \varepsilon \quad \text{and} \quad \|W_2^{\pi_2} - W_3^{\pi_3}\|_{\square,\mu''} < \delta_\square(\mathcal{W}_2, \mathcal{W}_3) + \varepsilon.$$

We define a measure μ on $S_1 \times S_2 \times S_3$ for any $E \subseteq S_1 \times S_2 \times S_3$ which is measurable for the product σ-algebra by

$$\mu(E) = \sum_{i,j,k} \frac{\mu'(A_i \times B_j)\mu''(B_j \times C_k)}{\mu_2(B_j)} \frac{\mu_1 \times \mu_2 \times \mu_3(E \cap (A_i \times B_j \times C_k))}{\mu_1(A_i)\mu_2(B_j)\mu_3(C_k)}.$$

By a straightforward calculation (see, for example, the paper by Janson, 2013, Lemma 6.5) the three mappings $\pi_l \colon (S_1 \times S_2 \times S_3, \mu) \to (S_l, \mu_l)$ for $l = 1, 2, 3$ are measure-preserving.

Furthermore, if $\widetilde{\mu}'$ is the pushforward measure of μ for the projection $\pi_{12} \colon S_1 \times S_2 \times S_3 \to S_1 \times S_2$, then

$$\widetilde{\mu}'(A_i \times B_j) = \mu'(A_i \times B_j) \quad \text{for all } i, j.$$

By Lemma 39 and since $\pi_{12} \colon (S_1 \times S_2 \times S_3, \mu) \to (S_1 \times S_2, \widetilde{\mu}')$ is measure-preserving,

$$\|W_1^{\pi_1} - W_2^{\pi_2}\|_{\square,S_1 \times S_2,\mu'} = \|W_1^{\pi_1} - W_2^{\pi_2}\|_{\square,S_1 \times S_2,\widetilde{\mu}'} = \|W_1^{\pi_1} - W_2^{\pi_2}\|_{\square,S_1 \times S_2 \times S_3,\mu}.$$

Hence,

$$\|W_1^{\pi_1} - W_2^{\pi_2}\|_{\square,S_1 \times S_2 \times S_3,\mu} < \delta_\square(\mathcal{W}_1, \mathcal{W}_2) + \varepsilon.$$

Similarly,

$$\|W_2^{\pi_2} - W_3^{\pi_3}\|_{\square,S_1 \times S_2 \times S_3,\mu} < \delta_\square(\mathcal{W}_2, \mathcal{W}_3) + \varepsilon.$$

Letting $\widehat{\mu}$ be the pushforward measure on $S_1 \times S_3$ of μ for the projection $\pi_{13} \colon S_1 \times S_2 \times S_3 \to S_1 \times S_3$, we have

$$\|W_1^{\pi_1} - W_3^{\pi_3}\|_{\square,S_1 \times S_3,\widehat{\mu}} = \|W_1^{\pi_1} - W_3^{\pi_3}\|_{\square,S_1 \times S_2 \times S_3,\mu}.$$

Since the cut norm $\|\cdot\|_\square$ clearly satisfies the triangle inequality,

$$\begin{aligned}\delta_\square(\mathcal{W}_1, \mathcal{W}_3) &\leq \|W_1^{\pi_1} - W_3^{\pi_3}\|_{\square,S_1 \times S_3,\widehat{\mu}} = \|W_1^{\pi_1} - W_3^{\pi_3}\|_{\square,S_1 \times S_2 \times S_3,\mu} \\ &\leq \|W_1^{\pi_1} - W_2^{\pi_2}\|_{\square,S_1 \times S_2 \times S_3,\mu} + \|W_2^{\pi_2} - W_3^{\pi_3}\|_{\square,S_1 \times S_2 \times S_3,\mu} \\ &< \delta_\square(\mathcal{W}_1, \mathcal{W}_2) + \delta_\square(\mathcal{W}_2, \mathcal{W}_3) + 2\varepsilon.\end{aligned}$$

Since ε was arbitrary this completes our proof for $\delta_\square$. The proof for δ_p is identical. ■

Lemma 42 *Let $\mathcal{W}_i = (W_i, \mathscr{S}_i)$ with $\mathscr{S}_i = (S_i, \mathcal{S}_i, \mu_i)$ be a graphon for $i = 1, 2$, such that $\mu_1(S_1) = \mu_2(S_2) \in (0, \infty]$. For $i = 1, 2$ let $\widetilde{\mathscr{S}_i} = (\widetilde{S}_i, \widetilde{\mathcal{S}}_i, \widetilde{\mu}_i)$ be an extension of $\mathscr{S}_i$, such that $\widetilde{\mu}_1(\widetilde{S}_1) = \widetilde{\mu}_2(\widetilde{S}_2) \in (0, \infty]$, and let $\widetilde{\mathcal{W}}_i$ be the trivial extension of $\mathcal{W}_i$ to $\widetilde{\mathscr{S}_i}$. Then $\delta_\square(\mathcal{W}_1, \mathcal{W}_2) = \delta_\square(\widetilde{\mathcal{W}}_1, \widetilde{\mathcal{W}}_2)$ and $\delta_1(\mathcal{W}_1, \mathcal{W}_2) = \delta_1(\widetilde{\mathcal{W}}_1, \widetilde{\mathcal{W}}_2)$, where $\delta_\square$ and δ_1 are as in Definition 5(i). If $p > 1$ and $\mathcal{W}_1$ and $\mathcal{W}_2$ are non-negative graphons in L^p, then the result holds for δ_p as well.*

Remark 43 *We remark that the assumption of non-negativity is necessary for the lemma to hold when $p > 1$. If for example $\mathcal{W}_1 = (\mathbf{1}, [0, 1])$ and $\mathcal{W}_2 = (-\mathbf{1}, [0, 1])$ are graphons over $[0, 1]$, and if $\widetilde{\mathcal{W}}_1$ and $\widetilde{\mathcal{W}}_2$ are the trivial extensions to $[0, 2]$, then $\delta_p(\mathcal{W}_1, \mathcal{W}_2) = 2$ and $\delta_p(\widetilde{\mathcal{W}}_1, \widetilde{\mathcal{W}}_2) = 2^{1/p}$.*

Proof We start with the proof for the cut metric. We will first prove the result for the case when $\widetilde{S}_i = \mathbb{N}$ and $S_i = S := \{1, \ldots, n\}$ for $i = 1, 2$ and some $n \in \mathbb{N}$, $\mathcal{S}_i$ and $\widetilde{\mathcal{S}}_i$ are the associated discrete σ-algebras, and $\widetilde{\mu}_i(x) = c$ for all $x \in \widetilde{S}_i$ and some $c \in (0, 1)$.

First we will argue that

$$\delta_\square(\mathcal{W}_1, \mathcal{W}_2) \geq \delta_\square(\widetilde{\mathcal{W}}_1, \widetilde{\mathcal{W}}_2).$$

By Definition 5(i) it is sufficient to prove that for each coupling measure μ on $S \times S$ we can define a coupling measure $\widetilde{\mu}$ on $\mathbb{N} \times \mathbb{N}$ such that $\|W_1^{\pi_1} - W_2^{\pi_2}\|_{\square,\mu} = \|\widetilde{W}_1^{\widetilde{\pi}_1} - \widetilde{W}_2^{\widetilde{\pi}_2}\|_{\square,\widetilde{\mu}}$. But

this is immediate, since we can define $\widetilde{\mu}$ such that $\widetilde{\mu}|_{S\times S} = \mu$, and $\widetilde{\mu}(A_1 \times A_2) = c|A_1 \cap A_2|$ for $A_i \subseteq \mathbb{N}\backslash S$.

Next we will prove that

$$\delta_\square(\mathcal{W}_1, \mathcal{W}_2) \leq \delta_\square(\widetilde{\mathcal{W}_1}, \widetilde{\mathcal{W}_2}). \tag{7}$$

Again by Definition 5(i), it will be sufficient to prove that given any coupling measure $\widetilde{\mu}$ on $\mathbb{N}\times\mathbb{N}$ we can find a coupling measure μ on $S \times S$ such that $\|W_1^{\pi_1} - W_2^{\pi_2}\|_{\square,\mu} \leq \|\widetilde{W}_1^{\pi_1} - \widetilde{W}_2^{\pi_2}\|_{\square,\widetilde{\mu}}$.

By the following argument we may approximate $\|\widetilde{W}_1^{\pi_1} - \widetilde{W}_2^{\pi_2}\|_{\square,\widetilde{\mu}}$ arbitrarily well by replacing $\widetilde{\mu}$ with a coupling measure which is supported on $(\widehat{S}\times\widehat{S}) \cup ((\mathbb{N}\backslash\widehat{S}) \times (\mathbb{N}\backslash\widehat{S}))$, where $\widehat{S} := \{1,\dots,K\}$ for some sufficiently large $K \in \mathbb{N}$. Indeed, by Corollary 35, given a coupling measure $\widetilde{\mu}$ on $\mathbb{N}\times\mathbb{N}$ and $K \in \mathbb{N}$, we can define a measure $\widehat{\mu}$ supported on $(\widehat{S}\times\widehat{S}) \cup ((\mathbb{N}\backslash\widehat{S}) \times (\mathbb{N}\backslash\widehat{S}))$ such that $\widehat{\mu} \geq \widetilde{\mu}$ on $(\widehat{S}\times\widehat{S}) \cup ((\mathbb{N}\backslash\widehat{S}) \times (\mathbb{N}\backslash\widehat{S}))$. It is easy to see from the construction of this measure in the proof of Corollary 35 that when K converges to infinity, the measure $\widehat{\mu}$ converges to $\widetilde{\mu}$ when restricted to $(S\times\mathbb{N}) \cup (\mathbb{N}\times S)$ (for example for the topology where we look at the maximum difference of the measure assigned to any set in $(S\times\mathbb{N}) \cup (\mathbb{N}\times S)$). Therefore the corresponding cut norms also converge. This shows that we may assume that $\widetilde{\mu}$ is supported on $(\widehat{S}\times\widehat{S}) \cup ((\mathbb{N}\backslash\widehat{S}) \times (\mathbb{N}\backslash\widehat{S}))$ for some $K \in \mathbb{N}$ when proving (7).

Let $\widetilde{\mu}'$ be the restriction of $\widetilde{\mu}$ to $\widehat{S}\times\widehat{S}$. Then $\|\widetilde{W}_1 - \widetilde{W}_2\|_{\square,\mathbb{N}\times\mathbb{N},\widetilde{\mu}} = \|\widehat{W}_1 - \widehat{W}_2\|_{\square,\widehat{S}\times\widehat{S},\widetilde{\mu}'}$ where $\widehat{\mathcal{W}_i} = (\widehat{W_i}, \widehat{\mathscr{S}_i})$ is the trivial extensions of $\mathcal{W}_i$ to the measure space $\widehat{\mathscr{S}_i}$ associated with $\widehat{S}_i$. We will prove that we may assume without loss of generality that $\widetilde{\mu}'$ corresponds to a permutation of $\widehat{S}$. By choosing $M \in \mathbb{N}$ sufficiently large we can approximate $\|\widehat{W}_1 - \widehat{W}_2\|_{\square,\widetilde{\mu}'}$ arbitrarily well by replacing $\widetilde{\mu}'$ with a measure such that each element $(i,j) \in \widehat{S}\times\widehat{S}$ has a measure which is an integer multiple of c/M; hence we may assume $\widetilde{\mu}'$ is on this form. Each such $\widetilde{\mu}'$ can be described in terms of a permutation σ' of $[KM]$ via $\widetilde{\mu}'((i,j)) = \sum_{\ell=1}^{KM} c/M\delta_{i,\lceil \ell/M\rceil}\delta_{j,\lceil\sigma'(\ell)/M\rceil}$. Let $\widehat{\mathcal{W}_i'} = (\widehat{W_i'}, [KM])$ be the graphon such that each $j \in [KM]$ has measure c/M, and such that $\widehat{W_i'} = (\widehat{W_i})^\phi$ for the measure-preserving map $\phi\colon [KM] \to [K]$ defined by $\phi(j) := \lceil j/M\rceil$. Using Proposition 39 and the above observation on describing $\widetilde{\mu}'((i,j))$ in terms of a permutation σ' of $[KM]$ we see that $\|\widehat{W}_1 - \widehat{W}_2\|_{\square,\widetilde{\mu}'} = \|\widehat{W}_1' - (\widehat{W}_2')^{\sigma'}\|_\square$. Upon replacing $\widehat{W}_i$ by $\widehat{W}_i'$ throughout the proof, we may assume that the measure $\widetilde{\mu}'$ is a permutation.

To complete the proof it is therefore sufficient to consider some permutation $\widehat{\sigma}$ of $\widehat{S}$ and prove that we can find a permutation σ of $\widehat{S}$ mapping S to S such that

$$\|\widehat{W}_1 - \widehat{W}_2^{\widehat{\sigma}}\|_\square \geq \|\widehat{W}_1 - \widehat{W}_2^{\sigma}\|_\square. \tag{8}$$

We modify the permutation $\widehat{\sigma}$ step by step to obtain a permutation mapping S to S. Abusing notation slightly we let $\widehat{\sigma}$ and σ denote the old and new, respectively, permutations in a single step. In each step choose $i_1, i_2 \leq n$ and $j_1, j_2 > n$ such that $\widehat{\sigma}(i_1) = j_1$ and $\widehat{\sigma}(j_2) = i_2$; if such i_1, i_2, j_1, j_2 do not exist we know that $\widehat{\sigma}$ maps S to S. Then define $\sigma(i_1) := i_2$ and $\sigma(j_2) := j_1$, and for $k \notin \{i_1, j_2\}$ define $\sigma(k) := \widehat{\sigma}(k)$. We have $\|\widehat{W}_1 - \widehat{W}_2^{\sigma}\|_\square \leq \|\widehat{W}_1 - \widehat{W}_2^{\widehat{\sigma}}\|_\square$ by the following argument. Let $U, V \subseteq \mathbb{N}$ be such that $\|\widehat{W}_1 - \widehat{W}_2^{\sigma}\|_\square = |\int_{U\times V} \left(\widehat{W}_1 - \widehat{W}_2^{\sigma}\right) dx\, dy|$. Since $\sigma(j_2) > n$ (implying that both $\widehat{W}_1$ and $\widehat{W}_2^{\sigma}$ are trivial on $(j_2\times\mathbb{N})$ and $(\mathbb{N}\times j_2)$)

the following identity holds if we define $U' := U\backslash\{j_2\}$ or $U' := U \cup \{j_2\}$, and if we define $V' := V\backslash\{j_2\}$ or $V' := V \cup \{j_2\}$:

$$\int_{U\times V} \left(\widehat{W}_1 - \widehat{W}_2^{\sigma}\right) dx\, dy = \int_{U'\times V'} \left(\widehat{W}_1 - \widehat{W}_2^{\sigma}\right) dx\, dy.$$

In other words, $\int_{U\times V}(\widehat{W}_1 - \widehat{W}_2^{\sigma})\, dx\, dy$ is invariant under adding or removing j_2 from U and/or V. Therefore we may assume without loss of generality that

$$j_2 \in U \text{ iff } i_1 \in U, \qquad j_2 \in V \text{ iff } i_1 \in V, \tag{9}$$

since if (9) is not satisfied we may redefine U and V such that (9) holds, and we still have $\|\widehat{W}_1 - \widehat{W}_2^{\sigma}\|_\square = |\int_{U\times V}(\widehat{W}_1 - \widehat{W}_2^{\sigma})\, dx\, dy|$. The assumption (9) implies that $\int_{U\times V}(\widehat{W}_1 - \widehat{W}_2^{\sigma})\, dx\, dy = \int_{U\times V}(\widehat{W}_1 - \widehat{W}_2^{\widehat{\sigma}})\, dx\, dy$, which implies (8) since we can obtain a permutation σ mapping S to S in finitely many steps as described above.

Now we will prove the lemma for general graphons. We will reduce the problem step by step to a problem with additional conditions on the measure spaces involved, until we have reduced the problem to the special case considered above.

First we show that we may assume $\mathscr{S}_i$ and $\widetilde{\mathscr{S}_i}$ are non-atomic. Define $S_i' := S_i \times [0,1]$ and $\widetilde{S}_i' := \widetilde{S}_i \times [0,1]$, let $\mathscr{S}_i'$ and $\widetilde{\mathscr{S}_i'}$ be the corresponding atomless product measure spaces when $[0,1]$ is equipped with Lebesgue measure, and let $\mathcal{W}_i' = (W_i', \mathscr{S}_i')$ and $\widetilde{\mathcal{W}}_i' = (\widetilde{W}_i', \widetilde{\mathscr{S}_i'})$ be graphons such that $W_i' = (W_i)^{\pi_i^1}$ and $\widetilde{W}_i' = (\widetilde{W}_i)^{\widetilde{\pi}_i^1}$, where $\pi_i^1\colon \mathscr{S}_i' \to \mathscr{S}_i$ and $\widetilde{\pi}_i^1\colon \widetilde{\mathscr{S}_i'} \to \widetilde{\mathscr{S}_i}$ are the projection maps on the first coordinates. By considering the natural coupling of $\widetilde{S}_i'$ and $\widetilde{S}_i$ it is clear that $\delta_\square(\widetilde{W}_i', \widetilde{W}_i) = 0$. It therefore follows from the triangle inequality that $\delta_\square(\widetilde{W}_1, \widetilde{W}_2) = \delta_\square(\widetilde{W}_1', \widetilde{W}_2')$. Similarly, $\delta_\square(W_1, W_2) = \delta_\square(W_1', W_2')$. In order to prove that $\delta_\square(\widetilde{W}_1, \widetilde{W}_2) = \delta_\square(W_1, W_2)$ it is therefore sufficient to prove that $\delta_\square(\widetilde{W}_1', \widetilde{W}_2') = \delta_\square(W_1', W_2')$. Since $\mathscr{S}_i'$ and $\widetilde{\mathscr{S}_i'}$ are atomless and $\widetilde{W}_i'$ is a trivial extension of W_i' it is therefore sufficient to prove the lemma for atomless measure spaces.

Next we will reduce the general case to the case when $\widetilde{\mu}_i(\widetilde{S}_i) = \infty$. If $\widetilde{\mu}_i(\widetilde{S}_i) < \infty$ we extend $\widetilde{\mathscr{S}_i}$ to a space $\widehat{\mathscr{S}_i}$ of infinite measure, and let $\widehat{\mathcal{W}_i}$ be the trivial extension of $\widetilde{\mathcal{W}_i}$ to $\widehat{\mathscr{S}_i}$. Assuming we have proved the lemma for the case when the extended measure spaces have infinite measure, it follows that

$$\delta_\square(\mathcal{W}_1, \mathcal{W}_2) = \delta_\square(\widehat{\mathcal{W}_1}, \widehat{\mathcal{W}_2}) = \delta_\square(\widetilde{\mathcal{W}_1}, \widetilde{\mathcal{W}_2});$$

hence the lemma also holds for the case when $\widetilde{\mu}_i(\widetilde{S}_i) < \infty$.

Next we prove that we may assume $\mu_i(S_i) < \infty$. We proceed similarly as in the previous paragraph, and assume $\mu_i(S_i) = \infty$. By Lemma 36 we may assume W_i are supported on sets of finite measure, and we let $\widehat{\mathscr{S}_i} = (\widehat{S}_i, \widehat{\mathcal{S}}_i, \widehat{\mu}_i)$ be a restriction of $\mathscr{S}_i$ such that $\mathrm{supp}(W_i) \subseteq \widehat{S}_i \times \widehat{S}_i$ and $\widehat{\mu}_i(\widehat{S}_i) < \infty$. Since $\mathscr{S}_i$ is non-atomic we may assume $\widehat{\mu}_1(\widehat{S}_1) = \widehat{\mu}_2(\widehat{S}_2)$. Define the graphon $\widehat{\mathcal{W}_i} = (\widehat{W}_i, \widehat{\mathscr{S}_i})$ to be such that $\mathcal{W}_i$ is the trivial extension of $\widehat{\mathcal{W}_i}$ to $\mathscr{S}_i$. Assuming we have proved the lemma for the case when $\mu_i(S_i) < \infty$, it follows that

$$\delta_\square(\mathcal{W}_1, \mathcal{W}_2) = \delta_\square(\widehat{\mathcal{W}_1}, \widehat{\mathcal{W}_2}) = \delta_\square(\widetilde{\mathcal{W}_1}, \widetilde{\mathcal{W}_2});$$

hence the lemma also holds for the case when $\mu_i(S_i) < \infty$.

Next we will prove that we may assume W_i is a step function for $i = 1, 2$, such that each step has the same measure $c > 0$. Step functions are dense in L^1, and hence it is immediate from Lemma 36 that we may assume W_i is a step function. We may assume that the measure of each step is a rational multiple of $\mu_i(S_i)$; if this is not the case we may adjust the steps slightly (because $\mathscr{S}_i$ is non-atomic, we can choose subsets of the steps of any desired measures, by Exercise 2 from §41 in the book of Halmos, 1974) to obtain this. Assuming each step has a measure which is a rational multiple of $\mu_i(S_i)$ we may subdivide each step such that each step obtains the same measure $c > 0$, again using the exercise in the book by Halmos (1974).

Assume W_i are step functions consisting of $k \in \mathbb{N}$ steps each having measure $c > 0$, and that $\mu_i(S_i) < \infty$ and $\widetilde{\mu}_i(\widetilde{S}_i) = \infty$. Let $\mathcal{W}_i' = (W_i', [n])$ (resp. $\widetilde{\mathcal{W}}_i' = (\widetilde{W}_i', \mathbb{N})$) be a graphon over $[n] := \{1, \dots, n\}$ (resp. $\mathbb{N}$) equipped with the discrete σ-algebra, such that each $j \in [n]$ (resp. $j \in \mathbb{N}$) has measure c, and such that $W_i = (W_i')^{\phi_i}$ (resp. $\widetilde{W}_i = (\widetilde{W}_i')^{\widetilde{\phi}_i}$) for a measure-preserving map $\phi_i \colon S_i \to [k]$ (resp. $\widetilde{\phi}_i \colon \widetilde{S}_i \to \mathbb{N}$). Then $\delta_\square(\mathcal{W}_1, \mathcal{W}_2) = \delta_\square(\mathcal{W}_1', \mathcal{W}_2')$ and $\delta_\square(\widetilde{\mathcal{W}}_1, \widetilde{\mathcal{W}}_2) = \delta_\square(\widetilde{\mathcal{W}}_1', \widetilde{\mathcal{W}}_2')$. By the special case we considered in the first paragraphs of the proof, $\delta_\square(\mathcal{W}_1', \mathcal{W}_2') = \delta_\square(\widetilde{\mathcal{W}}_1', \widetilde{\mathcal{W}}_2')$. Combining the above identities, our desired result $\delta_\square(\mathcal{W}_1, \mathcal{W}_2) = \delta_\square(\widetilde{\mathcal{W}}_1, \widetilde{\mathcal{W}}_2)$ follows.

To prove the result for the metric δ_p, we follow the steps above. The only place where the proof differs is in the proof of (8). Let σ, $\widehat{\sigma}$, and i_1, i_2, j_1, j_2 be as in the proof of (8). We would like to show that

$$\|\widehat{W}_1 - \widehat{W}_2^{\sigma}\|_p^p \le \|\widehat{W}_1 - \widehat{W}_2^{\widehat{\sigma}}\|_p^p.$$

Writing both sides as a sum over $(i, j) \in \widehat{S}^2$ we consider the following three cases separately: (i) $(i, j) \in (\widehat{S} \setminus \{i_1, j_2\})^2$, (ii) $(i, j) \in \{i_1, j_2\} \times (\widehat{S} \setminus \{i_1, j_2\})$ or $(i, j) \in (\widehat{S} \setminus \{i_1, j_2\}) \times \{i_1, j_2\}$, and (iii) $(i, j) \in \{i_1, j_2\} \times \{i_1, j_2\}$. In case (i) the terms are identical on the left side and on the right side. For dealing with case (ii) it is sufficient to prove that for an arbitrary $i \in (\widehat{S} \setminus \{i_1, j_2\})$,

$$\begin{aligned}|\widehat{W}_1(i_1, i) - \widehat{W}_2^{\sigma}(i_1, i)|^p + |\widehat{W}_1(j_2, i) - \widehat{W}_2^{\sigma}(j_2, i)|^p \\ \le |\widehat{W}_1(i_1, i) - \widehat{W}_2^{\widehat{\sigma}}(i_1, i)|^p + |\widehat{W}_1(j_2, i) - \widehat{W}_2^{\widehat{\sigma}}(j_2, i)|^p.\end{aligned}$$

This is equivalent to

$$|\widehat{W}_1(i_1, i) - \widehat{W}_2(i_2, \widehat{\sigma}(i))|^p \le |\widehat{W}_1(i_1, i)|^p + |\widehat{W}_2(i_2, \widehat{\sigma}(i))|^p.$$

This inequality is obviously true if either $p = 1$ or both $\widehat{W}_1$ and $\widehat{W}_2$ are non-negative. For case (iii) we need to show that

$$\sum_{i,j \in \{i_1, j_2\}} |\widehat{W}_1(i, j) - \widehat{W}_2^{\sigma}(i, j)|^p \le \sum_{i,j \in \{i_1, j_2\}} |\widehat{W}_1(i, j) - \widehat{W}_2^{\widehat{\sigma}}(i, j)|^p.$$

Writing out both sides we see that this is equivalent to

$$|\widehat{W}_1(i_1, i_1) - \widehat{W}_2(i_2, i_2)|^p \le |\widehat{W}_1(i_1, i_1)|^p + |\widehat{W}_2(i_2, i_2)|^p,$$

which is again true if either $p = 1$ or both $\widehat{W}_1$ and $\widehat{W}_2$ are non-negative. ■

Proof of Proposition 6 and Proposition 32 The existence of a coupling follows from Lemma 34 with $D_1 = D_2 = \emptyset$.

To prove the next statement of the proposition, i.e., that the value of $\delta_\square(\mathcal{W}_1, \mathcal{W}_2)$ is independent of the extensions $\widetilde{\mathscr{S}}_i$, we consider alternative extensions $\widehat{\mathscr{S}}_i = (\widehat{S}_i, \widehat{\mathcal{S}}_i, \widehat{\mu}_i)$ of $\mathscr{S}_i$ for $i = 1, 2$, and let $\widehat{\mathcal{W}}_i$ denote the trivial extension of $\mathcal{W}_i$ to $\widehat{\mathscr{S}}_i$. By Lemma 42 it is sufficient to consider the case when $\widetilde{\mu}_i(\widetilde{S}_i \setminus S_i) = \widehat{\mu}_i(\widehat{S}_i \setminus S_i) = \infty$, since if the extensions do not satisfy this property we can extend them to a space of infinite measure, and Lemma 42 shows that the cut norm is unchanged. It is sufficient to prove that, given any coupling measure $\widetilde{\mu}$ on $\widetilde{S}_1 \times \widetilde{S}_2$, we can find a coupling measure $\widehat{\mu}$ on $\widehat{S}_1 \times \widehat{S}_2$, such that

$$\|\widetilde{W}_1^{\pi_1} - \widetilde{W}_2^{\pi_2}\|_{\square, \widetilde{S}_1 \times \widetilde{S}_2, \widetilde{\mu}} = \|\widehat{W}_1^{\pi_1} - \widehat{W}_2^{\pi_2}\|_{\square, \widehat{S}_1 \times \widehat{S}_2, \widehat{\mu}}. \tag{10}$$

By Corollary 40, the left side of (10) only depends on $\widetilde{\mu}$ restricted to $S_1 \times S_2$; in a similar way, the right side only depends on $\widehat{\mu}$ restricted to $S_1 \times S_2$. We therefore can define an appropriate measure $\widehat{\mu}$ on $\widehat{S}_1 \times \widehat{S}_2$ by defining $\widehat{\mu}|_{S_1 \times S_2} = \widetilde{\mu}|_{S_1 \times S_2}$, and extending it to a coupling measure on $\widehat{S}_1 \times \widehat{S}_2$ by Lemma 34; this yields (10).

The function $\delta_\square$ is clearly symmetric and non-negative. To prove that it is a pseudometric it is therefore sufficient to prove that it satisfies the triangle inequality. This is immediate by Lemma 41 and the definition of $\delta_\square$ as given in Definition 5(ii).

The proof for the metric δ_1 follows exactly the same steps.

Using the statements of Lemma 42, Corollary 40, and Lemma 41 for $p > 1$, the above proof of Proposition 6 immediately generalizes to the invariant L^p metric δ_p as long as the graphons in question are non-negative graphons in L^p (in addition to being in L^1, as required by the definition of a graphon). This proves Proposition 32. ■

For two graphons $\mathcal{W} = (W, \mathscr{S})$ and $\widetilde{\mathcal{W}} = (\widetilde{W}, \widetilde{\mathscr{S}})$ for which $\mathscr{S} = \widetilde{\mathscr{S}}$, it is immediate that $\delta_\square(\mathcal{W}, \widetilde{\mathcal{W}}) \le \|W - \widetilde{W}\|_1$. The following lemma gives an analogous bound when W and $\widetilde{W}$ are defined on the same measurable space and $W = \widetilde{W}$, but the measures are not identical. If μ and $\widetilde{\mu}$ are two measures on the same measurable space $(S, \mathcal{S})$ and $a \ge 0$ we write $\mu \le a\widetilde{\mu}$ to mean that $\mu(A) \le a\widetilde{\mu}(A)$ for every $A \in \mathcal{S}$.

Lemma 44 *Let $\mathcal{W} = (W, \mathscr{S})$ with $\mathscr{S} = (S, \mathcal{S}, \mu)$ and $\widetilde{\mathcal{W}} = (W, \widetilde{\mathscr{S}})$ with $\widetilde{\mathscr{S}} = (S, \mathcal{S}, \widetilde{\mu})$ be graphons, and assume there is an $\varepsilon \in (0, 1)$ such that $\mu \le \widetilde{\mu} \le (1 + \varepsilon)\mu$. Then $\delta_\square(\mathcal{W}, \widetilde{\mathcal{W}}) \le 3\varepsilon\|W\|_{1,\mu}$.*

Proof To distinguish between the graphons $\mathcal{W}$ and $\widetilde{\mathcal{W}}$ we will write $\widetilde{\mathcal{W}} = (\widetilde{W}, \widetilde{\mathscr{S}})$ and $\widetilde{\mathscr{S}} = (\widetilde{S}, \widetilde{\mathcal{S}}, \widetilde{\mu})$, but recall throughout the proof that $\widetilde{W} = W$ and $(\widetilde{S}, \widetilde{\mathcal{S}}) = (S, \mathcal{S})$. Define $(S', \mathcal{S}') := (S, \mathcal{S})$, $\mu' := \widetilde{\mu} - \mu$, and $\mathscr{S}' := (S', \mathcal{S}', \mu')$, and let $\mathscr{S}''$ be the disjoint union of $\mathscr{S}$ and $\mathscr{S}'$. Let $\mathcal{W}'' = (W'', \mathscr{S}'')$ be the trivial extension of $\mathcal{W}$ to $\mathscr{S}''$, and note that $\mathcal{W}''$ and $\widetilde{\mathcal{W}}$ are graphons over spaces of equal total measure. Since $\delta_\square(\mathcal{W}'', \mathcal{W}) = 0$ it is sufficient to prove that $\delta_\square(\widetilde{\mathcal{W}}, \mathcal{W}'') \le 3\varepsilon\|W\|_{1,\mu}$. Let $\widehat{\mu}$ be the coupling measure on $\widetilde{S} \times S''$ such that if $\widetilde{A} \in \widetilde{\mathcal{S}}$, $A \in \mathcal{S}$, and $A' \in \mathcal{S}'$, then

$$\widehat{\mu}(\widetilde{A} \times (A \cup A')) = \mu(\widetilde{A} \cap A) + \mu'(\widetilde{A} \cap A'). \tag{11}$$

To complete the proof of the lemma it is sufficient to show that for all measurable sets $U'', V'' \subseteq \widetilde{S} \times S''$,

$$\left| \int_{U'' \times V''} (\widetilde{W}^{\pi_1} - (W'')^{\pi_2}) \, d\widehat{\mu} \, d\widehat{\mu} \right| \leq 3\varepsilon \|W\|_{1,\mu}, \tag{12}$$

where $\pi_1 \colon \widetilde{S} \times S'' \to \widetilde{S}$ (resp. $\pi_2 \colon \widetilde{S} \times S'' \to S''$) is the projection onto the first (resp. second) coordinate of $\widetilde{S} \times S''$. Let $U, V \subseteq \widetilde{S} \times S$ and $U', V' \subseteq \widetilde{S} \times S'$ be such that $U'' = U \cup U'$ and $V'' = V \cup V'$. Recall that since $S' = S$ we may also view U', V' as sets in $\widetilde{S} \times S$, and we denote these sets by U'_S, V'_S, respectively. By first using $W''|_{(S'' \times S'') \setminus (S \times S)} = 0$ while $\widehat{\mu}$-almost surely $\widetilde{W}^{\pi_1} = (W'')^{\pi_2}$ on $(\tilde{S} \times S)^2$, then using $\mu' \leq \varepsilon\mu$, and then using $\widetilde{W} = W$ and (11), we obtain the estimate (12):

$$\begin{aligned}
&\left| \int_{U'' \times V''} \left(\widetilde{W}^{\pi_1} - (W'')^{\pi_2}\right) d\widehat{\mu} \, d\widehat{\mu} \right| \\
&\quad = \left| \int_{U \times V'} \widetilde{W}^{\pi_1} \, d\widehat{\mu} \, d\widehat{\mu} + \int_{U' \times V} \widetilde{W}^{\pi_1} \, d\widehat{\mu} \, d\widehat{\mu} + \int_{U' \times V'} \widetilde{W}^{\pi_1} \, d\widehat{\mu} \, d\widehat{\mu} \right| \\
&\quad \leq \varepsilon \int_{U \times V'_S} |\widetilde{W}|^{\pi_1} \, d\widehat{\mu} \, d\widehat{\mu} + \varepsilon \int_{U'_S \times V} |\widetilde{W}|^{\pi_1} \, d\widehat{\mu} \, d\widehat{\mu} + \varepsilon^2 \int_{U'_S \times V'_S} |\widetilde{W}|^{\pi_1} \, d\widehat{\mu} \, d\widehat{\mu} \\
&\quad \leq 3\varepsilon \|W\|_1.
\end{aligned}$$

■

We close this appendix with a lemma that immediately implies Proposition 21.

Lemma 45 *Let $\varepsilon \geq 0$ and let $\mathcal{W} = (W, (S, \mathcal{S}, \mu))$ and $\mathcal{W}' = (W', (S', \mathcal{S}', \mu'))$ be two graphons with $\delta_\square(\mathcal{W}, \mathcal{W}') \leq \varepsilon^2/2$. Then*

$$\mu(\{D_W > \lambda + 2\varepsilon\}) - \varepsilon \leq \mu'(\{D_{W'} > \lambda + \varepsilon\}) \leq \mu(\{D_W > \lambda\}) + \varepsilon$$

for all $\lambda \geq 0$.

Proof Since the trivial extension of a graphon $\mathcal{W}$ only changes the measure of the set $\{D_W = 0\}$, we may assume without loss of generality that $\mu(S) = \mu'(S')$. Let π_1 and π_2 be the projections from $S \times S'$ onto the two coordinates, let $\varepsilon' > \varepsilon$, and let $\widehat{\mu}$ be a coupling of μ and μ' such that

$$\|W^{\pi_1} - (W')^{\pi_2}\|_{\square,\widehat{\mu}} \leq \frac{(\varepsilon')^2}{2}.$$

By definition of the cut metric, this implies that

$$\left| \int_U (D_W(x) - D_{W'}(x')) \, d\widehat{\mu}(x, x') \right| \leq \frac{(\varepsilon')^2}{2}$$

for all $U \subseteq S \times S'$. Applying this bound for $U = \{(x, x') \in S \times S' \colon D_W(x) - D_{W'}(x') \geq 0\}$ and $U = \{(x, x') \in S \times S' \colon D_W(x) - D_{W'}(x') \leq 0\}$, this implies that

$$\int_{S \times S'} \left| D_W(x) - D_{W'}(x') \right| d\widehat{\mu}(x, x') \leq (\varepsilon')^2,$$

which in turn implies that

$$\widehat{\mu}\left(\{(x, x') \in S \times S' \colon \left|D_W(x) - D_{W'}(x')\right| \geq \varepsilon'\}\right) \leq \varepsilon'.$$

As a consequence

$$\begin{aligned}
\mu(\{D_W > \lambda + 2\varepsilon'\}) - \varepsilon' &\leq \widehat{\mu}(\{(x, x') \colon D_W(x) > \lambda + 2\varepsilon' \text{ and } \left|D_W(x) - D_{W'}(x')\right| < \varepsilon'\}) \\
&\leq \widehat{\mu}(\{(x, x') \colon D_{W'}(x') > \lambda + \varepsilon' \text{ and } \left|D_W(x) - D_{W'}(x')\right| < \varepsilon'\}) \\
&\leq \mu'(\{D_{W'} > \lambda + \varepsilon'\}).
\end{aligned}$$

Taking $\varepsilon' \downarrow \varepsilon$ and using monotone convergence we obtain the first inequality in the statement of the lemma. The second is proved in the same way. ■

Proof of Proposition 21 Let $\varepsilon_n = \delta_\square(\mathcal{W}_n, \mathcal{W})$, and choose n large enough so that $\varepsilon_n < \lambda$. By Lemma 45,

$$\mu_n(\{D_{W_n} > \lambda\}) \leq \mu(\{D_W > \lambda - \varepsilon_n\}) + \varepsilon_n.$$

Since $\mu(\{D_W > \lambda\})$ is assumed to be continuous at λ, this gives

$$\limsup_{n \to \infty} \mu_n(\{D_{W_n} > \lambda\}) \leq \mu(\{D_W > \lambda\}).$$

The matching lower bound on the lim inf is proved in the same way. ■

Appendix B. Representation of Graphons Over $\mathbb{R}_+$

In this appendix we will prove that every graphon is equivalent to a graphon over $\mathbb{R}_+$ (Proposition 10), and prove that under certain assumptions on the underlying measure space of a graphon the cut metric can be defined in a number of equivalent ways (Proposition 48).

The first statement of the following lemma is a generalization of the analogous result for probability spaces, which was considered by Borgs, Chayes, and Lovász (2010, Corollary 3.3) and Janson (2013, Lemma 7.3). It will be used to prove Theorem 15 and Proposition 10. We proceed similarly to the proof by Janson (2013, Lemma 7.3), but in this case we also need to argue that underlying measure space of the constructed graphon is σ-finite, and we include an additional result on atomless measure spaces.

Lemma 46 *Every graphon $\mathcal{W} = (W, \mathscr{S})$ with $\mathscr{S} = (S, \mathcal{S}, \mu)$ is a pullback by a measure-preserving map of a graphon on some σ-finite Borel measure space. If $\mathscr{S}$ is atomless, the σ-finite Borel space can be taken to be atomless as well.*

Proof Let $S_0 := \emptyset$, and let $(S_k)_{k \in \mathbb{N}}$ be such that for each $k \in \mathbb{N}$, we have $S_k \in \mathcal{S}$, $S_k \subseteq S_{k+1}$, $\mu(S_k) < \infty$, and $\bigcup_{k \in \mathbb{N}} S_k = S$. We claim that we can find a sequence of sets $(A_i)_{i \in \mathbb{N}}$ satisfying the following properties: (i) if $\mathcal{A} := \{A_i : i \in \mathbb{N}\}$ and $\mathcal{S}_0 := \sigma(\mathcal{A})$, then W is $\mathcal{S}_0 \times \mathcal{S}_0$-measurable, (ii) for all $k \in \mathbb{N}$ there exists $i \in \mathbb{N}$ such that $A_i = S_k \setminus S_{k-1}$, (iii) for each $i \in \mathbb{N}$ there exists a $k \in \mathbb{N}$ such that $A_i \subseteq S_k \setminus S_{k-1}$, and (iv) $\bigcup_{i \in \mathbb{N}} A_i = S$. A set $\mathcal{A}$ satisfying (i) can be constructed by noting that each level set $\{(x_1, x_2) \in S \times S : W(x_1, x_2) < q\}$

for $q \in \mathbb{Q}$ is measurable with respect to $\sigma(\sigma(\mathcal{A}_q) \times \sigma(\mathcal{A}_q))$ for some countable set $\mathcal{A}_q$ (this follows, for example, by Lemma 3.4 in the paper of Borgs, Chayes, and Lovász, 2010). By adding the sets $S_k \setminus S_{k-1}$ to $\mathcal{A}$ we obtain a collection of sets satisfying (i) and (ii). Given a set $\widetilde{\mathcal{A}}$ satisfying (i) and (ii), we can easily obtain an $\mathcal{A}$ satisfying (i)–(iii) by replacing each $A \in \widetilde{\mathcal{A}}$ with the countable collection of sets $\{A \cap (S_k \setminus S_{k-1}) : k \in \mathbb{N}\}$. Finally, (ii) implies (iv).

Let $\mathcal{C} = \{0,1\}^\infty$ be the Cantor cube equipped with the product topology, and define $\phi\colon S \to \mathcal{C}$ by $\phi(x) := (\mathbf{1}_{x \in A_i})_{i \in \mathbb{N}}$. Let ν be the pushforward measure of μ onto $\mathcal{C}$ equipped with the Borel σ-algebra. We claim that ν is a σ-finite measure on $\mathcal{C}$. For each $k \in \mathbb{N}$ define $\widehat{C}_k := \{(a_i)_{i\in\mathbb{N}} \in \mathcal{C} : a_i = 0 \text{ if } A_i \not\subseteq S_k \setminus S_{k-1}\}$ and $\widetilde{C}_k := (\bigcup_{i \le k} \widehat{C}_i) \cup \widehat{C}_0$, where $\widehat{C}_0 := \mathcal{C} \setminus (\bigcup_{i\in\mathbb{N}} \widehat{C}_i) \subseteq \mathcal{C} \setminus \phi(S)$, and observe that all the subsets of $\mathcal{C}$ just defined are measurable. The claim will follow if we can prove that (a) $\nu(\widetilde{C}_k) < \infty$ for each $k \in \mathbb{N}$, and (b) $\bigcup_{k\in\mathbb{N}} \widetilde{C}_k = \mathcal{C}$. Property (a) is immediate since from the definition of ν, the fact $\nu(\widehat{C}_0) = 0$, and the properties (ii) and (iii) of $\mathcal{A}$, which imply that $\nu(\widehat{C}_k) = \mu(\phi^{-1}(\widehat{C}_k)) = \mu(S_k \setminus S_{k-1}) < \infty$. To prove (b) let $(a_i)_{i\in\mathbb{N}} \in \mathcal{C}$. We want to prove that $(a_i)_{i\in\mathbb{N}} \in \widetilde{C}_k$ for some $k \in \mathbb{N}$. If $(a_i)_{i\in\mathbb{N}} \notin \phi(S)$ we have $x \in \widehat{C}_0$, so $(a_i)_{i\in\mathbb{N}} \in \widetilde{C}_k$ for all $k \in \mathbb{N}$. If $(a_i)_{i\in\mathbb{N}} = \phi(x)$ for some $x \in S$, then $x \in S_k \setminus S_{k-1}$ for exactly one $k \in \mathbb{N}$, so $(a_i)_{i\in\mathbb{N}} = \phi(x) \in \widehat{C}_k \subseteq \widetilde{C}_k$.

The argument in the following paragraph is similar to the proof by Janson (2013, Lemma 7.3), but we repeat it for completeness. Since the σ-field on S generated by ϕ equals $\mathcal{S}_0$, the σ-field on $S \times S$ generated by $(\phi,\phi)\colon S^2 \to \mathcal{C}^2$ equals $\mathcal{S}_0 \times \mathcal{S}_0$. Since W is measurable with respect to $\mathcal{S}_0 \times \mathcal{S}_0$ we may use the Doob-Dynkin Lemma (see, for example, the book of Kallenberg, 2002, Lemma 1.13) to conclude that there exists a measurable function $V\colon \mathcal{C}^2 \to [0,1]$ such that $W = V^\phi$. We may assume V is symmetric upon replacing it by $\frac{1}{2}(V(x,x') + V(x',x))$. This completes the proof of the main assertion, since V is a graphon on a σ-finite Borel measure space.

Finally we will prove the last claim of the lemma, i.e., that if $\mathscr{S}$ is atomless we may take ν to be atomless as well. To prove this claim it is sufficient to establish that the set $\mathcal{A}$ in the above argument can be modified in such a way that $\nu(x) = 0$ for each $x \in \mathcal{C}$. Given a collection of sets $\mathcal{A}$ satisfying (i)–(iv) above we define a new collection of sets $\widetilde{\mathcal{A}}$ as follows. First define $\mathcal{A}_1 := \mathcal{A}$, and then define $\mathcal{A}_k$ for $k > 1$ inductively as follows. For any $k > 1$ and $A \in \mathcal{A}_{k-1}$, let $A^1 \in \mathcal{S}$ and $A^2 \in \mathcal{S}$ be disjoint sets with union A such that $\mu(A^1) = \mu(A^2) = \frac{1}{2}\mu(A)$; note that such sets A^1 and A^2 can be found since $\mathscr{S}$ is atomless. Then define $\mathcal{A}_k := \{A^1 : A \in \mathcal{A}_{k-1}\} \cup \{A^2 : A \in \mathcal{A}_{k-1}\}$, and finally define $\widetilde{\mathcal{A}} = \bigcup_{k\in\mathbb{N}} \mathcal{A}_k$. Then $\widetilde{\mathcal{A}}$ is countable, satisfies (i)–(iv), and by defining the measure ν using $\widetilde{\mathcal{A}}$ instead of $\mathcal{A}$ we have $\nu(x) = 0$ for every $x \in \mathcal{C}$. Proceeding as above with $\widetilde{\mathcal{A}}$ instead of $\mathcal{A}$ we get $W = V^\phi$, where V is a graphon over an atomless σ-finite Borel space. ■

Proposition 10 follows immediately from the following lemma, whose proof in turn follows a similar strategy as a proof by Janson (2013, Theorem 7.1).

Lemma 47 *Let $\mathcal{W} = (W, \mathscr{S})$ be a graphon over an arbitrary σ-finite space $\mathscr{S}$.*

(i) There are two graphons $\mathcal{W}' = (W', \mathbb{R}_+)$ and $\mathcal{W}'' = (W'', \mathscr{S}'')$ and measure-preserving maps $\phi\colon S \to S''$ and $\phi'\colon [0,\mu(S)) \to S''$ such that $W = (W'')^\phi$ and W' is the trivial extension of $(W'')^{\phi'}$ from $[0,\mu(S))$ to $\mathbb{R}_+$.

(ii) If $\mathscr{S}$ is a Borel measure space, then we can find a measure-preserving map

$$\phi'\colon [0,\mu(S)) \to S$$

such that $\mathcal{W}^{\phi'}$ is a graphon over $[0,\mu(S))$ equipped with the Borel σ-algebra and Lebesgue measure.

(iii) If $\mathscr{S}$ is an atomless Borel measure space, we may take ϕ' in (ii) to be an isomorphism between $\mathscr{S}$ and $[0,\mu(S))$.

Proof We start with the proof of (ii) and (iii). If $\mathscr{S}$ is an atomless Borel space the statement is immediate from Lemma 33. If $\mathscr{S}$ has atoms, we define a graphon $\widetilde{\mathcal{W}} = (\widetilde{W}, \widetilde{\mathscr{S}})$ where $\widetilde{\mathscr{S}} = (\widetilde{S}, \widetilde{\mathcal{S}}, \widetilde{\mu}) = (S \times [0,1], \mathcal{S} \times \mathcal{B}, \mu \times \lambda))$ is the product measure space and $\widetilde{W} := (W)^\pi$, with $\pi\colon S \times [0,1] \to S$ being the projection. Since $\widetilde{\mathscr{S}}$ is an atomless Borel space we may again use Lemma 33, giving the existence of an isomorphism $\psi\colon [0,\widetilde{\mu}(\widetilde{S})) \to \widetilde{S}$ such that $\widetilde{W}^\psi$ is a graphon over $[0,\widetilde{\mu}(\widetilde{S}))$ equipped with Lebesgue measure. Observing that $\widetilde{\mu}(\widetilde{S}) = \mu(S)$, we obtain statement (ii) with $\phi' = \pi \circ \psi$.

To prove (i) we use that by Lemma 46, $\mathcal{W}$ can be expressed as $(\mathcal{W}'')^\phi$ for a graphon $\mathcal{W}''$ on some Borel space $\mathscr{S}'' = (S'', \mathcal{S}'', \mu'')$ and some measure-preserving map $\phi\colon S \to S''$. We then apply the just proven statement (ii) to the graphon $\mathcal{W}''$, and define $\mathcal{W}'$ to be the trivial extension of $(\mathcal{W}'')^{\phi'}$ from $[0,\mu(S))$ to $\mathbb{R}_+$. ■

Proof of Proposition 10 The statement of the proposition follows from Lemma 47 by observing that $\delta_\square(\mathcal{W},\mathcal{W}') \le \delta_\square(\mathcal{W},\mathcal{W}'') + \delta_\square(\mathcal{W}'',\mathcal{W}') = \delta_\square((\mathcal{W}'')^\phi,\mathcal{W}'') + \delta_\square(\mathcal{W}'',(\mathcal{W}'')^{\phi'}) = 0$. ■

The following proposition provides equivalent definitions of the cut metric $\delta_\square$ under certain assumption on the underlying measure spaces. See papers by Borgs, Chayes, Lovász, Sós, and Vesztergombi (2008, Lemma 3.5) and Janson (2013, Theorem 6.9) for analogous results for probability spaces.

Proposition 48 *For $j = 1,2$ let $\mathcal{W}_j = (W_j, \mathscr{S}_j)$ with $\mathscr{S}_j = (S_j, \mathcal{S}_j, \mu_j)$ be a graphon satisfying $\mu_j(S_j) = \infty$. Then the following identities hold, and thus (a)–(e) provide alternative definitions of $\delta_\square$ under certain assumptions on the underlying measure spaces:*

(a) If S_j are Borel spaces, then $\delta_\square(\mathcal{W}_1,\mathcal{W}_2) = \inf_{\psi_1,\psi_2} \|W_1^{\psi_1} - W_2^{\psi_2}\|_\square$, where we take the infimum over measure-preserving $\psi_j\colon \mathbb{R}_+ \to S_j$ for $j = 1,2$, where $\mathbb{R}_+$ is equipped with the Borel σ-algebra and Lebesgue measure.

(b) If S_j are atomless Borel spaces, then $\delta_\square(\mathcal{W}_1,\mathcal{W}_2) = \inf_\psi \|W_1 - W_2^\psi\|_\square$, where we take the infimum over measure-preserving $\psi\colon S_1 \to S_2$.

(c) If S_j are atomless Borel spaces, then $\delta_\square(\mathcal{W}_1,\mathcal{W}_2) = \inf_\psi \|W_1 - W_2^\psi\|_\square$, where we take the infimum over isomorphisms $\psi\colon S_1 \to S_2$.

(d) If $S_j = \mathbb{R}_+$, then $\delta_\square(\mathcal{W}_1,\mathcal{W}_2) = \inf_{\widetilde{\sigma}} \|W_1 - W_2^{\widetilde{\sigma}}\|_\square$, where we take the infimum over all interval permutations $\widetilde{\sigma}$ (i.e., $\widetilde{\sigma}$ maps I_i to $I_{\sigma(i)}$ for some permutation σ of the non-negative integers, and $I_i := [ih, (i+1)h]$ for some $h > 0$).

(e) For $j = 1, 2$ let $(S_j^k)_{k\in\mathbb{N}}$ be increasing sets satisfying $\mu_j(S_j^k) < \infty$ and $\bigcup_{k\in\mathbb{N}} S_j^k = S_j$. Then $\delta_\square(\mathcal{W}_1, \mathcal{W}_2) = \lim_{k\to\infty} \delta_\square(\mathcal{W}_1|_{S_1^k}, \mathcal{W}_2|_{S_2^k})$, where $\mathcal{W}_j|_{S_j^k} := (W_j \mathbf{1}_{S_j^k\times S_j^k}, \mathscr{S}_j^k)$ and $\mathscr{S}_j^k$ is the restriction of $\mathscr{S}^k$ to S_j^k.

Proof of Proposition 48 Let $\delta_\square^{(a)}$, $\delta_\square^{(b)}$, $\delta_\square^{(c)}$, and $\delta_\square^{(d)}$ denote the right sides of the equalities in (a), (b), (c), and (d), respectively. For $j = 1, 2$ fix some arbitrary sequence $(S_j^k)_{k\in\mathbb{N}}$ satisfying $\mu_j(S_j^k) < \infty$ for all $k \in \mathbb{N}$, $S_j^k \subseteq S_j^{k+1}$, and $\bigcup_{k\in\mathbb{N}} S_j^k = S_j$. Define $\delta_\square^{(e)}$ and $\delta_\square^{(e')}$ by

$$\delta_\square^{(e)}(\mathcal{W}_1, \mathcal{W}_2) := \limsup_{k\to\infty} \delta_\square(\mathcal{W}_1^k, \mathcal{W}_2^k) \quad \text{and} \quad \delta_\square^{(e')}(\mathcal{W}_1, \mathcal{W}_2) := \liminf_{k\to\infty} \delta_\square(\mathcal{W}_1^k, \mathcal{W}_2^k),$$

where $\mathcal{W}_k^j = \mathcal{W}_j|_{S_j^k}$. By Lemma 33 it is sufficient to consider the case $\mathscr{S} = (\mathbb{R}_+, \mathcal{B}, \lambda)$ in (b) and (c), since we can consider graphons $(W_j^{\phi_j}, \mathbb{R}_+)$ on $\mathbb{R}_+$, which satisfy $\delta_\square((W_j^{\phi_j}, \mathbb{R}_+), \mathcal{W}_j) = 0$, by using measure-preserving transformations $\phi_j \colon \mathbb{R}_+ \to S_j$. Under this assumption we have $\delta_\square \le \delta_\square^{(b)} \le \delta_\square^{(c)} \le \delta_\square^{(d)}$, since we take the infimum over smaller and smaller sets of maps. By definition, $\delta_\square^{(e')} \le \delta_\square^{(e)}$. To complete the proof of the proposition it is therefore sufficient to prove the following results: (i) $\delta_\square^{(e)} \le \delta_\square \le \delta_\square^{(e')}$ for general σ-finite measure spaces $\mathscr{S}_1, \mathscr{S}_2$ of infinite measure, (ii) $\delta_\square^{(d)} \le \delta_\square$ for $\mathscr{S}_1 = \mathscr{S}_2 = (\mathbb{R}_+, \mathcal{B}, \lambda)$, and (iii) $\delta_\square^{(a)} = \delta_\square^{(c)}$.

We will start by proving (i). Since $\lim_{k\to\infty} \|W_j - W_j \mathbf{1}_{S_j^k\times S_j^k}\|_1 = 0$, Lemma 36 implies that it is sufficient to prove $\delta_\square^{(e)} \le \delta_\square \le \delta_\square^{(e')}$ for the case when $\operatorname{supp}(W_j) \subseteq S_j^k \times S_j^k$ for some $k \in \mathbb{N}$. Under this assumption $\delta_\square(\mathcal{W}_1, \mathcal{W}_2) = \delta_\square(\mathcal{W}_1|_{S_1^{k'}}, \mathcal{W}_2|_{S_2^{k'}})$ for all $k' \ge k$, and (i) follows.

Now we will prove (ii). Since $\lim_{M\to\infty} \|W_j - W_j \mathbf{1}_{|W_j|\le M} \mathbf{1}_{[0,M]^2}\|_1 = 0$ by the dominated convergence theorem, as above we may assume by Lemma 36 that there is an $M > 0$ such that W is bounded and $\operatorname{supp}(W_j) \subseteq [0, M]^2$ for $j = 1, 2$. For $j = 1, 2$ define $\widehat{\mathcal{W}}_j = (\widehat{W}_j, [0, M])$, where $\widehat{W}_j := W_j|_{[0,M]^2}$ is a bounded graphon on $[0, M]^2$. By Lemma 42 in the current paper and Lemma 3.5 of Borgs, Chayes, Lovász, Sós, and Vesztergombi (2008) (or, equivalently, Theorem 6.9 of Janson, 2013),

$$\delta_\square(\mathcal{W}_1, \mathcal{W}_2) = \delta_\square(\widehat{\mathcal{W}}_1, \widehat{\mathcal{W}}_2) = \inf_{\widehat{\sigma}} \|\widehat{W}_1 - \widehat{W}_2^{\widehat{\sigma}}\|_\square \ge \inf_{\widetilde{\sigma}} \|W_1 - W_2^{\widetilde{\sigma}}\|_\square,$$

where $\widehat{\sigma}$ is an interval permutation of $[0, M]$ and $\widetilde{\sigma}$ is an interval permutation of $\mathbb{R}_+$.

Finally we will prove (iii). By Lemma 47 there are measure-preserving maps $\phi_j \colon \mathbb{R}_+ \to S_j$ such that $\delta_\square(\mathcal{W}_j, (\mathcal{W}_j)^{\phi_j}) = 0$. The triangle inequality then implies that $\delta_\square(\mathcal{W}_1, \mathcal{W}_2) = \delta_\square((\mathcal{W}_1)^{\phi_1}, (\mathcal{W}_2)^{\phi_2})$. Since $(\mathcal{W}_1)^{\phi_1}$ and $(\mathcal{W}_2)^{\phi_2}$ are graphons over atomless Borel spaces, it follows that $\delta_\square^{(a)} = \delta_\square^{(c)}$. ∎

Remark 49 *The proof of the above proposition clearly generalizes to the metric δ_1, the only additional ingredient being the analogue of a result by Janson (2013, Theorem 6.9) for the metric δ_1 (Janson, 2013, Remark 6.13). Using the results and proof techniques of Borgs, Chayes, Cohn, and Ganguly (2015, Appendix A) instead of Janson (2013, Theorem 6.9), it can also be generalized to the metric δ_p for $p > 1$, again provided both graphons are non-negative and in L^p.*

We close this appendix by proving Proposition 7. In fact, we will prove a generalization of this proposition for the invariant L^p metric δ_p. The second statement of this proposition involves the distance $\delta_p(\mathcal{W}_1, \mathcal{W}_2)$ of graphons $\mathcal{W}_1 = (W_1, \mathbb{R}_+)$ and $\mathcal{W}_2 = (W_2, \mathbb{R}_+)$ that are not necessarily non-negative, which means we do not have Proposition 32 at our disposal to guarantee that δ_p is well defined. We avoid this problem by defining δ_p as in Proposition 48, i.e., by setting

$$\delta_p(\mathcal{W}_1, \mathcal{W}_2) := \inf_{\phi\colon \mathbb{R}_+\to\mathbb{R}_+} \|W_1 - W_2^\phi\|_p$$

with the infimum going over isomorphisms. Note that by Remark 49, for non-negative graphons in L^p, this definition is equivalent to the one given at the beginning of Appendix A.

Proposition 50 *Let $p \ge 1$, and let $\mathcal{W}_1$ and $\mathcal{W}_2$ be graphons in L^p. Then*

(i) $\delta_1(\mathcal{W}_1, \mathcal{W}_2) = 0$ if and only if $\delta_\square(\mathcal{W}_1, \mathcal{W}_2) = 0$, and

(ii) if $\mathcal{W}_1$ and $\mathcal{W}_2$ are non-negative or graphons over $\mathbb{R}_+$, then $\delta_p(\mathcal{W}_1, \mathcal{W}_2) = 0$ if and only if $\delta_\square(\mathcal{W}_1, \mathcal{W}_2) = 0$.

Proposition 50 (and hence Proposition 7) and Proposition 8 follow from the next proposition.

Proposition 51 *For $i = 1, 2$, let $\mathcal{W}_i = (W_i, \mathscr{S}_i)$ be a graphon over a Borel space $\mathscr{S}_i$ such that $\delta_\square(\mathcal{W}_1, \mathcal{W}_2) = 0$ and $\mu_1(S_1) = \mu_2(S_2)$. Then there exists a measure μ on $S_1 \times S_2$ such that*

(i) $\|W_1^{\pi_1} - W_2^{\pi_2}\|_{\square,\mu} = 0$,

(ii) the first (resp. second) marginal of μ is dominated by μ_1 (resp. μ_2), i.e., for any $A \in \mathcal{S}_1$ (resp. $A \in \mathcal{S}_2$) we have $\mu(A \times S_2) \le \mu_1(A)$ (resp. $\mu(S_1 \times A) \le \mu_2(A)$),

(iii) if $A \in \mathcal{S}_1$ is such that $\mu(A \times S_2) < \mu_1(A)$ then $\mu_1(A \cap E_1) > 0$, where

$$E_i := \left\{ x \in S_i \,:\, \int_{S_i} |W_i(x, x')|\, dx' = 0 \right\} \quad \text{for } i = 1, 2,$$

and the same property holds with the roles of $\mathcal{W}_1$ and $\mathcal{W}_2$ interchanged, and

(iv) in particular, if $\mu_1(E_1) = \mu_2(E_2) = 0$ with E_i as in (iii), then μ is a coupling measure.

Remark 52 *The analogous statement to Proposition 51 for graphons over probability spaces (see, for example, the paper of Janson, 2013, Theorem 6.16) states that when the underlying space is a Borel probability space, the infimum in the definition of the cut distance using couplings is attained. Proposition 51 says that the same result is true in our setting of σ-finite measure spaces if we make two additional assumptions: (a) the cut distance between the graphons is zero, and (b) $\mu_i(E_i) = 0$ for $i = 1, 2$, where E_i is defined in part (iii) of the proposition. We remark that both of these assumptions are necessary; see the examples following this remark.*

Janson (2016, Theorem 5.3) proves a related result, stating that if the cut distance of two graphons over σ-finite Borel spaces is zero, then there are trivial extensions of these graphons such that the extensions can be coupled so as to be equal almost everywhere. Proposition 51 implies a similar result, namely Proposition 8, which states that under this assumption, the restrictions of the two graphons to the sets $S_i \setminus E_i$ can be coupled so that they are equal a.e. To see this, we note that by Proposition 50 two graphons W_1, W_2 with cut distance zero have distance zero in the metric δ_1, which in turn implies that $|W_1|$ and $|W_2|$ have distance zero in δ_1 and hence in $\delta_\square$. By Lemma 45, this in turn implies that $\mu_1(S_1 \setminus E_1) = \mu_2(S_2 \setminus E_2)$, which allows us to use Proposition 51 to deduce the claim.

Example 53 *Assumption (a) in Remark 52 is necessary by the following counterexample, which illustrates that there are graphons $\mathcal{W}_1, \mathcal{W}_2$ over $\mathbb{R}_+$ such that $\|W_1^{\pi_1} - W_2^{\pi_2}\|_{\square,\mu} > \delta_\square(\mathcal{W}_1, \mathcal{W}_2)$ for all coupling measures μ. Let $\mathcal{W}_1 = (W_1, \mathbb{R}_+)$ be an arbitrary graphon such that W_1 is strictly positive everywhere, and let $\mathcal{W}_2 = (W_2, \mathbb{R}_+)$ be defined by $W_2(x,y) = W_1(x-1, y-1)$ for $x, y \geq 1$, $W_2(x,y) = -1$ for $x, y \in [0,1)$, and $W_2(x,y) = 0$ otherwise. First observe that $\delta_\square(\mathcal{W}_1, \mathcal{W}_2) \leq 1$, since if $\phi_n \colon \mathbb{R}_+ \to \mathbb{R}_+$ is defined by $\phi_n(x) = x + 1$ for $x \in [0, n]$, $\phi_n(x) = x - n$ for $x \in (n, n+1]$, and $\phi_n(x) = x$ for $x > n + 1$, then*

$$\lim_{n\to\infty} \|W_1 - W_2^{\phi_n}\|_\square = 1.$$

Then observe that $\|W_1^{\pi_1} - W_2^{\pi_2}\|_{\square,\mu} > 1$ for all coupling measures μ, since if $S = \mathbb{R}_+^2$ and $T = \mathbb{R}_+ \times [0,1]$ then

$$\begin{aligned}\|W_1^{\pi_1} - W_2^{\pi_2}\|_{\square,\mu} &\geq \left| \int_{S\times T} W_1^{\pi_1} - W_2^{\pi_2} \, d\mu \, d\mu \right| \\ &= \left| \int_{\mathbb{R}_+\times T} W_1(x, \pi_1(y)) \, d\lambda \, d\mu - \int_{\mathbb{R}_+\times[0,1]} W_2 \, d\lambda \, d\lambda \right| > 1.\end{aligned}$$

Assumption (b) is necessary by the following counterexample, which illustrates that there are non-negative graphons $\mathcal{W}_1, \mathcal{W}_2$ over $\mathbb{R}_+$ such that $\delta_\square(\mathcal{W}_1, \mathcal{W}_2) = 0$ and $\|W_1 - W_2\|_{\square,\mu} > 0$ for all coupling measures μ. Letting $\mathcal{W}_1$ and $\mathcal{W}_2$ be defined as above, except that $W_2(x,y) = 0$ for all $x, y \in \mathbb{R}_+$ for which either $x < 1$ or $y < 1$, we proceed as in case (a) to conclude that the graphons satisfy the desired property.

Proof of Proposition 51 First we note that for $\mu_1(S_1) = \mu_2(S_2) = 1$, the proposition follows immediately from a result of Janson (2013, Theorem 6.16), which in fact gives μ as a coupling of μ_1 and μ_2. The case $\mu_1(S_1) = \mu_2(S_2) = c < \infty$ with $c \neq 1$ can be reduced to the case $c = 1$ by considering the graphons $\mathcal{W}_i' = (W_i, \mathscr{S}_i')$ where $\mathscr{S}_i'$ is obtained from $\mathscr{S}_i$ by multiplying the measures μ_i by $1/c$, turning them into probability measures. All that is left to consider is therefore the case $\mu_1(S_1) = \mu_2(S_2) = \infty$.

Next we argue that we may assume $\mathscr{S}_i$ is atomless for $i = 1, 2$. Assuming the proposition has been proved for the case of atomless Borel measure spaces, and given graphons $\mathcal{W}_i$ over arbitrary Borel measure spaces, we define graphons $\widetilde{\mathcal{W}}_i$ over the measure space $\widetilde{\mathscr{S}}_i$ defined as the product of $\mathscr{S}_i$ and $[0,1]$, such that $\widetilde{\mathcal{W}}_i = \mathcal{W}_i^{\phi_i}$ for the projection map $\phi_i \colon S_i \times [0,1] \to S_i$. Assume that $\widetilde{\mu}$ is a measure on $\widetilde{S}_1 \times \widetilde{S}_2$ such that the statements of the proposition hold

for $\widetilde{\mu}$ and $\widetilde{\mathcal{W}}_1, \widetilde{\mathcal{W}}_2$. Defining a measure μ for $S_1 \times S_2$ by letting μ be the pushforward of $\widetilde{\mu}$ for the map $\widetilde{S}_1 \times \widetilde{S}_2 \to S_1 \times S_2$ sending $\big((x_1, r_1), (x_2, r_2)\big) \mapsto (x_1, x_2)$, one easily checks that μ is a measure satisfying the conclusions of the proposition for $\mathcal{W}_1$ and $\mathcal{W}_2$. It follows that we may assume the spaces $\mathscr{S}_i$ are atomless Borel measure spaces, and by Lemma 33 we may assume that they are $\mathbb{R}_+$ equipped with the Lebesgue measure; we will make these assumptions in the remainder of the proof. In particular, we will no longer use the notation μ_1 and μ_2 from the proposition statement, since they are now both Lebesgue measure λ; it will be convenient to use the notation μ_n for other purposes.

Consider a sequence of coupling measures $(\mu_n)_{n\in\mathbb{N}}$ such that $\|W_1^{\pi_1} - W_2^{\pi_1}\|_{\square,\mu_n} \to 0$. For any given $M > 0$, let $\mu_n^M = \mu_n|_{[0,M]^2}$. The measures μ_n^M are not necessarily coupling measures, but their marginals are dominated by the Lebesgue measure on $[0, M]$, and they satisfy $\lim_{n\to\infty} \|W_1^{\pi_1} - W_2^{\pi_2}\|_{\square,\mu_n^M} = 0$. Furthermore, as a sequence of measures of uniformly bounded total mass over a compact metrizable space, they have a subsequence that converges in the weak topology (Billingsley, 1999, Theorem 5.1), i.e., in the topology in which the integrals over all continuous functions on $[0, M]^2$ converge. Let μ^M be some subsequential limit, and note that as a limit of a sequence of measures having this property, the marginals of μ^M are dominated by the Lebesgue measure on $[0, M]$ as well. Note also that $\mu_n^M \times \mu_n^M$ converges weakly to $\mu^M \times \mu^M$ along any subsequence on which μ_n^M converges weakly to μ^M (Billingsley, 1999, Theorem 2.8). We will argue that

$$\left| \int_{A\times B} \Big(W_1^{\pi_1} - W_2^{\pi_2} \Big) \, d\mu^M \, d\mu^M \right| = 0 \tag{13}$$

for all measurable subsets $A, B \subseteq [0, M]^2$.

Indeed, given two such subsets and $\varepsilon > 0$, let $\widetilde{W}_i$ be continuous functions over $[0, M]^2$ such that $\|W_i - \widetilde{W}_i\|_{1,\lambda|_{[0,M]^2}} \leq \varepsilon$ for $i = 1, 2$, and let $f, g \colon [0, M]^2 \to [0, 1]$ be continuous functions such that $(\|\widetilde{W}_1\|_\infty + \|\widetilde{W}_2\|_\infty)\|\mathbf{1}_A - f\|_{1,\mu^M} \leq \varepsilon$ and $(\|\widetilde{W}_1\|_\infty + \|\widetilde{W}_2\|_\infty)\|\mathbf{1}_B - g\|_{1,\mu^M} \leq \varepsilon$ (existence of appropriate functions $\widetilde{W}_i, f, g$ follows from, for example, Stroock, 2011b, Corollary 3.2.15). Using the fact that $|\int f(x)g(y)(W_1^{\pi_1}(x,y) - W_2^{\pi_2}(x,y)) \, d\mu_n^M \, d\mu_n^M| \leq \|W_1^{\pi_1} - W_2^{\pi_2}\|_{\square,\mu_n^M}$ and the fact that the marginals of μ_n^M and μ^M are dominated by the Lebesgue measure on $[0, M]$, this allows us to conclude that for some sufficiently large n chosen from the subsequence along which μ_n^M converges,

$$\begin{aligned}\left| \int_{A\times B} \Big(W_1^{\pi_1} - W_2^{\pi_2} \Big) \, d\mu^M \, d\mu^M \right| &\leq \left| \int_{A\times B} \Big(\widetilde{W}_1^{\pi_1}(x,y) - \widetilde{W}_2^{\pi_2}(x,y) \Big) \, d\mu^M \, d\mu^M \right| + 2\varepsilon \\ &\leq \left| \int f(x)g(y) \Big(\widetilde{W}_1^{\pi_1}(x,y) - \widetilde{W}_2^{\pi_2}(x,y) \Big) \, d\mu^M \, d\mu^M \right| + 4\varepsilon \\ &\leq \left| \int f(x)g(y) \Big(\widetilde{W}_1^{\pi_1}(x,y) - \widetilde{W}_2^{\pi_2}(x,y) \Big) \, d\mu_n^M \, d\mu_n^M \right| + 5\varepsilon \\ &\leq \left| \int f(x)g(y) \Big(W_1^{\pi_1}(x,y) - W_2^{\pi_2}(x,y) \Big) \, d\mu_n^M \, d\mu_n^M \right| + 7\varepsilon \\ &\leq \|W_1^{\pi_1} - W_2^{\pi_2}\|_{\square,\mu_n^M} + 7\varepsilon \leq 8\varepsilon.\end{aligned}$$

Since ε was arbitrary, this proves (13).

For each $M \in \mathbb{N}$ we let μ^M be a measure as in the previous paragraph. We may assume the subsequence along which μ_n^{M+1} converges to μ^{M+1} is a subsequence of the subsequence along which μ_n^M converges to μ^M. This implies that $\mu^{M+1}|_{[0,M]^2} = \mu^M$ for all $M \in \mathbb{N}$, so there is a measure μ on $\mathbb{R}_+^2$ such that $\mu|_{[0,M]^2} = \mu^M$. Furthermore, there is a subsequence of $(\mu_n)_{n\in\mathbb{N}}$ converging weakly to μ, such that for any $M \in \mathbb{N}$ the measures $\mu_n|_{[0,M]^2}$ converge weakly to $\mu|_{[0,M]^2}$ along this subsequence, and as a limit of measures with these properties, the measure μ satisfies (ii), as well as

$$\sup_{A,B\subseteq\mathbb{R}_+^2} \left| \int_{A\times B} W_1^{\pi_1} - W_2^{\pi_2} \, d\mu \, d\mu \right| = 0,$$

where, *a priori*, the supremum is over measurable, bounded subsets $A, B \subset \mathbb{R}_+^2$. But it is easy to see that if the supremum over these sets is zero, then the same holds for the supremum over all measurable subsets $A, B \subseteq \mathbb{R}_+^2$ (use (ii) to conclude that the integrand is in L^1, which means it can be approximated by functions over bounded subsets of $\mathbb{R}_+^4$). The property (i) of μ follows.

It remains to prove that μ satisfies (iii), since (iv) follows immediately from (iii). Recall that the by definition of the measures μ_n,

$$\sup_{A_1,A_2,K} \left| \int_{(A_1\times[K,\infty))\times(A_2\times\mathbb{R}_+)} W_1^{\pi_1} - W_2^{\pi_2} \, d\mu_n \, d\mu_n \right| \to 0,$$

where the supremum is over $A_1, A_2 \subseteq \mathbb{R}_+$ and $K \geq 0$. Fix any $\varepsilon > 0$, and observe that for all $K > 1$ sufficiently large,

$$\sup_{A_1,A_2} \left| \int_{(A_1\times[K,\infty))\times(A_2\times\mathbb{R}_+)} W_2^{\pi_2} \, d\mu_n \, d\mu_n \right| \leq \sup_{A_1,A_2} \int_{[K,\infty)\times\mathbb{R}_+} |W_2| \, d\lambda \, d\lambda < \varepsilon,$$

so

$$\begin{aligned}\limsup_{n\to\infty} \sup_{A_1,A_2} &\left| \int_{(A_1\times[K,\infty))\times A_2} W_1(\pi_1(x), x') \, d\mu_n(x) \, d\lambda(x') \right| \\ &= \limsup_{n\to\infty} \sup_{A_1,A_2} \left| \int_{(A_1\times[K,\infty))\times(A_2\times\mathbb{R}_+)} W_1^{\pi_1} \, d\mu_n \, d\mu_n \right| < \varepsilon.\end{aligned}$$

Fix any $A_1, A_2 \subseteq \mathbb{R}_+$, and observe from the above that for K sufficiently large,

$$\limsup_{n\to\infty} \left| \int_{A_1\times A_2} W_1(x, x') \, d(\lambda - \mu_n^{1,K})(x) \, d\lambda(x') \right| < \varepsilon,$$

where $\mu_n^{1,K}$ is the projection of $\mu_n|_{\mathbb{R}_+\times[0,K]}$ onto the first coordinate. Choose $\widetilde{W}_1 \in C_c(\mathbb{R}_+^2)$ such that $\|\widetilde{W}_1 - W_1\|_1 < \varepsilon$, where $C_c(\mathbb{R}_+^2)$ is the space of continuous and compactly supported functions on $\mathbb{R}_+^2$ (the existence of such a function follows again from, for example, Stroock, 2011b, Corollary 3.2.15). For all $K > 1$ sufficiently large,

$$\left| \int_{A_1\times A_2} \widetilde{W}_1(x, x') \, d(\mu^1 - \mu^{1,K})(x) \, d\lambda(x') \right| < \varepsilon,$$

where $\mu^{1,K}$ (resp. μ^1) is the projection of $\mu|_{\mathbb{R}_+\times[0,K]}$ (resp. μ) onto the first coordinate. Next we claim that $\mu_n^{1,K}|_{[0,K']}$ converges weakly to $\mu^{1,K}|_{[0,K']}$ for any $K, K' > 0$. To see that, we need to show that for any continuous function $f\colon [0, K'] \to \mathbb{R}_+$ the associated integral converges when $n \to \infty$. To this end, we approximate the function $(x, x') \mapsto f(x)\mathbf{1}_{x'\in[0,K]}$ (with $f(x) = 0$ for $x > K'$) by a function $g\colon \mathbb{R}_+^2 \to \mathbb{R}$, where $g(x, x') = \widehat{f}(x)\chi(x')$, $\widehat{f}\colon \mathbb{R}_+ \to \mathbb{R}_+$ is a continuous function with support in $[0, K']$ approximating f and satisfying $\|\widehat{f}\|_\infty \leq \|f\|_\infty$, and $\chi\colon \mathbb{R}_+ \to [0, 1]$ is a continuous function with support in $[0, K]$ approximating the indicator function of the set $[0, K]$. Since $\widehat{f}$ and χ can be chosen to be arbitrarily close approximations in the L^1 norm and the marginals of μ_n are given by Lebesgue measure, this implies the claim. Therefore we can find $n_K \in \mathbb{N}$ depending on K, such that for all $n \geq n_K$

$$\left| \int_{A_1\times A_2} \widetilde{W}_1(x, x') \, d(\mu^{1,K} - \mu_n^{1,K})(x) \, d\lambda(x') \right| < \varepsilon.$$

Combining the above estimates and using the triangle inequality, we get that for sufficiently large K and $n \geq n_K$,

$$\begin{aligned}\left| \int_{A_1\times A_2} W_1(x, x') \, d(\lambda - \mu^1)(x) \, d\lambda(x') \right| \leq{}& \left| \int_{A_1\times A_2} W_1(x, x') - \widetilde{W}_1(x, x') \, d(\lambda - \mu^1)(x) \, d\lambda(x') \right| \\ &+ \left| \int_{A_1\times A_2} \widetilde{W}_1(x, x') \, d(\lambda - \mu_n^{1,K})(x) \, d\lambda(x') \right| \\ &+ \left| \int_{A_1\times A_2} \widetilde{W}_1(x, x') \, d(\mu_n^{1,K} - \mu^{1,K})(x) \, d\lambda(x') \right| \\ &+ \left| \int_{A_1\times A_2} \widetilde{W}_1(x, x') \, d(\mu^{1,K} - \mu^1)(x) \, d\lambda(x') \right| \\ &< 4\varepsilon.\end{aligned}$$

Since $\varepsilon > 0$ was arbitrary this implies that

$$\left| \int_{A_1\times A_2} W_1(x, x') \, d(\lambda - \mu^1)(x) \, d\lambda(x') \right| = 0.$$

Since $\lambda - \mu^1$ is absolutely continuous with respect to λ, we know by the Radon-Nikodym theorem that there is a non-negative function f such that $d(\lambda - \mu^1)(x) = f(x) \, d\lambda(x)$. The Lebesgue differentiation theorem now says that $W_1(x, x')f(x) = 0$ almost everywhere, which implies (iii). ∎

Proof of Proposition 50 Since $\delta_\square(\mathcal{W}_1, \mathcal{W}_2) \leq \delta_1(\mathcal{W}_1, \mathcal{W}_2)$, we only need to prove that $\delta_\square(\mathcal{W}_1, \mathcal{W}_2) = 0$ implies $\delta_1(\mathcal{W}_1, \mathcal{W}_2) = 0$ in order to prove (i). Assume first that the graphons are over $\mathbb{R}_+$, and let μ be as in Proposition 51. Then $W_1^{\pi_1} - W_2^{\pi_2} = 0$ μ-almost everywhere. For each $n \in \mathbb{N}$ let μ_n be some arbitrary coupling measure on $S_1 \times S_2$ such that $\mu_n|_{[0,n]^2} = \mu|_{[0,n]^2}$. Then $\lim_{n\to\infty} \|W_1^{\pi_1} - W_2^{\pi_2}\|_{1,\mu_n} = 0$, so $\delta_1(\mathcal{W}_1, \mathcal{W}_2) = 0$. To obtain the result for graphons over general measure spaces we use Proposition 10, the triangle inequality, and the fact that two graphons have distance zero for $\delta_\square$ and δ_1 if one is a pullback of the other.

For (ii) with graphons over $\mathbb{R}_+$ and $\delta_\square$ defined in terms of measure-preserving transformations, we will first prove that $\delta_\square(\mathcal{W}_1, \mathcal{W}_2) = 0$ implies $\delta_p(\mathcal{W}_1, \mathcal{W}_2) = 0$. This follows by the exact same argument as in the preceding paragraph, i.e., by using Proposition 51 to construct a measure μ and coupling measures μ_n on $S_1 \times S_2$.

Now we will prove that $\delta_p(\mathcal{W}_1, \mathcal{W}_2) = 0$ implies $\delta_\square(\mathcal{W}_1, \mathcal{W}_2) = 0$, still assuming the graphons are over $\mathbb{R}_+$ and that δ_p is defined in terms of measure-preserving transformations. By part (i) it is sufficient to show that $\delta_p(\mathcal{W}_1, \mathcal{W}_2) = 0$ implies $\delta_1(\mathcal{W}_1, \mathcal{W}_2) = 0$. Fix $\varepsilon > 0$ and let $A_1, A_2 \subseteq \mathbb{R}_+$ be such that $\|W_i - W_i \mathbf{1}_{A_i \times A_i}\|_1 < \varepsilon/2$ and $M := \lambda(A_1) + \lambda(A_2) < \infty$. By Hölder's inequality, for any isomorphism $\phi\colon \mathbb{R}_+ \to \mathbb{R}_+$ and $A := A_1 \cup \phi^{-1}(A_2)$,

$$\begin{aligned}\|W_1 - W_2^\phi\|_1 &\leq \|(\mathbf{1} - \mathbf{1}_{A\times A})(W_1 - W_2^\phi)\|_1 + \|(W_1 - W_2^\phi)\mathbf{1}_{A\times A}\|_1 \\ &\leq \varepsilon + \|W_1 - W_2^\phi\|_p \cdot M^{2-2/p}.\end{aligned}$$

Taking the infimum over ϕ we see that $\delta_1(\mathcal{W}_1, \mathcal{W}_2) \leq \varepsilon + M^{2-2/p}\delta_p(\mathcal{W}_1, \mathcal{W}_2) = \varepsilon$. Since ε was arbitrary, this shows that $\delta_1(\mathcal{W}_1, \mathcal{W}_2) = 0$.

We get (ii) for non-negative graphons and δ_p defined in terms of couplings by using Proposition 48 and Remark 49, and to move from graphons over $\mathbb{R}_+$ to graphons over a general σ-finite space we use Lemma 47. ■

Appendix C. Measurability properties of graph processes

Recall the definition of a graph process (Definition 23) and the measurable space of graphs $\mathbb{G}$, as well as what it means for two graph processes to be equal up to relabeling of the vertices (Definition 24). Before stating our main result about the measurability of the relation of being equal up to relabeling, we state and prove the following simple lemma.

Lemma 54 *Let $\mathcal{G} = (G_t)_{t\geq 0}$ be a graph process.*

(i) Let $V \subset V' \subset \mathbb{N}$ and $E \subset E' \subset \binom{\mathbb{N}}{2}$ be finite sets. Then there are increasing sequences $(a_k)_{k\in\mathbb{N}}$ and $(b_k)_{k\in\mathbb{N}}$ of real numbers such that

$$\{t \in \mathbb{R}_+ \colon V' \cap V(G_t) = V \text{ and } E' \cap E(G_t) = E\} = \bigcup_{k\in\mathbb{N}} [a_k, b_k).$$

Furthermore, for any set of the form $T = \bigcup_{k\in\mathbb{N}}[a_k, b_k)$ with a_k, b_k as above, the event that $T = \{t \in \mathbb{R}_+ \colon V' \cap V(G_t) = V \text{ and } E' \cap E(G_t) = E\}$ is measurable.

(ii) For $i = 1, 2$ let $V_i \subset V_i' \subset \mathbb{N}$ and $E_i \subset E_i' \subset \binom{\mathbb{N}}{2}$ be finite sets, let $\mathcal{G}^i = (G_t^i)_{t\geq 0}$ be a graph process, and let $T_i(\mathcal{G}^i)$ be the set of times for which $V_i' \cap V(G_t^i) = V_i$ and $E_i' \cap E(G_t^i) = E_i$. Then the event that $T_1(\mathcal{G}^1) = T_2(\mathcal{G}^2)$ is measurable.

(iii) The event that a specified vertex in $\mathbb{N}$ is isolated for all times is measurable.

(iv) If $\mathcal{G} = (G_t)_{t\geq 0}$ is projective, then the birth time $t_v \in [0, \infty]$ of any vertex $v \in \mathbb{N}$ is measurable, and under the assumptions (5) from Section 2.4, the map defined in (4) is measurable, where the σ-algebra used on the space of measures is defined above (4).

Proof The time set considered in (i) takes the required form since the set of graphs $\{G\colon V' \cap V(G) = V \text{ and } E' \cap E(G) = E\}$ is an open set in $\mathbb{G}$ and since $\mathcal{G} = (G_t)_{t\geq 0}$ is càdlàg. The measurability claim in (i) follows since the event in question occurs if and only if the two time sets have the same intersection with $\mathbb{Q}$. The statement (ii) follows by a similar argument. Both statements (iii) and (iv) immediately follow from (i). ■

Proposition 55 *The event that two graph processes $(G_t)_{t\geq 0}$ and $(\widehat{G}_t)_{t\geq 0}$ are equal up to relabeling is measurable.*

Proof The proposition is immediate in the case where $\bigcup_{t\geq 0} V(G_t)$ or $\bigcup_{t\geq 0} V(\widehat{G}_t)$ is finite, since the set of maps $\phi\colon [n] \to [n]$ is finite, and given any ϕ the event that this map satisfies the requirements of Definition 24 is measurable by Lemma 54. We may therefore assume that both $\bigcup_{t\geq 0} V(G_t)$ and $\bigcup_{t\geq 0} V(\widehat{G}_t)$ are infinite. We may further assume without loss of generality that $\bigcup_{t\geq 0} V(G_t) = \bigcup_{t\geq 0} V(\widehat{G}_t) = \mathbb{N}$; we may do this upon relabeling the vertices of both graph processes.

Next, we reduce the proof of the proposition to the case where no vertices are isolated for all times. For $V \subseteq \mathbb{N}$ let G_t^V denote the induced subgraph of G_t that has vertex set $V \cap V(G_t)$. Let $V_0 \subseteq \mathbb{N}$ (resp. $\widehat{V}_0 \subseteq \mathbb{N}$) denote the set of vertices for $(G_t)_{t\geq 0}$ (resp. $(\widehat{G}_t)_{t\geq 0}$) that are isolated for all times. Then $(G_t)_{t\geq 0}$ and $(\widehat{G}_t)_{t\geq 0}$ are equal up to relabeling of the vertices if and only if this property holds for $(G_t^{V_0})_{t\geq 0}$ and $(\widehat{G}_t^{\widehat{V}_0})_{t\geq 0}$ and for $(G_t^{\mathbb{N}\setminus V_0})_{t\geq 0}$ and $(\widehat{G}_t^{\mathbb{N}\setminus\widehat{V}_0})_{t\geq 0}$. To reduce to the case in which no vertices are isolated for all times, it is sufficient to show measurability of the event that $(G_t^{V_0})_{t\geq 0}$ and $(\widehat{G}_t^{\widehat{V}_0})_{t\geq 0}$ are equal up to relabeling of the vertices. We say that two vertices $i, j \in V_0$ are equivalent if $\{t \geq 0 : i \in V(G_t^{V_0})\} = \{t \geq 0 : j \in V(G_t^{V_0})\}$. Equivalence for two vertices $i, j \in \widehat{V}_0$ and for two vertices $i \in V_0$ and $j \in \widehat{V}_0$ is defined similarly. We observe that $(G_t^{V_0})_{t\geq 0}$ and $(\widehat{G}_t^{\widehat{V}_0})_{t\geq 0}$ are equal up to relabeling of the vertices if and only if each equivalence class has equal cardinality in V_0 and $\widehat{V}_0$. The latter event is measurable, since for any two vertices the event that these two vertices are equivalent is measurable by Lemma 54. Thus, we can assume that no vertices are permanently isolated.

To complete the proof, we must determine whether there exists a bijection $\phi_0\colon \mathbb{N} \to \mathbb{N}$ satisfying the properties of the map ϕ in Definition 24, i.e., whether there is a bijective map $\phi_0\colon \mathbb{N} \to \mathbb{N}$ such that for all times $t \geq 0$, $\phi_0(G_t) = \widehat{G}_t$. We will construct such a map by first constructing a sequence of maps ϕ_n defined on a growing sequence of domains, and then using a subsequence construction to transform them into a map $\phi_0\colon \mathbb{N} \to \mathbb{N}$ with the desired properties. We will show that this construction succeeds if and only if the two graph processes are equal up to relabeling.

To construct the maps ϕ_n, we define A_n^0 to be the set of injective maps $\phi\colon D \to \mathbb{N}$ such that D is finite, $\{1, \ldots, \lceil n/2 \rceil\} \subseteq D$, and $\{1, \ldots, \lfloor n/2 \rfloor\} \subseteq \phi(D)$. Let A_n to be the set of maps $\phi \in A_n^0$ such that $\phi(G_t^D) = \widehat{G}_t^{\phi(D)}$ for all $t \geq 0$. Note that A_n is non-empty for all n if the two graph processes are equal up to relabeling (just choose ϕ to be a restriction of the bijection ϕ_0). Note further that the set A_n^0 is countable, and that for each $\phi \in A_n^0$ the event that $\phi \in A_n$ is measurable by Lemma 54.

After these preparations, we are ready to construct the map ϕ_0. First we define $\phi_0(1)$. For each $j \in \mathbb{N}$ define the event B_j by $B_j = \bigcap_{n\in\mathbb{N}} B_{j,n}$, where $B_{j,n}$ is the event that there exists a map $\phi \in A_n$ for which $\phi(1) = j$. If the graph processes are equal up to relabeling, then B_j must occur for some j. We will prove that conversely, if B_j occurs for some j, then the graph processes are equal up to relabeling. Since B_j is a countable intersection of measurable events, this will finish our proof that the event that the two graph processes are equal up to relabeling is measurable.

To prove that the event B_j implies the existence of a bijection ϕ_0 such that $\phi_0(G_t) = \widehat{G}_t$ for all $t \geq 0$, we first note that the occurrence of B_j implies the existence of a sequence of maps $\phi_n \in A_n$ such that $\phi_n(1) = j$. Accordingly, we set $\phi_0(1) = j$. In the second step of the construction (explained below), we will determine $\phi_0^{-1}(1)$ by passing to a subsequence for which $\phi_n^{-1}(1)$ is constant. More generally, in the kth step of the construction, we will pass to a subsequence to ensure that $\phi_n(i)$ is constant for $1 \leq i \leq \lceil k/2 \rceil$ and $\phi_n^{-1}(i)$ is constant for $1 \leq i \leq \lfloor k/2 \rfloor$.

We will carry out this construction by induction on k. Suppose that we have defined $\phi_0(1), \ldots, \phi_0(\lceil k/2 \rceil)$ and $\phi_0^{-1}(1), \ldots, \phi_0^{-1}(\lfloor k/2 \rfloor)$ so that there exists a sequence $(\phi_n^k)_{n\geq k}$ of maps $\phi_n^k \in A_n$ for which

$$\begin{aligned} \phi_n^k(i) &= \phi_0(i) && \text{for all } n \geq k \text{ and all } i \leq \lceil k/2 \rceil, \text{ and} \\ (\phi_n^k)^{-1}(i) &= \phi_0^{-1}(i) && \text{for all } n \geq k \text{ and all } i \leq \lfloor k/2 \rfloor. \end{aligned}$$

Assume first that k is odd, in which case we need to define $\phi_0^{-1}(\lfloor (k+1)/2 \rfloor)$. Choose t in such a way that $\lfloor (k+1)/2 \rfloor$ is not isolated in $\widehat{G}_t$. Then $(\phi_n^k)^{-1}(\lfloor (k+1)/2 \rfloor)$ cannot be isolated in G_t either, and since G_t contains only a finite number of edges, we know there exists a finite set V such that for all $n \geq k$, $(\phi_n^k)^{-1}(\lfloor (k+1)/2 \rfloor) \in V$. But this implies that we can find a subsequence of $(\phi_n^k)_{n\geq k}$ on which $(\phi_n^k)^{-1}(\lfloor (k+1)/2 \rfloor)$ takes a fixed value, which we use to define $\phi_0^{-1}(\lfloor (k+1)/2 \rfloor)$. To conclude we need to prove the existence of a sequence $\phi_n^{k+1} \in A_n$ satisfying the induction hypothesis. For n in the subsequence obtained above we define $\phi_n^{k+1} = \phi_n^k$. To turn this subsequence into a sequence $\phi_n^{k+1} \in A_n$ defined for all $n \geq k+1$, we can simply reuse elements to fill in any gaps that occur before them, because $A_n \subset A_m$ for $m < n$. This completes the proof when k is odd, and the even case differs only in notation. ■

Appendix D. Random Graph Models

The main goal of this appendix is to establish Theorems 27 and 28. Proposition 56 will be used to prove left convergence of graphon processes in Section 2.5. It will also be applied in the proof of Theorem 28(i) in this appendix, where we need to consider the normalized number of edges in a graphon process.

A result of Lovász and Szegedy (2006, Corollary 2.6) in the setting of graphons over probability spaces is closely related to the following proposition. However, the proofs are different, even if both rely on martingale techniques. Note that in the course of proving the below proposition we give an alternative proof of a result of Veitch and Roy (2015, Theorem 5.3) for the special case of graphs with no self-edges. Recall from Section 2.5 that

for a simple graph F and a simple graph G we let $\mathrm{inj}(F, G)$ denote the number of injective adjacency preserving maps $\phi\colon V(F) \to V(G)$.

Proposition 56 *Let $\mathcal{W} = (W, \mathscr{S})$, where $W\colon S \times S \to [0,1]$ is a symmetric, measurable (but not necessarily integrable) function, and $\mathscr{S} = (S, \mathcal{S}, \mu)$ is a σ-finite measure space. Let F be a simple graph with vertex set $V(F) = \{1, \ldots, k\}$ for $k \geq 2$, such that F has no isolated vertices. Then a.s.*

$$\lim_{t\to\infty} t^{-k}\mathrm{inj}(F, G_t(\mathcal{W})) = \int_{S^k} \prod_{(i,j)\in E(F)} W(x_i, x_j)\, dx_1 \cdots dx_k, \tag{14}$$

where both sides should be read as elements of the extended non-negative reals $[0, \infty]$.

Remark 57 *Note that the proposition makes no integrability assumptions on W. All that is used is that W is measurable. As a consequence, the proposition can deal with situations where, say, the triangle density converges, even though $G_t(\mathcal{W})$ has infinitely many edges. An extreme example of such a behavior can be obtained by taking $\mathcal{W} = (W, \mathbb{R}_+)$ to be the "bipartite" graphon defined by $W(x,y) = \sum_{i,j\in\mathbb{N}} \mathbf{1}_{2i-2<x<2i-1}\mathbf{1}_{2j-1<y<2j} + \sum_{i,j\in\mathbb{N}} \mathbf{1}_{2i-1<x<2i}\mathbf{1}_{2j-2<y<2j-1}$, leading to a sequence of graphs $G_t(\mathcal{W})$ where every vertex has a.s. infinite degree, while all subgraph frequencies for graphs F that are not bipartite converge to zero.*

Proof of Proposition 56 We first prove the proposition under the assumption that the right side of (14) is finite. Throughout the proof we let $G_t := \widetilde{G}_t(\mathcal{W})$ be the graphon process generated by $\mathcal{W}$ with isolated vertices. Note that the left side of (14) is invariant under replacing $G_t(\mathcal{W})$ with $\widetilde{G}_t(\mathcal{W})$, because F has no isolated vertices. For each $t > 0$ define Y_{-t} to be the left side of (14) (with $G_t(\mathcal{W})$ replaced by G_t), i.e.,

$$Y_{-t} := t^{-k}\,\mathrm{inj}(F, G_t) = t^{-k} \sum_{v_1,\ldots,v_k\in V(G_t)} \prod_{(i,j)\in E(F)} \mathbf{1}_{(v_i,v_j)\in E(G_t)}.$$

As a first step, we will prove that for each $t > 0$,

$$\mathbb{E}[Y_{-t}] = \int_{S^k} \prod_{(i,j)\in E(F)} W(x_i, x_j)\, dx_1 \cdots dx_k. \tag{15}$$

We may assume that $\mu(S) < \infty$, since we can write S as a union of increasing sets S_m of finite measure for each $m \in \mathbb{N}$, and by the monotone convergence theorem it is sufficient to establish (15) with W replaced by $W\mathbf{1}_{S_m\times S_m}$, and with Y_{-t} defined in terms of graphs where we only consider vertices $v = (t, x)$ for which $x \in S_m$. If $N := |V(G_t)| < k$, then $Y_{-t} = 0$. If $N \geq k$, then

$$\mathbb{E}[Y_{-t} \,|\, N] = \frac{1}{t^k}\frac{N!}{(N-k)!}\frac{1}{\mu(S)^k}\int_{S^k} \prod_{(i,j)\in E(F)} W(x_i, x_j)\, dx_1 \cdots dx_k,$$

since we can form $\frac{N!}{(N-k)!}$ ordered sets of size k from $V(G_t)$, and the probability that a uniformly chosen injective map from $V(F)$ to $V(G_t)$ is a homomorphism, is given by

$$\frac{1}{\mu(S)^k}\int_{S^k} \prod_{(i,j)\in E(F)} W(x_i, x_j)\, dx_1 \cdots dx_k.$$

Since N has the law of a Poisson random variable with parameter $t\mu(S)$ we can conclude that (15) holds:

$$\begin{aligned}\mathbb{E}[Y_{-t}] &= \sum_{n=0}^{\infty} \mathbb{P}(N=n)\mathbb{E}[Y_{-t} \,|\, N=n] \\ &= \sum_{n=k}^{\infty} \frac{(t\mu(S))^n}{n!} e^{-t\mu(S)} \frac{1}{t^k} \frac{n!}{(n-k)!} \frac{1}{\mu(S)^k} \int_{S^k} \prod_{(i,j)\in E(F)} W(x_i,x_j)\, dx_1 \cdots dx_k \\ &= \int_{S^k} \prod_{(i,j)\in E(F)} W(x_i,x_j)\, dx_1 \cdots dx_k,\end{aligned}$$

implying in particular that Y_t is integrable for all $t < 0$ given that for now we assumed that the right side of (14) is finite. Note that in particular this implies that Y_t is a.s. finite, even though W may be such that the event that G_t has infinitely many edges has non-zero probability.

Let $\widehat{G}_t$ be identical to G_t, except that a vertex $v = (t,x) \in V(G_t)$ is labeled only with x. In order words, conditioning on a realization of $\widehat{G}_t$ is equivalent to conditioning on a realization of G_t, except that the time the different vertices were born (i.e., the time they appeared in the graphon process $(\widetilde{G}_t)_{t\geq 0}$) is unknown. Note that since S may have point masses multiple vertices of $\widehat{G}_t$ may have the same label, but they are still considered to be different. For $t \leq -1$, define $\mathcal{S}_t$ to be the σ-algebra generated by $(\widehat{G}_s)_{s\geq -t}$. Then $\mathcal{S}_s \subseteq \mathcal{S}_t$ for $s \leq t \leq -1$, so $(\mathcal{S}_t)_{t\leq -1}$ is a filtration, and Y_t is measurable with respect to $\widehat{G}_{-t}$ and hence $\mathcal{S}_t$; in other words, $(Y_t)_{t\leq -1}$ is adapted to the filtration.

Let $t > s > 0$. Given any k distinct vertices in $\widehat{G}_t$, the probability (conditioned on $(\widehat{G}_{t'})_{t'\geq t}$) that all k vertices are also in $\widehat{G}_s$ is given by $(s/t)^k$. Since $\widehat{G}_s$ is an induced subgraph of $\widehat{G}_t$, it follows that

$$\begin{aligned}\mathbb{E}[Y_{-s} \,|\, \mathcal{S}_{-t}] &= \mathbb{E}[Y_{-s} \,|\, (\widehat{G}_{t'})_{t'\geq t}] \\ &= \frac{1}{s^k} \sum_{v_1,\ldots,v_k \in V(\widehat{G}_t)} \prod_{(i,j)\in E(F)} \mathbf{1}_{(v_i,v_j)\in E(G_t)} \cdot \\ &\qquad\qquad \mathcal{P}\big(v_1,\ldots,v_k \in V(\widehat{G}_s) \,|\, v_1,\ldots,v_k \in V(\widehat{G}_t)\big) \\ &= Y_{-t},\end{aligned}$$

proving that $(Y_t)_{t<0}$ is a backwards martingale. The limit $Y_{-\infty} = \lim_{t\to\infty, t\in\mathbb{Q}} Y_{-t}$ exists almost surely (Kallenberg, 2002, Theorem 7.18). Since $\mathbb{E}[Y_{-t}] < \infty$ for all $t > 0$ we know that a.s., $Y_{-t} < \infty$ for all $t > 0$, which implies that a.s., $|E(G_t)| < \infty$ for all $t > 0$. Therefore $(Y_t)_{t<0}$ has finitely many discontinuities in any bounded interval, and is left-continuous with limits from the right. It follows that $Y_{-\infty} = \lim_{t\to\infty} Y_{-t}$; i.e., we do not need to take the limit along rationals.

To complete the proof it is sufficient to prove that the limit $Y_{-\infty}$ is equal to the right side of (15) almost surely. To establish this it is sufficient to prove that $Y_{-\infty}$ is equal to a deterministic constant almost surely, since $(Y_t)_{t<0}$ is uniformly integrable (Kallenberg, 2002, Theorem 7.21), which implies that $(Y_t)_{t<0}$ converges to $Y_{-\infty}$ also in L^1.

We will use the Kolmogorov 0-1 law (Stroock, 2011a, Theorem 1.1.2) to deduce this. For any $n \in \mathbb{N}$ define $\mathcal{V}_n := \{(s,x) \in \mathcal{V} : n-1 \leq s < n\}$. Let $\mathcal{F}_n$ be the σ-algebra generated by the set $\mathcal{V}_n$ and the edges between the vertex set $\mathcal{V}_n$ and the vertex set $\bigcup_{1\leq m\leq n} \mathcal{V}_m$. Since the randomness of the edges can be represented as an infinite sequence of independent uniform random variables, the σ-algebras $\mathcal{F}_n$ can be considered independent even if the edges considered in $\mathcal{F}_n$ join vertices in $\mathcal{V}_n$ and $\mathcal{V}_m$ for $m < n$. In order to apply the 0-1 law it is sufficient to prove that $Y_{-\infty}$ is measurable with respect to the σ-algebra generated by $\bigcup_{n\geq n_0} \mathcal{F}_n$ for all $n_0 \in \mathbb{N}$.

Define $Y_{-t,\geq n_0}$ in the same way as Y_{-t}, except that instead of summing over vertices in $V(G_t)$, we sum over vertices in $V(G_t) \cap \mathcal{V}_{\geq n_0}$, where $\mathcal{V}_{\geq n_0} = \bigcup_{n\geq n_0} \mathcal{V}_n$. Since $Y_{-t,\geq n_0}$ is measurable with respect to the σ-algebra generated by $\bigcup_{n\geq n_0} \mathcal{F}_n$, all we need to show is that for all $n \geq n_0$, a.s., $Y_{-t} - Y_{-t,\geq n_0} \to 0$ as $t \to \infty$. The difference between Y_{-t} and $Y_{-t,\geq n_0}$ can then be bounded by

$$t^{-k} \sum_{\substack{v_1,\ldots,v_k \in V(G_t) \\ t_i \leq n_0 \text{ for some } i\in[k]}} \prod_{(i,j)\in E(F)} \mathbf{1}_{(v_i,v_j)\in E(G_t)},$$

where t_i is the time label of v_i. Conditioned on $v_1,\ldots,v_k \in V(G_t)$, the probability that at least one of them has time label $t_i \leq t_0$ is bounded by kt_0/t. Continuing as in the proof of (15), we therefore obtain that

$$0 \leq \mathbb{E}[Y_{-t} - Y_{-t,\geq n_0}] \leq k\Big(\frac{t_0}{t}\Big) \int_{S^k} \prod_{(i,j)\in E(F)} W(x_i,x_j)\, dx_1 \cdots dx_k. \tag{16}$$

Since the limit $Y_{-\infty} = \lim_{t\to\infty} Y_{-t}$ exists, we can calculate $Y_{-\infty}$ along any sequence, say the sequence $(Y_{-n^2})_{n\in\mathbb{N}}$. The bound (16) combined with Markov's inequality and the Borel-Cantelli lemma therefore implies that $Y_{-\infty} = \lim_{n\to\infty} Y_{-n^2,\geq n_0}$, proving that $Y_{-\infty}$ is measurable with respect to the σ-algebra generated by $\bigcup_{n\geq n_0} \mathcal{F}_n$, as required for the application of the 0-1 law.

This completes the proof of the proposition under the assumption that the right side of (14) is finite. If the right side is infinite, we note that for any set A of finite measure (14) holds with $W\mathbf{1}_{A\times A}$ instead of W on both the left side and the right side. We can make the right side arbitrarily large by increasing A. The left side is monotone in A, and therefore the limit inferior of the left side (with W, not $W\mathbf{1}_{A\times A}$) is larger than any fixed constant, and hence is equal ∞. ■

Proof of Theorem 28 (i) Since the result of the theorem is immediate for $\|W\|_1 = 0$, we will assume throughout the proof that $\|W\|_1 > 0$. Since $\delta_\square(\widetilde{G}_t, G_t) = 0$ by (3), it is enough to prove the statement for either $(\widetilde{G}_t)_{t\geq 0}$ or $(G_t)_{t\geq 0}$.

If S has finite total mass, then, a.s., $|V(\widetilde{G}_t)|$ is finite for each fixed $t \geq 0$, and conditioned on the size of $|V(\widetilde{G}_t)|$, the graph $\widetilde{G}_t$ is a $\mathcal{W}$-random graph in the sense of the theory of dense graph convergence. The results of Borgs, Chayes, Lovász, Sós, and Vesztergombi (2008) imply that $\delta_\square(\widetilde{G}_t, \widetilde{\mathcal{W}}) \to 0$ and $\|W^{\widetilde{G}_t}\|_1 \to \|\widetilde{W}\|_1$ where $\widetilde{\mathcal{W}} = (\widetilde{W}, \widetilde{\mathscr{S}})$ is obtained from $\mathcal{W}$ by normalizing the measure to a probability measure (giving, in particular, $\|\widetilde{W}\|_1 = \|W\|_1/\pi(S)^2$.) Combined with Lemma 44, this implies that $\delta^{\mathrm{s}}_\square(\widetilde{G}_t, \mathcal{W}) \to 0$ when $\pi(S) < \infty$.

If $\pi(S) = \infty$, we use Lemma 47, and the observation that two graphons generate graphon processes with the same law if one graphon is a pullback of the other, to reduce the proof to the case $\mathscr{S} = (\mathbb{R}_+, \mathcal{B}, \lambda)$. Given $0 < \varepsilon < 1/2$ choose $M > 0$ such that $\|W - W\mathbf{1}_{[0,M]^2}\|_1 < \varepsilon\|W\|_1$, and define $\mathcal{W}_M$ to be the graphon $W_M = W\mathbf{1}_{[0,M]^2}$ over $[0, M]$, and $\widetilde{G}_t^M$ to be the induced subgraph of $\widetilde{G}_t$ on the set of vertices (s, x) such that $x \le M$. Define $\widetilde{W}^{\widetilde{G}_t^M,s} := W^{\widetilde{G}_t^M}(\lambda_M \cdot, \lambda_M \cdot)$ with $\lambda_M := M^{-1}\|W\|_1^{1/2}$. In the cut metric $\delta_\square$, the stretched graphon $\widetilde{W}^{\widetilde{G}_t^M,s}$ then converges to $\widetilde{W}_M^s := W_M(\|W\|_1^{1/2} \cdot, \|W\|_1^{1/2} \cdot)$, again by the convergence of $\mathcal{W}$-random graphs for $\mathcal{W}$ defined on a probability space.

Furthermore, by Proposition 56 applied to the graph F consisting of a single edge, we have that a.s., the number of edges in $\widetilde{G}_t^M$ divided by t^2 converges to $\frac{1}{2}\|W\mathbf{1}_{[0,M]^2}\|_1$, so in particular the time t_M where $\widetilde{G}_t^M$ has at least one edge is a.s. finite. For the rest of this proof, we will always assume that $t \ge t_M$.

Defining G_t' to be the graph obtained from $\widetilde{G}_t$ by removing all isolated vertices (s, x) from $V(\widetilde{G}_t)$ for which $x > M$, we note that by (3), it is sufficient to prove that $\delta_\square^s(G_t', \mathcal{W}) \to 0$. Recall that each vertex $v = (s, x)$ of G_t' corresponds to an interval when we define the stretched canonical graphon $\mathcal{W}^{G_t',s}$ of G_t'. Assume the intervals are ordered according the value of x; i.e., if the vertices $v = (s, x)$ and $v' = (s', x')$ satisfy $x < x'$, then the interval corresponding to v is to the left on the real line of the interval corresponding to v'. Noting that by our assumption $t \ge t_M$, there exists at last one vertex $v = (s, x)$ in G_t' such that $x \le M$, we define the graphon $\widetilde{\mathcal{W}}^{G_t',s} = (\widetilde{W}^{G_t',s}, \mathbb{R}_+)$ to be a "stretched" version of $\mathcal{W}^{G_t'}$ such that the vertices $v = (s, x)$ for which $x \in [0, M]$ correspond to the interval $[0, \lambda_M^{-1}]$. In other words, $\widetilde{W}^{G_t',s} = W^{G_t',s}(r_t \cdot, r_t \cdot)$ for some appropriately chosen constant $r_t > 0$. To calculate r_t, we note that $W^{G_t',s} = W^{G_t'}(\lambda \cdot, \lambda \cdot)$ with $\lambda = \|W^{G_t'}\|_1^{1/2} = |V(G_t')|^{-1}\sqrt{2|E(G_t')|}$ and $\widetilde{W}^{G_t',s} = W^{G_t'}(\lambda' \cdot, \lambda' \cdot)$ with $\lambda' = \lambda_M|V(\widetilde{G}_t^M)||V(G_t')|^{-1}$, giving $r_t = \lambda'/\lambda = \sqrt{\lambda_M^2|V(\widetilde{G}_t^M)|^2/(2|E(G_t')|)}$. Since $|V(\widetilde{G}_t^M)|$ is an exponential random variable with expectation Mt, and $|E(G_t')|/t^2 = |E(G_t)|/t^2 \to \frac{1}{2}\|W\|_1$ a.s. by Proposition 56, we have that, a.s., $r_t \to 1$ as $t \to \infty$. By the triangle inequality, Lemma 36, and the fact that $\widetilde{W}^{G_t',s}|_{[0,\lambda_M^{-1}]^2} = \widetilde{W}^{\widetilde{G}_t^M,s}$,

$$\begin{aligned}\delta_\square^s(\mathcal{W}, G_t') \le \|W^s - \widetilde{W}_M^s\|_1 + \delta_\square(\widetilde{\mathcal{W}}_M^s, \widetilde{W}^{\widetilde{G}_t^M,s}) \\ + \|\widetilde{W}^{G_t',s}|_{[0,\lambda_M^{-1}]^2} - \widetilde{W}^{G_t',s}\|_1 + \delta_\square(\widetilde{\mathcal{W}}^{G_t',s}, \mathcal{W}^{G_t',s}).\end{aligned} \tag{17}$$

The first term on the right side of (17) is bounded by ε by assumption, and the second converges to zero as already discussed above. The third term on the right side of (17) is the product of r_t^{-2} and the fraction of edges of G_t for which at least one vertex $v = (t, x)$ satisfies $x > M$. By Proposition 56 (applied with the random graphs $\widetilde{G}_t^M$ and $\widetilde{G}_t$ and the same simple graph F as above) and $\lim_{t\to\infty} r_t = 1$ it follows that this term is less than 2ε for all sufficiently large $t > 0$. The fourth term on the right side of (17) converges to zero by $\lim_{t\to\infty} r_t = 1$ and Lemma 44. Since $\varepsilon > 0$ was arbitrary we can conclude that $\lim_{t\to\infty} \delta_\square^s(\mathcal{W}, G_t) = 0$. ∎

Proof of Theorem 28 (ii) First we will show that the condition $\sum_{n=1}^\infty \mu(S_n)^{-1} = \infty$ is necessary. We will use proof by contradiction, and assume $\sum_{n=1}^\infty \mu(S_n)^{-1} < \infty$ and

a.s.-$\lim_{n\to\infty} \delta_\square^s(\mathcal{W}, G_n) = 0$. We will obtain the contradiction by proving that with positive probability $E(G_n) = \emptyset$ for all $n \in \mathbb{N}$ (which clearly contradicts a.s.-$\lim_{n\to\infty} \delta_\square^s(\mathcal{W}, G_n) = 0$). By rescaling the measure of $\mathscr{S}$ we may assume without loss of generality that $\|W\|_1 = 1$. Furthermore, we assume that $\mu(S) = \infty$ by extending $\mathscr{S}$ and W to a space of infinite measure. Note that the condition $\bigcup_i S_i = S$ will not hold after such an extension has been done, but we will not use this property in the proof. (The property $\bigcup_i S_i = S$ is applied only in the second part of the proof, where we show that the condition $\sum_{n=1}^\infty \mu(S_n)^{-1} = \infty$ is sufficient.)

First we will prove that there is a random $N \in \mathbb{N}$ such that $\mathcal{W}^{G_n,s} = \mathcal{W}^{G_N,s}$ (up to interval permutations) for all $n \ge N$. Since $(|E(G_n)|)_{n\in\mathbb{N}}$ is increasing, in order to do this it is sufficient to prove that $(|E(G_n)|)_{n\in\mathbb{N}}$ is bounded almost surely, and by monotone convergence, this in turn follows once we show that $\sup_{n\in\mathbb{N}} \mathbb{E}[|E(G_n)|] < \infty$. Letting $v_i \in V(G_i)$ denote the vertex added in step $i \in \mathbb{N}$, and defining $S_0 = \emptyset$, we obtain the desired result:

$$\begin{aligned}\mathbb{E}[|E(G_n)|] &= \sum_{1\le i<j\le n} \mathbb{P}[(v_i, v_j) \in E(G_n)] \\ &= \sum_{1\le i<j\le n} \frac{1}{\mu(S_i)\mu(S_j)}\|W\mathbf{1}_{S_i\times S_j}\|_1 \\ &\le \sum_{i',j'=1}^n \|W\mathbf{1}_{(S_{i'}\setminus S_{i'-1})\times(S_{j'}\setminus S_{j'-1})}\|_1 \sum_{i\ge i', j\ge j'} \frac{1}{\mu(S_i)\mu(S_j)} \\ &\le \|W\|_1 \left(\sum_{n=1}^\infty \mu(S_n)^{-1}\right)^2 \\ &< \infty.\end{aligned}$$

Since $\lim_{n\to\infty} \delta_\square^s(G_n, \mathcal{W}) = 0$ it follows that $\delta_\square(\mathcal{W}^{G_N,s}, \mathcal{W}) = 0$.

We saw in the above paragraph that $\delta_\square(\mathcal{W}^{G_N,s}, \mathcal{W}) = 0$ a.s. for some random $N \in \mathbb{N}$. Therefore there is a deterministic step graphon $\widehat{\mathcal{W}} = (\widehat{W}, \mathbb{R}_+)$ with values in $\{0, 1\}$ such that $\delta_\square(\widehat{\mathcal{W}}, \mathcal{W}) = 0$. Since the set $\{D_{\widehat{W}} > 0\}$ has finite measure, by Lemma 45, the set $A = \{D_W > 0\}$ has finite measure as well. After changing W on a set of measure 0, we have $\operatorname{supp} W \subseteq A \times A$. Note also that by Proposition 7 we have $\delta_1(\widehat{W}, W) = 0$.

For any $n \in \mathbb{N}$ the probability that a feature sampled from the measure μ_n is contained in A is given by $\mu(A \cap S_n)/\mu(S_n) \le \mu(A)/\mu(S_n)$. Hence the Borel-Cantelli lemma implies that finitely many vertices in $\bigcup_{n\ge1} V(G_n)$ have a feature in A. Therefore we can find a deterministic $n_0 \in \mathbb{N}$ such that $\mathbb{P}(x_n \notin A \text{ for all } n \ge n_0) > 0$. It follows that with uniformly positive probability conditioned on G_{n_0}, no edges are added to G_n after time n_0.

To conclude our proof (i.e., obtain a contradiction by proving that $E(G_n) = \emptyset$ with positive probability) it is therefore sufficient to prove that $E(G_{n_0}) = \emptyset$ with positive probability. We will do this by sampling a sequence of graphs $(\widehat{G}_n)_{n\in\mathbb{N}}$ from $\widehat{\mathcal{W}}$ which is close in law to $(G_n)_{n\in\mathbb{N}}$, and use that $E(\widehat{G}_n) = \emptyset$ with positive probability since $\widehat{W}$ is zero on a certain subdomain since the graphs we consider have no loops. (Note that our approach would not have worked if we allowed for loops; if, for example, $\widehat{W}|_{[0,1]^2} \equiv 1$ and $S_1, S_2 \subset [0, 1]$ we would have had $\mathbb{P}(E(\widehat{G}_n) = \emptyset) = 0$.) Let $\varepsilon > 0$, and recalling that $\delta_1(\widehat{\mathcal{W}}, \mathcal{W}) = 0$, choose a coupling measure $\widetilde{\mu}$ on $S \times \mathbb{R}_+$ such that $\|W^{\pi_1} - \widehat{W}^{\pi_2}\|_{1,\widetilde{\mu}} < \varepsilon$. By using $\widetilde{\mu}$ we can sample

two coupled sequences of graphs $(G_n)_{1\le n\le n_0}$ and $(\widehat{G}_n)_{1\le n\le n_0}$, such that the two sequences have a law which is close in total variation distance, $(G_n)_{1\le n\le n_0}$ has the law of the graphs in the statement of the theorem, and $(\widehat{G}_n)_{1\le n\le n_0}$ is sampled similarly as $(G_n)_{n\in\mathbb{N}}$ but with $\widehat{\mathcal{W}}$ instead of $\mathcal{W}$. More precisely, for each $n\in\{1,\dots,n_0\}$ we sample $(x,\widehat{x})\in S_n\times\mathbb{R}_+$ from the probability measure $\mu(S_n)^{-1}\widetilde{\mu}|_{S_n\times\mathbb{R}_+}$, we let x (resp. $\widehat{x}$) be the feature of the nth vertex of G_n (resp. $\widehat{G}_n$), and by using that $\|W^{\pi_1}-\widehat{W}^{\pi_2}\|_{1,\widetilde{\mu}}<\varepsilon$ we can couple $(G_n)_{1\le n\le n_0}$ and $(\widehat{G}_n)_{1\le n\le n_0}$ such that for each $n_1,n_2\in\{1,\dots,n_0\}$ for which $n_1\neq n_2$ we have

$$\begin{aligned}\mathcal{P}\Big(&\big\{(x_{n_1},x_{n_2})\in E(G_{n_0}),(\widehat{x}_{n_1},\widehat{x}_{n_2})\notin E(\widehat{G}_{n_0})\big\}\cup\\&\qquad\big\{(x_{n_1},x_{n_2})\notin E(G_{n_0}),(\widehat{x}_{n_1},\widehat{x}_{n_2})\in E(\widehat{G}_{n_0})\big\}\Big)\\&\le\mu(S_{n_1})^{-1}\mu(S_{n_2})^{-1}\int_{(S_{n_1}\times\mathbb{R}_+)\times(S_{n_2}\times\mathbb{R}_+)}\left|W^{\pi_1}-\widehat{W}^{\pi_2}\right|\,d\widetilde{\mu}\,d\widetilde{\mu}\\&<\mu(S_{n_1})^{-1}\mu(S_{n_2})^{-1}\varepsilon.\end{aligned}$$

Hence the total variation distance between the laws of $(G_n)_{1\le n\le n_0}$ and $(\widehat{G}_n)_{1\le n\le n_0}$ is bounded by $n_0^2\mu(S_1)^{-2}\varepsilon$. Since we can make this distance arbitrarily small by decreasing ε, in order to complete our proof it is sufficient to prove that $E(\widehat{G}_{n_0})=\emptyset$ with a uniformly positive probability for all coupling measures $\widetilde{\mu}$. Write $\mathbb{R}_+=\bigcup_{n=0}^N A_n$, such that $A_0,\dots,A_N$ correspond to the steps of the step function $\widehat{W}$, with, say, A_0 corresponding to the set of all x such that $\int\widehat{W}(x,y)\,dy=0$. For any choice of $\widetilde{\mu}$ we can find a $k=k_{\widetilde{\mu}}\in\{0,\dots,N\}$ such that $\widetilde{\mu}(S_1\times A_k)\ge\mu(S_1)/(N+1)$. Therefore there is a uniformly positive probability that all the vertices of $\widehat{G}_{n_0}$ have a feature in A_k. On this event we have $E(\widehat{G}_{n_0})=\emptyset$, since $\widehat{W}|_{A_k\times A_k}\equiv 0$ as the graphs we consider are simple (i.e., they do not have loops). This completes our proof that the condition $\sum_{n=1}^\infty\mu(S_n)^{-1}=\infty$ is necessary.

Now we will prove that the condition $\sum_{n=1}^\infty\mu(S_n)^{-1}=\infty$ is sufficient to guarantee that a.s.-$\lim_{n\to\infty}\delta^s_\square(\mathcal{W},G_n)=0$. We will couple $(G_n)_{n\in\mathbb{N}}$ to a graphon process $(\widetilde{G}_t)_{t\ge0}$ with isolated vertices. Fix $\varepsilon>0$, and choose $N\in\mathbb{N}$ sufficiently large such that $\|W-W\mathbf{1}_{S_N\times S_N}\|_\square<\varepsilon$. Sample $(\widetilde{G}_t)_{t\ge0}$, and independently from $(\widetilde{G}_t)_{t\ge0}$, sample $(G_n)_{1\le n\le N}$ as described in the statement of the theorem. Define $(t_n)_{n\ge N}$ inductively as follows

$$t_N=0,\qquad t_n=\inf\{t>t_{n-1}\,:\,\text{there exists } x\in S_n\text{ such that }(t,x)\in V(\widetilde{G}_t)\}.$$

Note that $t_n\to\infty$ a.s. as $n\to\infty$, because the increments t_n-t_{n-1} are independent and exponentially distributed with mean $\mu(S_n)^{-1}$, and $\sum_{n=N+1}^\infty\mu(S_n)^{-1}=\infty$ by assumption. For $n>N$ let $\widehat{G}_n$ be the induced subgraph of $\widetilde{G}_{t_n}$ whose vertex set is

$$V(\widehat{G}_n)=\{(t,x)\in V(\widetilde{G}_{t_n})\,:\,\text{there exists }\widetilde{n}\in\{N+1,\dots,n\}\text{ such that }t=t_{\widetilde{n}}\}.$$

For each $n>N$ let G_n be the union of G_N and $\widehat{G}_n$, such that the edge set of G_n is given by $E(G_N)\cup E(\widehat{G}_n)$ in addition to independently sampled edges between the vertices of G_N and the vertices of $\widehat{G}_n$, such that the probability of connecting vertices with features x and x' is $W(x,x')$, and such that G_{n-1} is an induced subgraph of G_n. It is immediate that $(G_n)_{n\in\mathbb{N}}$ has the same law as the sequence of graphs $(G_n)_{n\in\mathbb{N}}$ described in the statement of the theorem.

We will prove that $|E(G_n)\backslash E(\widetilde{G}_{t_n})|=o(|E(\widetilde{G}_{t_n})|)$ and $|E(\widetilde{G}_{t_n})\backslash E(G_n)|<\varepsilon|E(\widetilde{G}_{t_n})|$ for all large $n\in\mathbb{N}$. This is sufficient to complete the proof of the theorem, since $\varepsilon>0$ was arbitrary, since $\lim_{n\to\infty}\delta^s_\square(\mathcal{W},\widetilde{G}_{t_n})=0$ by part (i) of the theorem, and since $\delta^s_\square(G_n,\widetilde{G}_{t_n})\to0$ as $n\to\infty$ and $\varepsilon\to0$ by the following argument. Define $\widetilde{\mathcal{W}}^{G_n,s}:=(\widetilde{W}^{G_n,s},\mathbb{R}_+)$ and $\widetilde{W}^{G_n,s}:=W^{G_n,s}(r_n^{-1}\cdot,r_n^{-1}\cdot)$ for $r_n=|E(G_n)|^{1/2}|E(\widetilde{G}_{t_n})|^{-1/2}$; i.e., $\widetilde{W}^{G_n,s}$ is a stretched version of $W^{G_n,s}$ defined such that each vertex of G_n corresponds to an interval of length $(2|E(\widetilde{G}_{t_n})|)^{-1/2}$. Then each vertex corresponds to an interval of length $(2|E(\widetilde{G}_{t_n})|)^{-1/2}$ both for $\widetilde{W}^{G_n,s}$ and for $W^{\widetilde{G}_{t_n},s}$, so by ordering the vertices appropriately when defining the graphons we have $\|\widetilde{W}^{G_n,s}-W^{\widetilde{G}_{t_n},s}\|_1\le|E(G_n)\triangle E(\widetilde{G}_{t_n})||E(\widetilde{G}_{t_n})|^{-1}=o_n(1)+\varepsilon$. For sufficiently small $\varepsilon>0$ and large $n\in\mathbb{N}$ we have $|r_n-1|<\big||E(G_n)|^{1/2}-|E(\widetilde{G}_{t_n})|^{1/2}\big||E(\widetilde{G}_{t_n})|^{-1/2}<o_n(1)+\varepsilon$, and hence Lemma 44 implies that $\delta_\square(\mathcal{W}^{G_n,s},\widetilde{\mathcal{W}}^{G_n,s})<4\varepsilon$ for all sufficiently small $\varepsilon>0$ and sufficiently large $n\in\mathbb{N}$. Combining the above estimates we get that for all sufficiently small $\varepsilon>0$ and sufficiently large $n\in\mathbb{N}$,

$$\delta^s_\square(G_n,\widetilde{G}_{t_n})\le\delta_\square(\mathcal{W}^{G_n,s},\widetilde{\mathcal{W}}^{G_n,s})+\delta_\square(\widetilde{\mathcal{W}}^{G_n,s},\mathcal{W}^{\widetilde{G}_{t_n},s})\le4\varepsilon+\|\widetilde{W}^{G_n,s}-W^{\widetilde{G}_{t_n},s}\|_1\le6\varepsilon.$$

First we prove that conditioned on almost any realization of G_N, $|E(G_n)\backslash E(\widetilde{G}_{t_n})|=o(|E(\widetilde{G}_{t_n})|)$ as $n\to\infty$. Note that $E(G_n)\backslash E(\widetilde{G}_{t_n})$ consists of the edges in $E(G_N)$, plus independently sampled edges between $V(G_N)$ and $V(\widehat{G}_n)$. Since $V(\widehat{G}_n)\subset V(\widetilde{G}_{t_n})$, we overcount the latter if we independently sample one edge for each $v\in V(G_N)$ and $v'\in V(\widetilde{G}_{t_n})$, with the probability of an edge between v and v' given by W evaluated at the features of v and v'. Defining $\deg(v;\widetilde{G}_{t_n})$ to be the number of edges between $v\in V(G_N)$ and $V(\widetilde{G}_{t_n})$ obtained in this way, we thus have

$$|E(G_n)\backslash E(\widetilde{G}_{t_n})|\le|E(G_N)|+\sum_{v\in V(G_N)}\deg(v;\widetilde{G}_{t_n}).$$

By Proposition 56 applied with F being the simple connected graph with two vertices, $|E(\widetilde{G}_{t_n})|=\Theta(t_n^2)$. In order to prove that $|E(G_n)\backslash E(\widetilde{G}_{t_n})|=o(|E(\widetilde{G}_{t_n})|)$ it is therefore sufficient to prove that, conditioned on almost any realization of G_N, each vertex $v\in V(G_N)$ satisfies $\deg(v;\widetilde{G}_{t_n})\le Ct_n$ for all sufficiently large n and some $C>0$ depending on the feature of the vertex. Condition on a realization of G_N such that $\int_S W(x,y)\,d\mu(y)<\infty$ for all $x\in S$ such that x is the feature of some vertex in G_N. We will prove that if $x\in S$ is the feature of $v\in V(G_N)$ then a.s.

$$\lim_{t\to\infty}Y_{-t}=\int_S W(x,y)\,d\mu(y),\qquad\text{where }Y_{-t}:=t^{-1}\deg(v;\widetilde{G}_t)\text{ for all }t>0,\tag{18}$$

which is sufficient to imply the existence of an appropriate constant C. The convergence result (18) follows by noting that $(Y_t)_{t<0}$ is a backwards martingale with expectation $\int_S W(x,y)\,d\mu(y)$, which is left-continuous with right limits at each $t<0$; see the proof of Proposition 56 for a very similar argument. Hence the Kolmogorov 0-1 law implies (18). We can conclude that $|E(G_n)\backslash E(\widetilde{G}_{t_n})|=o(|E(\widetilde{G}_{t_n})|)$.

Now we prove $|E(\widetilde{G}_{t_n})\backslash E(G_n)|<\varepsilon|E(\widetilde{G}_{t_n})|$. Let $\overline{G}_{t_n}$ be the induced subgraph of $\widetilde{G}_{t_n}$ corresponding to vertices with feature in S_N. Then

$$|E(\widetilde{G}_{t_n})\backslash E(G_n)|\le|E(\widetilde{G}_{t_n})|-|E(\overline{G}_{t_n})|.$$

By applying Proposition 56 to each of the graphs $\overline{G}_{t_n}$ and $\widehat{G}_{t_n}$, and with F being the simple connected graph on two vertices, it follows that

$$\limsup_{n\to\infty} |E(\widetilde{G}_{t_n})|^{-1}|E(\widetilde{G}_{t_n})\backslash E(G_n)| \leq \lim_{n\to\infty} |E(\widetilde{G}_{t_n})|^{-1}(|E(\widetilde{G}_{t_n})| - |E(\overline{G}_{t_n})|)$$
$$= \|W\|_1 - \|W\mathbf{1}_{S_N\times S_N}\|_1 = \|W - W\mathbf{1}_{S_N\times S_N}\|_1 < \varepsilon.$$

■

Proof of Theorem 27 Assume that (i) holds, i.e., $\delta_\square(\mathcal{W}_1,\mathcal{W}_2) = 0$. We will prove that (ii) and (iii) also hold. It is sufficient to prove that (ii) holds, since (ii) implies (iii).

We first consider the case when $\mu_i(I_i) < \infty$ for $i = 1, 2$, where $I_i := \{x \in S_i : D_{W_i}(x) > 0\}$. Recall that by Proposition 21 we have $\mu_1(I_1) = \mu_2(I_2)$, so by restricting the graphon $\mathcal{W}_i$ to the space I_i for $i = 1, 2$ we obtain two graphons with cut distance zero over spaces of finite and equal measure. By definition of I_i, almost surely no vertices of $(\widetilde{G}_t)_{t\geq 0}$ will be isolated for all times, and it is proved that (ii) holds in, for example, a paper by Janson (2013, Theorem 8.10), who refers to papers by Borgs, Chayes, Lovász, Sós, and Vesztergombi (2008), Borgs, Chayes, and Lovász (2010), and Diaconis and Janson (2008) for the original proofs.

Next we consider the case where $\mu_1(I_1) = \mu_2(I_2) = \infty$. We may assume $\mu_i(S_i \setminus I_i) = 0$, since replacing the graphon $\mathcal{W}_i$ by its restriction to $S_i \setminus I_i$ amounts to removing vertices which are isolated for all times. Part (i) of Proposition 51 now implies that we can find a measure μ such that $W_1^{\pi_1} = W_2^{\pi_2}$ μ-almost everywhere. By the assumption $\mu_i(S_i \setminus I_i) = 0$, part (iv) of the proposition implies that μ is a coupling measure. Sampling a graphon process from $\mathcal{W}_i$ may be done by associating the vertex set with a Poisson point process on $(S_1 \times S_2) \times \mathbb{R}_+$ with intensity $\mu \times \lambda$, such that each $((x_1, x_2), t) \in (S_1 \times S_2) \times \mathbb{R}_+$ is associated with a vertex with feature x_i appearing at time t.

Now we will prove that (ii) or (iii) imply (i). We will only show that (ii) implies (i), since we can prove that (iii) implies (i) by the exact same argument. We assume (ii) holds, and couple $(\widetilde{G}_t^1)_{t\geq 0}$ and $(\widetilde{G}_t^2)_{t\geq 0}$ such that $\widetilde{G}_t^1 = \widetilde{G}_t^2$ for all $t \geq 0$. By Theorem 28(i) we know that $\lim_{t\to\infty} \delta_\square(\mathcal{W}_i, \mathcal{W}^{\widetilde{G}_t^i}) = 0$. Since $\mathcal{W}^{\widetilde{G}_t^1} = \mathcal{W}^{\widetilde{G}_t^2}$ for all $t \geq 0$ it follows by the triangle inequality that $\delta_\square(\mathcal{W}_1, \mathcal{W}_2) = 0$, so (i) holds. ■

Appendix E. Compactness

In this appendix we will establish Theorem 15.

Lemma 58 *Let $(\mathcal{W}_n)_{n\in\mathbb{N}}$ and $(\widetilde{\mathcal{W}}_n)_{n\in\mathbb{N}}$ be two sequences of graphons, with $\mathcal{W}_n = (W_n, \mathscr{S}_n)$, $\mathscr{S}_n = (S_n, \mathcal{S}_n, \mu_n)$, $\widetilde{\mathcal{W}}_n = (\widetilde{W}_n, \widetilde{\mathscr{S}}_n)$, and $\widetilde{\mathscr{S}}_n = (\widetilde{S}_n, \widetilde{\mathcal{S}}_n, \widetilde{\mu}_n)$, such that there are measure-preserving transformations $\phi_n \colon S_n \to \widetilde{S}_n$ for which $\lim_{n\to\infty} \|W_n - \widetilde{W}_n^{\phi_n}\|_1 = 0$. Furthermore, assume that either (i) ϕ_n is a bimeasurable bijection, or (ii) $S_n = \widetilde{S}_n \times [0, 1]$, where $[0, 1]$ is equipped with Lebesgue measure, and $\phi_n \colon S_n \to \widetilde{S}_n$ is the projection map. Then $(\mathcal{W}_n)_{n\in\mathbb{N}}$ has uniformly regular tails iff $(\widetilde{\mathcal{W}}_n)_{n\in\mathbb{N}}$ has uniformly regular tails.*

Proof Let $(\varepsilon_n)_{n\in\mathbb{N}}$ be a sequence of positive real numbers converging to zero, such that $\|W_n - \widetilde{W}_n^{\phi_n}\|_1 < \varepsilon_n$ for all $n \in \mathbb{N}$. First assume $(\widetilde{\mathcal{W}}_n)_{n\in\mathbb{N}}$ has uniformly regular tails. Given any $\varepsilon > 0$ let $M \geq 0$ be such that for all $n \in \mathbb{N}$ we can find $\widetilde{U}_n \in \widetilde{\mathcal{S}}_n$ satisfying $\widetilde{\mu}_n(\widetilde{U}_n) < M$ and $\|\widetilde{W}_n - \widetilde{W}_n\mathbf{1}_{\widetilde{U}_n\times\widetilde{U}_n}\|_1 < \varepsilon/2$. Define $U_n := \phi_n^{-1}(\widetilde{U}_n)$. Since ϕ_n is measure-preserving, $\mu_n(U_n) = \mu_n(\widetilde{U}_n) < M$. By first using $\|W_n - \widetilde{W}_n^{\phi_n}\|_1 < \varepsilon_n$ (which implies that $\|(W_n - \widetilde{W}_n^{\phi_n})\mathbf{1}_{U_n\times U_n}\|_1 < \varepsilon_n$) and then using that ϕ_n is measure-preserving we get

$$\begin{aligned}
&\|W_n - W_n\mathbf{1}_{U_n\times U_n}\|_1 \\
&\quad\leq \|W_n - \widetilde{W}_n^{\phi_n}\|_1 + \|\widetilde{W}_n^{\phi_n} - \widetilde{W}_n^{\phi_n}\mathbf{1}_{U_n\times U_n}\|_1 + \|(\widetilde{W}_n^{\phi_n} - W_n)\mathbf{1}_{U_n\times U_n}\|_1 \\
&\quad\leq \|\widetilde{W}_n^{\phi_n} - \widetilde{W}_n^{\phi_n}\mathbf{1}_{U_n\times U_n}\|_1 + 2\varepsilon_n \\
&\quad= \|\widetilde{W}_n - \widetilde{W}_n\mathbf{1}_{\widetilde{U}_n\times\widetilde{U}_n}\|_1 + 2\varepsilon_n \\
&\quad< \varepsilon/2 + 2\varepsilon_n.
\end{aligned}$$

The right side is less than ε for all sufficiently large $n \in \mathbb{N}$. Therefore $(\mathcal{W}_n)_{n\in\mathbb{N}}$ has uniformly regular tails.

Next assume $(\mathcal{W}_n)_{n\in\mathbb{N}}$ has uniformly regular tails. We consider the two cases (i) and (ii) separately. In case (i) it is immediate from the above result that $(\widetilde{\mathcal{W}}_n)_{n\in\mathbb{N}}$ has uniformly regular tails, since $\|\widetilde{W}_n - W_n^{\phi_n^{-1}}\|_1 < \varepsilon_n$. Now consider case (ii). Given any $\varepsilon > 0$ let $M > 0$ be such that for all $n \in \mathbb{N}$ we can find $U_n \in \mathcal{S}_n$ satisfying $\mu_n(U_n) < M/2$ and $\|W_n - W_n\mathbf{1}_{U_n\times U_n}\|_1 < \varepsilon/5$. Define $\widetilde{U}_n$ by

$$\widetilde{U}_n := \left\{x \in \widetilde{S}_n \,:\, \int_0^1 \mathbf{1}_{(x,s)\in U_n}\, ds > \frac{1}{2}\right\},$$

and define $U_n' := \phi_n^{-1}(\widetilde{U}_n)$. Note that $\widetilde{U}_n$ is a measurable set since $(x, s) \mapsto \mathbf{1}_{(x,s)\in U_n}$ is measurable. Then $\widetilde{\mu}_n(\widetilde{U}_n) < M$, since

$$\mu_n(U_n) = \int_{\widetilde{S}_n}\int_0^1 \mathbf{1}_{(x,s)\in U_n}\, ds\, d\widetilde{\mu}(x) \geq \int_{\widetilde{U}_n}\int_0^1 \mathbf{1}_{(x,s)\in U_n}\, ds\, d\widetilde{\mu}(x) \geq \int_{\widetilde{U}_n} \frac{1}{2}\, d\widetilde{\mu}(x) = \frac{1}{2}\widetilde{\mu}_n(\widetilde{U}_n).$$

Next we will argue that

$$\|\widetilde{W}_n - \widetilde{W}_n\mathbf{1}_{\widetilde{U}_n\times\widetilde{U}_n}\|_1 \leq 2\|\widetilde{W}_n^{\phi_n} - \widetilde{W}_n^{\phi_n}\mathbf{1}_{U_n\times U_n}\|_1. \tag{19}$$

If $(x, x') \in (\widetilde{S}_n \times \widetilde{S}_n)\backslash(\widetilde{U}_n \times \widetilde{U}_n)$ it holds by the definition of $\widetilde{U}_n$ that

$$\int_0^1\int_0^1 \mathbf{1}_{((x,s),(x',s'))\in(S_n\times S_n)\setminus(U_n\times U_n)}\, ds'\, ds = 1 - \int_0^1 \mathbf{1}_{(x,s)\in U_n}\, ds \int_0^1 \mathbf{1}_{(x',s)\in U_n}\, ds \geq \frac{1}{2},$$

which implies (19) by

$$\begin{aligned}
\|\widetilde{W}_n - \widetilde{W}_n \mathbf{1}_{\widetilde{U}_n\times\widetilde{U}_n}\|_1 &= \int_{\widetilde{S}_n\times\widetilde{S}_n} d\mu(x)\,d\mu(x')\left|\widetilde{W}_n(x,x')\right| \mathbf{1}_{(x,x')\in(\widetilde{S}_n\times\widetilde{S}_n)\setminus(\widetilde{U}_n\times\widetilde{U}_n)} \\
&\le 2\int_{\widetilde{S}_n\times\widetilde{S}_n} d\mu(x)\,d\mu(x')\Bigg(\left|\widetilde{W}_n(x,x')\right| \mathbf{1}_{(x,x')\in(\widetilde{S}_n\times\widetilde{S}_n)\setminus(\widetilde{U}_n\times\widetilde{U}_n)} \\
&\qquad \cdot \int_0^1\int_0^1 \mathbf{1}_{((x,s),(x',s'))\in(S_n\times S_n)\setminus(U_n\times U_n)}\,ds'\,ds\Bigg) \\
&\le 2\int_{\widetilde{S}_n\times\widetilde{S}_n} d\mu(x)\,d\mu(x')\left|\widetilde{W}_n(x,x')\right| \int_0^1\int_0^1 \mathbf{1}_{((x,s),(x',s'))\in(S_n\times S_n)\setminus(U_n\times U_n)}\,ds'\,ds \\
&= 2\left\|\widetilde{W}_n^{\phi_n}\mathbf{1}_{(S_n\times S_n)\setminus(U_n\times U_n)}\right\|_1 \\
&= 2\|\widetilde{W}_n^{\phi_n} - \widetilde{W}_n^{\phi_n}\mathbf{1}_{U_n\times U_n}\|_1.
\end{aligned}$$

Using that ϕ_n is measure-preserving, the triangle inequality, that $\|\widetilde{W}_n^{\phi_n} - W_n\|_1 < \varepsilon_n$, and the estimate (19) we get

$$\begin{aligned}
\|\widetilde{W}_n - \widetilde{W}_n\mathbf{1}_{\widetilde{U}_n\times\widetilde{U}_n}\|_1 &\le 2\|\widetilde{W}_n^{\phi_n} - \widetilde{W}_n^{\phi_n}\mathbf{1}_{U_n\times U_n}\|_1 \\
&\le 2\|W_n - W_n\mathbf{1}_{U_n\times U_n}\|_1 + 4\varepsilon_n < 2\varepsilon/5 + 4\varepsilon_n.
\end{aligned}$$

The right side is less than ε for all sufficiently large $n\in\mathbb{N}$, and thus $(\widetilde{\mathcal{W}}_n)_{n\in\mathbb{N}}$ has uniformly regular tails. ∎

Proof of Theorem 15 First we will prove that every $\delta_\square$-Cauchy sequence has uniformly regular tails. Let $(\mathcal{W}_n)_{n\in\mathbb{N}}$ with $\mathcal{W}_n = (W_n, \mathscr{S}_n)$ be a $\delta_\square$-Cauchy sequence of graphons, i.e., $\lim_{n,m\to\infty}\delta_\square(\mathcal{W}_n,\mathcal{W}_m) \to 0$. By Lemma 58 we may assume without loss of generality that $\mathscr{S}_n$ is atomless for all $n\in\mathbb{N}$. By Lemmas 46 and 33 we can find graphons $\widetilde{\mathcal{W}}_n = (\widetilde{W}_n, \mathbb{R}_+)$ and measure-preserving maps $\psi_n\colon S_n\to\mathbb{R}_+$ such that $W_n = (\widetilde{W}_n)^{\psi_n}$. Since $\delta_\square(\widetilde{\mathcal{W}}_n,\mathcal{W}_n) = 0$, $(\widetilde{\mathcal{W}}_n)_{n\in\mathbb{N}}$ is a Cauchy sequence. Given any $\varepsilon>0$ choose $N\in\mathbb{N}$ such that $\delta_\square(\widetilde{\mathcal{W}}_N,\widetilde{\mathcal{W}}_n) < \varepsilon/4$ for all $n\ge N$. For each $n\le N$ let $M_n\in\mathbb{R}_+$ be such that $\|\widetilde{W}_n - \widetilde{W}_n\mathbf{1}_{[0,M_n]^2}\|_1 < \varepsilon/3$, and define $M := \sup_{n\le N} M_n < \infty$. To prove that $(\widetilde{\mathcal{W}}_n)_{n\in\mathbb{N}}$ has uniformly regular tails it is sufficient to prove that for each $n\ge N$ we can find a Borel-measurable set $\widetilde{A}_n\subset\mathbb{R}_+$ such that

$$\lambda(\widetilde{A}_n)\le M, \qquad \|\widetilde{W}_n - \widetilde{W}_n\mathbf{1}_{\widetilde{A}_n\times\widetilde{A}_n}\|_1 < \varepsilon. \tag{20}$$

We can clearly find an appropriate set $\widetilde{A}_n$ for $n=N$; indeed, we can find a set $\widetilde{A}_N\subset\mathbb{R}_+$ such that the second bound holds with $\varepsilon/3$ instead of ε. By Proposition 48(c) we can find isomorphisms $\phi_n\colon\mathbb{R}_+\to\mathbb{R}_+$ such that $\|\widetilde{W}_N - \widetilde{W}_n^{\phi_n}\|_\square < \varepsilon/3$ for all $n\ge N$. Define $\widetilde{A}_n = \phi_n(\widetilde{A}_N)$, and note that

$$\|\widetilde{W}_n - \widetilde{W}_n\mathbf{1}_{\widetilde{A}_n\times\widetilde{A}_n}\|_\square = \|\widetilde{W}_n^{\phi_n} - \widetilde{W}_n^{\phi_n}\mathbf{1}_{\widetilde{A}_N\times\widetilde{A}_N}\|_\square \le \|\widetilde{W}_N - \widetilde{W}_N\mathbf{1}_{\widetilde{A}_N\times\widetilde{A}_N}\|_\square + \frac{2\varepsilon}{3} < \varepsilon.$$

Observing that for non-negative graphons the cut norm is equal to the L^1 norm, this gives that (20) is satisfied and $(\widetilde{\mathcal{W}}_n)_{n\in\mathbb{N}}$ has uniformly regular tails. Defining $A_n := \psi^{-1}(\widetilde{A}_n)$, we

have $\mu(A_n) < M$ and $\|W_n - W_n\mathbf{1}_{A_n\times A_n}\|_1 = \|\widetilde{W}_n - \widetilde{W}_n\mathbf{1}_{\widetilde{A}_n\times\widetilde{A}_n}\|_1 < \varepsilon$. Hence $(\mathcal{W}_n)_{n\in\mathbb{N}}$ has uniformly regular tails.

Now we will prove that uniform regularity of tails implies subsequential convergence for $\delta_\square$. We consider some sequence of graphons $(\mathcal{W}_n)_{n\in\mathbb{N}}$ with uniformly regular tails, and will prove that the sequence is subsequentially convergent for $\delta_\square$ towards some graphon $\mathcal{W}$. By Lemma 58 we may assume without loss of generality that $\mathscr{S}_n$ is atomless for all $n\in\mathbb{N}$, and by trivially extending $\mathcal{W}_n$ to a graphon over a space of infinite total mass if needed, we may assume that $\mu_n(S_n) = \infty$. Recall the definition of a partition of a measurable space, which was given as part of the discussion before the statement of Proposition 20. We will prove that we can find increasing sequences $(m_k)_{k\in\mathbb{N}}$ and $(M_k)_{k\in\mathbb{N}}$ with values in $\mathbb{N}$, such that for each $k,n\in\mathbb{N}$ there is a partition $\mathcal{P}_{n,k}$ of S_n and a graphon $\mathcal{W}_{n,k} = (W_{n,k},\mathbb{R}_+)$ such that the following hold:

(i) We have $\mathcal{P}_{n,k} = \{I^i_{n,k} : i = 0,\dots,m_k\}$, where $\mu_n(S_n\setminus I^0_{n,k}) = M_k$ and $\mu_n(I^i_{n,k}) = M_k/m_k$ for $i\in\{1,\dots,m_k\}$.

(ii) We have $\delta_\square(\mathcal{W}_n,\mathcal{W}_{n,k}) < 1/k$ for all $n\in\mathbb{N}$.

(iii) For each $i_1,i_2\in\{1,\dots,m_k\}$ the value of $W_{n,k}$ on $([i_1-1,i_1)\times[i_2-1,i_2))M_k/m_k$ is constant and equal to the value of $(W_n)_{\mathcal{P}_{n,k}}$ on $I^{i_1}_{n,k}\times I^{i_2}_{n,k}$. On the complement of $[0,M_k]^2$, we have $W_{n,k}=0$.

(iv) The partition $\mathcal{P}_{n,k+1}$ refines the partition $\mathcal{P}_{n,k}$. We number the elements of the partition $\mathcal{P}_{n,k+1}$ to be consistent with the refinement. More precisely, defining $r_k := (M_k/m_k)/(M_{k+1}/m_{k+1})\in\mathbb{N}$ to be the ratio of the partition sizes in the two partitions, we have $I^i_{n,k} = \bigcup_{j=(i-1)r_k+1}^{ir_k} I^j_{n,k+1}$ for every i with $0<i\le m_k$.

Partitions $\mathcal{P}_{n,k}$ and graphons $\mathcal{W}_{n,k}$ satisfying (i)–(iv) exist by the following argument. By the assumption of uniformly regular tails, for each $k\in\mathbb{N}$ we can find an $M_k\in\mathbb{N}$ such that for appropriate sets $I^0_{n,k}$ satisfying $\mu_n(S_n\setminus I^0_{n,k}) = M_k$ we have $\|W_n - W_n\mathbf{1}_{(S_n\setminus I^0_{n,k})\times(S_n\setminus I^0_{n,k})}\|_1 < 1/(3k)$ for all $n\in\mathbb{N}$. By Lemmas 46 and 33, for each $n,k\in\mathbb{N}$ the graphon $(W_n|_{(S_n\setminus I^0_{n,k})\times(S_n\setminus I^0_{n,k})}, S_n\setminus I^0_{n,k})$ is a pullback of a graphon $\widetilde{\mathcal{W}}_{n,k} = (\widetilde{W}_{n,k},[0,M_k])$ by a measure-preserving transformation $\varphi_{n,k}$. By applying Szemerédi regularity for equitable partitions to $\widetilde{W}_{n,k}$ (see, for example, the paper of Borgs, Chayes, Cohn, and Zhao, 2014a, Lemma 3.3) we can find appropriate $m_k\in\mathbb{N}$ and partitions $\widetilde{\mathcal{P}}_{n,k}$ of $[0,M_k]$ such that $\|(\widetilde{W}_{n,k} - (\widetilde{W}_{n,k})_{\widetilde{\mathcal{P}}_{n,k}})\|_\square < 1/(3k)$. Then the pullback of $(\widetilde{W}_{n,k})_{\widetilde{\mathcal{P}}_{n,k}}$ along $\varphi_{n,k}$ equals $(W_n)_{\mathcal{P}_{n,k}}$ for an appropriate partition of S_n satisfying (i), and

$$\begin{aligned}
\|W_n - (W_n)_{\mathcal{P}_{n,k}}\|_\square &\le \|W_n - W_n\mathbf{1}_{(S_n\setminus I^0_{n,k})\times(S_n\setminus I^0_{n,k})}\|_\square \\
&\quad + \|(W_n - (W_n)_{\mathcal{P}_{n,k}})\mathbf{1}_{(S_n\setminus I^0_{n,k})\times(S_n\setminus I^0_{n,k})}\|_\square \\
&\quad + \|(W_n)_{\mathcal{P}_{n,k}}\mathbf{1}_{(S_n\setminus I^0_{n,k})\times(S_n\setminus I^0_{n,k})} - (W_n)_{\mathcal{P}_{n,k}}\|_\square \\
&= \|W_n - W_n\mathbf{1}_{(S_n\setminus I^0_{n,k})\times(S_n\setminus I^0_{n,k})}\|_\square + \|\widetilde{W}_{n,k}^{\varphi_{n,k}} - (\widetilde{W}_{n,k})^{\varphi_{n,k}}_{\widetilde{\mathcal{P}}_{n,k}}\|_\square \\
&\quad + \left\|\left(W_n\mathbf{1}_{(S_n\setminus I^0_{n,k})\times(S_n\setminus I^0_{n,k})} - W_n\right)_{\mathcal{P}_{n,k}}\right\|_\square \\
&< 1/k.
\end{aligned}$$

Define $\mathcal{W}_{n,k}$ as described in (iii), and note that all requirements (i)–(iv) are satisfied since $\delta_\square(((W_n)_{\mathcal{P}_{n,k}},\mathscr{S}_n),\mathcal{W}_{n,k})=0$.

By compactness, for each $k\in\mathbb{N}$ there exists a step function $U_k\colon\mathbb{R}_+^2\to[0,1]$ (with support in $[0,M_k]^2$) such that $(W_{n,k})_{n\in\mathbb{N}}$ converges pointwise and in L^1 along a subsequence towards U_k. We may assume the subsequence along which $(W_{n,k+1})_{n\in\mathbb{N}}$ converges is contained in the subsequence along which $(W_{n,k})_{n\in\mathbb{N}}$ converges. Note that for each $i_1,i_2\in\{1,\dots,m_k\}$ the function U_k is constant on $([i_1-1,i_1]\times[i_2-1,i_2])M_k/m_k$. Furthermore, observe that if $k,k'\in\mathbb{N}$ and $k'\geq k$, the value of U_k at $([i_1-1,i_1]\times[i_2-1,i_2])M_k/m_k$ is equal to the average of $U_{k'}$ over this set. Define the graphon $\mathcal{U}_k$ by $\mathcal{U}_k:=(U_k,\mathbb{R}_+)$.

Choose $M>1$, and then choose k' such that $M_k\geq M_{k'}\geq M$ for all $k\geq k'$. Let (X,Y) be a uniformly random point in $[0,M_{k'}]^2$. By the observations in the preceding paragraph $(U_k(X,Y))_{k\geq k'}$ is a martingale. Hence the martingale convergence theorem implies that the limit $\lim_{k\to\infty}U_k(X,Y)$ exists a.s. Since M was arbitrary it follows that there is a set $E\subset\mathbb{R}_+^2$ of measure zero outside of which $(U_k)_{k\geq k'}$ converges pointwise. Define the graphon $\mathcal{U}:=(U,\mathbb{R}_+)$ as follows. For any $(x_1,x_2)\in\mathbb{R}_+^2\backslash E$ define $U(x_1,x_2):=\lim_{k\to\infty}U_k(x_1,x_2)$, and for any $(x_1,x_2)\in E$ define $U(x_1,x_2):=0$. Since the functions U_k are uniformly bounded, martingale convergence also implies that $U_k|_{[0,M_\ell]^2}$ converges to $U|_{[0,M_\ell]^2}$ in L^1 for each $\ell\in\mathbb{N}$.

Next we will show that $\lim_{k\to\infty}\|U_k-U\|_1=0$. Since $\lim_{k\to\infty}\|(U_k-U)\mathbf{1}_{[0,M_\ell]^2}\|_1=0$ for each $\ell\in\mathbb{N}$ it is sufficient to prove that $\|U_k\mathbf{1}_{\mathbb{R}_+^2\backslash[0,M_\ell]^2}\|_1<1/(3\ell)$ for all $k,\ell\in\mathbb{N}$ for which $k>\ell$. This follows by Fatou's lemma and the inequality

$$\begin{aligned}\int_{\mathbb{R}_+^2\backslash[0,M_\ell]^2}|W_{n,k}|&=\int_{[0,M_k]^2\backslash[0,M_\ell]^2}|W_{n,k}|\leq\int_{(S_n\backslash I^0_{n,k})^2\backslash(S_n\backslash I^0_{n,\ell})^2}|W_n|_{\mathcal{P}_{n,k}}\\&=\int_{(S_n\backslash I^0_{n,k})^2\backslash(S_n\backslash I^0_{n,\ell})^2}|W_n|<1/(3\ell).\end{aligned}$$

By the result of the preceding paragraph

$$\limsup_{k\to\infty}\delta_\square(\mathcal{U}_k,\mathcal{U})\leq\limsup_{k\to\infty}\|U_k-U\|_1=0,$$

and we conclude the proof by applying the triangle inequality to obtain

$$\liminf_{n\to\infty}\delta_\square(\mathcal{U},\mathcal{W}_n)\leq\limsup_{k\to\infty}\liminf_{n\to\infty}\Big(\delta_\square(\mathcal{U},\mathcal{U}_k)+\delta_\square(\mathcal{U}_k,\mathcal{W}_{n,k})+\delta_\square(\mathcal{W}_{n,k},\mathcal{W}_n)\Big)=0.$$

■

Appendix F. Basic Properties of Metric Convergent Sequences of Graphs

In this appendix we will establish Propositions 20 and 22. First we prove a lemma saying that for a set of graphs with uniformly regular tails we may assume the sets U in Definition 13 correspond to sets of vertices.

Lemma 59 *Let $\mathcal{G}$ be a set of graphs with uniformly regular tails. For every $\varepsilon>0$ there is an $M>0$ such that for each $G\in\mathcal{G}$ we can find a set $U\subset\mathbb{R}_+$ corresponding to a set of vertices for G such that $\|W^{G,s}-W^{G,s}\mathbf{1}_{U\times U}\|_1<\varepsilon$ and $\lambda(U)<M$.*

Proof Since $\mathcal{G}$ has uniformly regular tails we can find an $M>0$ such that for each $G\in\mathcal{G}$ there is a set $\widetilde{U}\subset\mathbb{R}_+$ (not necessarily corresponding to a set of vertices for G) such that $\|W^{G,s}-W^{G,s}\mathbf{1}_{\widetilde{U}\times\widetilde{U}}\|_1<\varepsilon/2$ and $\lambda(\widetilde{U})<M/2$. Recall that each vertex $i\in V(G)$ corresponds to an interval $I_i\subset\mathbb{R}_+$ for the stretched canonical graphon $W^{G,s}$, such that $\lambda(I_i)$ is proportional to the weight of the vertex. Given a set $\widetilde{U}\subset\mathbb{R}_+$ as above, define

$$U:=\bigcup_{i\in\mathcal{I}}I_i,\quad\text{where}\quad\mathcal{I}:=\{i\in V(G)\,:\,2\lambda(I_i\cap\widetilde{U})>\lambda(I_i)\}.$$

The lemma now follows by observing that $\lambda(U)\leq2\lambda(\widetilde{U})<M$ and

$$\begin{aligned}\|W^{G,s}-W^{G,s}\mathbf{1}_{U\times U}\|_1&=\sum_{i,j\in V(G)\,:\,(i,j)\notin\mathcal{I}\times\mathcal{I}}\beta_{i,j}\lambda(I_i)\lambda(I_j)\\&\leq2\sum_{i,j\in V(G)\,:\,(i,j)\notin\mathcal{I}\times\mathcal{I}}\beta_{ij}\big(\lambda(I_i)\lambda(I_j)-\lambda(I_i\cap\widetilde{U})\lambda(I_j\cap\widetilde{U})\big)\\&\leq2\|W^{G,s}-W^{G,s}\mathbf{1}_{\widetilde{U}\times\widetilde{U}}\|_1\\&<\varepsilon.\end{aligned}$$

■

Proof of Proposition 20 Define $M_n:=\inf\{M>0\,:\,\operatorname{supp}(W^{G_n,s})\subseteq[0,M]^2\}$. If $(G_n)_{n\in\mathbb{N}}$ is sparse, then $\liminf_{n\to\infty}M_n=\infty$. By Lemma 59, if $(G_n)_{n\in\mathbb{N}}$ has uniformly regular tails there exists an $M'>0$ such that if we order the vertices of G_n appropriately when defining the canonical graphon W^{G_n} of G_n, then $\|W^{G_n,s}\mathbf{1}_{[0,M']^2}\|_1>1/2$ for all $n\in\mathbb{N}$.

The graphons $\mathcal{W}^{G_n,r}$ and $\mathcal{W}^{G_n,s}$ are related by $W^{G_n,r}=\widetilde{M}_n^2W^{G_n,s}(\widetilde{M}_n\,\cdot\,,\widetilde{M}_n\,\cdot\,)$ for some $\widetilde{M}_n\geq M_n$ (with $\widetilde{M}_n=M_n$ if G_n has no isolated vertices; if G_n has isolated vertices corresponding to the end of the interval $[0,1]$ for the canonical graphon W^{G_n} we will have $\widetilde{M}_n>M_n$). If $\lim_{n\to\infty}\widetilde{M}_n=\infty$ and $\|W^{G_n,s}\mathbf{1}_{[0,M']^2}\|_1>1/2$ for all $n\in\mathbb{N}$, then

$$\|W^{G_n,r}\mathbf{1}_{[0,a_n]^2}\|_1>1/2\qquad\text{and}\qquad\lim_{n\to\infty}a_n=0,\qquad\text{where }a_n:=\min\big(M'\widetilde{M}_n^{-1},1\big).\quad(21)$$

The proof of (i) is complete if we can prove that (21) implies that $(G_n)_{n\in\mathbb{N}}$ is not uniformly upper regular. Assume the opposite, and let $K\colon(0,\infty)\to(0,\infty)$ and $(\eta_m)_{n\in\mathbb{N}}$ be as in the definition of uniform upper regularity. Let $\mathcal{P}_n$ be a partition of $\mathbb{R}_+$ such that one of the parts is $[0,a'_n]$, where $a'_n\geq a_n$ is chosen as small as possible such that $[0,a'_n]$ corresponds to an integer number of vertices of G_n for the canonical graphon. Then $\lim_{n\to\infty}a'_n=0$ since $\lim_{n\to\infty}a_n=0$ and $\lim_{n\to\infty}V(G_n)=\infty$. By the first part of (21) it follows that $(W^{G_n,r})_{\mathcal{P}_n}>K(1/2)$ on $[0,a'_n]^2$ for all sufficiently large n; hence for all sufficiently large n,

$$\|(W^{G_n,r})_{\mathcal{P}_n}\mathbf{1}_{|(W^{G_n,r})_{\mathcal{P}_n}|\geq K(1/2)}\|_1\geq\|(W^{G_n,r})_{\mathcal{P}_n}\mathbf{1}_{[0,a'_n]^2}\|_1=\|W^{G_n,r}\mathbf{1}_{[0,a'_n]^2}\|_1>1/2.$$

We have obtained a contradiction to the assumption of uniform upper regularity, and thus the proof of (i) is complete.

Defining $\rho_n:=\rho(G_n)$ (recall the definition of ρ in the beginning of Section 2), we have $W^{G_n,s}=W^{G_n}(\rho_n^{1/2}\,\cdot\,,\rho_n^{1/2}\,\cdot\,)$ and $W^{G_n,r}=\rho_n^{-1}W^{G_n}$. If $(G_n)_{n\in\mathbb{N}}$ is dense and has convergent

edge density the following limit exists and is positive: $\rho := \lim_{n\to\infty} \rho_n > 0$. It follows by Lemma 44 (resp. Lemma 36) that $(G_n)_{n\in\mathbb{N}}$ is a Cauchy sequence for $\delta^s_\square$ (resp. $\delta^r_\square$) iff $(\mathcal{W}^{G_n})_{n\in\mathbb{N}}$ is a Cauchy sequence for $\delta_\square$, since for any $n, m \in \mathbb{N}$,

$$\begin{aligned}\Big|\delta^s_\square(G_n, G_m) - \delta_\square\big((W^{G_n}(\rho^{1/2}\cdot\,,\rho^{1/2}\cdot\,),\mathbb{R}_+),(W^{G_m}(\rho^{1/2}\cdot\,,\rho^{1/2}\cdot\,),\mathbb{R}_+)\big)\Big| \\ \leq \delta_\square\big(W^{G_n,s},(W^{G_n}(\rho^{1/2}\cdot\,,\rho^{1/2}\cdot\,),\mathbb{R}_+)\big) \\ + \delta_\square\big((W^{G_m}(\rho^{1/2}\cdot\,,\rho^{1/2}\cdot\,),\mathbb{R}_+),W^{G_m,s}\big) \\ \to 0\end{aligned}$$

as $n, m \to \infty$, and a similar estimate holds with $\delta^r_\square$ instead of $\delta^s_\square$. This completes the proof of the first assertion of (ii).

To prove the second assertion of (ii) consider the following sequence of dense graphs $(G_n)_{n\in\mathbb{N}}$, which is a Cauchy sequence for $\delta^s_\square$ but not for $\delta^r_\square$. For odd n let G_n be a complete simple graph on n vertices, and for even n let G_n be the union of a complete graph on $n/2$ vertices and $n/2$ isolated vertices. This sequence converges to $\mathcal{W}_1 := (\mathbf{1}_{[0,1]^2}, \mathbb{R}_+)$ for $\delta^s_\square$, but does not converge for $\delta^r_\square$.

Conversely, the following sequence of dense graphs $(G_n)_{n\in\mathbb{N}}$ is a Cauchy sequence for $\delta^r_\square$ but not for $\delta^s_\square$. For odd n let G_n be a complete graph on n vertices, and for even n let G_n be an Erdős-Rényi graph with edge probability $1/2$. This sequence converges to $\mathcal{W}_1$ for $\delta^r_\square$, but does not converge for $\delta^s_\square$. ■

Proof of Proposition 22 We will assume throughout the proof that the graphs have no isolated vertices, since the case of $o|E(G_n)|$ vertices clearly follows from this case. We assume the average degree of $(G_n)_{n\in\mathbb{N}}$ is bounded above by $d \in \mathbb{N}$, and want to obtain a contradiction. When defining the canonical stretched graphon $W^{G_n,s}$ of G_n, each vertex of G_n corresponds to an interval of length $1/\sqrt{2|E(G_n)|}$. Since $|E(G_n)|/|V(G_n)| \leq d/2$ by assumption, the vertices of G_n correspond to an interval of length $|V(G_n)|/\sqrt{2|E(G_n)|} \geq \sqrt{2|E(G_n)|}/d$, which is too stretched out to be compatible with uniformly regular tails. Explicitly, given that G_n has no isolated vertices it follows that for any $M > 0$ and any Borel set $I \subset \mathbb{R}_+$ satisfying $\lambda(I) < M$,

$$\int_{(\mathbb{R}_+\setminus I)\times\mathbb{R}_+} W^{G_n,s} \geq \frac{\sqrt{2|E(G_n)|}/d - M}{\sqrt{2|E(G_n)|}}.$$

By the assumption that $\lim_{n\to\infty}|E(G_n)| = \infty$, the right side of this equation is greater than $1/(2d)$ for all sufficiently large $n \in \mathbb{N}$. Since $M > 0$ was arbitrary, this is not compatible with G_n having uniformly regular tails, which together with Theorem 15 gives a contradiction. ■

Appendix G. Exchangeability of Graphon Processes

The main goal of this appendix is to prove Theorem 26.

Lemma 60 *Let $(\widetilde{G}_n)_{n\in\mathbb{N}}$ be a sequence of simple graphs with uniformly regular tails, such that $|E(\widetilde{G}_n)| < \infty$ for all $n \in \mathbb{N}$ and $\lim_{n\to\infty}|E(\widetilde{G}_n)| = \infty$. Fix $d \in \mathbb{N}$, and for each $n \in \mathbb{N}$ let $\widehat{G}_n$ be an induced subgraph of $\widetilde{G}_n$ where all or some of the vertices of degree at most d are removed. Then $\lim_{n\to\infty}|E(\widehat{G}_n)|/|E(\widetilde{G}_n)| = 1$.*

Proof We wish to prove that $\varepsilon := \limsup_{n\to\infty}\varepsilon_n = 0$, where $\varepsilon_n := 1 - |E(\widehat{G}_n)|/|E(\widetilde{G}_n)|$. By taking a subsequence we may assume $\varepsilon = \lim_{n\to\infty}\varepsilon_n$. We will carry out the proof by contradiction, and assume $\varepsilon > 0$. By definition of $\widehat{G}_n$ there are at least $\varepsilon_n|E(\widetilde{G}_n)|$ edges of $\widetilde{G}_n$ which have at least one endpoint of degree at most d. Hence there are at least $\varepsilon_n|E(\widetilde{G}_n)|/d$ vertices with degree between 1 and d. In the canonical stretched graphon of $\widetilde{G}_n$, each vertex corresponds to an interval of length $(2|E(\widetilde{G}_n)|)^{-1/2}$. Hence the total length of the intervals corresponding to vertices of degree between 1 and d is at least $2^{-1/2}\varepsilon_n|E(\widetilde{G}_n)|^{1/2}d^{-1}$, which tends to infinity as $n \to \infty$. It follows that for each $M > 0$ and any sets $U_n \subset \mathbb{R}_+$ of measure at most M,

$$\|W^{\widetilde{G}_n,s} - W^{\widetilde{G}_n,s}\mathbf{1}_{U_n\times U_n}\|_1 \geq \big(2^{-1/2}\varepsilon_n|E(\widetilde{G}_n)|^{1/2}d^{-1} - M\big)\cdot(2|E(\widetilde{G}_n)|)^{-1/2},$$

which is at least $\varepsilon(2^{3/2}d)^{-1}$ when n is sufficiently large. Thus $(\widetilde{G}_n)_{n\in\mathbb{N}}$ does not have uniformly regular tails, and we have obtained the desired contradiction. ■

We will now prove Theorem 26. Note that we use a result of Kallenberg (2005, Theorem 9.25) for part of the argument, a result which is also used by Veitch and Roy (2015), but that we use it to prove Theorem 26, which characterizes exchangeable random graphs that have uniformly regular tails, while Veitch and Roy (2015) use it to characterize exchangeable random graphs that have finitely many edges for each finite time, but which do not necessarily have uniformly regular tails (a notion not considered by Veitch and Roy, 2015).

Proof of Theorem 26 First assume $(\widetilde{G}_t)_{t\geq 0}$ is a graphon process generated by $\mathcal{W}_\alpha$ with isolated vertices, where α is a random variable. We want to prove that $(\widetilde{G}_t)_{t\geq 0}$ has uniformly regular tails, and that the measure ξ is exchangeable. Regularity of tails is immediate from Theorems 28(i) and 15. Exchangeability is immediate by observing that the Poisson random measure $\mathcal{V}$ on $\mathbb{R}_+ \times S$ defined in the beginning of Section 2.4 is identical in law to $\{(\phi(t), x) : (t, x) \in \mathcal{V}\}$ for any measure-preserving transformation $\phi\colon \mathbb{R}_+ \to \mathbb{R}_+$ (in particular, for the case when ϕ corresponds to a permutation of intervals).

To prove the second part of the theorem assume that ξ is jointly exchangeable and that $(\widetilde{G}_t)_{t\geq 0}$ has uniformly regular tails. By joint exchangeability of ξ it follows from the representation theorem for jointly exchangeable random measures on $\mathbb{R}^2_+$ (Kallenberg, 2005, Theorem 9.24) that a.s.

$$\begin{aligned}\xi = &\sum_{i,j} f(\alpha, x_i, x_j, \zeta_{\{i,j\}})\delta_{t_i,t_j} + \beta\lambda_D + \gamma\lambda^2 \\ &+ \sum_{j,k}\big(g(\alpha, x_j, \chi_{j,k})\delta_{t_j,\sigma_{j,k}} + g'(\alpha, x_j, \chi_{j,k})\delta_{\sigma_{j,k},t_j}\big) \\ &+ \sum_j\big(h(\alpha, x_j)(\delta_{t_j}\otimes\lambda) + h'(\alpha, x_j)(\lambda\otimes\delta_{t_j})\big) \\ &+ \sum_k\Big(l(\alpha,\eta_k)\delta_{\rho_k,\rho'_k} + l'(\alpha,\eta_k)\delta_{\rho'_k,\rho_k}\Big),\end{aligned} \tag{22}$$

for some measurable functions $f \geq 0$ on $\mathbb{R}_+^4$, $g, g' \geq 0$ on $\mathbb{R}_+^3$, and $h, h', l, l' \geq 0$ on $\mathbb{R}_+^2$, a set of independent uniform random variables $(\zeta_{\{i,j\}})_{i,j\in\mathbb{N}}$ with values in $[0,1]$, independent unit rate Poisson processes $(t_j, x_j)_{j\in\mathbb{N}}$ and $(\sigma_{i,j}, \chi_{i,j})_{j\in\mathbb{N}}$ on $\mathbb{R}_+^2$ for $i \in \mathbb{N}$ and $(\rho_j, \rho'_j, \eta_j)_{j\in\mathbb{N}}$ on $\mathbb{R}_+^3$, an independent set of random variables $\alpha, \beta, \gamma \geq 0$, and λ (resp. λ_D) denoting Lebesgue measure on $\mathbb{R}_+$ (resp. the diagonal $x_1 = x_2 \geq 0$).

By the definition (4) in Section 2.4 of ξ as a sum of point masses, all the terms in (22) involving Lebesgue measure must be zero; i.e., except on an event of probability zero, $\beta = \gamma = 0$ and $h(\alpha, x_j) = h'(\alpha, x_j) = 0$ for all $j \in \mathbb{N}$. Recall that by (5) each vertex can be uniquely identified with the time $t \geq 0$ when it appeared in the graph, and that each point mass $\delta_{t,t'}$ with $t, t' \geq 0$ represents an edge between the two vertices associated with t and t'. Almost surely no two of the random variables $\rho_k, \rho'_k, t_i, t_j, \sigma_{j,k}$ for $i, j, k \in \mathbb{N}$ have the same value, and hence the functions f, g, g', l, l' take values in $\{0, 1\}$ almost everywhere. Furthermore, since the graphs $\widetilde{G}_t$ are undirected, we have $g = g'$ and $l = l'$, and f is symmetric in its second and third input argument.

First we will argue that the subgraphs $\widehat{G}_t$ of $\widetilde{G}_t$ corresponding to the terms

$$f(\alpha, x_i, x_j, \zeta_{\{i,j\}})\delta_{t_i,t_j}$$

have the law of a graphon process with isolated vertices generated by some (possibly random) graphon $\mathcal{W}$. Condition on the realization of α, and define the function $W_\alpha \colon \mathbb{R}_+^2 \to [0,1]$ by

$$W_\alpha(x, x') := \mathbb{P}(f(\alpha, x, x', \zeta_{\{i,j\}}) = 1 \,|\, \alpha) \qquad \text{for all } x, x' \in \mathbb{R}_+.$$

It follows that, conditioned on α such that $W_\alpha \in L^1$, $(\widehat{G}_t)_{t\geq 0}$ has the law of a graphon process generated by $\mathcal{W}_\alpha = (W_\alpha, \mathbb{R}_+)$. To conclude we need to prove that $W_\alpha \in L^1$ almost surely, which will be done in the next two paragraphs.

First we will argue that $(\widehat{G}_t)_{t\geq 0}$ has uniformly regular tails. Since no two of the random variables $\rho_k, \rho'_k, t_i, t_j, \sigma_{j,k}$ have the same value for $i, j, k \in \mathbb{N}$, each point mass δ_{ρ_k,ρ'_k} or $\delta_{\rho'_k,\rho_k}$ of ξ corresponds to an isolated edge, i.e., an edge between two vertices each of degree one, and each point mass $\delta_{\sigma_{j,k},t_j}$ or $\delta_{t_j,\sigma_{j,k}}$ of ξ corresponds to an edge between the vertex t_j in $\widehat{G}_t$ and a vertex of degree one; i.e., $\sum_{k\in\mathbb{N}}(\delta_{t_j,\sigma_{j,k}} + \delta_{\sigma_{j,k},t_j})$ corresponds to a star centered at the vertex associated with t_j. Note that $\widehat{G}_t$ and $\widetilde{G}_t$ satisfy the conditions of Lemma 60 with $d = 1$. Hence $\lim_{t\to\infty} |E(\widehat{G}_t)|/|E(\widetilde{G}_t)| = 1$, and since $\widetilde{G}_t$ has uniformly regular tails this implies that $\widehat{G}_t$ must also have uniformly regular tails.

We assume that W_α is not almost surely integrable, and will derive a contradiction. We condition on α such that $W_\alpha \notin L^1$, and to simplify notation we will write W instead of W_α. Let $\widehat{G}_t^+$ (resp. $\widehat{G}_t^-$) be the induced subgraph of $\widehat{G}_t$ consisting of the vertices for which the feature x satisfies $x \in I := \{x' \in \mathbb{R}_+ : \int_{\mathbb{R}_+} W(x, x')\, dx' \geq 1\}$ (resp. $x \notin I$). Let $\mathcal{W}^+ = (W^+, \mathbb{R}_+)$ and $\mathcal{W}^- = (W^-, \mathbb{R}_+)$ denote the corresponding graphons (we will see shortly that they are integrable), i.e., $W^+ = W\mathbf{1}_{I\times I}$ and $W^- = W\mathbf{1}_{I^c\times I^c}$. Since $|E(\widehat{G}_t)| < \infty$ a.s. for each $t \geq 0$ the measure ξ is locally finite a.s. We deduce from this that $\lambda(I) < \infty$ and $\|W^-\|_1 < \infty$ (Kallenberg, 2005, Theorem 9.25, (iii) and (iv)). By applying Proposition 56 this implies further that

$$\lim_{t\to\infty} t^{-2}|E(\widehat{G}_t^+)| = \frac{1}{2}\|W^+\|_1 \leq \frac{1}{2}\lambda(I)^2 < \infty,$$

$$\lim_{t\to\infty} t^{-2}|E(\widehat{G}_t^-)| = \frac{1}{2}\|W^-\|_1 < \infty,$$

and

$$\lim_{t\to\infty} t^{-2}|E(\widehat{G}_t)| = \infty.$$

It follows that if $\widetilde{E}(\widehat{G}_t) := E(\widehat{G}_t)\backslash(E(\widehat{G}_t^+) \cup E(\widehat{G}_t^-))$ is the set of edges having one endpoint in $V(\widehat{G}_t^+)$ and one endpoint in $V(\widehat{G}_t^-)$, we have

$$\lim_{t\to\infty} |\widetilde{E}(\widehat{G}_t)|/|E(\widehat{G}_t)| = 1. \tag{23}$$

For the stretched canonical graphon $\mathcal{W}^{\widehat{G}_t,s}$ the edges $\widetilde{E}(\widehat{G}_t)$ correspond to $A := (J_t \times J_t^c) \cup (J_t^c \times J_t) \subset \mathbb{R}_+^2$, where $J_t \subset \mathbb{R}_+$ corresponds to $V(\widehat{G}_t^+)$. Since $|V(\widehat{G}_t^+)| = \Theta(t)$, we have $\lambda(J_t) = |V(\widehat{G}_t^+)|(2|E(\widehat{G}_t)|)^{-1/2} = o_t(1)$. By (23) and $\|W^{\widehat{G}_t,s}\|_1 = 1$, we have $\lim_{t\to\infty} \|W^{\widehat{G}_t,s}\mathbf{1}_{A^c}\|_1 = 0$. Since $\lambda(J_t) = o_t(1)$ and $W^{\widehat{G}_t,s}$ takes values in $[0,1]$, we have $\lim_{t\to\infty} \|W^{\widehat{G}_t,s}\mathbf{1}_{A\cap U_t^2}\|_1 = 0$ for all sets $U_t \subset \mathbb{R}_+$ of bounded measure. We have obtained a contradiction to the hypothesis of uniform regularity of tails, since

$$\lim_{t\to\infty} \|W^{\widehat{G}_t,s}\mathbf{1}_{U_t^2}\|_1 \leq \lim_{t\to\infty} \|W^{\widehat{G}_t,s}\mathbf{1}_{U_t^2\cap A}\|_1 + \lim_{t\to\infty} \|W^{\widehat{G}_t,s}\mathbf{1}_{A^c}\|_1 = 0.$$

To complete the proof that $(\widetilde{G}_t)_{t\geq 0}$ has the law of a graphon process with isolated vertices, we need to argue that a.s.

$$l(\alpha, \eta_k) = g(\alpha, x_j, \chi_{j,k}) = 0 \qquad \text{for all } k, j \in \mathbb{N}. \tag{24}$$

Let $N_t \in \mathbb{N}_0$ denote the number of edges associated with terms of ξ of the form $\delta_{\rho_k,\rho'_k} + \delta_{\rho'_k,\rho_k}$, and let $\widetilde{N}_t$ denote the number of edges associated with terms of ξ of the form $\delta_{\sigma_{j,k},t_j} + \delta_{t_j,\sigma_{j,k}}$. Since $\widehat{G}_t$ and $\widetilde{G}_t$ satisfy the conditions of Lemma 60 with $d = 1$, and since Lemma 56 implies that a.s.-$\lim_{t\to\infty} |E(\widetilde{G}_t)|/t^2 = \frac{1}{2}\|W\|_1$, it follows that a.s.

$$\lim_{t\to\infty} N_t/t^2 = 0 \quad \text{and} \quad \lim_{t\to\infty} \widetilde{N}_t/t^2 = 0. \tag{25}$$

We will prove (24) by contradiction, and will consider each term separately. First assume $\lambda(\mathrm{supp}(l(\alpha, \cdot))) > 0$ with positive probability. Conditioned on a realization of α such that $p := \lambda(\mathrm{supp}(l(\alpha, \cdot))) > 0$, the random variable N_t is a Poisson random variable with expectation $t^2 p$. Hence $\lim_{t\to\infty} N_t/t^2 = p$, which is a contradiction to (25). It follows that $\lambda(\mathrm{supp}(l(\alpha, \cdot))) = 0$ a.s., and thus $l(\alpha, \eta_k) = 0$ for all $k \in \mathbb{N}$ a.s.

Now assume $\lambda(\mathrm{supp}(g(\alpha, \cdot, \cdot))) > 0$ with positive probability. Then there exists $\varepsilon > 0$ such that with positive probability there is a set $I \subset \mathbb{R}_+$ (depending on α) satisfying $\lambda(I) = \varepsilon$, and such that for all $x \in I$ it holds that $\lambda(I_x) > \varepsilon$, where $I_x := \{x' \in \mathbb{R}_+ : g(\alpha, x, x') = 1\}$. Consider the Poisson point process $(t_j, x_j)_{j\in\mathbb{N}}$ corresponding to the graphon process $\widetilde{G}_t$ with isolated vertices. The number of points $(t_j, x_j) \in [0, t] \times I$ evolves as a function of t as a Poisson process with rate $\varepsilon > 0$; hence the number of such points divided by t converges to ε a.s. For any given pair $(t_j, x_j) \in [0, t] \times I$ the number of points $(\sigma_{i,j}, \chi_{i,j}) \in [0, t] \times I_{x_j}$

for the Poisson point process $(\sigma_{i,j}, \chi_{i,j})_{i\in\mathbb{N}}$ has the law of a Poisson random variable with intensity greater than $t\varepsilon$. Hence,

$$t^{-2}\lim_{t\to\infty}\widetilde{N}_t = t^{-2}\lim_{t\to\infty}\sum_{j,k\,:\,t_j,\sigma_{j,k}\le t} g(\alpha, x_j, \chi_{j,k}) > \varepsilon^2.$$

This contradicts (25), and thus completes our proof that $(\widetilde{G}_t)_{t\ge 0}$ has the law of a graphon process with isolated vertices. ■

Remark 61 *In our proof above we observed that the assumption of exchangeability alone is not sufficient to prove that $(\widetilde{G}_t)_{t\ge 0}$ has the law of a graphon process with isolated vertices. More precisely, without this assumption we might have $W \notin L^1$ and the measure might also consist of the terms containing g, g' and l, l'. We observed in the proof that the terms containing l, l' correspond to isolated edges, and that the terms containing g, g' correspond to "stars" centered at a vertex in the graphon process. It is outside the scope of this paper to do any further analysis of these more general exchangeable graphs.*

Appendix H. Left Convergence of Graphon Processes

In this appendix we will prove Proposition 30. The following lemma will imply part (ii) of the proposition.

Lemma 62 *Let $\mathcal{W}$ be a bounded, non-negative graphon, and assume that $h(F_k, \mathcal{W}) < \infty$ for a star F_k with k leaves. Then $h(F, \mathcal{W}) < \infty$ for all simple, connected graphs F of maximal degree at most k.*

Proof We first note that if $h(F_k, \mathcal{W}) = \int D_W(x)^k \, d\mu(x) < \infty$ for a star with k leaves, then the same holds for all stars with at most k leaves, since we know that $D_W \in L^1(S)$ by our definition of a graphon. Also, using that W is bounded, we assume without loss of generality that F is a tree T of maximal degree $\Delta \le k$.

Designate one of the vertices, r, as the root of the tree, and choose a vertex u_1 such that no other vertex is further from the root. If u_1 has distance 1 from r, then T is a star and there is nothing to prove. Let u be u_1's grandparent, let v be its parent, and let $u_2, \dots, u_s$ for $1 \le s \le \Delta - 1$ be its siblings. Note that by our assumption on u_1, all the siblings $u_1, \dots, u_s$ are leaves. Furthermore, if their grandparent u is the root and the root has no other children, then T is again a star, so we can rule that out as well.

If we remove the edge uv from T, we obtain two disjoint trees T_1 and T_2, and as just argued, the one containing u is a tree with at least 2 vertices and maximal degree at most Δ, while the second one is a star, again of maximal degree at most Δ. Because $h(T, \mathcal{W}) \le \|W\|_\infty h(T_1, \mathcal{W}) h(T_2, \mathcal{W})$, the lemma now follows by induction. ■

Proof of Proposition 30 We will start by proving (i). Fix some simple connected graph F with k vertices. By Proposition 56 applied with F and the simple connected graph on two vertices, respectively,

$$\lim_{t\to\infty} t^{-k}\mathrm{inj}(F, G_t) = \|W\|_1^{k/2} h(F, \mathcal{W}), \qquad 2\lim_{t\to\infty} t^{-2}|E(G_t)| = \|W\|_1,$$

and hence

$$\lim_{t\to\infty} |2E(G_t)|^{-k/2}\mathrm{inj}(F, G_t) = h(F, \mathcal{W}), \tag{26}$$

proving (i).

(ii) Since $h(F, \mathcal{W}) = \int D_w^k$ if F is a star with k leaves, we can use Lemma 62 to conclude that $h(F, \mathcal{W}) < \infty$ for every simple connected graph F with at least two vertices. Express $\hom(F, G_t)$ as $\hom(F, G_t) = \sum_\Phi \mathrm{inj}(F/\Phi, G_t)$, where we sum over all equivalence relations Φ on $V(F)$. By Proposition 56 applied with F/Φ, we have

$$\lim_{t\to\infty} |2E(G_t)|^{-k/2}\mathrm{inj}(F/\Phi, G_t) = 0$$

unless Φ is the equivalence relation for which the number of equivalence classes equals $|V(F)|$. Hence the estimate (26) holds with hom in place of inj, which completes the proof of (ii).

Next we will prove (iii). Let F be a simple connected graph with at least three vertices, and assume $d \in \mathbb{N}$ is such that the degree of the vertices of G_n is bounded by d. We may assume G_n has no isolated vertices, since $h(F, G_n)$ is invariant under adding or deleting isolated vertices. Under the assumption of no isolated vertices, we have $|E(G_n)| \ge |V(G_n)|/2$. By boundedness of degrees, $\hom(F, G_n) \le |V(G_n)| d^{|V(F)|-1}$. Combining these estimates, $h(F, G_n) \le |V(G_n)|^{1-|V(F)|/2} d^{|V(F)|-1}$, from which the desired result follows.

Now we will prove (iv). We first construct an example of a sequence of graphs $(G_n)_{n\in\mathbb{N}}$ which converges for the stretched cut metric $\delta_\square^s$, but which is not left convergent. Let $(\widetilde{G}_n)_{n\in\mathbb{N}}$ be a sequence of simple dense graphs with $|V(\widetilde{G}_n)| \to \infty$ that is convergent in the $\delta_\square$ metric, and hence also in the $\delta_\square^s$ metric.

Define $G_n := \widetilde{G}_n$ for even n, and for odd n let G_n be the union of $\widetilde{G}_n$ and $|E(\widetilde{G}_n)|^{7/8}$ vertices of degree one, which are all connected to the same uniformly random vertex of $\widetilde{G}_n$. Then $(G_n)_{n\in\mathbb{N}}$ converges for $\delta_\square^s$ with the same limit as $(\widetilde{G}_n)_{n\in\mathbb{N}}$, since $\widetilde{G}_n$ is an induced subgraph of G_n and $|E(\widetilde{G}_n)|/|E(G_n)| \to 1$. On the other hand, $(G_n)_{n\in\mathbb{N}}$ is not left convergent, since if F is the simple connected graph with three vertices and two edges, then $\hom(F, G_n) = \Omega(|E(\widetilde{G}_n)|^{14/8})$ for odd n and hence $h(F, G_n) \to \infty$ along sequences of odd n, while $h(F, G_n)$ converges to a finite number by the fact that dense graph sequences which are convergent in the cut metric are left convergent.

Finally we will provide a counterexample in the reverse direction; i.e., we will construct a sequence of graphs $(G_n)_{n\in\mathbb{N}}$ which is left convergent, but does not converge for the stretched cut metric. Let $(\widetilde{G}_n)_{n\in\mathbb{N}}$ be left convergent and satisfy $\lim_{n\to\infty} |E(\widetilde{G}_n)| = \infty$, and let G_n be the union of $\widetilde{G}_n$ and $|E(\widetilde{G}_n)|$ isolated edges. Then $(G_n)_{n\in\mathbb{N}}$ is left convergent, since $\hom(F, G_n) = \hom(F, \widetilde{G}_n) + 2|E(\widetilde{G}_n)|$ when F is the simple connected graph on two vertices, and $\hom(F, G_n) = \hom(F, \widetilde{G}_n)$ when F has at least three vertices. On the other hand, $(G_n)_{n\in\mathbb{N}}$ is not convergent for $\delta_\square^s$, since it does not have uniformly regular tails. ■

References

D. J. Aldous. Representations for partially exchangeable arrays of random variables. *J. Multivariate Anal.*, 11(4):581–598, 1981. doi:10.1016/0047-259X(81)90099-3.

P. J. Bickel and A. Chen. A nonparametric view of network models and Newman-Girvan and other modularities. *Proc. Natl. Acad. Sci. USA*, 106(50):21068–21073, 2009. doi:10.1073/pnas.0907096106.

P. J. Bickel, A. Chen, and E. Levina. The method of moments and degree distributions for network models. *Ann. Statist.*, 39(5):2280–2301, 2011. doi:10.1214/11-AOS904.

P. Billingsley. *Convergence of Probability Measures*. John Wiley & Sons, Inc., New York, second edition, 1999. doi:10.1002/9780470316962.

B. Bollobás, S. Janson, and O. Riordan. The phase transition in inhomogeneous random graphs. *Random Structures Algorithms*, 31(1):3–122, 2007. doi:10.1002/rsa.20168.

B. Bollobás and O. Riordan. Metrics for sparse graphs. In *Surveys in Combinatorics 2009*, volume 365 of *London Math. Soc. Lecture Note Ser.*, pages 211–287. Cambridge Univ. Press, Cambridge, 2009. doi:10.1017/CBO9781107325975.009.

C. Borgs, J. T. Chayes, H. Cohn, and S. Ganguly. Consistent nonparametric estimation for heavy-tailed sparse graphs. Preprint, arXiv:1508.06675, 2015.

C. Borgs, J. T. Chayes, H. Cohn, and N. Holden. In preparation, 2017.

C. Borgs, J. T. Chayes, H. Cohn, and Y. Zhao. An L^p theory of sparse graph convergence I: limits, sparse random graph models, and power law distributions. Preprint, arXiv:1401.2906, 2014a.

C. Borgs, J. T. Chayes, H. Cohn, and Y. Zhao. An L^p theory of sparse graph convergence II: LD convergence, quotients, and right convergence. To appear in Ann. Probab., arXiv:1408.0744, 2014b.

C. Borgs, J. Chayes, J. Kahn, and L. Lovász. Left and right convergence of graphs with bounded degree. *Random Structures Algorithms*, 42(1):1–28, 2013. doi:10.1002/rsa.20414.

C. Borgs, J. Chayes, and L. Lovász. Moments of two-variable functions and the uniqueness of graph limits. *Geom. Funct. Anal.*, 19(6):1597–1619, 2010. doi:10.1007/s00039-010-0044-0.

C. Borgs, J. Chayes, L. Lovász, V. T. Sós, and K. Vesztergombi. Counting graph homomorphisms. In *Topics in discrete mathematics*, volume 26 of *Algorithms Combin.*, pages 315–371. Springer, Berlin, 2006. doi:10.1007/3-540-33700-8_18.

C. Borgs, J. T. Chayes, L. Lovász, V. T. Sós, and K. Vesztergombi. Convergent sequences of dense graphs I: Subgraph frequencies, metric properties and testing. *Adv. Math.*, 219 (6):1801–1851, 2008. doi:10.1016/j.aim.2008.07.008.

F. Caron and E. B. Fox. Sparse graphs using exchangeable random measures. Preprint, arXiv:1401.1137, 2014.

S. Chatterjee. Matrix estimation by universal singular value thresholding. *Ann. Statist.*, 43 (1):177–214, 2015. doi:10.1214/14-AOS1272.

D. S. Choi, P. J. Wolfe, and E. M. Airoldi. Stochastic blockmodels with a growing number of classes. *Biometrika*, 99(2):273–284, 2012. doi:10.1093/biomet/asr053.

E. Çınlar. *Probability and Stochastics*, volume 261 of *Graduate Texts in Mathematics*. Springer, New York, 2011. doi:10.1007/978-0-387-87859-1.

P. Diaconis and S. Janson. Graph limits and exchangeable random graphs. *Rend. Mat. Appl. (7)*, 28(1):33–61, 2008.

A. Frieze and R. Kannan. Quick approximation to matrices and applications. *Combinatorica*, 19(2):175–220, 1999. doi:10.1007/s004930050052.

C. Gao, Y. Lu, and H. H. Zhou. Rate-optimal graphon estimation. *Ann. Statist.*, 43(6): 2624–2652, 2015. doi:10.1214/15-AOS1354.

P. R. Halmos. *Measure Theory*, volume 18 of *Graduate Texts in Mathematics*. Springer, New York, 1974. doi:10.1007/978-1-4684-9440-2.

T. Herlau, M. N. Schmidt, and M. Mørup. Completely random measures for modelling block-structured sparse networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4260–4268. Curran Associates, Inc., 2016.

P. D. Hoff, A. E. Raftery, and M. S. Handcock. Latent space approaches to social network analysis. *J. Amer. Statist. Assoc.*, 97(460):1090–1098, 2002. doi:10.1198/016214502388618906.

D. Hoover. Relations on probability spaces and arrays of random variables. Preprint, Institute for Advanced Study, Princeton, NJ, 1979.

S. Janson. *Graphons, Cut Norm and Distance, Couplings and Rearrangements*, volume 4 of *New York Journal of Mathematics Monographs*. State University of New York, University at Albany, Albany, NY, 2013.

S. Janson. Graphons and cut metric on σ-finite measure spaces. Preprint, arXiv:1608.01833, 2016.

O. Kallenberg. *Foundations of Modern Probability*. Springer, New York, second edition, 2002.

O. Kallenberg. *Probabilistic Symmetries and Invariance Principles*. Springer, New York, 2005. doi:10.1007/0-387-28861-9.

O. Klopp, A. B. Tsybakov, and N. Verzelen. Oracle inequalities for network models and sparse graphon estimation. *Ann. Statist.*, 45(1):316–354, 2017. doi:10.1214/16-AOS1454.

L. Lovász and B. Szegedy. Limits of dense graph sequences. *J. Combin. Theory Ser. B*, 96 (6):933–957, 2006. doi:10.1016/j.jctb.2006.05.002.

L. Lovász and B. Szegedy. Szemerédi's lemma for the analyst. *Geom. Funct. Anal.*, 17(1): 252–270, 2007. doi:10.1007/s00039-007-0599-6.

K. Rohe, S. Chatterjee, and B. Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *Ann. Statist.*, 39(4):1878–1915, 2011. doi:10.1214/11-AOS887.

D. W. Stroock. *Probability Theory: An Analytic View*. Cambridge University Press, Cambridge, second edition, 2011a.

D. W. Stroock. *Essentials of Integration Theory for Analysis*, volume 262 of *Graduate Texts in Mathematics*. Springer, New York, 2011b. doi:10.1007/978-1-4614-1135-2.

V. Veitch and D. M. Roy. The class of random graphs arising from exchangeable random measures. Preprint, arXiv:1512.03099, 2015.

P. J. Wolfe and S. C. Olhede. Nonparametric graphon estimation. Preprint, arXiv:1309.5936, 2013.

Journal of Machine Learning Research 18 (2018) 1-43 Submitted 1/17; Revised 1/17; Published 4/18

Weighted SGD for ℓ_p Regression with Randomized Preconditioning *

Jiyan Yang JIYAN@STANFORD.EDU
Institute for Computational and Mathematical Engineering
Stanford University
Stanford, CA 94305, USA

Yin-Lam Chow YCHOW@STANFORD.EDU
Institute for Computational and Mathematical Engineering
Stanford University
Stanford, CA 94305, USA

Christopher Ré CHRISMRE@CS.STANFORD.EDU
Department of Computer Science
Stanford University
Stanford, CA 94305, USA

Michael W. Mahoney MMAHONEY@STAT.BERKELEY.EDU
International Computer Science Institute and Department of Statistics
University of California, Berkeley
Berkeley, CA 94720, USA

Editor: Zhihua Zhang

Abstract

In recent years, stochastic gradient descent (SGD) methods and randomized linear algebra (RLA) algorithms have been applied to many large-scale problems in machine learning and data analysis. SGD methods are easy to implement and applicable to a wide range of convex optimization problems. In contrast, RLA algorithms provide much stronger performance guarantees but are applicable to a narrower class of problems. We aim to bridge the gap between these two methods in solving *constrained* overdetermined linear regression problems—e.g., ℓ_2 and ℓ_1 regression problems.

- We propose a hybrid algorithm named pwSGD that uses RLA techniques for preconditioning and constructing an importance sampling distribution, and then performs an SGD-like iterative process with weighted sampling on the preconditioned system.
- By rewriting a deterministic ℓ_p regression problem as a stochastic optimization problem, we connect pwSGD to several existing ℓ_p solvers including RLA methods with algorithmic leveraging (RLA for short).
- We prove that pwSGD inherits faster convergence rates that only depend on the lower dimension of the linear system, while maintaining low computation complexity. Such SGD convergence rates are superior to other related SGD algorithm such as the weighted randomized Kaczmarz algorithm.
- Particularly, when solving ℓ_1 regression with size n by d, pwSGD returns an approximate solution with ϵ relative error in the objective value in $\mathcal{O}(\log n \cdot \mathrm{nnz}(A) + \mathrm{poly}(d)/\epsilon^2)$ time. This complexity is *uniformly* better than that of RLA methods in terms of both ϵ and d when the problem is unconstrained. In the presence of constraints, pwSGD only has to solve a sequence of much simpler and smaller optimization problem over the same constraints. In general this is more efficient than solving the constrained subproblem required in RLA.
- For ℓ_2 regression, pwSGD returns an approximate solution with ϵ relative error in the objective value and the solution vector measured in prediction norm in $\mathcal{O}(\log n \cdot \mathrm{nnz}(A) + \mathrm{poly}(d)\log(1/\epsilon)/\epsilon)$ time. We show that for unconstrained ℓ_2 regression, this complexity is comparable to that of RLA and is asymptotically better over several state-of-the-art solvers in the regime where the desired accuracy ϵ, high dimension n and low dimension d satisfy $d \geq 1/\epsilon$ and $n \geq d^2/\epsilon$.

We also provide lower bounds on the coreset complexity for more general regression problems, indicating that still new ideas will be needed to extend similar RLA preconditioning ideas to weighted SGD algorithms for more general regression problems. Finally, the effectiveness of such algorithms is illustrated numerically on both synthetic and real datasets, and the results are consistent with our theoretical findings and demonstrate that pwSGD converges to a medium-precision solution, e.g., $\epsilon = 10^{-3}$, more quickly.

1. Introduction

Many novel algorithms for large-scale data analysis and machine learning problems have emerged in recent years, among which stochastic gradient descent (SGD) methods and randomized linear algebra (RLA) algorithms have received much attention—both for their strong performance in practical applications and for their interesting theoretical properties (Bottou, 2010; Mahoney, 2011). Here, we consider the ubiquitous ℓ_1 and ℓ_2 regression problems, and we describe a novel RLA-SGD algorithm called pwSGD (preconditioned weighted SGD). Our new algorithm combines the advantages of both RLA and SGD methods for solving constrained overdetermined ℓ_1 and ℓ_2 regression problems.

Consider the overdetermined ℓ_p regression problem

$$\min_{x \in \mathcal{Z}} f(x) = \|Ax - b\|_p, \tag{1}$$

where $p \in [1, \infty]$, $A \in \mathbb{R}^{n \times d}$, $b \in \mathbb{R}^n$ and $n \gg d$. When $\mathcal{Z} = \mathbb{R}^d$, i.e., the solution space is unconstrained, the cases $p \in \{1, 2\}$ are respectively known as the Least Absolute Deviations (LAD, or ℓ_1) and Least-squares (LS, or ℓ_2) regression problems. Classically, the unconstrained ℓ_2 regression problem can be solved by eigenvector-based methods with worst-case running time $\mathcal{O}(nd^2)$ (Golub and Van Loan, 1996); or by iterative methods for which the running time depends on the condition number of A (Barrett et al., 1994; Kelley, 1995; Saad, 2003), while the unconstrained ℓ_1 regression problem can be formulated as a linear program (Portnoy and Koenker, 1997; Chen et al., 2001) and solved by an interior-point method (Portnoy and Koenker, 1997; Portnoy, 1997).

For these and other regression problems, SGD algorithms are widely used in practice because of their scalability and efficiency. In contrast, RLA algorithms have better theoretical guarantees but (thus far) have been less flexible, e.g., in the presence of constraints. For example, they may use an interior point method for solving a constrained subproblem, and this may be less efficient than SGD. (Without constraints, RLA methods can be used to construct subproblems to be solved exactly, or they can be used to construct preconditioners for the original problem; see Yang et al. (2016b) for details and implementations of these RLA methods to compute low, medium, and high precision solutions on up to terabyte-sized input data.) In this paper, we combine these two algorithmic approaches to develop a method that takes advantage of the strengths of both of these approaches.

*. A conference version of this paper appears under the same title in *Proceedings of ACM-SIAM Symposium on Discrete Algorithms*, Arlington, VA, 2016 (Yang et al., 2016a).

1.1 Overview of our main algorithm

Our main algorithm pwSGD is a hybrid method for solving constrained overdetermined ℓ_1 and ℓ_2 regression problems. It consists of two main steps. First, apply RLA techniques for preconditioning and construct an importance sampling distribution. Second, apply an SGD-like iterative phase with weighted sampling on the preconditioned system. Such an algorithm preserves the simplicity of SGD and the high quality theoretical guarantees of RLA. In particular, we prove that after preconditioning, the number of iterations required to converge to a target accuracy is fully predictable and only depends on the low dimension d, i.e., it is independent of the high dimension n. We show that, with a proper choice of preconditioner, pwSGD runs in $\mathcal{O}(\log n \cdot \mathrm{nnz}(A) + \mathrm{poly}(d)/\epsilon^2)$ time to return an approximate solution with ϵ relative error in the objective for constrained ℓ_1 regression; and in $\mathcal{O}(\log n \cdot \mathrm{nnz}(A) + \mathrm{poly}(d)\log(1/\epsilon)/\epsilon)$ time to return an approximate solution with ϵ relative error in the solution vector in prediction norm for constrained ℓ_2 regression. Furthermore, for unconstrained ℓ_2 regression, pwSGD runs in $\mathcal{O}(\log n \cdot \mathrm{nnz}(A) + d^3\log(1/\epsilon)/\epsilon)$ time to return an approximate solution with ϵ relative error in the objective.

To provide a quick overview of how pwSGD compares to existing algorithms, in Tables 1 and 2, we summarize the complexity required to compute a solution $\hat{x}$ with relative error, i.e., $(f(\hat{x}) - f(x^*))/f(x^*) = \epsilon$, of several solvers for unconstrained ℓ_1 and ℓ_2 regression. In Table 1, RLA with algorithmic leveraging (RLA for short) (Clarkson et al., 2013; Yang et al., 2014) is a popular method for obtaining a low-precision solution and randomized IPCPM is an iterative method for finding a higher-precision solution (Meng and Mahoney, 2013b) for unconstrained ℓ_1 regression. Clearly, pwSGD has a uniformly better complexity than that of RLA methods in terms of both d and ϵ, no matter which underlying preconditioning method is used. This makes pwSGD a more suitable candidate for getting a medium-precision, e.g., $\epsilon = 10^{-3}$, solution.

In Table 2, all the methods require constructing a sketch first. Among them, "low-precision" solvers refer to "sketching + direct solver" type algorithms; see Drineas et al. (2011); Clarkson and Woodruff (2013) for projection-based examples and Clarkson and Woodruff (2013); Drineas et al. (2012) for sampling-based examples. "High-precision" solvers refer to "sketching + preconditioning + iterative solver" type algorithms; see Avron et al. (2010); Meng et al. (2014) for examples. One can show that, when $d \geq 1/\epsilon$ and $n \geq d^2/\epsilon$, pwSGD is asymptotically better than all the solvers shown in Table 2. Moreover, although high-precision solvers are more efficient when a high-precision solution is desired, usually they are designed for unconstrained problems, whereas pwSGD also works for constrained problems.

We remark that, compared to general SGD algorithms, our RLA-SGD hybrid algorithm pwSGD *works for problems in a narrower range, i.e., ℓ_p regression, but inherits the strong theoretical guarantees of RLA. When solving ℓ_2 regression, for which traditional RLA methods are well designed,* pwSGD *has a comparable complexity. On the other hand, when solving ℓ_1 regression, due to the efficiency of SGD update,* pwSGD *has a strong advantage over traditional RLA methods.* See Sections 4.3 and 4.4 for more detailed discussions.

Finally, in Section 5, empirically we show that pwSGD performs favorably compared to other competing methods, as it converges to a medium-precision solution more quickly.

solver	complexity (general)	complexity (sparse)
RLA with algorithmic leveraging	$time(R) + \mathcal{O}(\mathrm{nnz}(A)\log n + \bar{\kappa}_1^{\frac{5}{4}} d^{\frac{17}{4}}/\epsilon^{\frac{5}{2}})$	$\mathcal{O}(\mathrm{nnz}(A)\log n + d^{\frac{69}{8}}\log^{\frac{25}{8}} d/\epsilon^{\frac{5}{2}})$
randomized IPCPM	$time(R) + nd^2 + \mathcal{O}((nd + \mathrm{poly}(d))\log(\bar{\kappa}_1 d/\epsilon))$	$nd^2 + \mathcal{O}((nd + \mathrm{poly}(d))\log(d/\epsilon))$
pwSGD	$time(R) + \mathcal{O}(\mathrm{nnz}(A)\log n + d^3\bar{\kappa}_1/\epsilon^2)$	$\mathcal{O}(\mathrm{nnz}(A)\log n + d^{\frac{13}{2}}\log^{\frac{5}{2}} d/\epsilon^2)$

Table 1: Summary of complexity of several unconstrained ℓ_1 solvers that use randomized linear algebra. The target is to find a solution $\hat{x}$ with accuracy $(f(\hat{x}) - f(x^*))/f(x^*) \leq \epsilon$, where $f(x) = \|Ax - b\|_1$. In the above, $time(R)$ denotes the time needed to compute a matrix R such that AR^{-1} is well-conditioned with condition number $\bar{\kappa}_1$ (Definition 1). The general complexity bound and the one using sparse reciprocal exponential transform (Woodruff and Zhang, 2013) as the underlying sketching method are presented. Here, we assume $n \gg d$ such that $n > d^3\log d$ and the underlying ℓ_1 regression solver in RLA with algorithmic leveraging algorithm takes $\mathcal{O}(n^{\frac{5}{4}}d^3)$ time to return a solution (Portnoy and Koenker, 1997). The complexity of each algorithm is computed by setting the failure probability to be a constant.

solver	complexity (SRHT)	complexity (CW)
low-precision solvers (projection)	$\mathcal{O}\left(nd\log(d/\epsilon) + d^3\log n(\log d + 1/\epsilon)\right)$	$\mathcal{O}\left(\mathrm{nnz}(A) + d^4/\epsilon^2\right)$
low-precision solvers (sampling)	$\mathcal{O}\left(nd\log n + d^3\log n\log d + d^3\log d/\epsilon\right)$	$\mathcal{O}\left(\mathrm{nnz}(A)\log n + d^4 + d^3\log d/\epsilon\right)$
high-precision solvers	$\mathcal{O}\left(nd\log n + d^3\log n\log d + nd\log(1/\epsilon)\right)$	$\mathcal{O}\left(\mathrm{nnz}(A) + d^4 + nd\log(1/\epsilon)\right)$
pwSGD	$\mathcal{O}\left(nd\log n + d^3\log n\log d + d^3\log(1/\epsilon)/\epsilon\right)$	$\mathcal{O}\left(\mathrm{nnz}(A)\log n + d^4 + d^3\log(1/\epsilon)/\epsilon\right)$

Table 2: Summary of complexity of several unconstrained ℓ_2 solvers that use randomized linear algebra. The target is to find a solution $\hat{x}$ with accuracy $(f(\hat{x}) - f(x^*))/f(x^*) \leq \epsilon$, where $f(x) = \|Ax - b\|_2$. Two sketching methods, namely, SRHT (Drineas et al., 2011; Tropp, 2011) and CW (Clarkson and Woodruff, 2013) are considered. Here, we assume $d \leq n \leq e^d$. The complexity of each algorithm is computed by setting the failure probability to be a constant.

1.2 Connection to related algorithms

As a side point of potentially independent interest, a connection between ℓ_p regression and stochastic optimization will allow us to unify our main algorithm pwSGD and some existing ℓ_p regression solvers under the same framework. In Figure 1, we present the basic structure of this framework, which provides a view of pwSGD from another perspective. To be more specific, we (in Proposition 4 formally) reformulate the deterministic ℓ_p regression problem in (1) as a stochastic optimization problem, i.e.,

$$\min_{y\in\mathcal{Y}} \|Uy - b\|_p^p = \min_{y\in\mathcal{Y}} \mathbb{E}_{\xi\sim P}\left[|U_\xi y - b_\xi|^p/p_\xi\right],$$

where U is a basis for the range space of A and ξ is a random variable over $\{1,\ldots,n\}$ with distribution $P = \{p_i\}_{i=1}^n$. As suggested in Figure 1, to solve this stochastic optimization problem, typically one needs to answer the following three questions.

- *(C1): How to sample: SAA (Sampling Average Approximation, i.e., draw samples in a batch mode and deal with the subproblem) or SA (Stochastic Approximation, i.e., draw a mini-batch of samples in an online fashion and update the weight after extracting useful information)?*
- *(C2): Which probability distribution P (uniform distribution or not) and which basis U (preconditioning or not) to use?*

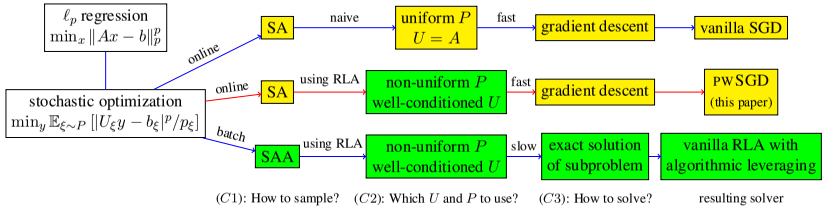


Figure 1: An overview of our framework for solving ℓ_p regression via stochastic optimization. To construct a solver, three choices have to be made. For ($C1$), the answer can be either SAA (Sampling Average Approximation, i.e., sample a batch of points and deal with the subproblem) or SA (Stochastic Approximation, i.e., sample a mini-batch in an online fashion and update the weight vector after extracting useful information). In ($C2$), the answer is determined by P, which denotes the underlying sampling distribution (uniform or nonuniform) and U, which denotes the basis with which to work (original or preconditioned system). Finally, for ($C3$), the answer determines how we solve the subproblem (in SAA) or what information we extract and how we update the weight (in SA).

- *($C3$): Which solver to use (e.g., how to solve the subproblem in SAA or how to update the weight in SA)?*

Some combinations of these choices may lead to existing solvers; see Figure 1 and Section 3 for more details. A natural question arises: is there a combination of these choices that leverages the algorithmic benefits of RLA preconditioning to improve the performance of SGD-type algorithms? Recall that RLA methods (in particular, those that exploit algorithmic averaging; see Appendix B and also Drineas et al. (2012); Yang et al. (2016b)) inherit strong theoretical guarantees because the underlying sampling distribution P captures most of the important information of the original system; moreover, such a carefully constructed leverage-based distribution is defined based on a well-conditioned basis U, e.g., an orthogonal matrix for $p = 2$. One immediate idea is to develop an SGD-like algorithm that uses the same choice of U and P as in RLA methods. This simple idea leads to our main algorithm pwSGD, which is an online algorithm ($C1$) that uses a non-uniform sampling distribution ($C2$) and performs a gradient descent update ($C3$) on a preconditioned system ($C2$), as Figure 1 suggests.

Indeed, for least-squares problems (unconstrained ℓ_2 regression), pwSGD is highly related to the weighted randomized Kaczmarz (RK) algorithm (Strohmer and Vershynin, 2009; Needell et al., 2014) in the way that both algorithms are SGD algorithm with non-uniform P but pwSGD runs on a well-conditioned basis U while randomized RK doesn't involve preconditioning. In Section 4.5 we show that this preconditioning step dramatically reduces the number of iterations required for pwSGD to converge to a (fixed) desired accuracy.

1.3 Main contributions

Now we are ready to state our main contributions.

- We reformulate the deterministic ℓ_p regression problem (1) into a stochastic optimization problem (4) and make connections to existing solvers including RLA methods with algorithmic leveraging and weighted randomized Kaczmarz algorithm (Sections 3 and 4.5).
- We develop a hybrid algorithm for solving *constrained* overdetermined ℓ_1 and ℓ_2 regression called pwSGD, which is an SGD algorithm with preconditioning and a non-uniform sampling distribution constructed using RLA techniques. We present several choices of the preconditioner and their tradeoffs. We show that with a suitable preconditioner, convergence rate of the SGD phase only depends on the low dimension d, and is independent of the high dimension n (Sections 4.1 and 4.2).
- We prove that pwSGD returns an approximate solution with ϵ relative error in the objective value in $\mathcal{O}(\log n \cdot \mathrm{nnz}(A) + \mathrm{poly}(d)/\epsilon^2)$ time for ℓ_1 regression. This complexity is *uniformly* better than that of RLA methods in terms of both ϵ and d when the problem is unconstrained. In the presence of constraints, pwSGD only has to solve a sequence of much simpler and smaller optimization problems over the same constraints, which in general can be more efficient than solving the constrained subproblem required in RLA (Sections 4.3 and 4.4).
- We prove that pwSGD returns a solution with ϵ relative error in the objective value and the solution vector measured in prediction norm in $\mathcal{O}(\log n \cdot \mathrm{nnz}(A) + \mathrm{poly}(d)\log(1/\epsilon)/\epsilon)$ time for ℓ_2 regression. We show that for unconstrained ℓ_2 regression, this complexity is asymptotically better than several state-of-the-art solvers in the regime where $d \geq 1/\epsilon$ and $n \geq d^2/\epsilon$ (Sections 4.3 and 4.4).
- Empirically, we show that when solving ℓ_1 and ℓ_2 regression problems, pwSGD inherits faster convergence rates and performs favorably in the sense that it obtains a medium-precision much faster than other competing SGD-like solvers do. Also, theories regarding several choices of preconditioners are numerically verified (Section 5).
- We show connections between RLA algorithms and coreset methods of empirical optimization problems under the framework of Feldman and Langberg (2011). We show that they are equivalent for ℓ_p regression and provide lower bounds on the coreset complexity for some more general regression problems. We also discuss the difficulties in extending similarly RLA preconditioning ideas to general SGD algorithms (Section 6).

1.4 Other prior related work

Numerous RLA algorithms have been proposed to solve ℓ_p regression problems (Yang et al., 2016b). RLA theories show that to achieve a relative-error bound, the required sampling size only depends on d, independent of n, and the running time also depends on the time to implement the random projection at the first step. Regarding the performance of unconstrained regression problems, in Dasgupta et al. (2009) the authors provide an algorithm that constructs a well-conditioned basis by ellipsoid rounding and a subspace-preserving sampling matrix for ℓ_p regression problems in $\mathcal{O}(nd^5 \log n)$ time; a sampling algorithm based on Lewis weights for ℓ_p regression have been proposed by Cohen and Peng (2015); the algorithms in Sohler and Woodruff (2011) and Clarkson et al. (2013) use the "slow" and "fast" Cauchy Transform to compute the low-distortion ℓ_1 embedding matrix and solve the over-constrained ℓ_1 regression problem in $\mathcal{O}(nd^{1.376+})$ and $\mathcal{O}(nd \log n)$ time, respectively; the algorithms in Drineas et al. (2012) estimate the leverage scores up to a small factor and solve the ℓ_2 regression problem in $\mathcal{O}(nd \log n)$ time respectively; and the algorithms in Clarkson and Woodruff (2013); Meng and Mahoney (2013a); Nelson and Nguyen (2013), solve the problem via sparse random projections in nearly input-sparsity time, i.e., $\mathcal{O}(\log n \cdot \mathrm{nnz}(A))$

time, plus lower-order terms, and a tighter analysis is provided by Cohen (2016). As for iterative algorithms, the algorithms in Avron et al. (2010); Meng et al. (2014) use randomized linear algebra to compute a preconditioner and call iterative solvers such as LSQR to solve the preconditioned problem.

In contrast, SGD algorithms update the solution vector in an iterative fashion and are simple to implement and scalable to large datasets (Bottou and Le Cun, 2004; Shalev-Shwartz and Srebro, 2008; Bottou and Bousquet, 2008). Moreover, these methods can be easily extended for problems with general convex loss functions and constraints, such as Pegasos (Shalev-Shwartz et al., 2007) for regularized SVM and stochastic coordinate descent (SCD) for ℓ_1 regularization (Shalev-Shwartz and Tewari, 2009). Several techniques, such as SAGE (Hu et al., 2009), AdaGrad (Duchi et al., 2011), SVRG (Johnson and Zhang, 2013), have recently been proposed to accelerate the convergence rate of SGD, and Niu et al. (2011) also show that SGD is favorable for parallel/distributed computation. More recently, several works, e.g., Zhao and Zhang (2015); Needell et al. (2014) regarding SGD with weighted sampling are proposed, in which the authors show that the performance of SGD can be improved by using a nonuniform sampling distribution.

In addition, as we point out in Section 4.2, pwSGD has a close relationship to second-order methods. It can be viewed as an algorithm with approximate Hessians obtained by sketching and stochastic gradients. This is related to the iterative Hessian sketching algorithm for solving constrained least squares problems proposed by Pilanci and Wainwright (2014) which is essentially a Newton-type algorithm with iterative sketched Hessians and batch gradients. Moreover, the idea of using approximate Hessians and stochastic gradients have been discussed in several recent papers. For example, (Moritz et al., 2016; Byrd et al., 2016; Curtis, 2016) exploit the idea of approximating Hessian with L-BFGS type updates and (variance-reduced) stochastic updates.

2. Preliminaries

For any matrix $A \in \mathbb{R}^{n\times d}$, we use A_i and A^j to denote the i-th row and j-th column of A, respectively. We assume A has full rank, i.e., $\text{rank}(A) = d$. Also denote by $\kappa(A)$ the usual condition number of A, by $\text{nnz}(A)$ the number of nonzero elements in A, and by $\text{poly}(d)$ a low-degree polynomial in d. We also use $[n]$ to denote the set of indices $1, \ldots, n$.

Throughout this subsection, the definitions are applied to general $p \in [1, \infty)$. We denote by $|\cdot|_p$ the element-wise ℓ_p norm of a matrix: $|A|_p = \left(\sum_{i=1}^n \sum_{j=1}^d |A_{ij}|^p\right)^{1/p}$. In particular, when $p = 2$, $|\cdot|_2$ is equivalent to the Frobenius norm.

The following two notions on well-conditioned bases and leverage scores are crucial to our methods. The first notion is originally introduced by Clarkson (2005) and stated more precisely in Dasgupta et al. (2009), and it is used to justify the well-posedness of a ℓ_p regression problem. These notions were introduced by Dasgupta et al. (2009).

Definition 1 ((α, β, p)-conditioning and well-conditioned basis) *A matrix $A \in \mathbb{R}^{n\times d}$ is (α, β, p)-conditioned if $|A|_p \le \alpha$ and for all $x \in \mathbb{R}^d$, $\beta\|Ax\|_p \ge \|x\|_q$, where $1/p + 1/q = 1$. Define $\bar{\kappa}_p(A)$ as the minimum value of $\alpha\beta$ such that A is (α, β, p)-conditioned. We say that a basis U for* range(A) *is a well-conditioned basis if $\bar{\kappa}_p = \bar{\kappa}_p(U)$ is a low-degree polynomial in d, independent of n.*

The notion of leverage scores captures how important each row in the dataset is, and is used in the construction of the sampling probability.

Definition 2 (ℓ_p leverage scores) *Given $A \in \mathbb{R}^{n\times d}$, suppose U is an (α, β, p) well-conditioned basis for* range(A)*. Then the i-th leverage score λ_i of A is defined as $\lambda_i = \|U_i\|_p^p$ for $i = 1, \ldots, n$.*

2.1 Preconditioning

Here, we briefly review the preconditioning methods that will be used in our main algorithms. A detailed summary of various preconditioning methods can be found in Yang et al. (2014, 2016b). The procedure for computing a preconditioner can be summarized in the following two steps.

- Given a matrix $A \in \mathbb{R}^{n\times d}$ with full rank, we first construct a sketch $SA \in \mathbb{R}^{s\times d}$ for A satisfying
$$\sigma_S \cdot \|Ax\|_p \le \|SAx\|_p \le \kappa_S\sigma_S \cdot \|Ax\|_p, \quad \forall x \in \mathbb{R}^d, \tag{2}$$
where κ_S is the distortion factor independent of n.
- Next, we compute the QR factorization of SA whose size only depends on d. Return R^{-1}.

The following lemma guarantees that the preconditioner satisfies that AR^{-1} is well-conditioned since κ_S and s depend on d only, independent of n.

Lemma 3 *Let R be the matrix returned by the above preconditioning procedure, then we have*
$$\bar{\kappa}_p(AR^{-1}) \le \kappa_S d^{\max\{\frac{1}{2}, \frac{1}{p}\}} s^{|\frac{1}{p} - \frac{1}{2}|}. \tag{3}$$

Various ways of computing a sketching matrix S satisfying (2) are proposed recently. It is worth mentioning that sketching algorithms that run in nearly input-sparsity time, i.e., in time proportional to $\mathcal{O}(\text{nnz}(A))$ to obtain such a sketch matrix for $p = 1$ and $p = 2$ are available via random projections composed of sparse matrices; see Clarkson and Woodruff (2013); Meng and Mahoney (2013a); Woodruff and Zhang (2013); Nelson and Nguyen (2013) for details. In Tables 5 and 6 in Appendix A we provide a short summary of these sketching methods and the resulting running time and condition number.

3. A connection to stochastic optimization

In this section, we describe our framework for viewing deterministic ℓ_p regression problems from the perspective of stochastic optimization. This framework will recover both RLA and SGD methods in a natural manner; and by combining these two approaches in a particular way we will obtain our main algorithm.

We reformulate overdetermined ℓ_p regression problems of the form (1) into a stochastic optimization problem of the form (4) [1], which reformulates a deterministic regression problem into a stochastic optimization problem. Note that the result holds for general $p \in [1, \infty)$.

Proposition 4 *Let $U \in \mathbb{R}^{n\times d}$ be a basis of the range space of A in the form $U = AF$, where $F \in \mathbb{R}^{d\times d}$. The constrained overdetermined ℓ_p regression problem* (1) *is equivalent to*
$$\min_{y\in\mathcal{Y}} \|Uy - b\|_p^p = \min_{y\in\mathcal{Y}} \mathbb{E}_{\xi\sim P}\left[H(y, \xi)\right], \tag{4}$$

1. Technically, this result is straightforward; but this reformulation allows us to introduce randomness—parameterized by a probability distribution P—into the deterministic problem (1) in order to develop randomized algorithms for it.

where ξ is a random variable over $\{1,\dots,n\}$ with distribution $P=\{p_i\}_{i=1}^n$, y is the decision variable in $\mathcal{Y}$, and $H(y,\xi)=|U_\xi y-b_\xi|^p/p_\xi$. The constraint set of y is $\mathcal{Y}=\{y\in\mathbb{R}^d|y=F^{-1}x, x\in\mathcal{Z}\}$.

With Proposition 4, as suggested in Figure 1, one can solve the overdetermined ℓ_p regression problem (1) by applying either SAA or SA, i.e., $(C1)$ on the stochastic optimization problem (4). In addition to the choice of SA versus SAA, one also has to choose U and P, i.e., $(C2)$, and determine the underlying solver, i.e., $(C3)$.

Assume that if SAA is used, then for $(C3)$ we solve the subproblem exactly, i.e., we compute a high-precision solution of the subproblem; this leads to a class of *randomized linear algebra* (RLA) algorithms for solving ℓ_p regression. Alternatively, if we assume that SA is used, then we extract the first-order information, i.e., sub-gradient of the sample, and update the weight in a gradient descent fashion; this leads to a family of *stochastic gradient descent* (SGD) algorithms for solving ℓ_p regression.

For $(C2)$, we need to choose a basis U that converts (1) into an equivalent problem represented by U and choose a distribution P for which the algorithm samples a row at every iteration accordingly. In general, different choices of U and P lead to different algorithms. In the following two subsections, we will discuss their effects on SAA and SA and make connections between existing solvers and our new solution methods. For simplicity, we assume there are no constraints, i.e., $\mathcal{Z}=\mathbb{R}^d$ (although much of this framework generalizes to nontrivial constraints).

3.1 Using RLA (SAA) to solve ℓ_p regression

In this subsection, we briefly discuss the algorithms induced by applying SAA to (4) with different choices of basis U and distribution P in Proposition 4.

We first show that the choice of the basis U has no effect on the resulting sampling algorithm. Let $S\in\mathbb{R}^{s\times n}$ be the equivalent sampling matrix in the sampling algorithm. That is,

$$S_{ij}=\begin{cases}1/p_j & \text{if the } j\text{-th row is sampled in the } i\text{-th iteration}\\ 0 & \text{otherwise.}\end{cases}$$

Then the subproblem can be cast as $\min_{y\in\mathcal{Y}}\|SUy-b\|_p^p$, which is equivalent to $\min_{x\in\mathcal{Z}}\|SAx-b\|_p^p$. Therefore, with a given distribution P, applying SAA to the stochastic optimization problem associated with any basis U is equivalent to applying SAA to the original problem with matrix A.

Next, we discuss the effect of the choice of P, i.e., the sampling distribution in SAA, on the required sampling size.

Naive choice of P One choice of P is a uniform distribution, i.e., $p_i=1/n$ for $i=1,\dots,n$. The resulting SAA algorithm becomes uniformly sampling s rows from the original n rows and solving the subproblem induced by the selected rows. If all the rows are equally "important", such an algorithm can be expected to work. However, consider the following toy example for which uniform sampling gives undesirable answers with high probability. Suppose the first row of the matrix contains the only nonzero element in the first column of the design matrix A. Since the only measurement of x_1 lies in the first row, in order to recover the optimal value, namely x_1^*, the first row in matrix A is crucial. However, when a uniform sampling scheme is used, the sampling size required in order to sample the first row is $\Omega(n)$. This implies that RLA with uniform sampling will fail with high probability unless the sampling size $s=\Omega(n)$.

Smarter choice of P In the above example, it is not hard to show that the leverage score of the first row is 1, i.e., it is much larger than the average value of the leverage scores. This inspires us to put more weights on "important" rows, i.e., rows with higher leverage scores. An immediate solution is to define P based on the leverage scores as follows:

$$p_i=\frac{\lambda_i}{\sum_{j=1}^n\lambda_j},$$

where λ_i is the i-th leverage score of A (which depends on whether one is working with ℓ_1, ℓ_2, or more general ℓ_p regression). Applying SAA with this distribution and solving the subproblem exactly recovers the recently proposed RLA methods with algorithmic leveraging for solving overdetermined ℓ_p regression problems; see Mahoney (2011); Dasgupta et al. (2009); Clarkson et al. (2013); Yang et al. (2014); Clarkson and Woodruff (2013); Meng and Mahoney (2013a); Ma et al. (2014) for details. (In RLA, this is simply solving the subproblem of the original problem, but in statistical learning theory, this has the interpretation of Empirirical Risk Minimization.) This algorithm is formally stated in Algorithm 3 in Appendix B. We also include its approximation-of-quality results from (Dasgupta et al., 2009) in Appendix B, which state that the resulting approximate solution $\hat{x}$ produces a $(1+\epsilon)$-approximation to the objective if the sampling size s is large enough. (Note, in particular, that "large enough" here means that when the desired accuracy and failure probability are fixed, the required sampling size only depends on the lower dimension d, independent of n.)

3.2 Using SGD (SA) to solve ℓ_p regression

Applying SA to (4) and updating the weight vector using first-order information results in a SGD algorithm. It is not hard to show that, given $U=AF$ and $P=\{p_i\}_{i=1}^n$, the update rule is as follows. Suppose the ξ_t-th row is sampled; then the weight vector x_t is updated by

$$x_{t+1}=x_t-\eta c_t H^{-1}A_{\xi_t},$$

where $H=\left(FF^\top\right)^{-1}\in\mathbb{R}^{d\times d}$, η is the step size, and c_t is a constant that depends on x_t and ξ_t.

Next, we discuss how different choices of U and P affect the convergence rates of the resulting SGD algorithms. For simplicity, we restrict our discussions to unconstrained ℓ_1 regressions.

Naive choice of U and P Consider the following choices of U and P that lead to undesirable convergence rates. Let $U=A$. If we apply the SGD with some distribution $P=\{p_i\}_{i=1}^n$, some simple arguments in the SGD convergence rate analysis lead to a relative approximation error of

$$\frac{f(\hat{x})-f(x^*)}{f(\hat{x})}=\mathcal{O}\left(\frac{\|x^*\|_2\max_{1\le i\le n}\|A_i\|_1/p_i}{\|Ax^*-b\|_1}\right), \tag{5}$$

where $f(x)=\|Ax-b\|_1$ and x^* is the optimal solution. When $\{p_i\}_{i=1}^n$ is the uniform distribution, (5) becomes $\mathcal{O}\left(n\frac{\|x^*\|_2\cdot M}{\|Ax^*-b\|_1}\right)$, where $M=\max_{1\le i\le n}\|A_i\|_1$ is the maximum ℓ_1 row norm of A. Alternatively, if one chooses p_i to be proportional to the row norms of A, i.e., $p_i=\frac{\|A_i\|_1}{\|A\|_1}$, then (5) becomes $\mathcal{O}\left(\frac{\|x^*\|_2\cdot\|A\|_1}{\|Ax^*-b\|_1}\right)$. Consider the following scenario. Given A and b, we continue to append samples (z,c) satisfying $z^\top x^*=c$ and $\|z\|_2\le M$ to A and b, respectively. This process will keep

x^*, M and $\|Ax^* - b\|_1$ unchanged. However, the value of n and $\|A\|_1$ will increase. Thus, in this case, the expected time for convergence of SGD with these naive sampling distributions might blow up as the size of the matrix grows.

Smarter choice of U and P To avoid this problem, we need to precondition the linear regression problem. If we work with a well-conditioned basis U for the range space of A and choose the sampling probabilities proportional to the row norms of U, i.e., leverage scores of A, then the resulting convergence rate on the relative error of the objective becomes $\mathcal{O}\left(\frac{\|y^*\|_2 \cdot \|U\|_1}{\|Uy^* - b\|_1}\right)$, where y^* is an optimal solution to the transformed problem. By Definition 1, if U is a well-conditioned basis, then one obtains $\|U\|_1 \leq \alpha$ and $\|y^*\|_\infty \leq \beta \|Uy^*\|_1$. Since the condition number $\alpha\beta$ of a well-conditioned basis depends only on d and since $\|Uy^* - b\|_1 / \|Uy^*\|_1$ is a constant, it implies that the resulting SGD inherits a convergence rate in a relative scale that depends on d and is independent of n.

The idea of using a preconditioner and a sampling distribution according to the leverage scores leads to our main algorithm.

4. Our Main Algorithm

In this section, we will state our main algorithm pwSGD (Algorithm 1) for solving the *constrained* overdetermined ℓ_1 and ℓ_2 regression problems. We now summarize the main steps of our main algorithm as follows.

First, we compute a well-conditioned basis U (Definition 1) for the range space of A implicitly via a conditioning method; see Tables 5 and 6 in Appendix A for a summary of recently proposed randomized conditioning methods. We refer this as the "implicit" method, i.e., it focuses on computing $R \in \mathbb{R}^{d \times d}$ such that $U = AR^{-1}$. A typical way of obtaining R is via the QR decomposition of SA where SA is a sketch of A; see Appendix A for more details.

Second, we either exactly compute or quickly approximate the leverage scores (Definition 2), i.e., the row norms of U as $\{\lambda_i\}_{i=1}^n$. To compute $\{\lambda_i\}_{i=1}^n$ exactly, we have to form the matrix U explicitly, which takes time $\mathcal{O}(nd^2)$. Alternatively, we can estimate the row norms of U without computing the product between A and R^{-1}, in order to further reduce the running time; see Appendix A for more details. We assume that $\{\lambda_i\}_{i=1}^n$ satisfy

$$(1-\gamma)\|U_i\|_p^p \leq \lambda_i \leq (1+\gamma)\|U_i\|_p^p, \tag{6}$$

where γ is the approximation factor of estimation. When the leverage scores are exact, the approximation factor $\gamma = 0$. From that, we can define a distribution P over $\{1, \ldots, n\}$ based on $\{\lambda_i\}_{i=1}^n$ as follows:

$$p_i = \frac{\lambda_i}{\sum_{j=1}^n \lambda_j}. \tag{7}$$

Third, in each iteration a new sample corresponding to a row of A is drawn according to distribution P and we apply an SGD process to solve the following equivalent problem with a specific choice of $F \in \mathbb{R}^{d \times d}$:

$$\min_{y \in \mathcal{Y}} h(y) = \|AFy - b\|_p^p = \mathbb{E}_{\xi \sim P}\left[|A_\xi Fy - b_\xi|^p / p_\xi\right]. \tag{8}$$

Here the matrix F is called the preconditioner for the linear system being solved; see Section 4.2 for several choices of F. Below, we show that with a suitable choice of F, the convergence rate of the

Algorithm 1 pwSGD— preconditioned weighted SGD for over-determined ℓ_1 and ℓ_2 regression

Input: $A \in \mathbb{R}^{n \times d}$, $b \in \mathbb{R}^n$ with $\text{rank}(A) = d$, $x_0 \in \mathcal{Z}$, η and T.
Output: An approximate solution vector to problem $\min_{x \in \mathcal{Z}} \|Ax - b\|_p^p$ for $p = 1$ or 2.
Compute $R \in \mathbb{R}^{d \times d}$ such that $U = AR^{-1}$ is a well-conditioned basis U as described in Section 2.1.
Compute or estimate $\|U_i\|_p^p$ with leverage scores λ_i, for $i \in [n]$, that satisfies (6).
Let $p_i = \frac{\lambda_i}{\sum_{j=1}^n \lambda_j}$, for $i \in [n]$.
Construct the preconditioner $F \in \mathbb{R}^{d \times d}$ based on R; see Section 4.2 for details.
for $t = 0, \ldots, T$ **do**
  Pick ξ_t from $[n]$ based on distribution $\{p_i\}_{i=1}^n$.

$$c_t = \begin{cases} \text{sgn}\left(A_{\xi_t} x_t - b_{\xi_t}\right) / p_{\xi_t} & \text{if } p = 1; \\ 2\left(A_{\xi_t} x_t - b_{\xi_t}\right) / p_{\xi_t} & \text{if } p = 2. \end{cases}$$

  Update x by

$$x_{t+1} = \begin{cases} x_t - \eta c_t H^{-1} A_{\xi_t} & \text{if } \mathcal{Z} = \mathbb{R}^d; \\ \arg\min\limits_{x \in \mathcal{Z}} \ \eta c_t A_{\xi_t} x + \frac{1}{2}\|x_t - x\|_H^2 & \text{otherwise.} \end{cases} \tag{11}$$

  where $H = \left(FF^\top\right)^{-1}$.
end for
Return $\bar{x}$ for $p = 1$ or x_T for $p = 2$.

SGD phase can be improved significantly. Indeed, we can perform the update rule in the original domain (with solution vector x instead of y), i.e., (11). Notice that if $\mathcal{Z} = \mathbb{R}^d$ and $F = I$, then the update rule can be simplified as

$$x_{t+1} = x_t - \eta c_t A_{\xi_t}. \tag{9}$$

If $\mathcal{Z} = \mathbb{R}^d$ and $F = R^{-1}$, then the update rule becomes

$$x_{t+1} = x_t - \eta c_t H^{-1} A_{\xi_t}, \tag{10}$$

where $H = (R^\top R)^{-1}$. In the presence of constraints, (11) only needs to solve an optimization problem with a quadratic objective over the same constraints whose size is independent of n.

Finally, the output is the averaged value over all iterates, i.e., $\bar{x} = \frac{1}{T}\sum_{t=1}^\top x_t$, for ℓ_1 regression, or the last iterate, i.e., x_T, for ℓ_2 regression.

4.1 Main results for ℓ_1 and ℓ_2 regression problems

The quality-of-approximation of Algorithm 1 is presented in Proposition 5 and Proposition 6 for ℓ_1 and ℓ_2 regression, respectively, in which we give the expected number of iterations that pwSGD needs for convergence within small tolerance. We show that pwSGD inherits a convergence rate of $\mathcal{O}\left(1/\sqrt{T}\right)$ for ℓ_1 regression and $\mathcal{O}\left(\log T/T\right)$ for ℓ_2 regression and the constant term only depends on the lower dimension d when $F = R^{-1}$. Worth mentioning is that for ℓ_2 regression, our bound on the solution vector is measured in prediction norm, i.e., $\|Ax\|_2$. For completeness, we

present the non-asymptotic convergence analysis of PWSGD in Proposition 14 and Proposition 15 in Appendix A. All the proofs can be found in Appendix C. The analysis of these results is based on the convergence properties of SGD; see Appendix D for technical details.

In the following results, R is the matrix computed in step 3 in Algorithm 1, $\{\lambda_i\}_{i\in[n]}$, are the leverage scores computed in step 4, F is the preconditioner chosen in step 6 in Algorithm 1 and $H = \left(FF^\top\right)^{-1}$. Denote by $\bar{\kappa}_p(U)$ the condition number of the well-conditioned basis $U = AR^{-1}$ and γ the approximation factor of the leverage scores λ_i, $i \in [n]$, that satisfies (6). For any vector $x \in \mathbb{R}^d$, denote by $\|x\|_H^2 = x^\top Hx$ the ellipsoidal norm of x induced by matrix $H = H^\top \succ 0$. For any non-singular matrix A, denote $\kappa(A) = \|A\|_2\|A^{-1}\|_2$ and $\hat{\kappa}(A) = |A|_1|A^{-1}|_1$. The exact form of the step-sizes used can be found in the proofs [2].

Proposition 5 *For $A \in \mathbb{R}^{n\times d}$ and $b \in \mathbb{R}^n$, define $f(x) = \|Ax-b\|_1$ and suppose $f(x^*) > 0$. Then there exists a step-size η such that after*

$$T = d\bar{\kappa}_1^2(U)\hat{\kappa}^2(RF)\frac{c_1^2c_2c_3^2}{\epsilon^2}$$

iterations, Algorithm 1 with $p = 1$ returns a solution vector estimate $\bar{x}$ that satisfies the expected relative error bound

$$\frac{\mathbb{E}\left[f(\bar{x})\right] - f(x^*)}{f(x^*)} \leq \epsilon.$$

Here, the expectation is taken over all the samples $\xi_1, \ldots, \xi_T$ and x^* is the optimal solution to the problem $\min_{x\in\mathcal{Z}} f(x)$. The constants in T are given by $c_1 = \frac{1+\gamma}{1-\gamma}$, $c_2 = \frac{\|x^*-x_0\|_H^2}{\|x^*\|_H^2}$ and $c_3 = \|Ax^*\|_1/f(x^*)$.

Proposition 6 *For $A \in \mathbb{R}^{n\times d}$ and $b \in \mathbb{R}^n$, define $f(x) = \|Ax-b\|_2$ and suppose $f(x^*) > 0$. Then there exists a step-size η such that after*

$$T = c_1\bar{\kappa}_2^2(U)\kappa^2(RF)\cdot\log\left(\frac{2c_2\kappa^2(U)\kappa^2(RF)}{\epsilon}\right)\cdot\left(1+\frac{\kappa^2(U)\kappa^2(RF)}{c_3\epsilon}\right)$$

iterations, Algorithm 1 with $p = 2$ returns a solution vector estimate x_T that satisfies the expected relative error bound

$$\frac{\mathbb{E}\left[\|A(x_T-x^*)\|_2^2\right]}{\|Ax^*\|_2^2} \leq \epsilon.$$

Furthermore, when $\mathcal{Z} = \mathbb{R}^d$ and $F = R^{-1}$, there exists a step-size η such that after

$$T = c_1\bar{\kappa}_2^2(U)\cdot\log\left(\frac{c_2\kappa^2(U)}{\epsilon}\right)\cdot\left(1+\frac{2\kappa^2(U)}{\epsilon}\right)$$

iterations, Algorithm 1 with $p = 2$ returns a solution vector estimate x_T that satisfies the expected relative error bound

$$\frac{\mathbb{E}\left[f(x_T)\right] - f(x^*)}{f(x^*)} \leq \epsilon.$$

2. The exact expression of the optimal stepsize contains unknown quantities such as x^*. In fact, this is also the case for many SGD-type algorithms. In practice, standard techniques for searching stepsizes can be used. In our experiments, we evaluate our algorithm using theoretically optimal stepsizes as well as stepsizes after grid searching.

Here, the expectation is taken over all the samples $\xi_1, \ldots, \xi_T$, and x^* is the optimal solution to the problem $\min_{x\in\mathcal{Z}} f(x)$. The constants in T are given by $c_1 = \frac{1+\gamma}{1-\gamma}$, $c_2 = \frac{\|x^*-x_0\|_H^2}{\|x^*\|_H^2}$, $c_3 = \|Ax^*\|_2^2/f(x^*)^2$.

The above results indicate two important properties of PWSGD. First recall that the condition number [3] $\bar{\kappa}_p(U)$ of the well-conditioned basis U is a polynomial of d that is independent of n. Thus with a preconditioner $F = R^{-1}$ and an appropriate step-size in PWSGD, the number of iterations T required to achieve an arbitrarily low relative error only depends on the *low dimension* d of the input matrix A. Second, PWSGD is *robust* to leverage score approximations, i.e., the expected convergence rate will only be affected by a small distortion factor even when the approximation has low accuracy, such as $\gamma = 0.5$.

Remark. For constrained ℓ_2 regression, the bound is on the solution vector measured in prediction norm. By the triangular inequality, this directly implies $(\mathbb{E}\left[f(x_T)\right] - f(x^*))/f(x^*) \leq \sqrt{c_3\epsilon}$.

Remark. Our approach can also be applied to other type of linear regression problems such as ridge regressions in which SGD can be invoked in a standard way. In this case, the "condition number" of SGD is lower than κ due to the regularization term. The randomized preconditioning methods discussed in Section 2.1 can be used but it is an "overkill'. More sophisticated preconditioning methods can be devised, e.g., based on ridge leverage scores (Cohen et al., 2015b).

4.2 The choice of the preconditioner F

As we can see, the preconditioner F plays an important role in our algorithm. It converts the original regression problem in (1) to the stochastic optimization problem in (8). From Proposition 5 and Proposition 6, clearly, different choices of F will lead to different convergence rates in the SGD phase (reflected in $\kappa(RF)$[4]) and additional computational costs (reflected in H in (11)).

When $F = R^{-1}$, the effect of $\kappa_2(RF)$ on T vanishes. In this case, H is also a good approximation to the Hessian $A^\top A$. This is because usually R is the R-factor in the QR decomposition of SA, where SA is a "sketch" of A satisfying (2) that shares similar properties with A. Together we have $H = R^\top R = (SA)^\top(SA) \approx A^\top A$. This implies (10) is close to the Newton-type update. However, as a tradeoff, since H^{-1} is a $d \times d$ dense matrix, an additional $\mathcal{O}(d^2)$ cost per iteration is required to perform SGD update (11).

On the other hand, when $F = I$, no matrix-vector multiplication is needed in updating x. However, based on the discussion above, one should expect $\kappa(R) = \kappa(SA)$ to be close to $\kappa(A)$. Then the term $\kappa(RF) = \kappa(R)$ can be large if A is poorly conditioned, which might lead to undesirable performance in SGD phase.

Besides the obvious choices of F such as R^{-1} and I, one can also choose F to be a diagonal preconditioner D that scales R to have unit column norms. According to van der Sluis (1969), the condition number after preconditioning $\kappa(RD)$ is always upper bounded by the original condition number $\kappa(R)$, while the additional cost per iteration to perform SGD updates with diagonal preconditioner is only $\mathcal{O}(d)$. In Section 5 we will illustrate the tradeoffs among these three choices of preconditioners empirically.

3. One can show that $\bar{\kappa}_2$ is a scaled version of the standard condition number κ. $\bar{\kappa}_1$ is also related to κ with $\bar{\kappa}_1 \geq \kappa/\sqrt{nd}$. This implies that in general $\bar{\kappa}_1$ can be large without preconditioning, e.g., the `buzz` dataset used in our experiments.

4. It is also reflected in $\hat{\kappa}(RF)$; however, it depends on $\kappa(RF)$ because one can show $m_1\kappa(RF) \leq \hat{\kappa}(RF) \leq m_2\kappa(RF)$, where m_1, m_2 are constants derived using matrix norm equivalences.

4.3 Complexities

Here, we discuss the complexity of pwSGD with $F = R^{-1}$. The running time of Algorithm 1 consists of three parts. First, for computing a matrix R such that $U = AR^{-1}$ is well-conditioned, Appendix A provides a brief overview of various recently proposed preconditioning methods for computing R for both ℓ_1 and ℓ_2 norms; see also Table 5 and Table 6 for their running time $time(R)$ and preconditioning quality $\bar{\kappa}_p(U)$. Particularly, there are several available sparse preconditioning methods that run in $\mathcal{O}(\mathrm{nnz}(A))$ plus lower order terms in d time (Clarkson and Woodruff, 2013; Meng and Mahoney, 2013a; Nelson and Nguyen, 2013; Yang et al., 2016b; Woodruff and Zhang, 2013). Second, to estimate the leverage scores, i.e., the row norms of AR^{-1}, Drineas et al. (2012); Clarkson et al. (2013) proposed several algorithms for approximating the ℓ_1 and ℓ_2 leverage scores without forming matrix U. For a target constant approximation quality, e.g., $\gamma = 0.5$ and $c_1 = \frac{1+\gamma}{1-\gamma} = 3$, the running time of these algorithms is $\mathcal{O}(\log n \cdot \mathrm{nnz}(A))$. Third, Proposition 5 and Proposition 6 provide upper bounds for the expected algorithmic complexity of our proposed SGD algorithm when a target accuracy is fixed. Combining these, we have the following results.

Proposition 7 *Suppose the preconditioner in step 3 of Algorithm 1, is chosen from Table 5 or Table 6, with constant probability, one of the following events holds for* pwSGD *with* $F = R^{-1}$*. To return a solution* $\tilde{x}$ *with relative error* ϵ *on the objective,*

- *It runs in* $time(R) + \mathcal{O}(\log n \cdot \mathrm{nnz}(A) + d^3\bar{\kappa}_1(U)/\epsilon^2)$ *for unconstrained* ℓ_1 *regression.*
- *It runs in* $time(R)+\mathcal{O}(\log n\cdot\mathrm{nnz}(A)+time_{update}\cdot d\bar{\kappa}_1(U)/\epsilon^2)$ *for constrained* ℓ_1 *regression.*
- *It runs in* $time(R) + \mathcal{O}(\log n \cdot \mathrm{nnz}(A) + d^3\log(1/\epsilon)/\epsilon)$ *for unconstrained* ℓ_2 *regression.*
- *It runs in* $time(R) + \mathcal{O}(\log n \cdot \mathrm{nnz}(A) + time_{update} \cdot d\log(1/\epsilon)/\epsilon^2)$ *for constrained* ℓ_2 *regression.*

In the above, $time(R)$ *denotes the time for computing the matrix* R *and* $time_{update}$ *denotes the time for solving the optimization problem in* (11)*.*

Notice that, since $time_{update}$ only depends on d, an immediate conclusion is that by using sparse preconditioning methods, to find an ϵ-approximate solution, pwSGD runs in $\mathcal{O}(\log n \cdot \mathrm{nnz}(A) + \mathrm{poly}(d)/\epsilon^2)$ time for ℓ_1 regression and in $\mathcal{O}(\log n \cdot \mathrm{nnz}(A) + \mathrm{poly}(d)\log(1/\epsilon)/\epsilon)$ time for ℓ_2 regression (in terms of solution vector in prediction norm for constrained problems or objective value for unconstrained problems).

Also, as can be seen in Proposition 7, for the complexity for ℓ_1 regression, the tradeoffs in choosing preconditioners from Table 5 are reflected here. On the other hand, for ℓ_2 regression, as all the preconditioning methods in Table 5 provide similar preconditioning quality, i.e., $\kappa(U) = \mathcal{O}(1)$, $time(R)$ becomes the key factor for choosing a preconditioning method. In Table 3, we summarize the complexity of pwSGD using various sketching methods for solving unconstrained ℓ_1 and ℓ_2 regression problems. The results are obtained by a direct combination of Tables 2, 5 and 6. We remark that, with decaying step-sizes, it is possible to improve the dependence on ϵ from $\log(1/\epsilon)/\epsilon$ to $1/\epsilon$ (Rakhlin et al., 2012).

Finally, we remind readers that Table 1 and 2 summarize the complexities of several related algorithms for unconstrained ℓ_1 and ℓ_2 regression. As we can see, pwSGD is more suitable for finding a medium-precision, e.g., $\epsilon = 10^{-3}$, solution. In particular, it has a dependency uniformly better than RLA methods for ℓ_1 regression. Moreover, unlike the high-precision solvers, pwSGD also works for constrained problems, in which case each iteration of pwSGD only needs to solve an optimization problem with quadratic objective over the same constraints.

type	sketch	complexity
ℓ_1	Dense Cauchy (Sohler and Woodruff, 2011)	$\mathcal{O}(nd^2\log n + d^3\log d + d^{\frac{11}{2}}\log^{\frac{3}{2}} d/\epsilon^2)$
ℓ_1	Fast Cauchy (Clarkson et al., 2013)	$\mathcal{O}(nd\log n + d^3\log^5 d + d^{\frac{17}{2}}\log^{\frac{9}{2}} d/\epsilon^2)$
ℓ_1	Sparse Cauchy (Meng and Mahoney, 2013a)	$\mathcal{O}(\mathrm{nnz}(A)\log n + d^7\log^5 d + d^{\frac{19}{2}}\log^{\frac{11}{2}} d/\epsilon^2)$
ℓ_1	Reciprocal Exponential (Woodruff and Zhang, 2013)	$\mathcal{O}(\mathrm{nnz}(A)\log n + d^3\log d + d^{\frac{13}{2}}\log^{\frac{5}{2}} d/\epsilon^2)$
ℓ_1	Lewis Weights (Cohen and Peng, 2015)	$\mathcal{O}(\mathrm{nnz}(A)\log n + d^3\log d + d^{\frac{9}{2}}\log^{\frac{1}{2}} d/\epsilon^2)$
ℓ_2	Gaussian Transform	$\mathcal{O}(nd^2 + d^3\log(1/\epsilon)/\epsilon)$
ℓ_2	SRHT (Tropp, 2011)	$\mathcal{O}(nd\log n + d^3\log n\log d + d^3\log(1/\epsilon)/\epsilon)$
ℓ_2	Sparse ℓ_2 embedding (Cohen, 2016)	$\mathcal{O}(\mathrm{nnz}(A)\log n + d^3\log d + d^3\log(1/\epsilon)/\epsilon)$
ℓ_2	Refinement Sampling (Cohen et al., 2015a)	$\mathcal{O}(\mathrm{nnz}(A)\log(n/d)\log d + d^3\log(n/d)\log d + d^3\log(1/\epsilon)/\epsilon)$

Table 3: Summary of complexity of pwSGD with different sketching methods for computing the preconditioner when solving unconstrained ℓ_1 and ℓ_2 regression problems. The target is to return a solution $\tilde{x}$ with relative error ϵ on the objective. Here, the complexity of each algorithm is calculated by setting the failure probability to be a constant.

4.4 Complexity comparison between pwSGD and RLA

As we pointed out in Section 3, pwSGD and RLA methods with algorithmic leveraging (Appendix B) (RLA for short) are closely related as they can be viewed as methods using SA and SAA to solve the stochastic optimization problem (4). Omitting the time for computing basis U and sampling distribution P, the comparison of complexity boils down to comparing $time_{sub}(s, d)$ (for RLA) and $time_{update} \cdot T$ (for pwSGD) where $time_{sub}(s, d)$ is the time needed to solve the same constrained regression problem with size s by d and $time_{update}$ denotes the time needed for to solve the optimization problem in (11). According to the theory, for the same target accuracy, the required s (sampling size) and T (number of iterations) are the same asymptotically, up to logarithmic factors; see Dasgupta et al. (2009); Yang et al. (2014); Drineas et al. (2011) and Section B for expression of s. When the problem is unconstrained, due to the efficiency of SGD, $time_{update} = \mathcal{O}(d^2)$ as indicated in (11). For ℓ_2 regression, due to the efficiency of the direct solver, $time_{sub}(s, d) = \mathcal{O}(sd^2)$. This explains why pwSGD and RLA (low-precision solvers (sampling)) have similar complexities as shown in Table 2. On the other hand, for unconstrained ℓ_1 regression, a typical ℓ_1 regression solver requires time $time_{sub}(s, d) > sd^2$. For example, if an interior point method is used (Portnoy and Koenker, 1997), $time_{sub}(s, d)$ is not even linear in s. This explains the advantage pwSGD has over RLA as shown in Table 1. We also note that in the presence of constraints, pwSGD may still be more efficient for solving ℓ_1 regression because roughly speaking, $time_{sub}(s, d)/s > time_{update}$.

4.5 Connection to weighted randomized Kaczmarz algorithm

As mentioned in Section 1, our algorithm pwSGD for least-squares regression is related to the weighted randomized Kaczmarz (RK) algorithm (Strohmer and Vershynin, 2009; Needell et al., 2014). To be more specific, weighted RK algorithm can be viewed as an SGD algorithm with constant step-size that exploits a sampling distribution based on row norms of A, i.e., $p_i = \|A_i\|_2^2/\|A\|_F^2$. In pwSGD, if the preconditioner $F = R^{-1}$ is used and the leverage scores are computed exactly, the resulting algorithm is equivalent to applying the weighted randomized Karczmarz algorithm on a well-conditioned basis $U = AR^{-1}$ since leverage scores are defined as the row norms of U.

Since the matrix A itself can be a basis for its range space, setting $U = A$ and $F = R = I$ in Proposition 6 indicates that weighted RK algorithm inherits a convergence rate that depends on condition number $\kappa(A)$ times the scaled condition number $\bar{\kappa}_2(A)$. Notice that in pwSGD, the preconditioning step implicitly computes a basis U such that both $\kappa(U)$ and $\bar{\kappa}(U)$ are low. One should expect the SGD phase in pwSGD inherits a faster convergence rate, as verified numerically in Section 5.

5. Experiments

In this section, we provide empirical evaluations of our main algorithm pwSGD. We evaluate its convergence rate and overall running time on both synthetic and real datasets. For pwSGD, we implement it with three different choices of the preconditioner F. Herein, throughout the experiments, by pwSGD-full, pwSGD-diag, pwSGD-noco, we respectively mean pwSGD with preconditioner $F = R^{-1}, D, I$; see Section 4.2 for details. Note that, for pwSGD, we use the methods from Clarkson and Woodruff (2013) for preconditioning. Also, we exactly compute the row norms of AR^{-1} and use them as the leverage scores. In each experiment, the initial solution vector estimate is set as zero. The above algorithms are then run in the following manner. Each epoch contains $\lceil n/10 \rceil$ iterations. At the beginning of each epoch, we sample $\lceil n/10 \rceil$ indices according to the underlying distribution without replacement and update the weight using the $\lceil n/10 \rceil$ row samples from the data matrix. Finally, the plots are generated by averaging the results over 20 independent trials.

5.1 Empirical evaluations on synthetic datasets

Theoretically the major advantage of pwSGD is the fast convergence rate. To evaluate its performance, we compare the convergence rate of relative error, i.e., $|\hat{f} - f^*|/f^*$, with other competing algorithms including vanilla SGD and fully weighted randomized Kaczmarz (weighted-RK) algorithm (Needell et al. (2014); Strohmer and Vershynin (2009)) for solving least-squares problem (unconstrained ℓ_2 regression). For each of these methods, given a target relative error $\epsilon = 0.1$ on the objective, i.e., $(\hat{f} - f^*)/f^* = 0.1$, we use the optimal step-size suggested in the theory. In particular, for pwSGD, we are showing the convergence rate of the SGD phase after preconditioning. We stop the algorithm when the relative error reaches ϵ. In this task, we use synthetic datasets for better control over the properties on input matrices A and b. Each dataset has size 1000 by 10 and is generated in one of the following two ways.

Synthetic 1 The design matrix A has skewed row norms and skewed leverage scores. That is, 5 rows have leverage scores and row norms significantly larger than the rest[5].

Synthetic 2 The design matrix A is of the form $A = U\Sigma V^\top$ where $U \in \mathbb{R}^{1000\times 10}$ and $V \in \mathbb{R}^{10\times 10}$ are random orthonormal matrices and $\Sigma \in \mathbb{R}^{10\times 10}$ is a diagonal matrix that controls $\kappa(A)$.

In both cases, the true solution x^* is a standard Gaussian vector and the response vector b is set to be Ax^* corrupted by some Gaussian noise with standard deviation 0.1.

In Figure 2, we present the results on two `Synthetic 1` datasets with condition number around 1 and 5. From the plots we can clearly see that among the methods we used, pwSGD-full and pwSGD-diag exhibit superior speed in terms of achieving the target accuracy. The relative ordering within pwSGD with three different preconditioners is consistent with the theory according to our discussions in Section 4.2. Since the datasets considered here are well-conditioned, the preconditioning phases in pwSGD-diag and pwSGD-full have similar effects and both methods perform well. However as suggested in Corollary 6, as the condition number increases (in comparison of the results in Figure 2(a) versus Figure 2(b)), other methods show degradations in convergence. Furthermore Figure 2(a) shows that the weighted-RK algorithm outperforms standard SGD. This is due to the fact that A in this dataset is well-conditioned but with non-uniform row norms.

We further investigate the relation between the condition number of A and convergence rate. As suggested in Proposition 6, for weighted SGD algorithm, the number of iterations required to solve an ℓ_2 regression problem is proportional to $\bar{\kappa}_2^2(A)\kappa^2(A) = \|(A^\top A)^{-1}\|_2^2\|A\|_2^2\|A\|_F^2 \leq \bar{\kappa}_2^4(A)$. To verify this hypothesis, we generate a sequence of A matrices using `Synthetic 2` dataset with increasing $\bar{\kappa}_2^4(A)$ values such that U and V in the sequence are constants.[6] This construction ensures that all other properties such as leverage scores and coherence (the largest leverage score) remain unchanged. Similar to Figure 2, we present the experimental results (number of iterations required for different methods versus $\bar{\kappa}_2^4(A)$) for the `Synthetic 2` dataset in Figure 3.

As shown in Figure 3, the required number of iterations of all the methods except for pwSGD-full scales linearly in $\bar{\kappa}_2^4(A)$. This phenomenon matches the result predicted in theory. A significant advantage of pwSGD-full over other methods is its robust convergence rate against variations in $\bar{\kappa}_2^4(A)$. This is mainly because its SGD phase operates on a well-conditioned basis after preconditioning and the preconditioning quality of pwSGD-full depends only on the low-dimension of A; thus increasing $\bar{\kappa}_2^4(A)$ has little effect on changing its convergence rate. Also, while the diagonal preconditioner in pwSGD-diag reduces the condition number, i.e., $\kappa(RD) \leq \kappa(R)$, its convergence rate still suffers from the increase of $\bar{\kappa}_2^4(A)$.

5.2 Time-accuracy tradeoffs

Next, we present the time-accuracy tradeoffs among these methods on the following two datasets described in Table 4.

name	#rows	# columns	$\kappa(A)$
`Year`[7]	5×10^5	90	2×10^3
`Buzz`[8]	5×10^5	77	10^8

Table 4: Summary of the two real datasets we evaluate in the experiments.

Here we test the performance of various methods in solving unconstrained ℓ_1 and ℓ_2 regression problems. Although there are no theoretical results to support the solution vector convergence on ℓ_1 regression problems with pwSGD, we still evaluate relative error in the solution vector. To further examine the performance of pwSGD methods, we also include AdaGrad, SVRG, and RLA methods with algorithmic leveraging (RLA for short) mentioned in Section 3 and Appendix B for comparisons. For AdaGrad, we use diagonal scaling and mirror descent update rule. For SVRG, we compute the full gradient every $\lfloor n/2 \rfloor$ iterations. As for implementation details, in all SGD-like

5. Note that, in general, there is no correlation between row norms and leverage scores unless the matrix has nearly orthonormal columns. For construction details of `Synthetic 1`, see the construction of NG matrices Section 5.3 in Yang et al. (2016b).

6. In `Synthetic 2`, U and V are fixed. Σ is of the form $\mathrm{diag}(\sigma_1, \ldots, \sigma_d)$ where $\sigma_i = 1 + (i-1)q$ for $i \in [d]$. We solve for q such that $\sum_{i=1}^d \sigma_i^2 = \bar{\kappa}_2^2(U)$ for any desired value $\bar{\kappa}_2^2(U)$.

7. `https://archive.ics.uci.edu/ml/datasets/YearPredictionMSD`

8. `https://archive.ics.uci.edu/ml/datasets/Buzz+in+social+media+`

algorithms, step-size tuning is done by grid-searching where at each trial the algorithm is run with a candidate step-size for enough iterations. Then the step-size that yields the lowest error within 10 seconds is used. The time/accuracy pair at every 2000 iterations is recorded. For RLA, we choose s from a wide range of values and record the corresponding time/accuracy pairs. The results on the two datasets are presented in Figures 4 and 5, respectively.

As we can see in Figures 4 and 5, in our algorithm PWSGD, a faster convergence comes with the additional cost of preconditioning. For example, the preconditioning phase of PWSGD takes approximately 0.5 seconds. Nevertheless, with a faster convergence rate in a well-conditioned basis, PWSGD-full still outperforms other methods in converging to a higher-precision solution at a given time span. As PWSGD-diag balances convergence rate and computational cost, it outperforms PWSGD-full at the early stage and yields comparable results to AdaGrad. As expected, due to the poor conditioning, SGD, weighted-RK, SVRG, and PWSGD-noco suffer from slow convergence rates. As for RLA methods, they have the same first step as in PWSGD, i.e., preconditioning and constructing the sampling distribution. For ℓ_1 regression, to obtain a fairly high-precision solution, the sampling size has to be fairly large, which might drastically increase the computation time for solving the sampled subproblem. This explains the advantage of PWSGD-full over RLA methods in Figure 4. It is worth mentioning that, although for ℓ_2 regression our theory provides relative error bound on the solution vector measured in the prediction norm, here we also see that PWSGD-full and PWSGD-diag display promising performance in approximating the solution vector measured in ℓ_2 norm.

We also notice that on `Buzz` (Figure 5), all the methods except for PWSGD-full and PWSGD-diag have a hard time converging to a solution with low solution error. This is due to the fact that `Buzz` has a high condition number. The advantage of applying a preconditioner is manifested.

Finally, notice that RLA uses a high performance direct solver to solve the mid-size subsampled problem for ℓ_2 regression. In this case PWSGD methods do not show significant advantages over RLA in terms of speed. For this reason we have not included RLA results in Figure 4(a) and 4(b). Nevertheless, PWSGD methods may still be favorable over RLA in terms of speed and feasibility when the size of the dataset becomes increasingly larger, e.g., 10^7 by 500.

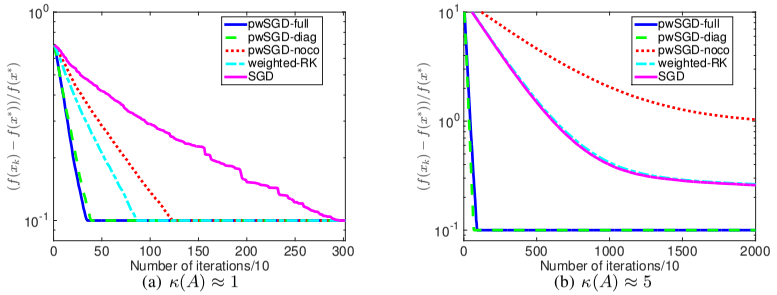


Figure 2: Convergence rate comparison of several SGD-type algorithms including PWSGD with three different choices of preconditioners for solving ℓ_2 regression on `Synthetic 1` datasets with condition number around 1 and 5, respectively. For each method, the optimal step-size is set according to the theory with target accuracy $|f(\hat{x})-f(x^*)|/f(x^*)=0.1$. The y-axis is showing the relative error on the objective, i.e., $|f(\hat{x})-f(x^*)|/f(x^*)$.

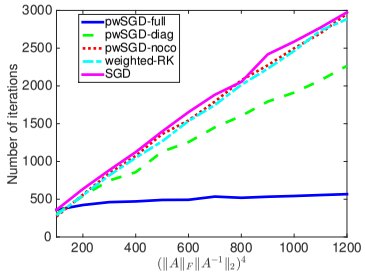


Figure 3: Convergence rate comparison of several SGD-type algorithms including PWSGD with three different choices of preconditioners for solving ℓ_2 regression on `Synthetic 2` datasets with increasing condition number. For each method, the optimal step-size is set according to the theory with target accuracy $|f(\hat{x})-f(x^*)|/f(x^*)=0.1$. The y-axis is showing the minimum number of iterations for each method to find a solution with the target accuracy.

5.3 Empirical evaluations with sparse ℓ_2 regression

Finally, we evaluate our algorithm on a constrained problem — sparse ℓ_2 regression, which is a special case of (1). The problem formulation is as follows. Given a matrix $A\in\mathbb{R}^{n\times d}$ and a vector

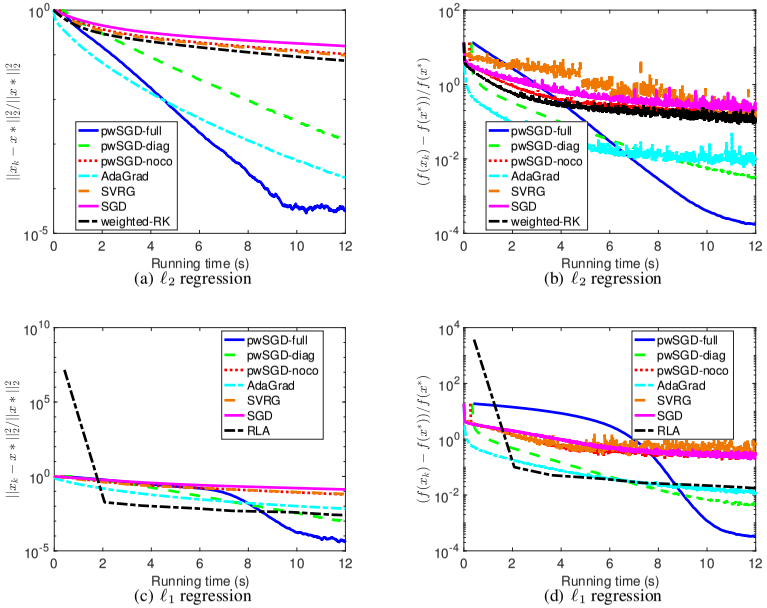


Figure 4: Time-accuracy tradeoffs of several algorithms including pwSGD with three different choices of preconditioners on `year` dataset. Both ℓ_1 and ℓ_2 regressions are tested and the relative error on both the objective value, i.e., $|f(\hat{x}) - f(x^*)|/f(x^*)$, and the solution vector, i.e., $\|\hat{x} - x^*\|_2^2/\|x^*\|_2^2$, are measured.

$b \in \mathbb{R}^n$, we want to solve the following constrained problem

$$\min_{\|x\|_1 \leq R} \|Ax - b\|_2, \tag{12}$$

where R controls the size of the ℓ_1-ball constraint.

When using pwSGD, according to (11) in Algorithm 1, at each iteration, a sparse ℓ_2 regression problem with size d by d needs to be solved. Here, to use the samples more efficiently, we use a mini-batch version of pwSGD. That is, in Step 8-10 of Algorithm 1, rather than picking only one row from A to compute the noisy gradient, we select m rows and average the scaled version of them. Doing this allows us to reduce the variance of the noisy gradient. In our experiments, we set $m = 200$.

In this task, the observation model is generated in the following manner, $b = Ax^* + e$ where $A \in \mathbb{R}^{n \times d}$ has independent standard normal entries, x^* has s nonzero entries and noise vector $e \in \mathbb{R}^n$ has independent standard normal entries. We evaluate both the optimization error $\|\hat{x} -$

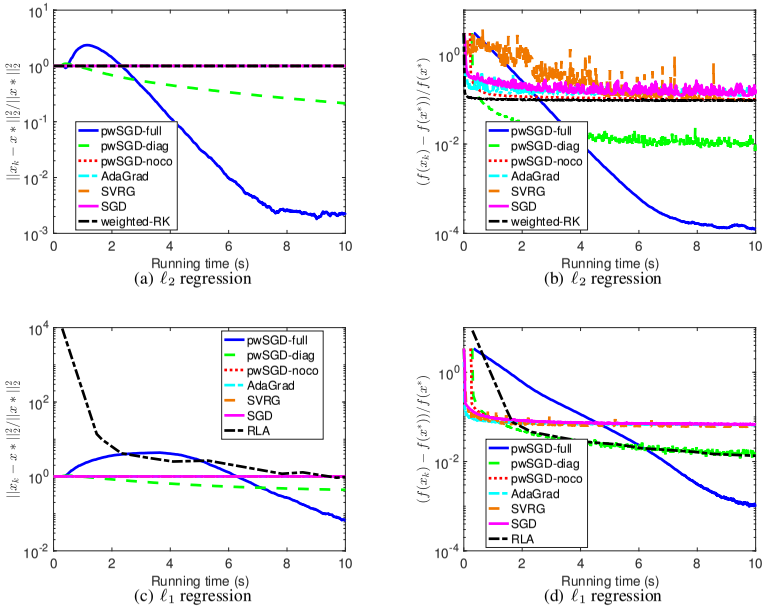


Figure 5: Time-accuracy tradeoffs of several algorithms including pwSGD with three different choices of preconditioners on `buzz` dataset. Both ℓ_1 and ℓ_2 regressions are tested and the relative error on both the objective value, i.e., $|f(\hat{x}) - f(x^*)|/f(x^*)$, and the solution vector, i.e., $\|\hat{x} - x^*\|_2^2/\|x^*\|_2^2$, are measured.

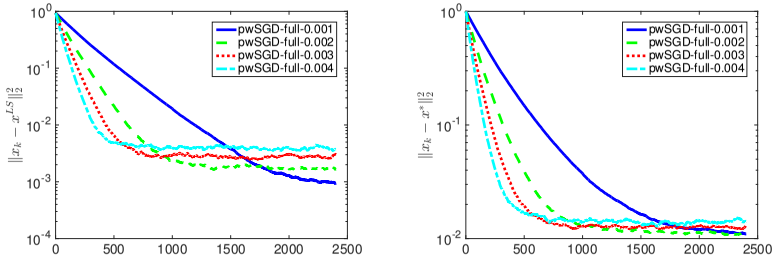
$x^{LS}\|_2$ and statistical error $\|\hat{x} - x^*\|_2$ of pwSGD-full with several choices of stepsize η where x^{LS} the optimal solution of problem (12). It is known that the least squares error of x^{LS} is $\|x^{LS} - x^*\|_2 \approx \sqrt{s\log(ed/s)/n}$ (Hastie et al., 2015). The statistical error can be bounded using the triangle inequality as shown below,

$$\|\hat{x} - x^*\|_2 \leq \|\hat{x} - x^{LS}\|_2 + \|x^{LS} - x^*\|_2.$$

Therefore, the statistical error $\|\hat{x} - x^*\|_2$ is dominated by the least squares error $\|x^{LS} - x^*\|_2$ when the optimization error $\|\hat{x} - x^{LS}\|_2$ is small.

In Figure 6, we show the results on a data instance with $n = 1e4$, $d = 400$ and $s = 30$. Here R is set to be $R = \|x^*\|_1$ for the experimental purpose. First, we briefly describe the effect of stepsize η. When a constant stepsize is used, typically, a smaller η allows the algorithm to converge to a more accurate solution with a slower convergence rate. This is verified by Figure 6(a) in which the performance of pwSGD-full with larger η's saturates earlier at a coarser level while $\eta = 0.001$ allows the algorithm to achieve a finer solution. Nevertheless, as discussed above, the statical error

is typically dominated by the least squares error. For our choice of (n, d, s), one can show that the least squares error $\|x^{LS} - x^*\|_2^2 \approx 0.01$. Therefore, the statistical error shown in Figure 6(b) is around 0.01 when the optimization error is small enough.



(a) Optimization error $\|x_k - x^{LS}\|_2^2$ (b) Statistical error $\|x_k - x^*\|_2^2$

Figure 6: Performance of pwSGD-full on a synthetic sparse ℓ_2 regression problem with difference choices of stepsize η. Both optimization error and statistical error are shown.

6. Connection with Coreset Methods

After viewing RLA and SGD from the stochastic optimization perspective and using that to develop our main algorithm, a natural question arises: can we do this for other types of problems? To do so, we need to define "leverage scores" for them, since they play a crucial role in this stochastic framework. Here, we first describe the coreset framework of Feldman and Langberg (2011). Then we show that—on ℓ_p regression problems—two key notions (leverage scores from RLA and sensitivities from coresets) correspond. Finally we will show what amounts to a negative result (i.e., a lower bound) for other problems. Note here, in this section, we work on constrained ℓ_p regression (1) with $p \in [1, \infty)$ and we use $\bar{A}$ to denote the augmented linear system $\begin{pmatrix} A & b \end{pmatrix}$.

6.1 Short summary of coreset methods

In Feldman and Langberg (2011), the authors propose a framework for computing a coreset of $\mathcal{F}$ to a given optimization problem of the form,

$$\text{cost}(\mathcal{F}, x) = \min_{x \in \mathcal{X}} \sum_{f \in \mathcal{F}} f(x),$$

where $\mathcal{F}$ is a set of functions from a set $\mathcal{X}$ to $[0, \infty)$. By Proposition 4, it is not hard to see, the ℓ_p regression problem (1) can be written as

$$\min_{x \in \mathcal{C}} \sum_{i=1}^{n} f_i(x),$$

Algorithm 2 Compute ϵ-coreset

Input: A class of functions $\mathcal{F}$, sampling size s.
Output: An ϵ-coreset to $\mathcal{F}$.
Initialize $\mathcal{D}$ as an empty set.
Compute the sensitivity $m(f)$ for each function $f \in \mathcal{F}$.
$M(\mathcal{F}) \leftarrow \sum_{f \in \mathcal{F}} m(f)$.
for $f \in \mathcal{F}$ **do**
  Compute probabilities $p(f) = \frac{m(f)}{M(\mathcal{F})}$.
end for
for $i = 1, \ldots, s$ **do**
  Pick f from $\mathcal{F}$ with probability $p(f)$.
  Add $f/(s \cdot p(f))$ to $\mathcal{D}$.
end for
Return $\mathcal{D}$.

where $f_i(x) = |\bar{A}_i x|^p$ and $\mathcal{C} = \{x \in \mathbb{R}^{d+1} | x_{d+1} = -1\}$, in which case one can define a set of functions $\mathcal{F} = \{f_i\}_{i=1}^n$.
Central to the coreset method of Feldman and Langberg (2011) is the following notion of sensitivity, which is used to construct importance sampling probabilities, as shown in Algorithm 2, and the dimension of the given class of function, which is based as Definition 6.1 in Feldman and Langberg (2011). They are defined as below.

Definition 8 *Given a set of function $\mathcal{F} = \{f_i\}_{i=1}^n$, the* sensitivity $m(f)$ *of each function is defined as $m(f) = \lfloor \sup_{x \in \mathcal{X}} n \cdot \frac{f(x)}{\text{cost}(\mathcal{F}, x)} \rfloor + 1$, and the* total sensitivity $M(\mathcal{F})$ *of the set of functions is defined as $M(\mathcal{F}) = \sum_{f \in \mathcal{F}} m(f)$.*

Definition 9 *The dimension of $\mathcal{F}$ is defined as the smallest integer d such that for any $G \subset \mathcal{F}$,*

$$|\{\textbf{Range}(G, x, r) \mid x \in \mathcal{X}, r \geq 0\}| \leq |G|^d,$$

where $\textbf{Range}(G, x, r) = \{g \in G \mid g(x) \leq r\}$.

The algorithm proposed in Feldman and Langberg (2011) is summarized in Algorithm 2 below, and the corresponding result of quality of approximation is presented in Theorem 10.

Theorem 10 *Given a set of functions $\mathcal{F}$ from $\mathcal{X}$ to $[0, \infty]$, if $s \geq \frac{cM(\mathcal{F})}{\epsilon^2}(\dim(\mathcal{F}') + \log\left(\frac{1}{\delta}\right))$, then with probability at least $1 - \delta$, Algorithm 2 returns an ϵ-coreset for $\mathcal{F}$. That is,*

$$(1 - \epsilon) \sum_{f \in \mathcal{F}} f(x) \leq \sum_{f \in \mathcal{D}} f(x) \leq (1 + \epsilon) \sum_{f \in \mathcal{F}} f(x),$$

where $\mathcal{F}' = \{f/s(f) \mid f \in \mathcal{F}\}$ is a rescaled version of $\mathcal{F}$.

6.2 Connections between RLA and coreset methods

In the following, we present two results on the connection between RLA with algorithmic leveraging, i.e., with sampling based on exact or approximate leverage scores, and coreset methods. These results originally appeared in Varadarajan and Xiao (2012). We include them here and give different proofs.

The first result shows that the sensitivities are upper bounded by a constant factor times the ℓ_p leverage scores. With this connection between leverage scores and sensitivities, it is not hard to see that applying Algorithm 2 to ℓ_p regression is exactly the same as applying Algorithm 3 (RLA sampling algorithm described in Appendix B).

Proposition 11 *Given $\bar{A} \in \mathbb{R}^{n\times(d+1)}$, let $f_i(x) = |\bar{A}_i x|^p$, for $i \in [n]$. Let λ_i be the i-th leverage score of $\bar{A}$. Then, the i-th sensitivity*

$$m(f_i) \leq n\beta^p\lambda_i + 1,$$

for $i \in [n]$ and the total sensitivity

$$M(\mathcal{F}) \leq n((\alpha\beta)^p + 1).$$

The second result is that, for the ℓ_p regression problem, the dimension of the class of functions $\dim(\mathcal{F}')$ is the same as the dimension of the subspace being considered, which is $\mathcal{O}(d)$. To be more specific, since all the $f \in \mathcal{F}'$ here are of the form $f(x) = |a^T x|^p$ for some vector $a \in \mathbb{R}^d$, we consider a broader class of functions, namely $\mathcal{A} = \{|a^T x|^p \mid a \in \mathbb{R}^d\}$, and compute its dimension.

Proposition 12 *Let $\mathcal{A} = \{|a^T x|^p \mid a \in \mathbb{R}^d\}$. We have*

$$\dim(\mathcal{A}) \leq d + 1.$$

With these results, in combination with Theorem 10, we can see that to compute a coreset $\mathcal{D}$, which leads to a $\left(\frac{1+\epsilon}{1-\epsilon}\right)$-approximate solution the ℓ_p regression using coreset method of (Feldman and Langberg, 2011), the required sampling complexity is the same (up to constants) as that of RLA sampling algorithm, as indicated by Theorem 16 (assuming $\gamma = 0$) in Appendix B.

6.3 Limitation of our approach

From the above, we see that for ℓ_p regression, a small coreset whose size only depends on d exists, and by solving it we can get a $(1+\epsilon)$-approximation solution. This results in the same sampling algorithm as in RLA. Also, the sensitivities defined in the framework can be used as a distribution when one converts a deterministic problem into a stochastic optimization problem. We want to see whether we can extend this scheme to other problems. Indeed, beyond ℓ_p regression, the coreset methods work for any kind of convex loss function (Feldman and Langberg, 2011). However, since it depends on the total sensitivity, the size of the coreset is not necessarily small. For RLA, this translates into requiring a very large sample size to construct a good subproblem. For example, for hinge loss, we have the following example showing that the size of the coreset has an exponential dependency on d.

Proposition 13 *Define $f_i(x) = f(x, a_i) = (x^T a_i)^+$, where $x, a_i \in \mathbb{R}^d$ for $i \in [n]$. There exists a set of vectors $\{a_i\}_{i=1}^d$ such that the total sensitivity of $\mathcal{F} = \{f_i\}_{i=1}^n$ is approximately 2^d.*

This result indicates that new ideas will be needed to extend similarly RLA preconditioning ideas to weighted SGD algorithms for other types of convex optimization problems. This should not be surprising, since RLA methods have been developed for randomized *linear* algebra problems, but it suggests several directions for follow-up work.

7. Conclusion

In this paper, we propose a novel RLA-SGD hybrid algorithm called pwSGD. We show that after a preconditioning step and constructing a non-uniform sampling distribution using RLA techniques, its SGD phase inherits fast convergence rates that only depend on the lower dimension of the input matrix. For ℓ_1 regression, pwSGD displays strong advantages over RLA methods in terms of the overall complexity. For ℓ_2 regression, it has a complexity comparable to that of several state-of-the-art solvers. Empirically we show that pwSGD is preferable when a medium-precision solution is desired. Finally, we provide lower bounds on the coreset complexity for more general regression problems, which point to specific directions for future work to extend our main results.

Acknowledgments. We would like to acknowledge the Army Research Office, the Defense Advanced Research Projects Agency, and the Department of Energy for providing partial support for this work.

References

H. Avron, P. Maymounkov, and S. Toledo. Blendenpik: Supercharging LAPACK's least-squares solver. *SIAM J. on Scientific Computing*, 32(3):1217–1236, 2010.

R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition*. SIAM, Philadelphia, 1994.

L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Computational Statistics (COMPSTAT)*, 2010.

L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In *Neural Information Processing Systems (NIPS)*, 2008.

L. Bottou and Y. Le Cun. Large scale online learning. In *Neural Information Processing Systems (NIPS)*, 2004.

R. H. Byrd, S. L. Hansen, J. Nocedal, and Y. Singer. A stochastic quasi-Newton method for large-scale optimization. *SIAM J. on Optimization*, 26(2):1008–1031, 2016.

S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159, 2001.

K. L. Clarkson. Subgradient and sampling algorithms for ℓ_1 regression. In *Symposium on Discrete Algorithms (SODA)*, 2005.

K. L. Clarkson and D. P. Woodruff. Low rank approximation and regression in input sparsity time. In *Symposium on Theory of Computing (STOC)*, 2013.

K. L. Clarkson, D. Eppstein, G. L. Miller, C. Sturtivant, and S. Teng. Approximating center points with iterated radon points. In *Symposium on Computational Geometry*, 1993.

K. L. Clarkson, P. Drineas, M. Magdon-Ismail, M. W. Mahoney, X. Meng, and D. P. Woodruff. The Fast Cauchy Transform and faster robust linear regression. In *Symposium on Discrete Algorithms (SODA)*, 2013.

M. B. Cohen. Nearly tight oblivious subspace embeddings by trace inequalities. In *Symposium on Discrete Algorithms (SODA)*, 2016.

M. B. Cohen and R. Peng. ℓ_p row sampling by Lewis weights. In *Symposium on the Theory of Computing (STOC)*, 2015.

M. B. Cohen, Y. T. Lee, C. Musco, C. Musco, R. Peng, and A. Sidford. Uniform sampling for matrix approximation. In *Conference on Innovations in Theoretical Computer Science (ITCS)*, 2015a.

M. B. Cohen, C. Musco, and C. Musco. Ridge leverage scores for low-rank approximation. *CoRR*, abs/1511.07263, 2015b.

F. Curtis. A self-correcting variable-metric algorithm for stochastic optimization. In *International Conference on Machine Learning (ICML)*, 2016.

A. Dasgupta, P. Drineas, B. Harb, R. Kumar, and M. W. Mahoney. Sampling algorithms and coresets for ℓ_p regression. *SIAM J. on Computing*, 38(5):2060–2078, 2009.

P. Drineas, M. W. Mahoney, S. Muthukrishnan, and T. Sarlós. Faster least squares approximation. *Numer. Math.*, 117(2):219–249, 2011.

P. Drineas, M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff. Fast approximation of matrix coherence and statistical leverage. *J. Machine Learning Research*, 13:3441–3472, 2012.

J. C. Duchi, S. Shalev-Shwartz, Y. Singer, and A. Tewari. Composite objective mirror descent. In *Conference on Learning Theory (COLT)*, 2010.

J. C. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Machine Learning Research*, 12:2121–2159, 2011.

D. Feldman and M. Langberg. A unified framework for approximating and clustering data. In *Symposium on Theory of Computing (STOC)*, 2011.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 1996.

T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press, 2015.

C. Hu, J. T. Kwok, and W. Pan. Accelerated gradient methods for stochastic optimization and online learning. In *Neural Information Processing Systems (NIPS)*, 2009.

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Neural Information Processing Systems (NIPS)*, 2013.

C. T. Kelley. *Iterative Methods for Solving Linear and Nonlinear Equations*. SIAM, Philadelphia, 1995.

P. Ma, M. W. Mahoney, and B. Yu. A statistical perspective on algorithmic leveraging. In *International Conference on Machine Learning (ICML)*, 2014.

M. W. Mahoney. *Randomized algorithms for matrices and data*. Foundations and Trends in Machine Learning. NOW Publishers, Boston, 2011.

X. Meng and M. W. Mahoney. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *Symposium on the Theory of Computing (STOC)*, 2013a.

X. Meng and M. W. Mahoney. Robust regression on MapReduce. In *International Conference on Machine Learning (ICML)*, 2013b.

X. Meng, M. A. Saunders, and M. W. Mahoney. LSRN: A parallel iterative solver for strongly over- or under-determined systems. *SIAM J. on Scientific Computing*, 36(2):C95–C118, 2014.

P. Moritz, R. Nishihara, and M. I. Jordan. A linearly-convergent stochastic L-BFGS algorithm. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.

D. Needell, R. Ward, and N. Srebro. Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. In *Neural Information Processing Systems (NIPS)*, 2014.

J. Nelson and H. L. Nguyen. OSNAP: faster numerical linear algebra algorithms via sparser subspace embeddings. In *Symposium on Foundations of Computer Science (FOCS)*, 2013.

F. Niu, B. Recht, C. Ré, and J. S Wright. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Neural Information Processing Systems (NIPS)*, 2011.

M. Pilanci and M. J. Wainwright. Iterative Hessian sketch: Fast and accurate solution approximation for constrained least-squares. *ArXiv e-prints*, 2014.

S. Portnoy. On computation of regression quantiles: Making the Laplacian tortoise faster. *Lecture Notes-Monograph Series, Vol. 31, L_1-Statistical Procedures and Related Topics*, pages 187–200, 1997.

S. Portnoy and R. Koenker. The Gaussian hare and the Laplacian tortoise: Computability of squared-error versus absolute-error estimators, with discussion. *Statistical Science*, 12(4):279–300, 1997.

A. Rakhlin, O. Shamir, and K. Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In *International Conference on Machine Learning (ICML)*, 2012.

Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, 2003.

N. Sauer. On the density of families of sets. *J. Combinatorial Theory, Series A*, 13(1):145–147.

S. Shalev-Shwartz and N. Srebro. SVM optimization: inverse dependence on training set size. In *International Conference on Machine Learning (ICML)*, 2008.

S. Shalev-Shwartz and A. Tewari. Stochastic methods for ℓ_1 regularized loss minimization. In *International Conference on Machine Learning (ICML)*, 2009.

S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub–gradient solver for SVM. In *International Conference on Machine Learning (ICML)*, 2007.

C. Sohler and D. P. Woodruff. Subspace embedding for the ℓ_1-norm with applications. In *Symposium on Theory of Computing (STOC)*, 2011.

T. Strohmer and R. Vershynin. A randomized Kaczmarz algorithm with exponential convergence. *J. Fourier Anal. Appl.*, 15(2), 2009.

J. A. Tropp. Improved analysis of the subsampled randomized Hadamard transform. *Adv. Adapt. Data Anal.*, 3(1-2):115–126, 2011.

A. van der Sluis. Condition numbers and equilibration of matrices. *Numerische Mathematik*, 14(1): 14–23, 1969.

K. Varadarajan and X. Xiao. On the sensitivity of shape fitting problems. In *Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, 2012.

D. P. Woodruff and Q. Zhang. Subspace embeddings and ℓ_p-regression using exponential random variables. *Conference on Learning Theory (COLT)*, 2013.

J. Yang, X. Meng, and M. W. Mahoney. Quantile regression for large-scale applications. *SIAM J. Scientific Computing*, 36(5):S78–S110, 2014.

J. Yang, Y. Chow, C. Ré, and M. W. Mahoney. Weighted SGD for ℓ_p regression with randomized preconditioning. In *Symposium on Discrete Algorithms (SODA)*, 2016a.

J. Yang, X. Meng, and M. W. Mahoney. Implementing randomized matrix algorithms in parallel and distributed environments. *Proceedings of the IEEE*, 104(1):58–92, 2016b.

P. Zhao and T. Zhang. Stochastic optimization with importance sampling. In *International Conference on Machine Learning (ICML)*, 2015.

Appendix A. Supplementary Details of Algorithm 1

As we discussed, we need to compute a well-conditioned basis implicitly and estimate its row norms, i.e., AR^{-1} and $\{\lambda_i\}_{i=1}^n$ in Steps 3 and 4 in Algorithm 1.

In Section 2.1 we have summarized the major steps for computing the preconditioner using sketching. Below in Table 5 we provide a short summary of preconditioning methods using various sketches along with the resulting running time and condition number. Note that the running time here denotes the total running for computing the matrix R which is the sketching time plus the time

for QR factorization of the sketch. Again, below $\bar{\kappa}_p(U)$ is the condition number of $U = AR^{-1}$ as defined in Definition 1 and $\kappa(U)$ is the standard condition number of U.

name	running time	$\bar{\kappa}_1(U)$
Dense Cauchy Transform (Sohler and Woodruff, 2011)	$\mathcal{O}(nd^2 \log d + d^3 \log d)$	$\mathcal{O}(d^{5/2} \log^{3/2} d)$
Fast Cauchy Transform (Clarkson et al., 2013)	$\mathcal{O}(nd \log d + d^3 \log d)$	$\mathcal{O}(d^{11/2} \log^{9/2} d)$
Sparse Cauchy Transform (Meng and Mahoney, 2013a)	$\mathcal{O}(\text{nnz}(A) + d^7 \log^5 d)$	$\mathcal{O}(d^{\frac{13}{2}} \log^{\frac{11}{2}} d)$
Reciprocal Exponential Transform (Woodruff and Zhang, 2013)	$\mathcal{O}(\text{nnz}(A) + d^3 \log d)$	$\mathcal{O}(d^{\frac{7}{2}} \log^{\frac{5}{2}} d)$
Lewis Weights (Cohen and Peng, 2015)	$\mathcal{O}(\text{nnz}(A) \log n + d^3 \log d)$	$\mathcal{O}(d^{\frac{3}{2}} \log^{\frac{1}{2}} d)$

Table 5: Summary of running time and condition number, for several different ℓ_1 conditioning methods. The failure probability of each chmethod is set to be a constant.

name	running time	$\kappa_2(U)$	$\bar{\kappa}_2(U)$
Gaussian Transform	$\mathcal{O}(nd^2)$	$\mathcal{O}(1)$	$\mathcal{O}(\sqrt{d})$
SRHT (Tropp, 2011)	$\mathcal{O}(nd \log n + d^3 \log n \log d)$	$\mathcal{O}(1)$	$\mathcal{O}(\sqrt{d})$
Sparse ℓ_2 Embedding (Clarkson and Woodruff, 2013)	$\mathcal{O}(\text{nnz}(A) + d^4)$	$\mathcal{O}(1)$	$\mathcal{O}(\sqrt{d})$
Sparse ℓ_2 Embedding[9] (Cohen, 2016)	$\mathcal{O}(\text{nnz}(A) \log d + d^3 \log d)$	$\mathcal{O}(1)$	$\mathcal{O}(\sqrt{d})$
Refinement Sampling (Cohen et al., 2015a)	$\mathcal{O}(\text{nnz}(A) \log(n/d) \log d + d^3 \log(n/d) \log d)$	$\mathcal{O}(1)$	$\mathcal{O}(\sqrt{d})$

Table 6: Summary of running time and condition number, for several different ℓ_2 conditioning methods. Here, we assume $d \leq n \leq e^d$. The failure probability of each method is set to be a constant.

Next, given the implicit representation of U by R, to compute the leverage scores $\|U_i\|_p^p$ exactly, one has to compute U which takes $\mathcal{O}(nd^2)$ time. Instead of forming U explicitly and "reading off" the row norms for computing the leverage scores, one can estimate the row norms of U up to a small factor by post-multiplying a random projection matrix; see Clarkson et al. (2013); Drineas et al. (2012) for the cases when $p = 1, 2$ respectively. The above process can be done in $\mathcal{O}(\text{nnz}(A) \cdot \log n)$ time.

Finally, we present two additional results regarding the non-asymptotic convergence rate of PWSGD on ℓ_1 and ℓ_2 regression, respectively. Notation is similar to the one used in Proposition 5 and Proposition 6.

Proposition 14 *For $A \in \mathbb{R}^{n \times d}$ and $b \in \mathbb{R}^n$, define $f(x) = \|Ax - b\|_1$. Algorithm 1 with $p = 1$ returns a solution vector estimate $\bar{x}$ that satisfies the following expected error bound*

$$\mathbb{E}\left[f(\bar{x})\right] - f(x^*) \leq \frac{1}{2\eta T}\|x^* - x_1\|_H^2 + \frac{\eta}{2}\left(c_1 \alpha \|RF\|_1\right)^2. \tag{13}$$

Hereby, the expectation is taken over all the samples $\xi_1, \ldots, \xi_T$ and x^* is an optimal solution to the problem $\min_{x \in \mathcal{Z}} f(x)$. The constant in the error bound is given by $c_1 = \frac{1+\gamma}{1-\gamma}$.

9. In Cohen (2016), the author analyzes a more general version of the original count-sketch like sparse ℓ_2 embedding (Clarkson and Woodruff, 2013). By setting the sparsity parameter differently, different running time complexities can be achieved.

Proposition 15 *For $A \in \mathbb{R}^{n\times d}$ and $b \in \mathbb{R}^n$, define $f(x) = \|Ax - b\|_2$. Algorithm 1 with $p = 2$ returns a solution vector estimate x_T that satisfies the following expected error bound*

$$\mathbb{E}\left[\|x_t - x^*\|_H^2\right] \leq \left(\frac{1 - 4\eta\left(1 - 2\eta c_1\alpha^2\|RF\|_2^2\right)}{\beta^2\|(RF)^{-1}\|_2^2}\right)^T \|x_0 - x^*\|_H^2 + \frac{2c_1\eta\bar{\kappa}_2^2(U)\kappa^2(RF)h(y^*)}{1 - 2c_1\eta\alpha^2\|RF\|_2^2}. \tag{14}$$

Hereby, $H = (F^{-1})^\top F^{-1}$ is the weighs of the ellipsoidal norm and the expectation is taken over all the samples $\xi_1, \ldots, \xi_T$ and x^* is an optimal solutions to the problem $\min_{x\in\mathcal{Z}} f(x)$. The constant in the error bound is given by $c_1 = \frac{1+\gamma}{1-\gamma}$.

Appendix B. RLA Methods with Algorithmic Leveraging

In this section, we present the RLA sampling algorithms with algorithmic leveraging for solving ℓ_p regression problems mentioned in Section 3. The main idea in this class of algorithms is to sample rows based on the leverage scores of $\begin{pmatrix} A & b \end{pmatrix}$ and solve the sample average approximation of the ℓ_p regression problem. This method is formally stated in Algorithm 3.

The following theorem (from Dasgupta et al. (2009)) states that if the sampling size s is large enough, the resulting approximation solution $\hat{x}$ produces a $\left(\frac{1+\epsilon}{1-\epsilon}\right)$-approximation to the original solution vector. The following theorem also shows that when the desired accuracy and confidence interval are fixed, the required sampling size only depends on the lower dimension d since α and β are independent of n.

Theorem 16 *Given input matrix $A \in \mathbb{R}^{n\times d}$ and vector $b \in \mathbb{R}^n$, let α, β be the condition numbers of the well-conditioned basis U for the range space of $\begin{pmatrix} A & b \end{pmatrix}$ and γ be the quality of approximation to the leverage scores satisfying* (6)*. Then when $\epsilon < 1/2$ and the sampling size satisfies the following condition*

$$s \geq \frac{1+\gamma}{1-\gamma}\frac{(32\alpha\beta)^p}{p^2\epsilon^2}\left((d+1)\log\left(\frac{12}{\epsilon}\right) + \log\left(\frac{2}{\delta}\right)\right), \tag{15}$$

Algorithm 3 returns a solution vector $\hat{x}$ that satisfies the following inequality with probability at least $1 - \delta$,

$$\|A\hat{x} - b\|_p \leq \left(\frac{1+\varepsilon}{1-\varepsilon}\right)\|Ax^* - b\|_p, \tag{16}$$

where $x^* \in \mathcal{Z}$ is an optimal solution to the original problem $\min_{x\in\mathcal{Z}} \|Ax - b\|_p$.

Remark. Compared to the RLA algorithm described in Section 3, the algorithm described here computes the leverage scored based on a basis for the range space of the augmented linear system $\bar{A} = \begin{pmatrix} A & b \end{pmatrix}$ rather than A. One can show similar results if the basis is computed for the range space of A.

Remark. As can be seen, the sampling size is $s = \mathcal{O}(\text{poly}(d)\log(1/\epsilon)/\epsilon^2)$ for a target accuracy ϵ. For unconstrained ℓ_2 regression, however, it can be shown that a sampling size $s = \mathcal{O}(\text{poly}(d)\log(1/\epsilon)/\epsilon)$ is sufficient to compute an ϵ-approximate solution; see Drineas et al. (2011); Clarkson and Woodruff (2013) for details.

Appendix C. Proofs

Here, we present the proofs of the theoretical results in the main text.

Algorithm 3 RLA methods with algorithmic leveraging for constrained ℓ_p regression

Input: $A \in \mathbb{R}^{n\times d}$, $b \in \mathbb{R}^n$ with $\text{rank}(\bar{A}) = k$ where $\bar{A} = \begin{pmatrix} A & b \end{pmatrix}$, $\mathcal{Z}$ and $s > 0$.
Output: An approximate solution $\hat{x} \in \mathbb{R}^d$ to problem $\text{minimize}_{x\in\mathcal{Z}} \|Ax - b\|_p^p$.
Compute $R \in \mathbb{R}^{k\times(d+1)}$ such that $\bar{A} = UR$ and U is an (α, β) well-conditioned basis U for the range space of $\bar{A}$.
Compute or estimate $\|U_i\|_p^p$ by λ_i satisfying (6) with γ, for $i \in [n]$.
Let $p_i = \frac{\lambda_i}{\sum_{j=1}^n \lambda_j}$, for $i \in [n]$.
Let $S \in \mathbb{R}^{s\times n}$ be a zero matrix.
for $i = 1, \ldots, s$ **do**
  Pick ξ_t from $[n]$ based on distribution $\{p_i\}_{i=1}^n$.
  Set $S_{i,\xi_t} = \left(\frac{1}{p_{\xi_t}}\right)^{\frac{1}{p}}$.
end for
Return $\hat{x} = \arg\min_{x\in\mathcal{Z}} \|SAx - Sb\|_p$.

C.1 Proof of Proposition 7

Consider the following three events:

- $\mathcal{E}_1$: Compute a matrix R such that $U = AR^{-1}$ has condition number $\bar{\kappa}_p$, and then compute $F = R^{-1}$ and $H = \left(FF^\top\right)^{-1}$.
- $\mathcal{E}_2$: Given R^{-1}, compute $\{\lambda_i\}_{i=1}^n$ as an estimation of row norms of AR^{-1} satisfying (6) with $\gamma = 0.5$.
- $\mathcal{E}_3$: For a given basis U with condition number $\bar{\kappa}_p(U)$ and $\{\lambda_i\}_{i=1}^n$ with approximation quality γ, pwSGD returns a solution with the desired accuracy with iterations $10T$ where T is specified in Proposition 5 or Proposition 6.

Since each preconditioning method shown in Table 5 successes with constant probability, $\mathcal{E}_1$ holds with a constant probability. Also, as introduced in Appendix A, $\mathcal{E}_2$ has a constant failure probability. Finally, by Markov inequality, we know that $\mathcal{E}_3$ holds with probability at least 0.9. As setting the failure probability of $\mathcal{E}_1$ and $\mathcal{E}_2$ to be arbitrarily small will not alter the results in big-O notation, we can ensure that, with constant probability, $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3$ holds.

Conditioned on the fact that $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3$ holds, to converge to the desired solution, for ℓ_1 regression, pwSGD runs in $\mathcal{O}(d\bar{\kappa}_1(U)/\epsilon^2)$ iterations. Since all the preconditioning methods in Table 6 provide $\kappa(U) = \mathcal{O}(1)$ and $\bar{\kappa}_2(U) = \mathcal{O}(\sqrt{d})$, for unconstrained ℓ_2 regression, it runs in $\mathcal{O}(d\log(1/\epsilon)/\epsilon)$ iterations. For constrained ℓ_2 regression, since an ϵ-approximate solution in terms of the solution vector measured in the prediction norm implies a $\sqrt{\epsilon}$-approximate solution on the objective, it runs in $\mathcal{O}(d\log(1/\epsilon)/\epsilon^2)$ iterations to return an ϵ-solution in the objective value.

The overall complexity is the sum of the complexity needed in each of the above events. For $\mathcal{E}_1$, it is $time(R)$ since the time for computing F and H is $\mathcal{O}(d^3)$ which can absorbed into $time(R)$ and they only have to be computed once. For $\mathcal{E}_2$, it is $\mathcal{O}(\text{nnz}(A)\cdot\log n)$. Finally, for $\mathcal{E}_3$, when the problem is unconstrained, $time_{update} = \mathcal{O}(d^2)$; when the problem is constrained, $time_{update} = \text{poly}(d)$. Combining these, we get the complexities shown in the statement. This completes the proof.

C.2 Proof of Proposition 14

The proof of this proposition is structured as follows. First we reformulate the problem using Proposition 4. Second we show that the sequence of solution vector estimates $\{x_t\}_{t=1}^T$ in Algorithm 1 is equivalent to the solution vector estimates $\{y_t\}_{t=1}^T$ obtained by running SGD on the equivalent problem. Third, we analyze the convergence rate of $\{y_t\}_{t=1}^T$ and conclude the error bound analysis.

Problem reformulation Suppose U is an ℓ_p well-conditioned basis for the range space of A and $A = UR$ for some nonsingular matrix R. Let P be the distribution defined based on the estimation of the corresponding leverage scores. That is, for $i \in [n]$,

$$p_i = \frac{\lambda_i}{\sum_{j=1}^n \lambda_j}, \tag{17}$$

where λ_i is an estimation of $\|U_i\|_p^p$ satisfying

$$(1-\gamma)\|U_i\|_p^p \le \lambda_i \le (1+\gamma)\|U_i\|_p^p. \tag{18}$$

This implies

$$\frac{1-\gamma}{1+\gamma}\frac{\|U_i\|_p^p}{\|U\|_p^p} \le p_i \le \frac{1+\gamma}{1-\gamma}\frac{\|U_i\|_p^p}{\|U\|_p^p}. \tag{19}$$

From Proposition 4, recall that for any non-singular matrix $F \in \mathbb{R}^{(d+1)\times(d+1)}$, the constrained ℓ_p regression problem

$$\min_{x\in\mathcal{Z}} f(x) := \|Ax-b\|_p^p \tag{20}$$

can be equivalently written as the following stochastic optimization problem,

$$\min_{y\in\mathcal{Y}} h(y) = \|URFy-b\|_p^p = \mathbb{E}_{\xi\sim P}\left[|U_\xi RFy - b_\xi|^p/p_\xi\right]. \tag{21}$$

Notice that by comparing to the objective function defined in (1) where $f(x) = \|Ax-b\|_p$, we rewrite $f(x)$ into the form of the sum of subfunctions, i.e., $f(x) = \|Ax-b\|_p^p$, so that SGD can be applied.

Equivalence of sequences By using the following linear transformation, one notices that the sequence $\{x_t\}_{t=1}^T$ obtained by (11) in Algorithm 1 has a one-to-one correspondence to the sequence $\{y_t\}_{t=1}^T$ obtained by running SGD on problem (21):

$$\begin{aligned} Fy_t &= x_t, \\ F\bar{y} &= \bar{x}, \\ Fy^* &= x^*. \end{aligned} \tag{22}$$

Thus with condition (22), immediately the objective function value has the following equivalence as well:

$$\begin{aligned} h(y_t) &= f(x_t), \\ h(\bar{y}) &= f(\bar{x}), \\ h(y^*) &= f(x^*), \end{aligned} \tag{23}$$

where $\bar{x} = \frac{1}{T}\sum_{i=1}^T x_t$, $\bar{y} = \frac{1}{T}\sum_{i=1}^T y_t$ and x^* and y^* are the optimal point to optimization problem (20) and (21) respectively.

Now we prove (22) by induction. By defining $Fy_0 = x_0$, one immediately shows that the equivalence condition holds at the base case ($t = 0$). Now by induction hypothesis, assume (22) holds for case $t = k$. Now for $t = k+1$, we show that x_{k+1} returned by Algorithm 1 and y_{k+1} returned by the update rule of SGD satisfy (22).

For simplicity, assume that at k-th iteration, the i-th row is picked. For subfunction $h_k(y) = |U_iRFy|^p - b_i/p_i$, its (sub)gradient is

$$g_k(y) = p\cdot \mathrm{sgn}(U_iRFy - b_i)\cdot(U_iRFy-b_i)^{p-1}\cdot U_iRF/p_i, \tag{24}$$

for which the SGD update rule becomes

$$y_{k+1} = \arg\min_{y\in\mathcal{Y}} \eta\langle y - y_k, c_kU_iRF\rangle + \frac{1}{2}\|y_k - y\|_2^2, \tag{25}$$

where $c_k = p\cdot\mathrm{sgn}(U_iRFy-b_i)\cdot(U_iRFy-b_i)^{p-1}/p_i$ is the corresponding (sub)gradient. Recall the linear transformation $Fy_k = x_k$, feasible set $\mathcal{Y} = \{y\in\mathbb{R}^k | y = F^{-1}x, x\in\mathcal{Z}\}$ and input matrix $A_i = U_iR$, the update rule (25) becomes

$$x_{k+1} = \arg\min_{x\in\mathcal{Z}} \eta c_kA_ix + \frac{1}{2}\|F^{-1}(x_k - x)\|_2^2. \tag{26}$$

The equation above is exactly the update performed in (11). In particular, when $\mathcal{Z} = \mathbb{R}^d$, i.e., in the unconstrained case, (26) has a closed-form solution as shown in (11). From the above analysis on the equivalence between (25) and (26), one notices x_{k+1} and y_{k+1} satisfy the relationship defined in (22), i.e., the induction hypothesis holds at $t = k+1$.

Therefore by induction, we just showed that condition (22), and therefore condition (23), hold for any t.

Convergence rate Based on the equivalence condition in (23), it is sufficient to analyze the performance of sequence $\{y_t\}_{t=1}^T$. When $p = 1$, the objective function is non-differentiable. Thus by substituting the subgradient of an ℓ_1 objective function to the update in (25), one notices that the SA method simply reduces to stochastic subgradient descent. We now analyze the convergence rate of running stochastic subgradient descent on problem (21) with $p = 1$.

Suppose the i-th row is picked at the t-th iteration. Recall that the (sub)gradient of the sample objective $|U_iRFy - b_i|/p_i$ in (25) is expressed as

$$g_t(y) = \mathrm{sgn}(U_iRFy - b_i)\cdot U_iRF/p_i. \tag{27}$$

Hence, by inequality (19), the norm of $g_t(y)$ is upper-bounded as follows:

$$\begin{aligned} \|g_t(y)\|_1 &= \|U_iRF\cdot\mathrm{sgn}(U_iRFy-b_i)\|_1/p_i \\ &\le |RF|_1\|U_i\|_1\frac{1+\gamma}{1-\gamma}\cdot\frac{|U|_1}{\|U_i\|_1} \le \alpha|RF|_1\frac{1+\gamma}{1-\gamma}. \end{aligned} \tag{28}$$

In above, we use the property of the well-conditioned basis U. Furthermore by Proposition 17 and the equivalence condition in (23), for $H = \left(FF^\top\right)^{-1}$ we have

$$\begin{aligned}\mathbb{E}\left[f(\bar{x})\right] - f(x^*) &= \mathbb{E}\left[h(\bar{y})\right] - h(y^*) \quad (29)\\ &\leq \frac{1}{2\eta(T+1)}\|y^* - y_0\|_2^2 + \frac{\eta}{2}\left(\alpha|RF|_1\frac{1+\gamma}{1-\gamma}\right)^2\\ &= \frac{1}{2\eta(T+1)}\|x^* - x_0\|_H^2 + \frac{\eta}{2}\left(\alpha|RF|_1\frac{1+\gamma}{1-\gamma}\right)^2, \quad (30)\end{aligned}$$

which completes the proof.

C.3 Proof of Proposition 5

By Proposition 17, when the step-size equals to

$$\eta = \frac{\|y^* - y_0\|_2}{\alpha|RF|_1\sqrt{T+1}}\frac{1-\gamma}{1+\gamma},$$

the expected error bound is given by

$$\mathbb{E}\left[h(\bar{y})\right] - h(y^*) \leq \alpha|RF|_1\frac{\|y^* - y_0\|_2}{\sqrt{T+1}}\frac{1+\gamma}{1-\gamma}. \quad (31)$$

By simple algebraic manipulations, we have that

$$\begin{aligned}\frac{1}{\sqrt{d}}\|y^*\|_2 &\leq \|y^*\|_\infty = \|(RF)^{-1}RFy^*\|_\infty \leq |(RF)^{-1}|_1\|RFy^*\|_\infty\\ &\leq \beta|(RF)^{-1}|_1\|URFy^*\|_1 = c_3\beta|(RF)^{-1}|_1 h(y^*), \quad (32)\end{aligned}$$

where $c_3 = \|URFy^*\|_1/h(y^*)$. In above, we use the property of the well-conditioned basis U.

Furthermore from inequality (31) and the equivalence condition in (23), the expected relative error bound can be upper-bounded by

$$\begin{aligned}\frac{\mathbb{E}\left[f(\bar{x})\right] - f(x^*)}{f(x^*)} &= \frac{\mathbb{E}\left[h(\bar{y})\right] - h(y^*)}{h(y^*)}\\ &\leq \frac{c_3\sqrt{d}\beta|(RF)^{-1}|_1}{\|y^*\|_2}\left(\alpha|RF|_1\frac{\|y^* - y_0\|_2}{\sqrt{T+1}}\frac{1+\gamma}{1-\gamma}\right)\\ &\leq |RF|_1|(RF)^{-1}|_1\frac{\|y^* - y_0\|_2}{\|y^*\|_2}\left(\frac{c_3\sqrt{d}\alpha\beta}{\sqrt{T+1}}\frac{1+\gamma}{1-\gamma}\right). \quad (33)\end{aligned}$$

Since the right hand side of the above inequality is a function of stopping time $T > 0$, for any arbitrarily given error bound threshold $\epsilon > 0$, by setting the right hand side to be ϵ, one obtains the following stopping condition:

$$\frac{\sqrt{d}\alpha\beta}{\sqrt{T+1}} = \frac{\epsilon}{c_1c_3\sqrt{c_2}|RF|_1|(RF)^{-1}|_1}, \quad (34)$$

where the above constants are given by

$$c_1 = \frac{1+\gamma}{1-\gamma},\ c_2 = \frac{\|x_0 - x^*\|_H^2}{\|x^*\|_H^2} = \frac{\|y_0 - y^*\|_2^2}{\|y^*\|_2^2}.$$

Rearranging the above terms we know that after

$$T \geq \frac{d\alpha^2\beta^2c_1^2c_2c_3^2}{\epsilon^2}|RF|_1^2|(RF)^{-1}|_1^2 \quad (35)$$

iterations, the relative expected error is upper-bounded by $\epsilon > 0$, i.e.,

$$\frac{\mathbb{E}\left[f(\bar{x})\right] - f(x^*)}{f(x^*)} \leq \epsilon. \quad (36)$$

This completes the proof.

C.4 Proof of Proposition 15

Similar to the proof of Proposition 14, the proof of this proposition is split into three parts: **Problem reformulation**, **Equivalence of sequences** and **Convergence rates**. From the proof of Proposition 14, one notices that the proofs in **Problem reformulation** and **Equivalence of sequences** hold for general p, and thus the proofs hold for the case when $p = 2$ as well. Now we proceed to the proof of the convergence rate. Again by the equivalence condition, we can show the convergence rate of solution vector estimate $\{x_t\}$ by showing the convergence rate achieved by the sequence $\{y_t\}$, i.e., the convergence rate of SGD of problem (21) for $p = 2$.

Throughout the rest of the proof, we denote

$$f(x) = \|Ax - b\|_2^2, \quad h(y) = \|AFy - b\|_2^2. \quad (37)$$

Denote by $H = \left(FF^\top\right)^{-1}$ the weighs of the ellipsoidal norm. Also recall that when the leverage scores satisfy the error condition in (6), we have the following condition

$$\frac{1-\gamma}{1+\gamma}\frac{\|U_i\|_2^2}{\|U\|_2^2} \leq p_i \leq \frac{1+\gamma}{1-\gamma}\frac{\|U_i\|_2^2}{\|U\|_2^2}. \quad (38)$$

Also, we assume that U is (α, β)-conditioned with $\bar{\kappa}_2(U) = \alpha\beta$. Based on Definition 1, we have

$$\begin{aligned}\alpha^2 &= \|U\|_F^2, \quad (39)\\ \beta^2 &= \|(U^\top U)^{-1}\|_2, \quad (40)\end{aligned}$$

and thus

$$\bar{\kappa}_2^2(U) = \|(U^\top U)^{-1}\|_2 \cdot \|U\|_F^2 = \alpha^2\beta^2. \quad (41)$$

Before deriving the convergence rate, we compute a few constants.

$$\mu = 2\sigma_{\min}^2(AF) = \frac{2}{\left\|((URF)^\top URF)^{-1}\right\|_2^2} \geq \frac{2}{\left\|(U^\top U)^{-1}\right\|_2 \cdot \|(RF)^{-1}\|_2^2} = \frac{2}{\beta^2 \cdot \|(RF)^{-1}\|_2^2}, \quad (42)$$

and

$$\sup_i L_i = \sup_i \frac{2\|A_iF\|_2^2}{p_i} = \sup_i \frac{2\|U_iRF\|_2^2}{p_i} \le 2c_1\|U\|_F^2 \cdot \|RF\|_2^2 = 2c_1\alpha^2 \cdot \|RF\|_2^2, \tag{43}$$

and

$$\begin{aligned}
\sigma^2 = \mathbb{E}_{i\sim\mathcal{D}}\left[\|g_i(y^*)\|^2\right] &= 4\sum_{i=1}^n (A_iFy^* - b_i)^2\|A_iF\|^2/p_i \\
&= 4\sum_{i=1}^n (U_iRFy^* - b_i)^2\|U_iRF\|^2/p_i \\
&\le 4c_1\|RF\|_2^2\|U\|_F^2\left(\sum_{i=1}^n (U_iRFy^* - b_i)^2\right) \\
&= 4c_1\|U\|_F^2 \cdot \|RF\|_2^2 \cdot h(y^*) \\
&= 4c_1\alpha^2 \cdot \|RF\|_2^2 \cdot h(y^*).
\end{aligned} \tag{44}$$

Equipped with these constant and from Proposition 18, we have the following error bound of the solution vector estimate $\{y_t\}_{t=1}^T$ generated by the weighted SGD algorithm

$$\begin{aligned}
&\mathbb{E}\left[\|x_T - x^*\|_H^2\right] \\
&= \mathbb{E}\left[\|y_T - y^*\|_2^2\right] \\
&\le \left(1 - 4\eta\sigma_{\min}^2(AF)\left(1 - \eta\sup_i \frac{2\|A_iF\|_2^2}{p_i}\right)\right)^T \|y_0 - y^*\|_2^2 \\
&\quad + \frac{2\eta\sum_{i=1}^n (A_iFy^* - b_i)^2\|A_iF\|_2^2/p_i}{\sigma_{\min}^2(AF)(1 - \eta\sup_i \frac{2\|A_iF\|_2^2}{p_i})} \\
&= \left(1 - 4\eta\sigma_{\min}^2(AF)\left(1 - \eta\sup_i \frac{2\|A_iF\|_2^2}{p_i}\right)\right)^T \|x_0 - x^*\|_H^2 \\
&\quad + \frac{2\eta\sum_{i=1}^n (A_iFy^* - b_i)^2\|A_iF\|_2^2/p_i}{\sigma_{\min}^2(AF)(1 - \eta\sup_i \frac{2\|A_iF\|_2^2}{p_i})} \\
&\le \left(\frac{1 - 4\eta\left(1 - 2\eta c_1\alpha^2\|RF\|_2^2\right)}{\beta^2\|(RF)^{-1}\|_2^2}\right)^T \|x_0 - x^*\|_H^2 + \frac{2c_1\eta\bar{\kappa}_2^2(U)\kappa^2(RF)h(y^*)}{1 - 2\eta c_1\alpha^2\|RF\|_2^2}.
\end{aligned} \tag{45}$$

Notice that the above equalities follow from the equivalence condition in (23). Combining the results from the above parts completes the proof of this lemma.

C.5 Proof of Proposition 6

Throughout the proof, we denote

$$f(x) = \|Ax - b\|_2^2, \quad h(y) = \|AFy - b\|_2^2. \tag{46}$$

Denote by $H = \left(FF^\top\right)^{-1}$ the weights of the ellipsoidal norm. Also recall the following constants defined in the statement of proposition

$$c_1 = \frac{1+\gamma}{1-\gamma},\ c_2 = \frac{\|y_0 - y^*\|_2^2}{\|y^*\|_2^2} = \frac{\|x_0 - x^*\|_H^2}{\|x^*\|_H^2},\ c_3 = \frac{\|Ax^*\|_2^2}{f(x^*)}. \tag{47}$$

Before diving into the detailed proof, we first show a useful inequality.

$$c_3h(y^*) = c_3f(x^*) = \|Ax^*\|_2^2 = \|URFy^*\|_2^2 \ge \mu\|y^*\|_2^2/2. \tag{48}$$

Now we show the first part. For an arbitrary target error $\epsilon > 0$, using (42), (43), (44) and setting

$$\frac{c_3\epsilon \cdot h(y^*)}{\|AF\|_2^2} \to \epsilon \tag{49}$$

in Corollary 19 we have that when the step-size is set to be

$$\eta = \frac{1}{4}\frac{c_3\epsilon \cdot \sigma_{\min}^2(AF) \cdot h(y^*)/\|AF\|_2^2}{\sum_{i=1}^n (A_iFy^* - b_i)^2\|A_iF\|_2^2/p_i + c_3\left(\epsilon \cdot h(y^*)/\|AF\|_2^2\right)\sigma_{\min}^2(AF)\sup_i \frac{\|A_iF\|_2^2}{p_i}}, \tag{50}$$

then after

$$\begin{aligned}
&\log\left(\frac{2\|y_0 - y^*\|_2^2}{c_3\epsilon \cdot h(y^*)/(\|U\|_2^2\|RF\|_2^2)}\right) \cdot \\
&\quad \left(c_1\alpha^2\beta^2\|RF\|_2^2\|(RF)^{-1}\|_2^2 + \frac{c_1\alpha^2\beta^4\|U\|_2^2\|RF\|_2^4\|(RF)^{-1}\|_2^4}{c_3\epsilon}\right) \\
&\le \log\left(\frac{2\|U\|_2^2\|RF\|_2^2 \cdot \|y_0 - y^*\|_2^2}{c_3\epsilon \cdot h(y^*)}\right)\left(c_1\bar{\kappa}_2^2(U)\kappa^2(RF) + \frac{c_1\bar{\kappa}_2^2(U)\kappa^2(U)\kappa^4(RF)}{c_3\epsilon}\right) \\
&\le \log\left(\frac{2c_2\kappa^2(U)\kappa^2(RF))}{\epsilon}\right)\left(c_1\bar{\kappa}_2^2(U)\kappa^2(RF)\right)\left(1 + \frac{\kappa^2(U)\kappa^2(RF)}{c_3\epsilon}\right)
\end{aligned} \tag{51}$$

iterations, the sequence $\{y_t\}_{k=1}^T$ generated by running weighted SGD algorithm satisfies the error bound

$$\|y_T - y^*\|_2^2 \le \frac{c_3\epsilon \cdot h(y^*)}{\|AF\|_2^2}. \tag{52}$$

Notice that in (51), we used (48). From this, we have

$$\begin{aligned}
\|A(x_T - x^*)\|_2^2 &= \|AFF^{-1}(x_T - x^*)\|_2^2 \\
&\le \|AF\|_2^2 \cdot \|x_T - x^*\|_H^2 \\
&= \|AF\|_2^2 \cdot \|y_T - y^*\|_2^2 \\
&= c_3\epsilon \cdot h(y^*) \\
&= \epsilon\|Ax^*\|_2^2.
\end{aligned} \tag{53}$$

For the second part, we show the result for general choice of F. The proof is basically the same as that of the first part except that we set

$$\frac{2\epsilon h(y^*)}{\|AF\|_2^2} \to \epsilon \tag{54}$$

in Corollary 19. The resulting step-size η and number of iterations required T become

$$\eta = \frac{1}{4}\frac{2\epsilon \cdot \sigma_{\min}^2(AF) \cdot h(y^*)/\|AF\|_2^2}{\sum_{i=1}^n (A_iFy^* - b_i)^2\|A_iF\|_2^2/p_i + \left(2\epsilon \cdot h(y^*)/\|AF\|_2^2\right)\sigma_{\min}^2(AF)\sup_i \frac{\|A_iF\|_2^2}{p_i}} \tag{55}$$

and

$$T = \log\left(\frac{c_2\kappa^2(U)\kappa^2(RF))}{\epsilon}\right)\left(c_1\bar{\kappa}_2^2(U)\kappa^2(RF)\right)\left(1+\frac{\kappa^2(U)\kappa^2(RF)}{2\epsilon}\right). \tag{56}$$

Setting $F = R^{-1}$ recovers the value of T shown in Proposition 6. The sequence $\{y_t\}_{k=1}^T$ generated by running weighted SGD algorithm satisfies the error bound

$$\|y_T - y^*\|_2^2 \le \frac{2\epsilon h(y^*)}{\|AF\|_2^2}. \tag{57}$$

Notice that when the problem is unconstrained, by smoothness of the objective $h(y)$, we have

$$h(y_T) - h(y^*) \le \|AF\|_2^2 \cdot \|y_T - y^*\|_2^2 \le 2\epsilon h(y^*). \tag{58}$$

Then by (23), we have

$$f(x_T) \le (1+2\epsilon)f(x^*) \le (1+2\epsilon+\epsilon^2)f(x^*). \tag{59}$$

This implies

$$\sqrt{f(x_T)} \le (1+\epsilon)\sqrt{f(x^*)}. \tag{60}$$

This completes the proof since $\sqrt{f(x)} = \|Ax-b\|_2$.

C.6 Proof of Theorem 10

Let G_f consist of m_f copies of g_f and $G = \bigcup_{f\in\mathcal{F}} G_f$. We may view the sampling step in Algorithm 2 as follows. Sample s items uniformly from G independently with replacement and denote the corresponding subset of samples by S. Then rescale every function in S by $M(\mathcal{F})/s$ and obtain $\mathcal{D}$.

By Theorem 4.1 in Feldman and Langberg (2011), we know that if the above intermediate set S is an $(\epsilon\cdot n/M(\mathcal{F}))-$approximation of the set G, then the resulting set $\mathcal{D}$ is a desired ϵ-coreset for $\mathcal{F}$. Indeed, S is such a set according to Theorem 6.10 in Feldman and Langberg (2011).

C.7 Proof of Proposition 11

We use A to denote $\bar{A}$ for the sake of simplicity. Also define the sensitivity at row index $i\in[n]$ as

$$s_i = n\cdot\sup_{x\in\mathcal{C}}\frac{|A_ix|^p}{\sum_{j=1}^n|A_jx|^p}. \tag{61}$$

Suppose $U\in\mathbb{R}^{n\times k}$ is an (α,β) well-conditioned basis of the range space of A satisfying $A=UR$, where $k=\text{rank}(A)$ and $R\in\mathbb{R}^{k\times(d+1)}$. Then from (61), we have that

$$\frac{s_i}{n} = \sup_{x\in\mathcal{C}}\frac{|A_ix|^p}{\|Ax\|_p^p} = \sup_{x\in\mathcal{C}}\frac{|U_iRx|^p}{\|URx\|_p^p} = \sup_{y\in\mathcal{C}'}\frac{|U_iy|^p}{\|Uy\|_p^p} \le \sup_{y\in\mathcal{C}'}\frac{\|U_i\|_p^p\|y\|_q^p}{\|y\|_q^p/\beta^p} = \beta^p\|U_i\|_p^p = \beta^p\cdot\lambda_i, \tag{62}$$

where $\mathcal{C}' = \{y\in\mathbb{R}^d | y = Rx, x\in\mathcal{C}\}$ is a one-to-one mapping. The first inequality follows from Hölder's inequality with $\frac{1}{p}+\frac{1}{q}=1$ and the properties of well-conditioned bases. According to the definition of sensitivity $m(f_i) = \lfloor s_i\rfloor + 1$, the above property implies

$$m(f_i) \le n\beta^p\lambda_i + 1. \tag{63}$$

which implies $M(\mathcal{F}) = \sum_{i=1}^n s_i \le (n\beta^p\sum_{i=1}^n\lambda_i) + n = n((\alpha\beta)^p+1)$, and completes the proof.

C.8 Proof of Proposition 12

According to Definition 9, we only have to show that for any arbitrary constant n and set of points $G=\{a_1,\ldots,a_n\}\subseteq\mathbb{R}^d$, the following condition holds:

$$|\{\textbf{Range}(G,x,r)|x\in\mathcal{X}, r\ge 0\}| \le n^{d+1},$$

where $\textbf{Range}(G,x,r) = \{a_i||a_i^\top x|^p\le r\}$ is the region located in the $p-$norm ellipsoid $|a_i^\top x|^p = r$. Since the following condition holds: $\{a_i||a_i^\top x|^p\le r\} = \{a_i||a_i^\top x|\le r^{\frac{1}{p}}\}$ and the constant r is non-negative and arbitrary. Without loss of generality, we assume $p=1$ in the above definition, i.e., $\textbf{Range}(G,x,r) = \{a_i||a_i^\top x|\le r\}$.

Notice that for every x and r, $\textbf{Range}(G,x,r)$ is a subset of G. Hence, we may view it as a binary classifier on G, denoted by $c_{x,r}$. Given $x\in\mathcal{X}$ and $r\ge 0$, for any $a_i\in G$ we have that

$$c_{x,r}(a_i) = \begin{cases}1, & \text{if } |a_i^\top x|\le r;\\ 0, & \text{otherwise.}\end{cases}$$

Therefore, one immediately sees that $|\{\textbf{Range}(G,x,r)|x\in\mathcal{X},r\ge0\}|$ is the shattering coefficient of $C := \{c_{x,r}|x\in\mathcal{X},r\ge0\}$ on n points, denoted by $s(C,n)$. To bound the shattering coefficient of C, we provide an upper bound based on its VC dimension.

We claim that the VC dimension of C is at most $d+1$. By contradiction, suppose there exists $n+2$ points such that any labeling on these $n+2$ points can be shattered by C. By Radon's Theorem (Clarkson et al., 1993), we can partition these points into two disjoint subsets, namely, V and W with size n_1 and n_2 respectively, where the intersection of their convex hulls is nonempty. Let b be a point located in the intersection of the convex hulls of V and W, which in general can be written as

$$b = \sum_{i=1}^{n_1}\lambda_iv_i = \sum_{i=1}^{n_2}\sigma_iw_i, \tag{64}$$

where $\lambda_i\ge0$, $\sigma_i\ge0$ and $\sum_{i=1}^{n_1}\lambda_i = \sum_{i=1}^{n_2}\sigma_i = 1$.

By the above assumption, we can find vector $x\in\mathbb{R}^n$ and nonnegative constant r such that the following conditions hold:

$$-r\le x^\top v_i\le r,\quad i=1,\ldots,n_1; \tag{65}$$

$$x^\top w_i > r \text{ or } x^\top w_i < -r,\quad i=1,\ldots,n_2. \tag{66}$$

By combining the conditions in (64), (65) and (66), we further obtain both inequalities

$$-r\le b^\top x\le r, \tag{67}$$

and

$$b^\top x < -r \text{ or } b^\top x > r, \tag{68}$$

which is clearly paradoxical! This concludes that the VC dimension of C is less than or equal to $d+1$. Furthermore, by Sauer's Lemma (Sauer), for $n\ge2$ the shattering coefficient $s(C,n) = |\{\textbf{Range}(G,x,r)|x\in\mathcal{X},r\ge0\}|$ is less than n^{d+1}, which completes the proof of this proposition.

C.9 Proof of Proposition 13

Without loss of generality, assume the low dimension d is even (because if d is odd, we can always add an extra arbitrary row to input matrix A and upper bound the size of the original total sensitivity set by the same analysis). Let $a_i \in [0,1]^d$ be a vector with exactly $d/2$ elements to be 1. For each $i \in [n]$, let $B_i = \{j | a_{ij} = 1\}$, where a_{ij} denotes the j-th element of vector a_i. For fixed i, define x as follows,

$$x_j = \begin{cases} 2/d, & \text{if } \ j \in B_i, \\ -d, & \text{otherwise.} \end{cases} \tag{69}$$

One immediately notices from the above expression that $x^\top a_i = 1$. Thus for $j \neq i$, $a_j \neq a_i$, there exists an index $k \in [d]$ such that $a_{jk} = 1$ but $a_{ik} = 0$. Furthermore the above condition implies

$$x^\top a_j = \sum_{l=1}^{d} x_l a_{jl} = \sum_{l \in B_j, l \neq k}^{d} x_l a_{jl} + \sum_{l \neq B_j}^{d} x_l a_{jl} + x_k a_{jk} \leq (d/2-1)(2/d) - d < 0, \tag{70}$$

which further implies $f_j(x) = x^\top a_j = 0$; Therefore, the i-th sensitivity becomes

$$s_i = \sup_x \frac{f_i(x)}{\sum_{i=j}^{n} f_j(x)} \geq 1. \tag{71}$$

Since the above condition holds for arbitrary index $i \in [n]$, and we have $\binom{d}{d/2}$ number of vectors a_i, i.e., $n = \binom{d}{d/2}$, this concludes that the size of the total sensitivity set is at least $\binom{d}{d/2} \approx 2^d$.

Appendix D. Stochastic Gradient Descent

Consider minimizing the following objective

$$\text{minimize}_{x \in \mathcal{X}} f(x) = \mathbb{E}_{i \sim P}\left[f_i(x)\right]. \tag{72}$$

Stochastic gradient descent (SGD) exploits the following update rule

$$x_{t+1} = \arg\min_{x \in \mathcal{X}} \eta \langle x - x_t, g_{\xi_t}(x_t) \rangle + \frac{1}{2}\|x - x_t\|_2^2, \tag{73}$$

where $\xi_t \in [n]$ is an index drawn according to P, $g_{\xi_t}(x) = \nabla f_{\xi_t}(x)$ and $\mathbb{E}_{\xi_t \sim P}\left[f_{\xi_t}(x)\right] = f(x)$. When $\mathcal{X} = \mathbb{R}^d$, the update rule (73) boils down to $x_{t+1} = x_t - \eta g_{\xi_t}(x_t)$. Note here, if $f_{\xi_t}(x)$ is not differentiable, we take $g_{\xi_t}(x_t)$ to be one of its sub-gradients, i.e., $g_{\xi_t}(x_t) \in \partial f_{\xi_t}(x_t)$. In this case, SGD boils down to stochastic sub-gradient method. For simplicity, we still refer to the algorithms as SGD.

In the following, we present two results regarding the convergence rate of SGD on problem with non-strongly convex objective and strongly convex objective, respectively.

D.1 Non-strongly convex case

Here we analyze the case where the objective function $f(x)$ is not strongly convex. Also, each sub-function is not necessary differentiable. That is, $g_i(x)$ can be a sub-gradient of function f_i at x.

Proposition 17 *Assume that $\frac{1}{2}\|\cdot\|_2^2 \geq \frac{\lambda}{2}\|\cdot\|^2$ for some norm $\|\cdot\|^2$. Also assume that $\|g_t(x_t)\|_* \leq M$ for any $t > 0$ where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$. The output $\bar{x} = \frac{1}{T+1}\sum_{t=1}^{T} x_t$ of SGD satisfies, for any $y \in \mathcal{X}$,*

$$\mathbb{E}\left[f(\bar{x})\right] - f(y) \leq \frac{\|y - x_0\|_2^2}{2\eta(T+1)} + \frac{\eta}{2\lambda}M^2. \tag{74}$$

In particular, when $\eta = \frac{\|y-x_0\|_2}{M}\sqrt{\frac{\lambda}{T+1}}$, we have

$$\mathbb{E}\left[f(\bar{x})\right] - f(y) \leq M\|y - x_0\|_2\sqrt{\frac{1}{(T+1)\lambda}}. \tag{75}$$

Proof From Lemma 1 in Duchi et al. (2010), at step t, we have that

$$\eta(f_t(x_t) - f_t(y)) \leq \frac{1}{2}\|y - x_t\|_2^2 - \frac{1}{2}\|y - x_{t+1}\|_2^2 + \frac{\eta^2}{2\lambda}\|g_t(x_t)\|_*^2. \tag{76}$$

Conditioned on x_t, taking the conditional expectation with respect to ξ_t on both sides, we have

$$\mathbb{E}\left[\eta(f_t(x_t) - f_t(y))|x_t\right] \leq \mathbb{E}\left[\frac{1}{2}\|y - x_t\|_2^2 - \frac{1}{2}\|y - x_{t+1}\|_2^2 + \frac{\eta^2}{2\lambda}\|g_t(x_t)\|_*^2|x_t\right]. \tag{77}$$

Noticing that $\mathbb{E}_{\xi_t \sim P}\left[f_t(x)\right] = f(x)$, we have

$$\eta f(x_t) - \eta f(y) \leq \frac{1}{2}\|y - x_t\|_2^2 + \mathbb{E}\left[-\frac{1}{2}\|y - x_{t+1}\|_2^2 + \frac{\eta^2}{2\lambda}\|g_t(x_t)\|_*^2|x_t\right]. \tag{78}$$

Then by taking the expectation over x_t and using the fact that $\|g_t(x_t)\|_* \leq M$, we have

$$\mathbb{E}\left[\eta f(x_t)\right] - \eta f(y) \leq \mathbb{E}\left[\frac{1}{2}\|y - x_t\|_2^2\right] - \mathbb{E}\left[\frac{1}{2}\|y - x_{t+1}\|_2^2\right] + \frac{\eta^2}{2\lambda}M^2. \tag{79}$$

Summing up the above equation with $t = 0, \ldots, T$ and noticing $\|y - x_{t+1}\|_2^2 \geq 0$, we have

$$\eta\sum_{t=0}^{T}\mathbb{E}\left[f(x_t)\right] - \eta(T+1)f(y) \leq \frac{1}{2}\|y - x_0\|_2^2 + \frac{\eta^2(T+1)}{2\lambda}M^2. \tag{80}$$

Finally by convexity of f, we have that

$$\mathbb{E}\left[f(\bar{x})\right] - f(y) \leq \frac{\|y - x_0\|_2^2}{2\eta(T+1)} + \frac{\eta}{2\lambda}M^2. \tag{81}$$

In particular with $\eta = \frac{\|y-y_1\|_2}{M}\sqrt{\frac{\lambda}{T+1}}$, we have

$$\mathbb{E}\left[f(\bar{x})\right] - f(y) \leq M\|y - x_0\|_2\sqrt{\frac{1}{(T+1)\lambda}}, \tag{82}$$

which completes the proof. ■

D.2 Strongly convex case

Here we analyze the case where the objective function $f(x)$ is strongly convex. We make the following two assumptions:

(A1) Function $f(x)$ is strongly convex with modulus μ. That is, for any $x, y \in \mathcal{X}$,

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y - x\|_2^2. \tag{83}$$

(A2) For each $i \in [n]$, the gradient of each sub-function $\nabla f_i(x)$ is Lipschitz continuous with constant L_i. That is, for any $x, y \in \mathcal{X}$,

$$\|\nabla f_i(y) - \nabla f_i(x)\|_2 \leq L_i \|y - x\|_2. \tag{84}$$

The following results also appeared in Needell et al. (2014).

Proposition 18 *Under assumption (A1), (A2), the sequence $\{x_t\}$ generated by SGD satisfies*

$$\mathbb{E}\left[\|x_T - x^*\|_2^2\right] \leq (1 - 2\eta\mu(1 - \eta \sup L_i))^T \|x_0 - x^*\|_2^2 + \frac{\eta\sigma^2}{\mu(1 - \eta \sup L_i)}, \tag{85}$$

where $\sigma^2 = \mathbb{E}_{i\sim\mathcal{D}}\left[\|\nabla f_i(x^*)\|_2^2\right]$ and x^* is the optimal solution to (72).

Proof The proof essentially follows the same lines of arguments as in Needell et al. (2014). The only difference is that, here we are working on the constrained problem where update rule (73) is equivalent to

$$x_{t+1} = \Pi_{\mathcal{X}}(x_t - \eta g_t(x_t)). \tag{86}$$

Notice that $\Pi_{\mathcal{X}}(x)$ is a projection operator to the feasible set $\mathcal{X}$ and it is non-expansive. This further implies

$$\|x_{t+1} - x^*\|_2^2 = \|\Pi_{\mathcal{X}}(x_t - \eta g_t(x_t)) - x^*\|_2^2 \leq \|x_t - \eta g_t(x_t) - x^*\|_2^2. \tag{87}$$

The rest of the proof follows analogous arguments in Needell et al. (2014). ■

Corollary 19 *Given a target accuracy $\epsilon > 0$, and let the step-size be $\eta = \frac{\epsilon\mu}{2\sigma^2 + 2\epsilon\mu \sup L_i}$. Then after*

$$T \geq \log\left(\frac{2\|x_0 - x^*\|^2}{\epsilon}\right)\left(\frac{\sigma^2}{\epsilon\mu^2} + \frac{\sup L_i}{\mu}\right) \tag{88}$$

iterations, we have that

$$\mathbb{E}\left[\|x_T - x^*\|_2^2\right] \leq \epsilon. \tag{89}$$

Proof The proof can be found in Needell et al. (2014). ■

Journal of Machine Learning Research 18 (2018) 1-54 Submitted 12/17; Revised 12/17; Published 4/18

Catalyst Acceleration for First-order Convex Optimization: from Theory to Practice

Hongzhou Lin HONGZHOU@MIT.EDU
Massachusetts Institute of Technology
Computer Science and Artificial Intelligence Laboratory
Cambridge, MA 02139, USA

Julien Mairal JULIEN.MAIRAL@INRIA.FR
Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK,*
Grenoble, 38000, France

Zaid Harchaoui ZAID@UW.EDU
University of Washington
Department of Statistics
Seattle, WA 98195, USA

Editor: Léon Bottou

Abstract

We introduce a generic scheme for accelerating gradient-based optimization methods in the sense of Nesterov. The approach, called Catalyst, builds upon the inexact accelerated proximal point algorithm for minimizing a convex objective function, and consists of approximately solving a sequence of well-chosen auxiliary problems, leading to faster convergence. One of the keys to achieve acceleration in theory and in practice is to solve these sub-problems with appropriate accuracy by using the right stopping criterion and the right warm-start strategy. We give practical guidelines to use Catalyst and present a comprehensive analysis of its global complexity. We show that Catalyst applies to a large class of algorithms, including gradient descent, block coordinate descent, incremental algorithms such as SAG, SAGA, SDCA, SVRG, MISO/Finito, and their proximal variants. For all of these methods, we establish faster rates using the Catalyst acceleration, for strongly convex and non-strongly convex objectives. We conclude with extensive experiments showing that acceleration is useful in practice, especially for ill-conditioned problems.

Keywords: convex optimization, first-order methods, large-scale machine learning

1. Introduction

A large number of machine learning and signal processing problems are formulated as the minimization of a convex objective function:

$$\min_{x\in\mathbb{R}^p} \left\{ f(x) \triangleq f_0(x) + \psi(x) \right\}, \tag{1}$$

where f_0 is convex and L-smooth, and ψ is convex but may not be differentiable. We call a function L-smooth when it is differentiable and its gradient is L-Lipschitz continuous.

∗. Institute of Engineering Univ. Grenoble Alpes

In statistics or machine learning, the variable x may represent model parameters, and the role of f_0 is to ensure that the estimated parameters fit some observed data. Specifically, f_0 is often a large sum of functions and (1) is a regularized empirical risk which writes as

$$\min_{x\in\mathbb{R}^p} \left\{ f(x) \triangleq \frac{1}{n}\sum_{i=1}^n f_i(x) + \psi(x) \right\}. \tag{2}$$

Each term $f_i(x)$ measures the fit between x and a data point indexed by i, whereas the function ψ acts as a regularizer; it is typically chosen to be the squared ℓ_2-norm, which is smooth, or to be a non-differentiable penalty such as the ℓ_1-norm or another sparsity-inducing norm (Bach et al., 2012).

We present a unified framework allowing one to accelerate gradient-based or first-order methods, with a particular focus on problems involving large sums of functions. By "accelerating", we mean generalizing a mechanism invented by Nesterov (1983) that improves the convergence rate of the gradient descent algorithm. When $\psi = 0$, gradient descent steps produce iterates $(x_k)_{k\geq 0}$ such that $f(x_k) - f^* \leq \varepsilon$ in $O(1/\varepsilon)$ iterations, where f^* denotes the minimum value of f. Furthermore, when the objective f is μ-strongly convex, the previous iteration-complexity becomes $O((L/\mu)\log(1/\varepsilon))$, which is proportional to the condition number L/μ. However, these rates were shown to be suboptimal for the class of first-order methods, and a simple strategy of taking the gradient step at a well-chosen point different from x_k yields the optimal complexity—$O(1/\sqrt{\varepsilon})$ for the convex case and $O(\sqrt{L/\mu}\log(1/\varepsilon))$ for the μ-strongly convex one (Nesterov, 1983). Later, this acceleration technique was extended to deal with non-differentiable penalties ψ for which the proximal operator defined below is easy to compute (Beck and Teboulle, 2009; Nesterov, 2013).

$$\mathrm{prox}_{\psi}(x) \triangleq \arg\min_{z\in\mathbb{R}^p} \left\{ \psi(z) + \frac{1}{2}\|x - z\|^2 \right\}, \tag{3}$$

where $\|.\|$ denotes the Euclidean norm.

For machine learning problems involving a large sum of n functions, a recent effort has been devoted to developing fast incremental algorithms such as SAG (Schmidt et al., 2017), SAGA (Defazio et al., 2014a), SDCA (Shalev-Shwartz and Zhang, 2012), SVRG (Johnson and Zhang, 2013; Xiao and Zhang, 2014), or MISO/Finito (Mairal, 2015; Defazio et al., 2014b), which can exploit the particular structure (2). Unlike full gradient approaches, which require computing and averaging n gradients $(1/n)\sum_{i=1}^n \nabla f_i(x)$ at every iteration, incremental techniques have a cost per-iteration that is independent of n. The price to pay is the need to store a moderate amount of information regarding past iterates, but the benefits may be significant in terms of computational complexity. In order to achieve an ε-accurate solution for a μ-strongly convex objective, the number of gradient evaluations required by the methods mentioned above is bounded by $O\left(\left(n + \frac{\bar{L}}{\mu}\right)\log(\frac{1}{\varepsilon})\right)$, where $\bar{L}$ is either the maximum Lipschitz constant across the gradients ∇f_i, or the average value, depending on the algorithm variant considered. Unless there is a big mismatch between $\bar{L}$ and L (global Lipschitz constant for the sum of gradients), incremental approaches significantly outperform the full gradient method, whose complexity in terms of gradient evaluations is bounded by $O\left(n\frac{L}{\mu}\log(\frac{1}{\varepsilon})\right)$.

Yet, these incremental approaches do not use Nesterov's extrapolation steps and whether or not they could be accelerated was an important open question when these methods were introduced. It was indeed only known to be the case for SDCA (Shalev-Shwartz and Zhang, 2016) for strongly convex objectives. Later, other accelerated incremental algorithms were proposed such as Katyusha (Allen-Zhu, 2017), or the method of Lan and Zhou (2017).

We give here a positive answer to this open question. By analogy with substances that increase chemical reaction rates, we call our approach "Catalyst". Given an optimization method $\mathcal{M}$ as input, Catalyst outputs an accelerated version of it, eventually the same algorithm if the method $\mathcal{M}$ is already optimal. The sole requirement on the method in order to achieve acceleration is that it should have linear convergence rate for strongly convex problems. This is the case for full gradient methods (Beck and Teboulle, 2009; Nesterov, 2013) and block coordinate descent methods (Nesterov, 2012; Richtárik and Takáč, 2014), which already have well-known accelerated variants. More importantly, it also applies to the previou incremental methods, whose complexity is then bounded by $\tilde{O}\left(\left(n+\sqrt{n\bar{L}/\mu}\right)\log(\frac{1}{\varepsilon})\right)$ after Catalyst acceleration, where $\tilde{O}$ hides some logarithmic dependencies on the condition number $\bar{L}/\mu$. This improves upon the non-accelerated variants, when the condition number is larger than n. Besides, acceleration occurs regardless of the strong convexity of the objective—that is, even if $\mu = 0$—which brings us to our second achievement.

Some approaches such as MISO, SDCA, or SVRG are only defined for strongly convex objectives. A classical trick to apply them to general convex functions is to add a small regularization term $\varepsilon\|x\|^2$ in the objective (Shalev-Shwartz and Zhang, 2012). The drawback of this strategy is that it requires choosing in advance the parameter ε, which is related to the target accuracy. The approach we present here provides a *direct support for non-strongly convex objectives*, thus removing the need of selecting ε beforehand. Moreover, we can immediately establish a faster rate for the resulting algorithm. Finally, some methods such as MISO are numerically unstable when they are applied to strongly convex objective functions with small strong convexity constant. By defining better conditioned auxiliary subproblems, Catalyst also provides better numerical stability to these methods.

A short version of this paper has been published at the NIPS conference in 2015 (Lin et al., 2015a); in addition to simpler convergence proofs and more extensive numerical evaluation, we extend the conference paper with a new Moreau-Yosida smoothing interpretation with significant theoretical and practical consequences as well as new practical stopping criteria and warm-start strategies.

The paper is structured as follows. We complete this introductory section with some related work in Section 1.1, and give a short description of the two-loop Catalyst algorithm in Section 1.2. Then, Section 2 introduces the Moreau-Yosida smoothing and its inexact variant. In Section 3, we introduce formally the main algorithm, and its convergence analysis is presented in Section 4. Section 5 is devoted to numerical experiments and Section 6 concludes the paper.

1.1 Related Work

Catalyst can be interpreted as a variant of the proximal point algorithm (Rockafellar, 1976; Güler, 1991), which is a central concept in convex optimization, underlying augmented Lagrangian approaches, and composite minimization schemes (Bertsekas, 2015; Parikh and

Boyd, 2014). The proximal point algorithm consists of solving (1) by minimizing a sequence of auxiliary problems involving a quadratic regularization term. In general, these auxiliary problems cannot be solved with perfect accuracy, and several notions of inexactness were proposed by Güler (1992); He and Yuan (2012) and Salzo and Villa (2012). The Catalyst approach hinges upon (i) an acceleration technique for the proximal point algorithm originally introduced in the pioneer work of Güler (1992); (ii) a more practical inexactness criterion than those proposed in the past.[1] As a result, we are able to control the rate of convergence for approximately solving the auxiliary problems with an optimization method $\mathcal{M}$. In turn, we are also able to obtain the computational complexity of the global procedure, which was not possible with previous analysis (Güler, 1992; He and Yuan, 2012; Salzo and Villa, 2012). When instantiated in different first-order optimization settings, our analysis yields systematic acceleration.

Beyond Güler (1992), several works have inspired this work. In particular, accelerated SDCA (Shalev-Shwartz and Zhang, 2016) is an instance of an inexact accelerated proximal point algorithm, even though this was not explicitly stated in the original paper. Catalyst can be seen as a generalization of their algorithm, originally designed for stochastic dual coordinate ascent approaches. Yet their proof of convergence relies on different tools than ours. Specifically, we introduce an approximate sufficient descent condition, which, when satisfied, grants acceleration to any optimization method, whereas the direct proof of Shalev-Shwartz and Zhang (2016), in the context of SDCA, does not extend to non-strongly convex objectives. Another useful methodological contribution was the convergence analysis of inexact proximal gradient methods of Schmidt et al. (2011) and Devolder et al. (2014). Finally, similar ideas appeared in the independent work (Frostig et al., 2015). Their results partially overlap with ours, but the two papers adopt rather different directions. Our analysis is more general, covering both strongly-convex and non-strongly convex objectives, and comprises several variants including an almost parameter-free variant.

Then, beyond accelerated SDCA (Shalev-Shwartz and Zhang, 2016), other accelerated incremental methods have been proposed, such as APCG (Lin et al., 2015b), SDPC (Zhang and Xiao, 2015), RPDG (Lan and Zhou, 2017), Point-SAGA (Defazio, 2016) and Katyusha (Allen-Zhu, 2017). Their techniques are algorithm-specific and cannot be directly generalized into a unified scheme. However, we should mention that the complexity obtained by applying Catalyst acceleration to incremental methods matches the optimal bound up to a logarithmic factor, which may be the price to pay for a generic acceleration scheme.

A related recent line of work has also combined smoothing techniques with outer-loop algorithms such as Quasi-Newton methods (Themelis et al., 2016; Giselsson and Fält, 2016). Their purpose was not to accelerate existing techniques, but rather to derive new algorithms for nonsmooth optimization.

To conclude this survey, we mention the broad family of extrapolation methods (Sidi, 2017), which allow one to extrapolate to the limit sequences generated by iterative algorithms for various numerical analysis problems. Scieur et al. (2016) proposed such an approach for convex optimization problems with smooth and strongly convex objectives. The approach we present here allows us to obtain global complexity bounds for strongly

1. Note that our inexact criterion was also studied, among others, by Salzo and Villa (2012), but their analysis led to the conjecture that this criterion was too weak to warrant acceleration. Our analysis refutes this conjecture.

Algorithm 1 Catalyst - Overview

input initial estimate x_0 in $\mathbb{R}^p$, smoothing parameter κ, optimization method $\mathcal{M}$.

1: Initialize $y_0 = x_0$.

2: **while** the desired accuracy is not achieved **do**

3: Find x_k using $\mathcal{M}$

$$x_k \approx \operatorname*{arg\,min}_{x \in \mathbb{R}^p} \left\{ h_k(x) \triangleq f(x) + \frac{\kappa}{2}\|x - y_{k-1}\|^2 \right\}. \tag{4}$$

4: Compute y_k using an extrapolation step, with β_k in $(0, 1)$

$$y_k = x_k + \beta_k(x_k - x_{k-1}).$$

5: **end while**

output x_k (final estimate).

convex and non strongly convex objectives, which can be decomposed into a smooth part and a non-smooth proximal-friendly part.

1.2 Overview of Catalyst

Before introducing Catalyst precisely in Section 3, we give a quick overview of the algorithm and its main ideas. Catalyst is a generic approach that wraps an algorithm $\mathcal{M}$ into an accelerated one $\mathcal{A}$, in order to achieve the same accuracy as $\mathcal{M}$ with reduced computational complexity. The resulting method $\mathcal{A}$ is an inner-outer loop construct, presented in Algorithm 1, where in the *inner loop* the method $\mathcal{M}$ is called to solve an auxiliary strongly-convex optimization problem, and where in the *outer loop* the sequence of iterates produced by $\mathcal{M}$ are *extrapolated* for faster convergence. There are therefore three main ingredients in Catalyst: a) a smoothing technique that produces strongly-convex sub-problems; b) an extrapolation technique to accelerate the convergence; c) a balancing principle to optimally tune the inner and outer computations.

Smoothing by infimal convolution Catalyst can be used on any algorithm $\mathcal{M}$ that enjoys a linear-convergence guarantee when minimizing strongly-convex objectives. However the objective at hand may be poorly conditioned or even might not be strongly convex. In Catalyst, we use $\mathcal{M}$ to *approximately minimize* an auxiliary objective h_k at iteration k, defined in (4), which is strongly convex and better conditioned than f. Smoothing by infimal convolution allows one to build a well-conditioned convex function F from a poorly-conditioned convex function f (see Section 3 for a refresher on Moreau envelopes). We shall show in Section 3 that a notion of *approximate Moreau envelope* allows us to define precisely the information collected when approximately minimizing the auxiliary objective.

Extrapolation by Nesterov acceleration Catalyst uses an extrapolation scheme " à la Nesterov " to build a sequence $(y_k)_{k\geq 0}$ updated as

$$y_k = x_k + \beta_k(x_k - x_{k-1}) \ ,$$

where $(\beta_k)_{k\geq 0}$ is a positive decreasing sequence, which we shall define in Section 3.

	Without Catalyst		With Catalyst	
	$\mu > 0$	$\mu = 0$	$\mu > 0$	$\mu = 0$
FG	$O\left(n\frac{L}{\mu}\log\left(\frac{1}{\varepsilon}\right)\right)$	$O\left(n\frac{L}{\varepsilon}\right)$	$\tilde{O}\left(n\sqrt{\frac{L}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right)$	$\tilde{O}\left(n\sqrt{\frac{L}{\varepsilon}}\right)$
SAG/SAGA	$O\left(\left(n+\frac{\bar{L}}{\mu}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$	$O\left(n\frac{\bar{L}}{\varepsilon}\right)$	$\tilde{O}\left(\left(n+\sqrt{\frac{n\bar{L}}{\mu}}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$	$\tilde{O}\left(\sqrt{\frac{n\bar{L}}{\varepsilon}}\right)$
MISO		not avail.		
SDCA				
SVRG				
Acc-FG	$O\left(n\sqrt{\frac{L}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right)$	$O\left(n\frac{L}{\sqrt{\varepsilon}}\right)$	no acceleration	
Acc-SDCA	$\tilde{O}\left(\left(n+\sqrt{\frac{n\bar{L}}{\mu}}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$	not avail.		

Table 1: Comparison of rates of convergence, before and after the Catalyst acceleration, in the strongly-convex and non strongly-convex cases, respectively. The notation $\tilde{O}$ hides logarithmic factors. The constant L is the global Lipschitz constant of the gradient's objective, while $\bar{L}$ is the average Lipschitz constants of the gradients ∇f_i, or the maximum value, depending on the algorithm's variants considered.

We shall show in Section 4 that we can get faster rates of convergence thanks to this extrapolation step when the smoothing parameter κ, the inner-loop stopping criterion, and the sequence $(\beta_k)_{k\geq 0}$ are carefully built.

Balancing inner and outer complexities The optimal balance between inner loop and outer loop complexity derives from the complexity bounds established in Section 4. Given an estimate about the condition number of f, our bounds dictate a choice of κ that gives the optimal setting for the inner-loop stopping criterion and all technical quantities involved in the algorithm. We shall demonstrate in particular the power of an appropriate warm-start strategy to achieve near-optimal complexity.

Overview of the complexity results Finally, we provide in Table 1 a brief overview of the complexity results obtained from the Catalyst acceleration, when applied to various optimization methods $\mathcal{M}$ for minimizing a large finite sum of n functions. Note that the complexity results obtained with Catalyst are optimal, up to some logarithmic factors (see Agarwal and Bottou, 2015; Arjevani and Shamir, 2016; Woodworth and Srebro, 2016).

2. The Moreau Envelope and its Approximate Variant

In this section, we recall a classical tool from convex analysis called the Moreau envelope or Moreau-Yosida smoothing (Moreau, 1962; Yosida, 1980), which plays a key role for understanding the Catalyst acceleration. This tool can be seen as a smoothing technique,

which can turn any convex lower semicontinuous function f into a smooth function, and an ill-conditioned smooth convex function into a well-conditioned smooth convex function.

The Moreau envelope results from the infimal convolution of f with a quadratic penalty:

$$F(x) \triangleq \min_{z\in\mathbb{R}^p} \left\{ f(z) + \frac{\kappa}{2}\|z-x\|^2 \right\}, \tag{5}$$

where κ is a positive regularization parameter. The proximal operator is then the unique minimizer of the problem—that is,

$$p(x) \triangleq \operatorname{prox}_{f/\kappa}(x) = \operatorname*{arg\,min}_{z\in\mathbb{R}^p} \left\{ f(z) + \frac{\kappa}{2}\|z-x\|^2 \right\}.$$

Note that $p(x)$ does not admit a closed form in general. Therefore, computing it requires to solve the sub-problem to high accuracy with some iterative algorithm.

2.1 Basic Properties of the Moreau Envelope

The smoothing effect of the Moreau regularization can be characterized by the next proposition (see Lemaréchal and Sagastizábal, 1997, for elementary proofs).

Proposition 1 (Regularization properties of the Moreau Envelope) *Given a convex continuous function f and a regularization parameter $\kappa > 0$, consider the Moreau envelope F defined in (5). Then,*

1. *F is convex and minimizing f and F are equivalent in the sense that*

$$\min_{x\in\mathbb{R}^p} F(x) = \min_{x\in\mathbb{R}^p} f(x)\ .$$

   Moreover the solution set of the two above problems coincide with each other.

2. *F is continuously differentiable even when f is not and*

$$\nabla F(x) = \kappa(x - p(x))\ . \tag{6}$$

   Moreover the gradient ∇F is Lipschitz continuous with constant $L_F = \kappa$.

3. *If f is μ-strongly convex, then F is μ_F-strongly convex with $\mu_F = \frac{\mu\kappa}{\mu+\kappa}$.*

Interestingly, F is friendly from an optimization point of view as it is convex and differentiable. Besides, F is κ-smooth with condition number $\frac{\mu+\kappa}{\mu}$ when f is μ-strongly convex. Thus F can be made arbitrarily well conditioned by choosing a small κ. Since both functions f and F admit the same solutions, a naive approach to minimize a non-smooth function f is to first construct its Moreau envelope F and then apply a smooth optimization method on it. As we will see next, Catalyst can be seen as an accelerated gradient descent technique applied to F with inexact gradients.

2.2 A Fresh Look at Catalyst

First-order methods applied to F provide us several well-known algorithms.

The proximal point algorithm. Consider gradient descent steps on F:

$$x_{k+1} = x_k - \frac{1}{L_F}\nabla F(x_k).$$

By noticing that $\nabla F(x_k) = \kappa(x_k - p(x_k))$ and $L_f = \kappa$, we obtain in fact

$$x_{k+1} = p(x_k) = \operatorname*{arg\,min}_{z\in\mathbb{R}^p} \left\{ f(z) + \frac{\kappa}{2}\|z-x_k\|^2 \right\},$$

which is exactly the proximal point algorithm (Martinet, 1970; Rockafellar, 1976).

Accelerated proximal point algorithm. If gradient descent steps on F yields the proximal point algorithm, it is then natural to consider the following sequence

$$x_{k+1} = y_k - \frac{1}{L_F}\nabla F(y_k) \quad \text{and} \quad y_{k+1} = x_{k+1} + \beta_{k+1}(x_{k+1} - x_k),$$

where β_{k+1} is Nesterov's extrapolation parameter (Nesterov, 2004). Again, by using the closed form of the gradient, this is equivalent to the update

$$x_{k+1} = p(y_k) \quad \text{and} \quad y_{k+1} = x_{k+1} + \beta_{k+1}(x_{k+1} - x_k),$$

which is known as the accelerated proximal point algorithm of Güler (1992).

While these algorithms are conceptually elegant, they suffer from a major drawback in practice: each update requires to evaluate the proximal operator $p(x)$. Unless a closed form is available, which is almost never the case, we are not able to evaluate $p(x)$ exactly. Hence an iterative algorithm is required for each evaluation of the proximal operator which leads to the inner-outer construction (see Algorithm 1). Catalyst can then be interpreted as an accelerated proximal point algorithm that calls an optimization method $\mathcal{M}$ to compute inexact solutions to the sub-problems. The fact that such a strategy could be used to solve non-smooth optimization problems was well-known, but the fact that it could be used for acceleration is more surprising. The main challenge that will be addressed in Section 3 is how to control the complexity of the inner-loop minimization.

2.3 The Approximate Moreau Envelope

Since Catalyst uses inexact gradients of the Moreau envelope, we start with specifying the inexactness criteria.

Inexactness through absolute accuracy. Given a proximal center x, a smoothing parameter κ, and an accuracy $\varepsilon > 0$, we denote the set of ε-approximations of the proximal operator $p(x)$ by

$$p^\varepsilon(x) \triangleq \{z \in \mathbb{R}^p \quad \text{s.t.} \quad h(z) - h^* \le \varepsilon\} \quad \text{where} \quad h(z) = f(z) + \frac{\kappa}{2}\|x-z\|^2, \tag{C1}$$

and h^* is the minimum function value of h.

Checking whether $h(z) - h^* \le \varepsilon$ may be impactical since h^* is unknown in many situations. We may then replace h^* by a lower bound that can be computed more easily. We may use the Fenchel conjugate for instance. Then, given a point z and a lower-bound

$d(z) \leq h^*$, we can guarantee $z \in p^\varepsilon(x)$ if $h(z) - d(z) \leq \varepsilon$. There are other choices for the lower bounding function d which result from the specific construction of the optimization algorithm. For instance, dual type algorithms such as SDCA (Shalev-Shwartz and Zhang, 2012) or MISO (Mairal, 2015) maintain a lower bound along the iterations, allowing one to compute $h(z) - d(z) \leq \varepsilon$.

When none of the options mentioned above are available, we can use the following fact, based on the notion of gradient mapping; see Section 2.3.2 of (Nesterov, 2004). The intuition comes from the smooth case: when h is smooth, the strong convexity yields

$$h(z) - \frac{1}{2\kappa}\|\nabla h(z)\|^2 \leq h^*.$$

In other words, the norm of the gradient provides enough information to assess how far we are from the optimum. From this perspective, the gradient mapping can be seen as an extension of the gradient for the composite case where the objective decomposes as a sum of a smooth part and a non-smooth part (Nesterov, 2004).

Lemma 2 (Checking the absolute accuracy criterion) *Consider a proximal center x, a smoothing parameter κ and an accuracy $\varepsilon > 0$. Consider an objective with the composite form (1) and we set function h as*

$$h(z) = f(z) + \frac{\kappa}{2}\|x - z\|^2 = \underbrace{f_0(z) + \frac{\kappa}{2}\|x - z\|^2}_{\triangleq\, h_0} + \psi(x).$$

For any $z \in \mathbb{R}^p$, we define

$$[z]_\eta = \operatorname{prox}_{\eta\psi}\left(z - \eta\nabla h_0\left(z\right)\right), \quad \textit{with} \quad \eta = \frac{1}{\kappa + L}. \tag{7}$$

Then, the gradient mapping of h at z is defined by $\frac{1}{\eta}(z - [z]_\eta)$ and

$$\frac{1}{\eta}\left\|z - [z]_\eta\right\| \leq \sqrt{2\kappa\varepsilon} \quad \textit{implies} \quad [z]_\eta \in p^\varepsilon(x).$$

The proof is given in Appendix B. The lemma shows that it is sufficient to check the norm of the gradient mapping to ensure condition (C1). However, this requires an additional full gradient step and proximal step at each iteration.

As soon as we have an approximate proximal operator z in $p^\varepsilon(x)$ in hand, we can define an approximate gradient of the Moreau envelope,

$$g(z) \triangleq \kappa(x - z), \tag{8}$$

by mimicking the exact gradient formula $\nabla F(x) = \kappa(x - p(x))$. As a consequence, we may immediately draw a link

$$z \in p^\varepsilon(x) \quad \Longrightarrow \quad \|z - p(x)\| \leq \sqrt{\frac{2\varepsilon}{\kappa}} \quad \Longleftrightarrow \quad \|g(z) - \nabla F(x)\| \leq \sqrt{2\kappa\varepsilon}, \tag{9}$$

where the first implication is a consequence of the strong convexity of h at its minimum $p(x)$. We will then apply the approximate gradient g instead of ∇F to build the inexact proximal point algorithm. Since the inexactness of the approximate gradient can be bounded by an absolute value $\sqrt{2\kappa\varepsilon}$, we call (C1) the absolute accuracy criterion.

Relative error criterion. Another natural way to bound the gradient approximation is by using a relative error, namely in the form $\|g(z) - \nabla F(x)\| \leq \delta'\|\nabla F(x)\|$ for some $\delta' > 0$. This leads us to the following inexactness criterion.

Given a proximal center x, a smoothing parameter κ and a relative accuracy δ in $[0, 1)$, we denote the set of δ-relative approximations by

$$g^\delta(x) \triangleq \left\{ z \in \mathbb{R}^p \quad \text{s.t.} \quad h(z) - h^* \leq \frac{\delta\kappa}{2}\|x - z\|^2 \right\}, \tag{C2}$$

At a first glance, we may interpret the criterion (C2) as (C1) by setting $\varepsilon = \frac{\delta\kappa}{2}\|x - z\|^2$. But we should then notice that the accuracy depends on the point z, which is is no longer an absolute constant. In other words, the accuracy varies from point to point, which is proportional to the squared distance between z and x. First one may wonder whether $g^\delta(x)$ is an empty set. Indeed, it is easy to see that $p(x) \in g^\delta(x)$ since $h(p(x)) - h^* = 0 \leq \frac{\delta\kappa}{2}\|x - p(x)\|^2$. Moreover, by continuity, $g^\delta(x)$ is closed set around $p(x)$. Then, by following similar steps as in (9), we have

$$z \in g^\delta(x) \quad \Longrightarrow \quad \|z - p(x)\| \leq \sqrt{\delta}\|x - z\| \leq \sqrt{\delta}(\|x - p(x)\| + \|p(x) - z\|).$$

By defining the approximate gradient in the same way $g(z) = \kappa(x - z)$ yields,

$$z \in g^\delta(x) \quad \Longrightarrow \quad \|g(z) - \nabla F(x)\| \leq \delta'\|\nabla F(x)\| \quad \text{with} \quad \delta' = \frac{\sqrt{\delta}}{1 - \sqrt{\delta}},$$

which is the desired relative gradient approximation.

Finally, the discussion about bounding $h(z) - h^*$ still holds here. In particular, Lemma 2 may be used by setting the value $\varepsilon = \frac{\delta\kappa}{2}\|x - z\|^2$. The price to pay is as an additional gradient step and an additional proximal step per iteration.

A few remarks on related works. Inexactness criteria with respect to subgradient norms have been investigated in the past, starting from the pioneer work of Rockafellar (1976) in the context of the inexact proximal point algorithm. Later, different works have been dedicated to more practical inexactness criteria (Auslender, 1987; Correa and Lemaréchal, 1993; Solodov and Svaiter, 2001; Fuentes et al., 2012). These criteria include duality gap, ε-subdifferential, or decrease in terms of function value. Here, we present a more intuitive point of view using the Moreau envelope.

While the proximal point algorithm has caught a lot of attention, very few works have focused on its accelerated variant. The first accelerated proximal point algorithm with inexact gradients was proposed by Güler (1992). Then, Salzo and Villa (2012) proposed a more rigorous convergence analysis, and more inexactness criteria, which are typically stronger than ours. In the same way, a more general inexact oracle framework has been proposed later by Devolder et al. (2014). To achieve the Catalyst acceleration, our main effort was to propose and analyze criteria that allow us to control the complexity for finding approximate solutions of the sub-problems.

3. Catalyst Acceleration

Catalyst is presented in Algorithm 2. As discussed in Section 2, this scheme can be interpreted as an inexact accelerated proximal point algorithm, or equivalently as an accelerated

gradient descent method applied to the Moreau envelope of the objective with inexact gradients. Since an overview has already been presented in Section 1.2, we now present important details to obtain acceleration in theory and in practice.

Algorithm 2 Catalyst

input Initial estimate x_0 in $\mathbb{R}^p$, smoothing parameter κ, strong convexity parameter μ, optimization method $\mathcal{M}$ and a stopping criterion based on a sequence of accuracies $(\varepsilon_k)_{k\geq 0}$, or $(\delta_k)_{k\geq 0}$, or a fixed budget T.

1: Initialize $y_0 = x_0$, $q = \frac{\mu}{\mu+\kappa}$. If $\mu > 0$, set $\alpha_0 = \sqrt{q}$, otherwise $\alpha_0 = 1$.

2: **while** the desired accuracy is not achieved **do**

3: Compute an approximate solution of the following problem with $\mathcal{M}$

$$x_k \approx \arg\min_{x\in\mathbb{R}^p} \left\{ h_k(x) \triangleq f(x) + \frac{\kappa}{2}\|x - y_{k-1}\|^2 \right\},$$

using the warm-start strategy of Section 3 and one of the following stopping criteria:

(a) *absolute accuracy:* find x_k in $p^{\varepsilon_k}(y_{k-1})$ by using criterion (C1);

(b) *relative accuracy:* find x_k in $g^{\delta_k}(y_{k-1})$ by using criterion (C2);

(c) *fixed budget:* run $\mathcal{M}$ for T iterations and output x_k.

4: Update α_k in $(0,1)$ by solving the equation

$$\alpha_k^2 = (1-\alpha_k)\alpha_{k-1}^2 + q\alpha_k. \tag{10}$$

5: Compute y_k with Nesterov's extrapolation step

$$y_k = x_k + \beta_k(x_k - x_{k-1}) \quad \text{with} \quad \beta_k = \frac{\alpha_{k-1}(1-\alpha_{k-1})}{\alpha_{k-1}^2 + \alpha_k}. \tag{11}$$

6: **end while**

output x_k (final estimate).

Requirement: linear convergence of the method $\mathcal{M}$. One of the main characteristic of Catalyst is to apply the method $\mathcal{M}$ to strongly-convex sub-problems, without requiring strong convexity of the objective f. As a consequence, Catalyst provides direct support for convex but non-strongly convex objectives to $\mathcal{M}$, which may be useful to extend the scope of application of techniques that need strong convexity to operate. Yet, Catalyst requires solving these sub-problems efficiently enough in order to control the complexity of the inner-loop computations. When applying $\mathcal{M}$ to minimize a strongly-convex function h, we assume that $\mathcal{M}$ is able to produce a sequence of iterates $(z_t)_{t\geq 0}$ such that

$$h(z_t) - h^* \leq C_{\mathcal{M}}(1-\tau_{\mathcal{M}})^t(h(z_0) - h^*), \tag{12}$$

where z_0 is the initial point given to $\mathcal{M}$, and $\tau_{\mathcal{M}}$ in $(0,1)$, $C_{\mathcal{M}} > 0$ are two constants. In such a case, we say that $\mathcal{M}$ admits a linear convergence rate. The quantity $\tau_{\mathcal{M}}$ controls the speed

of convergence for solving the sub-problems: the larger is $\tau_{\mathcal{M}}$, the faster is the convergence. For a given algorithm $\mathcal{M}$, the quantity $\tau_{\mathcal{M}}$ depends usually on the condition number of h. For instance, for the proximal gradient method and many first-order algorithms, we simply have $\tau_{\mathcal{M}} = O((\mu+\kappa)/(L+\kappa))$, as h is $(\mu+\kappa)$-strongly convex and $(L+\kappa)$-smooth. Catalyst can also be applied to randomized methods $\mathcal{M}$ that satisfy (12) in expectation:

$$\mathbb{E}[h(z_t) - h^*] \leq C_{\mathcal{M}}(1-\tau_{\mathcal{M}})^t(h(z_0) - h^*), \tag{13}$$

Then, the complexity results of Section 4 also hold in expectation. This allows us to apply Catalyst to randomized block coordinate descent algorithms (see Richtárik and Takáč, 2014, and references therein), and some incremental algorithms such as SAG, SAGA, or SVRG. For other methods that admit a linear convergence rates in terms of duality gap, such as SDCA, MISO/Finito, Catalyst can also be applied as explained in Appendix C.

Stopping criteria. Catalyst may be used with three types of stopping criteria for solving the inner-loop problems. We now detail them below.

(a) *absolute accuracy*: we predefine a sequence $(\varepsilon_k)_{k\geq 0}$ of accuracies, and stop the method $\mathcal{M}$ by using the absolute stopping criterion (C1). Our analysis suggests

- if f is μ-strongly convex,

$$\varepsilon_k = \frac{1}{2}(1-\rho)^k(f(x_0) - f^*) \quad \text{with} \quad \rho < \sqrt{q}\,.$$

- if f is convex but not strongly convex,

$$\varepsilon_k = \frac{f(x_0) - f^*}{2(k+2)^{4+\gamma}} \quad \text{with} \quad \gamma > 0\,.$$

Typically, $\gamma = 0.1$ and $\rho = 0.9\sqrt{q}$ are reasonable choices, both in theory and in practice. Of course, the quantity $f(x_0) - f^*$ is unknown and we need to upper bound it by a duality gap or by Lemma 2 as discussed in Section 2.3.

(b) *relative accuracy*: To use the relative stopping criterion (C2), our analysis suggests the following choice for the sequence $(\delta_k)_{k\geq 0}$:

- if f is μ-strongly convex,

$$\delta_k = \frac{\sqrt{q}}{2-\sqrt{q}}\,.$$

- if f is convex but not strongly convex,

$$\delta_k = \frac{1}{(k+1)^2}\,.$$

(c) *fixed budget*: Finally, the simplest way of using Catalyst is to fix in advance the number T of iterations of the method $\mathcal{M}$ for solving the sub-problems without

checking any optimality criterion. Whereas our analysis provides theoretical budgets that are compatible with this strategy, we found them to be pessimistic and impractical. Instead, we propose an aggressive strategy for incremental methods that simply consists of setting $T = n$. This setting was called the "one-pass" strategy in the original Catalyst paper (Lin et al., 2015a).

Warm-starts in inner loops. Besides linear convergence rate, an adequate warm-start strategy needs to be used to guarantee that the sub-problems will be solved in reasonable computational time. The intuition is that the previous solution may still be a good approximation of the current subproblem. Specifically, the following choices arise from the convergence analysis that will be detailed in Section 4.

Consider the minimization of the $(k+1)$-th subproblem $h_{k+1}(z) = f(z) + \frac{\kappa}{2}\|z - y_k\|^2$, we warm start the optimization method $\mathcal{M}$ at z_0 as following:

(a) when using criterion (C1) to find x_{k+1} in $p^{\varepsilon_k}(y_k)$,

- if f is smooth ($\psi = 0$), then choose $z_0 = x_k + \frac{\kappa}{\kappa+\mu}(y_k - y_{k-1})$.
- if f is composite as in (1), then define $w_0 = x_k + \frac{\kappa}{\kappa+\mu}(y_k - y_{k-1})$ and

$$z_0 = [w_0]_\eta = \text{prox}_{\eta\psi}\left(w_0 - \eta g\right) \text{ with } \eta = \frac{1}{L+\kappa} \text{ and } g = \nabla f_0(w_0) + \kappa(w_0 - y_k).$$

(b) when using criteria (C2) to find x_{k+1} in $g^{\delta_k}(y_k)$,

- if f is smooth ($\psi = 0$), then choose $z_0 = y_k$.
- if f is composite as in (1), then choose

$$z_0 = [y_k]_\eta = \text{prox}_{\eta\psi}(y_k - \eta\nabla f_0(y_k)) \quad \text{with} \quad \eta = \frac{1}{L+\kappa}.$$

(c) when using a fixed budget T, choose the same warm start strategy as in (b).

Note that the earlier conference paper (Lin et al., 2015a) considered the the warm start rule $z_0 = x_{k-1}$. That variant is also theoretically validated but it does not perform as well as the ones proposed here in practice.

Optimal balance: choice of parameter κ. Finally, the last ingredient is to find an optimal balance between the inner-loop (for solving each sub-problem) and outer-loop computations. To do so, we minimize our global complexity bounds with respect to the value of κ. As we shall see in Section 5, this strategy turns out to be reasonable in practice. Then, as shown in the theoretical section, the resulting rule of thumb is

We select κ by maximizing the ratio $\tau_{\mathcal{M}}/\sqrt{\mu+\kappa}$.

We recall that $\tau_{\mathcal{M}}$ characterizes how fast $\mathcal{M}$ solves the sub-problems, according to (12); typically, $\tau_{\mathcal{M}}$ depends on the condition number $\frac{L+\kappa}{\mu+\kappa}$ and is a function of κ.[2] In Table 2, we illustrate the choice of κ for different methods. Note that the resulting rule for incremental methods is very simple for the pracitioner: select κ such that the condition number $\frac{\bar{L}+\kappa}{\mu+\kappa}$ is of the order of n; then, the inner-complexity becomes $O(n\log(1/\varepsilon))$.

Method $\mathcal{M}$	Inner-complexity	$\tau_{\mathcal{M}}$	Choice for κ
FG	$O\left(n\frac{L+\kappa}{\mu+\kappa}\log\left(\frac{1}{\varepsilon}\right)\right)$	$\propto \frac{\mu+\kappa}{L+\kappa}$	$L - 2\mu$
SAG/SAGA/SVRG	$O\left(\left(n + \frac{\bar{L}+\kappa}{\mu+\kappa}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$	$\propto \frac{\mu+\kappa}{n(\mu+\kappa)+\bar{L}+\kappa}$	$\frac{\bar{L}-\mu}{n+1} - \mu$

Table 2: Example of choices of the parameter κ for the full gradient (FG) and incremental methods SAG/SAGA/SVRG. See Table 1 for details about the complexity.

4. Convergence and Complexity Analysis

We now present the complexity analysis of Catalyst. In Section 4.1, we analyze the convergence rate of the outer loop, regardless of the complexity for solving the sub-problems. Then, we analyze the complexity of the inner-loop computations for our various stopping criteria and warm-start strategies in Section 4.2. Section 4.3 combines the outer- and inner-loop analysis to provide the global complexity of Catalyst applied to a given optimization method $\mathcal{M}$.

4.1 Complexity Analysis for the Outer-Loop

The complexity analysis of the first variant of Catalyst we presented in (Lin et al., 2015a) used a tool called "estimate sequence", which was introduced by Nesterov (2004). Here, we provide a simpler proof. We start with criterion (C1), before extending the result to (C2).

4.1.1 Analysis for Criterion (C1)

The next theorem describes how the errors $(\varepsilon_k)_{k\geq 0}$ accumulate in Catalyst.

Theorem 3 (Convergence of outer-loop for criterion (C1)) *Consider the sequences $(x_k)_{k\geq 0}$ and $(y_k)_{k\geq 0}$ produced by Algorithm 2, assuming that x_k is in $p^{\varepsilon_k}(y_{k-1})$ for all $k \geq 1$, Then,*

$$f(x_k) - f^* \leq A_{k-1}\left(\sqrt{(1-\alpha_0)(f(x_0) - f^*) + \frac{\gamma_0}{2}\|x^* - x_0\|^2} + 3\sum_{j=1}^{k}\sqrt{\frac{\varepsilon_j}{A_{j-1}}}\right)^2,$$

2. Note that the rule for the non strongly convex case, denoted here by $\mu = 0$, slightly differs from Lin et al. (2015a) and results from a tighter complexity analysis.

where

$$\gamma_0 = (\kappa+\mu)\alpha_0(\alpha_0 - q) \quad \textit{and} \quad A_k = \prod_{j=1}^{k}(1-\alpha_j) \quad \textit{with } A_0 = 1\,. \tag{14}$$

Before we prove this theorem, we note that by setting $\varepsilon_k = 0$ for all k, the speed of convergence of $f(x_k) - f^*$ is driven by the sequence $(A_k)_{k\geq 0}$. Thus we first show the speed of A_k by recalling the Lemma 2.2.4 of Nesterov (2004).

Lemma 4 (Lemma 2.2.4 of Nesterov 2004) *Consider the quantities γ_0, A_k defined in (14) and the α_k's defined in Algorithm 2. Then, if $\gamma_0 \geq \mu$,*

$$A_k \leq \min\left\{(1-\sqrt{q})^k, \frac{4}{\left(2+k\sqrt{\frac{\gamma_0}{\kappa}}\right)^2}\right\}.$$

For non-strongly convex objectives, A_k follows the classical accelerated $O(1/k^2)$ rate of convergence, whereas it achieves a linear convergence rate for the strongly convex case. Intuitively, we are applying an inexact Nesterov method on the Moreau envelope F, thus the convergence rate naturally depends on the inverse of its condition number, which is $q = \frac{\mu}{\mu+\kappa}$. We now provide the proof of the theorem below.

Proof We start by defining an approximate sufficient descent condition inspired by a remark of Chambolle and Pock (2015) regarding accelerated gradient descent methods. A related condition was also used by Paquette et al. (2018) in the context of non-convex optimization.

Approximate sufficient descent condition. Let us define the function

$$h_k(x) = f(x) + \frac{\kappa}{2}\|x - y_{k-1}\|^2.$$

Since $p(y_{k-1})$ is the unique minimizer of h_k, the strong convexity of h_k yields: for any $k \geq 1$, for all x in $\mathbb{R}^p$ and any $\theta_k > 0$,

$$\begin{aligned}
h_k(x) &\geq h_k^* + \frac{\kappa+\mu}{2}\|x - p(y_{k-1})\|^2 \\
&\geq h_k^* + \frac{\kappa+\mu}{2}(1-\theta_k)\|x - x_k\|^2 + \frac{\kappa+\mu}{2}\left(1-\frac{1}{\theta_k}\right)\|x_k - p(y_{k-1})\|^2 \\
&\geq h_k(x_k) - \varepsilon_k + \frac{\kappa+\mu}{2}(1-\theta_k)\|x - x_k\|^2 + \frac{\kappa+\mu}{2}\left(1-\frac{1}{\theta_k}\right)\|x_k - p(y_{k-1})\|^2,
\end{aligned}$$

where the $(\mu+\kappa)$-strong convexity of h_k is used in the first inequality; Lemma 19 is used in the second inequality, and the last one uses the relation $h_k(x_k) - h_k^* \leq \varepsilon_k$. Moreover, when $\theta_k \geq 1$, the last term is positive and we have

$$h_k(x) \geq h_k(x_k) - \varepsilon_k + \frac{\kappa+\mu}{2}(1-\theta_k)\|x - x_k\|^2.$$

If instead $\theta_k \leq 1$, the coefficient $\frac{1}{\theta_k} - 1$ is non-negative and we have

$$-\frac{\kappa+\mu}{2}\left(\frac{1}{\theta_k}-1\right)\|x_k - p(y_{k-1})\|^2 \geq -\left(\frac{1}{\theta_k}-1\right)(h_k(x_k) - h_k^*) \geq -\left(\frac{1}{\theta_k}-1\right)\varepsilon_k.$$

In this case, we have

$$h_k(x) \geq h_k(x_k) - \frac{\varepsilon_k}{\theta_k} + \frac{\kappa+\mu}{2}(1-\theta_k)\|x - x_k\|^2.$$

As a result, we have for all value of $\theta_k > 0$,

$$h_k(x) \geq h_k(x_k) + \frac{\kappa+\mu}{2}(1-\theta_k)\|x - x_k\|^2 - \frac{\varepsilon_k}{\min\{1,\theta_k\}}.$$

After expanding the expression of h_k, we then obtain the approximate descent condition

$$f(x_k) + \frac{\kappa}{2}\|x_k - y_{k-1}\|^2 + \frac{\kappa+\mu}{2}(1-\theta_k)\|x - x_k\|^2 \leq f(x) + \frac{\kappa}{2}\|x - y_{k-1}\|^2 + \frac{\varepsilon_k}{\min\{1,\theta_k\}}. \tag{15}$$

Definition of the Lyapunov function. We introduce a sequence $(S_k)_{k\geq 0}$ that will act as a Lyapunov function, with

$$S_k = (1-\alpha_k)(f(x_k) - f^*) + \alpha_k\frac{\kappa\eta_k}{2}\|x^* - v_k\|^2. \tag{16}$$

where x^* is a minimizer of f, $(v_k)_{k\geq 0}$ is a sequence defined by $v_0 = x_0$ and

$$v_k = x_k + \frac{1-\alpha_{k-1}}{\alpha_{k-1}}(x_k - x_{k-1}) \quad \text{for } k \geq 1,$$

and $(\eta_k)_{k\geq 0}$ is an auxiliary quantity defined by

$$\eta_k = \frac{\alpha_k - q}{1-q}.$$

The way we introduce these variables allow us to write the following relationship,

$$y_k = \eta_k v_k + (1-\eta_k)x_k, \quad \text{for all } k \geq 0,$$

which follows from a simple calculation. Then by setting $z_k = \alpha_{k-1}x^* + (1-\alpha_{k-1})x_{k-1}$ the following relations hold for all $k \geq 1$.

$$\begin{aligned}
f(z_k) &\leq \alpha_{k-1}f^* + (1-\alpha_{k-1})f(x_{k-1}) - \frac{\mu\alpha_{k-1}(1-\alpha_{k-1})}{2}\|x^* - x_{k-1}\|^2, \\
z_k - x_k &= \alpha_{k-1}(x^* - v_k),
\end{aligned}$$

and also the following one

$$\begin{aligned}
\|z_k - y_{k-1}\|^2 &= \|(\alpha_{k-1} - \eta_{k-1})(x^* - x_{k-1}) + \eta_{k-1}(x^* - v_{k-1})\|^2 \\
&= \alpha_{k-1}^2\left\|\left(1 - \frac{\eta_{k-1}}{\alpha_{k-1}}\right)(x^* - x_{k-1}) + \frac{\eta_{k-1}}{\alpha_{k-1}}(x^* - v_{k-1})\right\|^2 \\
&\leq \alpha_{k-1}^2\left(1 - \frac{\eta_{k-1}}{\alpha_{k-1}}\right)\|x^* - x_{k-1}\|^2 + \alpha_{k-1}^2\frac{\eta_{k-1}}{\alpha_{k-1}}\|x^* - v_{k-1}\|^2 \\
&= \alpha_{k-1}(\alpha_{k-1} - \eta_{k-1})\|x^* - x_{k-1}\|^2 + \alpha_{k-1}\eta_{k-1}\|x^* - v_{k-1}\|^2,
\end{aligned}$$

where we used the convexity of the norm and the fact that $\eta_k \leq \alpha_k$. Using the previous relations in (15) with $x = z_k = \alpha_{k-1}x^* + (1-\alpha_{k-1})x_{k-1}$, gives for all $k \geq 1$,

$$\begin{aligned} f(x_k) + \frac{\kappa}{2}\|x_k - y_{k-1}\|^2 + \frac{\kappa+\mu}{2}(1-\theta_k)\alpha_{k-1}^2\|x^* - v_k\|^2 \\ \leq \alpha_{k-1}f^* + (1-\alpha_{k-1})f(x_{k-1}) - \frac{\mu}{2}\alpha_{k-1}(1-\alpha_{k-1})\|x^* - x_{k-1}\|^2 \\ + \frac{\kappa\alpha_{k-1}(\alpha_{k-1}-\eta_{k-1})}{2}\|x^* - x_{k-1}\|^2 + \frac{\kappa\alpha_{k-1}\eta_{k-1}}{2}\|x^* - v_{k-1}\|^2 + \frac{\varepsilon_k}{\min\{1,\theta_k\}}. \end{aligned}$$

Remark that for all $k \geq 1$,

$$\alpha_{k-1} - \eta_{k-1} = \alpha_{k-1} - \frac{\alpha_{k-1}-q}{1-q} = \frac{q(1-\alpha_{k-1})}{1-q} = \frac{\mu}{\kappa}(1-\alpha_{k-1}),$$

and the quadratic terms involving $x^* - x_{k-1}$ cancel each other. Then, after noticing that for all $k \geq 1$,

$$\eta_k\alpha_k = \frac{\alpha_k^2 - q\alpha_k}{1-q} = \frac{(\kappa+\mu)(1-\alpha_k)\alpha_{k-1}^2}{\kappa},$$

which allows us to write

$$f(x_k) - f^* + \frac{\kappa+\mu}{2}\alpha_{k-1}^2\|x^* - v_k\|^2 = \frac{S_k}{1-\alpha_k}. \tag{17}$$

We are left, for all $k \geq 1$, with

$$\frac{1}{1-\alpha_k}S_k \leq S_{k-1} + \frac{\varepsilon_k}{\min\{1,\theta_k\}} - \frac{\kappa}{2}\|x_k - y_{k-1}\|^2 + \frac{(\kappa+\mu)\alpha_{k-1}^2\theta_k}{2}\|x^* - v_k\|^2. \tag{18}$$

Control of the approximation errors for criterion (C1). Using the fact that

$$\frac{1}{\min\{1,\theta_k\}} \leq 1 + \frac{1}{\theta_k},$$

we immediately derive from equation (18) that

$$\frac{1}{1-\alpha_k}S_k \leq S_{k-1} + \varepsilon_k + \frac{\varepsilon_k}{\theta_k} - \frac{\kappa}{2}\|x_k - y_{k-1}\|^2 + \frac{(\kappa+\mu)\alpha_{k-1}^2\theta_k}{2}\|x^* - v_k\|^2. \tag{19}$$

By minimizing the right-hand side of (19) with respect to θ_k, we obtain the following inequality

$$\frac{1}{1-\alpha_k}S_k \leq S_{k-1} + \varepsilon_k + \sqrt{2\varepsilon_k(\mu+\kappa)}\alpha_{k-1}\|x^* - v_k\|,$$

and after unrolling the recursion,

$$\frac{S_k}{A_k} \leq S_0 + \sum_{j=1}^k \frac{\varepsilon_j}{A_{j-1}} + \sum_{j=1}^k \frac{\sqrt{2\varepsilon_j(\mu+\kappa)}\alpha_{j-1}\|x^* - v_j\|}{A_{j-1}}.$$

From Equation (17), the lefthand side is larger than $\frac{(\mu+\kappa)\alpha_{k-1}^2\|x^* - v_k\|^2}{2A_{k-1}}$. We may now define $u_j = \frac{\sqrt{(\mu+\kappa)}\alpha_{j-1}\|x^* - v_j\|}{\sqrt{2A_{j-1}}}$ and $a_j = 2\frac{\sqrt{\varepsilon_j}}{\sqrt{A_{j-1}}}$, and we have

$$u_k^2 \leq S_0 + \sum_{j=1}^k \frac{\varepsilon_j}{A_{j-1}} + \sum_{j=1}^k a_j u_j \quad \text{for all } k \geq 1.$$

This allows us to apply Lemma 20, which yields

$$\begin{aligned} \frac{S_k}{A_k} &\leq \left(\sqrt{S_0 + \sum_{j=1}^k \frac{\varepsilon_j}{A_{j-1}}} + 2\sum_{j=1}^k \sqrt{\frac{\varepsilon_j}{A_{j-1}}}\right)^2, \\ &\leq \left(\sqrt{S_0} + 3\sum_{j=1}^k \sqrt{\frac{\varepsilon_j}{A_{j-1}}}\right)^2 \end{aligned}$$

which provides us the desired result given that $f(x_k) - f^* \leq \frac{S_k}{1-\alpha_k}$ and that $v_0 = x_0$. ■

We are now in shape to state the convergence rate of the Catalyst algorithm with criterion (C1), without taking into account yet the cost of solving the sub-problems. The next two propositions specialize Theorem 3 to the strongly convex case and non strongly convex cases, respectively. Their proofs are provided in Appendix B.

Proposition 5 (μ-strongly convex case, criterion (C1))
In Algorithm 2, choose $\alpha_0 = \sqrt{q}$ and

$$\varepsilon_k = \frac{2}{9}(f(x_0) - f^*)(1-\rho)^k \quad \text{with} \quad \rho < \sqrt{q}.$$

Then, the sequence of iterates $(x_k)_{k\geq 0}$ satisfies

$$f(x_k) - f^* \leq \frac{8}{(\sqrt{q}-\rho)^2}(1-\rho)^{k+1}(f(x_0) - f^*).$$

Proposition 6 (Convex case, criterion (C1))
When $\mu = 0$, choose $\alpha_0 = 1$ and

$$\varepsilon_k = \frac{2(f(x_0) - f^*)}{9(k+1)^{4+\gamma}} \quad \text{with} \quad \gamma > 0.$$

Then, Algorithm 2 generates iterates $(x_k)_{k\geq 0}$ such that

$$f(x_k) - f^* \leq \frac{8}{(k+1)^2}\left(\frac{\kappa}{2}\|x_0 - x^*\|^2 + \frac{4}{\gamma^2}(f(x_0) - f^*)\right).$$

4.1.2 Analysis for Criterion (C2)

Then, we may now analyze the convergence of Catalyst under criterion (C2), which offers similar guarantees as (C1), as far as the outer loop is concerned.

Theorem 7 (Convergence of outer-loop for criterion (C2)) *Consider the sequences $(x_k)_{k\geq 0}$ and $(y_k)_{k\geq 0}$ produced by Algorithm 2, assuming that x_k is in $g^{\delta_k}(y_{k-1})$ for all $k \geq 1$ and δ_k in $(0,1)$. Then,*

$$f(x_k) - f^* \leq \frac{A_{k-1}}{\prod_{j=1}^k (1-\delta_j)} \left((1-\alpha_0)(f(x_0) - f^*) + \frac{\gamma_0}{2}\|x_0 - x^*\|^2 \right),$$

where γ_0 and $(A_k)_{k\geq 0}$ are defined in (14) in Theorem 3.

Proof Remark that x_k in $g^{\delta_k}(y_{k-1})$ is equivalent to x_k in $p^{\varepsilon_k}(y_{k-1})$ with an adaptive error $\varepsilon_k = \frac{\delta_k \kappa}{2}\|x_k - y_{k-1}\|^2$. All steps of the proof of Theorem 3 hold for such values of ε_k and from (18), we may deduce

$$\frac{S_k}{1-\alpha_k} - \frac{(\kappa+\mu)\alpha_{k-1}^2\theta_k}{2}\|x^* - v_k\|^2 \leq S_{k-1} + \left(\frac{\delta_k \kappa}{2\min\{1,\theta_k\}} - \frac{\kappa}{2} \right) \|x_k - y_{k-1}\|^2.$$

Then, by choosing $\theta_k = \delta_k < 1$, the quadratic term on the right disappears and the left-hand side is greater than $\frac{1-\delta_k}{1-\alpha_k} S_k$. Thus,

$$S_k \leq \frac{1-\alpha_k}{1-\delta_k} S_{k-1} \leq \frac{A_k}{\prod_{j=1}^k (1-\delta_j)} S_0,$$

which is sufficient to conclude since $(1-\alpha_k)(f(x_k) - f^*) \leq S_k$. ■

The next propositions specialize Theorem 7 for specific choices of sequence $(\delta_k)_{k\geq 0}$ in the strongly and non strongly convex cases.

Proposition 8 (μ-strongly convex case, criterion (C2))
In Algorithm 2, choose $\alpha_0 = \sqrt{q}$ and

$$\delta_k = \frac{\sqrt{q}}{2-\sqrt{q}}.$$

Then, the sequence of iterates $(x_k)_{k\geq 0}$ satisfies

$$f(x_k) - f^* \leq 2\left(1 - \frac{\sqrt{q}}{2}\right)^k (f(x_0) - f^*).$$

Proof This is a direct application of Theorem 7 by remarking that $\gamma_0 = (1-\sqrt{q})\mu$ and

$$S_0 = (1-\sqrt{q})\left(f(x_0) - f^* + \frac{\mu}{2}\|x^* - x_0\|^2 \right) \leq 2(1-\sqrt{q})(f(x_0) - f^*).$$

And $\alpha_k = \sqrt{q}$ for all $k \geq 0$ leading to

$$\frac{1-\alpha_k}{1-\delta_k} = 1 - \frac{\sqrt{q}}{2}$$

■

Proposition 9 (Convex case, criterion (C2))
When $\mu = 0$, choose $\alpha_0 = 1$ and

$$\delta_k = \frac{1}{(k+1)^2}.$$

Then, Algorithm 2 generates iterates $(x_k)_{k\geq 0}$ such that

$$f(x_k) - f^* \leq \frac{4\kappa\|x_0 - x^*\|^2}{(k+1)^2}. \tag{20}$$

Proof This is a direct application of Theorem 7 by remarking that $\gamma_0 = \kappa$, $A_k \leq \frac{4}{(k+2)^2}$ (Lemma 4) and

$$\prod_{i=1}^k \left(1 - \frac{1}{(i+1)^2}\right) = \prod_{i=1}^k \frac{i(i+2)}{(i+1)^2} = \frac{k+2}{2(k+1)} \geq \frac{1}{2}.$$

■

Remark 10 *In fact, the choice of δ_k can be improved by taking $\delta_k = \frac{1}{(k+1)^{1+\gamma}}$ for any $\gamma > 0$, which comes at the price of a larger constant in (20).*

4.2 Analysis of Warm-start Strategies for the Inner Loop

In this section, we study the complexity of solving the subproblems with the proposed warm start strategies. The only assumption we make on the optimization method $\mathcal{M}$ is that it enjoys linear convergence when solving a strongly convex problem—meaning, it satisfies either (12) or its randomized variant (13). Then, the following lemma gives us a relation between the accuracy required to solve the sub-problems and the corresponding complexity.

Lemma 11 (Accuracy vs. complexity) *Let us consider a strongly convex objective h and a linearly convergent method $\mathcal{M}$ generating a sequence of iterates $(z_t)_{t\geq 0}$ for minimizing h. Consider the complexity $T(\varepsilon) = \inf\{t \geq 0, h(z_t) - h^* \leq \varepsilon\}$, where $\varepsilon > 0$ is the target accuracy and h^* is the minimum value of h. Then,*

1. *If $\mathcal{M}$ is deterministic and satisfies (12), we have*

$$T(\varepsilon) \leq \frac{1}{\tau_{\mathcal{M}}} \log\left(\frac{C_{\mathcal{M}}(h(z_0) - h^*)}{\varepsilon} \right).$$

2. *If $\mathcal{M}$ is randomized and satisfies (13), we have*

$$\mathbb{E}[T(\varepsilon)] \leq \frac{1}{\tau_{\mathcal{M}}} \log\left(\frac{2C_{\mathcal{M}}(h(z_0) - h^*)}{\tau_{\mathcal{M}}\varepsilon} \right) + 1$$

The proof of the deterministic case is straightforward and the proof of the randomized case is provided in Appendix B.4. From the previous result, a good initialization is essential for fast convergence. More precisely, it suffices to control the initialization $\frac{h(z_0)-h^*}{\varepsilon}$ in order to bound the number of iterations $T(\varepsilon)$. For that purpose, we analyze the quality of various warm-start strategies.

4.2.1 Warm Start Strategies for Criterion (C1)

The next proposition characterizes the quality of initialization for (C1).

Proposition 12 (Warm start for criterion (C1)) *Assume that $\mathcal{M}$ is linearly convergent for strongly convex problems with parameter $\tau_{\mathcal{M}}$ according to (12), or according to (13) in the randomized case. At iteration $k+1$ of Algorithm 2, given the previous iterate x_k in $p^{\varepsilon_k}(y_{k-1})$, we consider the following function*

$$h_{k+1}(z) = f(z) + \frac{\kappa}{2}\|z - y_k\|^2,$$

which we minimize with $\mathcal{M}$, producing a sequence $(z_t)_{t\geq 0}$. Then,

- *when f is smooth, choose $z_0 = x_k + \frac{\kappa}{\kappa+\mu}(y_k - y_{k-1})$;*
- *when $f = f_0 + \psi$ is composite, choose $z_0 = [w_0]_\eta = \operatorname{prox}_{\eta\psi}(w_0 - \eta\nabla h_0(w_0))$ with $w_0 = x_k + \frac{\kappa}{\kappa+\mu}(y_k - y_{k-1})$, $\eta = \frac{1}{L+\kappa}$ and $h_0 = f_0 + \frac{\kappa}{2}\|\cdot - y_k\|^2$.*

We also assume that we choose α_0 and $(\varepsilon_k)_{k\geq 0}$ according to Proposition 5 for $\mu > 0$, or Proposition 6 for $\mu = 0$. Then,

1. *if f is μ-strongly convex, $h_{k+1}(z_0) - h_{k+1}^* \leq C\varepsilon_{k+1}$ where,*

$$C = \frac{L+\kappa}{\kappa+\mu}\left(\frac{2}{1-\rho} + \frac{2592(\kappa+\mu)}{(1-\rho)^2(\sqrt{q}-\rho)^2\mu}\right) \quad \text{if } f \text{ is smooth,} \tag{21}$$

   or

$$C = \frac{L+\kappa}{\kappa+\mu}\left(\frac{2}{1-\rho} + \frac{23328(L+\kappa)}{(1-\rho)^2(\sqrt{q}-\rho)^2\mu}\right) \quad \text{if } f \text{ is composite.} \tag{22}$$

2. *if f is convex with bounded level sets, there exists a constant $B > 0$ that only depends on f, x_0 and κ such that*

$$h_{k+1}(z_0) - h_{k+1}^* \leq B. \tag{23}$$

Proof We treat the smooth and composite cases separately.

Smooth and strongly-convex case. When f is smooth, by the gradient Lipschitz assumption,

$$h_{k+1}(z_0) - h_{k+1}^* \leq \frac{(L+\kappa)}{2}\|z_0 - p(y_k)\|^2.$$

Moreover,

$$\begin{aligned}
\|z_0 - p(y_k)\|^2 &= \left\|x_k + \frac{\kappa}{\kappa+\mu}(y_k - y_{k-1}) - p(y_k)\right\|^2 \\
&= \left\|x_k - p(y_{k-1}) + \frac{\kappa}{\kappa+\mu}(y_k - y_{k-1}) - (p(y_k) - p(y_{k-1}))\right\|^2 \\
&\leq 2\|x_k - p(y_{k-1})\|^2 + 2\left\|\frac{\kappa}{\kappa+\mu}(y_k - y_{k-1}) - (p(y_k) - p(y_{k-1}))\right\|^2.
\end{aligned}$$

Since x_k is in $p^{\varepsilon_k}(y_{k-1})$, we may control the first quadratic term on the right by noting that

$$\|x_k - p(y_{k-1})\|^2 \leq \frac{2}{\kappa+\mu}(h_k(x_k) - h_k^*) \leq \frac{2\varepsilon_k}{\kappa+\mu}.$$

Moreover, by the coerciveness property of the proximal operator,

$$\left\|\frac{\kappa}{\kappa+\mu}(y_k - y_{k-1}) - (p(y_k) - p(y_{k-1}))\right\|^2 \leq \|y_k - y_{k-1}\|^2,$$

see Appendix B.5 for the proof. As a consequence,

$$\begin{aligned}
h_{k+1}(z_0) - h_{k+1}^* &\leq \frac{(L+\kappa)}{2}\|z_0 - p(y_k)\|^2 \\
&\leq 2\frac{L+\kappa}{\mu+\kappa}\varepsilon_k + (L+\kappa)\|y_k - y_{k-1}\|^2,
\end{aligned} \tag{24}$$

Then, we need to control the term $\|y_k - y_{k-1}\|^2$. Inspired by the proof of accelerated SDCA of Shalev-Shwartz and Zhang (2016),

$$\begin{aligned}
\|y_k - y_{k-1}\| &= \|x_k + \beta_k(x_k - x_{k-1}) - x_{k-1} - \beta_{k-1}(x_{k-1} - x_{k-2})\| \\
&\leq (1+\beta_k)\|x_k - x_{k-1}\| + \beta_{k-1}\|x_{k-1} - x_{k-2}\| \\
&\leq 3\max\{\|x_k - x_{k-1}\|, \|x_{k-1} - x_{k-2}\|\},
\end{aligned}$$

The last inequality was due to the fact that $\beta_k \leq 1$. In fact,

$$\beta_k^2 = \frac{\left(\alpha_{k-1} - \alpha_{k-1}^2\right)^2}{\left(\alpha_{k-1}^2 + \alpha_k\right)^2} = \frac{\alpha_{k-1}^2 + \alpha_{k-1}^4 - 2\alpha_{k-1}^3}{\alpha_k^2 + 2\alpha_k\alpha_{k-1}^2 + \alpha_{k-1}^4} = \frac{\alpha_{k-1}^2 + \alpha_{k-1}^4 - 2\alpha_{k-1}^3}{\alpha_{k-1}^2 + \alpha_{k-1}^4 + q\alpha_k + \alpha_k\alpha_{k-1}^2} \leq 1,$$

where the last equality uses the relation $\alpha_k^2 + \alpha_k\alpha_{k-1}^2 = \alpha_{k-1}^2 + q\alpha_k$ from (10). Then,

$$\|x_k - x_{k-1}\| \leq \|x_k - x^*\| + \|x_{k-1} - x^*\|,$$

and by strong convexity of f

$$\frac{\mu}{2}\|x_k - x^*\|^2 \leq f(x_k) - f^* \leq \frac{36}{(\sqrt{q}-\rho)^2}\varepsilon_{k+1},$$

where the last inequality is obtained from Proposition 5. As a result,

$$\begin{aligned}
\|y_k - y_{k-1}\|^2 &\leq 9\max\left\{\|x_k - x_{k-1}\|^2, \|x_{k-1} - x_{k-2}\|^2\right\} \\
&\leq 36\max\left\{\|x_k - x^*\|^2, \|x_{k-1} - x^*\|^2, \|x_{k-2} - x^*\|^2\right\} \\
&\leq \frac{2592\,\varepsilon_{k-1}}{(\sqrt{q}-\rho)^2\mu}.
\end{aligned}$$

Since $\varepsilon_{k+1} = (1-\rho)^2\varepsilon_{k-1}$, we may now obtain (21) from (24) and the previous bound.

Smooth and convex case. When $\mu = 0$, Eq. (24) is still valid but we need to control $\|y_k - y_{k-1}\|^2$ in a different way. From Proposition 6, the sequence $(f(x_k))_{k\geq 0}$ is bounded by a constant that only depends on f and x_0; therefore, by the bounded level set assumption, there exists $R > 0$ such that

$$\|x_k - x^*\| \leq R, \quad \text{for all } k \geq 0.$$

Thus, following the same argument as the strongly convex case, we have

$$\|y_k - y_{k-1}\| \leq 36R^2 \quad \text{for all } k \geq 1,$$

and we obtain (23) by combining the previous inequality with (24).

Composite case. By using the notation of gradient mapping introduced in (7), we have $z_0 = [w_0]_\eta$. By following similar steps as in the proof of Lemma 2, the gradient mapping satisfies the following relation

$$h_{k+1}(z_0) - h^*_{k+1} \leq \frac{1}{2(\kappa+\mu)} \left\| \frac{1}{\eta}(w_0 - z_0) \right\|^2,$$

and it is sufficient to bound $\|w_0 - z_0\| = \|w_0 - [w_0]_\eta\|$. For that, we introduce

$$[x_k]_\eta = \text{prox}_{\eta\psi}(x_k - \eta(\nabla f_0(x_k) + \kappa(x_k - y_{k-1}))).$$

Then,

$$\|w_0 - [w_0]_\eta\| \leq \|w_0 - x_k\| + \|x_k - [x_k]_\eta\| + \|[x_k]_\eta - [w_0]_\eta\|, \tag{25}$$

and we will bound each term on the right. By construction

$$\|w_0 - x_k\| = \frac{\kappa}{\kappa+\mu}\|y_k - y_{k-1}\| \leq \|y_k - y_{k-1}\|.$$

Next, it is possible to show that the gradient mapping satisfies the following relation (see Nesterov, 2013),

$$\frac{1}{2\eta}\|x_k - [x_k]_\eta\|^2 \leq h_k(x_k) - h^*_k \leq \varepsilon_k.$$

And then since $[x_k]_\eta = \text{prox}_{\eta\psi}(x_k - \eta(\nabla f_0(x_k) + \kappa(x_k - y_{k-1})))$ and $[w_0]_\eta = \text{prox}_{\eta\psi}(w_0 - \eta(\nabla f_0(w_0) + \kappa(w_0 - y_k)))$. From the non expansiveness of the proximal operator, we have

$$\begin{aligned}
\|[x_k]_\eta - [w_0]_\eta\| &\leq \|x_k - \eta(\nabla f_0(x_k) + \kappa(x_k - y_{k-1})) - (w_0 - \eta(\nabla f_0(w_0) + \kappa(w_0 - y_k)))\| \\
&\leq \|x_k - \eta(\nabla f_0(x_k) + \kappa(x_k - y_{k-1})) - (w_0 - \eta(\nabla f_0(w_0) + \kappa(w_0 - y_{k-1})))\| \\
&\quad + \eta\kappa\|y_k - y_{k-1}\| \\
&\leq \|x_k - w_0\| + \eta\kappa\|y_k - y_{k-1}\| \\
&\leq 2\|y_k - y_{k-1}\|.
\end{aligned}$$

We have used the fact that $\|x - \eta\nabla h(x) - (y - \eta\nabla h(y))\| \leq \|x - y\|$. By combining the previous inequalities with (25), we finally have

$$\|w_0 - [w_0]_\eta\| \leq \sqrt{2\eta\varepsilon_k} + 3\|y_k - y_{k-1}\|.$$

Thus, by using the fact that $(a+b)^2 \leq 2a^2 + 2b^2$ for all a, b,

$$h_{k+1}(z_0) - h^*_{k+1} \leq \frac{L+\kappa}{\kappa+\mu}\left(2\varepsilon_k + 9(L+\kappa)\|y_k - y_{k-1}\|^2\right),$$

and we can obtain (22) and (23) by upper-bounding $\|y_k - y_{k-1}\|^2$ in a similar way as in the smooth case, both when $\mu > 0$ and $\mu = 0$.

■

Finally, the complexity of the inner loop can be obtained directly by combining the previous proposition with Lemma 11.

Corollary 13 (Inner-loop Complexity for Criterion (C1)) *Consider the setting of Proposition 12; then, the sequence $(z_t)_{t\geq 0}$ minimizing h_{k+1} is such that the complexity $T_{k+1} = \inf\{t \geq 0, h_{k+1}(z_t) - h^*_{k+1} \leq \varepsilon_{k+1}\}$ satisfies*

$$T_{k+1} \leq \frac{1}{\tau_{\mathcal{M}}}\log(C_{\mathcal{M}}C) \quad \text{if } \mu > 0 \quad \Longrightarrow \quad T_{k+1} = \tilde{O}\left(\frac{1}{\tau_{\mathcal{M}}}\right),$$

where C is the constant defined in (21) or in (22) for the composite case; and

$$T_{k+1} \leq \frac{1}{\tau_{\mathcal{M}}}\log\left(\frac{9C_{\mathcal{M}}(k+2)^{4+\eta}B}{2(f(x_0) - f^*)}\right) \quad \text{if } \mu = 0 \quad \Longrightarrow \quad T_{k+1} = \tilde{O}\left(\frac{\log(k+2)}{\tau_{\mathcal{M}}}\right),$$

where B is the uniform upper bound in (23). Furthermore, when $\mathcal{M}$ is randomized, the expected complexity $\mathbb{E}[T_{k+1}]$ is similar, up to a factor $2/\tau_{\mathcal{M}}$ in the logarithm—see Lemma 11, and we have $\mathbb{E}[T_{k+1}] = \tilde{O}(1/\tau_{\mathcal{M}})$ when $\mu > 0$ and $\mathbb{E}[T_{k+1}] = \tilde{O}(\log(k+2)/\tau_{\mathcal{M}})$. Here, $\tilde{O}(.)$ hides logarithmic dependencies in parameters $\mu, L, \kappa, C_{\mathcal{M}}$, $\tau_{\mathcal{M}}$ and $f(x_0) - f^*$.

4.2.2 Warm Start Strategies for Criterion (C2)

We may now analyze the inner-loop complexity for criterion (C2) leading to upper bounds with smaller constants and simpler proofs. Note also that in the convex case, the bounded level set condition will not be needed, unlike for criterion (C1). To proceed, we start with a simple lemma that gives us a sufficient condition for (C2) to be satisfied.

Lemma 14 (Sufficient condition for criterion (C2)) *If a point z satisfies*

$$h_{k+1}(z) - h^*_{k+1} \leq \frac{\delta_{k+1}\kappa}{8}\|p(y_k) - y_k\|^2,$$

then z is in $g^{\delta_{k+1}}(y_k)$.

Proof

$$\begin{aligned} h_{k+1}(z) - h_{k+1}^* &\leq \frac{\delta_{k+1}\kappa}{8}\|p(y_k) - y_k\|^2 \\ &\leq \frac{\delta_{k+1}\kappa}{4}\left(\|p(y_k) - z\|^2 + \|z - y_k\|^2\right) \\ &\leq \frac{\delta_{k+1}\kappa}{4}\left(\frac{2}{\mu+\kappa}(h_{k+1}(z) - h_{k+1}^*) + \|z - y_k\|^2\right) \\ &\leq \frac{1}{2}\left(h_{k+1}(z) - h_{k+1}^*\right) + \frac{\delta_{k+1}\kappa}{4}\|z - y_k\|^2. \end{aligned}$$

Rearranging the terms gives the desired result. ■

With the previous result, we can control the complexity of the inner-loop minimization with Lemma 11 by choosing $\varepsilon = \frac{\delta_{k+1}\kappa}{8}\|p(y_k) - y_k\|^2$. However, to obtain a meaningful upper bound, we need to control the ratio

$$\frac{h_{k+1}(z_0) - h_{k+1}^*}{\varepsilon} = \frac{8(h_{k+1}(z_0) - h_{k+1}^*)}{\delta_{k+1}\kappa\|p(y_k) - y_k\|^2}.$$

Proposition 15 (Warm start for criterion (C2)) *Assume that $\mathcal{M}$ is linearly convergent for strongly convex problems with parameter $\tau_{\mathcal{M}}$ according to (12), or according to (13) in the randomized case. At iteration $k+1$ of Algorithm 2, given the previous iterate x_k in $g^{\delta_k}(y_{k-1})$, we consider the following function*

$$h_{k+1}(z) = f(z) + \frac{\kappa}{2}\|z - y_k\|^2,$$

which we minimize with $\mathcal{M}$, producing a sequence $(z_t)_{t\geq 0}$. Then,

- *when f is smooth, set $z_0 = y_k$;*
- *when $f = f_0 + \psi$ is composite, set $z_0 = [y_k]_\eta = \operatorname{prox}_{\eta\psi}(y_k - \eta\nabla f_0(y_k))$ with $\eta = \frac{1}{L+\kappa}$.*

Then,

$$h_{k+1}(z_0) - h_{k+1}^* \leq \frac{L+\kappa}{2}\|p(y_k) - y_k\|^2. \tag{26}$$

Proof When f is smooth, the optimality conditions of $p(y_k)$ yield $\nabla h_{k+1}(p(y_k)) = \nabla f(p(y_k)) + \kappa(p(y_k) - y_k) = 0$. As a result,

$$\begin{aligned} h_{k+1}(z_0) - h_{k+1}^* &= f(y_k) - \left(f(p(y_k)) + \frac{\kappa}{2}\|p(y_k) - y_k\|^2\right) \\ &\leq f(p(y_k)) + \langle\nabla f(p(y_k)), y_k - p(y_k)\rangle + \frac{L}{2}\|y_k - p(y_k)\|^2 \\ &\quad - \left(f(p(y_k)) + \frac{\kappa}{2}\|p(y_k) - y_k\|^2\right) \\ &= \frac{L+\kappa}{2}\|p(y_k) - y_k\|^2. \end{aligned}$$

When f is composite, we use the inequality in Lemma 2.3 of Beck and Teboulle (2009): for any z,

$$h_{k+1}(z) - h_{k+1}(z_0) \geq \frac{L+\kappa}{2}\|z_0 - y_k\|^2 + (L+\kappa)\langle z_0 - y_k, y_k - z\rangle,$$

Then, we apply this inequality with $z = p(y_k)$, and thus,

$$\begin{aligned} h_{k+1}(z_0) - h_{k+1}^* &\leq -\frac{L+\kappa}{2}\|z_0 - y_k\|^2 - (L+\kappa)\langle z_0 - y_k, y_k - p(y_k)\rangle \\ &\leq \frac{L+\kappa}{2}\|p(y_k) - y_k\|^2. \end{aligned}$$

■

We are now in shape to derive a complexity bound for criterion (C2), which is obtained by combining directly Lemma 11 with the value $\varepsilon = \frac{\delta_{k+1}\kappa}{8}\|p(y_k) - y_k\|^2$, Lemma 14, and the previous proposition.

Corollary 16 (Inner-loop Complexity for Criterion (C2)) *Consider the setting of Proposition 15 when $\mathcal{M}$ is deterministic; assume further that α_0 and $(\delta_k)_{k\geq 0}$ are chosen according to Proposition 8 for $\mu > 0$, or Proposition 9 for $\mu = 0$.*

Then, the sequence $(z_t)_{t\geq 0}$ is such that the complexity $T_{k+1} = \inf\{t \geq 0, z_t \in g^{\delta_{k+1}}(y_k)\}$ satisfies

$$T_{k+1} \leq \frac{1}{\tau_{\mathcal{M}}}\log\left(4C_{\mathcal{M}}\frac{(L+\kappa)}{\kappa}\frac{2-\sqrt{q}}{\sqrt{q}}\right) \quad \text{when } \mu > 0,$$

and

$$T_{k+1} \leq \frac{1}{\tau_{\mathcal{M}}}\log\left(4C_{\mathcal{M}}\frac{(L+\kappa)}{\kappa}(k+2)^2\right) \quad \text{when } \mu = 0.$$

When $\mathcal{M}$ is randomized, the expected complexity is similar, up to a factor $2/\tau_{\mathcal{M}}$ in the logarithm—see Lemma 11, and we have $\mathbb{E}[T_{k+1}] = \tilde{O}(1/\tau_{\mathcal{M}})$ when $\mu > 0$ and $\mathbb{E}[T_{k+1}] = \tilde{O}(\log(k+2)/\tau_{\mathcal{M}})$.

The inner-loop complexity is asymptotically similar with criterion (C2) as with criterion (C1), but the constants are significantly better.

4.3 Global Complexity Analysis

In this section, we combine the previous outer-loop and inner-loop convergence results to derive a global complexity bound. We treat here the strongly convex ($\mu > 0$) and convex ($\mu = 0$) cases separately.

4.3.1 Strongly Convex Case

When the problem is strongly convex, we remark that the subproblems are solved in a constant number of iterations $T_k = T = \tilde{O}\left(\frac{1}{\tau_{\mathcal{M}}}\right)$ for both criteria (C1) and (C2). This means that the iterate x_k in Algorithm 2 is obtained after $s = kT$ iterations of the method $\mathcal{M}$. Thus, the true convergence rate of Catalyst applied to $\mathcal{M}$ is of the form

$$f_s - f^* = f\left(x_{\frac{s}{T}}\right) - f^* \leq C'(1-\rho)^{\frac{s}{T}}(f(x_0) - f^*) \leq C'\left(1 - \frac{\rho}{T}\right)^s(f(x_0) - f^*), \tag{27}$$

where $f_s = f(x_k)$ is the function value after s iterations of $\mathcal{M}$. Then, choosing κ consists of maximizing the rate of convergence (27). In other words, we want to maximize $\sqrt{q}/T = \tilde{O}(\sqrt{q}\tau_{\mathcal{M}})$. Since $q = \frac{\mu}{\mu+\kappa}$, this naturally lead to the maximization of $\tau_{\mathcal{M}}/\sqrt{\mu+\kappa}$. We now state more formally the global convergence result in terms of complexity.

Proposition 17 (Global Complexity for strongly convex objectives) *When f is μ-strongly convex and all parameters are chosen according to Propositions 5 and 12 when using criterion (C1), or Propositions 8 and 15 for (C2), then Algorithm 2 finds a solution $\hat{x}$ such that $f(\hat{x}) - f^* \leq \varepsilon$ in at most $N_{\mathcal{M}}$ iterations of a deterministic method $\mathcal{M}$ with*

1. *when criterion (C1) is used,*

$$N_{\mathcal{M}} \leq \frac{1}{\tau_{\mathcal{M}}\rho} \log\left(C_{\mathcal{M}} C\right) \cdot \log\left(\frac{8(f(x_0) - f^*)}{(\sqrt{q}-\rho)^2\varepsilon}\right) = \tilde{O}\left(\frac{1}{\tau_{\mathcal{M}}\sqrt{q}} \log\left(\frac{1}{\varepsilon}\right)\right),$$

   where $\rho = 0.9\sqrt{q}$ and C is the constant defined in (21) or (22) for the composite case;

2. *when criterion (C2) is used,*

$$N_{\mathcal{M}} \leq \frac{2}{\tau_{\mathcal{M}}\sqrt{q}} \log\left(4C_{\mathcal{M}}\frac{L+\kappa}{\kappa}\frac{2-\sqrt{q}}{\sqrt{q}}\right) \cdot \log\left(\frac{2(f(x_0) - f^*)}{\varepsilon}\right) = \tilde{O}\left(\frac{1}{\tau_{\mathcal{M}}\sqrt{q}} \log\left(\frac{1}{\varepsilon}\right)\right).$$

Note that similar results hold in terms of expected number of iterations when the method $\mathcal{M}$ is randomized (see the end of Proposition 12).

Proof Let K be the number of iterations of the outer-loop algorithm required to obtain an ε-accurate solution. From Proposition 5, using (C1) criterion yields

$$K \leq \frac{1}{\rho} \log\left(\frac{8(f(x_0) - f^*)}{(\sqrt{q}-\rho)^2\varepsilon}\right).$$

From Proposition 8, using (C2) criterion yields

$$K \leq \frac{2}{\sqrt{q}} \log\left(\frac{2(f(x_0) - f^*)}{\varepsilon}\right).$$

Then since the number of runs of $\mathcal{M}$ is constant for any inner loop, the total number $N_{\mathcal{M}}$ is given by KT where T is respectively given by Corollaries 13 and 16. ■

4.3.2 Convex, but not Strongly Convex Case

When $\mu = 0$, the number of iterations for solving each subproblems grows logarithmically, which means that the iterate x_k in Algorithm 2 is obtained after $s =\leq kT\log(k+2)$ iterations of the method $\mathcal{M}$, where T is a constant. By using the global iteration counter $s = kT\log(k+2)$, we finally have

$$f_s - f^* \leq C' \frac{\log^2(s)}{s^2}\left(f(x_0) - f^* + \frac{\kappa}{2}\|x_0 - x^*\|^2\right). \tag{28}$$

This rate is *near-optimal*, up to a logarithmic factor, when compared to the optimal rate $O(1/s^2)$. This may be the price to pay for using a generic acceleration scheme. As before, we detail the global complexity bound for convex objectives in the next proposition.

Proposition 18 (Global complexity for convex objectives) *When f is convex and all parameters are chosen according to Propositions 6 and 12 when using criterion (C1), or Propositions 9 and 15 for criterion (C2), then Algorithm 2 finds a solution $\hat{x}$ such that $f(\hat{x}) - f^* \leq \varepsilon$ in at most $N_{\mathcal{M}}$ iterations of a deterministic method $\mathcal{M}$ with*

1. *when criterion (C1) is applied*

$$N_{\mathcal{M}} \leq \frac{1}{\tau_{\mathcal{M}}} K \log\left(\frac{9C_{\mathcal{M}}BK^{4+\gamma}}{2(f(x_0) - f^*)}\right) = \tilde{O}\left(\frac{1}{\tau_{\mathcal{M}}}\sqrt{\frac{\kappa}{\varepsilon}} \log\left(\frac{1}{\varepsilon}\right)\right),$$

   where,

$$K_\varepsilon = \sqrt{\frac{8\left(\frac{\kappa}{2}\|x_0 - x^*\|^2 + \frac{4}{\gamma^2}(f(x_0) - f^*)\right)}{\varepsilon}};$$

2. *when criterion (C2) is applied,*

$$\begin{aligned} N_{\mathcal{M}} &\leq \frac{1}{\tau_{\mathcal{M}}}\sqrt{\frac{4\kappa\|x_0 - x^*\|^2}{\varepsilon}} \log\left(\frac{16C_{\mathcal{M}}(L+\kappa)\|x_0 - x^*\|^2}{\varepsilon}\right) \\ &= \tilde{O}\left(\frac{1}{\tau_{\mathcal{M}}}\sqrt{\frac{\kappa}{\varepsilon}} \log\left(\frac{1}{\varepsilon}\right)\right). \end{aligned}$$

Note that similar results hold in terms of expected number of iterations when the method $\mathcal{M}$ is randomized (see the end of Proposition 15).

Proof Let K denote the number of outer-loop iterations required to achieve an ε-accurate solution. From Proposition 6, when (C1) is applied, we have

$$K \leq \sqrt{\frac{8\left(\frac{\kappa}{2}\|x_0 - x^*\|^2 + \frac{4}{\gamma^2}(f(x_0) - f^*)\right)}{\varepsilon}}.$$

From Proposition 9, when (C2) is applied, we have

$$K \leq \sqrt{\frac{4\kappa\|x_0 - x^*\|^2}{\varepsilon}}.$$

Since the number of runs in the inner loop is increasing, we have

$$N_{\mathcal{M}} = \sum_{i=1}^{K} T_i \leq KT_K.$$

Respectively apply T_K obtained from Corollary 13 and Corollary 16 gives the result. ■

Theoretical foundations of the choice of κ. The parameter κ plays an important rule in the global complexity result. The linear convergence parameter $\tau_{\mathcal{M}}$ depends typically on κ since it controls the strong convexity parameter of the subproblems. The natural way to choose κ is to minimize the global complexity given by Proposition 17 and Proposition 18, which leads to the following rule

Choose κ to maximize $\frac{\tau_{\mathcal{M}}}{\sqrt{\mu+\kappa}}$,

where $\mu = 0$ when the problem is convex but not strongly convex. We now illustrate two examples when applying Catalyst to the classical gradient descent method and to the incremental approach SVRG.

Gradient descent. When $\mathcal{M}$ is the gradient descent method, we have

$$\tau_{\mathcal{M}} = \frac{\mu+\kappa}{L+\kappa}.$$

Maximizing the ratio $\frac{\tau_{\mathcal{M}}}{\sqrt{\mu+\kappa}}$ gives

$$\kappa = L - 2\mu, \quad \text{when } L > 2\mu.$$

Consequently, the complexity in terms of gradient evaluations for minimizing the finite sum (2), where each iteration of $\mathcal{M}$ cost n gradients, is given by

$$N_{\mathcal{M}} = \begin{cases} \tilde{O}\left(n\sqrt{\frac{L}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right) & \text{when } \mu > 0; \\ \tilde{O}\left(n\sqrt{\frac{L}{\varepsilon}}\log\left(\frac{1}{\varepsilon}\right)\right) & \text{when } \mu = 0. \end{cases}$$

These rates are near-optimal up to logarithmic constants according to the first-order lower bound (Nemirovskii and Yudin, 1983; Nesterov, 2004).

SVRG. For SVRG (Xiao and Zhang, 2014) applied to the same finite-sum objective,

$$\tau_{\mathcal{M}} = \frac{1}{n + \frac{\bar{L}+\kappa}{\mu+\kappa}}.$$

Thus, maximizing the corresponding ratio gives

$$\kappa = \frac{\bar{L}-\mu}{n+1} - \mu, \quad \text{when } \bar{L} > (n+2)\mu.$$

Consequently, the resulting global complexity, here in terms of expected number of gradient evaluations, is given by

$$\mathbb{E}[N_{\mathcal{M}}] = \begin{cases} \tilde{O}\left(\sqrt{n\frac{\bar{L}}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right) & \text{when } \mu > 0; \\ \tilde{O}\left(\sqrt{\frac{n\bar{L}}{\varepsilon}}\log\left(\frac{1}{\varepsilon}\right)\right) & \text{when } \mu = 0. \end{cases}$$

Note that we treat here only the case $\bar{L} > (n+2)\mu$ to simplify, see Table 1 for a general results. We also remark that Catalyst can be applied to similar incremental algorithms such as SAG/SAGA (Schmidt et al., 2017; Defazio et al., 2014a) or dual-type algorithm MISO/Finito (Mairal, 2015; Defazio et al., 2014b) or SDCA Shalev-Shwartz and Zhang (2012). Moreover, the resulting convergence rates are near-optimal up to logarithmic constants according to the first-order lower bound (Woodworth and Srebro, 2016; Arjevani and Shamir, 2016).

4.3.3 Practical Aspects of the Theoretical Analysis

So far, we have not discussed the fixed budget criterion mentioned in Section 3. The idea is quite natural and simple to implement: we predefine the number of iterations to run for solving each subproblems and stop worrying about the stopping condition. For example, when $\mu > 0$ and $\mathcal{M}$ is deterministic, we can simply run $T_{\mathcal{M}}$ iterations of $\mathcal{M}$ for each subproblem where $T_{\mathcal{M}}$ is greater than the value given by Corollaries 13 or 16, then the criterions (C1) and (C2) are guaranteed to be satisfied. Unfortunately, the theoretical bound of $T_{\mathcal{M}}$ is relatively poor and does not lead to a practical strategy. On the other hand, using a more aggressive strategy such as $T_{\mathcal{M}} = n$ for incremental algorithms, meaning one pass over the data, seems to provide outstanding results, as shown in the experimental part of this paper.

Finally, one could argue that choosing κ according to a worst-case convergence analysis is not necessarily a good choice. In particular, the convergence rate of the method $\mathcal{M}$, driven by the parameter $\tau_{\mathcal{M}}$ is probably often under estimated in the first place. This suggests that using a smaller value for κ than the one we have advocated earlier is a good thing. In practice, we have observed that indeed Catalyst is often robust to smaller values of κ than the theoretical one, but we have also observed that the theoretical value performs reasonably well, as we shall see in the next section.

5. Experimental Study

In this section, we conduct various experiments to study the effect of the Catalyst acceleration and its different variants, showing in particular how to accelerate SVRG, SAGA, and MISO. In Section 5.1, we describe the data sets and formulations considered for our evaluation, and in Section 5.2, we present the different variants of Catalyst. Then, we study different questions: which variant of Catalyst should we use for incremental approaches? (Section 5.3); how do various incremental methods compare when accelerated with Catalyst? (Section 5.4); what is the effect of Catalyst on the test error when Catalyst is used to minimize a regularized empirical risk? (Section 5.5); is the theoretical value for κ appropriate? (Section 5.6). The code used for all our experiments is available at `https://github.com/hongzhoulin89/Catalyst-QNing/`.

5.1 Data sets, Formulations, and Metric

Data sets. We consider six machine learning data sets with different characteristics in terms of size and dimension to cover a variety of situations.

name	covtype	alpha	real-sim	rcv1	MNIST-CKN	CIFAR-CKN
n	581 012	250 000	72 309	781 265	60 000	50 000
d	54	500	20 958	47 152	2 304	9 216

While the first four data sets are standard ones that were used in previous work about optimization methods for machine learning, the last two are coming from a computer vision application. MNIST and CIFAR-10 are two image classification data sets involving 10 classes. The feature representation of each image was computed using an unsupervised

convolutional kernel network Mairal (2016). We focus here on the the task of classifying class #1 vs. the rest of the data set.

Formulations. We consider three common optimization problems in machine learning and signal processing, which admit a particular structure (large finite sum, composite, strong convexity). For each formulation, we also consider a training set $(b_i, a_i)_{i=1}^n$ of n data points, where the b_i's are scalars in $\{-1, +1\}$ and the a_i are feature vectors in $\mathbb{R}^p$. Then, the goal is to fit a linear model x in $\mathbb{R}^p$ such that the scalar b_i can be well predicted by the inner-product $\approx a_i^\top x$, or by its sign. Specifically, the three formulations we consider are listed below.

- **ℓ_2^2-regularized Logistic Regression**:

$$\min_{x\in\mathbb{R}^p} \quad \frac{1}{n}\sum_{i=1}^n \log\left(1+\exp(-b_i\, a_i^T x)\right) + \frac{\mu}{2}\|x\|^2,$$

  which leads to a μ-strongly convex smooth optimization problem.

- **ℓ_1-regularized Linear Regression (LASSO)**:

$$\min_{x\in\mathbb{R}^p} \quad \frac{1}{2n}\sum_{i=1}^n (b_i - a_i^T x)^2 + \lambda\|x\|_1,$$

  which is non smooth and convex but not strongly convex.

- **$\ell_1 - \ell_2^2$-regularized Linear Regression (Elastic-Net)**:

$$\min_{x\in\mathbb{R}^p} \quad \frac{1}{2n}\sum_{i=1}^n (b_i - a_i^T x)^2 + \lambda\|x\|_1 + \frac{\mu}{2}\|x\|^2,$$

  which is based on the Elastic-Net regularization (Zou and Hastie, 2005) and leading to a strongly-convex optimization problem.

Each feature vector a_i is normalized, and a natural upper-bound on the Lipschitz constant L of the un-regularized objective can be easily obtained with $L_{\text{logistic}} = 1/4$ and $L_{\text{lasso}} = 1$. The regularization parameter μ and λ are choosing in the following way:

- For **Logistic Regression**, we find an optimal regularization parameter μ^* by 10-fold cross validation for each data set on a logarithmic grid $2^i/n$, with $i \in [-12, 3]$. Then, we set $\mu = \mu^*/2^3$ which corresponds to a small value of the regularization parameter and a relatively ill-conditioned problem.

- For **Elastic-Net**, we set $\mu = 0.01/n$ to simulate the ill-conditioned situation and add a small l_1-regularization penalty with $\lambda = 1/n$ that produces sparse solutions.

- For the **Lasso problem**, we consider a logarithmic grid $10^i/n$, with $i = -3, -2, \ldots, 3$, and we select the parameter λ that provides a sparse optimal solution closest to 10% non-zero coefficients, which leads to $\lambda = 10/n$ or $100/n$.

Note that for the strongly convex problems, the regularization parameter μ yields a lower bound on the strong convexity parameter of the problem.

Metric used. In this chapter, and following previous work about incremental methods (Schmidt et al., 2017), we plot objective values as a function of the number of gradients evaluated during optimization, which appears to be the computational bottleneck of all previously mentioned algorithms. Since no metric is perfect for comparing algorithms' speed, we shall make the two following remarks, such that the reader can interpret our results and the limitations of our study with no difficulty.

- Ideally, CPU-time is the gold standard but CPU time is implementation-dependent and hardware-dependent.

- We have chosen to count only gradients computed with random data access. Thus, computing n times a gradient f_i by picking each time one function at random counts as "n gradients", whereas we ignore the cost of computing a full gradient $(1/n)\sum_{i=1}^n \nabla f_i$ at once, where the f_i's can be accessed in sequential order. Similarly, we ignore the cost of computing the function value $f(x) = (1/n)\sum_{i=1}^n f_i(x)$, which is typically performed every pass on the data when computing a duality gap. While this assumption may be inappropriate in some contexts, the cost of random gradient computations was significantly dominating the cost of sequential access in our experiments, where (i) data sets fit into memory; (ii) computing full gradients was done in C++ by calling BLAS2 functions exploiting multiple cores.

5.2 Choice of Hyper-parameters and Variants

Before presenting the numerical results, we discuss the choice of default parameters used in the experiments as well as different variants.

Choice of method $\mathcal{M}$. We consider the acceleration of incremental algorithms which are able to adapt to the problem structure we consider: large sum of functions and possibly non-smooth regularization penalty.

- The proximal SVRG algorithm of Xiao and Zhang (2014) with stepsize $\eta = 1/L$.

- The SAGA algorithm Defazio et al. (2014a) with stepsize $\eta = 1/3L$.

- The proximal MISO algorithm of Lin et al. (2015a).

Choice of regularization parameter κ. As suggested by the theoretical analysis, we take κ to minimize the global complexity, leading to the choice

$$\kappa = \frac{L-\mu}{n+1} - \mu.$$

Stopping criteria for the inner loop. The choice of the accuracies are driven from the theoretical analysis described in paragraph 3. Here, we specify it again for the clarity of presentation:

- **Stopping criterion (C1).** Stop when $h_k(z_t) - h_k^* \leq \varepsilon_k$, where

$$\varepsilon_k = {}^5 \begin{cases} \frac{1}{2}(1-\rho)^k f(x_0) \;\text{ with }\; \rho = 0.9\sqrt{\frac{\mu}{\mu+\kappa}} & \text{when } \mu > 0; \\ \frac{f(x_0)}{2(k+1)^{4.1}} & \text{when } \mu = 0. \end{cases}$$

The duality gap $h(w_t) - h^*$ can be estimated either by evaluating the Fenchel conjugate function or by computing the squared norm of the gradient.

- **Stopping criterion (C2).** Stop when $h_k(z_t) - h_k^* \leq \delta_k \cdot \frac{\kappa}{2}\|z_t - y_{k-1}\|^2$, where

$$\delta_k = \begin{cases} \frac{\sqrt{q}}{2-\sqrt{q}} \text{ with } q = \frac{\mu}{\mu+\kappa} & \text{when } \mu > 0; \\ \frac{1}{(k+1)^2} & \text{when } \mu = 0. \end{cases}$$

- **Stopping criterion (C3) .** Perform exactly one pass over the data in the inner loop without checking any stopping criteria.[4]

Warm start for the inner loop. This is an important point to achieve acceleration which was not highlighted in the conference paper (Lin et al., 2015a). At iteration $k+1$, we consider the minimization of

$$h_{k+1}(z) = f_0(z) + \frac{\kappa}{2}\|z - y_k\|^2 + \psi(z).$$

We warm start according to the strategy defined in Section 3. Let x_k be the approximate minimizer of h_k, obtained from the last iteration.

- **Initialization for (C1).** Let us define $\eta = \frac{1}{L+\kappa}$, then initialize at

$$z_0^{C1} = \begin{cases} w_0 \triangleq x_k + \frac{\kappa}{\kappa+\mu}(y_k - y_{k-1}) & \text{if } \psi = 0; \\ [w_0]_\eta & \text{otherwise.} \end{cases}$$

  where $[w_0]_\eta = \text{prox}_{\eta\psi}(w_0 - \eta g)$ with $g = \nabla f_0(w_0) + \kappa(w_0 - y_k)$.

- **Initialization for (C2).** Intialize at

$$z_0^{C2} = \begin{cases} y_k & \text{if } \psi = 0; \\ [y_k]_\eta = \text{prox}_{\eta\psi}(y_k - \eta\nabla f_0(y_k)) & \text{otherwise.} \end{cases}$$

- **Initialization for (C3).** Take the best initial point among x_k and z_0^{C1}

$$z_0^{C3} \text{ such that } h_k(z_0^{C3}) = \min\{h_k(x_{k-1}), h_k(z_0^{C1})\}.$$

- **Initialization for (C1*).** Use the strategy (C1) with z_0^{C3}.

The warm start at z_0^{C3} requires to choose the best point between the last iterate x_k and the point z_0^{C1}. The motivation is that since the one-pass strategy is an aggressive heuristic, the solution of the subproblems may not be as accurate as the ones obtained with other criterions. Allowing using the iterate x_k turned out to be significantly more stable in practice. Then, it is also natural to use a similar strategy for criterion (C1), which we call (C1*). Using a similar strategy for (C2) turned out not to provide any benefit in practice and is thus omitted from the list here.

5. Here we upper bound $f(x_0) - f^*$ by $f(x_0)$ since f is always positive in our models.
4. This stopping criterion is heuristic since one pass may not be enough to achieve the required accuracy. What we have shown is that with a large enough $T_{\mathcal{M}}$, then the convergence will be guaranteed. Here we take heuristically $T_{\mathcal{M}}$ as one pass.

5.3 Comparison of Stopping Criteria and Warm-start Strategies

First, we evaluate the performance of the previous strategies when applying Catalyst to SVRG, SAGA and MISO. The results are presented in Figures 1, 2, and 3, respectively.

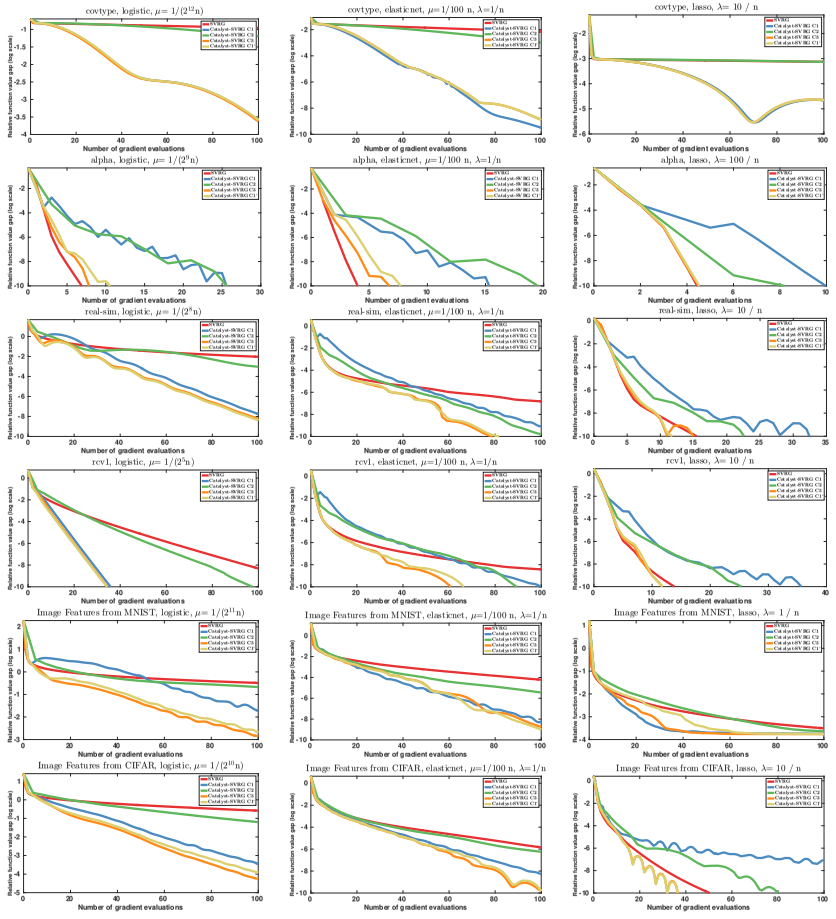


Figure 1: Experimental study of different stopping criterions for Catalyst-SVRG. We plot the value $f(x_k)/f^* - 1$ as a function of the number of gradient evaluations, on a logarithmic scale; the optimal value f^* is estimated with a duality gap.

Observations for Catalyst-SVRG. We remark that in most of the cases, the curve of (C3) and (C1*)are superimposed, meaning that one pass through the data is enough for solving the subproblem up to the required accuracy. Moreover, they give the best performance among all criterions. Regarding the logistic regression problem, the acceleration is significant (even huge for the covtype data set) except for alpha, where only (C3) and (C1*)do not degrade significantly the performance. For sparse problems, the effect of acceleration is more mitigated, with 7 cases out of 12 exhibiting important acceleration and 5 cases no acceleration. As before, (C3) and (C1*)are the only strategies that never degrade performance.

One reason explaining why acceleration is not systematic may be the ability of incremental methods to adapt to the unknown strong convexity parameter $\mu' \geq \mu$ hidden in the objective's loss, or local strong convexity near the solution. When $\mu'/L \geq 1/n$, we indeed obtain a well-conditioned regime where acceleration should not occur theoretically. In fact the complexity $O(n \log(1/\varepsilon))$ is already optimal in this regime, see Arjevani and Shamir (2016); Woodworth and Srebro (2016). For sparse problems, conditioning of the problem with respect to the linear subspace where the solution lies might also play a role, even though our analysis does not study this aspect. Therefore, this experiment suggests that adaptivity to unknown strong convexity is of high interest for incremental optimization.

Observations for Catalyst-SAGA. Our conclusions with SAGA are almost the same as with SVRG. However, in a few cases, we also notice that criterion C1 lacks stability, or at least exhibits some oscillations, which may suggest that SAGA has a larger variance compared to SVRG. The difference in the performance of (C1) and (C1*)can be huge, while they differ from each other only by the warm start strategy. Thus, *choosing a good initial point for solving the sub-problems is a key for obtaining acceleration in practice.*

Observations for Catalyst-MISO. The warm-start strategy of MISO is different from primal algorithms because parameters for the dual function need to be specified. The most natural way for warm starting the dual functions is to set

$$d_{k+1}(x) = d_k(x) + \frac{\kappa}{2}\|x - y_k\|^2 - \frac{\kappa}{2}\|x - y_{k-1}\|^2,$$

where d_k is the last dual function of the previous subproblem h_k. This gives the warm start

$$z_0 = \text{prox}\left(x_k + \frac{\kappa}{\kappa + \mu}(y_k - y_{k-1})\right).$$

For other choices of z_0, the dual function needs to be recomputed from scratch, which is computationally expensive and unstable for ill-conditioned problems. Thus, we only present the experimental results with respect to criterion (C1) and the one-pass heuristic (C3) . As we observe, a huge acceleration is obtained in logistic regression and Elastic-net formulations. For Lasso problem, the original Prox-MISO is not defined since the problem is not strongly convex. Thus, in order to make a comparison, we compare with Catalyst-SVRG which shows that the acceleration achieves a similar performance. This aligns with the theoretical result stating that Catalyst applied to incremental algorithms yields a similar convergence rate. Notice also that the original MISO algorithm suffers from numerical stability in this ill-conditioned regime chosen for our experiments. Catalyst not only accelerates MISO, but it also stabilizes it.

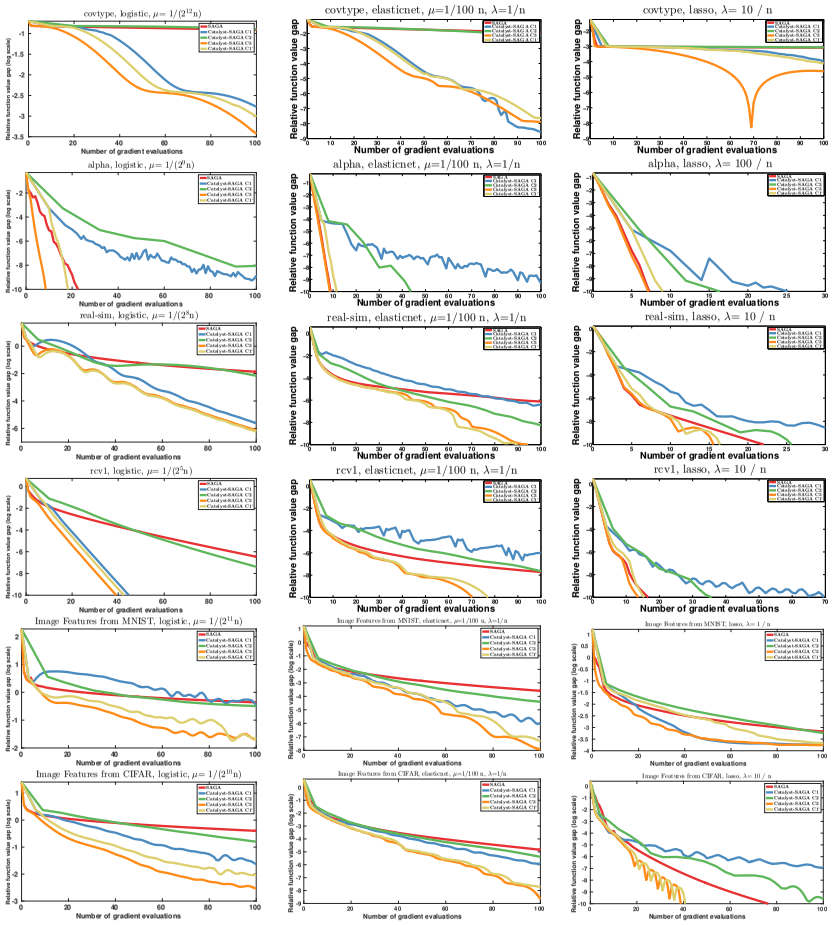


Figure 2: Experimental study of different stopping criterions for Catalyst-SAGA, with a similar setting as in Figure 1.

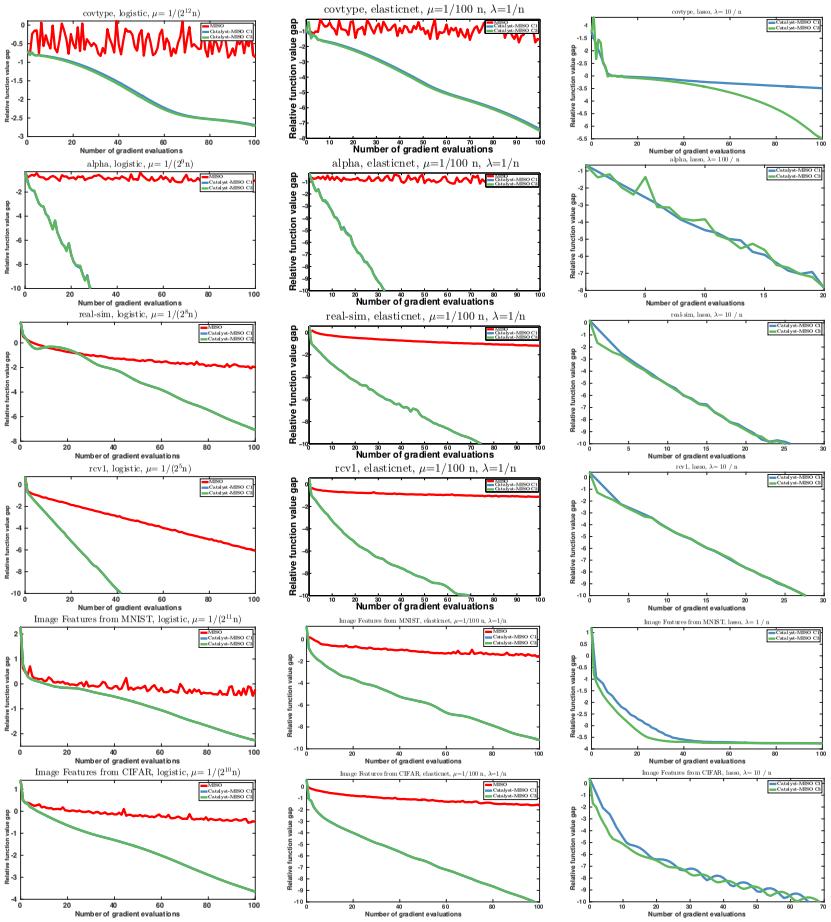


Figure 3: Experimental study of different stopping criterions for Catalyst-MISO, with a similar setting as in Figure 1

5.4 Acceleration of Existing Methods

Then, we put the previous curves into perspective and make a comparison of the performance before and after applying Catalyst across methods. We show the best performance among the three developed stopping criteria, which corresponds to be (C3) .

Observations. In Figure 4, we observe that by applying Catalyst, we accelerate the original algorithms up to the limitations discussed above (comparing the dashed line and the solid line of the same color). In three data sets (covtype, real-sim and rcv1), significant improvements are achieved as expected by the theory for the ill-conditioned problems in logistic regression and Elastic-net. For data set alpha, we remark that an relative accuracy in the order 10^{-10} is attained in less than 10 iterations. This suggests that the problems is in fact well-conditioned and there is some hidden strong convexity for this data set. Thus, the incremental algorithms like SVRG or SAGA are already optimal under this situation and no further improvement can be obtained by applying Catalyst.

5.5 Empirical Effect on the Generalization Error

A natural question that applies to all incremental methods is whether or not the acceleration that we may see for minimizing an empirical risk on *training* data affects the objective function and the test accuracy on new unseen *test* data. To answer this question, we consider the logistic regression formulation with the regularization parameter μ^* obtained by cross-validation. Then, we cut each data set into 80% of training data and set aside 20% of the data point as test data.

Observations on the test loss and test accuracy. The left column of Figure 5 shows the loss function on the training set, where acceleration is significant in 5 cases out of 6. The middle column shows the loss function evaluated on the test set, but on a non-logarithmic scale since the optimal value of the test loss is unknown. Acceleration appears in 4 cases out of 6. Regarding the test accuracy, an important acceleration is obtained in 2 cases, whereas it is less significant or negligible in the other cases.

5.6 Study of the Parameter κ

Finally, we evaluate the performance for different values of κ.

Observations for different choices of κ. We consider a logarithmic grid $\kappa = 10^i \kappa_0$ with $i = -2, -1, \cdots, 2$ and κ_0 is the optimal κ given by the theory. We observe that for ill-conditioned problems, using optimal choice κ_0 provides much better performance than other choices, which confirms the theoretical result. For the data set of alpha or Lasso problems, we observe that the best choice is given by the smallest $\kappa = 0.01\kappa_0$. This suggests that, as discussed before, there is a certain degree of strong convexity present in the objective even without any regularization.

6. Conclusion

We have introduced a generic algorithm called Catalyst that allows us to extend Nesterov's acceleration to a large class of first-order methods. We have shown that it can be effective

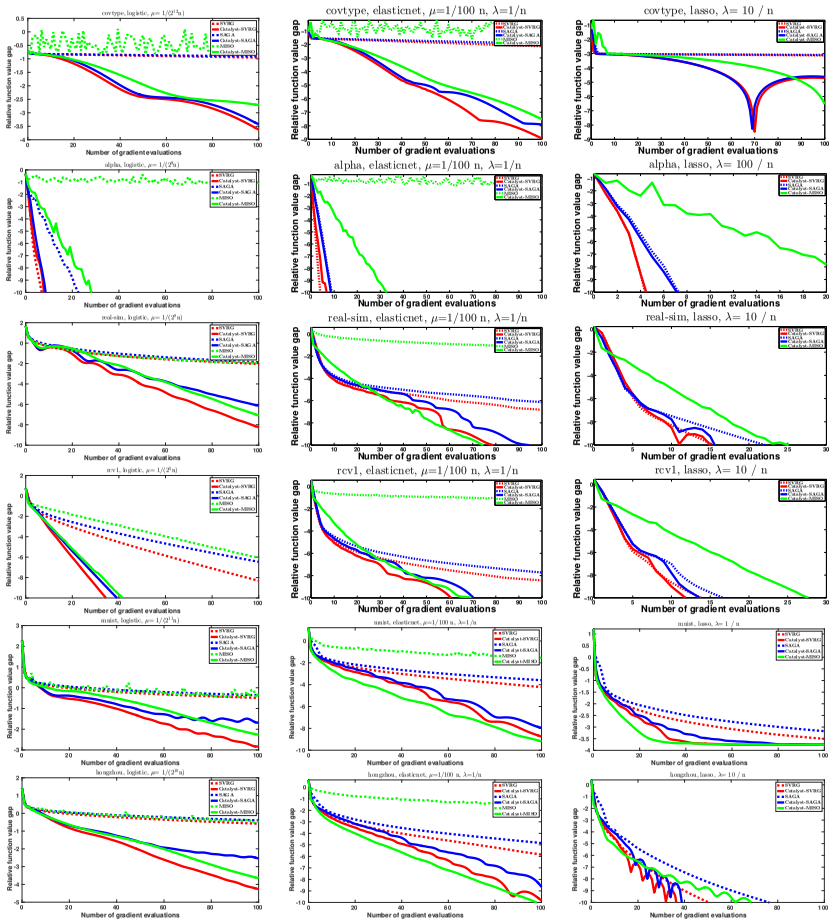


Figure 4: Experimental study of the performance of Catalyst applying to SVRG, SAGA and MISO. The dashed lines correspond to the original algorithms and the solid lines correspond to accelerated algorithms by applying Catalyst. We plot the relative function value gap $(f(x_k) - f^*)/f^*$ in the number of gradient evaluations, on a logarithmic scale.

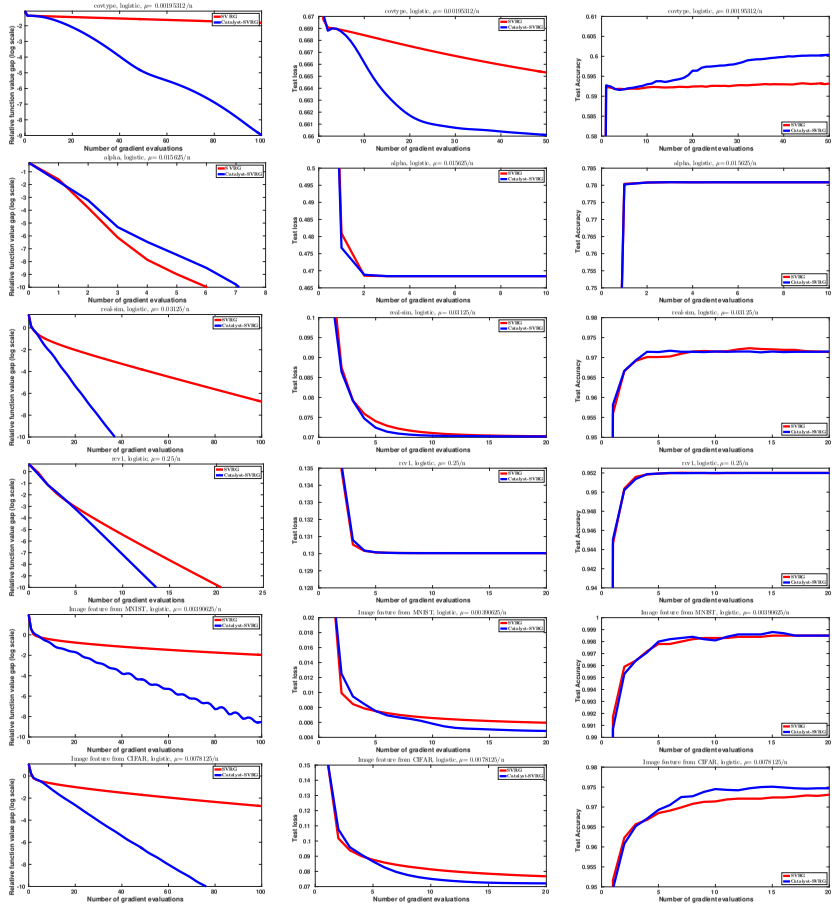


Figure 5: Empirical effect on the generalization error. For a logistic regression experiment, we report the value of the objective function evaluated on the training data on the left column, the value of the loss evaluated on a test set on the middle column, and the classification error evaluated on the test set on the right.

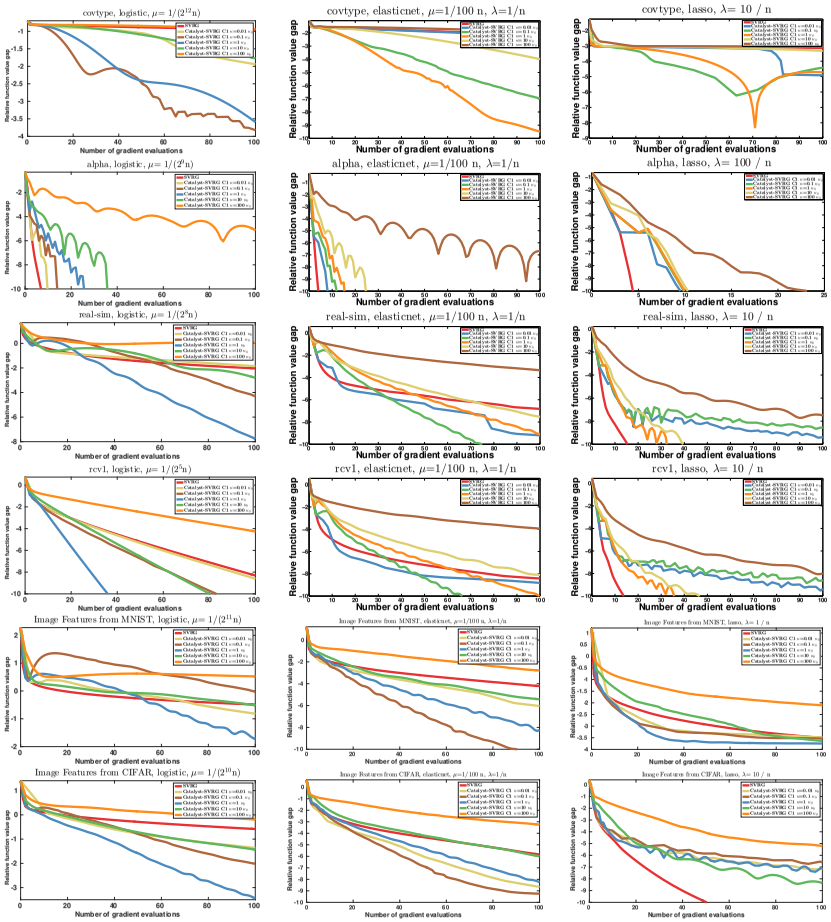


Figure 6: Evaluations of Catalyst-SVRG for different κ using stopping criterion C1, where κ_0 is the theoretical choice given by the complexity analysis.

in practice for ill-conditioned problems. Besides acceleration, Catalyst also improves the numerical stability of a given algorithm, by applying it to auxiliary problems that are better conditioned than the original objective. For this reason, it also provides support to convex, but not strongly convex objectives, to algorithms that originally require strong convexity. We have also studied experimentally many variants to identify the ones that are the most effective and the simplest to use in practice. For incremental methods, we showed that the "almost-parameter-free" variant, consisting in performing a single pass over the data at every outer-loop iteration, was the most effective one in practice.

Even though we have illustrated Catalyst in the context of finite-sum optimization problems, the main feature of our approach is its versatility. Catalyst could also be applied to other algorithms that have not been considered so far and give rise to new accelerated algorithms.

Acknowledgments

The authors would like to thank Dmitriy Drusvyatskiy, Anatoli Juditsky, Sham Kakade, Arkadi Nemirovski, Courtney Paquette, and Vincent Roulet for fruitful discussions. Hongzhou Lin and Julien Mairal were supported by the ERC grant SOLARIS (# 714381) and a grant from ANR (MACARON project ANR-14-CE23-0003-01). Zaid Harchaoui was supported by NSF Award CCF-1740551 and the program "Learning in Machines and Brains" of CIFAR. This work was performed while Hongzhou Lin was at Inria and Univ. Grenoble Alpes.

Appendix A. Useful Lemmas

Lemma 19 (Simple lemma on quadratic functions) *For all vectors x, y, z in $\mathbb{R}^p$ and $\theta > 0$,*

$$\|x - y\|^2 \geq (1-\theta)\|x - z\|^2 + \left(1 - \frac{1}{\theta}\right)\|z - y\|^2.$$

Proof

$$\begin{aligned}
\|x - y\|^2 &= \|x - z + z - y\|^2 \\
&= \|x - z\|^2 + \|z - y\|^2 + 2\langle x - z, z - y\rangle \\
&= \|x - z\|^2 + \|z - y\|^2 + \left\|\sqrt{\theta}(x - z) + \frac{1}{\sqrt{\theta}}(z - y)\right\|^2 - \theta\|x - z\|^2 - \frac{1}{\theta}\|z - y\|^2 \\
&\geq (1-\theta)\|x - z\|^2 + \left(1 - \frac{1}{\theta}\right)\|z - y\|^2.
\end{aligned}$$

■

Lemma 20 (Simple lemma on non-negative sequences) *Consider a increasing sequence $(S_k)_{k\geq 0}$ and two non-negative sequences $(a_k)_{k\geq 0}$ and $(u_k)_{k\geq 0}$ such that for all k,*

$$u_k^2 \leq S_k + \sum_{i=1}^{k} a_i u_i. \tag{29}$$

Then,

$$S_k + \sum_{i=1}^{k} a_i u_i \leq \left(\sqrt{S_k} + \sum_{i=1}^{k} a_i \right)^2. \tag{30}$$

Proof This lemma is identical to the Lemma A.10 in the original Catalyst paper (Lin et al., 2015a), inspired by a lemma of Schmidt et al. (2011) for controlling errors of inexact proximal gradient methods.

We give here an elementary proof for completeness based on induction. The relation (30) is obviously true for $k = 0$. Then, we assume it is true for $k-1$ and prove the relation for k. We remark that from (29),

$$\left(u_k - \frac{a_k}{2}\right)^2 \leq S_k + \sum_{i=1}^{k-1} a_i u_i + \frac{a_k^2}{4},$$

and then

$$u_k \leq \sqrt{S_k + \sum_{i=1}^{k-1} a_i u_i + \frac{a_k^2}{4}} + \frac{a_k}{2}.$$

We may now prove the relation (30) by induction,

$$\begin{aligned}
S_k + \sum_{i=1}^{k} a_i u_i &\leq S_k + \sum_{i=1}^{k-1} a_i u_i + a_k \left(\frac{a_k}{2} + \sqrt{S_k + \sum_{i=1}^{k-1} a_i u_i + \frac{a_k^2}{4}} \right) \\
&\leq S_k + \sum_{i=1}^{k-1} a_i u_i + a_k \left(a_k + \sqrt{S_k + \sum_{i=1}^{k-1} a_i u_i} \right) \\
&\leq \left(\sqrt{S_k + \sum_{i=1}^{k-1} a_i u_i} + a_k \right)^2 \\
&= \left(\sqrt{(S_k - S_{k-1}) + (S_{k-1} + \sum_{i=1}^{k-1} a_i u_i)} + a_k \right)^2 \\
&\leq \left(\sqrt{(S_k - S_{k-1}) + \left(\sqrt{S_{k-1}} + \sum_{i=1}^{k-1} a_i \right)^2} + a_k \right)^2 \quad \text{(by induction)} \\
&\leq \left(\sqrt{S_k} + \sum_{i=1}^{k} a_i \right)^2.
\end{aligned}$$

The last inequality is obtained by developing the square $\left(\sqrt{S_{k-1} + \sum_{i=1}^{k-1} a_i} \right)^2$ and use the increasing assumption $S_{k-1} \leq S_k$. ■

Lemma 21 (Growth of the sequence $(A_k)_{k\geq 0}$)
Let $(A_k)_{k\geq 0}$ be the sequence defined in (14) where $(\alpha_k)_{k\geq 0}$ is produced by (10) with $\alpha_0 = 1$ and $\mu = 0$. Then, we have the following bounds for all $k \geq 0$,

$$\frac{2}{(k+2)^2} \leq A_k \leq \frac{4}{(k+2)^2}.$$

Proof The righthand side is directly obtained from Lemma 4 by noticing that $\gamma_0 = \kappa$ with the choice of α_0. Using the recurrence of α_k, we have for all $k \geq 1$,

$$\alpha_k^2 = (1 - \alpha_k)\alpha_{k-1}^2 = \prod_{i=1}^{k} (1 - \alpha_i)\alpha_0^2 = A_k \leq \frac{4}{(k+2)^2}.$$

Thus, $\alpha_k \leq \frac{2}{k+2}$ for all $k \geq 1$ (it is also true for $k = 0$). We now have all we need to conclude the lemma:

$$A_k = \prod_{i=1}^{k} (1 - \alpha_i) \geq \prod_{i=1}^{k} \left(1 - \frac{2}{i+2}\right) = \frac{2}{(k+2)(k+1)} \geq \frac{2}{(k+2)^2}.$$

■

Appendix B. Proofs of Auxiliary Results

B.1 Proof of Lemma 2

Proof Let us introduce the notation $h'(z) \triangleq \frac{1}{\eta}(z - [z]_\eta)$ for the gradient mapping at z. The first order conditions of the convex problem defining $[z]_\eta$ give

$$h'(z) - \nabla h_0(z) \in \partial\psi([z]_\eta).$$

Then, we may define

$$\begin{aligned}
u &\triangleq \frac{1}{\eta}(z - [z]_\eta) - (\nabla h_0(z) - \nabla h_0([z]_\eta)), \\
&= h'(z) - \nabla h_0(z) + \nabla h_0([z]_\eta) \in \partial h([z]_\eta).
\end{aligned}$$

Then, by strong convexity,

$$\begin{aligned}
h^* &\geq h([z]_\eta) + u^\top (p(x) - [z]_\eta) + \frac{\kappa + \mu}{2} \|p(x) - [z]_\eta\|^2 \\
&\geq h([z]_\eta) - \frac{1}{2(\kappa + \mu)} \|u\|^2.
\end{aligned}$$

Moreover,

$$\begin{aligned}
\|u\|^2 &= \left\| \frac{1}{\eta}(z - [z]_\eta) \right\|^2 - \frac{2}{\eta} \langle z - [z]_\eta, \nabla h_0(z) - \nabla h_0([z]_\eta) \rangle + \|\nabla h_0(z) - \nabla h_0([z]_\eta)\|^2 \\
&\leq \|h'(z)\|^2 - \|\nabla h_0(z) - \nabla h_0([z]_\eta)\|^2 \\
&\leq \|h'(z)\|^2,
\end{aligned}$$

where the first inequality comes from the relation (Nesterov, 2004, Theorem 2.1.5) using the fact h_0 is $(1/\eta)$-smooth

$$\|\nabla h_0(z) - \nabla h_0([z]_\eta)\|^2 \leq \frac{1}{\eta}\langle z - [z]_\eta, \nabla h_0(z) - \nabla h_0([z]_\eta)\rangle.$$

Thus,

$$h([z]_\eta) - h^* \leq \frac{1}{2(\kappa+\mu)}\|u\|^2 \leq \frac{1}{2(\kappa+\mu)}\|h'(z)\|^2.$$

As a result,

$$\|h'(z)\| \leq \sqrt{2\kappa\varepsilon} \quad \Rightarrow \quad h([z]_\eta) - h^* \leq \varepsilon.$$

■

B.2 Proof of Proposition 5

Proof We simply use Theorem 3 and specialize it to the choice of parameters. The initialization $\alpha_0 = \sqrt{q}$ leads to a particularly simple form of the algorithm, where $\alpha_k = \sqrt{q}$ for all $k \geq 0$. Therefore, the sequence $(A_k)_{k\geq 0}$ from Theorem 3 is also simple since we indeed have $A_k = (1-\sqrt{q})^k$. Then, we remark that $\gamma_0 = \mu(1-\sqrt{q})$ and thus, by strong convexity of f,

$$S_0 = (1-\sqrt{q})\left(f(x_0) - f^* + \frac{\mu}{2}\|x_0 - x^*\|^2\right) \leq 2(1-\sqrt{q})(f(x_0) - f^*).$$

Therefore,

$$\begin{aligned}
\sqrt{S_0} + 3\sum_{j=1}^k \sqrt{\frac{\varepsilon_j}{A_{j-1}}} &\leq \sqrt{2(1-\sqrt{q})(f(x_0)-f^*)} + 3\sum_{j=1}^k \sqrt{\frac{\varepsilon_j}{A_{j-1}}} \\
&= \sqrt{2(1-\sqrt{q})(f(x_0)-f^*)}\left[1 + \sum_{j=1}^k \underbrace{\left(\sqrt{\frac{1-\rho}{1-\sqrt{q}}}\right)^j}_{\eta}\right] \\
&= \sqrt{2(1-\sqrt{q})(f(x_0)-f^*)}\,\frac{\eta^{k+1}-1}{\eta-1} \\
&\leq \sqrt{2(1-\sqrt{q})(f(x_0)-f^*)}\,\frac{\eta^{k+1}}{\eta-1}.
\end{aligned}$$

Therefore, Theorem 3 combined with the previous inequality gives us

$$\begin{aligned}
f(x_k) - f^* &\leq 2A_{k-1}(1-\sqrt{q})(f(x_0)-f^*)\left(\frac{\eta^{k+1}}{\eta-1}\right)^2 \\
&= 2\left(\frac{\eta}{\eta-1}\right)^2(1-\rho)^k(f(x_0)-f^*) \\
&= 2\left(\frac{\sqrt{1-\rho}}{\sqrt{1-\rho}-\sqrt{1-\sqrt{q}}}\right)^2(1-\rho)^k(f(x_0)-f^*) \\
&= 2\left(\frac{1}{\sqrt{1-\rho}-\sqrt{1-\sqrt{q}}}\right)^2(1-\rho)^{k+1}(f(x_0)-f^*).
\end{aligned}$$

Since $\sqrt{1-x}+\frac{x}{2}$ is decreasing in $[0,1]$, we have $\sqrt{1-\rho}+\frac{\rho}{2} \geq \sqrt{1-\sqrt{q}}+\frac{\sqrt{q}}{2}$. Consequently,

$$f(x_k) - f^* \leq \frac{8}{(\sqrt{q}-\rho)^2}(1-\rho)^{k+1}(f(x_0)-f^*).$$

■

B.3 Proof of Proposition 6

Proof The initialization $\alpha_0 = 1$ leads to $\gamma_0 = \kappa$ and $S_0 = \frac{\kappa}{2}\|x^* - x_0\|^2$. Then,

$$\begin{aligned}
\sqrt{\frac{\gamma_0}{2}\|x_0 - x^*\|^2} + 3\sum_{j=1}^k \sqrt{\frac{\varepsilon_j}{A_{j-1}}} &\leq \sqrt{\frac{\kappa}{2}\|x_0 - x^*\|^2} + 3\sum_{j=1}^k \sqrt{\frac{(j+1)^2\varepsilon_j}{2}} \quad \text{(from Lemma 21)} \\
&\leq \sqrt{\frac{\kappa}{2}\|x_0 - x^*\|^2} + \sqrt{f(x_0)-f^*}\left(\sum_{j=1}^k \frac{1}{(j+1)^{1+\gamma/2}}\right),
\end{aligned}$$

where the last inequality uses Lemma 21 to upper-bound the ratio ε_j/A_j. Moreover,

$$\sum_{j=1}^k \frac{1}{(j+1)^{1+\gamma/2}} \leq \sum_{j=2}^\infty \frac{1}{j^{1+\gamma/2}} \leq \int_1^\infty \frac{1}{x^{1+\gamma/2}}\,\mathrm{d}x = \frac{2}{\gamma}.$$

Then applying Theorem 3 yields

$$\begin{aligned}
f(x_k) - f^* &\leq A_{k-1}\left(\sqrt{\frac{\kappa}{2}\|x_0 - x^*\|^2} + \frac{2}{\gamma}\sqrt{f(x_0)-f^*}\right)^2 \\
&\leq \frac{8}{(k+1)^2}\left(\frac{\kappa}{2}\|x_0 - x^*\|^2 + \frac{4}{\gamma^2}(f(x_0)-f^*)\right).
\end{aligned}$$

The last inequality uses $(a+b)^2 \leq 2(a^2+b^2)$. ■

B.4 Proof of Lemma 11

Proof We abbreviate $\tau_{\mathcal{M}}$ by τ and $C = C_{\mathcal{M}}(h(z_0) - h^*)$ to simplify the notation. Set

$$T_0 = \frac{1}{\tau} \log\left(\frac{1}{1-e^{-\tau}}\frac{C}{\varepsilon}\right).$$

For any $t \geq 0$, we have

$$\mathbb{E}[h(z_t) - h^*] \leq C(1-\tau)^t \leq C\, e^{-t\tau}.$$

By Markov's inequality,

$$\mathbb{P}[h(z_t) - h^* > \varepsilon] = \mathbb{P}[T(\varepsilon) > t] \leq \frac{\mathbb{E}[h(z_t) - h^*]}{\varepsilon} \leq \frac{C\, e^{-t\tau}}{\varepsilon}.$$

Together with the fact $\mathbb{P} \leq 1$ and $t \geq 0$. We have

$$\mathbb{P}[T(\varepsilon) \geq t+1] \leq \min\left\{\frac{C}{\varepsilon}e^{-t\tau}, 1\right\}.$$

Therefore,

$$\begin{aligned}
\mathbb{E}[T(\varepsilon)] &= \sum_{t=1}^{\infty} \mathbb{P}[T(\varepsilon) \geq t] = \sum_{t=1}^{T_0} \mathbb{P}[T(\varepsilon) \geq t] + \sum_{t=T_0+1}^{\infty} \mathbb{P}[T(\varepsilon) \geq t] \\
&\leq T_0 + \sum_{t=T_0}^{\infty} \frac{C}{\varepsilon} e^{-t\tau} = T_0 + \frac{C}{\varepsilon} e^{-T_0\tau} \sum_{t=0}^{\infty} e^{-t\tau} \\
&= T_0 + \frac{C}{\varepsilon}\frac{e^{-\tau T_0}}{1-e^{-\tau}} = T_0 + 1.
\end{aligned}$$

A simple calculation shows that for any $\tau \in (0,1)$, $\frac{\tau}{2} \leq 1 - e^{-\tau}$ and then

$$\mathbb{E}[T(\varepsilon)] \leq T_0 + 1 = \frac{1}{\tau}\log\left(\frac{1}{1-e^{-\tau}}\frac{C}{\varepsilon}\right) + 1 \leq \frac{1}{\tau}\log\left(\frac{2C}{\tau\varepsilon}\right) + 1.$$

■

B.5 Proof of coerciveness property of the proximal operator

Lemma 22 *Given a μ-strongly convex function $f : \mathbb{R}^p \to \mathbb{R}$ and a positive parameter $\kappa > 0$. For any x, $y \in \mathbb{R}^p$, the following inequality holds,*

$$\frac{\kappa}{\kappa+\mu}\langle y - x, p(y) - p(x)\rangle \geq \|p(y) - p(x)\|^2,$$

where $p(x) = \arg\min_{z \in \mathbb{R}^p} \left\{ f(z) + \frac{\kappa}{2}\|z - x\|^2 \right\}$.

Proof By the definition of $p(x)$, we have $0 \in \partial f(p(x)) + \kappa(p(x) - x)$, meaning that $\kappa(x - p(x)) \in \partial f(p(x))$. By strong convexity of f,

$$\langle \kappa(y - p(y)) - \kappa(x - p(x)), p(y) - p(x)\rangle \geq \mu\|p(y) - p(x)\|^2.$$

Rearranging the terms yields the desired inequality. ■

As a consequence,

$$\begin{aligned}
&\left\|\frac{\kappa}{\kappa+\mu}(y_k - y_{k-1}) - (p(y_k) - p(y_{k-1}))\right\|^2 \\
&\quad= \left\|\frac{\kappa}{\kappa+\mu}(y_k - y_{k-1})\right\|^2 - 2\frac{\kappa}{\kappa+\mu}\langle y_k - y_{k-1}, p(y_k) - p(y_{k-1})\rangle + \|p(y_k) - p(y_{k-1})\|^2 \\
&\quad\leq \left\|\frac{\kappa}{\kappa+\mu}(y_k - y_{k-1})\right\|^2 \\
&\quad\leq \|y_k - y_{k-1}\|^2.
\end{aligned}$$

Appendix C. Catalyst for MISO/Finito/SDCA

In this section, we present the application of Catalyst to MISO/Finito (Mairal, 2015; Defazio et al., 2014b), which may be seen as a variant of SDCA (Shalev-Shwartz and Zhang, 2016). The reason why these algorithms require a specific treatment is due to the fact that their linear convergence rates are given in a different form than (12); specifically, Theorem 4.1 of Lin et al. (2015a) tells us that MISO produces a sequence of iterates $(z_t)_{t\geq 0}$ for minimizing the auxiliary objective $h(z) = f(z) + \frac{\kappa}{2}\|z - y\|^2$ such that

$$\mathbb{E}[h(z_t)] - h^* \leq C_{\mathcal{M}}(1 - \tau_{\mathcal{M}})^{t+1}(h^* - d_0(z_0)),$$

where d_0 is a lower-bound of h defined as the sum of a simple quadratic function and the composite regularization ψ. More precisely, these algorithms produce a sequence $(d_t)_{t\geq 0}$ of such lower-bounds, and the iterate z_t is obtained by minimizing d_t in closed form. In particular, z_t is obtained from taking a proximal step at a well chosen point w_t, providing the following expression,

$$z_t = \mathrm{prox}_{\psi/(\kappa+\mu)}(w_t).$$

Then, linear convergence is achieved for the duality gap

$$\mathbb{E}[h(z_t) - h^*] \leq \mathbb{E}[h(z_t) - d_t(z_t)] \leq C_{\mathcal{M}}(1 - \tau_{\mathcal{M}})^t(h^* - d_0(z_0)).$$

Indeed, the quantity $h(z_t) - d_t(z_t)$ is a natural upper-bound on $h(z_t) - h^*$, which is simple to compute, and which can be naturally used for checking the criterions (C1) and (C2). Consequently, the expected complexity of solving a given problem is slightly different compared to Lemma 11.

Lemma 23 (Accuracy vs. complexity) *Let us consider a strongly convex objective h and denote $(z_t)_{t\geq 0}$ the sequence of iterates generated by MISO/Finito/SDCA. Consider the*

complexity $T(\varepsilon) = \inf\{t \geq 0, h(z_t) - d_t(z_t) \leq \varepsilon\}$*, where* $\varepsilon > 0$ *is the target accuracy and* h^* *is the minimum value of* h*. Then,*

$$\mathbb{E}[T(\varepsilon)] \leq \frac{1}{\tau_{\mathcal{M}}} \log\left(\frac{2C_{\mathcal{M}}(h^* - d_0(z_0))}{\tau_{\mathcal{M}}\varepsilon}\right) + 1,$$

where d_0 *is a lower bound of* f *built by the algorithm.*

For the convergence analysis, the outer-loop complexity does not change as long as the algorithm finds approximate proximal points satisfying criterions (C1) and (C2). It is then sufficient to control the inner loop complexity. As we can see, we now need to bound the dual gap $h^* - d_0(z_0)$ instead of the primal gap $h(z_0) - h^*$, leading to slightly different warm start strategies. Here, we show how to restart MISO/Finito.

Proposition 24 (Warm start for criterion (C1)) *Consider applying Catalyst with the same parameter choices as in Proposition 12 to MISO/Finito. At iteration* $k+1$ *of Algorithm 2, assume that we are given the previous iterate* x_k *in* $p^{\varepsilon_k}(y_{k-1})$*, the corresponding dual function* $d(x)$ *and its prox-center* w_k *satisfying* $x_k = \mathrm{prox}_{\psi/(\kappa+\mu)}(w_k)$*. Then, initialize the sequence* $(z_t)_{t\geq 0}$ *for minimizing* $h_{k+1} = f + \frac{\kappa}{2}\|\cdot - y_k\|^2$ *with,*

$$z_0 = \mathrm{prox}_{\psi/(\kappa+\mu)}\left(w_k + \frac{\kappa}{\kappa+\mu}(y_k - y_{k-1})\right),$$

and initialize the dual function as

$$d_0(x) = d(x) + \frac{\kappa}{2}\|x - y_k\|^2 - \frac{\kappa}{2}\|x - y_{k-1}\|^2.$$

Then,

1. *when* f *is* μ*-strongly convex, we have* $h^*_{k+1} - d_0(z_0) \leq C\varepsilon_{k+1}$ *with the same constant as in (21) and (22), where* d_0 *is the dual function corresponding to* z_0*;*
2. *when* f *is convex with bounded level sets, there exists a constant* $B > 0$ *identical to the one of (23) such that*
$$h^*_{k+1} - d_0(z_0) \leq B.$$

Proof The proof is given in Lemma D.5 of Lin et al. (2015a), which gives

$$h^*_{k+1} - d_0(z_0) \leq \varepsilon_k + \frac{\kappa^2}{2(\kappa+\mu)}\|y_k - y_{k-1}\|^2.$$

This term is smaller than the quantity derived from (24), leading to the same upper bound. ■

Proposition 25 (Warm start for criterion (C2)) *Consider applying Catalyst with the same parameter choices as in Proposition 15 to MISO/Finito. At iteration* $k+1$ *of Algorithm 2, we assume that we are given the previous iterate* x_k *in* $g^{\delta_k}(y_{k-1})$ *and the*

corresponding dual function $d(x)$*. Then, initialize the sequence* $(z_t)_{t\geq 0}$ *for minimizing* $h_{k+1} = f + \frac{\kappa}{2}\|\cdot - y_k\|^2$ *by*

$$z_0 = \mathrm{prox}_{\psi/(\kappa+\mu)}\left(y_k - \frac{1}{\kappa+\mu}\nabla f_0(y_k)\right),$$

where $f = f_0 + \psi$ *and* f_0 *is the smooth part of* f*, and set the dual function* d_0 *by*

$$d_0(x) = f_0(y_k) + \langle \nabla f_0(y_k), x - y_k\rangle + \frac{\kappa+\mu}{2}\|x - y_k\|^2 + \psi(x).$$

Then,

$$h^*_{k+1} - d_0(z_0) \leq \frac{(L+\kappa)^2}{2(\mu+\kappa)}\|p(y_k) - y_k\|^2. \tag{31}$$

Proof Since $p(y_k)$ is the minimum of h_{k+1}, the optimality condition provides

$$-\nabla f_0(p(y_k)) - \kappa(p(y_k) - y_k) \in \partial\psi(p(y_k)).$$

Thus, by convexity,

$$\begin{aligned}
&\psi(p(y_k)) + \langle -\nabla f_0(p(y_k)) - \kappa(p(y_k) - y_k), z_0 - p(y_k)\rangle \leq \psi(z_0),\\
&f_0(p(y_k)) + \frac{\kappa}{2}\|p(y_k) - y_k\|^2 + \langle \nabla f_0(p(y_k)) + \kappa(p(y_k) - y_k), y_k - p(y_k)\rangle \leq f_0(y_k).
\end{aligned}$$

Summing up gives

$$h^*_{k+1} \leq f_0(y_k) + \psi(z_0) + \langle \nabla f_0(p(y_k)) + \kappa(p(y_k) - y_k), z_0 - y_k\rangle.$$

As a result,

$$\begin{aligned}
h^*_{k+1} - d_0(z_0) &\leq f_0(y_k) + \psi(z_0) + \langle \nabla f_0(p(y_k)) + \kappa(p(y_k) - y_k), z_0 - y_k\rangle - d_0(z_0)\\
&= \langle \nabla f_0(p(y_k)) + \kappa(p(y_k) - y_k) - \nabla f_0(y_k), z_0 - y_k\rangle - \frac{\kappa+\mu}{2}\|z_0 - y_k\|^2\\
&\leq \frac{1}{2(\kappa+\mu)}\|\nabla \underbrace{f_0(p(y_k)) - \nabla f_0(y_k)}_{\|\cdot\|\leq L\|p(y_k)-y_k\|} + \kappa(p(y_k) - y_k)\|^2\\
&\leq \frac{(L+\kappa)^2}{2(\mu+\kappa)}\|p(y_k) - y_k\|^2.
\end{aligned}$$

■

The bound obtained from (31) is similar to the one form Proposition 15, and differs only in the constant factor. Thus, the inner loop complexity in Section 4.2.2 still holds for MISO/Finito up to a constant factor. As a consequence, the global complexity of MISO/Finito applied to Catalyst is similar to one obtained by SVRG, yielding an acceleration for ill-conditioned problems.

References

A. Agarwal and L. Bottou. A lower bound for the optimization of finite sums. In *Proceedings of the International Conferences on Machine Learning (ICML)*, 2015.

Z. Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. In *Proceedings of Symposium on Theory of Computing (STOC)*, 2017.

Y. Arjevani and O. Shamir. Dimension-free iteration complexity of finite sum optimization problems. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

A. Auslender. Numerical methods for nondifferentiable convex optimization. In *Nonlinear Analysis and Optimization*, volume 30, pages 102–126. Springer, 1987.

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

D. P. Bertsekas. *Convex Optimization Algorithms*. Athena Scientific, 2015.

A. Chambolle and T. Pock. A remark on accelerated block coordinate descent for computing the proximity operators of a sum of convex functions. *SMAI Journal of Computational Mathematics*, 1:29–54, 2015.

R. Correa and C. Lemaréchal. Convergence of some algorithms for convex minimization. *Mathematical Programming*, 62(1):261–275, 1993.

A. Defazio. A simple practical accelerated method for finite sums. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

A. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems (NIPS)*, 2014a.

A. Defazio, J. Domke, and T. S. Caetano. Finito: A faster, permutable incremental gradient method for big data problems. In *Proceedings of the International Conferences on Machine Learning (ICML)*, 2014b.

O. Devolder, F. Glineur, and Y. Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1-2):37–75, 2014.

R. Frostig, R. Ge, S. M. Kakade, and A. Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *Proceedings of the International Conferences on Machine Learning (ICML)*, 2015.

M. Fuentes, J. Malick, and C. Lemaréchal. Descentwise inexact proximal algorithms for smooth optimization. *Computational Optimization and Applications*, 53(3):755–769, 2012.

P. Giselsson and M. Fält. Nonsmooth minimization using smooth envelope functions. *arXiv:1606.01327*, 2016.

O. Güler. On the convergence of the proximal point algorithm for convex minimization. *SIAM Journal on Control and Optimization*, 29(2):403–419, 1991.

O. Güler. New proximal point algorithms for convex minimization. *SIAM Journal on Optimization*, 2(4):649–664, 1992.

B. He and X. Yuan. An accelerated inexact proximal point algorithm for convex minimization. *Journal of Optimization Theory and Applications*, 154(2):536–548, 2012.

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

G. Lan and Y. Zhou. An optimal randomized incremental gradient method. *Mathematical Programming*, 2017.

C. Lemaréchal and C. Sagastizábal. Practical aspects of the Moreau–Yosida regularization: Theoretical preliminaries. *SIAM Journal on Optimization*, 7(2):367–385, 1997.

H. Lin, J. Mairal, and Z. Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2015a.

Q. Lin, Z. Lu, and L. Xiao. An accelerated randomized proximal coordinate gradient method and its application to regularized empirical risk minimization. *SIAM Journal on Optimization*, 25(4):2244–2273, 2015b.

J. Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2):829–855, 2015.

J. Mairal. End-to-end kernel learning with supervised convolutional kernel networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

B. Martinet. Brève communication. Régularisation d'inéquations variationnelles par approximations successives. *Revue française d'informatique et de recherche opérationnelle, série rouge*, 4(3):154–158, 1970.

J.-J. Moreau. Fonctions convexes duales et points proximaux dans un espace hilbertien. *CR Acad. Sci. Paris Sér. A Math*, 255:2897–2899, 1962.

A. Nemirovskii and D. B. Yudin. *Problem complexity and method efficiency in optimization*. John Wiley & Sons, 1983.

Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer, 2004.

Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

C. Paquette, H. Lin, D. Drusvyatskiy, J. Mairal, and Z. Harchaoui. Catalyst acceleration for gradient-based non-convex optimization. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

N. Parikh and S. P. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2014.

P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014.

R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

S. Salzo and S. Villa. Inexact and accelerated proximal point algorithms. *Journal of Convex Analysis*, 19(4):1167–1192, 2012.

M. Schmidt, N. Le Roux, and F. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 160(1):83–112, 2017.

D. Scieur, A. d' Aspremont, and F. Bach. Regularized nonlinear acceleration. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

S. Shalev-Shwartz and T. Zhang. Proximal stochastic dual coordinate ascent. *preprint arXiv:1211.2717*, 2012.

S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Mathematical Programming*, 155(1):105–145, 2016.

A. Sidi. *Vector Extrapolation Methods with Applications*. Society for Industrial and Applied Mathematics, 2017.

M. V. Solodov and B. F. Svaiter. A unified framework for some inexact proximal point algorithms. *Numerical Functional Analysis and Optimization*, 22(7-8):1013–1035, 2001.

A. Themelis, L. Stella, and P. Patrinos. Forward-backward envelope for the sum of two nonconvex functions: Further properties and nonmonotone line-search algorithms. *arXiv:1606.06256*, 2016.

B. E. Woodworth and N. Srebro. Tight complexity bounds for optimizing composite objectives. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

L. Xiao and T. Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

K. Yosida. Functional analysis. *Berlin-Heidelberg*, 1980.

Y. Zhang and L. Xiao. Stochastic primal-dual coordinate method for regularized empirical risk minimization. In *Proceedings of the International Conferences on Machine Learning (ICML)*, 2015.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

Journal of Machine Learning Research 18 (2018) 1-29 Submitted 7/17; Revised 11/17; Published 4/18

Gaussian Lower Bound for the Information Bottleneck Limit

Amichai Painsky AMICHAI.PAINSKY@MAIL.HUJI.AC.IL

Naftali Tishby TISHBY@CS.HUJI.AC.IL

School of Computer Science and Engineering and
The Interdisciplinary Center for Neural Computation
The Hebrew University of Jerusalem
Givat Ram, Jerusalem 91904, Israel

Editor: Samuel Kaski

Abstract

The Information Bottleneck (IB) is a conceptual method for extracting the most compact, yet informative, representation of a set of variables, with respect to the target. It generalizes the notion of minimal sufficient statistics from classical parametric statistics to a broader information-theoretic sense. The IB curve defines the optimal trade-off between representation complexity and its predictive power. Specifically, it is achieved by minimizing the level of mutual information (MI) between the representation and the original variables, subject to a minimal level of MI between the representation and the target. This problem is shown to be in general NP hard. One important exception is the multivariate Gaussian case, for which the Gaussian IB (GIB) is known to obtain an analytical closed form solution, similar to Canonical Correlation Analysis (CCA). In this work we introduce a Gaussian lower bound to the IB curve; we find an embedding of the data which maximizes its "Gaussian part", on which we apply the GIB. This embedding provides an efficient (and practical) representation of any arbitrary data-set (in the IB sense), which in addition holds the favorable properties of a Gaussian distribution. Importantly, we show that the optimal Gaussian embedding is bounded from above by non-linear CCA. This allows a fundamental limit for our ability to Gaussianize arbitrary data-sets and solve complex problems by linear methods.

Keywords: Information Bottleneck, Canonical Correlations, ACE, Gaussianization, Mutual Information Maximization, Infomax

1. Introduction

The problem of extracting the relevant aspects of complex data is a long standing staple in statistics and machine learning. The Information Bottleneck (IB) method, presented by Tishby et al. (1999), approaches this problem by extending its classical notion to a broader information-theoretic setup. Specifically, given the joint distribution of a set of explanatory variables $\underline{X}$ and a target variable $\underline{Y}$ (which may also be of a higher dimension), the IB method strives to find the most compressed representation of $\underline{X}$, while preserving information about $\underline{Y}$. Thus, $\underline{Y}$ implicitly regulates the compression of $\underline{X}$, so that its compressed representation maintains a level of relevance as explanatory variables with respect to $\underline{Y}$.

The IB problem is formally defined as follows:

$$\begin{aligned} \min_{P(\underline{T}|\underline{X})} \quad & I(\underline{X};\underline{T}) \\ \text{subject to} \quad & I(\underline{T};\underline{Y}) \geq I_Y \end{aligned} \tag{1}$$

where $\underline{T}$ is the compressed representation of $\underline{X}$ and the minimization is over the mapping of $\underline{X}$ to $\underline{T}$, defined by the conditional probability $P(\underline{T}|\underline{X})$. Here, I_Y is a constant parameter that sets the level of information to be preserved between the compressed representation and the target. Solving this problem for a range of I_Y values defines the *IB curve* – a continuous concave curve which provides the optimal trade-off between representation complexity (regarded as $I(\underline{X};\underline{T})$) and predictive power ($I(\underline{T};\underline{Y})$).

The IB method showed to be a powerful tool in a variety of machine learning domains and related areas (Slonim and Tishby, 2000; Friedman et al., 2001; Sinkkonen and Kaski, 2002; Slonim et al., 2005; Hecht et al., 2009). It is also applicable to other fields such as neuroscience (Schneidman et al., 2001) and optimal control (Tishby and Polani, 2011). Recently, Tishby and Zaslavsky (2015) and Shwartz-Ziv and Tishby (2017) demonstrated its abilities in analyzing and optimizing the performance of deep neural networks.

Generally speaking, solving the IB problem (1) for an arbitrary joint distribution is not a simple task. In the introduction of the IB method, Tishby et al. (1999) defined a set of self-consistent equations which formulate the necessary conditions for the optimal solution of (1). Further, they provided an iterative Arimoto– Blahut like algorithm which shows to converge to local optimum. In general, these equations do not hold a tractable solution and are usually approximated by different means (Slonim, 2002). An extensive attention was given to the simpler categorical setup, where the IB curve is somewhat easier to approximate. Here, $\underline{X}$ and $\underline{Y}$ take values on a finite set and $\underline{T}$ represents (soft and informative) clusters of $\underline{X}$ (Slonim, 2002). Naturally, the IB problem also applies for continuous variables. In this case, approximating the solution to the self-consistent equations is even more involved. A special exception is the Gaussian case, where $\underline{X}$ and $\underline{Y}$ are assumed to follow a jointly normal distribution and the *Gaussian IB* problem (GIB) is analytically solved by linear projections to the canonical correlation vector space (Chechik et al., 2005). However, evaluating the IB curve for arbitrary continuous random variables is still considered a highly complicated task where most attempts focus on approximating or bounding it (Rey and Roth, 2012; Chalk et al., 2016). A detailed discussion regarding currently known methods is provided in the following section.

In this work we present a novel Gaussian lower bound to the IB curve, which applies to all types of random variables (continuous, nominal and categorical). Our bound strives to maximize the "jointly Gaussian part" of the data and apply the analytical GIB to it. Specifically, we seek for two transformations, $\underline{U} = \phi(\underline{X})$ and $\underline{V} = \psi(\underline{Y})$ so that $\underline{U}$ and $\underline{V}$ are highly correlated and "as jointly Gaussian as possible". In addition, we ask that the transformations preserve as much information as possible between $\underline{X}$ and $\underline{Y}$. This way, we maximize the portion of the data that can be explained by linear means, $I(\underline{U};\underline{V}) \leq I(\underline{X};\underline{Y})$, specifically using the GIB.

In fact, our results go beyond the specific context of the information bottleneck. In this work we tackle the fundamental question of linearizing non-linear problems. In other words, we ask ourselves whether it is possible to "push" all the information in the data

to its second moments. This problem has received a great amount of attention over the years. For example, Schneidman et al. (2006) discuss this problem in the context of neural networks; they provide preliminary evidence that in the vertebrate retina, weak pairwise correlations may describe the collective (non-linear) behavior of neurons. In this work, we provide both fundamental limits and constructive algorithms for maximizing the part of the data that can be optimally analyzed by linear means. This basic property holds both theoretical and practical implications, as it defines the maximal portion which allows favorable analytical properties in many applications. Interestingly, we show that even if we allow the transformations $\underline{U} = \phi(\underline{X})$ and $\underline{V} = \psi(\underline{Y})$ to increase the dimensions of $\underline{X}$ and $\underline{Y}$, our ability to linearize the problem is still limited, and governed by the non-linear canonical correlations (Breiman and Friedman, 1985) of the original variables.

Our suggested approach may also be viewed as an extension of the *Shannon lower bound* (Cover and Thomas, 2012), for evaluating the mutual information. In his seminal work, Shannon provided an analytical Gaussian lower bound for the generally involved rate distortion function. He showed that the rate distortion function $R(D)$ can be bounded from below by $h(X) - \frac{1}{2}\log(2\pi eD)$ where X is the compressed source, $h(X)$ is its corresponding deferential entropy and $\frac{1}{2}\log(2\pi eD)$ is the differential entropy of an independent Gaussian noise with a maximal distortion level D. This bound holds some favorable theoretical properties (Cover and Thomas, 2012) and serves as one of the most basic tools for approximating the rate distortion function to this very day. In this work, we use a similar rationale and derive a Gaussian lower bound for the mutual information of two random variables, which holds an analytical expression just like the Shannon's bound. We then extend our result to the entire IB curve and discuss its theoretical properties and practical considerations. A matlab implementation of our suggested approach is publicly available at the first author's web-page[1].

The rest of this manuscript is organized as follows: In Section 2 we review previous work on the IB method for continuous random variables. Section 3 defines our suggested lower bound and formulates it as an optimization problem. We then propose a set of solutions and bounds to this problem, as we distinguish between the easier univariate case (Section 4) and the more involved multivariate case (Section 5). Finally, in Section 6 we extend our results to the entire IB curve.

2. Related work

As discussed in the previous section, solving the IB problem for continuous variables is in general a difficult task. A special exception is where $\underline{X}$ and $\underline{Y}$ follow a jointly normal distribution. Chechik et al. (2005) show that in this case, the Gaussian IB problem (GIB) is solved by a noisy linear projection, $T = A\underline{X} + \underline{\zeta}$. Specifically, assume that $\underline{X}$ and $\underline{Y}$ are of dimensions n_X and n_Y respectively and denote the covariance matrix of $\underline{X}$ as $C_{\underline{X}}$ while the conditional covariance matrix of $\underline{X}|\underline{Y}$ is $C_{\underline{X}|\underline{Y}}$. Then, $\underline{\zeta}$ is a Gaussian random vector with a zero mean and a unit covariance matrix, independent of $\underline{X}$. The matrix A is defined as follows:

$$A = \begin{Bmatrix} [0^T;\ldots;0^T] & 0 \le \beta \le \beta_1^C \\ [a_1 v_1^T; 0^T; \ldots; 0^T] & \beta_1^C \le \beta \le \beta_2^C \\ [a_1 v_1^T; a_2 v_2^T; 0^T; \ldots; 0^T] & \beta_2^C \le \beta \le \beta_3^C \\ \vdots & \vdots \end{Bmatrix}. \quad (2)$$

where $\{v_1^T, v_2^T, \ldots, v_{n_x}^T\}$ are the left eigenvectors of $C_{\underline{X}|\underline{Y}} C_{\underline{X}}^{-1}$, sorted by their corresponding ascending eigenvalues $\lambda_1, \ldots, \lambda_{n_X}$, $\beta_I^C = \frac{1}{\lambda_i}$ are the critical β values, a_i are defined by $a_i = \sqrt{\frac{\beta(1-\lambda_i)-1}{\lambda_i r_i}}$. $r_i = v_i^T C_{\underline{X}} v_i$ and 0^T is an n_X row vector of zeros. Notice that the critical values β correspond to the slope of the IB curve, as they represent the Lagrange multipliers of the IB problem.

Unfortunately, this solution is limited to jointly Gaussian random variables. In fact, it can be shown that a closed form analytical solution (for continuous random variables) may only exist under quite restrictive assumptions on the underlaying distribution. Moreover, as the IB curve is so challenging to evaluate in the general case, most known attempts either focus on extending the GIB to other distributions under varying assumptions, or approximate the IB curve by different means.

Rey and Roth (2012) reformulate the IB problem in terms of probabilistic copulas. They show that under a Gaussian copula assumption, an analytical solution (which extends the GIB) applies to joint distributions with arbitrary marginals. This formulation provides several interesting insights on the IB problem. However, its practical implications are quite limited as the Gaussian copula assumption is very restrictive. In fact, it implicitly requires that the joint distribution would maintain a Gaussian structure. As we show in the following sections, this assumption makes the problem significantly easier and does not hold in general.

Chalk et al. (2016) provide a lower bound to the IB curve by using an approximate variational scheme, analogous to variational expectation maximization. Their method relaxes the IB problem by restricting the class of distributions, $P(\underline{Y}|\underline{T})$ and $P(\underline{T})$ to a set of parametric models. This way, the relaxed IB problem may be solved in EM-like steps; their suggested algorithm iteratively maximizes the objective over the mappings (for fixed parameters) and then maximize the set of parameters, for fixed mappings. Chalk et al. (2016) show that this method can be effectively applied to "sparse" data in which $\underline{X}$ and $\underline{Y}$ are generated by sparsely occurring latent features. However, in the general case, their suggested bound strongly depends on the assumption that the chosen parametric models provide reasonable approximations for the optimal distributions. This assumption is obviously quite restrictive. Moreover, it is usually difficult to validate, as the optimal distributions are unknown. Kolchinsky et al. (2017) take a somewhat similar approach, as they suggest a variational upper bound to the IB curve. The main difference between the two methods relies on the variational approximation of objective, $I(\underline{X};\underline{Y})$. However, they are both prone to the same difficulties stated above.

Alemi et al. (2016) propose an additional variational inference method to construct a lower bound to the IB curve. Here, they re-parameterize the IB problem followed by Monte Carlo sampling, to get an unbiased estimate of the IB objective gradient. This allows them to apply deep neural networks in order to parameterize any given distribution. However,

1. https://sites.google.com/site/amichaipainsky/software

this method fails to provide guarantees on the obtained bound, as a result of the suggested stochastic gradient descent optimization approach.

Achille and Soatto (2018) relax the bottleneck problem by introducing an additional *total correlation* (TC) regularization term that strives to maximize the independence among the components of the representation T. They show that under the assumption that the Lagrange multipliers of the TC and MI constraints are identical, the relaxed problem may be solved by adding auxiliary variables. However, this assumption is usually invalid, and the suggested method fails to provide guarantees on the difference between the obtained objective and original IB formulation.

In this work we suggest a novel lower bound to the IB curve which provides both theoretical and practical guarantees. In addition, we introduce upper and lower bounds to our suggested solution that are very easy to attain. This way we allow immediate benchmarks to the IB curve using common off-the-shelf methods.

3. Problem formulation

Throughout this manuscript we use the following standard notation: underlines denote vector quantities, where their respective components are written without underlines but with index. For example, the components of the n-dimensional vector $\underline{X}$ are $X_1, X_2, \ldots X_n$. Random variables are denoted with capital letters while their realizations are denoted with the respective lower-case letters. The mutual information of two random variables is defined as $I(\underline{X};\underline{Y}) = h(\underline{X}) + h(\underline{Y}) - h(\underline{X},\underline{X})$ where $h(\underline{X}) = -\int_{\underline{X}} f_{\underline{X}}(\underline{x}) \log f_{\underline{X}}(\underline{x}) d\underline{x}$ is the differential entropy of $\underline{X}$ and $f_{\underline{X}}(\underline{x})$ is its probability density function.

We begin by introducing a Gaussian lower bound to the mutual information $I(\underline{X};\underline{Y})$. We then extend our result to the entire IB curve.

3.1 Problem statement

Let $\underline{X} \in \mathbb{R}^{d_x}, \underline{Y} \in \mathbb{R}^{d_y}$ be two multivariate random vectors with a joint cumulative distribution function (CDF) $F_{XY}(x,y)$ and mutual information $I(\underline{X},\underline{Y})$. In the following sections we focus on bounding $I(\underline{X},\underline{Y})$ from below with an analytical expression. Let $\underline{U} = \phi(\underline{X})$ and $\underline{V} = \psi(\underline{Y})$ be two transformations of $\underline{X}$ and $\underline{Y}$, respectively. Assume that $\underline{U} \in \mathbb{R}^{d_u}$ and $\underline{V} \in \mathbb{R}^{d_v}$ are *separately normally distributed*. This means that $\underline{U} \sim N\left(\mu_{\underline{U}}, C_{\underline{U}}\right)$ and $\underline{V} \sim N\left(\mu_{\underline{V}}, C_{\underline{V}}\right)$ but the vector $[\underline{U},\underline{V}]^T$ is not necessarily normally distributed. This allows us to derive the following basic inequality

$$\begin{aligned} I(\underline{X},\underline{Y}) \geq & I(\underline{U},\underline{V}) = h(\underline{U}) + h(\underline{V}) - h(\underline{U},\underline{V}) \geq \\ & h(\underline{U}) + h(\underline{V}) - h(\underline{U}_{jg},\underline{V}_{jg}) = \frac{1}{2}\log\left(\frac{\left|C_{[\underline{U},\underline{V}]}\right|}{|C_{\underline{U}}||C_{\underline{V}}|}\right) \end{aligned} \tag{3}$$

where the first inequality follows from the Data Processing lemma (Cover and Thomas, 2012) and the second inequality follows from $\left[\underline{U}_{jg},\underline{V}_{jg}\right]^T$ being jointly Gaussian (jg) distributed with the same covariance matrix as $[\underline{U},\underline{V}]^T$, $C_{[\underline{U}_{jg},\underline{V}_{jg}]} = C_{[\underline{U},\underline{V}]}$, so that $h(\underline{U}_{jg},\underline{V}_{jg}) \geq h(\underline{U},\underline{V})$ (Cover and Thomas, 2012). Notice that (3) can also be derived from an *information geometry* (IG) view point, as shown by Cardoso (2003).

Equality is attained in (3) iff $I(\underline{X},\underline{Y}) = I(\underline{U},\underline{V})$ (no information is lost in the transformation) and $\underline{U} = \phi(\underline{X})$, $\underline{V} = \psi(\underline{Y})$ are jointly normally distributed. In other words, in order to preserve all the information we must find ϕ and ψ that capture all the mutual information, and at the same time make $\underline{X}$ and $\underline{Y}$ jointly normal. This is obviously a complicated task as ϕ and ψ only operate on $\underline{X}$ and $\underline{Y}$ separately. Therefore, we are interested in maximizing this lower bound as much as possible:

$$\begin{aligned} \max_{\phi,\psi} \quad & \log\left(\frac{\left|C_{[\underline{U},\underline{V}]}\right|}{|C_{\underline{U}}||C_{\underline{V}}|}\right) \\ \text{subject to} \quad & \underline{U} = \phi(\underline{X}) \sim N\left(0, C_{\underline{U}}\right) \\ & \underline{V} = \psi(\underline{Y}) \sim N\left(0, C_{\underline{V}}\right) \end{aligned} \tag{4}$$

where the constraints imply that $\underline{U}$ and $\underline{V}$ are separately normally distributed random vectors with zero means and covariance matrices $C_{\underline{U}}$ and $C_{\underline{V}}$, respectively. In other words, we would like to maximize Cardoso (2003) IG bound by applying two transformations, ϕ and ψ, to the original variables. This would allow us to achieve a tighter result.

Notice that our objective is invariant to the means of $\underline{U},\underline{V}$ so they are chosen to be zero. In addition, it is easy to show that our objective is invariant to linear transformations of $\underline{U},\underline{V}$. This means we can equivalently assume that $C_{\underline{U}}, C_{\underline{V}}$ are identity covariance matrices. As shown by Kay (1992) and others (Klami and Kaski, 2005; Chechik et al., 2005), maximizing the objective of (4) is equivalent to maximizing the canonical correlations, $\text{cov}(U_i, V_i)$. Therefore, our problem may be written as

$$\begin{aligned} \max_{\phi,\psi} \quad & \sum_{i=1}^{k} E\left(U_i V_i\right) \\ \text{subject to} \quad & \underline{U} = \phi(\underline{X}) \sim N\left(0, I\right) \\ & \underline{V} = \psi(\underline{Y}) \sim N\left(0, I\right) \end{aligned} \tag{5}$$

where $k = \min\{d_u, d_v\}$. This problem may also be viewed as a variant of the well-known CCA problem (Hotelling, 1936), where we optimize over nonlinear transformations ϕ and ψ, and impose additional normality constraints. As in CCA, this problem can be solved iteratively by gradually finding the the optimal canonical components in each step (subject to the normality constraint), while maintaining orthogonality with the components that were previously found. For simplicity of the presentation we begin by solving (5) in the univariate (1-D) case. Then, we generalize to the multivariate case. In each of these setups we present a solution to the problem, followed by simpler upper and lower bounds.

4. The univariate case

In the univariate case we assume that $d_x = d_y = k = 1$. We seek ϕ, ψ such that

$$\begin{aligned} \max_{\phi,\psi} \quad & \rho = E(UV) \\ \text{subject to} \quad & U = \phi(X) \sim N(0,1) \\ & V = \psi(Y) \sim N(0,1). \end{aligned} \tag{6}$$

As a first step towards this goal, let us relax our problem by replacing the normality constraint with simpler second order statistics constraints,

$$\begin{aligned} \max_{\phi,\psi} \quad & \rho = E(UV) \\ \text{subject to} \quad & U = \phi(X),\ E(U) = 0,\ E(U^2) = 1 \\ & V = \psi(Y),\ E(V) = 0,\ E(V^2) = 1. \end{aligned} \tag{7}$$

As mentioned above, this problem is a non-linear extension of CCA, which traces back to early work by Lancaster (1963). As this problem is also a relaxed version of our original task (6), it may serve us as an upper bound. This means that the optimum of (7), denoted as ρ_{ub}, necessarily bound from above ρ_*, the optimum of (6).

4.1 Alternation Conditional Expectation (ACE)

Breiman and Friedman (1985) show that the optimal solution to (7) is achieved by a simple alternating conditional expectation procedure, named ACE. Assume that $\psi(Y)$ is fixed, known and satistfies the constraints. Then, we optimize (7) only over ϕ and by Cauchy-Schwarz inequality, we have that

$$E(\phi(X)\psi(Y)) = E_x\left(\phi(X)E(\psi(Y)|X)\right) \leq \sqrt{\text{var}(\phi(X))}\sqrt{\text{var}(E(\psi(Y)|X))}$$

with equality iff $\phi(X) = c \cdot E(\psi(Y)|X)$. Therefore, choosing the constant c to satisfy the unit variance constraint we achieve $\phi(X) = \frac{E(\psi(Y)|X)}{\sqrt{var(E(\psi(Y)|X))}}$. In the same manner we may fix $\phi(X)$ and attain $\psi(Y) = \frac{E(\phi(X)|Y)}{\sqrt{var(E(\phi(X)|Y))}}$. These coupled equations are in fact necessary conditions for the optimality of ϕ and ψ, leading to an alternating procedure in which at each step we fix one transformation and optimize the other. Breiman and Friedman (1985) prove that this procedure convergences to the global optimum using Hilbert space algebra. They show that the transformations ϕ and ψ may be represented in a zero-mean and finite variance Hilbert space, while the conditional expectation projection is linear, closed, and shown to be self-adjoint and compact under mild assumptions. Then, the coupled equations may be formulate as an eigen problem in the Hilbert space, for which there exists a unique and optimal solution.

The following lemma defines a strict connection between the non-linear canonical correlations and the Gaussinized IB problem.

Lemma 1 *Let ρ_{ub} be the solution to (7). If $I(X;Y) > -\log\left(1-\rho_{ub}^2\right)$, then there are no transformations ϕ, ψ such that $U = \phi(X)$ and $V = \psi(Y)$ are jointly normally distributed and preserve all of the mutual information, $I(X;Y)$.*

Proof Let ρ_* be the solution to (6). As mentioned above, $\rho_{ub} \geq \rho_*$. Therefore, $I(X;Y) > -\log\left(1-\rho_{ub}^2\right) > -\log\left(1-\rho_*^2\right)$. This means that the inequality (3) cannot be achieved with equality. Hence, there are no transformations $U = \phi(X)$ and $V = \psi(Y)$ so that U and V are jointly normal and preserve all of the mutual information, $I(X;Y)$. ■

Lemma 1 suggests that if the optimal transformations of the relaxed problem (which can be obtained by ACE) fails to capture all the mutual information between X and Y, then there

are no transformations that can project X and Y onto jointly normal variables without losing information. Moreover, notice that the maximal level of correlation ρ_{ub} cannot be further increased, even if we allow $\underline{U} = \phi(X)$ and $\underline{V} = \phi(Y)$ to reside in greater dimensions. This means that Lemma 1 holds for any $\phi : R^{d_x} \to R^{d_u}$ and $\psi : R^{d_y} \to R^{d_v}$, such that $d_u, d_v \geq 0$.

4.2 Alternating Gaussinized Conditional Expectations (AGCE)

Let us go back to our original problem, which strives to maximize the correlation between U and V, subject to marginal normality constraints (6). Here we follow Breiman and Friedman (1985), and suggest an alternating optimization procedure.

Let us fix $\psi(Y)$ and optimize (6) with respect to $\phi(X)$. As before, we can write the correlation objective as $E(\phi(X)\psi(Y)) = E_x\left(\phi(X)E(\psi(Y)|X)\right)$. Since $E(\phi(X)^2)$ is constrained to be equal to 1 while $E\left(E(\psi(Y)|X)^2\right)$ is fixed, maximizing $E_x\left(\phi(X)E(\psi(Y)|X)\right)$ is equivalent to minimizing $E_x\left(\phi(X) - E(\psi(Y)|X)\right)^2$. For simplicity, denote $\bar{X} \equiv E(\psi(Y)|X)$. Then, our optimization problem can be reformulated as

$$\begin{aligned} \min_{\phi} \quad & E\left(\phi(\bar{X}) - \bar{X}\right)^2 \\ \text{subject to} \quad & \bar{X} \sim F_{\bar{X}} \\ & \phi(\bar{X}) \sim N(0,1) \end{aligned} \tag{8}$$

where $F_{\bar{X}}$ is the (fixed) CDF of $\bar{X} \equiv E(\psi(Y)|X)$. Notice that ϕ is necessarily a function of $\bar{X}$ alone (as opposed to X), for simple optimization considerations. Assuming that $\bar{X}$ and $U = \phi(\bar{X})$ are two separable metric spaces such that any probability measure on $\bar{X}$ (or U) is a Radon measure (i.e. they are Radon spaces), then (8) is simply an optimal transportation problem (Monge, 1781) with a strictly convex cost function (mean square error). We refer to $\phi^*(\bar{X})$ that minimizes (8) as the optimal map.

The optimal transportation problem was presented by Monge (1781) and has generated an important branch of mathematics. The problem originally studied by Monge was the following: assume we are given a pile of sand (in $\mathbb{R}^3$) and a hole that we have to completely fill up with that sand. Clearly the pile and the hole must have the same volume and different ways of moving the sand will give different costs of the operation. Monge wanted to minimize the cost of this operation. Formally, the optimal transportation problem is defined as

$$\inf\left\{\int_{\bar{X}} c(\bar{X}, \phi(\bar{X}))d\mu(\bar{X}) \middle| \phi_*(\mu) = \nu\right\}$$

where μ and ν are the probability measures of $\bar{X}$ and U respectively, $c(\cdot,\cdot)$ is some cost function and $\phi_*(\mu)$ denotes the push forward of μ by the map ϕ. Clearly, (8) is a special case of the optimal transportation problem where the $\mu = F_{\bar{X}}$, ν is a standard normal distribution and the cost function is the euclidean distance between the two.

Assume that $\bar{X} \in \mathbb{R}$ has finite p^{th} moments for $1 \leq p < \infty$ and a strictly continuous CDF, $F_{\bar{X}}$ (that is $\bar{X}$ is a strictly continuous random variable). Then, Rachev and Rüschendorf (1998) show that the optimal map (which minimizes (8)) is exactly $\phi^*(\bar{X}) = \Phi_N^{-1} \circ F_{\bar{X}}(\bar{X})$ where Φ_N^{-1} is the inverse CDF of a standard normal distribution. As shown by Rachev and

Rüschendorf (1998), the optimal map is unique and achieves

$$E\left(\left(\phi^*(\bar{X})-\bar{X}\right)^2\right)=\int_0^1\left(F_{\bar{X}}(s)-\Phi_N(s)\right)^2 ds. \tag{9}$$

Notice that the optimal map may be generalized to the multivariate case, as discussed in the next Section. The solution to the optimal transportation problem is in fact the "optimal projection" of our problem (8). Further, it allows us to quantify how much we lose from imposing the marginal normality constraint, compared with ACE's optimal projection.

Notice that the optimal map, $\phi^*(\bar{X})=\Phi_N^{-1}\circ F_{\bar{X}}(\bar{X})$, is simply marginal Gaussianization of $\bar{X}$: applying $\bar{X}$'s CDF to itself results in a uniformly distributed random variable, while Φ_N^{-1} shapes this uniform distribution into a standard normal. In other words, while the optimal projection of $\psi(Y)$ on X is its conditional expectation, the optimal projection under a normality constraint is simply a Gaussianization of the conditional expectation. The uniqueness of the optimal map leads to the following necessary conditions for an optimal solution to (6),

$$\begin{aligned}\phi(X)&=\Phi_N^{-1}\circ F_{E(\psi(Y)|X)}(E(\psi(Y)|X))\\ \psi(Y)&=\Phi_N^{-1}\circ F_{E(\phi(X)|Y)}(E(\phi(X)|Y)).\end{aligned} \tag{10}$$

As in ACE, these necessary conditions imply an alternating projection algorithm, namely, the Alternating Gaussinized Conditional Expectation (AGCE). Here, we begin by randomly choosing a transformation that only satisfies the normality constraint $\psi(Y)\sim N(0,1)$. Then, we iterate by fixing one of the transformation while optimizing the other, according to (10). We terminate once $E(\phi(X)\psi(Y))$ fails to increase, which means that we converged to a set of transformations that satisfy the necessary conditions for optimal solution. Algorithm 1 summarizes our suggested approach. Notice that in every step of our procedure, we may either:

1. Increase our objective value, as a result of the optimal map for (8).
2. Maintain with the same objective value and with the same transformation that was found in of the previous iteration, as we converged to (10).

This means that our alternating method generates a monotonically increasing sequence of objective values. Moreover, as written in Section 4, this sequence is bounded from above by the optimal correlation given by ACE. Therefore, according to the monotone convergence theorem, our suggested method converges to a local optimum.

Unfortunately, as opposed to ACE, our projection operator is not linear and we cannot claim for global optimality. We see that for different random initializations we converge to (a limited number) of local optima. Yet, AGCE provides an effective tool for finding local maximizers of (4), which together with MCMC (Gilks, 2005) initializations (or any other random search mechanisms) is capable of finding the global optimum.

4.3 Off-the-shelf lower bound

Although the AGCE method provides a (locally) optimal solution to (4), we would still like to consider a simpler "off-the-shelf" mechanism that is easier to implement and gives a lower

Algorithm 1 Alternating Gaussinized Conditional Expectations (AGCE) for the univariate case

Require: F_{XY}, the joint distribution function of X and Y.
Require: $g:\mathbb{R}\to\mathbb{R}$, a random mapping.
Set $\psi(Y)=\Phi_N^{-1}\circ F_{g(Y)}(g(Y))$.
Set $\phi(X)=\Phi_N^{-1}\circ F_{E(\psi(Y)|X)}(E(\psi(Y)|X))$.
Set $\rho=E(\phi(X)\psi(Y))$.
Set $T=0$
while $T\neq 1$ **do**
  Set $\psi(Y)=\Phi_N^{-1}\circ F_{E(\phi(X)|Y)}(E(\phi(X)|Y))$.
  Set $\phi(X)=\Phi_N^{-1}\circ F_{E(\psi(Y)|X)}(E(\psi(Y)|X))$.
  if $E(\phi(X)\psi(Y))\not>\rho$ **then**
    $T=1$
  else
    $\rho=E(\phi(X)\psi(Y))$
  end if
end while
return $\phi(X),\psi(Y),\rho$

bound to the best we can hope for. Here, we tackle (4) in two phases. In the first phase we would like to maximize the correlation objective, $E(UV)$, subject to the relaxed second order statistics constraints (as defined in (7)). Then, we enforce the marginal normality constraints by simply applying *separate Gaussianization* to the outcome of the first phase. In other words, we first apply ACE to increase our objective as much as possible, and then separately Gaussianize the results to meet the normality constraints, hoping this process does not reduce our objective "too much". Notice that in this univariate case, separate Gaussianization is achieved according to Theorem 2:

Theorem 2 *Let X be any random variable $X\sim F_X(x)$ and $\theta\sim Unif[0,1]$ be statistically independent of it. In order to shape X to a normal distribution the following applies:*

1. *Assume X is a non-atomic distribution ($F_X(x)$ is strictly increasing) then*

$$\Phi_N^{-1}\circ F_X(X)\sim N(0,1)$$

2. *Assume X is discrete or a mixture probability distribution then*

$$\Phi_N^{-1}\circ\left(F_X(X)-\theta P_X(x)\right)\sim N(0,1)$$

The proof of this theorem can be located in Appendix 1 of (Shayevitz and Feder, 2011). Theorem 2 implies that if X is strictly continuous then we may achieve a normal distribution by applying $\Phi_N^{-1}\circ F_X(X)$ to it, as discussed in the previous section. Otherwise, we shall handle its CDF's singularity points by randomly scattering them in a uniform manner, followed by applying Φ_N^{-1} to the random variable we achieved. Notice that this process do not allow any flexibility in the Gaussianization process. However, we show that in the

multivariate case (Section (5.3)) the equivalent process is quite flexible and allows us to control the correlation objective.

Further, notice that this lower bound is by no means a candidate for an optimal solution to (6), as it does not meet the necessary conditions described in (10). Yet, by finding both an upper and lower bounds (through ACE, and then separately Gaussianizing the result of ACE) we may immediately achieve the range in which the optimal solution necessarily resides. Assuming this range is not too large, one may settle for a sub-optimal solution without a need to apply AGCE at all.

4.4 Illustrative example

We now demonstrate our suggested methodology with a simple illustrative example. Let $X \sim N(0,1)$, $W \sim N(0,\epsilon^2)$ and $Z \sim N(\mu_z,1)$ be three normally distributed random variables, all independent of each other. Let P be a Bernoulli distributed random variable with a parameter $\frac{1}{2}$, independent of X, W and Z. Define Y as:

$$Y = \left\{ \begin{array}{ll} \text{X+W} & \text{P=0} \\ \text{Z} & \text{P=1} \end{array} \right\}.$$

Then, Y is a balanced Gaussian mixture with parameters

$$\theta_y = \left\{\mu_1 = 0, \sigma_1^2 = 1+\epsilon^2, \mu_2 = \mu_z, \sigma_2^2 = 1\right\}.$$

The joint probability density function of X and Y is also a balanced two-dimensional Gaussian mixture with parameters

$$\theta_{xy} = \left\{\mu_1 = \left[\begin{array}{c} 0 \\ 0 \end{array}\right], C_1 = \left[\begin{array}{cc} 1 & 1 \\ 1 & 1+\epsilon^2 \end{array}\right], \mu_2 = \left[\begin{array}{c} 0 \\ \mu_z \end{array}\right], C_2 = I\right\}.$$

Let us further assume that μ_z is large enough, and ϵ^2 is small enough, so that the overlap between the two Gaussian is negligible. For example, we set $\mu_z = 10$ and $\epsilon = 0.1$. The correlation between X and Y is easily shown to be $\rho_{xy} = \frac{1/2}{\sqrt{1+1/2\epsilon^2+1/4\mu_z}} = 0.098$. The mutual information between X and Y is defined as

$$I(X;Y) = h(X) + h(Y) - h(X;Y).$$

Since we assume that the Gaussians in the mixture practically do not overlap, we have that

$$h(Y) = -\int f_Y(y)\log f_Y(y)dy \approx \frac{1}{4}\log\left(2\pi e(1+\epsilon^2)\right) + \frac{1}{4}\log\left(2\pi e\right) + 1. \tag{11}$$

In the same manner,

$$h(X,Y) = -\int f_{X,Y}(x,y)\log f_{X,Y}(x,y)dxdy \approx \tag{12}$$
$$\frac{1}{4}\log\left((2\pi e)^2|C_1|\right) + \frac{1}{4}\log\left((2\pi e)^2|C_2|\right) + 1.$$

Plugging $\mu_z = 10$ and $\epsilon = 0.1$ we have that

$$I(X;Y) = h(X) + h(Y) - h(X;Y) \approx 1.66\text{bits}. \tag{13}$$

The scatter plot on the left of Figure 1 illustrates 10,000 independent draws of X and Y, where the blue circles corresponds to the "correlated samples" ($P = 0$) while the blue crosses are the "noise" ($P = 1$).

Before we proceed to apply our suggested methods, let us first examine two benchmark options for separate Gaussianization. As an immediate option, we may always apply separate Gaussianization, directly to X and Y, denoted as U_a and V_a respectively. This corresponds to Cardoso (2003) information geometry bound. Since X is already normally distributed we may set $U_a = X$ and only apply Gaussinization to Y. Let $V_a = \psi(Y)$ be the Gaussianization of Y. This means that

$$V_a = \Phi_N^{-1}\left(F_Y\left(Y\right)\right) = \Phi_N^{-1}\left(\Phi_{GM(\theta_y)}(Y)\right)$$

where $\Phi_{GM(\theta_y)}$ is the cumulative distribution function a Gaussian Mixture with the parameters θ_y described above. Therefore,

$$\rho_{u_a,v_a} = E(XV) = \frac{1}{2}E\left(X\Phi_N^{-1}\left(\Phi_{GM(\theta_y)}(X+W)\right)\right).$$

Although it is not possible to obtain a closed form solution to this expectation, it may be numerically evaluated quite easily, as X and W are independent. Assuming $\mu_z = 10$ and $\epsilon = 0.1$ we get that $\rho_{u_a,v_a} \approx 0.288$ and our lower bound on the mutual information, as appears in (3), is $I_g \equiv -\frac{1}{2}\log\left(1-\rho_{u_a,v_a}^2\right) \approx 0.0628$bits. The middle scatter plot of Figure 1 presents this separate marginal Gaussianization of the previously drawn 10,000 samples of X and Y. Notice that the marginal Gaussianization is a monotonic transformation, so that the Y samples are not being shuffled and maintain the separation between the two parts of the mixture. While the red circles are now "half Gaussian", the blue crosses are shaped in a curvy manner, so that their marginal distribution (projected on the y axis) is also a "half Gaussian", leading to a normal marginal distribution of Y. We notice that while the mutual information between X and Y is 1.66 bits, the lower bound attained by this naive Gaussianization approach is close to zero. This is obviously an unsatisfactory result.

A second benchmark alternative for separate Gaussianization is to take advantage of the Gaussian mixture properties. Since we assume that the two Gaussians of Y are practically separable, we may distinguish between observations from the two Gaussians. Therefore, we can simply reduce μ_z from the Z samples (the red circles), and normalize the observations of $X + W$. This way the transformed Y becomes a Gaussian mixture of two co-centered standard Gaussians, and no further Gaussianization is necessary. For $\mu_z = 10$ and $\epsilon = 0.1$, this leads to a correlation of

$$\rho_{u_b,v_b} = \frac{1}{2}E\left(\frac{1}{\sqrt{1+\epsilon^2}}(X+W)X\right) = \frac{1}{2}\frac{1}{\sqrt{1+\epsilon^2}} = 0.497 \tag{14}$$

and a corresponding mutual information lower bound of $I_g = 0.204$ bits. However, notice that the suggested transformation is not invertible and may cause a reduction in mutual information. Specifically, we now have that the joint distribution of $U_b = X$ and V_b follows a Gaussian mixture model with parameters:

$$\theta_{u_b,v_b} = \left\{\mu_1 = \left[\begin{array}{c} 0 \\ 0 \end{array}\right], C_1 = \left[\begin{array}{cc} 1 & \frac{1}{\sqrt{1+\epsilon^2}} \\ \frac{1}{\sqrt{1+\epsilon^2}} & 1 \end{array}\right], \mu_2 = \left[\begin{array}{c} 0 \\ 0 \end{array}\right], C_2 = I\right\}.$$

Therefore,

$$h(U_b, V_b) = -\int f_{U_b,V_b}(u,v)\log\left(f_{U_b,V_b}(u,v)\right)dudv = \tag{15}$$
$$-\int \phi_{GN(\theta_{u_b,v_b})}(u,v)\log\phi_{GN(\theta_{u_b,v_b})}(u,v)dudv \approx 3.1384\text{bits}$$

where $\phi_{GN(\theta_{u_b,v_b})}(u,v)$ is the probability density function of a Gaussian mixture with the parameters θ_{u_b,v_b} described above, and the last approximation step is due to numerical integration. This leads to $I(U_b;V_b) = 0.95$ bits.

To conclude, although the mutual information is reduced from 1.66 bits to 0.95 bits, the suggested bound increased quite dramatically, from 0.0628 bits to 0.204 bits. The right plot of Figure 1 demonstrates this customized separate Gaussianization (as it only applies for this specific setup) to the previously sampled X and Y. Again, we emphasis that this solution is not applicable in general, and is only feasible due to the specific nature of this Gaussian mixture model.

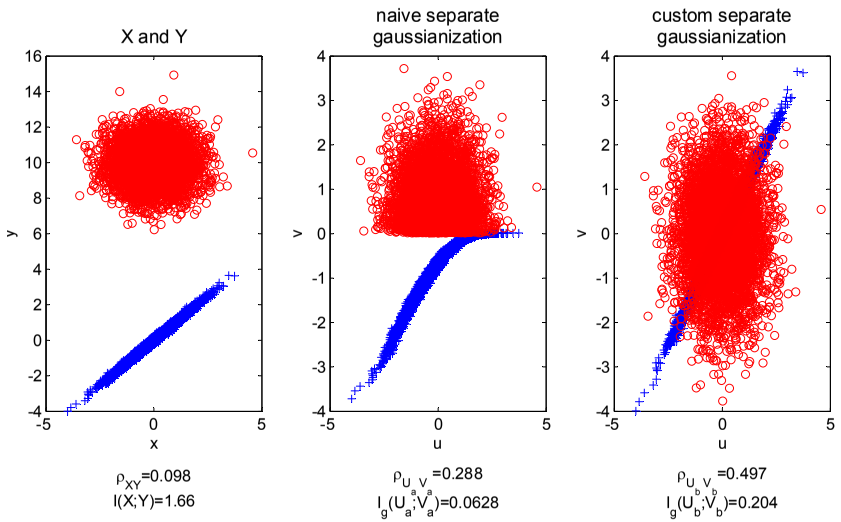


Figure 1: Univariate Gaussianization: Left: scatter of X and Y. Middle: naive separate Gaussianization to X and Y. Right: separate Gaussianization which considers the Gaussian Mixture model of X and Y, as described in the text.

Let us now turn to our suggested methods, as described in detail in the previous sections. We begin by applying the ACE procedure (Section 4.1), to attain an upper bound on our problem (6). Not surprisingly, ACE converges to a solution in which the samples of Y that are independent of X (the ones that come from Z) are set to zero,

while the rest are normalized to achieve an unit variance. Therefore, the resulting correlation is $\rho_{ub} = \frac{1/2}{\sqrt{1/2(1+\epsilon^2)}} = 0.703$. This results further implies that we can never find a Gaussianization procedure that will capture all the information between X and Y, as $I(X;Y) > -\log\left(1-\rho_{ub}^2\right) = 0.4917$ bits, according to Lemma 1. The left scatter plot of Figure 2 demonstrates the outcome of the ACE procedure, applied to the drawn $10,000$ samples of X and Y.

Next, we apply our suggested AGCE routine, described in Section 4.2. As discussed above, the AGCE only converges to a local optimum. Therefore, we initialize it with several random transformation (including the ACE solution that we just found). We notice that the number of convergence points is very limited and results in almost similar maxima. The middle scatter plot of Figure 4.2 shows the best result we achieve, leading to a correlation coefficient of 0.66 and a lower bound on a corresponding Gaussian lower bound (3) of 0.411 bits. This result demonstrates the power of our suggested approach, as it significantly improves the benchmarks, even compared with the U_b, V_b that considers the separable Gaussian mixture nature of our samples.

Finally, we evaluate a lower bound for (6), as described in Section 4.3. Here, we simply apply separate Gaussianization to the outcome of the ACE procedure. This results in $\rho_{lb} = 0.646$ and a corresponding $I_g = 0.389$. The right scatter plot of Figure 2 shows the Gaussianized samples the we achieve. We notice that this lower bound is not significantly lower than AGCE, suggesting that in some cases we may settle for this less involved method.

To conclude, our suggested solution surpasses the benchmarks quite easily, as we increase the lower bound from 0.204 bits using the custom Gaussianization procedure to 0.411 bits using our suggested solution. We notice that all of the discussed procedures result in joint distributions that are quite far from normal. This is not surprising, since X and Y were highly "non-normal" to begin with. Specifically, all of the suggested procedures lose information, compared with the original $I(X;Y) = 1.66$. However, our suggested solution minimizes this loss, and may be considered "more jointly normal" than others, in this regards.

5. The multivariate case

Let us now consider the multivariate case where both $\underline{X} \in \mathbb{R}^{d_x}$ and $\underline{Y} \in \mathbb{R}^{d_y}$ are random vectors with a joint CDF $F_{\underline{X},\underline{Y}}$. One of the fundamental differences from the univariate case is that Gaussianizing each of these vectors (even separately) is not a simple task. In other words, finding a transformation $\phi : \mathbb{R}^{d_x} \to \mathbb{R}^{d_u}$ such that $\underline{U} = \phi(\underline{X})$ is normally distributed may be theoretically straight-forward but practically involved.

For the simplicity of the presentation, assume that $\underline{X} = [X_1, X_2]^T$ is a two dimensional, strictly continuous, random vector. Then, Gaussianization may be achieved in two steps: first, apply marginal Gaussianization to X_1, so that $U_1 = \Phi_N^{-1} \circ F_{X_1}(X_1)$. Then, apply marginal Gaussianization on X_2, conditioned on each possible realization of the previous component, $U_2|u_1 = \Phi_N^{-1} \circ F_{X_2|U_1}(X_2|U_1 = u_1)$. This results in a jointly normally distributed vector $\underline{U} = [U_1, U_2]^T$. While this procedure is theoretically simple, it is quite problematic to apply in practice, as it requires Gaussianizing each and every conditional CDF. This is obviously impossible, given a finite number of samples. Yet, it gives us a constructive

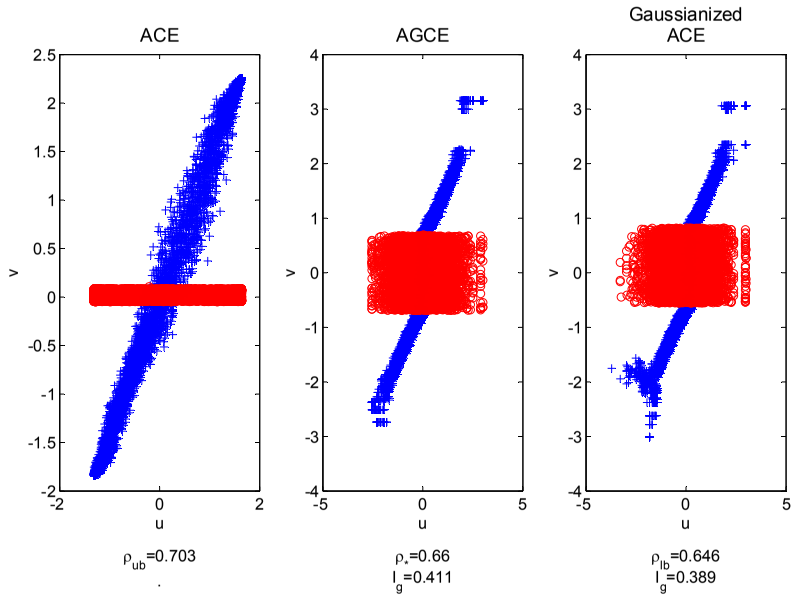


Figure 2: Our Suggested Univariate Gaussianization Schemes: Left: upper bound by ACE. Middle: (local) optimal solution by AGCE. Right: lower bound by separate Guassianization to ACE.

method, assuming that all the CDF's are known. In the following sections we shall present several alternatives for Gaussianization in a finite sample size setup.

5.1 Upper bound by ACE

As in the univariate case, we begin our analysis by relaxing the normality constraints with softer second order statistics constraints. This leads to an immediate multivariate generalization of the ACE procedure:

We begin by extracting the first canonical pair, which satisfies $U_1 = c \cdot E(V_1|\underline{X})$ and $V_1 = c \cdot E(U_1|\underline{Y})$. As in the univariate case, c is a normalization coefficient (the square root of the variance of the conditional expectation), and the optimization is done by alternating projections. Then, we shall extract the second pair of canonical components, subject to an orthogonality constraint with the first pair. It is easy to show that if V_2 is orthogonal to V_1, then $U_2 = c \cdot E(V_2|\underline{X})$ is orthogonal to U_1, and obviously maximizes the correlation with V_2. Therefore, we may extract the second canonical pair by first randomly assigning a zero-mean and unit variance V_2 that is also orthogonal to V_1 (by the Gram-Schmidt procedure, for example), followed by alternating conditional expectations with respect to V_2 and U_2, in the same manner as we did with the first pair. We continue this way for the rest of the

canonical pairs. As in the univariate case, convergence to a global maximum is guaranteed from the same Hilbert space arguments. As before, the multivariate ACE sets an upper bound to (5) as it maximizes a relaxed version of this problem.

Lemma 3 *Let $\underline{U}_*, \underline{V}_*$ be the outcome of multivariate ACE procedure (the canonical vectors). Assuming that $I(\underline{X};\underline{Y}) > \log\left|C_{[\underline{U}_*,\underline{V}_*]}\right|$, then there are no transformations such that $\underline{U} = \phi(\underline{X})$ and $\underline{V} = \psi(\underline{Y})$ follow a jointly normal distribution and preserve all of the mutual information, $I(X;Y)$.*

The proof of Lemma 3 follows exactly from the proof of Lemma 1. Here again, the multivariate ACE objective, $\log\left|C_{[\underline{U}_*,\underline{V}_*]}\right|$, cannot be further increased by artificially inflating the dimension of the problem. Therefore, Lemma 3 holds for any $\phi : \mathbb{R}^{d_x} \to \mathbb{R}^{d_u}$ and $\psi : \mathbb{R}^{d_y} \to \mathbb{R}^{d_v}$, such that $d_u, d_v \geq 0$.

5.2 multivariate AGCE

As with the multivariate ACE, we propose a generalized multivariate procedure for AGCE. We begin by extracting the first pair, in the same manner as we did in the univariate case. That is, we find a pair U_1 and V_1 that satisfies

$$\begin{aligned} U_1 &= \Phi_N^{-1} \circ F_{E(U_1|\underline{X})}(E(U_1|\underline{X})) \\ V_1 &= \Phi_N^{-1} \circ F_{E(V_1|\underline{Y})}(E(V_1|\underline{Y})) \end{aligned} \tag{16}$$

by applying the alternating optimization scheme. As we proceed to the second pair, we require that U_2 is both orthogonal and jointly normally distributed with U_1 (same goes for V_2 with respect to V_1). This means that the second pair needs not only to be orthogonal, but also statistically independent with the first pair. In other words, assuming V_2 is fixed, our basic projection step is

$$\begin{aligned} \max_{\phi_2} \quad & E\left(\phi_2(\underline{X})V_2\right) \\ \text{subject to} \quad & \phi_2(\underline{X}) \sim N(0,1) \\ & \phi_2(\underline{X}) \perp\!\!\!\perp \phi_1(\underline{X}). \end{aligned} \tag{17}$$

Let us denote a subspace $\tilde{\underline{X}} \subset \underline{X}$ that is statistically independent of $U_1 = \phi_1(\underline{X})$. Then, the problem of maximizing $E\left(\phi_2(\tilde{\underline{X}})V_2\right)$ subject to $\phi_2(\tilde{\underline{X}}) \sim N(0,1)$ is again solved by the optimal map, $\phi_2(\tilde{\underline{X}}) = \Phi_N^{-1} \circ F_{E(V_2|\tilde{\underline{X}})}(E(V_2|\tilde{\underline{X}}))$. Therefore, the remaining task is to find the "best" subspace $\tilde{\underline{X}} \subset \underline{X}$, so that $E\left(\phi_2(\tilde{\underline{X}})V_2\right)$ is maximal, when plugging the optimal map.

Proposition 4 *Let $U_1 = u_1$ be the value (realization) of U_1. Let $\tilde{\underline{X}} = g(\underline{X}, u_1)$ be a subspace of $\underline{X}$, independent of U_1. If $g(\underline{X}, u_1)$ is an invertible function with respect to $\underline{X}$ given u_1, then $\tilde{\underline{X}}$ is an optimal subspace for maximizing $E\left(\phi_2(\tilde{\underline{X}})V_2\right)$ subject to $\phi_2(\tilde{\underline{X}}) \sim N(0,1)$.*

Proof Assume there exists a different subspace $\underline{\tilde{X}}' = g'(\underline{X}, u_1)$ so that

$$\max_{\phi_2'} E\left(\phi_2'(\underline{\tilde{X}}')V_2\right) > \max_{\phi_2} E\left(\phi_2(\underline{\tilde{X}})V_2\right)$$

subject to the normality constraint. Since g is invertible we have that $\underline{X} = g^{-1}(\underline{\tilde{X}}, u_1)$. Therefore, $\underline{\tilde{X}}' = g'\left(g^{-1}(\underline{\tilde{X}}, u_1)\right) \equiv f(\underline{\tilde{X}}, u_1)$. Plugging this to the inequality above leads to

$$\max_{\phi_2'} E\left(\phi_2'(f(\underline{\tilde{X}}, u_1))V_2\right) > \max_{\phi_2} E\left(\phi_2(\underline{\tilde{X}})V_2\right)$$

which obviously contradicts the optimality of maximization over ϕ_2. ■

Therefore, we are left with finding $\underline{\tilde{X}} = g(\underline{X}, u_1)$ that is a subspace of $\underline{X}$, independent of U_1 and invertible with respect to $\underline{X}$ given u_1. For simplicity of the presentation, let us first assume that X is univariate. Then, the function $g(X, u_1) = F_{X|U_1}(X|U_1 = u1)$ is independent of U_1 (as it holds the same (uniform) distribution, regardless to the value of U_1), and invertible given u_1 (assuming that the conditional CDF's $F_{X|U_1}(X|U_1 = u1)$ are continuous for every u_1). Going back to the multivariate $\underline{X} \in \mathbb{R}^{d_x}$, we may follow the same rationale by choosing a single d_x-dimensional distribution that all the conditional CDF's, $F_{\underline{X}|U_1}$ will be shaped to. For simplicity we choose a d_x-dimensional uniform distribution, denoted by its CDF as F_{unif}. Then, $g_*\left(F_{\underline{X}|U_1}, u_1\right) = F_{unif}$, where $g_*(P, x) = Q$ refers to a mapping that pushes forward the distribution P into Q, given x. Specifically, if $p(w)$ and $q(w)$ are the corresponding density functions of the (absolutely continuous) CDF's P and Q respectively, then we know from basic probability theory that the push forward transformation S satisfies

$$p(w) = q\left(S(w)\right)\left|J_S\left(S(w)\right)\right|$$

where J_S is the Jacobian operator of the map S.

To conclude, in order to construct $\underline{\tilde{X}}$ that is independent of U_1 and invertible given u_1, we need to push forward all the conditional CDF's $F_{\underline{X}|U_1}(\underline{X}|U_1 = u_1)$ into a predefined distribution (say, uniform). Then, the optimal map $\phi_2(\underline{\tilde{X}})$ that maximizes $E\left(\phi_2(\underline{\tilde{X}})V_2\right)$ subject to $\phi_2(\underline{\tilde{X}}) \sim N(0,1)$ is given by $\phi_2(\underline{\tilde{X}}) = \Phi_N^{-1} \circ F_{E(V_2|\underline{\tilde{X}})}(E(V_2|\underline{\tilde{X}}))$. In the same manner, we may find $\underline{\tilde{Y}}$ that is independent of V_1 and invertible given v_1, and carry on with the alternating projections. This process continues for all the Gaussinized canonical components and converges to a local optimum, from the same considerations described in the univariate case.

It is important to notice that while this procedure may be considered practically infeasible (as it requires estimating the conditional CDF's), it is equivalently impractical as the multivariate Gaussianization considered in the beginning of this section. Yet, it gives us a local optimum for our problem, assuming that the joint probability distribution is known.

5.3 Off-the-shelf lower bound in the multivariate case

In the same manner as with the univariate case, we may apply a simple off-the-shelf lower bound to (4) by first maximizing the objective as much as we can (using multivariate ACE) followed by Gaussianizing the outcome vectors, hoping we do not reduce the objective

"too much". However, as mentioned in the beginning of Section 5, applying multivariate Gaussianization may be practically infeasible. Therefore, we begin this section by reviewing practical multivariate Gaussianization methodologies. Then, we use these ideas to suggest a practical lower bound, which unlike the univariate case, is not oblivious to our objective.

5.3.1 Practical multivariate Gaussianization

The Gaussianization procedure strives to find a transformation $\underline{Z} = \mathcal{G}(\underline{X})$ so that $\underline{Z} \sim N(0, I)$. A reasonable a cost function for describing "how Gaussian" $\underline{Z}$ really is, may be the Kullback Leibler divergence (KLD) between $\underline{Z}$'s PDF, $f_{\underline{Z}}(\underline{z})$, and a standard normal distribution,

$$J(\underline{Z}) = D_{KL}\left(f_{\underline{Z}}(\underline{z})||f_N(\underline{Z})\right) = \int_{\underline{Z}} f_{\underline{Z}}(\underline{z}) \log\left(\frac{f_{\underline{Z}}(\underline{z})}{f_N(\underline{Z})}\right) dz$$

where $f_N(\underline{Z})$ is the PDF of a standard normal distribution. As shown by Chen and Gopinath (2001), $J(\underline{Z})$ may be decomposed into

$$J(\underline{Z}) = D_{KL}\left(f_{\underline{Z}}(\underline{z})||\prod_{i=1}^{d_z} f_{Z_i}(z_i)\right) + \sum_{i=1}^{d_z} D_{KL}\left(f_{Z_i}(z_i)||f_N(z_i)\right) \tag{18}$$

where the first term quantifies how independent are the components of $\underline{Z}$, and the second term indicates how normally distributed they are. This decomposition led Chen and Gopinath (2001) to an iterative algorithm. In each iteration, their suggested approach applies Independent Component Analysis (Hyvärinen et al., 2004), to minimize the first term, followed by marginal Gaussianization of each component (as we describe for the univariate case), to minimize the second term. Chen and Gopinath show that minimizing one term does not effect the other, which leads to a monotonically decreasing procedure that converges once $\underline{Z}$ is normally distributed.

Notice that the Independent Component Analysis (ICA) is a linear operator. Therefore, if $\underline{Z}$ can be linearly decomposed into independent components, then Chen and Gopinath's Gaussianizion process converges in a single step. Moreover, notice that this Gaussianization process does not require estimating the multivariate distribution. However, it does require estimating the marginals, f_{Z_i} which is considered a much easier task, in general.

A similar yet different multivariate Gaussianization approach was suggested by Laparra et al. (2011). Here, the authors propose to replace the computationally costly ICA with a simple random rotation matrix. This way, they abandon the effort of minimizing the first term of (18), and only shuffle the components so that consequent marginal Gaussianization would further decrease $J(\underline{Z})$. Although this approach takes more iterations to converge to a normal distribution (as in each iteration, only the second term of (18) is being minimized), it holds several favorable properties. First, the overall run-time is dramatically shorter, since applying random rotations is much faster then applying linear ICA. Second, it implies a degree of freedom in choosing the rotation matrix, as the suggested random matrix is just one example of linear shuffling of the components.

5.3.2 Bi-terminal multivariate Gaussianization

Going back to our problem, we would like to Gaussinize $\underline{U}_*$ and $\underline{V}_*$, the outcomes of the multivariate ACE procedure described above. Ideally, we would like to do so while refraining

(as much as we can) from reducing our objective,

$$\log\left(\frac{\left|C_{[\underline{U}_*,\underline{V}_*]}\right|}{|C_{\underline{U}_*}||C_{\underline{V}_*}|}\right). \tag{19}$$

Following the Gaussianization procedures described in the previous section, we suggest an iterative process, where in each iteration we apply a rotation matrix to both vectors, followed by marginal Gaussianization to each of the components of the two vectors. It is easy to show that (19) is invariant to full rank linear transformations. However, it may be effected by the (non-linear) marginal Gaussianization of the components (as described in Theorem 2). Therefore, we would like to find rotation matrices that minimize the effect of the consequent marginal Gaussianization step. This problem is far from trivial. In fact, due to the complicated nature of the marginal Gaussianization procedure, it is quite impossible to minimize the effect of the marginal Gaussianization a-priori, without actually applying it and see how it effects (19). Therefore, we suggest a stochastic search mechanism, which allows us to construct a "reasonable" rotation matrix.

Our suggested mechanism works as follow: At each iteration we begin by drawing two random rotation matrices R_1 and R_2 for the two vectors we are to Gaussianize, just like Laparra et al. (2011). We apply marginal Gaussianization to all the components and evaluate our objective (19). Then, we randomly choose two dimensions and an angle, θ, and construct a corresponding rotation matrix $\tilde{R}$ that rotates the space spanned by the two dimensions in θ degrees. We apply $\tilde{R}\cdot R_1$ to our vector, followed by marginal Gaussianization, and again evaluate (19). If the objective increases we assign $R_1 = \tilde{R}\cdot R_1$. We repeat this process a configurable number of times, for the two vectors we are to Gaussianize.

Notice that our suggested procedure applies a stochastic hill climbing (SHC) search in each step: it randomly searches for the best rotation matrix by gradually composing "small" rotation steps (of two dimensions and an angle), as the complete search space is practically infinite. This procedure guarantees to converge to two multivariate normal vectors, as shown by Laparra et al. (2011), under the reasonable assumption that R_1 and R_2 do not repeatedly converge to identity matrices. Our suggested approach is described in detail in Algorithm 2.

As we see in our experiments, the Bi-terminal Gaussianization process is superior to naively applying a Gaussianization procedure to each of the vectors separately (as suggested by Chen and Gopinath (2001) or Laparra et al. (2011)), in all the cases we examine.

5.4 Illustrative examples

We now examine our suggests multivariate approach in different setups. As in the univariate case, we draw samples from a given model and bound from below the mutual information $I(\underline{X},\underline{Y})$ according to (3). First, we apply the multivariate ACE procedure (Section 5.1) to achieve an upper bound to our objective. Then, we apply separate Gaussianization to ACE's outcome, to attain an immediate lower bound (Section 5.3.1). Further, we tighten this lower bound by replacing the separate Gaussianization with bi-terminal Gaussianization to ACE's outcome (Section 5.3.2). Since our multivariate AGCE procedure (Section 5.2) is practically infeasible, we refrain from using it. This would be further justified later in our results, as we see that the gap between the lower and upper bounds is relatively small. In

Algorithm 2 Bi-terminal multivariate Gaussianization

Require: $\underline{X} \in \mathbb{R}^{d_x}$, $\underline{Y} \in \mathbb{R}^{d_y}$.
Require: Th, a Gaussianization convergence threshold and N, the SHC parameter.
Set $\underline{U} = \underline{X}$ and $\underline{V} = \underline{Y}$.
Set $J_U = J(\underline{U})$ and $J_V = J(\underline{V})$ according to (18).
while $J_U \geq Th$ OR $J_V \geq Th$ **do**
  Draw a rotation martix R_1 of dimensions $d_x \times d_x$ and set $\underline{U}^* = R_1\underline{U}$.
  Draw a rotation matrix R_2 of dimensions $d_y \times d_y$ and set $\underline{V}^* = R_2\underline{V}$.
  Apply marginal Gaussianization to $\underline{U}^*$ and $\underline{V}^*$.
  Set $\rho^* = \log\left(\frac{\left|C_{[\underline{U}^*,\underline{V}^*]}\right|}{|C_{\underline{U}^*}||C_{\underline{V}^*}|}\right)$.
  for all $n = 1$ to N **do**
    Draw an angle θ.
    Draw (without replacement) two dimensions d_a, d_b from the set $\{1,\ldots,d_x\}$.
    Construct a rotation matrix $\tilde{R}$ from θ, d_a, d_b and set $\tilde{\underline{U}} = \tilde{R}R_1\underline{U}$.
    Apply marginal Gaussianization to $\tilde{\underline{U}}$.
    Set $\tilde{\rho} = \log\left(\frac{\left|C_{[\tilde{\underline{U}},\underline{V}^*]}\right|}{|C_{\tilde{\underline{U}}}||C_{\underline{V}^*}|}\right)$.
    if $\tilde{\rho} > \rho^*$ **then**
      Set $R_1 = \tilde{R}R_1$, $\underline{U}^* = \tilde{\underline{U}}$ and $\rho^* = \tilde{\rho}$.
    end if
    Draw an angle θ.
    Draw (without replacement) two dimensions d_a, d_b from the set $\{1,\ldots,d_y\}$.
    Construct a rotation matrix $\tilde{R}$ from θ, d_a, d_b and set $\tilde{\underline{V}} = \tilde{R}R_2\underline{U}$.
    Apply marginal Gaussianization to $\tilde{\underline{V}}$.
    Set $\tilde{\rho} = \log\left(\frac{\left|C_{[\underline{U}^*,\tilde{\underline{V}}]}\right|}{|C_{\underline{U}^*}||C_{\tilde{\underline{V}}}|}\right)$.
    if $\tilde{\rho} > \rho$ **then**
      Set $R_2 = \tilde{R}R_2$, $\underline{V}^* = \tilde{\underline{V}}$ and $\rho^* = \tilde{\rho}$.
    end if
  end for
  Set $\underline{U} = \underline{U}^*$ and $\underline{V} = \underline{V}^*$.
  Set $J_U = J(\underline{U})$ and $J_V = J(\underline{V})$ according to (18).
end while
return $\underline{U}, \underline{V}, \rho^*$.

all of our experiments, our benchmark would be a direct separate Gaussianization of $\underline{X}$ and $\underline{Y}$, as an immediate alternative.

We begin with a simple toy example. Let $\underline{X} \sim N(0, I)$ and $\underline{W} \sim N(0, I)$ be independent random vectors. Define $\underline{Y} = \underline{X} + \underline{W}$, so that $\underline{X}$ and $\underline{Y}$ are jointly normally distributed. Further, we "scramble" $\underline{X}$ and $\underline{Y}$ by applying invertible, yet non-monotonic, transformations to each of them separately. We ask that the transformations are invertible to guarantee that the (analytically derived) mutual information is preserved. We further require non-monotonic transformations since marginal Gaussianization is invariant to monotonic functions (see Proposition 5), which would make this experiment too easy. In this experiment, we multiply all the observations in the range $[-1, 1]$ by -1. This operation simply mirrors these observations with respect to the origin.

Proposition 5 *Let $\tilde{X} = g(X)$ be a monotonic transformation on $X \in \mathbb{R}$. Then Gaussinizing $\tilde{X}$ is equivalent to Gaussinizing X.*

Proof Let $\tilde{V} = \Phi_N^{-1}\left(F_{\tilde{X}}\left(\tilde{X}\right)\right)$ be the Gaussianization $\tilde{X}$ and $V = \Phi_N^{-1}\left(F_X\left(X\right)\right)$ is the Gaussianization of X. Assume that g is monotonically increasing. Then,

$$F_{\tilde{X}}(a) = P(\tilde{X} \leq a) = P(g(X) \leq a) = P(X \leq g^{-1}(a)).$$

Therefore, $F_{\tilde{X}}\left(\tilde{X}\right) = F_X\left(g^{-1}(\tilde{X})\right) = F_X(X)$ and $\tilde{V} = V$. An equivalent derivation holds for the monotonically decreasing case. ■

Before we proceed, it is important to briefly comment on the implications of the finite sample size in our multivariate experiments. The ACE procedure estimates conditional expectations at each of its iterations. This estimation task is known to be quite challenging in a finite sample size regime. Breiman and Friedman (1985) suggest a *k nearest neighbor* estimator which guarantees favorable consistency properties. Unfortunately, this solution suffers from the curse of dimensionality (Hastie et al., 2005). Therefore, as the dimension of our problem increases, we cannot turn to ACE and have to settle for suboptimal solutions. In our experiments, we use the kernel CCA (Lai and Fyfe, 2000) as an alternative to ACE when the dimension size is greater than $d = 5$. The kernel CCA (KCCA) is a non-linear generalization to the classical CCA which embeds the data in a high-dimensional Hilbert space and applies CCA in that space. It is known to significantly improve the flexibility of CCA while avoiding over-fitting of the data. Notice that other non-linear CCA extensions, such as *Deep CCA* (Andrew et al., 2013) or *nonparametric CCA* (Michaeli et al., 2016), may also apply as a finite sample size alternative to ACE.

We now demonstrate our suggested approach to the jointly Gaussian model discussed above. The left plot of Figure 3 demonstrates the results we achieve for different dimension sizes d. The black line on the top is $I(\underline{X}, \underline{Y})$, which can be analytically derived. The red curve with the squares at the bottom is separate Gaussianization of $\underline{X}$ and $\underline{Y}$, which results in a very poor lower bound to the mutual information due to the non-monotonic nature of the transformation that we apply. The green curve with the circles is ACE, while the dashed blue curve is separate Gaussianization of ACE. Finally, the blue line between them is bi-terminal Gaussianization of ACE. As we can see, ACE succeeds in recovering the jointly Gaussian

representation of $\underline{X}$ and $\underline{Y}$, which makes further Gaussianization redundant. Unfortunately, for $d > 5$ we can no longer apply ACE and turn to KCCA instead. We use a Gaussian kernel with varying parameters to achieve the reported results. Since the KCCA attains a suboptimal representation it is followed by Gaussianization, which further decreases our objective. Here, we notice the improved effect of the bi-terminal Gaussianization, compared with separate Gaussianization.

Next, we turn to a more challenging exponential model. In this model, each component of $\underline{X}$ and $\underline{W}$ is exponentially distributed with a unit parameter, while all the components are independent of each other. Again, we define $\underline{Y} = \underline{X} + \underline{W}$ so that $\underline{Y}$ is Gamma distributed. This allows us to analytically derive $I(\underline{X}, \underline{Y})$. As before, we apply an invertible non-monotonic transformation to each of the components of $\underline{X}$ and $\underline{Y}$. Notice that this time we mirror the observations in the range $[0, 2]$ with respect to 1. We then apply a linear rotation, so that the components are no longer independent. The plot in the middle of Figure 3 demonstrates the results we achieve. As before, we notice that separate Gaussianzation of $\underline{X}$ and $\underline{Y}$ preforms very poorly. On the other hand, ACE as well does not succeed in maintaining the MI. This means that no Gaussianzation procedure would allow jointly normal representation of $\underline{X}$ and $\underline{Y}$ without losing information (Lemma 3). Still, by applying bi-terminal Gaussianization to ACE's results we are able to capture more than half of the information in the worst case (for $d = 5$, where ACE still applies). As before, we witness a reduction of performance when turning from ACE to KCCA.

Finally, we go back to the multivariate extension of the Gaussian mixture model described in Section 4.4 and apply our suggested procedures. Again, we witness the same behavior described in the previous experiments. In addition, our results indicate that in this model, the Gaussian part of the MI is significantly smaller, compared with the exponential model. This further demonstrates the ability of our method to quantify how well an arbitrary distribution may be represented as jointly normal.

6. Gaussian lower bound for the Information Bottleneck Curve

We now extended our derivation to the Information Bottleneck (IB) curve. We show that by maximizing the Gaussian lower bound of the mutual information (3), we allow a maximization of a Gaussian lower bound to the entire IB curve. We prove this in two steps. First, we show that the IB curve of $\phi(\underline{X}), \psi(\underline{Y})$ bounds from below the IB curve of X and Y, for any choice of ϕ, ψ (specifically, $\phi(\underline{X}) \sim N$ and $\psi(\underline{Y}) \sim N$, in our case). This property is referred to as the *data processing lemma for the IB curve*. Then, we show that the IB curve of jointly normal random variables bounds from below the IB curve of separately normal random variable. Finally, by applying the GIB (Chechik et al., 2005) to the maximally correlated jointly normal random variables that satisfy (3), we attain the desired Gaussian lower bound for the IB of $\underline{X}$ and $\underline{Y}$.

Lemma 6 *(data processing lemma for the IB Curve): Let (20) be the equivalent maximization problem of the IB problem (1):*

$$\begin{aligned} \max_{T} \quad & I(T(\underline{X}); \underline{Y}) \\ \textit{subject to} \quad & I(T(\underline{X}); \underline{X}) \leq I_X. \end{aligned} \tag{20}$$

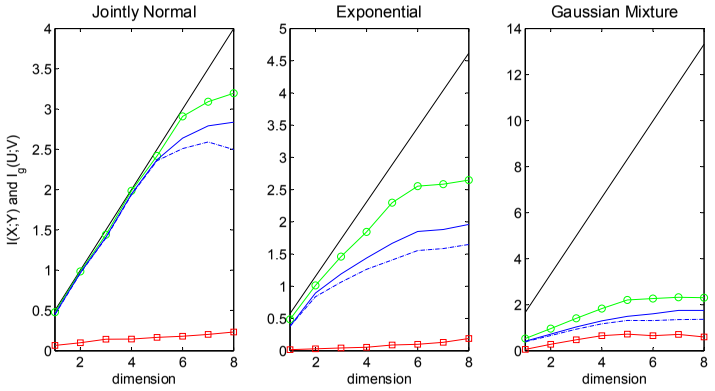


Figure 3: Multivariate Gaussianization experiments: The black line on the top of each plot is $I(\underline{X},\underline{Y})$. The red curve with the squares at the bottom is separate Gaussianization of $\underline{X}$ and $\underline{Y}$. The green curve with the circles is ACE, while the dashed blue curve is separate Gaussianization of ACE. The blue line in between is bi-terminal Gaussianization of ACE.

Denote its solution as $I_*^\beta(\underline{X};\underline{Y})$. *Then,* $I_*^\beta(\underline{X};\underline{Y}) \geq I_*^\beta(\phi(\underline{X});\psi(\underline{Y}))$ *for any* ϕ,ψ, *and with equality iff* $I(\underline{X};\underline{Y}) = I(\phi(\underline{X});\psi(\underline{Y}))$.

Proof We prove this lemma by showing that $I_*^\beta(\underline{X};\underline{Y}) \geq I_*^\beta(\underline{X};\psi(\underline{Y})) \geq I_*^\beta(\psi(\underline{X});\psi(\underline{Y}))$. We start with the first inequality. According to the data processing lemma, we have that $I(T(\underline{X});\underline{Y}) \geq I(T(\underline{X});\psi(\underline{Y}))$. Notice that for convenience, we emphasize that T is indeed a mapping of X alone. In addition, since our constraint (1) is independent of Y, we have that $I_*^\beta(\underline{X};\underline{Y}) \geq I_*^\beta(\underline{X};\psi(\underline{Y}))$, as desired. Second, denote the minimizer of (1) as $I_*(\underline{X};\underline{Y})$. Assume that there exists such ϕ that

$$I_*(\underline{X};\underline{Y}) > I_*(\phi(\underline{X});\underline{Y}). \tag{21}$$

This means that for $I(T(\underline{X});\underline{Y}) \geq I_Y$ and $I(T'(\phi(\underline{X}));\underline{Y}) \geq I_Y$ we have that $I(T(\underline{X});\underline{X}) > I(T'(\phi(\underline{X}));\phi(\underline{X}))$ where T and T' are the optimizers of (1) with respect to $(\underline{X},\underline{Y})$ and $(\phi(\underline{X}),\underline{Y})$, for a given I_Y, respectively. Let us set $\bar{T} \equiv T' \circ \phi$ and apply this transformation to $\underline{X}$. Then, we have that the constraint of (1) is met, as $I(\bar{T}(\underline{X});\underline{Y}) \equiv I(T'(\phi(\underline{X}));\underline{Y}) \geq I_Y$. In addition, we have that

$$I(\bar{T}(\underline{X});\underline{X}) \equiv I(T'(\phi(\underline{X}));\underline{X}) = I(T'(\phi(\underline{X}));\phi(\underline{X}))$$

where the second equality follows from T' being independent of $\underline{X}$, given $\phi(\underline{X})$. Therefore, $\bar{T} = T' \circ \phi$ is a better optimizer to (1) with respect to $\underline{X}$ and $\underline{Y}$, then T. This contradicts the optimality of T as a minimizer of (1), which means that the assumption in (21) is false. Therefore, $I_*(\underline{X};\underline{Y}) \leq I_*(\phi(\underline{X});\underline{Y})$ which means that $I_*^\beta(\underline{X};\underline{Y}) \geq I_*^\beta(\phi(\underline{X});\underline{Y})$ for any $\underline{Y}$ (specifically, $\phi(\underline{Y})$) and our proof is concluded. ■

Lemma 7 *Let* $\underline{U}$ *and* $\underline{V}$ *be separately Gaussian random vectors with a joint covariance matrix* $C_{[\underline{U},\underline{V}]}$ *(that is,* $\underline{U} \sim N$ *and* $\underline{V} \sim N$ *but* $[\underline{U},\underline{V}]^T$ *is not normally distributed). Let* $\underline{U}_{jg},\underline{V}_{jg}$ *be two jointly normally distributed random vectors with the same covariance matrix,* $C_{[\underline{U}_{jg},\underline{V}_{jg}]} = C_{[\underline{U},\underline{V}]}$. *Then, the IB curve of* $\underline{U}_{jg}$ *and* $\underline{V}_{jg}$ *bounds from below the IB curve of* $\underline{U}$ *and* $\underline{V}$.

Proof Let $\big(I(\underline{U}_{jg};\underline{T}), I(\underline{T};\underline{V}_{jg})\big)$ be a point of the IB curve of $\underline{U}_{jg}$ and $\underline{V}_{jg}$. Since $\underline{U}_{jg}$ and $\underline{V}_{jg}$ are jointly normally distributed, T is necessarily a linear transformation of $\underline{U}_{jg}$, with additive independent Gaussian noise (Chechik et al., 2005). Specifically, $T = A\underline{U}_{jg} + \underline{\zeta}$, where $\underline{\zeta} \sim N(0,I)$, independent of $\underline{U}_{jg}$ and $\underline{V}_{jg}$.

Further, let $\underline{T}' = A\underline{U} + \underline{\zeta}$ be the same transformation, applied of $\underline{U}$. Since $\underline{U}$ and $\underline{V}$ are not jointly normal, the point $(I(\underline{U};\underline{T}'), I(\underline{T}';\underline{V}))$ is below the IB curve of $\underline{U}$ and $\underline{V}$. First, notice that

$$I(\underline{U};\underline{T}') \equiv I(\underline{U};A\underline{U}+\underline{\zeta}) = I(\underline{U}_{jg};A\underline{U}_{jg}+\underline{\zeta}) \equiv I(\underline{U}_{jg},\underline{T})$$

where the second equality follows from $\underline{U}$ and $\underline{U}_{jg}$ having the same distribution. In addition, since $C_{[\underline{U}_{jg},\underline{V}_{jg}]} = C_{[\underline{U},\underline{V}]}$ we have that $C_{[A\underline{U}_{jg}+\underline{\zeta},\underline{V}_{jg}]} = C_{[A\underline{U}+\underline{\zeta},\underline{V}]}$. Therefore, $I(A\underline{U}+\underline{\zeta};\underline{V}) \geq I(A\underline{U}_{jg}+\underline{\zeta};\underline{V}_{jg})$, in the same manner as the in (3). This means that $I(\underline{T}';\underline{V}) \geq I(\underline{T};\underline{V}_{jg})$. To conclude, we showed that for the two pairs, $\big(I(\underline{U}_{jg};\underline{T}), I(\underline{T};\underline{V}_{jg})\big)$ and $(I(\underline{U};\underline{T}'), I(\underline{T}';\underline{V}))$, we have that $I(\underline{U};\underline{T}') = I(\underline{U}_{jg},\underline{T})$ while $I(\underline{T}';\underline{V}) \geq I(\underline{T};\underline{V}_{jg})$, as desired. ■

The two theorems above guarantee that the IB curve of $\underline{X}$ and $\underline{Y}$ is bounded from below by the IB curve of $\underline{U}_{jg}$ and $\underline{V}_{jg}$, where $C_{[\underline{U}_{jg},\underline{V}_{jg}]} = C_{[\underline{U},\underline{V}]}$, and $\underline{U} = \phi(\underline{X}) \sim N$, $\underline{V} = \psi(\underline{Y}) \sim N$. Therefore, in order to maximize this lower bound, one needs to maximize the correlation between $\underline{U}$ and $\underline{V}$, subject to a normality constraint, as discussed throughout this manuscript. Moreover, once we have found a pair of $(\underline{U}_{jg},\underline{V}_{jg})$ with a maximal correlation, we may directly apply the GIB to it, as shown by Chechik et al. (2005), to achieve the optimal Gaussian lower bound oIB curve for $\underline{X}$ and $\underline{Y}$.

6.1 Examples

We now demonstrate our suggested Gaussian lower bound for the IB curve in two different setups. Here, we would like to compare our bound with the "true" IB curve, and with an additional benchmark off-the-shelf lower bound. As discussed in Section 1, computing the exact IB curve (for a general joint distribution) is not a simple task. This task becomes even more complicated when dealing with continuous random variables. In fact, to the best of our knowledge, all currently known methods provide approximated curves, which do not claim to converge to the exact IB curve. Moreover, these methods fail to provide any guarantees on the extent of their divergence from the true IB curve. Therefore, in our experiments, we apply the commonly used reverse annealing technique (Slonim, 2002) in order to approximate the "true" IB curve. The reverse annealing algorithm is initiated by computing the mutual information between $\underline{X}$ and $\underline{Y}$, which corresponds to extreme point where $I_Y \to \infty$ on the IB curve. Then, I_Y is gradually decreased and the solution of the IB problem (1) with the previous value of I_Y serves as a starting point to the currently solved I_Y. This results in a greedy "no-regret" optimization method, which in general, fails to

converge to the exact IB curve. However, in some special cases (such as the GIB), it can be shown that the optimal solution for a given value of I_Y is, in fact, the optimal starting point for a smaller value of I_Y. In the general case, it is implicitly assumed to be a reasonable local optimization domain. Since the reverse annealing was originally designed for discrete random variables, we apply discretization (via Gaussian quadratures) to our probability distributions is all of our experiments.

We begin by revisiting the exponential model, described in Section 5.4. In this model, X and W are independent exponentially distributed random variables with a unit parameter. We define $Y = X + W$ so that Y is Gamma distributed. As in Section 5.4 we apply an invertible non-monotonic transformation to X and Y, to make this problem more challenging. Since approximating the IB curve is involved enough for continuous random variables, we limit our attention to the simplest univariate case.

The plot on the left of Figure 4 demonstrates the results we achieve. The black curve on top is the approximated IB curve, using the reverse annealing procedure. The red curve on the bottom is a benchmark lower bound, achieved by simply applying the GIB to X and Y, as if they were jointly Gaussian. The blue curve in the middle is our suggested Gaussian lower bound (Section 4.2). As we can see, our suggested bound surpasses the GIB quite remarkably. This is mainly due to the non-monotonic transformation we apply, which makes the joint distribution highly non-Gaussian. We further notice that our bound is quite tight for smaller I_Y's (closer to the origin) but increasingly diverges as I_Y increases. The reason is that more compressed representations are more "degenerate" and are easier to Gaussianize while maintaining reasonably high correlations.

Next, we revisit the more challenging Gaussian mixture model, described in Section 4.4. The right plot in Figure 4 demonstrates the results we achieve. As before, we notice that our suggested lower bound surpasses the naive benchmark, while demonstrating favorable performance closer to the origin. Comparing the two models, we notice that the Gaussian mixture is more difficult to bound from below using our suggested method. This result is not surprising, given the gap in our ability to bound from below the mutual information in these two models, as discussed in Section 5.4.

7. Discussion and conclusion

In this work we address the fundamental problem of normalizing non-Gaussian data, while trying to avoid loss of information. This allows us to solve complex problems by linear means, as we push information to the data's second moments. We show that our ability to do so is strongly governed by the non-linear canonical correlations of the data. In other words, if the non-linear canonical coefficients of the data fail to maintain its mutual information, then it is impossible to describe its high order dependencies just by second order statistics. This result is of high interest to a broad variety of applications, as solving non-linear problems by linear means is a common alternative in many scientific and engineering fields. Further, we provide a variety of methods to quantify the minimal amount of information that may be lost when normalizing the data. We show that in many cases, our suggested approach is able to preserve a significant portion of the information, even for highly non-Gaussian joint distributions. Our results improve upon Cardoso (2003) information geometry bound, as we show that a tighter bound may be obtained by the AGCE method.

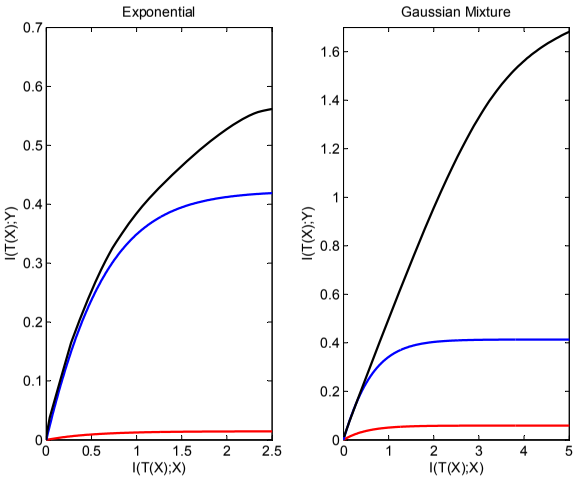


Figure 4: Bounding the Information Bottleneck curve for Exponential and Gaussian Mixture distributions: The black line is the approximated IB curve and the blue line is our suggested Gaussian lower bound. The red curve is achieved by applying the GIB directly to X, Y.

It is important to mention that while our suggested approach is theoretically found, it exhibits several practical limitation in a finite sample-size setup. This is a direct result of our use of the ACE algorithm, which suffers from the curse of dimensionality when applied to high-dimensional data. Therefore, we further examine different non-linear CCA methods, which are less vulnerable to this problem. However, these methods fail to converge to the optimal canonical coefficients.

Finally, we show that our results may be generalized to bound from below the entire information bottleneck curve. This allows a practical alternative for different approximation methods and restrictive solutions to the involved IB problem in the continuous case. Our experiments show that the suggested Gaussian lower bound provides a meaningful benchmark to the IB curve, even in highly non-Gaussian setups.

8. Acknowledgments

This research was supported by a Fellowship from the Israeli Center of Research Excellence in Algorithms to Amichai Painsky. The authors thank Nori Jacoby for early discussions on the subject.

References

Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through noisy computation. *IEEE Transactions on Pattern Analysis and Machine*

Intelligence, 2018.

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *International Conference on Machine Learning*, pages 1247–1255, 2013.

Leo Breiman and Jerome H Friedman. Estimating optimal transformations for multiple regression and correlation. *Journal of the American statistical Association*, 80(391):580–598, 1985.

Jean-François Cardoso. Dependence, correlation and gaussianity in independent component analysis. *Journal of Machine Learning Research*, 4:1177–1203, 2003.

Matthew Chalk, Olivier Marre, and Gasper Tkacik. Relevant sparse codes with variational information bottleneck. In *Advances in Neural Information Processing Systems*, pages 1957–1965, 2016.

Gal Chechik, Amir Globerson, Naftali Tishby, and Yair Weiss. Information bottleneck for Gaussian variables. *Journal of Machine Learning Research*, 6:165–188, 2005.

Scott Saobing Chen and Ramesh A Gopinath. Gaussianization. In *Advances in Neural Information Processing Systems*, pages 423–429, 2001.

Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

Nir Friedman, Ori Mosenzon, Noam Slonim, and Naftali Tishby. Multivariate information bottleneck. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 152–161. Morgan Kaufmann Publishers Inc., 2001.

Walter R Gilks. *Markov chain monte carlo*. Wiley Online Library, 2005.

Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

Ron M Hecht, Elad Noor, and Naftali Tishby. Speaker recognition by Gaussian information bottleneck. In *INTERSPEECH*, pages 1567–1570, 2009.

Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent component analysis*, volume 46. John Wiley & Sons, 2004.

Jim Kay. Feature discovery under contextual supervision using mutual information. In *Neural Networks, 1992. IJCNN., International Joint Conference on*, volume 4, pages 79–84. IEEE, 1992.

Arto Klami and Samuel Kaski. Non-parametric dependent components. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, 2005.

Artemy Kolchinsky, Brendan D Tracey, and David H Wolpert. Nonlinear information bottleneck. *arXiv preprint arXiv:1705.02436*, 2017.

Pei Ling Lai and Colin Fyfe. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10(05):365–377, 2000.

HO Lancaster. Correlations and canonical forms of bivariate distributions. *The Annals of Mathematical Statistics*, 34(2):532–538, 1963.

Valero Laparra, Gustavo Camps-Valls, and Jesús Malo. Iterative Gaussianization: from ICA to random rotations. *IEEE Transactions on Neural Networks*, 22(4):537–549, 2011.

Tomer Michaeli, Weiran Wang, and Karen Livescu. Nonparametric canonical correlation analysis. In *International Conference on Machine Learning*, pages 1967–1976, 2016.

Gaspard Monge. *Mémoire sur la théorie des déblais et des remblais*. De l'Imprimerie Royale, 1781.

Svetlozar T Rachev and Ludger Rüschendorf. *Mass Transportation Problems: Volume I: Theory*, volume 1. Springer Science & Business Media, 1998.

Mélanie Rey and Volker Roth. Meta-Gaussian information bottleneck. In *Advances in Neural Information Processing Systems*, pages 1916–1924, 2012.

Elad Schneidman, Noam Slonim, Naftali Tishby, R deRuyter van Steveninck, and William Bialek. Analyzing neural codes using the information bottleneck method. *Advances in Neural Information Processing Systems*, 2001.

Elad Schneidman, Michael J Berry, Ronen Segev, and William Bialek. Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature*, 440 (7087):1007–1012, 2006.

Ofer Shayevitz and Meir Feder. Optimal feedback communication via posterior matching. *IEEE Transactions on Information Theory*, 57(3):1186–1222, 2011.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

Janne Sinkkonen and Samuel Kaski. Clustering based on conditional distributions in an auxiliary space. *Neural Computation*, 14(1):217–239, 2002.

Noam Slonim. *The information bottleneck: Theory and applications*. PhD thesis, Hebrew University of Jerusalem, 2002.

Noam Slonim and Naftali Tishby. Document clustering using word clusters via the information bottleneck method. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 208–215. ACM, 2000.

Noam Slonim, Gurinder Singh Atwal, Gašper Tkačik, and William Bialek. Information-based clustering. *Proceedings of the National Academy of Sciences of the United States of America*, 102(51):18297–18302, 2005.

Naftali Tishby and Daniel Polani. Information theory of decisions and actions. In *Perception-action cycle*, pages 601–636. Springer, 2011.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *IEEE Information Theory Workshop (ITW)*, pages 1–5, 2015.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. In *Proceedings of 37th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

Journal of Machine Learning Research 18 (2018) 1-5 Submitted 7/17; Revised 4/18; Published 4/18

tick: a Python Library for Statistical Learning, with an emphasis on Hawkes Processes and Time-Dependent Models

Emmanuel Bacry emmanuel.bacry@polytechnique.edu
Martin Bompaire martin.bompaire@polytechnique.edu
Philip Deegan philip.deegan@polytechnique.edu
Stéphane Gaïffas stephane.gaiffas@polytechnique.edu
Søren V. Poulsen soren.poulsen@polytechnique.edu
Centre de Mathématiques Appliquées
École polytechnique
UMR 7641, 91128 Palaiseau, France

Editor: Geoff Holmes

Abstract

This paper introduces `tick`, is a statistical learning library for Python 3, with a particular emphasis on time-dependent models, such as point processes, tools for generalized linear models and survival analysis. The core of the library provides model computational classes, solvers and proximal operators for regularization. It relies on a C++ implementation and state-of-the-art optimization algorithms to provide very fast computations in a single node multi-core setting. Source code and documentation can be downloaded from `https://github.com/X-DataInitiative/tick`.

Keywords: Statistical Learning; Python; Hawkes processes; Optimization; Generalized linear models; Point Process; Survival Analysis

1. Introduction

The aim of the `tick` library is to provide for the Python community a large set of tools for statistical learning, previously not available in any framework. Though `tick` focuses on time-dependent modeling, it actually introduces a set of tools that go way beyond this particular set of models, thanks to a highly modular optimization toolbox. It benefits from thorough documentation (including tutorials with many examples), and a strongly tested Python API that brings to the scientific community cutting-edge algorithms with a high level of customization. Optimization algorithms such as SVRG (Johnson and Zhang, 2013) or SDCA (Shalev-Shwartz and Zhang, 2013) are among the several optimization algorithms available in `tick` that can be applied (in a modular way) to a large variety of models. An emphasis is placed on time-dependent models: from the Cox regression model (Andersen et al., 2012), a very popular model in survival analysis, to Hawkes processes, used in a wide range of applications such as geophysics (Ogata, 1988), finance (Bacry et al., 2015) and more recently social networks (Zhou et al., 2013; Xu et al., 2016). To the best of our knowledge, `tick` is the most comprehensive library that deals with Hawkes processes, since it brings parametric and nonparametric estimators of these models to a new accessibility level.

2. Existing Libraries

The `tick` library follows, whenever possible, `scikit-learn`'s API (Pedregosa et al., 2011; Buitinck et al., 2013) which is well-known for its completeness and ease of use. However, while `scikit-learn` targets a wide spectrum, `tick` has a more specific objective: implementing highly-optimized algorithms with a particular emphasis on time-dependent modeling (not proposed in `scikit-learn`). The `tick` optimization toolbox relies on state-of-the-art optimization algorithms, and is implemented in a very modular way. It allows more possibilities than other `scikit-learn` API based optimization libraries such as `lightning`[1].

A wide variety of time-dependent models are taken care of by `tick`, which makes it the most comprehensive library that deals with Hawkes processes for instance, by including the main inference algorithms from the literature. Despite the growing interest in Hawkes models, very few open source packages are available. There are mainly three of them. The library `pyhawkes`[2] proposes a small set of Bayesian inference algorithms for Hawkes process. `hawkes` R[3] is an R-based library that provides a single estimation algorithm, and is hardly optimized. Finally, `PtPack`[4] a C++ library which proposes parametric maximum likelihood estimators, with sparsity-inducing regularizations. However, since `tick` is a Python library, it addresses a different community to PtPack. Moreover, as illustrated below, `tick` provides better performance than `PtPack`.

3. Package Architecture

The `tick` library has four main modules: `tick.hawkes` for Hawkes processes (see Section 4 for a detailed review), `tick.linear_model` with linear, logistic and Poisson regression, `tick.robust` for robust linear models and `tick.survival` for survival analysis. Each of these modules provide simulation tools and learners to easily learn from data. The core of `tick` is made of easy to combine penalization techniques (`tick.prox` module) and several convex solvers (`tick.solver`), to train almost any available model in the library, see Table 1 for a non-exhaustive list of possible combinations. An exhaustive list is available on the documentation web page[5], and is given in Figure 7 of the supplementary material.

4. Hawkes

Distributing a comprehensive open source library for Hawkes processes is one of the primary aims of the `tick` library: it provides many non-parametric and parametric estimation algorithms that are listed in Table 2 as well as simulation tools for many kernel types as shown in Figure 5 of the supplementary material. This diversity of algorithms is illustrated in Figure 1 in which we show how two kernels of different shapes are estimated by four different algorithms. A first use case for modeling high-frequency financial data is given in Figure 2 (with the associated Python code), while a second use-case about propagation analysis of earthquake aftershocks can be found in Figure 3.

1. `http://contrib.scikit-learn.org/lightning`
2. `https://github.com/slinderman/pyhawkes`
3. `https://cran.r-project.org/web/packages/hawkes/hawkes.pdf`
4. `https://github.com/dunan/MultiVariatePointProcess`
5. `https://x-datainitiative.github.io/tick/`

Model	Proximal operator	Solver
Linear regression	SLOPE	Gradient Descent
Logistic regression	L1 (Lasso)	Stochastic Variance Reduced Gradient
Poisson regression	Total Variation	Stochastic Gradient Descent
Cox regression	Group L1	Accelerated Gradient Descent
Hawkes with exp. kernels	L2 (Ridge)	Stochastic Dual Coordinate Ascent

Table 1: `tick` allows the user to combine many models, prox and solvers

Non Parametric	Parametric
EM (Lewis and Mohler, 2011)	Single exponential kernel
Basis kernels (Zhou et al., 2013)	Sum of exponentials kernels
Wiener-Hopf (Bacry and Muzy, 2014)	Sum of gaussians kernels (Xu et al., 2016)
NPHC (Achab et al., 2017)	ADM4 (Zhou et al., 2013)

Table 2: Models and estimation techniques for Hawkes processes available in `tick`

5. Benchmarks

In Figure 4, we benchmark computational times for both simulation and estimation of Hawkes processes (with exponential kernels) using `hawkes` R (where only simulation is available), `PtPack` and `tick` on respectively 2, 4 and 16 cores. The model fits in plots "Fit" and "Multicore fit" are compared on simulated 16-dimensional Hawkes processes, with an increasing number of events: small=5×10^4, medium=2×10^5, large=10^6, xlarge= 5×10^7, while 200, 400 and 750 dimensional Hawkes processes are fitted in plot "High-dimensional fitting". We observe that `tick` outperforms by several orders of magnitudes both `hawkes` R and `PtPack`, in particular for large datasets. Benchmarks against `scikit-learn` for logistic regression are also provided in Figure 6 from the supplementary material.

Acknowledgments

The authors thank the Data-science Initiative of École polytechnique and Intel® for supporting `tick` development

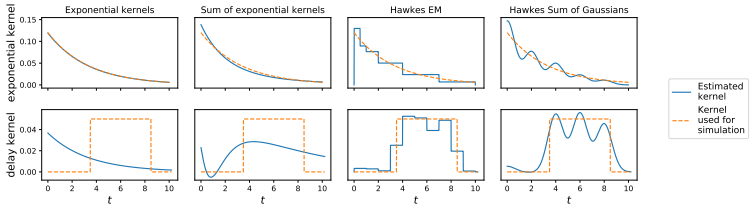


Figure 1: Illustration of different kernel shapes and estimations obtained by `tick` on two 1-dimensional simulated Hawkes processes.


```
timestamps_list = fetch_hawkes_bund_data()

hawkes_learner = HawkesConditionalLaw(
  claw_method="log", quad_method="log", n_quad=50,
  delta_lag=0.1, min_lag=5e-4, max_lag=500,
  min_support=1e-4, max_support=1, n_threads=4)

hawkes_learner.fit(timestamps_list)

plot_hawkes_kernel_norms(
  hawkes_learner, node_names= "P_u", "P_d", "T_a", "T_b" )
```

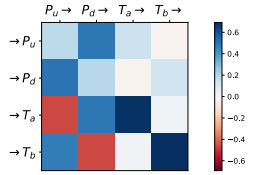


Figure 2: Kernel norms of a Hawkes process fitted on high-frequency financial data from the Bund market (Bacry et al., 2016) where P_u (resp. P_d) counts the number of upward (resp. downward) mid-price moves and T_a (resp. T_b) counts the number of market orders at the ask (resp. bid) that do not move the price.

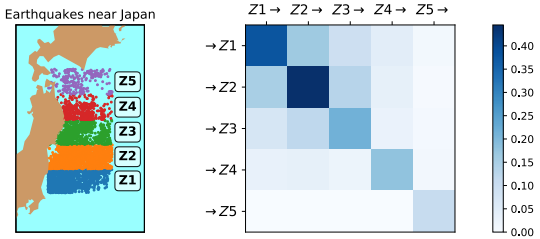


Figure 3: Modeling of earthquake propagation with Hawkes processes on a dataset from Ogata (1988). The left hand side gives the location of the earthquakes while right hand side illustrates the propagation matrix, namely how likely an earthquake in a given zone will trigger an aftershock in another zone.

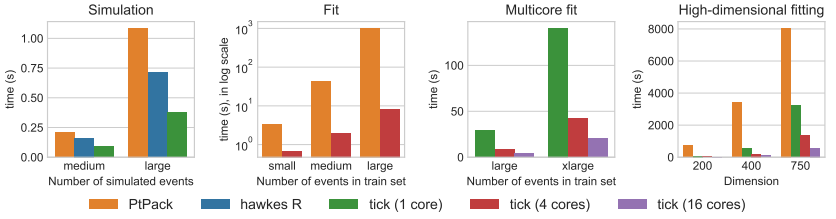


Figure 4: Computational timings of `tick` versus `PtPack` and `hawkes` R. `tick` strongly outperforms both libraries for simulation and fitting (note that the "Fit" graph is in log-scale). "Multicore fit" and "High-dimensional fitting" plots show that `tick` benefits from multi-core environments to speed up computations.

References

M. Achab, E. Bacry, S. Gaïffas, I. Mastromatteo, and J.-F. Muzy. Uncovering causality from multivariate hawkes integrated cumulants. In *International Conference on Machine Learning*, pages 1–10, 2017.

P. K. Andersen, O. Borgan, R. D. Gill, and N. Keiding. *Statistical Models based on Counting Processes*. Springer Science, 2012.

E. Bacry and J.-F. Muzy. Second order statistics characterization of hawkes processes and non-parametric estimation. *arXiv preprint arXiv:1401.0903*, 2014.

E. Bacry, I. Mastromatteo, and J.-F. Muzy. Hawkes processes in finance. *Market Microstructure and Liquidity*, 1(01):1550005, 2015.

E. Bacry, T. Jaisson, and J.-F. Muzy. Estimation of slowly decreasing hawkes kernels: application to high-frequency order book dynamics. *Quantitative Finance*, 16(8):1179–1201, 2016.

L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J VanderPlas, A. Joly, B. Holt, and G. Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

E. Lewis and G. Mohler. A nonparametric em algorithm for multiscale hawkes processes. *preprint*, pages 1–16, 2011.

Y. Ogata. Statistical models for earthquake occurrences and residual analysis for point processes. *Journal of the American Statistical Association*, 83(401):9–27, 1988.

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

S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14(Feb):567–599, 2013.

H. Xu, M. Farajtabar, and H. Zha. Learning granger causality for hawkes processes. In *Proceedings of International Conference on Machine Learning*, pages 1717–1726, 2016.

K. Zhou, H. Zha, and L. Song. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *AISTATS*, volume 31, pages 641–649, 2013.

Journal of Machine Learning Research 18 (2018) 1-5 Submitted 10/17; Revised 4/18; Published 4/18

SGDLibrary: A MATLAB library for stochastic optimization algorithms

Hiroyuki Kasai KASAI@IS.UEC.AC.JP
Graduate School of Informatics and Engineering
The University of Electro-Communications
Tokyo, 182-8585, Japan

Editor: Geoff Holmes

Abstract

We consider the problem of finding the minimizer of a function $f : \mathbb{R}^d \to \mathbb{R}$ of the finite-sum form $\min f(w) = 1/n \sum_i^n f_i(w)$. This problem has been studied intensively in recent years in the field of machine learning (ML). One promising approach for large-scale data is to use a stochastic optimization algorithm to solve the problem. SGDLibrary is a readable, flexible and extensible pure-MATLAB library of a collection of stochastic optimization algorithms. The purpose of the library is to provide researchers and implementers a comprehensive evaluation environment for the use of these algorithms on various ML problems.

Keywords: Stochastic optimization, stochastic gradient, finite-sum minimization problem, large-scale optimization problem

1. Introduction

This work aims to facilitate research on stochastic optimization for large-scale data. We particularly address a regularized finite-sum minimization problem defined as

$$\min_{w \in \mathbb{R}^d} f(w) := \frac{1}{n} \sum_{i=1}^{n} f_i(w) = \frac{1}{n} \sum_{i=1}^{n} L(w, x_i, y_i) + \lambda R(w), \tag{1}$$

where $w \in \mathbb{R}^d$ represents the model parameter and n denotes the number of samples (x_i, y_i). $L(w, x_i, y_i)$ is the loss function and $R(w)$ is the regularizer with the regularization parameter $\lambda \geq 0$. Widely diverse machine learning (ML) models fall into this problem. Considering $L(w, x_i, y_i) = (w^T x_i - y_i)^2$, $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$ and $R(w) = \|w\|_2^2$, this results in an ℓ_2-norm regularized linear regression problem (a.k.a. ridge regression) for n training samples $(x_1, y_1), \cdots, (x_n, y_n)$. In the case of binary classification with the desired class label $y_i \in \{+1, -1\}$ and $R(w) = \|w\|_1$, an ℓ_1-norm regularized logistic regression (LR) problem is obtained as $f_i(w) = \log(1 + \exp(-y_i w^T x_i)) + \lambda \|w\|_1$, which encourages the sparsity of the solution of w. Other problems covered are matrix completion, support vector machines (SVM), and sparse principal components analysis, to name but a few.

Full gradient descent (a.k.a. steepest descent) with a step-size η is the most straightforward approach for (1), which updates as $w_{k+1} \leftarrow w_k - \eta \nabla f(w_k)$ at the k-th iteration. However, this is expensive when n is extremely large. In fact, one needs a sum of n calculations of the inner product $w^T x_i$ in the regression problems above, leading to $\mathcal{O}(nd)$ cost overall per iteration. For this issue, a popular and effective alternative is *stochastic gradient descent* (SGD), which updates as $w_{k+1} \leftarrow w_k - \eta \nabla f_i(w_k)$ for the i-th sample uniformly at random (Robbins and Monro, 1951; Bottou, 1998). SGD assumes an *unbiased estimator* of the full gradient as $\mathbb{E}_i[\nabla f_i(w^k)] = \nabla f(w^k)$. As the update rule represents, the calculation cost is independent of n, resulting in $\mathcal{O}(d)$ per iteration. Furthermore, *mini-batch* SGD (Bottou, 1998) calculates $1/|\mathcal{S}_k| \sum_{i \in \mathcal{S}_k} \nabla f_i(w_k)$, where $\mathcal{S}_k$ is the set of samples of size $|\mathcal{S}_k|$. SGD needs a *diminishing* step-size algorithm to guarantee convergence, which causes a severe slow convergence rate (Bottou, 1998). To accelerate this rate, we have two active research directions in ML; *Variance reduction* (VR) techniques (Johnson and Zhang, 2013; Roux et al., 2012; Shalev-Shwartz and Zhang, 2013; Defazio et al., 2014; Nguyen et al., 2017) that explicitly or implicitly exploit a full gradient estimation to reduce the variance of the noisy stochastic gradient, leading to superior convergence properties. Another promising direction is to modify deterministic *second-order* algorithms into stochastic settings, and solve the potential problem of *first-order* algorithms for *ill-conditioned* problems. A direct extension of *quasi-Newton* (QN) is known as online BFGS (Schraudolph et al., 2007). Its variants include a regularized version (RES) (Mokhtari and Ribeiro, 2014), a limited memory version (oLBFGS) (Schraudolph et al., 2007; Mokhtari and Ribeiro, 2015), a stochastic QN (SQN) (Byrd et al., 2016), an incremental QN (Mokhtari et al., 2017), and a non-convex version. Lastly, hybrid algorithms of the SQN with VR are proposed (Moritz et al., 2016; Kolte et al., 2015). Others include (Duchi et al., 2011; Bordes et al., 2009).

The performance of stochastic optimization algorithms is strongly influenced not only by the distribution of data but also by the step-size algorithm (Bottou, 1998). Therefore, we often encounter results that are completely different from those in papers in every experiment. Consequently, an evaluation framework to test and compare the algorithms at hand is crucially important for fair and comprehensive experiments. One existing tool is *Lightning* (Blondel and Pedregosa, 2016), which is a Python library for large-scale ML problems. However, its supported algorithms are limited, and the solvers and the problems such as classifiers are mutually connected. Moreover, the implementations utilize Cython, which is a C-extension for Python, for efficiency. Subsequently, they decrease users' readability of code, and also make users' evaluations and extensions more complicated. SGDLibrary is a readable, flexible and extensible pure-MATLAB library of a collection of stochastic optimization algorithms. The library is also operable on GNU Octave. The purpose of the library is to provide researchers and implementers a collection of state-of-the-art stochastic optimization algorithms that solve a variety of large-scale optimization problems such as linear/non-linear regression problems and classification problems. This also allows researchers and implementers to easily extend or add solvers and problems for further evaluation. To the best of my knowledge, no report in the literature and no library describe a comprehensive experimental environment specialized for stochastic optimization algorithms. The code is available at `https://github.com/hiroyuki-kasai/SGDLibrary`.

2. Software architecture

The software architecture of SGDLibrary follows a typical *module-based* architecture, which separates *problem descriptor* and *optimization solver*. To use the library, the user selects one problem descriptor of interest and no less than one optimization solvers to be compared.

Problem descriptor: The problem descriptor, denoted as `problem`, specifies the problem of interest with respect to w, noted as `w` in the library. This is implemented by MATLAB `classdef`. The user does nothing other than calling a problem definition function, for instance, `logistic_regression()` for the ℓ_2-norm regularized LR problem. Each problem definition includes the functions necessary for solvers; (i) (full) cost function $f(w)$, (ii) mini-batch stochastic derivative `v`$=1/|\mathcal{S}|\nabla f_{i\in\mathcal{S}}(w)$ for the set of samples $\mathcal{S}$. (iii) stochastic Hessian (Bordes et al., 2009), and (iv) stochastic Hessian-vector product for a vector `v`. The built-in problems include, for example, ℓ_2-norm regularized multidimensional linear regression, ℓ_2-norm regularized linear SVM, ℓ_2-norm regularized LR, ℓ_2-norm regularized softmax classification (multinomial LR), ℓ_1-norm multidimensional linear regression, and ℓ_1-norm LR. The problem descriptor provides additional specific functions. For example, the LR problem includes the prediction and the classification accuracy calculation functions.

Optimization solver: The optimization solver implements the main routine of the stochastic optimization algorithm. Once a solver function is called with one selected problem descriptor `problem` as the first argument, it solves the optimization problem by calling some corresponding functions via `problem` such as the cost function and the stochastic gradient calculation function. Examples of the supported optimization solvers in the library are listed in categorized groups as; (i) **SGD methods:** Vanila SGD (Robbins and Monro, 1951), SGD with classical momentum, SGD with classical momentum with Nesterov's accelerated gradient (Sutskever et al., 2013), AdaGrad (Duchi et al., 2011), RMSProp, AdaDelta, Adam, and AdaMax, (ii) **Variance reduction (VR) methods:** SVRG (Johnson and Zhang, 2013), SAG (Roux et al., 2012), SAGA (Defazio et al., 2014), and SARAH (Nguyen et al., 2017), (iii) **Second-order methods**: SQN (Bordes et al., 2009), oBFGS-Inf (Schraudolph et al., 2007; Mokhtari and Ribeiro, 2015), oBFGS-Lim (oLBFGS) (Schraudolph et al., 2007; Mokhtari and Ribeiro, 2015), Reg-oBFGS-Inf (RES) (Mokhtari and Ribeiro, 2014), and Damp-oBFGS-Inf, (iv) **Second-order methods with VR**: SVRG-LBFGS (Kolte et al., 2015), SS-SVRG (Kolte et al., 2015), and SVRG-SQN (Moritz et al., 2016), and (v) **Else**: BB-SGD and SVRG-BB. The solver function also receives optional parameters as the second argument, which forms a *struct*, designated as `options` in the library. It contains elements such as the maximum number of epochs, the batch size, and the step-size algorithm with an initial step-size. Finally, the solver function returns to the caller the final solution `w` and rich statistical information, such as a record of the cost function values, the optimality gap, the processing time, and the number of gradient calculations.

Others: SGDLibrary accommodates a *user-defined* step-size algorithm. This accommodation is achieved by setting as `options.stepsizefun=@my_stepsize_alg`, which is delivered to solvers. Additionally, when the regularizer $R(w)$ in the minimization problem (1) is a non-smooth regularizer such as the ℓ_1-norm regularizer $\|w\|_1$, the solver calls the *proximal operator* as `problem.prox(w,stepsize)`, which is the wrapper function defined in each problem. The ℓ_1-norm regularized LR problem, for example, calls the *soft-threshold* function as `w = prox(w,stepsize)=soft_thresh(w,stepsize*lambda)`, where `stepsize` is the step-size η and `lambda` is the regularization parameter $\lambda > 0$ in (1).

3. Tour of the SGDLibrary

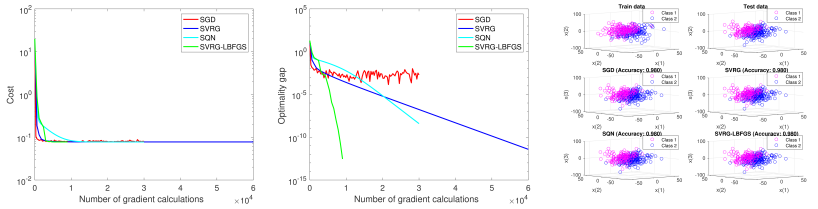
We embark on a tour of SGDLibrary exemplifying the ℓ_2-norm regularized LR problem. The LR model generates n pairs of (x_i, y_i) for an unknown model parameter w, where x_i is an input d-dimensional vector and $y_i \in \{-1, 1\}$ is the binary class label, as $P(y_i|x_i, w) = 1/(1 + \exp(-y_i w^T x_i))$. The problem seeks w that fits the regularized LR model to the generated data (x_i, y_i). This problem is cast as a minimization problem as $\min f(w) := 1/n \sum_{i=1}^{n} \log[1 + \exp(-y_i w^T x_i)] + \lambda/2\|w\|^2$. The code for this problem is in Listing 1.

```
% generate synthetic 300 samples of dimension 3 for logistic regression
d = logistic_regression_data_generator(300,3);
% define logistic regression problem
problem = logistic_regression(d.x_train,d.y_train,d.x_test,d.y_test);

options.w_init = d.w_init;                          % set initial value
options.step_init = 0.01;                           % set initial stepsize
options.verbose = 1;                                % set verbose mode
[w_sgd, info_sgd] = sgd(problem, options);          % perform SGD solver
[w_svrg, info_svrg] = svrg(problem, options);       % perform SVRG solver
[w_svrg, info_svrg] = sqn(problem, options);      % perform SQN solver
% display cost vs. number of gradient evaluations
display_graph('grad_calc_count','cost',{'SGD','SVRG'},...
                {w_sgd,w_svrg},{info_sgd,info_svrg});
```

Listing 1: Demonstration code for logistic regression problem.

First, we generate train/test datasets `d` using `logistic_regression_data_generator()`, where the input feature vector is with $n = 300$ and $d = 3$. $y_i \in \{-1, 1\}$ is its class label. The LR problem is defined by calling `logistic_regression()`, which internally contains the functions for cost value, the gradient and the Hessian. This is stored in `problem`. Then, we execute solvers, i.e., SGD and SVRG, by calling solver functions, i.e., `sgd()` and `svrg()` with `problem` and `options` after setting some options into the `options` struct. They return the final solutions of `{w_sgd,w_svrg}` and the statistical information `{info_sgd,info_svrg}`. Finally, `display_graph()` visualizes the behavior of the cost function values in terms of the number of gradient evaluations. It is noteworthy that each algorithm requires a different number of evaluations of samples in each epoch. Therefore, it is common to use this value to evaluate the algorithms instead of the number of iterations. Illustrative results additionally including SQN and SVRG-LBFGS are presented in Figure 1, which are generated by `display_graph()`, and `display_classification_result()` specialized for classification problems. Thus, SGDLibrary provides rich visualization tools as well.



(a) Cost function value (b) Optimality gap (c) Classification result

Figure 1: Results of ℓ_2-norm regularized logistic regression problem.

References

M. Blondel and F. Pedregosa. Lightning: large-scale linear classification, regression and ranking in Python, 2016. URL `https://doi.org/10.5281/zenodo.200504`.

A. Bordes, L. Bottou, and P. Callinari. SGD-QN: Careful quasi-Newton stochastic gradient descent. *JMLR*, 10:1737–1754, 2009.

L. Bottou. Online algorithm and stochastic approximations. In David Saad, editor, *On-Line Learning in Neural Networks*. Cambridge University Press, 1998.

R. H. Byrd, S. L. Hansen, J. Nocedal, and Y. Singer. A stochastic quasi-Newton method for large-scale optimization. *SIAM J. Optim.*, 26(2), 2016.

A. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NIPS*, 2014.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12:2121–2159, 2011.

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS*, 2013.

R. Kolte, M. Erdogdu, and A. Ozgur. Accelerating SVRG via second-order information,. In *OPT2015*, 2015.

A. Mokhtari and A. Ribeiro. RES: Regularized stochastic BFGS algorithm. *IEEE Trans. on Signal Process.*, 62(23):6089–6104, 2014.

A. Mokhtari and A. Ribeiro. Global convergence of online limited memory BFGS. *JMLR*, 16:3151–3181, 2015.

A. Mokhtari, M. Eizen, and A. Ribeiro. An incremental quasi-Newton method with a local superlinear convergence rate. In *ICASSP*, 2017.

P. Moritz, R. Nishihara, and M. I. Jordan. A linearly-convergent stochastic L-BFGS algorithm. In *AISTATS*, 2016.

L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takac. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *ICML*, 2017.

H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Statistics*, 22 (3):400–407, 1951.

N. L. Roux, M. Schmidt, and F. R. Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *NIPS*, 2012.

N. N. Schraudolph, J. Yu, and S. Gunter. A stochastic quasi-Newton method for online convex optimization. In *AISTATS*, 2007.

S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *JMLR*, 14:567–599, 2013.

I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, 2013.

Journal of Machine Learning Research 18 (2018) 1-34 Submitted 7/16; Revised 7/17; Published 4/18

Reward Maximization Under Uncertainty: Leveraging Side-Observations on Networks

Swapna Buccapatnam SB646F@ATT.COM
AT&T Labs Research, Middletown, NJ 07748, USA

Fang Liu LIU.3977@OSU.EDU
Atilla Eryilmaz ERYILMAZ.2@OSU.EDU
Department of Electrical and Computer Engineering
The Ohio State University
Columbus, OH 43210, USA

Ness B. Shroff SHROFF.11@OSU.EDU
Department of Electrical and Computer Engineering and Computer Science Engineering
The Ohio State University
Columbus, OH 43210, USA

Editor: Shie Mannor

Abstract

We study the stochastic multi-armed bandit (MAB) problem in the presence of side-observations across actions that occur as a result of an underlying network structure. In our model, a bipartite graph captures the relationship between actions and a common set of unknowns such that choosing an action reveals observations for the unknowns that it is connected to. This models a common scenario in online social networks where users respond to their friends' activity, thus providing side information about each other's preferences. Our contributions are as follows: 1) We derive an asymptotic lower bound (with respect to time) as a function of the bi-partite network structure on the regret of any *uniformly good policy* that achieves the maximum long-term average reward. 2) We propose two policies - a randomized policy; and a policy based on the well-known upper confidence bound (UCB) policies - both of which explore each action at a rate that is a function of its network position. We show, under mild assumptions, that these policies achieve the asymptotic lower bound on the regret up to a multiplicative factor, independent of the network structure. Finally, we use numerical examples on a real-world social network and a routing example network to demonstrate the benefits obtained by our policies over other existing policies.

Keywords: Multi-armed Bandits, Side Observations, Bipartite Graph, Regret Bounds

1. Introduction

Multi-armed bandit (MAB) problems are well-known models of sequential decision-making under uncertainty (Lai and Robbins, 1985) and have lately been used to model new and exciting decision problems in content recommendation systems, online advertising platforms, and social networks, among others. In the classical MAB setting, at each time, a bandit policy must choose an action from a set of actions with unknown probability distributions. Choosing an action gives a random reward drawn from the distribution of the action. The regret of any policy is defined as the difference between the total reward obtained from the action with the highest average reward and the given policy's total reward. The goal is to find policies that minimize the expected regret over time.

In this work, we consider an important extension to the classical MAB problem, where choosing an action not only generates a reward from that action, but also reveals important information for a subset of the remaining actions. We model this relationship between different actions using a bipartite graph between the set of actions and a common set of unknowns (see Figure 2). The reward from each action is a known function of a subset of the unknowns (called its parents) and choosing an action reveals observations from each of its parents. Our main objective in this work is to leverage such a structure to improve scalability of bandit policies in terms of the action/decision space.

Such an information structure between actions becomes available in a variety of applications. For example, consider the problem of *routing* in communication networks, where packets are to be sent over a set of links from source to destination (called a path or a route) in order to minimize the delay. Here, the total delay on each path is the sum of individual link delays, which are unknown. In addition, traveling along a path reveals observations for delays on each of constituent links. Hence, each path provides additional information for all other paths that share some of their links with it. In this example, actions correspond to a set of feasible paths and the set of unknowns corresponds to random delays on all the links in the network.

Another example occurs in advertising in online social networks through promotional offers. Suppose a user is offered a promotion/discounted price for a product in return for advertising it to his friends/neighbors in an online social network. The influence of the user is then measured by the friends that respond to his message through comments/likes, etc. Each user has an intrinsic unknown probability of responding to such messages on social media. Here, the set of actions correspond to the set of users (to whom promotions are given) and the set of unknowns are the users' intrinsic responsiveness to such promotions.

In this work, we aim to characterize the asymptotic lower bound on the regret for a general stochastic multi-armed bandit problem in the presence of such an information structure and investigate policies that achieve this lower bound by taking the network structure into account. Our main contributions are as follows:

- We model the MAB problem in the presence of additional structure and derive an asymptotic (with respect to time) lower bound (as a function of the network structure) on the regret of any uniformly good policy which achieves the maximum long term average reward. This lower bound is presented in terms of the optimal value of a linear program (LP).
- Motivated by the LP lower bound, we propose and investigate the performance of a randomized policy, we call ϵ_t-greedy-LP policy, as well as an upper confidence bound based policy, we call UCB-LP policy. Both of these policies *explore each action at a rate that is a function of its location in the network.* We show under some mild assumptions that these policies are optimal in the sense that they achieve the asymptotic lower bound on the regret up to a multiplicative constant that is independent of the network structure.

The model considered in this work is an important first step in the direction of more general models of interdependence across actions. For this model, we show that as the

number of actions becomes large, significant benefits can be obtained from policies that explicitly take network structure into account. While ϵ_t-greedy-LP policy explores actions at a rate proportional to their network position, its exploration is oblivious to the average rewards of the sub-optimal actions. On the other hand, UCB-LP policy takes into account both the upper confidence bounds on the mean rewards as well as network position of different actions at each time.

2. Related Work

The seminal work of Lai and Robbins (1985) showed that the asymptotic lower bound on the regret of any uniformly good policy scales logarithmically with time with a multiplicative constant that is a function of the distributions of actions. Further, Lai and Robbins (1985) provide constructive policies called Upper Confidence Bound (UCB) policies based on the concept of optimism in the face of uncertainty that asymptotically achieve the lower bound. More recently, Auer et al. (2002) considered the case of bounded rewards and propose simpler sample-mean-based UCB policies and a decreasing-ϵ_t-greedy policy that achieve logarithmic regret uniformly over time, rather than only asymptotically as in the previous works.

The traditional multi-armed bandit policies incur a regret that is linear in the number of suboptimal arms. This makes them unsuitable in settings such as content recommendation, advertising, etc, where the action space is typically very large. To overcome this difficulty, richer models specifying additional information across reward distributions of different actions have been studied, such as dependent bandits by Pandey et al. (2007), $\mathcal{X}$-armed bandits by Bubeck et al. (2011), linear bandits by Rusmevichientong and Tsitsiklis (2010), contextual side information in bandit problems by Li et al. (2010), combinatorial bandits by Chen et al. (2013) etc..

The works of Mannor and Shamir (2011), Caron et al. (2012), and Buccapatnam et al. (2014) proposed to handle the large number of actions by assuming that choosing an action reveals observations from a larger set of actions. In this setting, actions are embeded in a network and choosing an action provides observations for all the immediate neighbors in the network. The policies proposed in Mannor and Shamir (2011) achieve the best possible regret in the adversarial setting (see Bubeck and Cesa-Bianchi (2012) for a survey of adversarial MABs) with side-observations, and the regret bounds of these policies are in terms of the independence number of the network. The stochastic version of this problem is introduced in Caron et al. (2012) and Buccapatnam et al. (2014), which improves upon the results in Caron et al. (2012). In Buccapatnam et al. (2014), the authors derive a lower bound on regret in stochastic network setting for any uniformly good policy and propose two policies that achieve this lower bound in these settings up to a multiplicative constant. Our current work extends the setting in Caron et al. (2012); Buccapatnam et al. (2014) to a more general and important graph feedback structure between the set of actions and a set of common unknowns, which may or may not coincide with the set of actions available to the decision maker. The setting of Mannor and Shamir (2011), Caron et al. (2012), and Buccapatnam et al. (2014) is a special case of this general feedback structure, where the set of unknowns and the set of actions coincide.

More recently, Cohen et al. (2016), have studied the multi-armed bandit problem with a graph based feedback structure similar to Mannor and Shamir (2011), and Buccapatnam et al. (2014). However, they assume that the graph structure is never fully revealed. In contrast, in many cases such as the problem of routing in communication networks and the problem of influence maximization in social networks, the graph structure is revealed or learnt apriori and is known. When the graph structure is known, the authors in Buccapatnam et al. (2014) propose algorithms for the stochastic setting whose regret performance is bounded by the domination number of the graph. In contrast, the algorithms proposed in Cohen et al. (2016) assume that the graph is unknown and achieve a regret that is upper bounded by the independence number of the graph. (Note that the independence number of a graph is larger than or equal to the domination number). Our current work proposes a general feedback structure of which Buccapatnam et al. (2014) and Cohen et al. (2016) can be viewed as a special case. Moreover, we present algorithms that benefit significantly from the knowledge of the graph feedback structure.

The setting of combinatorial bandits (CMAB) by Chen et al. (2013) is also closely related to our work. In CMAB, a subset of base actions with unknown distributions form super actions and in each round, choosing a super action reveals outcomes of its constituent actions. The reward obtained is a function of these outcomes. The number of super actions and their composition in terms of base actions is assumed to be arbitrary and the policies do not utilize the underlying network structure between base actions and super actions. In contrast, in our work, we derive a regret lower bound in terms of the underlying network structure and propose policies that achieve this bound. This results in markedly improved performance when the number of super actions is not substantially larger than the number of base actions.

3. Problem Formulation

In this section, we formally define the general bandit problem in the presence of side observations across actions. Let $\mathcal{N} = \{1, \ldots, N\}$ denote the collection of *base-arms* with unknown distributions. Subsets of base-arms form *actions*, and are indexed by $\mathcal{K} = \{1, \ldots, K\}$. A decision maker must choose an action $j \in \mathcal{K}$ at each time t and observes the rewards of related base-arms. Let $X_i(t)$ be the reward of base-arm i observed by the decision maker (on choosing some action) at time t. We assume that $\{X_i(t), t \geq 0\}$ are independent and identically distributed (i.i.d.) for each i and $\{X_i(t), \forall i \in \mathcal{N}\}$ are independent for each time t. Let $V_j \subseteq \mathcal{N}$ be the subset of base-arms that are observed when playing action j. Then, we define $S_i = \{j : i \in V_j\}$ as the support of base-arm i, i.e., the decision maker gets observations for base-arm i on playing action $j \in S_i$. When the decision maker chooses action j at time t, he or she observes one realization for each of the random variables $X_i(t)$, $i \in V_j$. The reward of the played action j depends on the outcomes of its related base-arms subset, denoted by $\mathcal{K}_j \subseteq \mathcal{N}$, and some known function $f_j(\cdot)$. Note that $\mathcal{K}_j \subseteq V_j$ because there may be some base-arms that can be observed by action j but not counted as reward in general (see Figure 2 for a concrete example). Let the vector $\vec{X}_j(t) = [X_i(t)]_{i \in \mathcal{K}_j}$ denote the collection of random variables associated with the reward of action j. Then the reward from playing action j at time t is given by $f_j(\vec{X}_j(t))$. We assume that the reward is bounded in $[0, 1]$ for each action. Note that we only assume that the reward function $f_j(\cdot)$ is bounded

and the specific form of $f_j(\cdot)$ and $\mathcal{K}_j$ are determined by the decision maker or the specific problem. Let μ_j be the mean of reward on playing action j.

Side-observation model : The actions $\mathcal{K}$ and base-arms $\mathcal{N}$ form nodes in a network G, represented by a bipartite graph $(\mathcal{K},\mathcal{N},E)$ and the collection $\{\mathcal{K}_j\}_{j\in\mathcal{K}}$. The $N \times K$ adjacency matrix $E = [e_{ij}]$ is defined by $e_{ij} = 1$ if $i \in V_j$ and $e_{ij} = 0$ otherwise. If there is an edge between action j and base-arm i, i.e., $e_{ij} = 1$, then we can observe a realization of base-arm i when choosing action j. Intuitively, the bipartite graph determined by $\{V_j\}_{j\in\mathcal{K}}$ describes the side-observation relationships while the collection $\{\mathcal{K}_j\}_{j\in\mathcal{K}}$ captures the reward structure. Without loss of generality, we assume that $\cup_{i\in\mathcal{K}}\mathcal{K}_i = \mathcal{N}$, which means that there are no useless (dummy) unknown base-arms in the network.

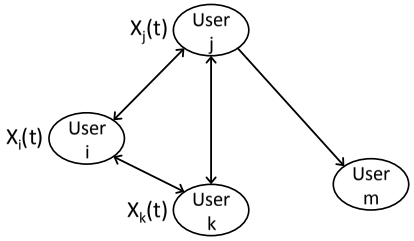


Figure 1: At time t, suppose that the decision maker chooses user i to offer a promotion. He then receives a response $X_i(t)$ from user i. Using the social interconnections, he also observes responses $X_j(t)$ and $X_k(t)$ of i's neighbors j and k.

Figure 1 illustrates the side-observation model for the example of targeting users in an online social network. Such side observations are made possible in settings of online social networks like Facebook by surveying or tracking a user's neighbors' reactions (likes, dislikes, no opinion, etc.) to the user's activity. This is possible when the online social network has a survey or a like/dislike indicator that generates side observations. For example, when user i is offered a promotion, her neighbors may be queried as follows: "User i was recently offered a promotion. Would you also be interested in the offer?[1]"

Figure 2 shows the bipartite graph generated from the example shown in Figure 1. The set of base-arms is the set of users since they act independently according to their own preferences in the promotion, which are unknown to the decision maker. The set of actions is also the set of users because the decision maker wants to target the users with the maximum expected reward. When action j (user j) is chosen, the decision maker observes $X_i(t)$, $X_j(t)$, $X_k(t)$ and $X_m(t)$ from user i, j, k and m since $V_j = \{i,j,k,m\}$. The reward of playing action j depends on $\mathcal{K}_j$ and $f_j(\cdot)$. Suppose $f_j(\vec{X}_j(t)) = \sum_{i\in\mathcal{K}_j} X_i(t)$. Then $\mathcal{K}_j = \{i,j,k,m\}$ means that the reward is the sum of all positive feedbacks. It is also

1. Since, the neighbors do not have any information on whether the user i accepted the promotion, they act independently according to their own preferences in answering this survey. The network itself provides a better way for surveying and obtaining side observations.

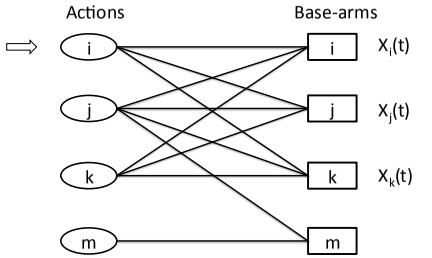


Figure 2: Bipartite graph for the example of targeting users in online social network.

possible that the decision maker set $\mathcal{K}_j = \{j\}$, which means that the reward of playing action j is only the observation from the user j.

The reward function can be quite general (but bounded) to accommodate different settings. Also, the bipartite graph can be more general than social networks in two key ways: 1) Connecting two users in two-hop neighborhood means that the reaction of the friend of my friend is also observable, which is true in Facebook. 2) Connecting two users, say i and j, with similar preference profiles means that the network actively recommends the promotion received by user i to user j even though they are not friends. This has been widely applied in recommender systems such as Yelp.

Objective: An allocation strategy or policy $\boldsymbol{\phi}$ chooses the action to be played at each time. Formally, $\boldsymbol{\phi}$ is a sequence of random variables $\{\phi(t), t \geq 0\}$, where $\phi(t) \in \mathcal{K}$ is the action chosen by policy $\boldsymbol{\phi}$ at time t. Let $T_j^{\phi}(t)$ be the total number of times action j is chosen up to time t by policy $\boldsymbol{\phi}$. For each action, rewards are only obtained when the action is chosen by the policy (side-observations do not contribute to the total reward). Then, the regret of policy $\boldsymbol{\phi}$ at time t for a fixed $\boldsymbol{\mu} = (\mu_1,\ldots,\mu_K)$ is defined by

$$R_{\boldsymbol{\mu}}^{\phi}(t) = \mu^* t - \sum_{j=1}^{K} \mu_j \mathbb{E}[T_j^{\phi}(t)] = \sum_{j=1}^{K} \Delta_j \mathbb{E}[T_j^{\phi}(t)],$$

where $\Delta_j \triangleq \mu^* - \mu_j$ and $\mu^* \triangleq \max_{j\in\mathcal{K}} \mu_j$. Henceforth, we drop the superscript ϕ unless it is required. The objective is to find policies that minimize the rate at which the regret grows as a function of time for every fixed network G. We focus our investigation on the class of uniformly good policies (Lai and Robbins, 1985) defined below:

Uniformly good policies: An allocation rule $\boldsymbol{\phi}$ is said to be uniformly good if for every fixed $\boldsymbol{\mu}$, the following condition is satisfied as $t \to \infty$:

$$R_{\boldsymbol{\mu}}(t) = o(t^b), \text{ for every } b > 0.$$

The above condition implies that uniformly good policies achieve the optimal long term average reward of μ^*. Next, we define two structures that will be useful later to bound the performance of allocation strategies in terms of the network structure G.

Definition 1 *A hitting set D is a subset of $\mathcal{K}$ such that $S_i \cap D \neq \emptyset$, $\forall i \in \mathcal{N}$. Then the hitting set number is $\gamma(G) = \inf_{D \subseteq \mathcal{K}}\{|D| : S_i \cap D \neq \emptyset,\ \forall i \in \mathcal{N}\}$. For example, the set $\{i, m\}$ is a hitting set in Figure 2.*

Definition 2 *A clique C is a subset of $\mathcal{K}$ such that $\mathcal{K}_j \subseteq V_i$, $\forall i, j \in C$. This means that for every action i in C, we can observe the reward of playing any action j in C. A clique cover $\mathcal{C}$ of a network G is a partition of all its nodes into sets $C \in \mathcal{C}$ such that the sub-network formed by each C is a clique. Let $\bar{\chi}(G)$ be the smallest number of cliques into which the nodes of the network G can be partitioned, also called the clique partition number.*

Proposition 3 *For any network G with bipartite graph $(\mathcal{K}, \mathcal{N}, E)$ and $\{\mathcal{K}_j\}_{j\in\mathcal{K}}$, if $\cup_{j\in\mathcal{K}}\mathcal{K}_j = \mathcal{N}$, then $\gamma(G) \leq \bar{\chi}(G)$.*

Proof Let $\mathcal{C} = \{C_1, C_2, ..., C_m\}$ be a clique cover with cardinality m, i.e., $|\mathcal{C}| = m$ and each C_k is a clique for $k = 1, ..., m$. Pick arbitrarily an element a_k from C_k for each k. Define $\mathcal{H} = \{a_k : k = 1, ..., m\}$. Now it remains to show that $\mathcal{H}$ is a hitting set, which implies $\gamma(G) \leq \bar{\chi}(G)$. We prove this by contradiction.

Suppose $\mathcal{H}$ is not a hitting set, then $\exists i \in \mathcal{N}$ s.t. $S_i \cap \mathcal{H} = \emptyset$. Since $\cup_{j\in\mathcal{K}}\mathcal{K}_j = \mathcal{N}$, $\exists j \in \mathcal{K}$ s.t. $i \in \mathcal{K}_j$. $\mathcal{C}$ is a clique cover, then $\exists k(j) \in \{1, 2, ..., m\}$ such that $j \in C_{k(j)}$. By the construction of $\mathcal{H}$, there exists $a_{k(j)} \in \mathcal{H} \cap C_{k(j)}$. By the definition of clique, we have $\mathcal{K}_j \subseteq V_{a_{k(j)}}$. Thus, we have $a_{k(j)} \in S_i$ since $i \in \mathcal{K}_j$. It follows that $S_i \cap \mathcal{H} \neq \emptyset$, which contradicts to $S_i \cap \mathcal{H} = \emptyset$. Hence, $\mathcal{H}$ is a hitting set. ■

In the next section, we obtain an asymptotic lower bound on the regret of uniformly good policies for the setting of MABs with side-observations. This lower bound is expressed as the optimal value of a linear program (LP), where the constraints of the LP capture the connectivity of each action in the network.

4. Regret Lower Bound in the Presence of Side Observations

In order to derive a lower bound on the regret, we need some mild regularity assumptions (Assumptions 1, 2, and 3) on the distributions F_i (associated with base-arm i) that are similar to the ones in Lai and Robbins (1985). Let the probability distribution F_i have a univariate density function $g(x; \theta_i)$ with unknown parameters θ_i, for each $i \in \mathcal{N}$. Let $D(\theta||\sigma)$ denote the Kullback Leibler (KL) distance between distributions with density functions $g(x; \theta)$ and $g(x; \sigma)$ and with means $u(\theta)$ and $u(\sigma)$ respectively.

Assumption 1 *(Finiteness) We assume that $g(\cdot;\cdot)$ is such that $0 < D(\theta||\sigma) < \infty$ whenever $u(\sigma) > u(\theta)$.*

Assumption 2 *(Continuity) For any $\epsilon > 0$ and θ, σ such that $u(\sigma) > u(\theta)$, there exists $\eta > 0$ for which $|D(\theta||\sigma) - D(\theta||\rho)| < \epsilon$ whenever $u(\sigma) < u(\rho) < u(\sigma) + \eta$.*

Assumption 3 *(Denseness) For each $i \in \mathcal{N}$, $\theta_i \in \Theta$ where the set Θ satisfies: for all $\theta \in \Theta$ and for all $\eta > 0$, there exists $\theta' \in \Theta$ such that $u(\theta) < u(\theta') < u(\theta) + \eta$.*

Let $\vec{\theta}$ be the vector $[\theta_1, \ldots, \theta_N]$. Define $\Theta_i = \{\vec{\theta} : \exists k \in S_i$ such that $\mu_k(\vec{\theta}) < \mu^*(\vec{\theta})\}$. So, not all actions that support base-arm i are optimal. Suppose $\vec{\theta} \in \Theta_i$. For base arm i, define the set

$$\mathcal{B}_i(\theta_i) = \{\theta_i' : \exists k \in S_i \text{ such that } \mu_k(\vec{\theta}_i') > \mu^*(\vec{\theta})\},$$

where $\vec{\theta}_i' = [\theta_1, \ldots, \theta_i', \ldots \theta_N]$. $\vec{\theta}_i'$ differs from $\vec{\theta}$ only in the i^{th} parameter. In this set $\mathcal{B}_i(\theta_i)$, base-arm i contributes towards a unique optimal action. Define constant $J_i(\theta_i) = \inf\{D(\theta_i||\theta_i') : \theta_i' \in \mathcal{B}_i(\theta_i)\}$. This is well-defined when $\mathcal{B}_i(\theta_i) \neq \emptyset$.

The following proposition is obtained using Theorem 2 in Lai and Robbins (1985). It provides an asymptotic lower bound on the regret of any uniformly good policy under the model described in Section 3:

Proposition 4 *Suppose Assumptions 1, 2, and 3 hold. Let $\mathcal{U} = \{j : \mu_j < \mu^*\}$ be the set of suboptimal actions. Also, let $\Delta_j = \mu^* - \mu_j$. Then, under any uniformly good policy $\boldsymbol{\phi}$, the expected regret is asymptotically bounded below as follows:*

$$\liminf_{t\to\infty} \frac{R_{\boldsymbol{\mu}}(t)}{\log(t)} \geq c_{\boldsymbol{\mu}}, \tag{1}$$

where $c_{\boldsymbol{\mu}}$ is the optimal value of the following linear program (LP) P_1:

$$\begin{aligned} P_1 :\ & \min \sum_{j\in\mathcal{U}} \Delta_j w_j, \\ \textit{subject to:}\ & \sum_{j\in S_i} w_j \geq \frac{1}{J_i(\theta_i)},\ \forall i \in \mathcal{N}, \\ & w_j \geq 0,\ \forall j \in \mathcal{K}. \end{aligned}$$

Proof *(Sketch)* Let $M_i(t)$ be the total number of observations corresponding to base-arm i available at time t. Then, by modifying the proof of Theorem 2 of Lai and Robbins (1985), we have that, for $i \in \mathcal{N}$,

$$\liminf_{t\to\infty} \frac{\mathbb{E}[M_i(t)]}{\log(t)} \geq \frac{1}{J_i(\theta_i)}.$$

An observation is received for base-arm i whenever any action in S_i is chosen. Hence, $M_i(t) = \sum_{j\in S_i} T_j(t)$. These two facts give us the constraints in LP P_1. See Appendix A for the full proof. ■

The linear program given in P_1 contains the graphical information that governs the lower bound. However, it requires the knowledge of $\vec{\theta}$, which is unknown. This motivates the construction of the following linear program, LP P_2, which preserves the graphical structure while eliminating the distributional dependence on $\vec{\theta}$.

$$\begin{aligned} P_2 : \min & \sum_{j\in\mathcal{K}} z_j \\ \text{subject to:}\ & \sum_{j\in S_i} z_j \geq 1,\ \forall i \in \mathcal{N}, \\ \text{and}\ & z_j \geq 0,\ \forall j \in \mathcal{K}. \end{aligned}$$

Let $\mathbf{z}^* = (z_j^*)_{j\in\mathcal{K}}$ be the optimal solution of LP P_2. In Sections 5 and 6, we use the above LP P_2 to modify the ϵ-greedy policy in Auer et al. (2002) and UCB policy in Auer and Ortner (2010) for the setting of side-observations. We provide regret guarantees of these modified policies in terms of the optimal value $\sum_{j\in\mathcal{K}} z_j^*$ of LP P_2. We note that the linear program P_2 is, in fact, the LP relaxation of the minimum hitting set problem on network G. Since, any hitting set of network G is a feasible solution to the LP P_2, we have that the optimal value of the LP $\sum_{j\in\mathcal{K}} z_j^* \leq \gamma(G) \leq \bar{\chi}(G)$.

Proposition 5 *Consider an Erdos-Renyi random bipartite graph $(\mathcal{K}, \mathcal{N}, E)$ such that each entry of the matrix E equals 1 with probability p, where $0 < p < 1$. Suppose $\cup_{j\in\mathcal{K}}\mathcal{K}_j = \mathcal{N}$, i.e., there are no useless base-arms in the network, then $\sum_{j\in\mathcal{K}} z_j^*$ is upper-bounded by $\log_{\frac{1}{1-p}} N$ as $N \to \infty$ in probability.*

Proof *(sketch)* Since $\sum_{j\in\mathcal{K}} z_j^* \leq \gamma(G)$, it remains to be shown that $\gamma(G)$ is upper bounded by the above result. Suppose there are no useless base-arms in the network. Then the set of all actions is a hitting set. Based on this observation, we construct a repeated experiment to generate actions sequentially. Then we define a stopping time τ as the first time that all the generated actions form a hitting set. Hence, we show the asymptotic result of τ as the upper bound of $\gamma(G)$. See full proof in Appendix B. ■

In the next proposition, we provide a lower bound on $c_{\boldsymbol{\mu}}$ in Equation (1) using the optimal solution $\mathbf{z}^* = (z_j^*)_{j\in\mathcal{K}}$ of LP P_2.

Proposition 6 *Let $\mathcal{U} = \{j : \mu_j < \mu^*\}$ be the set of suboptimal actions. Let $\mathcal{O} = \{j : \mu_j = \mu^*\}$ be the set of optimal actions. Then,*

$$\frac{\max_{i\in\mathcal{N}} J_i(\theta_i)}{\min_{j\in\mathcal{U}} \Delta_j} c_{\boldsymbol{\mu}} + |\mathcal{O}| \geq \sum_{j\in\mathcal{K}} z_j^* \geq \frac{\min_{i\in\mathcal{N}} J_i(\theta_i)}{\max_{j\in\mathcal{U}} \Delta_j} c_{\boldsymbol{\mu}}. \tag{2}$$

Proof *(Sketch)* Using the optimal solution of LP P_1, we construct a feasible solution satisfying constraints in LP P_2 for base-arms in $\mathcal{N}$. The feasible solution constructed in this way gives an upper bound on the optimal value of LP P_2 in terms of the optimal value of LP P_1. For the lower bound, we use the fact that any feasible solution of P_2, in particular $\mathbf{z}^*$, can be used to construct a feasible solution of P_1. See Appendix C for the full proof. ■

We note that $\sum_{j\in\mathcal{K}} z_j^* = \Theta(c_\mu)$ completely captures the time dependence of the regret on network structure under the following assumption:

Assumption 4 *The quantities $|\mathcal{O}|$, $\min_{j\in\mathcal{U}} \Delta_j$, and $\min_{i\in\mathcal{N}} J_i(\theta_i)$ are constants that are independent of network size K and N.*

Note that the constants in the above assumption are unknown to the decision maker. In the next section, we propose the ϵ_t-greedy-LP policy which achieves the regret lower bound of $c_{\boldsymbol{\mu}} \log(t)$ up to a multiplicative constant factor that is independent of the network structure and time.

5. Epsilon-greedy-LP policy

Motivated by the LPs P_1 and P_2, we propose a *network-aware* randomized policy called the ϵ_t-greedy-LP policy. We provide an upper bound on the regret of this policy and show that it achieves the asymptotic lower bound, up to a constant multiplier, independent of the network structure. Let $\bar{f}_j(t)$ be the empirical average of observations (rewards and side-observations combined) available for action j up to time t. The ϵ_t-greedy-LP policy is described in Algorithm 1. The policy consists of two iterations - exploitation and exploration, where the exploration probability decreases as $1/t$, similarly to that of the ϵ_t-greedy policy proposed by Auer et al. (2002). However, in our policy, we choose the exploration probability for action j to be proportional to z_j^*/t, where $\mathbf{z}^*$ is the optimal solution of LP P_2, while in the original policy in Auer et al. (2002), the exploration probability is uniform over all actions.

Algorithm 1 : ϵ_t-greedy-LP

Input: $c > 0$, $0 < d < 1$, optimal solution $\mathbf{z}^*$ of LP P_2.

for each time t **do**

Update $\bar{f}_j(t)$ for each $j \in \mathcal{K}$, where $\bar{f}_j(t)$ is the empirically average over all the observations of action j.

Let $\epsilon(t) = \min\left(1, \frac{c\sum_{j\in\mathcal{K}} z_j^*}{d^2 t}\right)$ and $a^* = \arg\max_{j\in\mathcal{K}} \bar{f}_j(t)$.

Sample a from the distribution such that $\mathbb{P}\{a = j\} = \frac{z_j^*}{\sum_{i\in\mathcal{K}} z_i^*}$ for all $j \in \mathcal{K}$.

Play action $\phi(t)$ such that

$$\phi(t) = \begin{cases} a, & \text{with probability } \epsilon(t) \\ a^*, & \text{with probability } 1-\epsilon(t) \end{cases} \tag{3}$$

end for

The following proposition provides performance guarantees on the ϵ_t-greedy-LP policy:

Proposition 7 *For $0 < d < \min_{j\in\mathcal{U}} \Delta_j$, any $c > 0$, and $\alpha > 1$, the probability with which a suboptimal action j is selected by the ϵ_t-greedy-LP policy, described in Algorithm 1, for all $t > t' = \frac{c\sum_{i\in\mathcal{K}} z_i^*}{d^2}$ is at most*

$$\left(\frac{c}{d^2 t} z_j^*\right) + \frac{2\lambda c\delta}{\alpha d^2}\left(\frac{et'}{t}\right)^{cr/\alpha d^2} \log\left(\frac{e^2 t}{t'}\right) + \frac{4}{\Delta_j^2}\left(\frac{et'}{t}\right)^{\frac{c\Delta_j^2}{2\alpha d^2}}, \tag{4}$$

where $r = \frac{3(\alpha-1)^2}{8\alpha-2}$, $\lambda = \max_{j\in\mathcal{K}} |\mathcal{K}_j|$, and $\delta = \max_{i\in\mathcal{N}} |S_i|$ is the maximum degree of the supports in the network. Note that α is a parameter we introduce in the proof, which is used to determine a range for the choice of parameter c as shown in Corollary 8.

Proof *(Sketch)* Since $\mathbf{z}^*$ satisfies the constraints in LP P_2, there is sufficient exploration within each suboptimal action's neighborhood. The proof is then a combination of this fact

and the proof of Theorem 3 in Auer et al. (2002). In particular, we derive an upper bound for the probability that suboptimal action j is played at each time and then sum over the time. See Appendix D for the full proof. ■

In the above proposition, for large enough c, we see that the second and third terms are $O(1/t^{1+\epsilon})$ for some $\epsilon > 0$ (Auer et al., 2002). Using this fact, the following corollary bounds the expected regret of the ϵ_t-greedy-LP policy:

Corollary 8 *Choose parameters c and d such that,*

$$0 < d < \min_{j \in \mathcal{U}} \Delta_j, \quad \text{and} \quad c > \max(2\alpha d^2/r, 4\alpha),$$

for any $\alpha > 1$. Then, the expected regret at time T of the ϵ_t-greedy-LP policy described in Algorithm 1 is at most

$$\left(\frac{c}{d^2} \sum_{j \in \mathcal{U}} \Delta_j z_j^* \right) \log(T) + O(K), \tag{5}$$

where the $O(K)$ term captures constants independent of time but dependent on the network structure. In particular, the $O(K)$ term is at most

$$\sum_{j \in \mathcal{U}} \left[\frac{\pi^2 \lambda c \delta \Delta_j}{3\alpha d^2} \left(et'\right)^{cr/\alpha d^2} + \frac{2\pi^2}{3\Delta_j} \left(et'\right)^{\frac{c\Delta_j^2}{2\alpha d^2}} \right],$$

where t', r, λ and δ are defined in Proposition 7.

Remark 9 *Under Assumption 4, we can see from Proposition 6 and Corollary 8 that, ϵ_t-greedy-LP algorithm is* order optimal *achieving the lower bound $\Omega\left(\sum_{j \in \mathcal{K}} z_j^* \log(T)\right) = \Omega\left(c_{\boldsymbol{\mu}} \log(T)\right)$ as the network and time scale.*

While the ϵ_t-greedy-LP policy is network aware, its exploration is oblivious to the observed average rewards of the sub-optimal actions. Further, its performance guarantees depend on the knowledge of $\min_{j \in \mathcal{U}} \Delta_j$, which is the difference between the best and the second best optimal actions. On the other hand, the UCB-LP policy proposed in the next section is network-aware taking into account the average rewards of suboptimal actions. This could lead to better performance compared to ϵ_t-greedy-LP policy in certain situations, for example, when the action with greater z_j^* is also highly suboptimal.

6. UCB-LP policy

In this section we develop the UCB-LP policy defined in Algorithm 2 and obtain upper bounds on its regret. The UCB-LP policy is based on the improved UCB policy proposed in Auer and Ortner (2010), which can be summarized as follows: the policy estimates the values of Δ_i in each round by a value $\tilde{\Delta}_m$ which is initialized to 1 and halved in each round m. By each round m, the policy draws $n(m)$ observations for each action in the set of

actions not eliminated by round m, where $n(m)$ is determined by $\tilde{\Delta}_m$. Then, it eliminates those actions whose UCB indices perform poorly. Our policy differs from the one in Auer and Ortner (2010) by accounting for the presence of side-observations - this is achieved by choosing each action according to the optimal solution of LP P_2, while ensuring that $n(m)$ observations are available for each action not eliminated by round m.

Algorithm 2 : UCB-LP policy

Input: Set of actions $\mathcal{K}$, time horizon T, and optimal solution $\mathbf{z}^*$ of LP P_2.

Initialization: Let $\tilde{\Delta}_0 := 1$, $A_0 := \mathcal{K}$, and $B_0 := \mathcal{K}$

for round $m = 0, 1, 2, \ldots, \lfloor \frac{1}{2} \log_2 \frac{T}{e} \rfloor$ **do**

Action Selection: Let $n(m) := \left\lceil \frac{2 \log(T \tilde{\Delta}_m^2)}{\tilde{\Delta}_m^2} \right\rceil$

If $|B_m| = 1$: choose the single action in B_m until time T.

Else If $\sum_{i \in \mathcal{K}} z_i^* \leq 2|B_m|\tilde{\Delta}_m : \forall j \in A_m$, choose action j $\left[z_j^*(n(m) - n(m-1)) \right]$ times.

Else For each action j in B_m, choose j for $[n(m) - n(m-1)]$ times.

Update $\bar{f}_j(m)$ and $T_j(m)$ for each $j \in \mathcal{K}$, where $\bar{f}_j(m)$ is the empirical average reward of action j, and $T_j(m)$ is the total number of observations for action j up to round m.

Action Elimination:
To get B_{m+1}, delete all actions j in B_m for which

$$\bar{f}_j(m) + \sqrt{\frac{\log(T\tilde{\Delta}_m^2)}{2T_j(m)}} < \max_{a \in B_m} \left\{ \bar{f}_a(m) - \sqrt{\frac{\log(T\tilde{\Delta}_m^2)}{2T_a(m)}} \right\},$$

Reset:
The set A_{m+1} is given as $A_{m+1} = \bigcup_{i \in D_{m+1}} S_i$, where $D_{m+1} = \bigcup_{j \in B_{m+1}} \mathcal{K}_j$.
Let $\tilde{\Delta}_{m+1} = \frac{\tilde{\Delta}_m}{2}$.

end for

The following proposition provides performance guarantees on the expected regret due to UCB-LP policy:

Proposition 10 *For action j, define round m_j as follows:*

$$m_j := \min \left\{ m : \tilde{\Delta}_m < \frac{\Delta_j}{2} \right\}.$$

Define $\bar{m} = \min\left\{m : \sum_{j\in\mathcal{K}} z_j^* > \sum_{j:m_j>m} 2^{-m+1}\right\}$ *and the set* $B = \{j \in \mathcal{U} : m_j > \bar{m}\}$. *Then, the expected regret due to the UCB-LP policy described in Algorithm 2 is at most*

$$\sum_{j\in\mathcal{U}\setminus B} \Delta_j z_j^* \frac{32\log(T\hat{\Delta}_j^2)}{\hat{\Delta}_j^2} + \sum_{j\in B} \frac{32\log(T\Delta_j^2)}{\Delta_j} + O(K^2), \tag{6}$$

where $\hat{\Delta}_j = \max\{2^{-\bar{m}+2}, \min_{a:j\in G_a}\{\Delta_a\}\}$, $G_a = \cup_{i\in\mathcal{K}_a} S_i$, *and* (z_j^*) *is the solution of LP* P_2. *The* $O(K^2)$ *term captures constants independent of time. Further, under Assumption 4, the regret is also at most*

$$O\left(\sum_{j\in\mathcal{K}} z_j^* \log(T)\right) + O(K^2), \tag{7}$$

where (z_j^*) *entirely captures the time dependence on network structure.*

Proof *(Sketch)* The $\log(T)$ term in the regret follows from the fact that, with high probability, each suboptimal action j is eliminated (from the set B_m) on or before the first round m such that $\tilde{\Delta}_m < \Delta_j/2$. See Appendix E for the full proof. ■

Remark 11 *While* ϵ_t*-greedy-LP does not require knowledge of the time horizon* T, *UCB-LP policy requires the knowledge of* T. *UCB-LP policy can be extended to the case of an unknown time horizon similar to the suggestion in Auer and Ortner (2010). Start with* $T_0 = 2$ *and at end of each* T_l, *set* $T_{l+1} = T_l^2$. *The regret bound for this case is shown in Proposition 19 in Appendix F.*

Next, we briefly describe the policies UCB-N and UCB-MaxN proposed in Caron et al. (2012). In the UCB-N policy, at each time, the action with the highest UCB index is chosen similar to UCB1 policy in Auer et al. (2002). In UCB-MaxN policy, at each time t, the action i with the highest UCB index is identified and its neighboring action j with the highest empirical average reward at time t is chosen.

Remark 12 *The regret upper bound of UCB-N policy is*

$$\inf_{\mathcal{C}} \sum_{C\in\mathcal{C}} \frac{8\max_{j\in C}\Delta_j}{\min_{j\in C}\Delta_j^2}\log(T) + O(K),$$

where $\mathcal{C}$ *is a clique covering of the sub-network of suboptimal actions. The regret upper bound for UCB-MaxN is the same as that for UCB-N with an* $O(|\mathcal{C}|)$ *term instead of the time-invariant* $O(K)$ *term. We show a better regret performance for UCB-LP policy and* ϵ_t*-greedy-LP policies with respect to the* $\log(T)$ *term because* $\sum_{i\in\mathcal{K}} z_i^* \le \gamma(G) \le \bar{\chi}(G)$. *However, the time-invariant term in our policies is* $O(K)$ *and* $O(K^2)$, *can be worse than the time-invariant term* $O(|\mathcal{C}|)$ *in UCB-MaxN.*

Remark 13 *All uniformly good policies that ignore side-observations incur a regret that is at least* $\Omega(|\mathcal{U}|\log(t))$ *Lai and Robbins (1985), where* $|\mathcal{U}|$ *is the number of suboptimal actions. This could be significantly higher than the guarantees on the regret of both* ϵ_t*-greedy-LP policy and UCB-LP policy for a rich network structure as discussed in Remark 12.*

Remark 14 *In our model, we assumed that the side observations are always available. However, in reality, side observations may only be obtained sporadically. Suppose that when action* j *is chosen, side-observations of base-arms* $i \in \mathcal{K}_j$ *are obtained almost surely and that of base-arms* $i \in V_j \setminus \mathcal{K}_j$ *are obtained with a known probability* p_j. *In this case, Proposition 4 holds with the replacement of LP* P_1 *with LP* P_1' *as follows:*

$$\begin{aligned} P_1':\ & \min \sum_{j\in\mathcal{U}} \Delta_j w_j, \\ \textit{subject to:}\ & \sum_{j\in S_i} (w_j \mathbb{1}_{\{i\in\mathcal{K}_j\}} + p_j w_j \mathbb{1}_{\{i\notin\mathcal{K}_j\}}) \ge \frac{1}{J_i(\theta_i)},\ \forall i \in \mathcal{N}, \\ & w_j \ge 0,\ \forall j \in \mathcal{K}. \end{aligned}$$

Both of our policies work for this setting by changing the LP P_2 *to* P_2' *as follows:*

$$\begin{aligned} P_2':\ & \min \sum_{j\in\mathcal{K}} z_j \\ \textit{subject to:}\ & \sum_{j\in S_i} (z_j \mathbb{1}_{\{i\in\mathcal{K}_j\}} + p_j z_j \mathbb{1}_{\{i\notin\mathcal{K}_j\}}) \ge 1,\ \forall i \in \mathcal{N}, \\ & \textit{and } z_j \ge 0,\ \forall j \in \mathcal{K}. \end{aligned}$$

The regret bounds of our policies will now depend on the optimal solution of LP P_2'.

7. Numerical Results

7.1 Algorithm Performance on Data Trace

We consider the Flixster network dataset for the numerical evaluation of our algorithms. The authors in Jamali and Ester (2010) collected this social network data, which contains about 1 million users and 14 million links. We use graph clustering by Dhillon et al. (2007) to identify two strongly clustered sub-networks of sizes 1000 and 2000 nodes. Both these sub-networks have a degree distribution that is a straight line on a log-log plot indicating a power law distribution commonly observed in social networks. [2]

Our empirical setup is as follows. Let $\mathcal{N}$ be the set of users and $\mathcal{K} = \mathcal{N}$. To be specific, each user in the network is offered a promotion at each time, and accepts the promotion with probability $\mu_i \in [0.3, 0.9]$. Let S_i be the set of one-hop neighbors in the social network of user i (including user i). This is the setting when the Flixster has a

2. We note that the social network of interest may or may not display a power law behavior. We find that the subgraphs of the Flixster network have a degree distribution that is a straight line on a log-log plot indicating a power law distribution display while the authors in Ugander et al. (2011) show that the degree distribution of the global Facebook network is not a straight line on log-log plot.

survey or a like/dislike indicator that generates side observations of user's neighborhood. Let $\mathcal{K}_j = \{j\}$ and $f_j(X_j) = X_j$, which means that the decision maker receives a random reward of 1 if the chosen user j accepts the promotion or 0 reward otherwise. μ_j is chosen uniformly at random from $[0.3, 0.8]$ and there are 50 randomly chosen users with optimal $\mu_j = 0.9$.

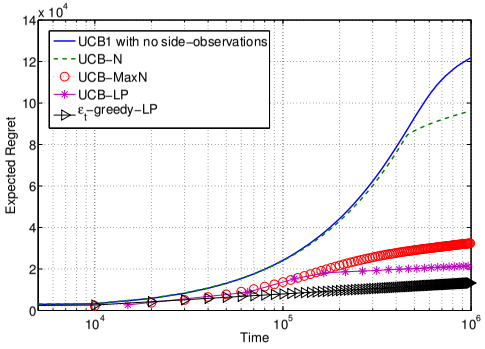


Figure 3: Regret comparison of all the policies for a network of size 1000.

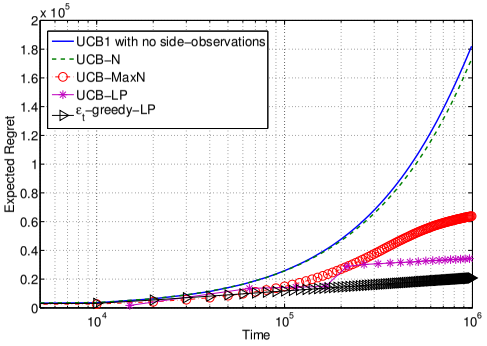


Figure 4: Regret comparison of all the policies for a network of size 2000.

Figures 3 and 4 show the regret performance as a function of time for the two sub-networks of sizes 1000 and 2000 respectively. Note that the average regret is taken over 1000 experiments. For the ϵ_t-greedy-LP policy, we let $c = 5$ and $d = 0.2$ (we choose $d = 0.2$ to show that our algorithm seems to have good performance in more general settings, even when the bounds in the Proposition 7 are not known or used). For both networks, we see that our policies outperform the UCB-N and UCB-MaxN policies Caron et al. (2012)

(UCB-N and UCB-MaxN policies can also be viewed as special cases of those proposed in Chen et al. (2013) for this specific combinatorial structure). We also observe that the improvement obtained by UCB-N policy over the baseline UCB1 policy is marginal. It has been shown (Cooper et al., 2005) that for power law graphs, both $\gamma(G)$ and $\bar{\chi}(G)$ scale linearly with N, although $\gamma(G)$ has a lower slope. Our numerical results show that our policies outperform existing policies even for the Flixster network.

As we show in Corollary 8 and Proposition 10, ϵ_t-greedy-LP and UCB-LP have the same upper bound $O(\sum_{j \in \mathcal{U}} z_j^* \log T)$. It is hard to say which algorithm outperforms the other one. In the Flixster network, we see that the ϵ_t-greedy-LP policy performs better than the UCB-LP policy. As we show in Section 7.2, UCB-LP performs better than ϵ_t-greedy-LP. In addition, the regret gap between the ϵ_t-greedy-LP and UCB-LP is not large compared to their gain to UCB-N and UCB-maxN.

7.2 Algorithm Performance on Synthetic Data

We consider a routing problem defined on a communication network, which is demonstrated as an undirected graph consisting of 6 nodes in Figure 5. We assume that node 1 is the source node and node 6 is the destination node. The decision maker repeatedly sends packets from the source node to the destination node. There exist 13 simple paths from the source node to the destination node. The delay of each path is the sum of delays over all the constituent links. The goal of the decision maker is to identify the path with the smallest expected delay and minimize the regret as much as possible.

Solving the routing problem is a direct application of this work once we let the set of paths be the set of actions and the set of links be the set of base-arms. We assume that the delay of each link i, denoted by $X_i(t)$, is an independent and identically distributed sequence of random variables (drawn from the steady-state distribution) over discrete time t. Then, this is a stochastic bandit problem with side-observations since playing one action (path) reveals some observations of some base-arms (links) that contribute to other actions (paths). For example, choosing path $(1, 2, 4, 6)$ reveals the delay performance of link $(1, 2)$ and $(2, 4)$, which are included in the path $(1, 2, 4, 5, 6)$.

In the routing problem, there are 13 paths and 12 directed links (note that some links are never used in all the paths). Thus, we set $K = 13$ and $N = 12$ in the simulation. Then, we construct the set V_j for each action j such that $i \in V_j$ if path j traverses the link i. And, we set $\mathcal{K}_j = V_j$ for each $j \in \mathcal{K}$ since the delay of a path is the total delay of the traversed links. Let B be the upper bound of all the action delays. Then, we choose the function $f_j(\vec{X}_j(t)) = 1 - \sum_{i \in \mathcal{K}_j} X_i(t)/B$ as the reward of playing action j at time t. In the simulation, we assume that the delay of link i is sampled from a Bernoulli distribution with mean u_i. Each u_i is independently sampled from a uniform distribution from 0 to 1, which realizes the problem instance in Figure 5[3]. One can check that the optimal action (shortest path) is the path $(1, 3, 5, 6)$ given the ground truth $\{u_i\}_{i \in \mathcal{N}}$. We let $B = 5$ in the simulation.

We apply the UCB1, UCB-N, Cohen (Cohen et al., 2016) and our policies to the problem instance in Figure 5 and the regret performance, averaged over 1000 experiments, is shown in Figure 6. We do not represent the result of the UCB-MaxN because it degenerates to

3. The number indicates the mean delay and the arrow indicates the direction of the link

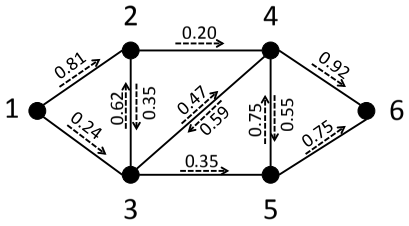


Figure 5: Routing problem on a communication network

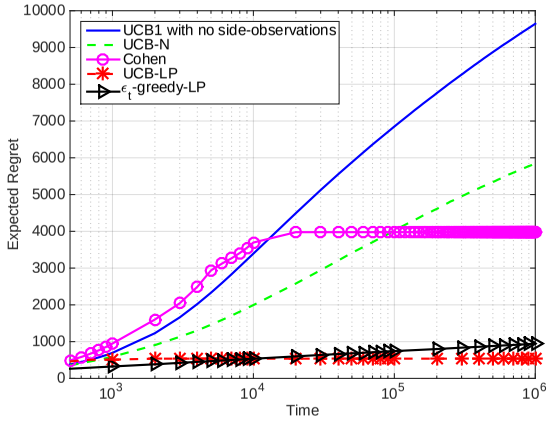


Figure 6: Regret comparison of all the policies for the routing problem

the UCB-N policy. The reason is that there is no non-trivial clique (clique with more than one element) in this problem due to the setting $\mathcal{K}_j = V_j$ and $\mathcal{K}_j \neq \mathcal{K}_a$ for any $j, a \in \mathcal{K}$. Intuitively, there does not exist two paths that can observe the outcome of each other. For the ϵ_t-greedy-LP policy, we let $c = 4$ and $d = 0.05$. From the regret performance, we observe that the improvement obtained by the UCB-N policy against the UCB1 policy is considerably greater than the results in Figure 3 and Figure 4. The reason behind this is that the bipartite graph in the routing problem is more dense and network size is small in the routing problem, which enables the UCB-N policy to take the advantage of side-observations. Overall, we see that our policies outperform the UCB-N policy and Cohen policy because our policies take the network structure into consideration, which enables us to trade off the exploration with exploitation more efficiently.

7.3 Asymptotic Behavior

We run a simulation to verify the result provided in Proposition 5. For each base-arm size N, we sequentially generate action j such that $e_{ij} = 1$ with probability p for any $i \in \mathcal{N}$ independently. Stopping time τ is the number of actions we have generated so that there are no useless base-arms in the network. Note that τ is an upper bound of $\gamma(G)$ and $\sum_{j\in\mathcal{K}} z_j^*$ as shown in Appendix B. Then, we solve the linear program P2 to obtain $\sum_{j\in\mathcal{K}} z_j^*$ and find a hitting set by a greedy algorithm.[4]

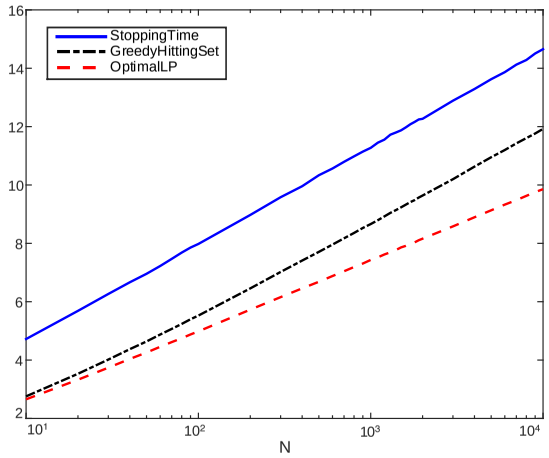


Figure 7: Erdos-Renyi random graph with p=0.5

Figure 7 shows the average result over 1000 samples for each N when $p = 0.5$. The numerical result verifies our theoretical result that $\sum_{j\in\mathcal{K}} z_j^*$ is upper-bounded by a logarithmic function of N asymptotically in Erdos-Renyi random graph. The reason why we are interested in the scaling order of $\sum_{j\in\mathcal{K}} z_j^*$ is that the traditional UCB1 policy suffers from the curse of dimensionality when applied in the real world, such as recommendation systems with thousands of items. However, our policies show a regret of $O(\sum_{j\in\mathcal{K}} z_j^* \log T)$, and $\sum_{j\in\mathcal{K}} z_j^*$ is upper-bounded by a logarithmic function of the number of unknowns, which makes our policies scalable in some large networks.

8. Summary

In this work, we introduce an important structural form of feedback available in many multiarmed bandits using the bipartite network structure. We obtained an asymptotic (with respect to time) lower bound as a function of the network structure on the regret of

4. It is well known that hitting set problem is NP-complete. So we employ the greedy algorithm which brings in the node with the largest degree in the network during each iteration.

any uniformly good policy. Further, we proposed two policies: 1) the ϵ_t-greedy-LP policy, and 2) the UCB-LP policy, both of which are optimal in the sense that they achieve the asymptotic lower bound on the regret, up to a multiplicative constant that is independent of the network structure. These policies can have a better regret performance than existing policies for some important network structures. The ϵ_t-greedy-LP policy is a network-aware any-time policy, but its exploration is oblivious to the average rewards of the suboptimal actions. On the other hand, UCB-LP considers both the network structure and the average rewards of actions.

Important avenues of future work include the case of dynamic graphs – what would be the lower bound and corresponding algorithms if the graph structure remains known but changes with time? Recently Tossou et al. (2017) presented a novel extension of Thompson sampling algorithm for the setting of immediate neighbor feedback studied in Mannor and Shamir (2011); Caron et al. (2012); Buccapatnam et al. (2014). It would be interesting to see how to adapt Thompson sampling algorithm for the bipartite graph feedback structure introduced in our current work.

In what follows, we give the proofs of all propositions stated in the earlier sections. These proofs make use of Lemmas 15, 16, and 17, and Proposition 18 given in Section F.

Appendix A. Proof of Proposition 4

Let $\mathcal{U} = \{j : \mu_j < \mu^*\}$ be the set of suboptimal actions. Also, let $\Delta_j = \mu^* - \mu_j$. Also, $T_j(t)$ is the total number of times action j is chosen up to time t by policy $\boldsymbol{\phi}$. Let $M_i(t)$ be the total number of observations corresponding to base-arm i available at time t. From Proposition 18 given in the Appendix, we have,

$$\liminf_{t\to\infty} \frac{\mathbb{E}[M_i(t)]}{\log(t)} \geq \frac{1}{J_i(\theta_i)}, \ \forall i \in \mathcal{N}. \tag{8}$$

An observation is received for base-arm i whenever any action in S_i is chosen. Hence,

$$M_i(t) = \sum_{j\in S_i} T_j(t). \tag{9}$$

Now, from Equations (8) and (9), for each $i \in \mathcal{N}$,

$$\liminf_{t\to\infty} \frac{\sum_{j\in S_i} \mathbb{E}[T_j(t)]}{\log(t)} \geq \frac{1}{J_i(\theta_i)}. \tag{10}$$

Using (10), we get the constraints of LP P_1. Further, we have from definition of regret that,

$$\liminf_{t\to\infty} \frac{R_\mu(t)}{\log(t)} = \liminf_{t\to\infty} \sum_{j\in\mathcal{U}} \Delta_j \frac{\mathbb{E}[T_j(t)]}{\log(t)}.$$

The above equation along with the constraints of the LP P_1 obtained from (10) gives us the required lower bound on regret.

Appendix B. Proof of Proposition 5

Here we consider a E-R random graph with each entry of the matrix E equals 1 with probability p, where $0 < p < 1$. Consider the following discrete stochastic process. χ_n are i.i.d., such that $\chi_n \subseteq [N]$ is sampled by the following steps: for each $i = 1, 2, .., N$, $i \in \chi_n$ with probability p. Let $q = 1 - p$. Then let τ be a stopping time defined as

$$\tau = \min\{n \geq 1, \cup_{j=1}^{n} \chi_j = [N]\} \tag{11}$$

The complement cdf of τ is the following.

$$P(\tau > n) = 1 - (1 - q^n)^N \tag{12}$$

1. Fix N. Given $0 < p < 1$, then $0 < q < 1$. Thus, $P(\tau = \infty) = 0$
2. What is the upper bound of $E(\tau)$?

$$(1 - q^n)^N > \exp(-\frac{q^n N}{1 - q^n}) \ \ (since \ \ln(1 - x) > -\frac{x}{1 - x} \ \ for \ \ 0 < x < 1) \tag{13}$$

Thus, we have

$$P(\tau > n) < 1 - \exp(-\frac{q^n N}{1-q^n}) \leq \frac{q^n N}{1-q^n} \quad (since\ 1 - e^x \leq -x) \tag{14}$$

Then we can bound the expectation of τ.

$$E(\tau) = \sum_{n=1}^{\infty} P(\tau > n) < \sum_{n=1}^{\infty} q^n \frac{N}{1-q} = \frac{qN}{(1-q)^2} \tag{15}$$

3. What is the upper bound of $P(\tau \leq n)$?

$$P(\tau \leq n) = (1-q^n)^N \leq \exp(-q^n N) \tag{16}$$

4. Does τ converge as N goes to infinity?
Given $\epsilon > 0$, as N goes to ∞,

$$P(\tau \leq (1-\epsilon)\log_{1/q} N) \leq \exp(-q^{(1-\epsilon)\log_{1/q} N} N) = \exp(-N^{\epsilon}) \to 0 \tag{17}$$

$$P(\tau > (1+\epsilon)\log_{1/q} N) \leq \frac{q^{(1+\epsilon)\log_{1/q} N} N}{1-q} = \frac{1}{(1-q)N^{\epsilon}} \to 0 \tag{18}$$

Since ϵ is arbitrary, we can have

$$P(\tau = \log_{1/q} N) \to 1 \ \ as\ N \to \infty. \tag{19}$$

That is to say τ converges to $\log_{1/q} N$ in probability.

Suppose there are no useless base-arms in the network, i.e. $[K]$ is a hitting set. Then τ is less than K with probability 1. Given this information, $\gamma(G)$ should be upper bounded by $\log_{1/q} N$ as N goes to infinity.

Appendix C. Proof of Proposition 6

Let $(z_j^*)_{j\in\mathcal{K}}$ be the optimal solution of LP P_2. We will first prove the upper bound in Equation 2. Using the optimal solution $(w_j^*)_{j\in\mathcal{K}}$ of LP P_1, we construct a feasible solution satisfying constraints in LP P_2 in the following way: For actions $j \in \mathcal{K}$, let $z_j = \left(\max_{i\in\mathcal{N}} J_i(\theta_i)\right) w_j^*$. Then $(z_j)_{j\in\mathcal{K}}$ satisfy constraints for all base-arms $i \in \mathcal{N}$ because w_j^* satisfy constraints of LP P_1.
The feasible solution constructed in this way gives an upper bound on the optimal value of LP P_2. Hence,

$$\begin{aligned}
\sum_{j\in\mathcal{K}} z_j^* &\leq \sum_{j\in\mathcal{U}} z_j + |\mathcal{O}| \\
&\leq \sum_{j\in\mathcal{U}} \left(\max_{i\in\mathcal{N}} J_i(\theta_i)\right) w_j^* + |\mathcal{O}| \\
&\leq \frac{\max_{i\in\mathcal{N}} J_i(\theta_i)}{\min_{j\in\mathcal{U}} \Delta_j} \sum_{j\in\mathcal{U}} \Delta_j w_j^* + |\mathcal{O}| \\
&\leq \frac{\max_{i\in\mathcal{N}} J_i(\theta_i)}{\min_{j\in\mathcal{U}} \Delta_j} c_{\boldsymbol{\mu}} + |\mathcal{O}|
\end{aligned}$$

For the lower bound, any feasible solution of P_2, in particular $\mathbf{z}^*$, can be used to construct a feasible solution of P_1. For actions $j \in \mathcal{K}$, let $w_j = \frac{z_j^*}{\min_{i\in\mathcal{N}} J_i(\theta_i)}$. Then $(w_j)_{j\in\mathcal{K}}$ satisfies the constraints of LP P_1 and hence gives an upper bound on its optimal value. Therefore, we have

$$\begin{aligned}
c_{\boldsymbol{\mu}} &= \sum_{j\in\mathcal{U}} \Delta_j w_j^*, \\
&\leq \sum_{j\in\mathcal{K}} \frac{\Delta_j z_j^*}{\min_{i\in\mathcal{N}} J_i(\theta_i)} \\
&\leq \sum_{j\in\mathcal{K}} \frac{\max_{a\in\mathcal{U}} \Delta_a z_j^*}{\min_{i\in\mathcal{N}} J_i(\theta_i)}
\end{aligned}$$

which gives us the required lower bound.

Appendix D. Proof of Proposition 7

Since $\mathbf{z}^*$ satisfies the constraints in LP P_2, there is sufficient exploration within each suboptimal action's neighborhood. The proof is then a combination of this fact and the proof of Theorem 3 in Auer et al. (2002). Let $\bar{f}_j(t)$ be the random variable denoting the sample mean of all observations available for action j at time t. Let $\bar{f}^*(t)$ be the random variable denoting the sample mean of all observations available for an optimal action at time t. Fix a suboptimal action j. For some $\alpha > 1$, define m_i for each base-arm i as follows,

$$m_i = \frac{1}{\alpha} \frac{\sum_{j\in S_i} z_j^*}{\sum_{j\in\mathcal{K}} z_j^*} \sum_{m=1}^{t} \epsilon(m)$$

Let $\phi(t)$ be the action chosen by ϵ_t-greedy-LP policy at time t. The event $\{\phi(t) = j\}$ implies that either sampling a random action j for exploration or playing the best observed action j for exploitation. Then,

$$\mathbb{P}[\phi(t) = j] \leq \frac{\epsilon(t) z_j^*}{\sum_{a\in\mathcal{K}} z_a^*} + (1 - \epsilon(t))\mathbb{P}[\bar{f}_j(t) \geq \bar{f}^*(t)]$$

The event $\{\bar{f}_j(t) \geq \bar{f}^*(t)\}$ implies that either $\{\bar{f}_j(t) \geq \mu_j + \frac{\Delta_j}{2}\}$ or $\{\bar{f}^*(t) \leq \mu^* - \frac{\Delta_j}{2}\}$ since $\mu_j + \frac{\Delta_j}{2} = \mu^* - \frac{\Delta_j}{2}$. We also have that,

$$\mathbb{P}[\bar{f}_j(t) \geq \bar{f}^*(t)] \leq \mathbb{P}\left[\bar{f}_j(t) \geq \mu_j + \frac{\Delta_j}{2}\right] + \mathbb{P}\left[\bar{f}^*(t) \leq \mu^* - \frac{\Delta_j}{2}\right].$$

The analysis of both the terms in the right hand side of the above expression is similar. Let $O_i^R(t)$ be the total number of observations available for base-arm i from the exploration iterations of the policy up to time t. Let $O_i(t)$ be the total number of observations available for base-arm i up to time t. By concentration inequalities, the probability that the empirical mean deviate from the expectation can be bounded given the number of observations. The

number of observations for action j is lower-bounded by the number of observations from the exploration iterations. Hence, we have,

$$\begin{aligned}
\mathbb{P}\left[\bar{f}_j(t) \geq \mu_j + \frac{\Delta_j}{2}\right] &= \sum_{m=1}^{t} \mathbb{P}\left[\min_{i\in\mathcal{K}_j} O_i(t) = m; \bar{f}_j(t) \geq \mu_j + \frac{\Delta_j}{2}\right] \\
&= \sum_{m=1}^{t} \mathbb{P}\left[\bar{f}_j(t) \geq \mu_j + \frac{\Delta_j}{2} | \min_{i\in\mathcal{K}_j} O_i(t) = m\right] \mathbb{P}\left[\min_{i\in\mathcal{K}_j} O_i(t) = m\right] \\
&\leq \sum_{m=1}^{t} \mathbb{P}\left[\min_{i\in\mathcal{K}_j} O_i(t) = m\right] e^{\frac{-\Delta_j^2 m}{2}} \\
&\text{(follows from Chernoff-Hoeffding bound in Lemma 15)} \\
&\leq \sum_{m=1}^{t} \mathbb{P}\left[\min_{i\in\mathcal{K}_j} O_i^R(t) \leq m\right] e^{\frac{-\Delta_j^2 m}{2}} \\
&\leq \sum_{m=1}^{\lfloor m_0 \rfloor} \mathbb{P}\left[\min_{i\in\mathcal{K}_j} O_i^R(t) \leq m\right] + \sum_{m=\lfloor m_0 \rfloor+1}^{t} e^{\frac{-\Delta_j^2 m}{2}} \\
&\leq m_0 \mathbb{P}\left[\min_{i\in\mathcal{K}_j} O_i^R(t) \leq m_0\right] + \frac{2}{\Delta_j^2} e^{\frac{-\Delta_j^2 m_0}{2}} \\
&\qquad \left(\text{since } \sum_{m+1}^{\infty} e^{-ku} \leq \frac{1}{k} e^{-km}\right) \\
&\leq \sum_{i\in\mathcal{K}_j} m_0 \mathbb{P}\left[O_i^R(t) \leq m_0\right] + \frac{2}{\Delta_j^2} e^{\frac{-\Delta_j^2 m_0}{2}},
\end{aligned}$$

where $m_0 = \min_{i\in\mathcal{N}} m_i$.

Recall that $O_i^R(t)$ is the total number of observations for base-arm i from exploration. Now, we derive the bounds for the expectation and variance of $O_i^R(t)$ in order to use Bernstein's inequality.

$$\begin{aligned}
\mathbb{E}\left[O_i^R(t)\right] &= \sum_{m=1}^{t} \epsilon(m) \sum_{j\in S_i} \frac{z_j^*}{\sum_{j\in\mathcal{K}} z_j^*} \\
&= \frac{\sum_{j\in S_i} z_j^*}{\sum_{j\in\mathcal{K}} z_j^*} \sum_{m=1}^{t} \epsilon(m) = \alpha m_i \\
&\geq \alpha m_0
\end{aligned}$$

$$\begin{aligned}
var\left[O_i^R(t)\right] &= \sum_{m=1}^{t} \left(\epsilon(m) \sum_{j\in S_i} \frac{z_j^*}{\sum_{j\in\mathcal{K}} z_j^*}\right) \left(1 - \epsilon(m) \sum_{j\in S_i} \frac{z_j^*}{\sum_{j\in\mathcal{K}} z_j^*}\right) \\
&\leq \sum_{m=1}^{t} \epsilon(m) \sum_{j\in S_i} \frac{z_j^*}{\sum_{j\in\mathcal{K}} z_j^*} \\
&= \mathbb{E}[O_i^R(t)] = \alpha m_i
\end{aligned}$$

Now, using Bernstein's inequality given in Lemma 16, we have

$$\begin{aligned}
\mathbb{P}\left[O_i^R(t) \leq m_0\right] &= \mathbb{P}\left[O_i^R(t) \leq \mathbb{E}[O_i^R(t)] + m_0 - \alpha m_i\right] \\
&\leq \mathbb{P}\left[O_i^R(t) \leq \mathbb{E}[O_i^R(t)] + m_i - \alpha m_i\right] \\
&\leq \exp\left(-\frac{(\alpha-1)^2 m_i^2}{2\alpha m_i + \frac{2}{3}(\alpha-1)m_i}\right) \\
&= \exp\left(-\frac{3(\alpha-1)^2}{8\alpha-2} m_i\right) = \exp(-r m_i)
\end{aligned}$$

where $r = \frac{3(\alpha-1)^2}{8\alpha-2}$. Now, we will obtain upper and lower bounds on m_i by plugging in the definition of $\epsilon(m)$. For the upper bound, for any $t > t' = \frac{c\sum_{i\in\mathcal{K}} z_i^*}{d^2}$,

$$\begin{aligned}
m_i &= \frac{\sum_{j\in S_i} z_j^*}{\alpha \sum_{j\in\mathcal{K}} z_j^*} \sum_{m=1}^{t} \epsilon(m) \\
&= \frac{\sum_{j\in S_i} z_j^*}{\alpha \sum_{j\in\mathcal{K}} z_j^*} t' + \frac{\sum_{j\in S_i} z_j^*}{\alpha \sum_{j\in\mathcal{K}} z_j^*} \sum_{m=t'+1}^{t} \frac{c\sum_{i\in\mathcal{K}} z_i^*}{d^2 m} \\
&\leq \frac{c\delta}{\alpha d^2} \left(1 + \sum_{m=t'+1}^{t} \frac{1}{m}\right) \\
&\leq \frac{c\delta}{\alpha d^2} \log\left(\frac{e^2 t}{t'}\right).
\end{aligned}$$

where $\delta = \max_{i\in\mathcal{N}} |S_i|$, denoting the maximum degree of the supports in the network. In the above, $\sum_{j\in S_i} z_j^* \leq \delta$ because $z_j^* \leq 1$, which is due to the fact that $\left(z_j^*\right)_{j\in\mathcal{K}}$ is the optimal solution of LP P_2. Next, for the lower bound, we use the fact that $\sum_{j\in S_i} z_j^* \geq 1$ for all i because $\left(z_j^*\right)_{j\in\mathcal{K}}$ satisfies the constraints of LP P_2. Thus

$$\begin{aligned}
m_i &\geq \frac{\sum_{j\in S_i} z_j^*}{\alpha \sum_{j\in\mathcal{K}} z_j^*} \sum_{m=t'+1}^{t} \frac{c\sum_{i\in\mathcal{K}} z_i^*}{d^2 m} \\
&\geq \frac{c}{\alpha d^2} \sum_{m=t'+1}^{t} \frac{1}{m} \\
&\geq \frac{c}{\alpha d^2} \log\left(\frac{t}{et'}\right).
\end{aligned}$$

Let $\lambda = \max_{j\in\mathcal{K}} |\mathcal{K}_j|$. Hence, combining the inequalities above,

$$\begin{aligned}
\mathbb{P}\left[\bar{f}_j(t) \geq \mu_j + \frac{\Delta_j}{2}\right] &\leq \sum_{i\in\mathcal{K}_j} m_0 \mathbb{P}\left[O_i^R(t) \leq m_0\right] + \frac{2}{\Delta_j^2} e^{-\frac{\Delta_j^2 m_0}{2}} \\
&\leq \sum_{i\in\mathcal{K}_j} m_0 \left(\frac{et'}{t}\right)^{cr/\alpha d^2} + \frac{2}{\Delta_j^2} e^{-\frac{\Delta_j^2 m_0}{2}} \\
&\leq \lambda \frac{c\delta}{\alpha d^2}\left(\log\left(\frac{e^2 t}{t'}\right)\right)\left(\frac{et'}{t}\right)^{cr/\alpha d^2} + \frac{2}{\Delta_j^2}\left(\frac{et'}{t}\right)^{\frac{c\Delta_j^2}{2\alpha d^2}}
\end{aligned}$$

Now, similarly for the optimal action, we have, for all $t > t'$

$$\mathbb{P}\left[\bar{f}^*(t) \leq \mu^* - \frac{\Delta_j}{2}\right] \leq \frac{\lambda c\delta}{\alpha d^2}\left(\frac{et'}{t}\right)^{cr/\alpha d^2} \log\left(\frac{e^2 t}{t'}\right) + \frac{2}{\Delta_j^2}\left(\frac{et'}{t}\right)^{\frac{c\Delta_j^2}{2\alpha d^2}}.$$

Combining everything, we have for any suboptimal action j, for all $t > t'$

$$\begin{aligned}
\mathbb{P}[\phi(t) = j] &\leq \frac{\epsilon(t) z_j^*}{\sum_{a\in\mathcal{K}} z_a^*} + (1-\epsilon(t))\, P[\bar{f}_j(t) \geq \bar{f}^*(t)] \\
&\leq \frac{c z_j^*}{d^2 t} + P[\bar{f}_j(t) \geq \bar{f}^*(t)] \\
&\leq \frac{c z_j^*}{d^2 t} + \frac{2\lambda c\delta}{\alpha d^2}\left(\frac{et'}{t}\right)^{cr/\alpha d^2} \log\left(\frac{e^2 t}{t'}\right) + \frac{4}{\Delta_j^2}\left(\frac{et'}{t}\right)^{\frac{c\Delta_j^2}{2\alpha d^2}}
\end{aligned}$$

Appendix E. Proof of Proposition 10

The proof technique is similar to that in Auer and Ortner (2010). We will analyze the regret by conditioning on two disjoint events. The first event is that each suboptimal action a is eliminated by an optimal action on or before the first round m such that $\tilde{\Delta}_m < \Delta_a/2$. This happens with high probability and leads to logarithmic regret. The compliment of the first event yields linear regret in time but occurs with probability proportional to $1/T$. The main difference from the proof in Auer and Ortner (2010) is that on the first event, the number of times we choose each action j is proportional to $z_j^* \log(T)$ in the exploration iterations (i.e., when $|B_m| > 1$) of the policy. This gives us the required upper bound in terms of optimal solution $\mathbf{z}^*$ of LP P_2.

Let $*$ denote any optimal action. Let m^* denote the round in which the last optimal action $*$ is eliminated. For each suboptimal action j, define round $m_j := \min\{m : \tilde{\Delta}_m < \frac{\Delta_j}{2}\}$. For an optimal action j, $m_j = \infty$ by convention. Then, by the definition of m_j, for all rounds $m < m_j$, $\Delta_j \leq 2\tilde{\Delta}_m$, and

$$\frac{2}{\Delta_j} < 2^{m_j} = \frac{1}{\tilde{\Delta}_{m_j}} \leq \frac{4}{\Delta_j} < \frac{1}{\tilde{\Delta}_{m_j+1}} = 2^{m_j+1}. \tag{20}$$

From Lemma 17 in the Appendix, the probability that action j is not eliminated in round m_j by $*$ is at most $\frac{2}{T\tilde{\Delta}_{m_j}^2}$.

Let $I(t)$ be the action chosen at time t by the UCB-LP policy.
Let E_{m^*} be the event that all suboptimal actions with $m_j \leq m^*$ are eliminated by $*$ on or before their respective m_j. Then, the complement of E_{m^*}, denoted as $E_{m^*}^c$, is the event that there exists some suboptimal action j with $m_j \leq m^*$, which is not eliminated by round m_j. Let E_j^c be the event that action j is not eliminated by round m_j by $*$. Let $m_f = \lfloor \frac{1}{2}\log_2 \frac{T}{e} \rfloor$ and $I(t)$ denote the action chosen at time t by the policy. Recall that regret is denoted by $R_{\boldsymbol{\mu}}(T)$. Let $\mathbb{P}[m^* = m]$ be denoted by p_m. Hence, $\sum_{m=0}^{m_f} p_m = 1$.

$$\begin{aligned}
\mathbb{E}\left[R_{\boldsymbol{\mu}}(T)\right] &= \sum_{m=0}^{m_f} \mathbb{E}\left[R_{\boldsymbol{\mu}}(T) | \{m^* = m\}\right] \mathbb{P}[m^* = m] \\
&= \sum_{m=0}^{m_f}\sum_{t=1}^{T}\sum_{j\in\mathcal{U}} \Delta_j \mathbb{P}\left[I(t) = j | \{m^* = m\}\right] p_m \\
&= \sum_{m=0}^{m_f}\sum_{t=1}^{T}\sum_{j\in\mathcal{U}} \Delta_j \mathbb{P}\left[\{I(t) = j\} \cap E_{m^*} | \{m^* = m\}\right] p_m \\
&\quad + \sum_{m=0}^{m_f}\sum_{t=1}^{T}\sum_{j\in\mathcal{U}} \Delta_j \mathbb{P}\left[\{I(t) = j\} \cap E_{m^*}^c | \{m^* = m\}\right] p_m \\
&= (i) + (ii)
\end{aligned}$$

Next we will show that term (i) leads to logarithmic regret while term (ii) leads to a constant regret with time.
First, consider the term (ii) of the regret expression. For each $j \in \mathcal{U}$, we have,

$$\begin{aligned}
&\sum_{m=0}^{m_f}\sum_{t=1}^{T} \mathbb{P}\left[\{I(t) = j\} \cap E_{m^*}^c | \{m^* = m\}\right] \mathbb{P}[m^* = m] \\
&\leq \sum_{m=0}^{m_f}\sum_{t=1}^{T} \mathbb{P}\left[\{I(t) = j\} \cap \left(\cup_{a\in\mathcal{U}: m_a \leq m^*} E_a^c\right) | \{m^* = m\}\right] p_m \\
&\leq \sum_{m=0}^{m_f}\sum_{t=1}^{T} \Big(\mathbb{P}\left[\{I(t) = j\} | \left(\cup_{a\in\mathcal{U}: m_a \leq m^*} E_a^c\right), \{m^* = m\}\right] \\
&\qquad\qquad \mathbb{P}\left[\cup_{a\in\mathcal{U}: m_a \leq m^*} E_a^c | \{m^* = m\}\right] p_m\Big) \\
&\leq T\mathbb{P}\left[\cup_{a\in\mathcal{U}} E_a^c | \{m^* = m_f\}\right] \sum_{m=0}^{m_f} p_m \\
&\leq T \sum_{a\in\mathcal{U}} \frac{2}{T\tilde{\Delta}_{m_a}^2}, \\
&\left(\text{using Lemma 17, } \mathbb{P}\left[E_a^c | \{m^* = m_f\}\right] \leq \frac{2}{T\tilde{\Delta}_{m_a}^2}\right) \\
&\leq \sum_{a\in\mathcal{U}} \frac{32}{\Delta_a^2},
\end{aligned}$$

where the last inequality follows from Equation (20). Hence, the term (ii) of regret is

$$\sum_{m=0}^{m_f}\sum_{t=1}^{T}\sum_{j\in\mathcal{U}}\Delta_j\mathbb{P}\left[\{I(t)=j\}\cap E^c_{m^*}|\{m^*=m\}\right]p_m \leq \sum_{j\in\mathcal{U}}\Delta_j\sum_{a\in\mathcal{U}}\frac{32}{\Delta_a^2}=O(K^2). \tag{21}$$

Next, we consider the term (i). Recall that, in this term, we consider the case that all suboptimal actions j with $m_j \leq m^*$ are eliminated by $*$ on or before m_j.

$$\begin{aligned}
(i) &= \sum_{m=0}^{m_f}\sum_{t=1}^{T}\sum_{j\in\mathcal{U}}\Delta_j\mathbb{P}\left[\{I(t)=j\}\cap E_{m^*}|\{m^*=m\}\right]p_m\\
&= \sum_{m=0}^{m_f}\mathbb{E}\left[R_{\boldsymbol{\mu}}(T)|\{m^*=m\},E_{m^*}\right]\mathbb{P}[E_{m^*}|\{m^*=m\}]p_m\\
&\leq \sum_{m=0}^{m_f}\Big(\mathbb{E}\left[\text{Regret from }\{j:m_j\leq m^*\}|\{m^*=m\},E_{m^*}\right]\\
&\quad+\mathbb{E}\left[\text{Regret from }\{j:m_j>m^*\}|\{m^*=m\},E_{m^*}\right]\Big)p_m\\
&\leq \sum_{m=0}^{m_f}\Big(\mathbb{E}\left[\text{Regret from }\{j:m_j\leq m_f\}|\{m^*=m_f\},E_{m_f}\right]\\
&\quad+\mathbb{E}\left[\text{Regret from }\{j:m_j>m^*\}|\{m^*=m\},E_{m^*}\right]\Big)p_m\\
&\leq \mathbb{E}\left[R_{\boldsymbol{\mu}}(T)|\{m^*=m_f\},E_{m_f}\right]\sum_{m=0}^{m_f}p_m\\
&+\sum_{m=0}^{m_f}\mathbb{E}\left[\text{Regret from }\{j:m_j>m^*\}|\{m^*=m\},E_{m^*}\right]p_m\\
&= (ia)+(ib)
\end{aligned}$$

Once again, we will consider the above two terms separately. For the term (ia), under the event E_{m_f}, each suboptimal action j is eliminated by $*$ by round m_j. Define round $\bar{m}$ and the set B as follows:

$$\begin{aligned}
\bar{m} &= \min\{m:\sum_{j\in\mathcal{K}}z_j^*>\sum_{a:m_a>m}2^{-m+1}\},\\
B &= \{j\in\mathcal{U}:m_j>\bar{m}\}.
\end{aligned}$$

After round $\bar{m}$, Algorithm 2 chooses only those actions with $m_j>\bar{m}$. Also, by the definition of the **Reset** phase of Algorithm 2, we have that any suboptimal action $j\notin B$ is chosen (i.e. appears in the set A_m at round m) only until it is not in A_m or until $\bar{m}$, whichever happens

first. Define $n_j=\min\{\bar{m},\max_{a:j\in G_a}\{m_a\}\}$ for each suboptimal action j, where $G_a=\bigcup_{i\in\mathcal{K}_a}S_i$ for action a. Then any suboptimal action $j\notin B$ is chosen for at most n_j rounds.

$$\begin{aligned}
(ia) &= \mathbb{E}\left[R_{\boldsymbol{\mu}}(T)|\{m^*=m_f\},E_{m_f}\right]\\
&\leq \sum_{j\in\mathcal{U}\setminus B}\Delta_j z_j^*\frac{2\log(T\tilde{\Delta}^2_{n_j})}{\tilde{\Delta}^2_{n_j}}+\sum_{j\in B}\Delta_j\frac{2\log(T\tilde{\Delta}^2_{m_j})}{\tilde{\Delta}^2_{m_j}}\\
&\leq \sum_{j\in\mathcal{U}\setminus B}\Delta_j z_j^*\frac{32\log(T\hat{\Delta}^2_j)}{\hat{\Delta}^2_j}+\sum_{j\in B}\Delta_j\frac{32\log(T\Delta^2_j)}{\Delta^2_j},
\end{aligned} \tag{22}$$

where $\hat{\Delta}_j=\max\{2^{-\bar{m}+2},\min_{a:j\in G_a}\{\Delta_a\}\}$ and (z_j^*) is the solution of LP P_2.
Finally, we consider the term (ib). Note that $T_j(m)\geq n(m),\forall j\in B_m,\forall m$. An optimal action $*$ is not eliminated in round m^* if (25) holds for $m=m^*$. Hence, using (26) and (27), the probability p_m that $*$ is eliminated by a suboptimal action in any round m^* is at most $\frac{2}{T\tilde{\Delta}^2_{m^*}}$. Hence, term (ib) is given as:

$$\begin{aligned}
&\sum_{m=0}^{m_f}\mathbb{E}\left[\text{Regret from }\{j:m_j>m^*\}|\{m^*=m\},E_{m^*}\right]p_m\\
&\leq \sum_{m=0}^{m_f}\sum_{j\in\mathcal{U}:m_j\geq m}\frac{2}{T\tilde{\Delta}^2_m}.T\max_{a\in\mathcal{U}}\Delta_a\\
&\leq \max_{a\in\mathcal{U}}\Delta_a\sum_{m=0}^{m_f}\sum_{j\in\mathcal{U}:m_j\geq m}\frac{2}{\tilde{\Delta}^2_m}\\
&\leq \sum_{j\in\mathcal{U}}\sum_{m=0}^{m_j}\frac{2}{\tilde{\Delta}^2_m}\\
&\leq \sum_{j\in\mathcal{U}}2^{2m_j+2}\leq\sum_{j\in\mathcal{U}}\frac{64}{\Delta^2_j}=O(K).
\end{aligned} \tag{23}$$

Now we get the result (6) by combining the bounds in (21), (22), and (23).
Further, the definition of set B ensures that we have

$$\sum_{j\in B}\Delta_j\leq\sum_{j\in\mathcal{K}}z_j^*.$$

Also, using the Assumption 4, $\frac{32\Delta_j\log(T\hat{\Delta}^2_j)}{\hat{\Delta}^2_j},\frac{32\log(T\Delta^2_j)}{\Delta^2_j}$ are bounded by $C\log(T)$, where $C=\frac{32}{\min_{j\in\mathcal{U}}\Delta^2_j}$, is a constant independent of network structure. When one checks the feasibility of C, note that $\hat{\Delta}_j\geq\min_{a:j\in G_a}\Delta_a$ by definition and $\Delta_j\leq 1$ for any j since the

rewards are bounded by 1. Hence, (22) can be bounded as:

$$\begin{aligned}\sum_{j\in\mathcal{U}\setminus B}\Delta_j z_j^*\frac{32\log(T\hat{\Delta}_j^2)}{\hat{\Delta}_j^2}+\sum_{j\in B}\Delta_j\frac{32\log(T\Delta_j^2)}{\Delta_j^2}&\\ \leq\sum_{j\in\mathcal{U}\setminus B}z_j^*C\log(T)+\sum_{j\in B}\Delta_j C\log(T)&\\ \leq\sum_{j\in\mathcal{U}\setminus B}z_j^*C\log(T)+\sum_{j\in B}2^{-\bar{m}+1}C\log(T)&\\ \leq 2\sum_{j\in\mathcal{K}}z_j^*C\log(T).&\end{aligned}\tag{24}$$

Hence, we get (7) from (24), (21), and (23).

Appendix F. Supplementary Material

$S_n=\frac{1}{n}\sum_{j=1}^n X_j$ denotes the sample mean of the random variables $X_1,\ldots,X_n$. The first two lemmas below state the Chernoff-Hoeffding inequality and Bernstein's inequality.

Lemma 15 *Let $X_1,\ldots,X_n$ be a sequence of random variables with support $[0,1]$ and $\mathbb{E}[X_t]=\mu$ for all $t\leq n$. Let $S_n=\frac{1}{n}\sum_{j=1}^n X_j$. Then, for all $\epsilon>0$, we have,*

$$\begin{aligned}\mathbb{P}[S_n\geq\mu+\epsilon]&\leq e^{-2n\epsilon^2}\\ \mathbb{P}[S_n\leq\mu-\epsilon]&\leq e^{-2n\epsilon^2}.\end{aligned}$$

Lemma 16 *Let $X_1,\ldots,X_n$ be a sequence of random variables with support $[0,1]$ and $\sum_{k=1}^t var[X_k|X_1,\ldots,X_{k-1}]\leq\sigma^2$ for all $t\leq n$. Let $S_n=\sum_{j=1}^n X_j$. Then, for all $\epsilon>0$, we have,*

$$\begin{aligned}\mathbb{P}[S_n\geq\mathbb{E}[S_n]+\epsilon]&\leq\exp\left\{-\frac{\epsilon^2}{2\sigma^2+\frac{2}{3}\epsilon}\right\}\\ \mathbb{P}[S_n\leq\mathbb{E}[S_n]-\epsilon]&\leq\exp\left\{-\frac{\epsilon^2}{2\sigma^2+\frac{2}{3}\epsilon}\right\}.\end{aligned}$$

The next lemma is used in the proof of Proposition 10.

Lemma 17 *The probability that action j is not eliminated in round m_j by $*$ is at most $\frac{2}{T\tilde{\Delta}_{m_j}^2}$.*

Proof Let $\bar{f}_j(m)$ be the sample mean of all observations for action j available in round m. Let $\bar{f}^*(m)$ be the sample mean of the optimal action. The constraints of LP P_2 ensure that at the end of each round m, for all actions in B_m, we have at least $n(m):=\left\lceil\frac{2\log(T\tilde{\Delta}_m^2)}{\tilde{\Delta}_m^2}\right\rceil$ observations. The reason is as follows. The set A_m contains set B_m. In particular, $A_m=\cup_{i\in D_m}S_i$ and $D_m=\cup_{j\in B_m}\mathcal{K}_j$. If each action j in A_m is played z_j^* times, then all the base-arms in D_m have at least 1 observations according the constraints of LP P_2. Thus,

the actions in B_m have at least 1 observations. In sum, for all actions in B_m, we have at least $n(m)-n(m-1)$ observations at round m. Thus, we have at least $n(m)$ observations for all actions in B_m.

Now, for $m=m_j$, if we have,

$$\bar{f}_j(m)\leq\mu_j+\sqrt{\frac{\log(T\tilde{\Delta}_m^2)}{2n(m)}}\quad\text{and}\quad\bar{f}^*(m)\geq\mu^*-\sqrt{\frac{\log(T\tilde{\Delta}_m^2)}{2n(m)}},\tag{25}$$

then, action j is eliminated by $*$ in round m_j. In fact, in round m_j, we have

$$\sqrt{\frac{\log(T\tilde{\Delta}_{m_j}^2)}{2n(m_j)}}\leq\frac{\tilde{\Delta}_{m_j}}{2}<\frac{\Delta_j}{4}.$$

Hence, in the elimination phase of the UCB-LP policy, if (25) holds for action j in round m_j, we have,

$$\begin{aligned}\bar{f}_j(m_j)+\sqrt{\frac{\log(T\tilde{\Delta}_{m_j}^2)}{2n(m_j)}}&\leq\mu_j+2\sqrt{\frac{\log(T\tilde{\Delta}_{m_j}^2)}{2n(m_j)}}\\ &<\mu_j+\Delta_j-2\sqrt{\frac{\log(T\tilde{\Delta}_{m_j}^2)}{2n(m_j)}}\\ &=\mu^*-2\sqrt{\frac{\log(T\tilde{\Delta}_{m_j}^2)}{2n(m_j)}}\\ &\leq\bar{f}^*(m_j)-\sqrt{\frac{\log(T\tilde{\Delta}_{m_j}^2)}{2n(m_j)}},\end{aligned}$$

and action j is eliminated. Hence, the probability that action j is not eliminated in round m_j is the probability that either one of the inequalities in (25) do not hold. Using Chernoff-Hoeffding bound (Lemma 15), we can bound this as follows,

$$\mathbb{P}\left[\bar{f}_j(m)>\mu_j+\sqrt{\frac{\log(T\tilde{\Delta}_m^2)}{2n(m)}}\right]\leq\frac{1}{T\tilde{\Delta}_m^2}\tag{26}$$

$$\mathbb{P}\left[\bar{f}^*(m)<\mu^*-\sqrt{\frac{\log(T\tilde{\Delta}_m^2)}{2n(m)}}\right]\leq\frac{1}{T\tilde{\Delta}_m^2}.\tag{27}$$

Summing the above two inequalities for $m=m_j$ gives us that the probability that action j is not eliminated in round m_j by $*$ is at most $\frac{2}{T\tilde{\Delta}_{m_j}^2}$. ■

The next proposition is a modified version of Theorem 2 in Lai and Robbins (1985). We use it to obtain the regret lower bound in Proposition 4.

Proposition 18 *Suppose Assumptions 1, 2, and 3 hold. Let $M_i(t)$ be the total number of observations for such a base-arm i, for which $\vec{\theta}\in\Theta_i$. Then, under any uniformly good policy $\boldsymbol{\phi}$, we have that*

$$\liminf_{t\to\infty}\frac{\mathbb{E}[M_i(t)]}{\log(t)}\geq\frac{1}{J_i(\theta_i)}.$$

Proof By definition of $J_i(\theta_i)$, for every $\epsilon > 0$, there exists a $\theta_i' \in \mathcal{B}_i(\theta_i)$ such that $J_i(\theta_i) < D(\theta_i||\theta_i') < (1+\epsilon)J_i(\theta_i)$.

Now, under $\vec{\theta}_i' = [\theta_1, \ldots, \theta_i', \ldots \theta_N]$, there exists an action $k \in \mathcal{S}_i$ such that k is the unique optimal action. Then, for any uniformly good policy, for $0 < b < \delta$,

$$\mathbb{E}_{\vec{\theta}_i'}[t - T_k(t)] = o(t^b)$$

and therefore,

$$\mathbb{P}_{\vec{\theta}_i'}\left[T_k(t) < (1-\delta)\log(t)/D(\theta_i||\theta_i')\right] = o(t^{b-1}),$$

similar to the asymptotic lower bound proof in Lai and Robbins (1985).

Let $M_i(t)$ be the total number of observations for base-arm i. Then $M_i(t) \geq T_k(t)$, since choosing any action in $\mathcal{S}_i$ gives observations for i. Hence,

$$\mathbb{P}_{\vec{\theta}_i'}\left[M_i(t) < (1-\delta)\log(t)/D(\theta_i||\theta_i')\right] = o(t^{b-1}),$$

Now the rest of the proof of Theorem 2 in Lai and Robbins (1985) applies directly to $M_i(t)$. We will repeat it below for completeness. Let $(Y_i(r))_{r\geq 1}$ be the observations drawn from distribution F_i and define

$$L_m = \sum_{r=1}^{m} \log\left(\frac{g(Y_i(r);\theta_i)}{g(Y_i(r);\theta_i')}\right).$$

Now, we have that $\mathbb{P}_{\vec{\theta}_i'}[C_t] = o(t^{b-1})$ where $C_t = \{M_i(t) < (1-\delta)\log(t)/D(\theta_i||\theta_i')$ and $L_{M_i(t)} \leq (1-b)\log(t)\}$.

Now, we use the change of measure arguments.

$$\mathbb{P}_{\vec{\theta}_i'}[M_1(t) = m_1, \ldots, M_N(t) = m_N, L_{m_i} \leq (1-b)\log(t)] \tag{28}$$

$$= \int_{\{M_1(t)=m_1,\ldots,M_N(t)=m_N, L_{m_i}\leq(1-b)\log(t)\}} \Pi_{r=1}^{m_i} \frac{g(Y_i(r);\theta_i')}{g(Y_i(r);\theta_i)} dP_{\vec{\theta}_i} \tag{29}$$

$$\geq \exp(-(1-b)\log(t))\mathbb{P}_{\vec{\theta}_i}[M_1(t) = m_1, \ldots, M_N(t) = m_N, L_{m_i} \leq (1-b)\log(t)] \tag{30}$$

Since C_t is a disjoint union of events of the form $\{M_1(t) = m_1, \ldots, M_N(t) = m_N, L_{m_i} \leq (1-b)\log(t)\}$ with $m_i < (1-\delta)\log(t)/D(\theta_i||\theta_i')$, it follows that

$$\mathbb{P}_{\vec{\theta}}[C_t] \leq t^{1-b}\mathbb{P}_{\vec{\theta}_i'}[C_t] \to 0.$$

So far, we show that the probability of the event C_t goes to 0 as t goes to infinity. If we show the event $\{L_{M_i(t)} \leq (1-b)\log(t)|M_i(t) < (1-\delta)\log(t)/D(\theta_i||\theta_i')\}$ occurs almost surely, then we show the probability of $\{M_i(t) < (1-\delta)\log(t)/D(\theta_i||\theta_i')\}$ goes to 0 as t goes to infinity, which is the desired result. By strong law of large numbers $L_m/m \to D(\theta_i||\theta_i')$ as $m \to \infty$ and $\max_{r\leq m} L_r/m \to D(\theta_i||\theta_i')$ almost surely. Now, since $1-b > 1-\delta$, it follows that as $t \to \infty$,

$$\mathbb{P}_{\vec{\theta}}\left[L_r > (1-b)\log(t) \text{ for some } r < (1-\delta)\log(t)/D(\theta_i||\theta_i')\right] \to 0. \tag{31}$$

Hence, we have that as $t \to \infty$,

$$\mathbb{P}_{\vec{\theta}}\left[M_i(t) < (1-\delta)\log(t)/D(\theta_i||\theta_i')\right] \to 0.$$

By choosing ϵ, δ appropriately, this translates to

$$\liminf_{t\to\infty} \frac{\mathbb{E}[M_i(t)]}{\log(t)} \geq \frac{1}{J_i(\theta_i)}.$$

■

Proposition 19 *When the horizon is unknown, start the UCB-LP algorithm with $\tilde{T}_0 = 2$ and increase $\tilde{T}$ after reaching $\tilde{T}$ steps by setting $\tilde{T}_{l+1} = \tilde{T}_l^2$. The regret of unknown horizon UCB-LP is bounded by*

$$\sum_{j\in\mathcal{U}\setminus B} \frac{64\Delta_j z_j^*}{\hat{\Delta}_j^2}\log(T\hat{\Delta}_j^2) + \sum_{j\in B}\frac{64\log(T\Delta_j^2)}{\Delta_j} + O(K^2\log_2\log_2 T). \tag{32}$$

Proof When the horizon is unknown, start the UCB-LP algorithm with $\tilde{T}_0 = 2$ and increase $\tilde{T}$ after reaching $\tilde{T}$ steps by setting $\tilde{T}_{l+1} = \tilde{T}_l^2$. Thus, $\tilde{T}_l = 2^{2^l}$ until reaching horizon T. Also, the period in which horizon is reached is denoted by L. Note that $2 \leq L \leq \log_2\log_2 T$.

In any period $l, (0 \leq l \leq L)$, UCB-LP uses $\tilde{T}_l$ as input. Note that $\bar{m}, m_j, B$ and $\hat{\Delta}_j$ are independent of $\tilde{T}_l$, thus l. Recall that regret of UCB-LP is bounded by (6). The regret of UCB-LP with unknown horizon is upper bounded by the summation over all the periods.

$$\sum_{l=0}^{L}\left[\sum_{j\in\mathcal{U}\setminus B}\Delta_j z_j^*\frac{32\log(\tilde{T}_l\hat{\Delta}_j^2)}{\hat{\Delta}_j^2} + \sum_{j\in B}\frac{32\log(\tilde{T}_l\Delta_j^2)}{\Delta_j} + O(K^2)\right] = (i) + (ii) + (iii).$$

First, we consider the term (i). We can plug in the definition of $\tilde{T}_l$ into (i).

$$\begin{aligned}
(i) &= \sum_{l=0}^{L}\sum_{j\in\mathcal{U}\setminus B}\Delta_j z_j^*\frac{32\log(\tilde{T}_l\hat{\Delta}_j^2)}{\hat{\Delta}_j^2} \\
&= \sum_{j\in\mathcal{U}\setminus B}\frac{32\Delta_j z_j^*}{\hat{\Delta}_j^2}\sum_{l=0}^{L}\log(2^{2^l}\hat{\Delta}_j^2) \\
&= \sum_{j\in\mathcal{U}\setminus B}\frac{32\Delta_j z_j^*}{\hat{\Delta}_j^2}\left((\log 2)\sum_{l=0}^{L}2^l + (L+1)\log\hat{\Delta}_j^2\right) \\
&\leq \sum_{j\in\mathcal{U}\setminus B}\frac{32\Delta_j z_j^*}{\hat{\Delta}_j^2}\left(2^{L+1}(\log 2) + (L+1)\log\hat{\Delta}_j^2\right) \\
&\leq \sum_{j\in\mathcal{U}\setminus B}\frac{32\Delta_j z_j^*}{\hat{\Delta}_j^2}\left(2\log T + (L+1)\log\hat{\Delta}_j^2\right) \quad (since\ L \leq \log_2\log_2 T) \\
&\leq \sum_{j\in\mathcal{U}\setminus B}\frac{64\Delta_j z_j^*}{\hat{\Delta}_j^2}\log(T\hat{\Delta}_j^2) \quad (since\ (L+1)\log\hat{\Delta}_j^2 \leq 2\log\hat{\Delta}_j^2)
\end{aligned}$$

Similarly, we have that $(ii) \leq \sum_{j \in B} \frac{64 \log(T\Delta_j^2)}{\Delta_j}$. Now, we directly sum up the bound for term (iii).

$$(iii) \leq \sum_{l=0}^{L} O(K^2) \leq (L+1)O(K^2) = O(K^2 \log_2 \log_2 T).$$

Hence, by combining the results above, the regret of unknown horizon is bounded by

$$\sum_{j \in \mathcal{U} \setminus B} \frac{64 \Delta_j z_j^*}{\hat{\Delta}_j^2} \log(T\hat{\Delta}_j^2) + \sum_{j \in B} \frac{64 \log(T\Delta_j^2)}{\Delta_j} + O(K^2 \log_2 \log_2 T).$$

■

References

Peter Auer and Ronald Ortner. UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1-2):55–65, 2010.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, May 2002. ISSN 0885-6125.

Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. *X*-armed bandits. *Journal of Machine Learning Research*, 12:1655–1695, 2011.

Swapna Buccapatnam, Atilla Eryilmaz, and Ness B. Shroff. Stochastic bandits with side observations on networks. *SIGMETRICS Perform. Eval. Rev.*, 42(1):289–300, June 2014. ISSN 0163-5999.

S. Caron, B. Kveton, M. Lelarge, and S. Bhagat. Leveraging side observations in stochastic bandits. In *UAI*, pages 142–151. AUAI Press, 2012.

Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework and applications. In *Proceedings of the 30th International Conference on Machine Learning*, pages 151–159, 2013.

Alon Cohen, Tamir Hazan, and Tomer Koren. Online learning with feedback graphs without the graphs. *CoRR*, abs/1605.07018, 2016.

Colin Cooper, Ralf Klasing, and Michele Zito. Lower bounds and algorithms for dominating sets in web graphs. *Internet Mathematics*, 2:275–300, 2005.

Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors: A multilevel approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11):1944–1957, 2007.

M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 135–142. ACM, 2010.

T. L. Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 661–670. ACM, 2010.

Shie Mannor and Ohad Shamir. From bandits to experts: On the value of side-observations. In *NIPS*, pages 684–692, 2011.

Sandeep Pandey, Deepayan Chakrabarti, and Deepak Agarwal. Multi-armed bandit problems with dependent arms. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 721–728, New York, NY, USA, 2007. ACM.

Paat Rusmevichientong and John N. Tsitsiklis. Linearly parameterized bandits. *Math. Oper. Res.*, 35(2):395–411, 2010.

Aristide C. Y. Tossou, Christos Dimitrakakis, and Devdatt Dubhashi. Thompson sampling for stochastic bandits with graph feedback. *CoRR*, abs/ 1701.04238, 2017.

Johan Ugander, Brian Karrer, Lars Backstrom, and Cameron Marlow. The anatomy of the facebook social graph. *CoRR*, abs/1111.4503, 2011.

Journal of Machine Learning Research 18 (2018) 1-58 Submitted 1/17; Revised 11/17; Published 4/18

Simultaneous Clustering and Estimation of Heterogeneous Graphical Models

Botao Hao HAO22@PURDUE.EDU
Department of Statistics
Purdue University
West Lafayette, IN 47906, USA

Will Wei Sun WSUN@BUS.MIAMI.EDU
Department of Management Science
University of Miami School of Business Administration
Miami, FL 33146, USA

Yufeng Liu YFLIU@EMAIL.UNC.EDU
Department of Statistics and Operations Research
Department of Genetics
Department of Biostatistics
Carolina Center for Genome Sciences
Lineberger Comprehensive Cancer Center
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599, USA

Guang Cheng CHENGG@PURDUE.EDU
Department of Statistics
Purdue University
West Lafayette, IN 47906, USA

Editor: Koji Tsuda

Abstract

We consider joint estimation of multiple graphical models arising from heterogeneous and high-dimensional observations. Unlike most previous approaches which assume that the cluster structure is given in advance, an appealing feature of our method is to learn cluster structure while estimating heterogeneous graphical models. This is achieved via a high dimensional version of Expectation Conditional Maximization (ECM) algorithm (Meng and Rubin, 1993). A joint graphical lasso penalty is imposed on the conditional maximization step to extract both homogeneity and heterogeneity components across all clusters. Our algorithm is computationally efficient due to fast sparse learning routines and can be implemented without unsupervised learning knowledge. The superior performance of our method is demonstrated by extensive experiments and its application to a Glioblastoma cancer dataset reveals some new insights in understanding the Glioblastoma cancer. In theory, a non-asymptotic error bound is established for the output directly from our high dimensional ECM algorithm, and it consists of two quantities: *statistical error* (statistical accuracy) and *optimization error* (computational complexity). Such a result gives a theoretical guideline in terminating our ECM iterations.

Keywords: Clustering, finite-sample analysis, graphical models, high-dimensional statistics, non-convex optimization.

1. Introduction

Graphical models have been widely employed to represent conditional dependence relationships among a set of variables. The structure recovery of an undirected Gaussian graph is known to be equivalent to recovering the support of its corresponding precision matrix (Lauritzen, 1996). In the situation where data dimension is comparable to or much larger than the sample size, the penalized likelihood method is proven to be an effective way to learn the structure of graphical models (Yuan and Lin, 2007; Friedman et al., 2008; Shojaie and Michailidis, 2010a,b). When observations come from several distinct subpopulations, a naive way is to estimate each graphical model separately. However, separate estimation ignores the information of common structure shared across different subpopulations, and thus can be inefficient in some real applications. For instance, in the glioblastoma multiforme (GBM) cancer dataset from The Cancer Genome Atlas Research Network (TCGA, 2008), Verhaak et al. (2010) showed that GBM cancer could be classified into four subtypes. Based on this cluster structure, it has been suggested that although the graphs across four subtypes differ in some edges, they share many common structures. In this case, the naive procedure can be suboptimal (Danaher et al., 2014; Lee and Liu, 2015). Such applications have motivated recent studies on joint estimation methods (Guo et al., 2011; Danaher et al., 2014; Lee and Liu, 2015; Qiu et al., 2016; Wang, 2015; Cai et al., 2016a; Peterson et al., 2015) that encourage common structure in estimating heterogeneous graphical models. However, all aforementioned approaches crucially rely on an assumption that the class label of each sample is known in advance.

For certain problems, prior knowledge of the class membership may be available. But this may not be the case for the massive data with complex and unknown population structures. For instance, in online advertising, an important task is to find the most suitable advertisement (ad) for a given user in a specific online context. This could increase the chance of users' favorable actions (e.g., click the ad, inquire about or purchase a product). In recent years, user clustering has gained increasing attention due to its superior performance of ad targeting. This is because users with similar attributes, such as gender, age, income, geographic information, and online behaviors, tend to behave similarly to the same ad (Yan et al., 2009). Moreover, it is very important to understand conditional dependence relationships among user attributes in order to improve ad targeting accuracy (Wang et al., 2015a). Such conditional dependence relationships are expected to share commonality across different groups (user homogeneity) while maintaining some levels of uniqueness within each group (user heterogeneity) (Jeziorski and Segal, 2015). In this online advertising application, previously mentioned joint estimation methods are no longer applicable as they need to know the user cluster structure in advance. Furthermore, with the data being continuously collected, the number of underlying user clusters grows with the sample size (Chen et al., 2009). This provides another reason for simultaneously conducting user clustering and joint graphical model estimation, which is much needed in the era of big data.

Our contributions in this paper are two-fold. On the methodological side, we propose a general framework of **S**imultaneous **C**lustering **A**nd estimatio**N** of heterogeneous graphical models (SCAN). SCAN is a likelihood based method which treats the underlying class label as a latent variable. Based on a high-dimensional version of Expectation Conditional

Maximization (ECM) algorithm (Meng and Rubin, 1993), we are able to conduct clustering and sparse graphical model learning at the same time. In each iteration of the ECM algorithm, the expectation step performs cluster analysis by estimating missing labels and the conditional maximization step conducts feature selection and joint estimation of heterogeneous graphical models via a penalization procedure. With an iteratively updating process, the estimation for both cluster structure and sparse precision matrices becomes more and more refined. Our algorithm is computationally efficient by taking advantage of the fast sparse learning in the conditional maximization step. Moreover, it can be implemented in a user-friendly fashion, without the need of additional unsupervising learning knowledge.

As a promising application, we apply the SCAN method on the GBM cancer dataset to simultaneously cluster the GBM patients and construct the gene regulatory network of each subtype. Our method greatly outperforms the competitors in clustering accuracy and delivers new insights in understanding the GBM disease. Figure 1 reports four gene networks estimated from the SCAN method. The black lines are links shared in all four subtypes, and the color lines are uniquely presented in some subtypes. Our findings generally agree with the GBM disease literature (Verhaak et al., 2010). Besides common edges of all subtypes, we have discovered some unique gene connections that were not found through separate estimation (Danaher et al., 2014; Lee and Liu, 2015). This new finding suggests further investigation on their possible impact on the GBM disease. See Section 4.5 for more discussions.

On the theoretical side, we develop non-asymptotic statistical analysis for the output directly from the high dimensional ECM algorithm. This is nontrivial due to the non-convexity of the likelihood function. In this case, there is no guarantee that the sample-based estimator is close to the maximum likelihood estimator. Hence, we need to directly evaluate the estimation error in each iteration. Let $\mathbf{\Theta}$ represent vectorized cluster means $\boldsymbol{\mu}_k$ and precision matrices $\mathbf{\Omega}_k$, see (3) for a formal definition. Given an appropriate initialization $\mathbf{\Theta}^{(0)}$, the finite sample error bound of the t-th step solution $\mathbf{\Theta}^{(t)}$ consists of two parts:

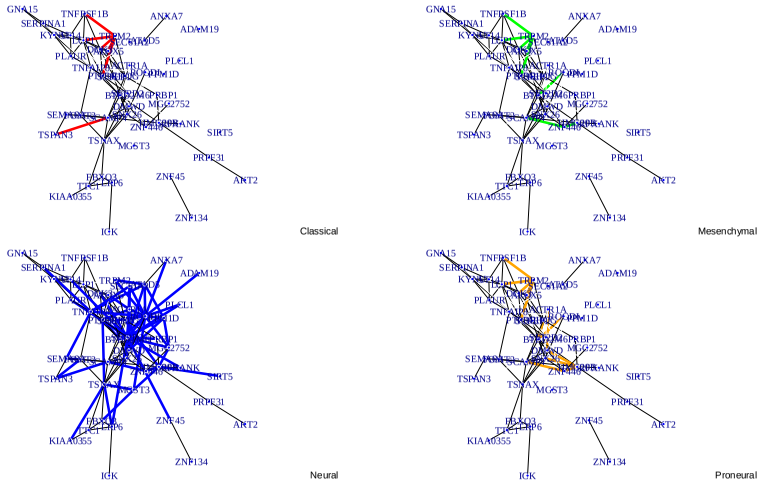
$$\left\|\mathbf{\Theta}^{(t)}-\mathbf{\Theta}^{*}\right\|_2 \leq \underbrace{C\cdot\varepsilon\left(n,p,K,\Psi(\mathcal{M})\right)}_{\text{Statistical Error(SE)}}+\underbrace{\kappa^t\left\|\mathbf{\Theta}^{(0)}-\mathbf{\Theta}^{*}\right\|_2}_{\text{Optimization Error(OE)}}, \tag{1}$$

with high probability. Here, K is the number of clusters, $\Psi(\mathcal{M})$ measures the sparsity of cluster means and precision matrices, and $\kappa\in(0,1)$ is a contraction coefficient. The above theoretical analysis is applicable to any decomposable penalty used in the conditional maximization step.

The error bound (1) enables us to monitor the dynamics of estimation error in each iteration. Specifically, the optimization error decays geometrically with the iteration number t, while the statistical error remains the same when t grows. Therefore, the maximal number of iterations T is implied, beyond which the optimization error is dominated by the statistical error such that consequently the whole error bound is in the same order as the statistical error. In particular,

$$\sum_{k=1}^{K}\left(\left\|\boldsymbol{\mu}_k^{(T)}-\boldsymbol{\mu}_k^{*}\right\|_2+\left\|\mathbf{\Omega}_k^{(T)}-\mathbf{\Omega}_k^{*}\right\|_F\right)=O_P\left(\underbrace{\sqrt{\frac{K^5 d\log p}{n}}}_{\text{Cluster means error}}+\underbrace{\sqrt{\frac{K^3(Ks+p)\log p}{n}}}_{\text{Precision matrices error}}\right),$$

Figure 1: Estimated gene networks corresponding to the Classical, Mesenchymal, Neural and Proneural clusters from our SCAN method applying to the Glioblastoma Cancer Data. In each network, the black lines are the links shared in all four groups. The color lines are the edges shared by some subtypes.



where d and s are the sparsity for a single cluster mean and precision matrix. This result indicates that, after T steps, the SCAN estimator will fall within statistical precision of the true parameter $\{\boldsymbol{\mu}_k^{*},\mathbf{\Omega}_k^{*}\}$. It is worth mentioning that our theory allows the number of clusters K to diverge polynomially with the sample size, reflecting a typical big data scenario. When K is fixed, our statistical rate for the precision matrix estimation under the Frobenius norm, i.e., $O_P(\sqrt{(s+p)\log p/n})$, achieves the optimal rate established in Theorem 7 of Cai et al. (2016b), which is the best rate we could obtain even when the true cluster structure is given.

In the literature, a related line of research focuses on methodological developments of high-dimensional clustering. Pan and Shen (2007) and Sun et al. (2012) introduced regularized model-based clustering and regularized K-means clustering, and Zhou et al. (2009) proposed a network-based clustering approach by imposing a graphical lasso to each individual precision matrix estimation. However, the regularized model-based clustering assumes an identical covariance matrix in each cluster, while the network-based clustering treats each graphical model estimation separately. As pointed out in Danaher et al. (2014) and Lee and Liu (2015), ignoring the network information of other clusters may lead to suboptimal graphical model estimation. During the submission of our paper, we became aware of an independent work by Gao et al. (2016) who also considered the multiple precision ma-

trices estimation via a Gaussian mixture model. Different from ours, Gao et al. (2016) did not enforce the sparsity in the cluster means, which would inevitably lead to sub-optimal estimators in high-dimensional clustering (Yi and Caramanis, 2015; Wang et al., 2015b). Most importantly, no theoretical guarantee was provided in Zhou et al. (2009) and Gao et al. (2016). On the other hand, our SCAN method is more general than these existing methods since we allow the sparsity in both cluster means and precision matrices, and our theoretical analysis of the general SCAN framework sheds some lights on the behavior of these existing method, See Remark 1 for more discussions. In addition, in terms of the heterogeneous graphical model estimation, Saegusa and Shojaie (2016) proposed an interesting two-stage method which used hierarchical clustering to obtain cluster memberships and then estimated the multiple graphical models based on the attained cluster assignments. Despite its simplicity, it is unclear how the performance of clustering in the first stage could affect the performance of precision matrix estimation in the second stage. In comparison, our approach unifies clustering and parameter estimation into one optimization framework, which allows us to quantify both estimation errors in each iteration.

Another line of related work is the theoretical analysis of EM algorithm (Balakrishnan et al., 2016; Yi and Caramanis, 2015; Wang et al., 2015b). Specifically, Balakrishnan et al. (2016) studied the low-dimensional Gaussian mixture model, while Wang et al. (2015b) and Yi and Caramanis (2015) considered its high dimensional extensions. However, their methods are not applicable for the estimation of heterogeneous graphical models due to the assumed identity covariance matrix. In fact, our consideration of the general covariance matrix demands more challenging technical analysis since simultaneous estimation of cluster means and covariance matrices induces a bi-convex optimization beyond the non-convexity of the EM algorithm itself. This also explains why ECM is needed instead of EM. To address these technical issues, key ingredients of our theoretical analysis are to bound the dual norm of the gradient of an auxiliary Q-function and employ nice properties of bi-convex optimization (Boyd et al., 2011) in the regularized M-estimation framework (Negahban et al., 2012). See Section 3 for more details.

In terms of notation, we use $[K]$ to denote the set $\{1, 2, \ldots, K\}$. For a vector $\boldsymbol{\mu} \in \mathbb{R}^p$, $\|\boldsymbol{\mu}\|_2$ is its Euclidean norm. For a matrix $\boldsymbol{X} \in \mathbb{R}^{p_1 \times p_2}$, we denote $\|\boldsymbol{X}\|_F$ and $\|\boldsymbol{X}\|_2$ as its Frobenius norm and spectral norm, respectively, and define its matrix max norm as $\|\boldsymbol{X}\|_{\max} = \max_{i,j} |X_{ij}|$ and its max induced norm as $\|\boldsymbol{X}\|_\infty = \max_{i=1,\ldots,p_1} \sum_{j=1}^{p_2} |X_{ij}|$, which is simply the maximum absolute row sum of the matrix. For a square matrix $\boldsymbol{A} \in \mathbb{R}^{p \times p}$, let $\sigma_{\min}(\boldsymbol{A})$ and $\sigma_{\max}(\boldsymbol{A})$ be its smallest and largest eigenvalue respectively and $|\boldsymbol{A}|$ be its determinant. For a sub-Gaussian random variable Z, we use $\|Z\|_{\psi_2}$ and $\|Z\|_{\psi_1}$ to denote its Orlicz norm. Specifically, $\|Z\|_{\psi_2} = \sup_{p \geq 1} p^{-1/2} (\mathbb{E}|Z|^p)^{1/p}$ and $\|Z\|_{\psi_1} = \sup_{p \geq 1} p^{-1} (\mathbb{E}|Z|^p)^{1/p}$. For two sequences $\{a_n\}$ and $\{b_n\}$ of positive numbers, $a_n \lesssim b_n$ refers to the case that $a_n \leq C b_n$ for some uniform constant C. We write $1(\cdot)$ as an indicator function. Throughout this paper, we use $C, C_1, C_2, \ldots D, D_1, D_2, \ldots$ to denote generic absolute constants, whose values may vary at different places.

The rest of this article is organized as follows. Section 2 introduces heterogeneous graphical models and the SCAN method. Section 3 provides some statistical guarantees for the output directly from the SCAN method. Section 4 shows some simulation results as well as a real data analysis on the Glioblastoma cancer data. Section 5 gives some discussions

for future works. The appendix is devoted to the technical details of the main theorems, and the online supplementary material contains all the supporting lemmas and their proofs.

2. Methodology

In this section, we introduce the SCAN method that simultaneously conducts high-dimensional clustering and estimation of heterogeneous graphical models.

2.1 Heterogeneous Graphical Models

We start our discussions from heterogeneous graphical models with known labels. Assume we are given K groups of data sets $\mathcal{A}_1, \ldots, \mathcal{A}_K$ and the samples in the k-th group are generated i.i.d. from the following Gaussian distribution:

$$f_k(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = (2\pi)^{-p/2} |\boldsymbol{\Sigma}_k|^{-1/2} \exp\left\{ -\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_k) \right\}, k = 1, \ldots, K. \quad (2)$$

Let $\boldsymbol{\Omega}_k = \boldsymbol{\Sigma}_k^{-1}$ be the k-th precision matrix with the ij-th entry ω_{kij}. For the k-th pair of parameters $(\boldsymbol{\mu}_k, \boldsymbol{\Omega}_k)$, i.e.,

$$\boldsymbol{\mu}_k = \begin{pmatrix} \mu_{k1} \\ \vdots \\ \mu_{kp} \end{pmatrix}, \boldsymbol{\Omega}_k = \begin{pmatrix} \omega_{k11} & \cdots & \omega_{k1p} \\ \vdots & \ddots & \vdots \\ \omega_{kp1} & \cdots & \omega_{kpp} \end{pmatrix},$$

we write $\boldsymbol{\Theta}_k := \mathrm{vec}(\boldsymbol{\mu}_k, \boldsymbol{\Omega}_k) = (\mu_{k1}, \ldots, \mu_{kp}, \omega_{k11}, \ldots, \omega_{kp1}, \ldots, \omega_{k1p}, \ldots, \omega_{kpp}) \in \mathbb{R}^{p^2+p}$ as its vectorized representation, and write the parameter of interest $\boldsymbol{\Theta}$ as

$$\boldsymbol{\Theta} = \left(\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_K\right)^\top \in \mathbb{R}^{K(p^2+p)}. \quad (3)$$

Note that the degrees of freedom of $\boldsymbol{\Theta}$ are $K(0.5p^2 + 1.5p)$, including K sets of p means, p variances, as well as $p(p-1)/2$ covariances.

In some cases, there may also exist some common structure across K precision matrices. Danaher et al. (2014) formulated the joint estimation of heterogeneous graphical models as

$$\underset{\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_K \succ 0}{\operatorname{argmax}} \sum_{k=1}^{K} \sum_{\boldsymbol{x} \in \mathcal{A}_k} \log f_k(\boldsymbol{x}; \boldsymbol{\Theta}_k) - \mathcal{P}(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_K), \quad (4)$$

where $\mathcal{P}(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_K)$ is an entry-wise penalty which encourages both sparsity of each individual precision matrix and similarity among all precision matrices.

In practice, the cluster label is not always available. A probabilistic model is thus needed to accommodate the latent structure in the data. Assume the observation $\boldsymbol{x}_i; i = 1, \ldots, n$, from unlabeled heterogeneous population has the underlying density

$$f(\boldsymbol{x}, \boldsymbol{\Theta}) = \sum_{k=1}^{K} \pi_k f_k(\boldsymbol{x}; \boldsymbol{\Theta}_k), \quad (5)$$

where π_k is the probability that an observation $\boldsymbol{x}_i$ belongs to the k-th subpopulation. Here, for simplicity we assume the number of cluster K is identifiable. In order to ensure the

identifiability of fixed-dimensional Gaussian graphical models, some sufficient conditions such as the strong identifiability condition was imposed on the density functions. However these conditions are hard to verify in practice. In fact, the identifiability issue for high dimensional mixture model is still an open problem (Ho and Nguyen, 2015) and is beyond the scope of this paper.

Consider the penalized log-likelihood function for the *observed data*

$$\log \mathcal{L}(\boldsymbol{\Theta}|\boldsymbol{X}) := \frac{1}{n}\sum_{i=1}^{n}\log\left(\sum_{k=1}^{K}\pi_k f_k\Big(\boldsymbol{x}_i;\boldsymbol{\mu}_k,(\boldsymbol{\Omega}_k)^{-1}\Big)\right) - \mathcal{R}(\boldsymbol{\Theta}).$$

Our **S**imultaneous **C**lustering **A**nd estimatio**N** (SCAN) method aims to solve

$$\max_{\pi_k,\boldsymbol{\mu}_k,\boldsymbol{\Omega}_k} \log \mathcal{L}(\boldsymbol{\Theta}|\boldsymbol{X}). \tag{6}$$

For an illustration, we take

$$\mathcal{R}(\boldsymbol{\Theta}) = \lambda_1 \underbrace{\sum_{k=1}^{K}\sum_{j=1}^{p}|\mu_{kj}|}_{\mathcal{P}_1(\boldsymbol{\Theta})} + \lambda_2 \underbrace{\sum_{k=1}^{K}\sum_{i\neq j}|\omega_{kij}|}_{\mathcal{P}_2(\boldsymbol{\Theta})} + \lambda_3 \underbrace{\sum_{i\neq j}(\sum_{k=1}^{K}\omega_{kij}^2)^{1/2}}_{\mathcal{P}_3(\boldsymbol{\Theta})}, \tag{7}$$

where $\mathcal{P}_1(\boldsymbol{\Theta})$ and $\mathcal{P}_2(\boldsymbol{\Theta})$ impose sparsity of the estimated cluster mean and precision matrix, and $\mathcal{P}_3(\boldsymbol{\Theta})$ encourages similarity among all estimated precision matrices. The above three tuning parameters can be tuned efficiently via adaptive BIC. More details can be found in Section 4.1.

Remark 1 *It is worth mentioning that our SCAN method is applicable to penalty functions other than* (7)*. For instance, the cluster mean penalty can be replaced by the group lasso penalty in Sun et al. (2012) or the ℓ_0-norm penalty in Shen et al. (2012). The group graphical lasso penalty for the precision matrix estimation can be substituted by the structural pursuit penalty in Zhu et al. (2014) or the weighted bridge penalty in Rothman and Forzani (2014). As shown in Section 2.2, only a slight modification of our algorithm is needed to accommodate other penalty functions. We also note that SCAN reduces to the regularized model-based clustering (Pan and Shen, 2007) when $\lambda_2 = \lambda_3 = 0$, reduces to the method by Zhou et al. (2009) when $\lambda_3 = 0$, and reduces to the method by Gao et al. (2016) when $\lambda_1 = 0$. Consequently, the technical tools developed for the SCAN estimator in Section 3 are also applicable to these special cases.*

2.2 ECM Algorithm

In this subsection, we introduce an efficient ECM algorithm to solve the general non-convex optimization problem in (6). The ECM replaces each M-step with an conditional maximization (CM) step in which each parameter $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Omega}_k$ is maximized separately, by fixing other parameters.

Denote the latent cluster assignment matrix as $\boldsymbol{L}$, where $L_{ik} = 1(\boldsymbol{x}_i \in \mathcal{A}_k)$; $i = 1,\ldots,n$, $k = 1,\ldots,K$. If the cluster label L_{ik} is available, the penalized log-likelihood function for the *complete data* can be formulated as

$$\log \mathcal{L}(\boldsymbol{\Theta}|\boldsymbol{X},\boldsymbol{L}) := \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} L_{ik}\Big[\log\pi_k + \log f_k(\boldsymbol{x}_i;\boldsymbol{\Theta}_k)\Big] - \mathcal{R}(\boldsymbol{\Theta}).$$

In the expectation step, the conditional expectation of the penalized log-likelihood function is computed as

$$\mathbb{E}_{\boldsymbol{L}|\boldsymbol{X},\boldsymbol{\Theta}^{(t-1)}}\Big[\log\mathcal{L}(\boldsymbol{\Theta}|\boldsymbol{X},\boldsymbol{L})\Big] = Q_n(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t-1)}) - \mathcal{R}(\boldsymbol{\Theta}), \tag{8}$$

where $\mathcal{R}(\boldsymbol{\Theta})$ is the penalty in (7) and

$$Q_n(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t-1)}) := \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} L_{\boldsymbol{\Theta}^{(t-1)},k}(\boldsymbol{x}_i)\Big[\log\pi_k + \log f_k(\boldsymbol{x}_i;\boldsymbol{\Theta}_k)\Big], \tag{9}$$

with the class label being computed based on the parameter $\boldsymbol{\Theta}^{(t-1)}$ and $\pi_k^{(t-1)}$ obtained at the previous iteration, that is,

$$L_{\boldsymbol{\Theta}^{(t-1)},k}(\boldsymbol{x}_i) = \frac{\pi_k^{(t-1)} f_k(\boldsymbol{x}_i;\boldsymbol{\Theta}_k^{(t-1)})}{\sum_{k=1}^{K}\pi_k^{(t-1)} f_k(\boldsymbol{x}_i;\boldsymbol{\Theta}_k^{(t-1)})}. \tag{10}$$

In the conditional maximization step, maximizing (8) with respect to π_k, $\boldsymbol{\mu}_k$, $\boldsymbol{\Omega}_k$ yields the update of parameters. In particular, the update of π_k is given as

$$\pi_k^{(t)} = \sum_{i=1}^{n}\frac{L_{\boldsymbol{\Theta}^{(t-1)},k}(\boldsymbol{x}_i)}{n}, \tag{11}$$

and the update of $\boldsymbol{\mu}_k$ is given in the following Lemma.

Lemma 2 *Let $\boldsymbol{\mu}_k^{(t)} := \arg\max_{\boldsymbol{\mu}_k} Q_n(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t-1)}) - \mathcal{R}(\boldsymbol{\Theta})$ and denote $n_k := \sum_{i=1}^{n} L_{\boldsymbol{\Theta}^{(t-1)},k}(\boldsymbol{x}_i)$. We have, for $j = 1,\ldots,p$,*

$$\mu_{kj}^{(t)} = \begin{cases} g_{1,j}(\boldsymbol{x};\boldsymbol{\Theta}_k^{(t-1)}) - \frac{n\lambda_1}{n_k\omega_{kjj}^{(t-1)}}\, sign(\mu_{kj}^{(t-1)}) & \text{if } \Big|\sum_{i=1}^{n} g_{2,j}(\boldsymbol{x}_i;\boldsymbol{\Theta}_k^{(t-1)})\Big| > \lambda_1; \\ 0 & \text{otherwise,} \end{cases}$$

where

$$g_{1,j}(\boldsymbol{x};\boldsymbol{\Theta}_k^{(t-1)}) = \frac{\sum_{i=1}^{n} L_{\boldsymbol{\Theta}^{(t-1)},k}(\boldsymbol{x}_i)\big(\sum_{l=1}^{p} x_{il}\omega_{klj}^{(t-1)}\big)}{\omega_{kjj}^{(t-1)} n_k} - \frac{\sum_{l=1}^{p}\mu_{kl}^{(t-1)}\omega_{klj}^{(t-1)}}{\omega_{kjj}^{(t-1)}} + \mu_{kj}^{(t-1)},$$

$$g_{2,j}(\boldsymbol{x}_i;\boldsymbol{\Theta}_k^{(t-1)}) = L_{\boldsymbol{\Theta}^{(t-1)},k}(\boldsymbol{x}_i)\Big(\sum_{l=1,l\neq j}^{p}(x_{il}-\mu_{kl}^{(t-1)})\omega_{klj}^{(t-1)} + x_{ij}\omega_{kjj}^{(t-1)}\Big).$$

Note that if the lasso penalty is replaced with other penalty functions, then the update formula of $\boldsymbol{\mu}_k^{(t)}$ in Lemma 2 can be modified accordingly. Given the pseudo sample covariance matrix $\widetilde{S}_k$, we are able to develop an update formula for $\boldsymbol{\Omega}_k$ by establishing its connection with joint estimation of heterogeneous graphical models (4).

Lemma 3 *The solution of maximizing (8) with respect to $(\mathbf{\Omega}_1, \ldots, \mathbf{\Omega}_K)$ is equivalent to*

$$(\mathbf{\Omega}_1^{(t)}, \ldots, \mathbf{\Omega}_K^{(t)}) := \arg \max_{\mathbf{\Omega}_1, \ldots, \mathbf{\Omega}_K \succ 0} \sum_{k=1}^{K} n_k \Big[\log\det(\mathbf{\Omega}_k) - trace(\widetilde{S}_k \mathbf{\Omega}_k) \Big] - \mathcal{R}(\mathbf{\Theta}), \tag{12}$$

where $\widetilde{S}_k$ is a pseudo sample covariance matrix defined as

$$\widetilde{S}_k := \frac{\sum_{i=1}^{n} L_{\mathbf{\Theta}^{(t-1)},k}(\boldsymbol{x}_i)(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(t-1)})^\top (\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(t-1)})}{\sum_{i=1}^{n} L_{\mathbf{\Theta}^{(t-1)},k}(\boldsymbol{x}_i)}.$$

The solution for (12) can be solved efficiently via the ADMM algorithm by slightly modifying the joint graphical lasso algorithm in Danaher et al. (2014). Since Danaher et al. (2014) do not impose the symmetry condition for precision matrix update, $\{\mathbf{\Omega}_k^{(T)}\}_{k=1}^K$ in general is not necessarily symmetric. Following the symmetrization strategy in Cai et al. (2011) and Cai et al. (2016a), we symmetrize $\mathbf{\Omega}_k^{(t)}$ by

$$\omega_{kij}^{(t)} = \omega_{kij}^{(t)} I(|\omega_{kij}^{(t)} \leq \omega_{kij}^{(t)}|) + \omega_{kji}^{(t)} I(|\omega_{kij}^{(t)} > \omega_{kij}^{(t)}|), \tag{13}$$

where $\omega_{kij}^{(t)}$ is the ij-th entry of $\mathbf{\Omega}_k^{(t)}$ and $I(\cdot)$ is the indicator function. This step will not affect the convergence rate of the final estimator, which is illustrated in Cai et al. (2011) and Cai et al. (2016a). We summarize the high-dimensional ECM algorithm for solving the SCAN method in Table 1. Our algorithm is computationally efficient due to fast sparse learning routines shown in Lemmas 2 and 3.

Table 1: The SCAN Algorithm

Input: $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, number of clusters K, tuning parameters $\lambda_1, \lambda_2, \lambda_3$.
Output: Cluster label $\boldsymbol{L}$, cluster mean $\boldsymbol{\mu}_k$ and precision matrix $\mathbf{\Omega}_k$.
Step 1: Initialize cluster mean $\boldsymbol{\mu}_k^{(0)}$, positive definite precision matrix $\mathbf{\Omega}_k^{(0)}$, and set $\pi_k^{(0)} = 1/K$, for each $k \in [K]$.
Step 2: Until some termination conditions are met, for iteration $t = 1, 2, \ldots$
(a) E-step. Find the cluster assignment $L_{\mathbf{\Theta}^{(t-1)},k}(\boldsymbol{x}_i)$ as in (10).
(b) CM-step. Given $L_{\mathbf{\Theta}^{(t-1)},k}(\boldsymbol{x}_i)$, update $\pi_k^{(t)}$, $\boldsymbol{\mu}_k^{(t)}$, and $\mathbf{\Omega}_k^{(t)}$ in (11), Lemma 2, Lemma 3, respectively. Symmetrize $\mathbf{\Omega}_k^{(t)}$ by (13).

In all of our experiments, we obtain $(\boldsymbol{\mu}_k^{(0)}, \mathbf{\Omega}_k^{(0)})$ by random initialization, which is computationally efficient and practically reliable. In the theoretical study, we require the initialization to be of a constant distance to the truth. See Remark 14 for more discussions. Moreover, in the implementation, ECM step in Step 2 is terminated when the updated parameters are close to their previous values:

$$\sum_{k=1}^{K} \left\{ \frac{\|\boldsymbol{\mu}_k^{(t)} - \boldsymbol{\mu}_k^{(t-1)}\|_2}{\|\boldsymbol{\mu}_k^{(t)}\|_2} + \frac{\|\mathbf{\Omega}_k^{(t)} - \mathbf{\Omega}_k^{(t-1)}\|_F}{\|\mathbf{\Omega}_k^{(t)}\|_F} \right\} \leq 0.01.$$

Remark 4 *In the existing high-dimensional EM algorithms where the covariance matrix is assumed to be an identity matrix (Wang et al., 2015b; Yi and Caramanis, 2015), sample-splitting procedures have been routinely used in the M-step in order to facilitate the theoretical analysis. Although it simplifies theoretical developments, such a sample-splitting procedure does not take advantage of full samples in the M-step and is hard to implement in practice. Our Algorithm 1 is able to avoid this sample-splitting step but still enjoys nice theoretical properties. See Corollary 18 for more discussions on its statistical guarantee.*

3. Statistical Guarantee

In this section, we establish statistical guarantee for the SCAN estimator based on sample-based analysis of (9) and population-based analysis of (16). Here, we consider the high-dimensional setting where $p \gg n$ and K is allowed to diverge with n.

We start by introducing some useful notation. Denote the index set of diagonal components of K precision matrices by

$$\mathcal{G} = \bigcup_{k=1}^{K} \mathcal{G}_k, \text{with } \mathcal{G}_k = \big(k(p+1), k(2p+2), \ldots, k(p^2+p)\big), \tag{14}$$

that is, $\mathbf{\Theta}_{\mathcal{G}} = (\omega_{111}, \ldots, \omega_{1pp}, \ldots, \omega_{K11}, \ldots, \omega_{Kpp}) \in \mathbb{R}^{Kp}$. Let $\mathcal{O}$ be the complete index set of $\mathbf{\Theta}$ and $\mathcal{G}^c = \mathcal{O} \setminus \mathcal{G}$ be the complement set of $\mathcal{G}$. Denote $\mathcal{U}_k := \{i : \mu_{ki}^* \neq 0\}$ where $\boldsymbol{\mu}_k^*$ is the true mean parameter, $\mathcal{V}_k := \{(i,j) : i \neq j, \omega_{kij}^* \neq 0\}$ where $\mathbf{\Omega}_k^*$ is the true precision matrix and $\mathcal{S}_1 = \bigcup_{k=1}^K \mathcal{U}_k$, $\mathcal{S}_2 = \bigcup_{k=1}^K \mathcal{V}_k$. Define $\Xi \subseteq \mathbb{R}^{K(p^2+p)}$ as some non-empty convex set of parameters. Denote the support space $\mathcal{M}$ as

$$\begin{aligned} \mathcal{M} \;:=\; \Big\{ \boldsymbol{V} \in \Xi \;\Big|\; & \mu_{ki} = 0 \text{ for all } i \notin \mathcal{S}_1, \\ & \omega_{kij} = 0 \text{ for all pairs } (i,j) \notin \mathcal{S}_2, k = 1 \ldots, K \Big\}, \end{aligned} \tag{15}$$

where $\boldsymbol{V}$ follows the same definition style used for $\mathbf{\Theta}$ in (3). Denote the sparsity parameters:

$$\begin{aligned} s &:= \#\{(i,j) : \omega_{kij}^* \neq 0, i, j = 1 \ldots p, i \neq j, k = 1, \ldots, K\}, \\ d &:= \#\{i : \mu_{ik}^* \neq 0, i = 1, \ldots, p, k = 1, \ldots, K\}. \end{aligned}$$

3.1 Population-Based Analysis

We define a corresponding population version of Q_n in (9) as

$$Q(\mathbf{\Theta}' | \mathbf{\Theta}) := \mathbb{E}\left[\sum_{k=1}^{K} L_{\mathbf{\Theta},k}(\boldsymbol{X}) [\log \pi_k' + \log f_k(\boldsymbol{X}; \mathbf{\Theta}_k')] \right]. \tag{16}$$

Without loss of generality, we assume the true prior probability $\pi_k^* = 1/K$ for each $k = 1, \ldots, K$. Recall that the update of weights in (11) is independent of the updates of other parameters. Consequently, according to (2), maximizing $Q(\mathbf{\Theta}' | \mathbf{\Theta})$ over $(\boldsymbol{\mu}_k', \mathbf{\Omega}_k')$ is equivalent to maximizing

$$\sum_{k=1}^{K} \mathbb{E}\left[L_{\mathbf{\Theta},k}(\boldsymbol{X}) \left\{ \frac{1}{2} \log\det(\mathbf{\Omega}_k') - \frac{1}{2} (\boldsymbol{X} - \boldsymbol{\mu}_k')^\top \mathbf{\Omega}_k' (\boldsymbol{X} - \boldsymbol{\mu}_k') \right\} \right]. \tag{17}$$

Clearly, the update of $(\boldsymbol{\mu}_l^{'}, \boldsymbol{\Omega}_l^{'})$ is independent of the update of $(\boldsymbol{\mu}_t^{'}, \boldsymbol{\Omega}_t^{'})$ for any $t \neq l$. This enables us to characterize the update of each pair of parameters separately. For any $k = 1, \ldots, K$, define

$$M_{\boldsymbol{\mu}_k^{'}}(\boldsymbol{\Omega}_k^{'}) := \arg\max_{\boldsymbol{\mu}_k^{'}} Q(\boldsymbol{\Theta}^{'}|\boldsymbol{\Theta}) \text{ and } M_{\boldsymbol{\Omega}_k^{'}}(\boldsymbol{\mu}_k^{'}) := \arg\max_{\boldsymbol{\Omega}_k^{'}} Q(\boldsymbol{\Theta}^{'}|\boldsymbol{\Theta}).$$

We show in Lemma 5 that the population update of $\boldsymbol{\mu}_k^{'}$ is independent of $\boldsymbol{\Omega}_k^{'}$, while the population update of $\boldsymbol{\Omega}_k^{'}$ is a function of $\boldsymbol{\mu}_k^{'}$.

Lemma 5 *For any $k = 1, \ldots, K$, we have*

$$M_{\boldsymbol{\mu}_k^{'}}(\boldsymbol{\Omega}_k^{'}) = \big[\mathbb{E}[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})]\big]^{-1}\mathbb{E}[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})\boldsymbol{X}], \tag{18}$$

$$M_{\boldsymbol{\Omega}_k^{'}}(\boldsymbol{\mu}_k^{'}) = \mathbb{E}[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})]\left[\mathbb{E}[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})(\boldsymbol{X}-\boldsymbol{\mu}_k^{'})(\boldsymbol{X}-\boldsymbol{\mu}_k^{'})^{\top}]\right]^{-1}. \tag{19}$$

The difficulty of simultaneous clustering and estimation can be characterized by the following *sufficiently separable condition*. Define $\mathcal{B}_\alpha(\boldsymbol{\Theta}^*) := \big\{\boldsymbol{\Theta} \in \Xi : \big\|\boldsymbol{\Theta} - \boldsymbol{\Theta}^*\big\|_2 \leq \alpha\big\}$.

Condition 6 (*Sufficiently Separable Condition*) *Denote $W = \max_j W_j$, $W^{'} = \max_j W_j^{'}$, $W^{''} = \max_j W_j^{''}$ with $W_j, W_j^{'}, W_j^{''}$ defined in* (S.4), (S.7) *and* (S.8), *respectively. We assume K clusters are sufficiently separable such that given an appropriately small parameter $\gamma > 0$, it holds a.s.*

$$L_{\boldsymbol{\Theta},k}(\boldsymbol{X}) \cdot L_{\boldsymbol{\Theta},j}(\boldsymbol{X}) \leq \frac{\gamma}{24(K-1)\sqrt{\max\{W, W^{'}, W^{''}\}}}, \tag{20}$$

for each pair $\{(j,k), j,k \in [K], j \neq k\}$ and any $\boldsymbol{\Theta} \in \mathcal{B}_\alpha(\boldsymbol{\Theta}^*)$.

Condition 6 requires that K clusters are sufficiently separable in the sense that $\boldsymbol{X}$ belongs to the k-th cluster with probability either close to zero or close to one such that $L_{\boldsymbol{\Theta},k}(\boldsymbol{X}) \cdot L_{\boldsymbol{\Theta},j}(\boldsymbol{X})$ is close to zero. In the special case that $K = 2$ and $\boldsymbol{\Omega}_1^* = \boldsymbol{\Omega}_2^* = 1_p$, Balakrishnan et al. (2016) requires $\|\boldsymbol{\mu}_1^* - \boldsymbol{\mu}_2^*\|_2$ is sufficiently large. Our Condition 6 extends it to general K and general precision matrices. Note that the condition (20) is related with the number of clusters K. As K grows, the clustering problem gets harder and hence a stronger sufficiently separable condition is needed.

The next lemma guarantees that the curvature of $Q(\cdot|\boldsymbol{\Theta})$ is similar to that of $Q(\cdot|\boldsymbol{\Theta}^*)$ when $\boldsymbol{\Theta}$ is close to $\boldsymbol{\Theta}^*$, which is a key ingredient in our population-based analysis.

Lemma 7 (*Gradient Stability*) *Under Condition 6, the function $\{Q(\cdot|\boldsymbol{\Theta}), \boldsymbol{\Theta} \in \Xi\}$ satisfies,*

$$\big\|\nabla Q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) - \nabla Q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}^*)\big\|_2 \leq \tau \cdot \big\|\boldsymbol{\Theta} - \boldsymbol{\Theta}^*\big\|_2, \tag{21}$$

with parameter $\tau \leq \gamma/12$ for any $\boldsymbol{\Theta} \in \mathcal{B}_\alpha(\boldsymbol{\Theta}^*)$. The gradient $\nabla Q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta})$ is taken with respect to the first variable of $Q(\cdot|\cdot)$.

3.2 Sample-Based Analysis

In this section, we analyze the sample-base function Q_n, defined as the objective function in (9). The statistical error comes from the approximation by using sample-base function Q_n to population-base function Q. We need one regularity condition to ensure that Q_n is strongly concave in a specific Euclidean ball.

Condition 8 *There exist some positive constants β_1, β_2 such that $0 < \beta_1 < \min_{k\in[K]} \sigma_{\min}(\boldsymbol{\Omega}_k^*) < \max_{k\in[K]} \sigma_{\max}(\boldsymbol{\Omega}_k^*) < \beta_2$.*

Lemma 9 verifies the restricted strong concavity condition of Q_n. Note that (22) corresponds to the restricted eigenvalue condition in sparse linear regression (Negahban et al., 2012).

Lemma 9 (*Restricted Strong Concavity*) *Suppose that Condition 8 holds. Then for any $\boldsymbol{\Theta} \in \mathcal{B}_\alpha(\boldsymbol{\Theta}^*)$, with probability at least $1-\delta$, each $\boldsymbol{\Theta}' \in \mathbb{C} := \{\boldsymbol{\Theta}' \mid \|\boldsymbol{\Theta}' - \boldsymbol{\Theta}^*\|_2 \leq 2\alpha\}$ satisfies*

$$Q_n(\boldsymbol{\Theta}'|\boldsymbol{\Theta}) - Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) - \Big\langle \nabla Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}), \boldsymbol{\Theta}' - \boldsymbol{\Theta}^* \Big\rangle \leq -\frac{\gamma}{2}\Big\|\boldsymbol{\Theta}' - \boldsymbol{\Theta}^*\Big\|_2^2, \tag{22}$$

with sufficiently large n, where $\gamma = c \cdot \min\{\beta_1, 0.5(\beta_2 + 2\alpha)^{-2}\}$ is the strong concavity parameter for some constant c.

Define $\mathcal{P}(\boldsymbol{\Theta}) = M_1\mathcal{P}_1(\boldsymbol{\Theta}) + M_2\mathcal{P}_2(\boldsymbol{\Theta}) + M_3\mathcal{P}_3(\boldsymbol{\Theta})$ for some positive constants M_1, M_2, M_3. Let $\mathcal{P}^*$ be the dual norm of $\mathcal{P}$, which is defined as $\mathcal{P}^*(\boldsymbol{\Theta}) = \sup_{\mathcal{P}(\boldsymbol{\Theta}')\leq 1}\langle \boldsymbol{\Theta}', \boldsymbol{\Theta}\rangle$. For simplicity, write $\|\cdot\|_{\mathcal{P}^*} = \mathcal{P}^*(\cdot)$.

Condition 10 *For any fixed $\boldsymbol{\Theta} \in \mathcal{B}_\alpha(\boldsymbol{\Theta}^*)$, with probability at least $1-\delta_1$,*

$$\Big\|\nabla Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) - \nabla Q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta})\Big\|_{\mathcal{P}^*} \leq \varepsilon_1, \tag{23}$$

and with probability at least $1-\delta_2$, we have

$$\Big\|\Big[\nabla Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) - \nabla Q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta})\Big]_{\mathcal{G}}\Big\|_2 \leq \varepsilon_2, \tag{24}$$

where $\mathcal{G}$ is the diagonal index set defined in (14). Here ε_1 and ε_2 are functions of $n, p, K, \delta_1, \delta_2$.

Intuitively, ε_1 and ε_2 quantify the difference between the population-based and sample-based conditional maximization step. Note that $\mathcal{P}$ does not penalize diagonal elements of each precision matrix, thus

$$\Big\|\nabla Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) - \nabla Q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta})\Big\|_{\mathcal{P}^*} = \Big\|\Big[\nabla Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) - \nabla Q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta})\Big]_{\mathcal{G}^c}\Big\|_{\mathcal{P}^*}.$$

Our analysis makes use of the property of dual norm to bridge the SCAN penalty term and the targeted error term in L_2 norm. Note that our SCAN penalty does not penalize diagonal terms of precision matrices, and hence it can be treated as a norm only if it is applied to the parameter $\boldsymbol{\Theta}$ without diagonal terms of precision matrices. Otherwise, it is a semi-norm. For this purpose, we separate all the diagonal terms from $\boldsymbol{\Theta}$. Therefore, our statistical error is split by two parts: one from the sparse estimate of cluster means and non-diagonal terms

in precision matrices, and another from the estimate of diagonal terms of precision matrices. In Lemma S.1, ε_1 and ε_2 will be specifically calculated for our proposed SCAN penalty. In the high dimensional ECM algorithm, there is no explicit form for the CM-step update due to the existence of the penalty term. This is a crucial difference from the low-dimensional EM algorithm in Balakrishnan et al. (2016). Fortunately, the decomposability of SCAN penalty enables us to quantify statistical errors by evaluating the gradient of Q-function.

3.3 Statistical Error versus Optimization Error

In this section, we provide the final theoretical guarantee for the high-dimensional ECM algorithm by combining the population and sample-based analysis.

Definition 11 (*Support Space Compatibility Constant*) *For the support subspace $\mathcal{M} \subseteq \mathbb{R}^{K(p^2+p)}$ defined in (15), we define*

$$\nu(\mathcal{M}) = \sup_{\mathbf{\Theta} \in \mathcal{M} \setminus \{0\}} \frac{\mathcal{P}(\mathbf{\Theta})}{\|\mathbf{\Theta}\|_2}. \tag{25}$$

Remark 12 *The support space compatibility constant $\nu(M)$ is a variant of subspace compatibility constant originally proposed by Negahban et al. (2012) and Wainwright (2014). Actually, $\nu(M)$ can be interpreted as a notion of intrinsic dimensionality of M. In order to bound the statistical error, we need some measures for the complexity of parameter $\mathbf{\Theta}$ reflected by the penalty term. One possible way is to specify a model subspace $\mathcal{M}$ and require $\mathbf{\Theta}$ lie in the space. By choosing the support space $\mathcal{M}$ of parameter of interest $\mathbf{\Theta}$, the support space compatibility constant $\nu(\mathcal{M})$ can measure the complexity of $\mathbf{\Theta}$ relative to the penalty term $\mathcal{P}$ and square norm. The larger $\nu(\mathcal{M})$ is, the more samples are needed to guarantee statistical consistency. For examples, if the penalty $\mathcal{P}$ is L_1 penalty with s-sparse coordinate support space $\mathcal{M}'$, then we have $\nu(\mathcal{M}') = \sqrt{s}$. In the context of group lasso penalty, we have $\nu(\mathcal{M}') = \sqrt{|S|}$, where S is the index set of active groups. For our SCAN penalty, $\nu(\mathcal{M})$ is specifically calculated by $M_1\sqrt{Kd} + (M_2\sqrt{K} + M_3)\sqrt{s}$, where d, s are the common sparsity parameters for single cluster means and precision matrices accordingly and M_1, M_2, M_3 are some absolute constants.*

We first provide a general theory that applies to any decomposable penalty, such as the group lasso penalty in Sun et al. (2012) and fused graphical lasso penalty in Danaher et al. (2014). The theoretical result of our SCAN penalty will be discussed in Corollary 18.

Theorem 13 *Suppose Conditions 6, 8, 10 hold and $\mathbf{\Theta}^*$ lies in the interior of Ξ. Let $\kappa = 6\tau/\gamma$, where τ, γ are calculated in Lemma 7 and Lemma 9. Consider our SCAN algorithm in Table 1 with initialization $\mathbf{\Theta}^{(0)}$ falling into a ball $\mathcal{B}_\alpha(\mathbf{\Theta}^*)$ for some constant radius $\alpha > 0$ and assume the tuning parameters satisfy $\lambda_1 = M_1\lambda_n^{(t)}$, $\lambda_2 = M_2\lambda_n^{(t)}$, $\lambda_3 = M_3\lambda_n^{(t)}$, and*

$$\lambda_n^{(t)} = \varepsilon + \kappa\frac{\gamma}{\nu(\mathcal{M})}\left\|\mathbf{\Theta}^{(t-1)} - \mathbf{\Theta}^*\right\|_2. \tag{26}$$

If the sample size n is large enough such that $\varepsilon \le (1-\kappa)\gamma\alpha/(6\nu(\mathcal{M}))$, then $\mathbf{\Theta}^{(t)}$ satisfies, with probability at least $1 - t\delta'$,

$$\left\|\mathbf{\Theta}^{(t)} - \mathbf{\Theta}^*\right\|_2 \le \underbrace{\frac{6\nu(\mathcal{M})}{(1-\kappa)\gamma}\varepsilon}_{\text{Statistical Error(SE)}} + \underbrace{\kappa^t\left\|\mathbf{\Theta}^{(0)} - \mathbf{\Theta}^*\right\|_2}_{\text{Optimiation Error(OE)}}, \tag{27}$$

where $\delta' = \delta + \delta_1 + \delta_2$ with δ, δ_1, δ_2 defined in Lemma 9 and Condition 10 and $\varepsilon = \varepsilon_1 + \varepsilon_2/\nu(\mathcal{M})$.

The above theoretical result suggests that the estimation error in each iteration consists *statistical error* and *optimization error*. From the definition of τ in Lemma 7, κ is less than 0.5 so that it is a contractive parameter. With a relatively good initialization, even though ECM algorithm may be trapped into a local optima after enough iterations, it can be guaranteed to be within a small neighborhood of the truth, in the sense of statistical accuracy. In addition, with a proper choice of δ', the final probability $1 - t\delta'$ will converge to 1; see Corollary 18 for details.

Remark 14 *To our limited knowledge, there is no existing literature to guarantee the global convergence of ECM algorithm in a general case. Compromisingly, we have to require some constraints on the initial value. In our framework, the only requirement for the initial value is to fall into a ball with constant radius to the truth. Such a condition has also been imposed in EM algorithms (Balakrishnan et al., 2016; Wang et al., 2015b; Yi and Caramanis, 2015) and can be fulfilled by some spectral-based initializations (Zhang et al., 2014).*

Remark 15 *In Theorem 13, we introduce an iterative turning procedure (26) which appeared in high dimensional regularized M-estimation (Negahban et al., 2012), and was also applied in Yi and Caramanis (2015) to facilitate their theoretical analysis.*

The error bound in (27) measures the estimation error in each iteration. Here, optimization error decays geometrically with the iteration number t, while the statistical error remains the same when t grows. Therefore, this enables us to provide a meaningful choice of the maximal number of iterations T beyond which the optimization error is dominated by the statistical error such that the whole error bound is in the same order of the statistical error.

In the following corollary, taking the SCAN penalty as an example, we provide a closed form of the maximal number of iterations T and also an explicit form of the estimation error.

Condition 16 *The largest element of cluster means and precision matrices are both bounded, that is, for some positive constants c_1 and c_2,*

$$\|\boldsymbol{\mu}^*\|_\infty := \max_{k\in[K]}\|\boldsymbol{\mu}_k^*\|_\infty < c_1 \text{ and } \|\mathbf{\Omega}^*\|_{\max} := \max_{k\in[K]}\|\mathbf{\Omega}_k^*\|_{\max} < c_2.$$

Condition 17 *Suppose that the number of clusters K satisfies $K^2 = o(p(\log n)^{-1})$.*

Corollary 18 *Suppose Conditions 6, 8, 16 and 17 hold. If sample size n is sufficiently large such that*

$$n \geq \left(\frac{6(CK\|\mathbf{\Omega}^*\|_\infty + C'K^{1.5})(\sqrt{Kd} + \sqrt{Ks} + \sqrt{K}) + C''K^{1.5}\sqrt{p}}{(1-\kappa)\gamma\alpha} \right)^2 \log p,$$

and the iteration step t is large enough such that

$$t \geq T = \log_{1/\kappa} \frac{\left\|\mathbf{\Theta}^{(0)} - \mathbf{\Theta}^*\right\|_2}{\varphi(n,p,K)},$$

where $\varphi(n,p,K) = 6\widetilde{C}((1-\kappa)\gamma)^{-1}\|\mathbf{\Omega}^*\|_\infty(\sqrt{Kd}+\sqrt{Ks+p})\sqrt{K^3\log p/n}$ for some positive constant $\widetilde{C}$, the optimization error in (27) is dominated by the statistical error, and

$$\sum_{k=1}^{K}\left(\left\|\boldsymbol{\mu}_k^{(T)} - \boldsymbol{\mu}_k^*\right\|_2 + \left\|\mathbf{\Omega}_k^{(T)} - \mathbf{\Omega}_k^*\right\|_F\right) \leq \frac{12\widetilde{C}}{(1-\kappa)\gamma}\left(\underbrace{\|\mathbf{\Omega}^*\|_\infty\sqrt{\frac{K^5 d\log p}{n}}}_{\text{Cluster means error}} + \underbrace{\|\mathbf{\Omega}^*\|_\infty\sqrt{\frac{K^3(Ks+p)\log p}{n}}}_{\text{Precision matrices error}}\right),$$

with probability converging to 1.

Remark 19 *If K is fixed, the above upper bound reduces to*

$$\sum_{k=1}^{K}\left(\left\|\boldsymbol{\mu}_k^{(T)} - \boldsymbol{\mu}_k^*\right\|_2 + \left\|\mathbf{\Omega}_k^{(T)} - \mathbf{\Omega}_k^*\right\|_F\right) \lesssim \left(\underbrace{\|\mathbf{\Omega}^*\|_\infty\sqrt{\frac{d\log p}{n}}}_{\text{Cluster means error}} + \underbrace{\|\mathbf{\Omega}^*\|_\infty\sqrt{\frac{(s+p)\log p}{n}}}_{\text{Precision matrices error}}\right). \tag{28}$$

Consider the class of precision matrix $\mathcal{Q} := \{\mathbf{\Omega} : \mathbf{\Omega} \succ 0, \|\mathbf{\Omega}\|_\infty \leq C_\mathcal{Q}\}$ as in Cai et al. (2016b). When $C_\mathcal{Q}$ does not depend on n, p, our rate $\sqrt{(s+p)\log p/n}$ in (28) is minimax optimal for estimating s-sparse precision matrix under Frobenius norm (see Theorem 7 in Cai et al. (2016b)). The same rate has also been obtained in Saegusa and Shojaie (2016) for multiple precision matrix estimation when the true cluster structure is assumed to be given in advance. Moreover, our cluster mean error rate $\sqrt{d\log p/n}$ is minimax optimal for estimating d-sparse cluster means; see Wang et al. (2015b). In short, Corollary 18 indicates that our procedure is able to achieve optimal statistical rates for both cluster means and multiple precision matrices even when the true cluster structure is unknown.

Remark 20 *As a by-product, we establish the variable selection consistency of $\mathbf{\Omega}_k^{(T)}$, which ensures that our precision matrix estimator can asymptotically identify true connected links. Assume $\|\mathbf{\Omega}_k^*\|_\infty$ is bounded and the minimal signal in the true precision matrix satisfies $\omega_{\min} := \min_{(i,j)\in\mathcal{V}_k, k=1,\ldots,K} w_{kij}^* > 2r_n$, where $r_n = (\sqrt{K^5 d} + \sqrt{K^3(Ks+p)})\sqrt{\log p/n}$. The latter condition is weaker than that assumed in Guo et al. (2011), where they require a constant lower bound of $\omega_{\min}$. To ensure the model selection consistency, we threshold the precision matrix estimator $\mathbf{\Omega}_k^{(T)}$ such that $\widetilde{\omega}_{kij} = \omega_{kij}^{(T)} 1\{|\omega_{kij}^{(T)}| > r_n\}$ as in Bickel and Levina (2008) and Lee and Liu (2015). See Theorem S.2 in the online supplementary for some results on the selection consistency result.*

4. Numerical Study

In this section, we discuss an efficient tuning parameter selection procedure and demonstrate the superior numerical performance of our method. We compare our algorithm with three clustering and graphical model estimation methods:

- Standard K-means clustering (MacQueen, 1967).
- Algorithm in Zhou et al. (2009) which applies graphical lasso for each precision matrix estimation.
- A two-stage approach which first uses K-means clustering to obtain the clusters and then applies joint graphical lasso (Danaher et al., 2014) to estimate precision matrices.

For a fair comparison, we assume the number of clusters K is given in all methods.

4.1 Selection of Tuning Parameters

In our simultaneous clustering and graph estimation formulation, three tuning parameters $\Lambda := \{\lambda_1, \lambda_2, \lambda_3\}$ need to be appropriately determined so that both the clustering and network estimation performance can be optimized. In our framework, the tuning parameters are selected through the following adaptive BIC-type selection criterion. For a set of tuning parameters $\Lambda := \{\lambda_1, \lambda_2, \lambda_3\}$, the adaptive BIC criterion is defined as

$$\mathrm{BIC}(\Lambda) = -2\log\widehat{L}(\Lambda) + \log(n)\mathrm{df}_\Lambda(\boldsymbol{\mu}) + 2\mathrm{df}_\Lambda(\mathbf{\Omega}), \tag{29}$$

where $\widehat{L}(\Lambda)$ is the sample likelihood function and $\{\mathrm{df}_\Lambda(\boldsymbol{\mu}), \mathrm{df}_\Lambda(\mathbf{\Omega})\}$ is the degrees of freedom of the model. Here, $\{\mathrm{df}_\Lambda(\boldsymbol{\mu}), \mathrm{df}_\Lambda(\mathbf{\Omega})\}$ can be approximated by the size of selected variables in the final estimator. Therefore, according to the Gaussian mixture model assumption, the adaptive BIC criterion in (29) can be computed as

$$-2\sum_{i=1}^{n}\log\left(\sum_{k=1}^{K}\widehat{\pi}_k f_k\left(\boldsymbol{x}_i; \widehat{\boldsymbol{\mu}}_k, (\widehat{\mathbf{\Omega}}_k)^{-1}\right)\right) + \sum_{k=1}^{K}\{\log n \cdot s_{1k} + 2s_{2k}\},$$

where $s_{1k} = \mathrm{Card}\{i : \widehat{\mu}_{ki} \neq 0\}$, $s_{2k} = \mathrm{Card}\{(i,j) : \widehat{\Omega}_{kij} \neq 0, 1 \leq i < j \leq p\}$ and $\widehat{\pi}_k, \widehat{\boldsymbol{\mu}}_k, \widehat{\mathbf{\Omega}}_k$ are final updates from Algorithm 1. We choose a smaller weight for the degrees of freedom of precision matrices as suggested in Danaher et al. (2014). The mixing weight π is not counted into the degrees of freedom since it only contributes a constant factor.

In our experiment, we choose the optimal set of parameters minimizing the BIC value in (29). In the high-dimensional scenario where p is very large, calculation of BIC over a

grid search for all $\lambda_1, \lambda_2, \lambda_3$ may be computationally expensive. Following Danaher et al. (2014), we suggest a line search over λ_1, λ_2 and λ_3. In detail, we fix λ_2 and λ_3 at their median value of the given range and conduct a grid search over λ_1. Then with tuned λ_1 and median value of λ_3, we conduct a grid search over λ_2. The line search for λ_3 is the same. In our simulations, we choose the tuning range $10^{-2+2t/15}$ with $t = 0, 1, \ldots, 15$ for all $\lambda_1, \lambda_2, \lambda_3$.

4.2 Illustration

In this subsection, we demonstrate the importance of simultaneous clustering and estimation in improving both the clustering performance and the estimation accuracy of multiple precision matrices.

The simulated data consists of $n = 1000$ observations from 2 clusters, and among them 500 observations are from $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$ and the rest 500 observations are from $\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$ with $\boldsymbol{\mu}_1 = (0, 1)^\top$, $\boldsymbol{\mu}_2 = (0, -1)^\top$, and

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}.$$

The standard K-means algorithm treats the data space as isotropic (distances unchanged by translations and rotations) (Raykov et al., 2016). This means that data points in each cluster are modeled as lying within a sphere around the cluster centroid. A sphere has the same radius in each dimension. However, the non-diagonal covariance matrix in the mixture model makes the cluster structure highly non-spherical. Thus, the K-means algorithm is expected to produce an unsatisfactory clustering result. This is illustrated in Figure 2 where K-means clustering clearly obtains wrong clusters. On the other hand, by incorporating the precision matrix estimation into clustering, our method is able to identify two correct clusters.

Figure 2: The first plot represents the true clusters shown in red and black in the example of Section 4.2. The middle and right plots show the clusters obtained from the standard K-means clustering (Kmeans) and our SCAN method.

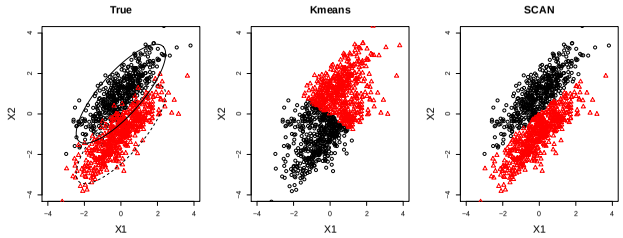
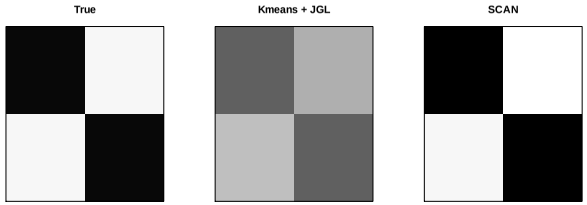


Figure 3 illustrates the estimation performance of precision matrices based on the clusters estimated from the K-means clustering and our method. Clearly, our SCAN method

delivers an estimator with improved accuracy when compared to the two stage method which applies joint graphical lasso (JGL) to the clusters obtained from the K-means clustering. This suggests that an accurate clustering is critical for the estimation performance of heterogeneous graphical models.

Figure 3: The true precision matrix and the estimated precision matrices from the two stage method (Kmeans + JGL) and our SCAN method in the example of Section 4.2.



4.3 Effect of Sample Size and Dimension

We investigate the effect of sample size and dimension in terms of the estimation error and computational time. First, we empirically demonstrate the derived upper bound (28) for the estimation error by drawing the error pattern of our precision matrix estimator against sample size and dimension. The setting is the same as Section 4.2 except that we consider a tri-diagonal convariance structure. The results are summarized in Figure 4. In the first plot, we fix the dimension to be 10 and vary the sample size from 400 to 2000. In the second plot, we fix the sample size to be 5000 and vary the dimension from 5 to 50. The box plot refers to the the actual numerical values of precision matrix estimation errors, and the red dot is the theoretical error rate in each scenario. These results demonstrate that the empirical errors match very well with the theoretical error bound.

Second, we compare the average running time of our SCAN algorithm with varying sample sizes and dimensions. Figure 5 shows that our algorithm scales linearly with the sample size and roughly linearly with the dimension. This illustrates the efficiency and scalability of our proposed algorithm.

4.4 Simulations

In this subsection, we conduct extensive simulation studies to evaluate the performance of our algorithm. To assess the clustering performance of various methods, we compute the following clustering error (CE) which calculates the distance between an estimated clustering assignment $\widehat{\psi}$ and the true assignment ψ of the sample data $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ (Wang, 2010; Sun et al., 2012),

$$\mathrm{CE}(\widehat{\psi}, \psi) := \binom{n}{2}^{-1} \Big| \{(i, j) : 1(\widehat{\psi}(\boldsymbol{X}_i) = \widehat{\psi}(\boldsymbol{X}_j)) \neq 1(\psi(\boldsymbol{X}_i) = \psi(\boldsymbol{X}_j)); i < j\} \Big|,$$

Figure 4: Comparison of the numerical error and the theoretical error rates of our SCAN method. The left panel displays the precision matrix estimation error with varying sample sizes. The right panel displays the precision matrix estimation error with varying dimensions.

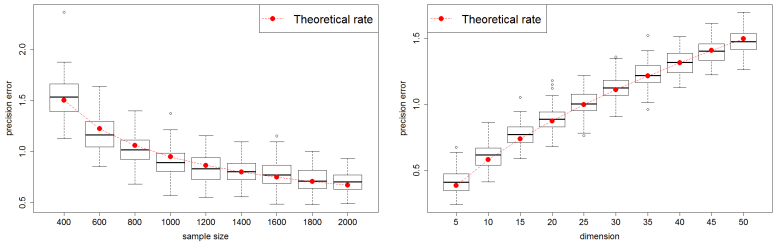
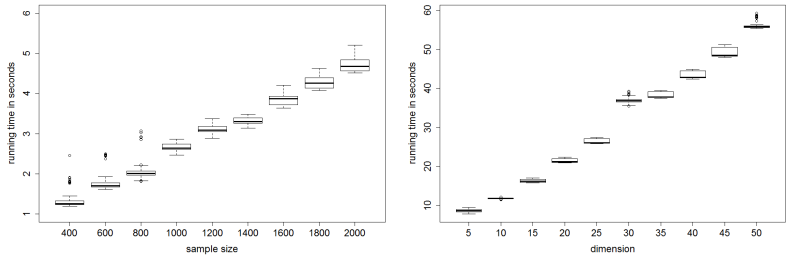


Figure 5: Running time of our algorithm. The left panel is the running time with varying sample sizes and fixed dimension $p = 10$. The right panel is the running time with varying dimensions and fixed sample size $n = 5000$.



where $|\mathcal{A}|$ is the cardinality of set $\mathcal{A}$. To measure the estimation quality, we calculate the precision matrix error (PME) and cluster mean error (CME)

$$\text{PME} := \frac{1}{K}\sum_{k=1}^{K}\left\|\widehat{\boldsymbol{\Omega}}^{(k)} - \boldsymbol{\Omega}^{(k)}\right\|_F; \ \text{CME} := \frac{1}{K}\sum_{k=1}^{K}\left\|\widehat{\boldsymbol{\mu}}^{(k)} - \boldsymbol{\mu}^{(k)}\right\|_2.$$

Finally, to compare the variable selection performance, we compute the true positive rate (TPR, percentage of true edges selected) and the false positive rate (FPR, percentage of false edges selected)

$$\text{TPR} := \frac{1}{K}\sum_{k=1}^{K}\frac{\sum_{i<j}1(\omega_{kij}\neq 0, \widehat{\omega}_{kij}\neq 0)}{\sum_{i<j}1(\omega_{kij}\neq 0)},$$
$$\text{FPR} := \frac{1}{K}\sum_{k=1}^{K}\frac{\sum_{i<j}1(\omega_{kij}= 0, \widehat{\omega}_{kij}\neq 0)}{\sum_{i<j}1(\omega_{kij}= 0)}.$$

In the simulation, a three-class problem is considered. We illustrate three different types of network structures. In the first scenario, the network is assumed to have some regular structures. We generate a 5-block tridiagonal precision matrix with p features for the precision matrix. To allow the similarity of precision matrices across clusters, we set the off-diagonal entry of $\Omega_1, \Omega_2, \Omega_3$ as η, 0.99η, and 1.01η, respectively. The diagonal entries of Ω_1, Ω_2, and Ω_3 are all 1.

In the second and third scenarios, followed by Danaher et al. (2014), we simulate each network consisting of disjointed modules since many large networks in the real life exhibit a modular structure comprised of many disjointed or loosely connected components of relatively small size (Peng et al., 2009). Thus, each of three networks is generated with p features, which has ten equally sized unconnected subnetworks. Among the ten subnetworks, eight have the same structure and edge values across all the three classes, one remains the same only for the first two classes and the last one appears only for the first class. For the cluster structure of subnetwork, we consider two scenarios: power-law network and chain network, which are generated using the algorithm in Peng et al. (2009) and Fan et al. (2009). The detail construction is described as below.

Power-law network. Given an undirected network structure above, the initial ten-block precision matrix $(w^1_{ij})_{p\times p}$ is generated by

$$w^1_{ij} = \begin{cases} 1 & i \neq j; \\ 0 & i \neq j, \text{ no edge}; \\ \text{Unif}([-0.4,-0.1]\cup[0.1,0.4]) & i \neq j, \text{ edge exits}; \end{cases}$$

To ensure positive definiteness and symmetry, we divide each off-diagonal entry by 0.9 times the sum of the absolute values of off-diagonal entries in its row and average this rescaled matrix with its transpose. Denote the final transformed matrix by $\boldsymbol{A}$. The covariance matrix corresponding to the first class is created by

$$\boldsymbol{\Sigma}_{1ij} = d_{ij}\frac{\boldsymbol{A}^{-1}_{ij}}{\sqrt{\boldsymbol{A}^{-1}_{ii}\boldsymbol{A}^{-1}_{jj}}} \tag{30}$$

where $d_{ij} = 0.9$ for non-diagonal entry and $d_{ij} = 1$ for diagonal entry. For the covariance matrix corresponding to the second class, we create $\boldsymbol{\Sigma}_2$ be identical to $\boldsymbol{\Sigma}_1$ but reset one of ten block matrix to the identity matrix. Similarly, we reset one additional block matrix for $\boldsymbol{\Sigma}_3$.

Chain network. In the scenario, each of ten blocks of the first covariance matrix $\boldsymbol{\Sigma}_1$ is constructed in the following way. The ij-th element of each block has the form

$\sigma_{ij} = \exp(-a|s_i - s_j|)$, where $s_1 < s_2 < \cdots < s_{p/10}$ for some $a > 0$. This is related to the autoregressive process of order one. In our case, we choose $a = 1$ and $s_i - s_{i-1} \sim \text{Unif}(0.5, 1)$ for $i = 2, \ldots, p/10$. Similarly, we create $\mathbf{\Sigma}_2$ be identical to $\mathbf{\Sigma}_1$ but reset one of ten block matrix to the identity matrix and reset one additional block matrix for $\mathbf{\Sigma}_3$.

After the networks are constructed, the samples are generated as follows. First, the cluster membership Y_i's are uniformly sampled from $\{1, 2, 3\}$. Given the cluster label, we generate each sample $\boldsymbol{X}_i \sim \mathcal{N}(\boldsymbol{\mu}(Y_i), \mathbf{\Sigma}(Y_i))$. Here, the cluster mean $\boldsymbol{\mu}(Y_i)$ is sparse, where its first 10 variables are of the form

$$(\mu \mathbf{1}_5^\top, -\mu \mathbf{1}_5^\top)^\top 1(Y_i = 1) + \mu \mathbf{1}_{10} 1(Y_i = 2) + (-\mu \mathbf{1}_5^\top, -\mu \mathbf{1}_5^\top)^\top 1(Y_i = 3),$$

with $\mathbf{1}_5$ being a 5-dimensional vector of all ones, and its last $p - 10$ variables are zeros. For the first scenario, we consider 3 simulation models with varying choices of μ and η:

- Model 1: $\mu = 0.8$ and $\eta = 0.3$,
- Model 2: $\mu = 1$ and $\eta = 0.3$,
- Model 3: $\mu = 1$ and $\eta = 0.4$.

Here μ controls the separability of the three clusters with larger μ corresponding to an easier clustering problem, and η represents the similarity level of precision matrices across clusters. For the second and third scenarios, we considered three simulation models with sequential choices of μ:

- Models 4,7: $\mu = 0.7$,
- Models 5,8: $\mu = 0.8$,
- Models 6,9: $\mu = 0.9$.

The number of features p is equal to 100 and sample size is equal to 300. The results are averaged over 50 experiments. The code is written in R and implemented on an Intel Xeon-E5 processor with 64 GB of RAM. The average computation time for SCAN of a single run took one and half minute.

In the experiment, our method selected the tuning parameters via the BIC criterion in Section 4.1. For a fair comparison, we also used the same tuning parameters λ_1, λ_2 in Zhou et al. (2009), and the same λ_2, λ_3 in the joint graphical lasso penalty of the two-stage approach. We repeated the procedure 50 times and reported the averaged clustering errors, estimation errors, and variable selection errors for each method as well as their standard errors. Table 2 is for regular network, Table 3 is for power-law networks and Table 4 is for chain networks. As shown in Table 3 and Table 4, the standard K-means clustering method has the largest clustering error due to a violation of its diagonal covariance matrix assumption. This will result in poor estimation for multiple precision matrices. The method of Zhou et al. (2009) improves the clustering performance of the standard K-means by using a graphical lasso in the precision matrix estimation. However, it obtains a relatively large precision matrix estimation error and very bad false positive rate since it

ignores the similarity across different precision matrices. In contrast, our SCAN algorithm achieves the best clustering accuracy and best precision matrix estimation accuracy for both scenarios. This is due to our simultaneous clustering and estimation strategy as well as the consideration of similarity of precision matrices across clusters. This experiment shows that a satisfactory clustering algorithm is critical to achieve accurate estimations of heterogeneous graphical models, and alternatively good estimation of the graphical model can also improve the clustering performance. This explains the success of our simultaneous method in terms of both clustering and graphical model estimation.

Table 2: Simulation results of regular network. The clustering errors (CE), cluster mean errors (CME), precision matrix errors (PME), true positive rates (TPR) and false positive rates (FPR) of precision matrix estimation of four methods. The minimal clustering error and minimal estimation error in each simulation are shown in bold.

Models	Methods	CE	CME	PME	TPR /FPR
	K-means	$0.166_{0.011}$	$2.256_{0.108}$	NA	NA /NA
Model 1	K-means + JGL	$0.166_{0.011}$	$2.256_{0.108}$	$8.206_{0.090}$	$0.985_{0.001}$ /$0.023_{0.001}$
$\mu = 0.8$	Zhou et al. (2009)	$0.104_{0.007}$	$1.190_{0.052}$	$10.458_{0.0509}$	$0.960_{0.002}$ /$0.107_{0.001}$
$\eta = 0.3$	SCAN	$\mathbf{0.071_{0.007}}$	$\mathbf{1.120_{0.063}}$	$\mathbf{7.620_{0.072}}$	$\mathbf{0.993_{0.001}}$ /$\mathbf{0.022_{0.001}}$
	K-means	$0.210_{0.009}$	$3.428_{0.114}$	NA	NA/NA
Model 2	K-means + JGL	$0.210_{0.009}$	$3.428_{0.114}$	$12.099_{0.317}$	$0.989_{0.001}$ /$0.039_{0.003}$
$\mu = 1$	Zhou et al. (2009)	$0.125_{0.012}$	$1.860_{0.118}$	$12.833_{0.253}$	$0.993_{0.001}$ /$0.119_{0.006}$
$\eta = 0.3$	SCAN	$\mathbf{0.058_{0.012}}$	$\mathbf{1.476_{0.145}}$	$\mathbf{10.301_{0.332}}$	$\mathbf{0.997_{0.001}}$ /$\mathbf{0.036_{0.002}}$
	K-means	$0.021_{0.002}$	$1.289_{0.013}$	NA	NA /NA
Model 3	K-means + JGL	$0.021_{0.002}$	$1.289_{0.013}$	$7.639_{0.061}$	$0.993_{0.001}$ /$0.029_{0.002}$
$\mu = 1$	Zhou et al. (2009)	$0.021_{0.002}$	$0.968_{0.018}$	$10.115_{0.047}$	$0.968_{0.001}$ /$0.106_{0.001}$
$\eta = 0.4$	SCAN	$\mathbf{0.014_{0.001}}$	$\mathbf{0.956_{0.018}}$	$\mathbf{7.614_{0.061}}$	$\mathbf{0.993_{0.001}}$ /$\mathbf{0.029_{0.002}}$

Table 3: Simulation results of power-law network. The clustering errors (CE), cluster mean errors (CME), precision matrix errors (PME), true positive rates (TPR) and false positive rates (FPR) of precision matrix estimation of four methods. The minimal clustering error and minimal estimation error in each simulation are shown in bold.

Models	Methods	CE	CME	PME	TPR /FPR
Model 4 $\mu = 0.7$	K-means	$0.331_{0.007}$	$3.282_{0.047}$	NA	NA /NA
	K-means + JGL	$0.331_{0.007}$	$3.282_{0.047}$	$49.516_{0.159}$	$0.575_{0.002}$ /$0.034_{0.002}$
	Zhou et al. (2009)	$0.311_{0.006}$	$2.494_{0.055}$	$50.945_{0.164}$	$0.578_{0.002}$ /$0.134_{0.002}$
	SCAN	$\mathbf{0.283_{0.008}}$	$\mathbf{2.385_{0.065}}$	$\mathbf{48.845_{0.146}}$	$0.577_{0.003}$ /$0.032_{0.002}$
Model 5 $\mu = 0.8$	K-means	$0.228_{0.010}$	$2.777_{0.111}$	NA	NA/NA
	K-means + JGL	$0.228_{0.010}$	$2.777_{0.111}$	$48.601_{0.132}$	$0.582_{0.002}$ /$0.044_{0.003}$
	Zhou et al. (2009)	$0.186_{0.011}$	$1.837_{0.113}$	$49.289_{0.122}$	$0.584_{0.001}$ /$0.131_{0.001}$
	SCAN	$\mathbf{0.156_{0.012}}$	$\mathbf{1.789_{0.119}}$	$\mathbf{47.729_{0.118}}$	$0.583_{0.002}$ /$0.041_{0.002}$
Model 6 $\mu = 0.9$	K-means	$0.083_{0.010}$	$1.624_{0.120}$	NA	NA /NA
	K-means + JGL	$0.083_{0.010}$	$1.624_{0.120}$	$46.879_{0.093}$	$0.589_{0.002}$ /$0.070_{0.003}$
	Zhou et al. (2009)	$0.050_{0.002}$	$1.003_{0.018}$	$47.503_{0.003}$	$0.591_{0.001}$ /$0.128_{0.001}$
	SCAN	$\mathbf{0.045_{0.002}}$	$\mathbf{1.003_{0.018}}$	$\mathbf{46.356_{0.086}}$	$0.589_{0.001}$ /$0.068_{0.003}$

Table 4: Simulation results of chain network. The clustering errors (CE), cluster mean errors (CME), precision matrix errors (PME), true positive rates (TPR) and false positive rates (FPR) of precision matrix estimation of four methods. The minimal clustering error and minimal estimation error in each simulation are shown in bold.

Models	Methods	CE	CME	PME	TPR /FPR
Model 7 $\mu = 0.7$	K-means	$0.277_{0.005}$	$2.705_{0.070}$	NA	NA /NA
	K-means + JGL	$0.277_{0.005}$	$2.705_{0.070}$	$25.608_{0.183}$	$0.995_{0.000}$ /$0.033_{0.001}$
	Zhou et al. (2009)	$0.267_{0.006}$	$1.815_{0.075}$	$29.341_{0.109}$	$0.991_{0.001}$ /$0.131_{0.002}$
	SCAN	$\mathbf{0.231_{0.007}}$	$\mathbf{1.652_{0.087}}$	$\mathbf{25.110_{0.106}}$	$0.991_{0.001}$ /$0.031_{0.001}$
Model 8 $\mu = 0.8$	K-means	$0.200_{0.008}$	$2.124_{0.098}$	NA	NA/NA
	K-means + JGL	$0.200_{0.008}$	$2.124_{0.098}$	$24.499_{0.127}$	$0.996_{0.000}$ /$0.042_{0.001}$
	Zhou et al. (2009)	$0.168_{0.004}$	$1.055_{0.076}$	$27.494_{0.121}$	$0.995_{0.001}$ /$0.131_{0.001}$
	SCAN	$\mathbf{0.140_{0.004}}$	$\mathbf{1.046_{0.038}}$	$\mathbf{23.804_{0.085}}$	$0.996_{0.000}$ /$0.039_{0.001}$
Model 9 $\mu = 0.9$	K-means	$0.123_{0.005}$	$1.465_{0.040}$	NA	NA /NA
	K-means + JGL	$0.123_{0.005}$	$1.465_{0.040}$	$23.663_{0.097}$	$0.997_{0.000}$ /$0.044_{0.001}$
	Zhou et al. (2009)	$0.116_{0.003}$	$1.031_{0.022}$	$26.476_{0.090}$	$0.996_{0.001}$ /$0.131_{0.001}$
	SCAN	$\mathbf{0.098_{0.003}}$	$\mathbf{1.025_{0.022}}$	$\mathbf{23.425_{0.083}}$	$0.998_{0.000}$ /$0.043_{0.002}$

4.5 Glioblastoma Cancer Data Analysis

In this section, we apply our simultaneous clustering and graphical model estimation method to a Glioblastoma cancer dataset. We aim to cluster the glioblastoma multiforme (GBM) patients and construct the gene regulatory network of each subtype in order to improve our understanding of the GBM disease.

The raw gene expression dataset measures 17814 levels of mRNA expression of 482 GBM patients. Each patient belongs to one of four subgroups of GBM: Classical, Mesenchymal, Neural, and Proneural (Verhaak et al., 2010). Although they are biologically different, these four subtypes share many similarities since they are all GBM diseases. For our analysis, we considered the 840 signature genes established by Verhaak et al. (2010). Following the preprocess procedures in Lee and Liu (2015), we excluded the genes with no subtype information or the genes with missing values. We then applied the sure independence screening analysis (Fan and Lv, 2008) to finally include 50 genes in our analysis. These 50 signature genes are highly distinctive for these four subtypes. In the analysis, we pretended that the subtype information of each patient was unknown and evaluated the clustering accuracy of various clustering methods by comparing the estimated groups with the true subtypes. In all methods, we fixed $K = 4$. Moreover, we set the tuning parameters $\lambda_1 = 0.065, \lambda_2 = 0.238$, and $\lambda_3 = 0.138$ in our SCAN algorithm. For a fair comparison, we also used the same λ_1, λ_2 in Zhou et al. (2009), and the same λ_2, λ_3 in the joint graphical lasso of the two-stage method.

Table 5 reported the clustering errors of all methods as well as the number of informative variables in the corresponding estimated means and precision matrices. The standard K-means clustering has the large clustering error due to its ignorance of the network structure in the precision matrices. Therefore, the consequent joint graphical lasso method of the network reconstruction is less reliable. The method in Zhou et al. (2009) performed even worse. This is because their method estimates each precision matrix individually without borrowing information from each other. In this gene network example, all of the four graphical models share many edges due to the commonality in the GBM diseases. Zhou et al. (2009)'s method may suffer from the small sample size. Our method is able to achieve the best clustering performance due to the procedure of simultaneous clustering and heterogeneous graphical model estimation.

Table 5: The clustering errors and the number of selected features in cluster mean and precision matrix of various methods in the Glioblastoma Cancer Data.

Methods	Clustering Error	$\sum_k \|\widehat{\boldsymbol{\mu}}^{(k)}\|_0$	$\sum_k \|\widehat{\boldsymbol{\Omega}}^{(k)}\|_0$
K-means	0.262	200	NA
Zhou et al. (2009)	0.336	106	1820
K-means + JGL	0.262	200	1360
SCAN	**0.222**	128	1452

To evaluate the ability of reconstructing gene regulatory network of each subtype, we report the four gene networks estimated from our SCAN method in Figure 1. The black lines are links shared in all subtypes, and the color lines are uniquely presented in some subtypes. Clearly, most edges are black lines, which indicates the common structure of all subtypes. For instance, the link between ZNF45 and ZNF134 is significant across all the four subtypes. Those two genes belong to ZNF gene family. They are known to play roles in making

zinc finger proteins, which are regulatory proteins that are functional important to many cellulars. As they play roles in the same biological process, it is reasonable to expect this link is shared by all GBM subtypes. There are two links that shared by three subtypes except neural subtype: TNFRSF1B↔TRPM2, PTPRC↔ TRPM2. One link uniquely appears in Proneural subtype: ACTR1A ↔DWED and one link FBXO3↔HMG20B is uniquely shown in neural subtype. These findings agree with the existing results in Verhaak et al. (2010). It has been shown that the PTPRC is a well-described microglia marker and is highly exposed in the set of murine astrocytic samples which are strongly associated with the Mesenchymal group. In addition, TRPM2 and TNFRSF1B are shown frequently in the GOTERM category of Mesenchymal group but less likely to appear in Neural group. And FBXO3 is only significant in the cell part of neural subtype. Furthermore, ACTR1A is only found in the intracellular non-membrane-bound organelle and protein binding of Proneural subtype in the supplemental material of Verhaak et al. (2010). It would also be of interest to investigate unique gene links that were not discovered in existing literatures for better understanding of GBM diseases.

5. Discussion

In this paper, we propose a new SCAN method for simultaneous clustering and estimation of heterogeneous graphical models with common structures. We describe the theoretical properties of SCAN and we show that the estimation error bound of our SCAN algorithm consists of statistical error and optimization error, which explicitly addresses the trade-off between statistical accuracy and computational complexity. In our experiments, the tuning parameters can be chosen via an efficient BIC-type criterion. For future work, it is of interest to investigate the model selection consistency of these tuning parameters and study the distributed implementation of ECM algorithm based on the work in Wolfe et al. (2008).

APPENDIX

In this section, we provide detailed proofs of key results: Theorem 13 and Corollary 18. The proofs of other lemmas and theorems are deferred to the online supplementary.

Appendix A. Proof of Theorem 13

First we introduce some notation. Recall the definition of support space $\mathcal{M}$ in (15). The orthogonal complement of support space $\mathcal{M}$, namely, is defined as the set

$$\mathcal{M}^{\perp} := \left\{ \boldsymbol{\Theta}' \in \Xi \mid \langle \boldsymbol{V}, \boldsymbol{\Theta}' \rangle = 0 \text{ for all } \boldsymbol{V} \in \mathcal{M} \right\}.$$

The projection operator $\Pi_{\mathcal{M}}(\boldsymbol{\Theta}) : \Xi \to \Xi$ is defined as

$$\Pi_{\mathcal{M}}(\boldsymbol{\Theta}) := \arg\min_{\boldsymbol{V} \in \mathcal{M}} \|\boldsymbol{V} - \boldsymbol{\Theta}\|_2 .$$

To simplify the notation, we frequently use the shorthand $\boldsymbol{\Theta}_{\mathcal{M}} = \Pi_{\mathcal{M}}(\boldsymbol{\Theta})$ and $\boldsymbol{\Theta}_{\mathcal{M}^{\perp}} = \Pi_{\mathcal{M}^{\perp}}(\boldsymbol{\Theta})$.

In order to efficiently solve the high-dimensional regularized problem, we explore some good properties enjoyed by SCAN penalty in Lemma 21 and Lemma 22. Similar properties can be derived by any decomposable penalty.

Lemma 21 *The SCAN penalty $\mathcal{P}$ is convex and decomposable with respect to $(\mathcal{M}, \mathcal{M}^{\perp})$. In detail,*

$$\mathcal{P}(\boldsymbol{\Theta}_1 + \boldsymbol{\Theta}_2) = \mathcal{P}(\boldsymbol{\Theta}_1) + \mathcal{P}(\boldsymbol{\Theta}_2), \text{ for any } \boldsymbol{\Theta}_1 \in \mathcal{M}, \boldsymbol{\Theta}_2 \in \mathcal{M}^{\perp}.$$

The dual norm of SCAN penalty $\mathcal{P}$ is given by

$$\mathcal{P}^*(\boldsymbol{\Theta}) := \max_{i,j,k,i \neq j} \left(M_1 \sqrt{\mu_{kj}^2}, M_2 \sqrt{\omega_{kij}^2}, M_3 \left(\sum_{k=1}^{K} \omega_{kij}^2 \right)^{1/2} \right). \tag{31}$$

Proof of Lemma 21: The convexity of SCAN comes from the convexity of lasso penalty for cluster means and the convexity of group graphical lasso penalty for precision matrices. The decomposability and derivation of dual norm is obvious from the definition. Also see Wainwright (2014). ■

Lemma 22 *For all vectors $\boldsymbol{\Theta}$ belonging to support space $\mathcal{M}$, $\mathcal{P}(\boldsymbol{\Theta}_{\mathcal{M}})$ satisfies the following inequality:*

$$\mathcal{P}(\boldsymbol{\Theta}_{\mathcal{M}}) \leq \nu(\mathcal{M}) \|\boldsymbol{\Theta}_{\mathcal{M}}\|_2 , \tag{32}$$

where $\nu(\mathcal{M}) = M_1\sqrt{Kd} + (M_2\sqrt{K} + M_3)\sqrt{s}$ is the support space compatibility constant *defined in (25) .*

Proof of Lemma 22: The detailed proof of Lemma 22 is discussed in S.V. ■

Next lemma is a key step to establish our main theorem. It quantifies the estimation error in one iteration step. According to this lemma, one can precisely understand how the statistical error and optimization error accumulate with more and more iterations.

Lemma 23 *Suppose $\boldsymbol{\Theta}^*$ lies in the interior of Ξ. If $\boldsymbol{\Theta}^{(t-1)} \in \mathcal{B}_{\alpha}(\boldsymbol{\Theta}^*)$, with choice of $\lambda_n^{(t)} = \varepsilon + \tau\|\boldsymbol{\Theta}^{(t-1)} - \boldsymbol{\Theta}^*\|_2/\nu(\mathcal{M})$, final estimation error satisfies $\|\boldsymbol{\Theta}^{(t)} - \boldsymbol{\Theta}^*\|_2 \leq 6\nu(\mathcal{M})\lambda_n^{(t)}/\gamma$ with probability at least $1-\delta'$ for all $t = 1, 2, \ldots$. Here τ, λ and $\nu(\mathcal{M})$ are defined in Lemma 7, Lemma 9 and Lemma 22 accordingly.*

Proof of Lemma 23: Proof is postponed to section B.1. ■

Equipped with Lemmas 23, we are able to precisely quantify the final estimation error after t iteration steps. This can be achieved by mathematical induction. For simplicity, define $\kappa := 6\tau/\gamma$. When $t = 1$, we have $\boldsymbol{\Theta}^{(0)} \in \mathcal{B}_{\alpha}(\boldsymbol{\Theta}^*)$. Applying Lemma 23 yields that

$$\begin{aligned} \left\| \boldsymbol{\Theta}^{(1)} - \boldsymbol{\Theta}^* \right\|_2 &\leq \frac{6\lambda_n^{(1)}\nu(\mathcal{M})}{\gamma} \\ &= \frac{6\nu(\mathcal{M})}{\gamma}\varepsilon + \kappa \left\| \boldsymbol{\Theta}^{(0)} - \boldsymbol{\Theta}^* \right\|_2 . \end{aligned}$$

Suppose the following inequality is true for some $t \geq 1$,

$$\left\| \boldsymbol{\Theta}^{(t)} - \boldsymbol{\Theta}^* \right\|_2 \leq \frac{1-\kappa^t}{1-\kappa} \frac{6\nu(\mathcal{M})}{\gamma}\varepsilon + \kappa^t \left\| \boldsymbol{\Theta}^{(0)} - \boldsymbol{\Theta}^* \right\|_2 ,$$

with probability at least $1-t\delta'$. We need to verify when $t=t+1$, the above inequality still holds. First, we show that $\mathbf{\Theta}^{(t)}$ is within a ball of $\mathbf{\Theta}^*$ with radius α. Under the assumption that $\varepsilon \leq (1-\kappa)\alpha\gamma/(6\nu(\mathcal{M}))$ for sufficient large n, we have

$$\begin{aligned}\left\|\mathbf{\Theta}^{(t)}-\mathbf{\Theta}^*\right\|_2 &\leq \frac{1-\kappa^t}{1-\kappa}\frac{6\nu(\mathcal{M})}{\gamma}\frac{(1-\kappa)\alpha\gamma}{6\nu(\mathcal{M})}+\kappa^t\left\|\mathbf{\Theta}^{(0)}-\mathbf{\Theta}^*\right\|_2\\ &\leq (1-\kappa^t)\alpha+\kappa^t\alpha=\alpha.\end{aligned}$$

Consequently, we have $\mathbf{\Theta}^{(t)}\in\mathcal{B}_\alpha(\mathbf{\Theta}^*)$. Applying Lemma 23 with $t+1$ implies that

$$\begin{aligned}\left\|\mathbf{\Theta}^{(t+1)}-\mathbf{\Theta}^*\right\|_2 &\leq \frac{6\nu(\mathcal{M})\varepsilon}{\gamma}+\kappa\left\|\mathbf{\Theta}^{(t)}-\mathbf{\Theta}^*\right\|_2\\ &\leq \frac{6\nu(\mathcal{M})\varepsilon}{\gamma}+\kappa\left(\frac{1-\kappa^t}{1-\kappa}\frac{6\nu(\mathcal{M})\varepsilon}{\gamma}+\kappa^t\left\|\mathbf{\Theta}^{(0)}-\mathbf{\Theta}^*\right\|_2\right)\\ &= \frac{1-\kappa^{t+1}}{1-\kappa}\frac{6\nu(\mathcal{M})\varepsilon}{\gamma}+\kappa^{t+1}\left\|\mathbf{\Theta}^{(0)}-\mathbf{\Theta}^*\right\|_2,\end{aligned}$$

with probability at least $1-(t+1)\delta'$. Therefore, we reach the conclusion that

$$\begin{aligned}\left\|\mathbf{\Theta}^{(t)}-\mathbf{\Theta}^*\right\|_2 &\leq \frac{1-\kappa^t}{1-\kappa}\frac{6\nu(\mathcal{M})}{\gamma}\varepsilon+\kappa^t\left\|\mathbf{\Theta}^{(0)}-\mathbf{\Theta}^*\right\|_2\\ &\leq \frac{6\nu(\mathcal{M})\varepsilon}{(1-\kappa)\gamma}+\kappa^t\left\|\mathbf{\Theta}^{(0)}-\mathbf{\Theta}^*\right\|_2,\end{aligned}$$

with probability at least $1-t\delta'$. This concludes the proof of Theorem 13. ■

A.1 Proof of Corollary 18

It is worth to notice that sufficiently large iterations ensure that the optimization error will be dominated by statistical error finally as $\kappa<1/2$. First we provide a stopping rule T. Plugging $\varepsilon_1,\varepsilon_2$ from (S.14) & (S.15) into statistical error part and letting $\delta=1/p$, we have:

$$\begin{aligned}SE=&\frac{1}{1-\kappa}\frac{6}{\gamma}\left[\left(\sqrt{Kd}+(\sqrt{K}+1)\sqrt{s}\right)\left(CK\|\mathbf{\Omega}^*\|_\infty+C'K^{1.5}\right)\sqrt{\frac{\log p}{n}}\right]\\ &+\frac{1}{1-\kappa}\frac{6}{\gamma}\left[C''\sqrt{p}\sqrt{\frac{K^3\log p}{n}}\right].\end{aligned}$$

Note that under Condition 17, $K=o(p)$. Then SE is simplified by

$$SE\leq\frac{6\widetilde{C}}{(1-\kappa)\gamma}\|\mathbf{\Omega}^*\|_\infty\left(\sqrt{Kd}+\sqrt{Ks+p}\right)\sqrt{\frac{K^3\log p}{n}},$$

for some constant $\widetilde{C}$. For simplicity, let's denote

$$\varphi(n,p,K)=\frac{6\widetilde{C}}{(1-\kappa)\gamma}\|\mathbf{\Omega}^*\|_\infty\left(\sqrt{Kd}+\sqrt{Ks+p}\right)\sqrt{\frac{K^3\log p}{n}}.$$

Therefore, the bound (27) suggests a reasonable choice of the number of iterations. In particular, when

$$t\geq T=\log_{1/\kappa}\left(\frac{\left\|\mathbf{\Theta}^{(0)}-\mathbf{\Theta}^*\right\|_2}{\varphi(n,p,K)}\right), \tag{33}$$

the optimization error is dominated by statistical error. Final estimation error will be upper bounded by

$$\left\|\mathbf{\Theta}^{(T)}-\mathbf{\Theta}^*\right\|_2\leq\frac{12\widetilde{C}}{(1-\kappa)\gamma}\left(\|\mathbf{\Omega}^*\|_\infty\sqrt{\frac{K^5d\log p}{n}}+\|\mathbf{\Omega}^*\|_\infty\sqrt{\frac{K^3(Ks+p)\log p}{n}}\right),$$

with probability at least $1-T(26K^2+8K+1)/p$. Plugging in the expression of T in (33), the probability term is bounded by:

$$\begin{aligned}\frac{T(26K^2+8K+1)}{p} &\lesssim \frac{\log_{1/\kappa}\left(n/\left(\left(\sqrt{Kd}+\sqrt{Ks+p}\right)\sqrt{K^3\log p}\right)\right)K^2}{p}\\ &\lesssim \frac{K^2\log_{1/\kappa}n}{p}.\end{aligned}$$

Under Condition 17, $T(26K^2+8K+1)/p$ goes to zero as K and p diverging. Putting pieces together, we have

$$\left\|\mathbf{\Theta}^{(T)}-\mathbf{\Theta}^*\right\|_2\leq\frac{12\widetilde{C}}{(1-\kappa)\gamma}\left(\|\mathbf{\Omega}^*\|_\infty\sqrt{\frac{K^5d\log p}{n}}+\|\mathbf{\Omega}^*\|_\infty\sqrt{\frac{K^3(Ks+p)\log p}{n}}\right),$$

which implies

$$\begin{aligned}&\sum_{k=1}^K\left(\left\|\boldsymbol{\mu}_k^{(T)}-\boldsymbol{\mu}_k^*\right\|_2+\left\|\mathbf{\Omega}_k^{(T)}-\mathbf{\Omega}_k^*\right\|_F\right)\\ &\leq\frac{12\widetilde{C}}{(1-\kappa)\gamma}\left(\|\mathbf{\Omega}^*\|_\infty\sqrt{\frac{K^5d\log p}{n}}+\|\mathbf{\Omega}^*\|_\infty\sqrt{\frac{K^3(Ks+p)\log p}{n}}\right),\end{aligned}$$

with probability converging to 1. It ends the proof of Corollary 18. ■

Appendix B. Proof of Key Lemmas

B.1 Proof of Lemma 23

We first consider an unsymmetrized version of $\mathbf{\Theta}^{(t)}$. Our proof makes use of the function $f:\Xi\to\mathbb{R}$ given by:

$$f(\Delta):=Q_n(\mathbf{\Theta}^*+\Delta|\mathbf{\Theta}^{(t-1)})-Q_n(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)})-\lambda_n^{(t)}\left(\mathcal{P}(\mathbf{\Theta}^*+\Delta)-\mathcal{P}(\mathbf{\Theta}^*)\right).$$

This function helps us evaluate the error between the iterative estimator $\mathbf{\Theta}^{(t)}$ and the true parameter $\mathbf{\Theta}^*$. In addition, we exploit the following fact:

$$\begin{cases} f(0) = 0 \\ f(\widehat{\Delta}) \geq 0 \text{ when } \widehat{\Delta} = \mathbf{\Theta}^{(t)} - \mathbf{\Theta}^*. \end{cases} \tag{34}$$

The second property is from the optimality of $\mathbf{\Theta}^{(t)}$ in terms of the sample version objective function. In detail,

$$\mathbf{\Theta}^{(t)} = \arg\max_{\mathbf{\Theta}'} Q_n(\mathbf{\Theta}'|\mathbf{\Theta}^{(t-1)}) - \lambda_n^{(t)}\mathcal{P}(\mathbf{\Theta}'). \tag{35}$$

Correspondingly, there is a classical result named self-consistency property for population version objective function in McLachlan and Krishnan (2007), which in detail is

$$\mathbf{\Theta}^* = \arg\max_{\mathbf{\Theta}'} Q(\mathbf{\Theta}'|\mathbf{\Theta}^*). \tag{36}$$

The whole proof follows two steps. In Step I, we show that $f(\Delta) < 0$ if $\|\Delta\|_2 = \xi$. Next in Step II, we show that the error term $\widehat{\Delta}$ must satisfy $\|\widehat{\Delta}\|_2 < \xi$ under the result in Step I.

Step I: we begin to establish an upper bound on $f(\Delta)$ over the set $\mathbb{C}(\xi) := \{\Delta : \|\Delta\|_2 = \xi\}$ for the chosen radius $\xi = 6\lambda_n^{(t)}\nu(\mathcal{M})/\gamma$. From the assumption on n, when n is large enough,

$$\varepsilon \leq \frac{(1-\kappa)\alpha\gamma}{6\nu(\mathcal{M})} \leq \frac{(2-\kappa)\alpha\gamma}{6\nu(\mathcal{M})},$$
$$\frac{6\nu(\mathcal{M})\varepsilon}{\gamma} \leq (2-\kappa)\alpha.$$

On the other hand, as $\|\mathbf{\Theta}^{(t-1)} - \mathbf{\Theta}^*\|_2 \leq \alpha$, ξ satisfies,

$$\xi = \frac{6\nu(\mathcal{M})\varepsilon}{\gamma} + \kappa \left\|\mathbf{\Theta}^{(t-1)} - \mathbf{\Theta}^*\right\|_2 \leq 2\alpha.$$

It is sufficient to show that $\mathbb{C}(\xi) \subseteq \mathbb{C} = \{\Delta | \|\Delta\|_2 \leq 2\alpha\}$. According to Lemma 9, replacing $\mathbf{\Theta}' - \mathbf{\Theta}^*$ by Δ, then any $\Delta \in \mathbb{C}(\xi)$ enjoys restricted strong concavity property, which implies:

$$Q_n(\mathbf{\Theta}^* + \Delta|\mathbf{\Theta}^{(t-1)}) - Q_n(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)}) \leq \left\langle \nabla Q_n(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)}), \Delta \right\rangle - \frac{\gamma}{2}\|\Delta\|_2^2,$$

with probability at least $1-\delta$. Subtracting $\lambda_n^{(t)}(\mathcal{P}(\mathbf{\Theta}^* + \Delta) - \mathcal{P}(\mathbf{\Theta}^*))$ from both sides, we construct an upper bound of $f(\Delta)$ in the right side,

$$f(\Delta) \leq \underbrace{\left\langle \nabla Q_n(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)}), \Delta \right\rangle}_{(i)} - \lambda_n^{(t)} \underbrace{\left(\mathcal{P}(\mathbf{\Theta}^* + \Delta) - \mathcal{P}(\mathbf{\Theta}^*)\right)}_{(ii)} - \frac{\gamma}{2}\|\Delta\|_2^2.$$

Bounding (i)**:** Note that Q_n is a sample version Q-function but $\mathbf{\Theta}^*$ comes from population version Q-function (36). So we use $\nabla Q(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)})$ as a bridge to connect the sample-based

analysis and population-based analysis together.

$$\begin{aligned}(i) \leq\ & |\big\langle \nabla Q_n(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)}) - \nabla Q(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)}) \\ & \qquad + \nabla Q(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)}) - \nabla Q(\mathbf{\Theta}^*|\mathbf{\Theta}^*), \Delta \big\rangle| \\ \leq\ & \underbrace{|\big\langle \nabla Q_n(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)}) - \nabla Q(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)}), \Delta \big\rangle|}_{\text{Statistical Error(SE)}} \\ & \qquad + \underbrace{|\big\langle \nabla Q(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)}) - \nabla Q(\mathbf{\Theta}^*|\mathbf{\Theta}^*), \Delta \big\rangle|}_{\text{Optimization Error(OE)}}.\end{aligned}$$

Note that $\mathbf{\Theta}^*$ lies in the interior of Ξ. According to the self-consistency property (36), $\nabla Q(\mathbf{\Theta}^*|\mathbf{\Theta}^*) = 0$ which implies the first inequality holds. This decomposition for (i) leads to the optimization error part and statistical error part.

For simplicity, we write $h(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)}) = \nabla Q_n(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)}) - \nabla Q(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)})$. Since the group graphical lasso penalty does not penalize the diagonal element, it is a semi-norm. Recall that both Δ and $h(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)})$ are $K(p^2+p)$ dimensional vectors. Then by the definition of $\mathcal{G}$ and $\mathcal{G}^c$ in (14), statistical error can be decomposed further by:

$$\begin{aligned}\text{SE} \leq\ & \left|\langle h(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)})_{\mathcal{G}^c}, \Delta_{\mathcal{G}^c}\rangle\right| + \left|\langle h(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)})_{\mathcal{G}}, \Delta_{\mathcal{G}}\rangle\right| \\ \leq\ & \left\|h(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)})_{\mathcal{G}^c}\right\|_{\mathcal{P}^*} \cdot \mathcal{P}(\Delta_{\mathcal{G}^c}) + \|h(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)})_{\mathcal{G}}\|_2 \cdot \|\Delta_{\mathcal{G}}\|_2 \\ \leq\ & \|h(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)})\|_{\mathcal{P}^*} \cdot \mathcal{P}(\Delta) + \|h(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)})_{\mathcal{G}}\|_2 \cdot \|\Delta\|_2.\end{aligned}$$

The second inequality comes from the generalized Cauchy-Schwarz inequality. After excluding the diagonal terms from precision matrices, $\mathcal{P}(\Delta_{\mathcal{G}^c})$ can be treated as a norm. The last inequality is because both the penalties $\mathcal{P}$ and $\mathcal{P}^*$ do not penalize the diagonal term of precision matrices. Under statistical error Condition 10,

$$\text{SE} \leq \varepsilon_1\mathcal{P}(\Delta) + \varepsilon_2\|\Delta\|_2, \tag{37}$$

with probability at least $1 - (\delta_1 + \delta_2)$.

On the other hand, from the assumption that $\mathbf{\Theta}^{(t-1)}$ is in the $\mathcal{B}_\alpha(\mathbf{\Theta}^*)$, we are able to apply the Gradient Stability condition in Lemma 7 to bound OE.

$$\begin{aligned}\text{OE} \leq\ & \|\nabla Q(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)}) - \nabla Q(\mathbf{\Theta}^*|\mathbf{\Theta}^*)\|_2 \cdot \|\Delta\|_2 \\ \leq\ & \tau\|\mathbf{\Theta}^{(t-1)} - \mathbf{\Theta}^*\|_2 \cdot \|\Delta\|_2.\end{aligned} \tag{38}$$

Therefore, putting (37) and (38) together, (i) is upper bounded by

$$(i) \leq \varepsilon_1\mathcal{P}(\Delta) + \varepsilon_2\|\Delta\|_2 + \tau\|\mathbf{\Theta}^{(t-1)} - \mathbf{\Theta}^*\|_2 \cdot \|\Delta\|_2, \tag{39}$$

with probability at least $1 - (\delta_1 + \delta_2)$.

Bounding (ii)**:** The decomposability of SCAN penalty in Lemma 21 implies $\mathcal{P}(\mathbf{\Theta}^* + \Delta) = \mathcal{P}(\mathbf{\Theta}^* + \Delta_{\mathcal{M}}) + \mathcal{P}(\Delta_{\mathcal{M}^\perp})$. By triangle inequality, it is sufficient to bound (ii),

$$\begin{aligned}(ii) =\ & \mathcal{P}(\mathbf{\Theta}^* + \Delta_{\mathcal{M}}) + \mathcal{P}(\Delta_{\mathcal{M}^\perp}) - \mathcal{P}(\mathbf{\Theta}^*) \\ \geq\ & \mathcal{P}(\mathbf{\Theta}^*) - \mathcal{P}(\Delta_{\mathcal{M}}) + \mathcal{P}(\Delta_{\mathcal{M}^\perp}) - \mathcal{P}(\mathbf{\Theta}^*) \\ =\ & \mathcal{P}(\Delta_{\mathcal{M}^\perp}) - \mathcal{P}(\Delta_{\mathcal{M}}).\end{aligned} \tag{40}$$

Combining (39) and (40), $f(\Delta)$ is upper bounded by:

$$f(\Delta) \leq \varepsilon_1 \mathcal{P}(\Delta) + \varepsilon_2 \|\Delta\|_2 + \tau \|\mathbf{\Theta}^{(t-1)} - \mathbf{\Theta}^*\|_2 \cdot \|\Delta\|_2 - \lambda_n^{(t)} \left(\mathcal{P}(\Delta_{\mathcal{M}^\perp}) - \mathcal{P}(\Delta_{\mathcal{M}})\right) - \frac{\gamma}{2}\|\Delta\|_2^2.$$

Triangle inequality implies $\mathcal{P}(\Delta) \leq \mathcal{P}(\Delta_{\mathcal{M}}) + \mathcal{P}(\Delta_{\mathcal{M}^\perp})$. After combining some terms, the right hand side above could be further bounded by:

$$f(\Delta) \leq -\frac{\gamma}{2}\|\Delta\|_2^2 + (\lambda_n^{(t)} + \varepsilon_1)\mathcal{P}(\Delta_{\mathcal{M}}) + (\varepsilon_1 - \lambda_n^{(t)})\mathcal{P}(\Delta_{\mathcal{M}^\perp}) + \varepsilon_2\|\Delta\|_2 + \tau\|\mathbf{\Theta}^{(t-1)} - \mathbf{\Theta}^*\|_2 \cdot \|\Delta\|_2, \tag{41}$$

with probability at least $1 - (\delta + \delta_1 + \delta_2)$. Let $\delta' = \delta + \delta_1 + \delta_2$. According to Lemma 22, we have the inequality $\mathcal{P}(\Delta_{\mathcal{M}}) \leq \nu(\mathcal{M})\|\Delta_{\mathcal{M}}\|_2$. By the definition of $\Pi_{\mathcal{M}}(\Delta)$, we have

$$\|\Delta_{\mathcal{M}}\|_2 = \|\Pi_{\mathcal{M}}(\Delta) - \Pi_{\mathcal{M}}(0)\|_2 \leq \|\Delta - 0\|_2 = \|\Delta\|_2.$$

Then $\mathcal{P}(\Delta_{\mathcal{M}})$ is further bounded by

$$\mathcal{P}(\Delta_{\mathcal{M}}) \leq \nu(\mathcal{M})\|\Delta\|_2. \tag{42}$$

Substituting (42) into (41), we obtain:

$$f(\Delta) \leq \left(\varepsilon_1 + \frac{\varepsilon_2 + \tau\|\mathbf{\Theta}^{(t-1)} - \mathbf{\Theta}^*\|_2}{\nu(\mathcal{M})}\right)\nu(\mathcal{M})\|\Delta\|_2 - \frac{\gamma}{2}\|\Delta\|_2^2 + \lambda_n^{(t)}\nu(\mathcal{M})\|\Delta\|_2 + (\varepsilon_1 - \lambda_n^{(t)})\mathcal{P}(\Delta_{\mathcal{M}^\perp}),$$

with at least probability $1 - \delta'$. Recall that we choose

$$\lambda_n^{(t)} = \varepsilon + \frac{\tau\|\mathbf{\Theta}^{(t-1)} - \mathbf{\Theta}^*\|_2}{\nu(\mathcal{M})}, \epsilon = \epsilon_1 + \frac{\epsilon_2}{\nu(\mathcal{M})}.$$

From the construction of $\lambda_n^{(t)}$, the inequality $\varepsilon_1 - \lambda_n^{(t)} < 0$ always holds. Therefore, the upper bound for $f(\Delta)$ can be simplified by

$$\begin{aligned} f(\Delta) &\leq -\frac{\gamma}{2}\|\Delta\|_2^2 + 2\lambda_n^{(t)}\nu(\mathcal{M})\|\Delta\|_2 \\ &= -\frac{6(\lambda_n^{(t)}\nu(\mathcal{M}))^2}{\gamma} < 0. \end{aligned}$$

where the above equality is due to $\Delta \in \mathbb{C}(\xi)$. Now we reach the conclusion that $f(\Delta) < 0$ for all vectors $\Delta \in \mathbb{C}(\xi)$.

Step II: Now we start to prove the following statement: if for some optimal solution $\mathbf{\Theta}^{(t)}$ in (35), the corresponding error term $\widehat{\Delta} = \mathbf{\Theta}^{(t)} - \mathbf{\Theta}^*$ satisfies the inequality $\|\widehat{\Delta}\|_2 > \xi$, there must exist some vectors $\widetilde{\Delta}$ which belong to $\mathbb{C}(\xi)$ such that $f(\widetilde{\Delta}) \geq 0$. Before our forward proofs, let's state a lemma which describe the curvature of function $Q_n(\cdot|\mathbf{\Theta}^{(t-1)})$.

Lemma 24 *$Q_n(\cdot|\mathbf{\Theta}^{(t-1)})$ satisfies the following inequality a.s.:*

$$Q_n(\mathbf{\Theta}^{(1)}|\mathbf{\Theta}^{(t-1)}) - Q_n\left(\mathbf{\Theta}^{(2)}|\mathbf{\Theta}^{(t-1)}\right) \leq \left\langle \nabla Q_n\left(\mathbf{\Theta}^{(2)}|\mathbf{\Theta}^{(t-1)}\right), \mathbf{\Theta}^{(1)} - \mathbf{\Theta}^{(2)}\right\rangle.$$

when $(\mathbf{\Theta}^{(1)}, \mathbf{\Theta}^{(2)}) = (\mathbf{\Theta}^{(t)}, t^*\mathbf{\Theta}^{(t)} + (1-t^*)\mathbf{\Theta}^*)$ *or* $(\mathbf{\Theta}^*, t^*\mathbf{\Theta}^{(t)} + (1-t^*)\mathbf{\Theta}^*)$.

Proof of Lemma 24: The detailed proof of Lemma 24 is discussed in S.VI. ■

The lemma tells us that we only require sample-based Q-function to be point-wise concave rather than global concave. If $\|\widehat{\Delta}\|_2 > \xi$, then the line joining $\widehat{\Delta}$ to 0 must intersect the set $\mathbb{C}(\xi)$ at some intermediate points $t^*\widehat{\Delta}$, for some $t^* \in (0,1)$. According to Lemma 24,

$$\begin{aligned} &Q_n(\mathbf{\Theta}^{(t)}|\mathbf{\Theta}^{(t-1)}) - Q_n(t^*\mathbf{\Theta}^{(t)} + (1-t^*)\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)}) \\ &\quad \leq \left\langle \nabla Q_n(t^*\mathbf{\Theta}^{(t)} + (1-t^*)\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)}), (1-t^*)(\mathbf{\Theta}^{(t)} - \mathbf{\Theta}^*)\right\rangle \\ &Q_n(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)}) - Q_n\left(t^*\mathbf{\Theta}^{(t)} + (1-t^*)\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)}\right) \\ &\quad \leq \left\langle \nabla Q_n\left(t^*\mathbf{\Theta}^{(t)} + (1-t^*)\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)}\right), -t^*(\mathbf{\Theta}^{(t)} - \mathbf{\Theta}^*)\right\rangle. \end{aligned}$$

Adding the above two inequalities together with proper scaling, we can get

$$t^*Q_n(\mathbf{\Theta}^{(t)}|\mathbf{\Theta}^{(t-1)}) + (1-t^*)Q_n(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)}) \leq Q_n(t^*\mathbf{\Theta}^{(t)} + (1-t^*)\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)}).$$

According to the convexity of $\mathcal{P}(\mathbf{\Theta})$,

$$\begin{aligned} &\mathcal{P}\left(\mathbf{\Theta}^* + t^*\widehat{\Delta}\right) - \mathcal{P}\left(\mathbf{\Theta}^*\right) = \mathcal{P}\left(t^*\mathbf{\Theta}^{(t)} + (1-t^*)\mathbf{\Theta}^*\right) - \mathcal{P}(\mathbf{\Theta}^*) \\ &\leq t^*\mathcal{P}(\mathbf{\Theta}^{(t)}) + (1-t^*)\mathcal{P}(\mathbf{\Theta}^*) - \mathcal{P}(\mathbf{\Theta}^*) = t^*\left(\mathcal{P}(\mathbf{\Theta}^{(t)}) - \mathcal{P}(\mathbf{\Theta}^*)\right). \end{aligned}$$

Putting the above pieces together, it is shown that

$$\begin{aligned} f(t^*\widehat{\Delta}) &= Q_n\left(t^*\mathbf{\Theta}^{(t)} + (1-t^*)\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)}\right) - Q_n\left(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)}\right) \\ &\qquad - \lambda_n^{(t)}\left(\mathcal{P}(\mathbf{\Theta}^* + \Delta) - \mathcal{P}(\mathbf{\Theta}^*)\right) \\ &\geq t^*\left(Q_n(\mathbf{\Theta}^{(t)}|\mathbf{\Theta}^{(t-1)}) - Q_n(\mathbf{\Theta}^*|\mathbf{\Theta}^{(t-1)}) - \lambda_n^{(t)}\left(\mathcal{P}(\mathbf{\Theta}^{(t)}) - \mathcal{P}(\mathbf{\Theta}^*)\right)\right) \\ &= t^*f(\widehat{\Delta}). \end{aligned}$$

On the other hand, the optimality property (34) guarantees $f(\widehat{\Delta}) \geq 0$, and hence $f(t^*\widehat{\Delta}) \geq 0$ as well. Thus, we have constructed a vector $\widetilde{\Delta} = t^*\widehat{\Delta}$ with the claimed properties. This proves the statement in the beginning of Step II. Therefore, combining with the result in Step I, the contradiction of the statement in Step II implies that

$$\left\|\mathbf{\Theta}^{(t)} - \mathbf{\Theta}^*\right\|_2 \leq \xi = \frac{6\lambda_n^{(t)}\nu(\mathcal{M})}{\gamma}, \tag{43}$$

with probability at least $1 - \delta'$. This concludes the proof of Lemma 23. ■

References

Sivaraman Balakrishnan, Martin J Wainwright, and Bin Yu. Statistical guarantees for the em algorithm: From population to sample-based analysis. *Annals of Statistics*, page To appear, 2016.

P. Bickel and E. Levina. Covariance regularization by thresholding. *Annals of Statistics*, 36:2577–2604, 2008.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3:1–122, 2011.

T. Tony Cai, Hongzhe Li, Weidong Liu, and Jichun Xie. Joint estimation of multiple high-dimensional precision matrices. *Statistica Sinica*, 26(2):445–464, 2016a.

T. Tony Cai, Weidong Liu, and Harrison H. Zhou. Estimating sparse precision matrix: Optimal rates of convergence and adaptive estimation. *Ann. Statist.*, 44(2):455–488, 04 2016b.

Tony Cai, Weidong Liu, and Xi Luo. A constrained 1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494): 594–607, 2011.

Y. Chen, D. Pavlov, and J. Canny. Large-scale behavioral targeting. In *ACM SIGKDD*, pages 209–218, 2009.

Patrick Danaher, Pei Wang, and Daniela M. Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B*, 76(2):373–397, 2014. ISSN 1467-9868. doi: 10.1111/rssb.12033.

J. Fan and J. Lv. Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B*, 70:849–911, 2008.

Jianqing Fan, Yang Feng, and Yichao Wu. Network exploration via the adaptive lasso and scad penalties. *The Annals of Applied Statistics*, 3(2):521, 2009.

J. Friedman, H. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9:432–441, 2008.

C. Gao, Y. Zhu, X. Shen, and W. Pan. Estimation of multiple networks in gaussian mixture models. *Electronic Journal of Statistics*, 10:1133–1154, 2016.

Jian Guo, Elizaveta Levina, George Michailidis, and Ji Zhu. Joint estimation of multiple graphical models. *Biometrika*, 98(1):1–15, 2011. doi: 10.1093/biomet/asq060.

Nhat Ho and XuanLong Nguyen. Identifiability and optimal rates of convergence for parameters of multiple types in finite mixtures. *arXiv preprint arXiv:1501.02497*, 2015.

R. A. Horn and C. R. Johnson. *Matrix Analysis*. New York: Cambridge Univ. Press, 1988.

P. Jeziorski and I. Segal. What makes them click: Empirical analysis of consumer demand for search advertising. *American Economic Journal*, 7:24–53, 2015.

S.L. Lauritzen. *Graphical Models*. Oxford Science Publications, 1996.

Wonyul Lee and Yufeng Liu. Joint estimation of multiple precision matrices with common structures. *Journal of Machine Learning Research*, 16:1035–1062, 2015.

J. MacQueen. Some methods for clasification and analysis of multivariate observations. *In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley Series in Probability and Statistics., 2007.

Xiao-Li Meng and Donald B. Rubin. Maximum likelihood estimation via the ecm algorithm: A general framework. *Biometrika*, 80(2):267–278, 1993.

Sahand N. Negahban, Pradeep Ravikumar, Martin J. Wainwright, and Bin Yu. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statist. Sci.*, 27(4):538–557, 11 2012. doi: 10.1214/12-STS400.

W. Pan and X. Shen. Penalized model-based clustering with application to variable selection. *Journal of Machine Learning Research*, 8:1145–1164, 2007.

Jie Peng, Pei Wang, Nengfeng Zhou, and Ji Zhu. Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association*, 104(486):735–746, 2009.

Christine Peterson, Francesco C. Stingo, and Marina Vannucci. Bayesian inference of multiple gaussian graphical models. *Journal of the American Statistical Association*, 110(509): 159–174, 2015.

Huitong Qiu, Fang Han, Han Liu, and Brian Caffo. Joint estimation of multiple graphical models from high dimensional time series. *Journal of the Royal Statistical Society: Series B*, 78(2):487–504, 2016. ISSN 1467-9868.

Yordan P Raykov, Alexis Boukouvalas, Fahd Baig, and Max A Little. What to do when k-means clustering fails: a simple yet principled alternative algorithm. *PloS one*, 11(9): e0162259, 2016.

A. J. Rothman and L. Forzani. On the existence of the weighted bridge penalized gaussian likelihood precision matrix estimator. *Electronic Journal of Statistics*, 8:2693–2700, 2014.

Takumi Saegusa and Ali Shojaie. Joint estimation of precision matrices in heterogeneous populations. *Electronic Journal of Statistics*, page To appear, 2016.

X. Shen, W. Pan, and Y. Zhu. Likelihood-based selection and sharp parameter estimation. *Journal of the American Statistical Association*, 107:223–232, 2012.

Ali Shojaie and George Michailidis. Penalized likelihood methods for estimation of sparse high-dimensional directed acyclic graphs. *Biometrika*, 97(3):519–538, 2010a.

Ali Shojaie and George Michailidis. Penalized principal component regression on graphs for analysis of subnetworks. *Advances in Neural Information Processing Systems*, pages 2155–2163, 2010b.

Wei Sun, Junhui Wang, and Yixin Fang. Regularized k-means clustering of high-dimensional data and its asymptotic consistency. *Electron. J. Statist.*, 6:148–167, 2012. doi: 10.1214/12-EJS668.

TCGA. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, 455:1061–1068, 2008.

R. G. Verhaak, K. A. Hoadley, E. Purdom, V. Wang, Y. Qi, M. D. Wilkerson, C. R. Miller, L. Ding, T. Golub, J. P. Mesirov, G. Alexe, M. Lawrence, M. OKelly, P. Tamayo, B. A. Weir, S. Gabriel, W. Winckler, S. Gupta, L. Jakkula, H. S. Feiler, J. G. Hodgson, C. D. James, J. N. Sarkaria, C. Brennan, A. Kahn, P. T. Spellman, R. K. Wilson, T. P. Speed, J. W. Gray, M. Meyerson, G. Getz, C. M. Perou, D. N. Hayes, and TCGA. Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in pdgfra, idh1, egfr, and nf1. *Cancer Cell*, 17:98–110, 2010.

Roman Vershynin. *Compressed sensing*, chapter Introduction to the non-asymptotic analysis of random matrices, pages 210–268. Cambridge Univ. Press, 2012.

Martin J. Wainwright. Structured regularizers for high-dimensional problems: Statistical and computational issues. *Annual Review of Statistics and Its Application*, 1(1):233–253, 2014. doi: 10.1146/annurev-statistics-022513-115643.

Junhui Wang. Consistent selection of the number of clusters via crossvalidation. *Biometrika*, 97:893–904, 2010.

Junhui Wang. Joint estimation of sparse multivariate regression and conditional graphical models. *Statistica Sinica*, 25:831–851, 2015.

Pengyuan Wang, Wei Sun, Dawei Yin, Jimmy Yang, and Yi Chang. Robust tree-based causal inference for complex ad effectiveness analysis. In *Proceedings of 8th ACM Conference on Web Search and Data Mining*, 2015a.

Zhaoran Wang, Quanquan Gu, Yang Ning, and Han Liu. High dimensional em algorithm: Statistical optimization and asymptotic normality. *Advances in Neural Information Processing Systems*, pages 2512–2520, 2015b.

J. Wolfe, A. Haghighi, and D. Klein. Fully distributed em for very large datasets. *The International Conference on Machine Learning*, pages 1184–1191, 2008.

J. Yan, N. Liu, G. Wang, W. Zhang, Y. Jiang, and Z. Chen. How much can behavioral targeting help online advertising? In *International ACM WWW Conference*, pages 261–270, 2009.

Xinyang Yi and Constantine Caramanis. Regularized em algorithms: A unified framework and statistical guarantees. *Advances in Neural Information Processing Systems*, pages 1567–1575, 2015.

M. Yuan and Y. Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94:19–35, 2007.

Yuchen Zhang, Xi Chen, Denny Zhou, and Michael I Jordan. Spectral methods meet em: A provably optimal algorithm for crowdsourcing. *Advances in Neural Information Processing Systems*, pages 1260–1268, 2014.

Hui Zhou, Wei Pan, and Xiaotong Shen. Penalized model-based clustering with unconstrained covariance matrices. *Electron. J. Statist.*, 3:1473–1496, 2009. doi: 10.1214/09-EJS487.

Y. Zhu, X. Shen, and W. Pan. Structural pursuit over multiple undirected graphs. *Journal of the American Statistical Association*, 109:1683–1696, 2014.

Online Supplementary

This supplementary contains supporting lemmas and their proofs for the theoretical developments in the main paper.

Appendix A. Proof of Several Lemmas and Selection Consistency

S.I Proof of Lemma 5

The result follows by setting the derivative of $Q(\boldsymbol{\Theta}^{'}|\boldsymbol{\Theta})$ with respect to $\boldsymbol{\mu}_k^{'}$ or $\boldsymbol{\Omega}_k^{'}$ as zero. In particular, solving

$$\frac{\partial Q(\boldsymbol{\Theta}^{'}|\boldsymbol{\Theta})}{\partial \boldsymbol{\mu}_k^{'}} = \mathbb{E}[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})\boldsymbol{\Omega}_k^{'}(\boldsymbol{X}-\boldsymbol{\mu}_k^{'})] = 0,$$

implies that

$$\arg\max_{\boldsymbol{\mu}_k^{'}} Q(\boldsymbol{\Theta}^{'}|\boldsymbol{\Theta}) = \frac{[\boldsymbol{\Omega}_k^{'}]^{-1}\mathbb{E}[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})\boldsymbol{\Omega}_k^{'}\boldsymbol{X}]}{\mathbb{E}[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})]} = \frac{\mathbb{E}[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})\boldsymbol{X}]}{\mathbb{E}[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})]}.$$

Similarly, solving

$$\frac{\partial Q(\boldsymbol{\Theta}^{'}|\boldsymbol{\Theta})}{\partial \boldsymbol{\Omega}_k^{'}} = \frac{1}{2}\mathbb{E}[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})][\boldsymbol{\Omega}_k^{'}]^{-1} - \frac{1}{2}\mathbb{E}[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})(\boldsymbol{X}-\boldsymbol{\mu}_k^{'})(\boldsymbol{X}-\boldsymbol{\mu}_k^{'})^{\top}] = 0,$$

implies (19). This ends the proof of Lemma 5. ■

S.II Proof of Lemma 7

We consider k-th group first

$$\left\|\nabla_{\boldsymbol{\Theta}_k^{'}} Q(\boldsymbol{\mu}_k^*,\boldsymbol{\Omega}_k^*|\boldsymbol{\Theta}^*) - \nabla_{\boldsymbol{\Theta}_k^{'}} Q(\boldsymbol{\mu}_k^*,\boldsymbol{\Omega}_k^*|\boldsymbol{\Theta})\right\|_2 \leq \tau\left\|\boldsymbol{\Theta}-\boldsymbol{\Theta}^*\right\|_2, \tag{S.1}$$

for any $\boldsymbol{\Theta} \in \mathbb{B}_\alpha(\boldsymbol{\Theta}^*)$. Remind that $\boldsymbol{\Theta}_k^{'} = \mathrm{vec}(\boldsymbol{\mu}_k, \boldsymbol{\Omega}_k) \in \mathbb{R}^{p^2+p}$. According to the derivation in the proof of Lemma 5, we have

$$\nabla_{\boldsymbol{\Theta}_k^{'}} Q(\boldsymbol{\Theta}_k^{'}|\boldsymbol{\Theta}) = \begin{pmatrix} \mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})\boldsymbol{\Omega}_k^{'}(\boldsymbol{X}-\boldsymbol{\mu}_k^{'})\right] \\ \mathrm{vec}\left\{\frac{1}{2}\mathbb{E}[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})]\boldsymbol{\Omega}_k^{'-1} - \frac{1}{2}\mathbb{E}[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})(\boldsymbol{X}-\boldsymbol{\mu}_k^{'})(\boldsymbol{X}-\boldsymbol{\mu}_k^{'})^{\top}]\right\}^{\top} \end{pmatrix}.$$

Define $D_L(\boldsymbol{\Theta}^*,\boldsymbol{\Theta}) = L_{\boldsymbol{\Theta}^*,k}(\boldsymbol{X}) - L_{\boldsymbol{\Theta},k}(\boldsymbol{X})$. Therefore, the square of the left hand side of $(S.1)$ can be simplified to

$$\begin{aligned} & \left\|\nabla_{\boldsymbol{\Theta}_k^{'}} Q(\boldsymbol{\mu}_k^*,\boldsymbol{\Omega}_k^*|\boldsymbol{\Theta}^*) - \nabla_{\boldsymbol{\Theta}_k^{'}} Q(\boldsymbol{\mu}_k^*,\boldsymbol{\Omega}_k^*|\boldsymbol{\Theta})\right\|_2^2 \\ = & \underbrace{\left\|\mathbb{E}\left[D_L(\boldsymbol{\Theta}^*,\boldsymbol{\Theta})\boldsymbol{\Omega}_k^*(\boldsymbol{X}-\boldsymbol{\mu}_k^*)\right]\right\|_2^2}_{I} \\ & + \underbrace{\left\|\frac{1}{2}\mathbb{E}[D_L(\boldsymbol{\Theta}^*,\boldsymbol{\Theta})\boldsymbol{\Omega}_k^{*-1} - \frac{1}{2}\mathbb{E}[D_L(\boldsymbol{\Theta}^*,\boldsymbol{\Theta})(\boldsymbol{X}-\boldsymbol{\mu}_k^*)(\boldsymbol{X}-\boldsymbol{\mu}_k^*)^{\top}]\right\|_F^2}_{II}. \end{aligned}$$

If we can show $I \leq \tau_1\|\boldsymbol{\Theta}-\boldsymbol{\Theta}^*\|_2^2$ and $II \leq \tau_2\|\boldsymbol{\Theta}-\boldsymbol{\Theta}^*\|_2^2$, then we have $\tau = \sqrt{\tau_1+\tau_2}$ since

$$\left\|\nabla_{\boldsymbol{\Theta}_k^{'}} Q(\boldsymbol{\mu}_k^*,\boldsymbol{\Omega}_k^*|\boldsymbol{\Theta}^*) - \nabla_{\boldsymbol{\Theta}_k^{'}} Q(\boldsymbol{\mu}_k^*,\boldsymbol{\Omega}_k^*|\boldsymbol{\Theta})\right\|_2 \leq \sqrt{\tau_1+\tau_2}\|\boldsymbol{\Theta}-\boldsymbol{\Theta}^*\|_2.$$

Bounding I: We apply Taylor expansion to simplify $D_L(\boldsymbol{\Theta}^*,\boldsymbol{\Theta})$. Remind that, by assumption, $\pi_k = 1/K$, and hence we have

$$L_{\boldsymbol{\Theta},k}(\boldsymbol{X}) = \frac{\pi_k f_k(\boldsymbol{X};\boldsymbol{\Theta}_k)}{\sum_{k=1}^K \pi_k f_k(\boldsymbol{X};\boldsymbol{\Theta}_k)} = \frac{|\boldsymbol{\Omega}_k|^{1/2}\exp\left\{-\frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu}_k)^{\top}\boldsymbol{\Omega}_k(\boldsymbol{X}-\boldsymbol{\mu}_k)\right\}}{\sum_{k=1}^K |\boldsymbol{\Omega}_k|^{1/2}\exp\left\{-\frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu}_k)^{\top}\boldsymbol{\Omega}_k(\boldsymbol{X}-\boldsymbol{\mu}_k)\right\}}.$$

Then, Taylor expansion of $L_{\boldsymbol{\Theta},k}(\boldsymbol{X})$ around $\boldsymbol{\Theta}_k^*$ leads to

$$L_{\boldsymbol{\Theta},k}(\boldsymbol{X}) = L_{\boldsymbol{\Theta}^*,k}(\boldsymbol{X}) + [\nabla_{\boldsymbol{\Theta}} L_{\boldsymbol{\Theta}_t,k}(\boldsymbol{X})]^{\top}(\boldsymbol{\Theta}-\boldsymbol{\Theta}^*), \tag{S.2}$$

where $\boldsymbol{\Theta}_t = \boldsymbol{\Theta}^* + t\Delta$ with $t \in [0,1]$ and $\Delta = \boldsymbol{\Theta}-\boldsymbol{\Theta}^*$. Here the derivative of $L_{\boldsymbol{\Theta},k}(\boldsymbol{X})$ with respect to $\boldsymbol{\Theta} = (\boldsymbol{\Theta}_1,\ldots,\boldsymbol{\Theta}_K)$ can be written as

$$\nabla_{\boldsymbol{\Theta}} L_{\boldsymbol{\Theta},k}(\boldsymbol{X}) = \left([\nabla_{\boldsymbol{\Theta}_1} L_{\boldsymbol{\Theta},k}(\boldsymbol{X})]^{\top},\ldots,[\nabla_{\boldsymbol{\Theta}_K} L_{\boldsymbol{\Theta},k}(\boldsymbol{X})]^{\top}\right)^{\top}, \tag{S.3}$$

where

$$\nabla_{\boldsymbol{\Theta}_j} L_{\boldsymbol{\Theta},k}(\boldsymbol{X}) = \begin{cases} -L_{\boldsymbol{\Theta},k}(\boldsymbol{X})\cdot L_{\boldsymbol{\Theta},j}(\boldsymbol{X})\cdot\delta_{\boldsymbol{\Theta}_j}(\boldsymbol{X}) \text{ when } j \neq k; \\ L_{\boldsymbol{\Theta},k}(\boldsymbol{X})[1-L_{\boldsymbol{\Theta},k}(\boldsymbol{X})]\cdot\delta_{\boldsymbol{\Theta}_k}(\boldsymbol{X}) \text{ when } j = k, \end{cases}$$

and, for $j = 1\ldots,K$, and $\boldsymbol{\Theta}_j = \mathrm{vec}(\mu_j,\boldsymbol{\Omega}_j)$,

$$\delta_{\boldsymbol{\Theta}_j}(\boldsymbol{X}) = \begin{pmatrix} \boldsymbol{\Omega}_j(\boldsymbol{X}-\boldsymbol{\mu}_j) \\ \frac{1}{2}\mathrm{vec}\left\{\boldsymbol{\Omega}_j^{-1} - (\boldsymbol{X}-\boldsymbol{\mu}_j)(\boldsymbol{X}-\boldsymbol{\mu}_j)^{\top}\right\} \end{pmatrix}.$$

Next we apply this Taylor expansion to bound I. According to $(S.2)$, we have

$$\begin{aligned} I &= \left\|\mathbb{E}\left[\boldsymbol{\Omega}_k^*(\boldsymbol{X}-\boldsymbol{\mu}_k^*)[\nabla_{\boldsymbol{\Theta}} L_{\boldsymbol{\Theta}_t,k}(\boldsymbol{X})]^{\top}(\boldsymbol{\Theta}-\boldsymbol{\Theta}^*)\right]\right\|_2^2 \\ &= \left\|\mathbb{E}\left[\boldsymbol{\Omega}_k^*(\boldsymbol{X}-\boldsymbol{\mu}_k^*)[\nabla_{\boldsymbol{\Theta}} L_{\boldsymbol{\Theta}_t,k}(\boldsymbol{X})]^{\top}\right]\right\|_2^2 \cdot \|\boldsymbol{\Theta}-\boldsymbol{\Theta}^*\|_2^2 \\ &\leq \underbrace{\sup_{t\in[0,1]} \mathbb{E}\left[\|\boldsymbol{\Omega}_k^*(\boldsymbol{X}-\boldsymbol{\mu}_k^*)\|_2^2 \cdot \|\nabla_{\boldsymbol{\Theta}} L_{\boldsymbol{\Theta}_t,k}(\boldsymbol{X})\|_2^2\right]}_{\tau_1} \cdot \|\boldsymbol{\Theta}-\boldsymbol{\Theta}^*\|_2^2. \end{aligned}$$

By the definition of $\nabla_{\boldsymbol{\Theta}} L_{\boldsymbol{\Theta}_t,k}(\boldsymbol{X})$, which equals to (S.3) with $\boldsymbol{\Theta} = \boldsymbol{\Theta}_t$, we have

$$\begin{aligned} \|\nabla_{\boldsymbol{\Theta}} L_{\boldsymbol{\Theta}_t,k}(\boldsymbol{X})\|_2^2 &= \underbrace{\sum_{j\neq k}[L_{\boldsymbol{\Theta}_t,k}(\boldsymbol{X})L_{\boldsymbol{\Theta}_t,j}(\boldsymbol{X})]^2\cdot[\delta_{\boldsymbol{\Theta}_{tj}}(\boldsymbol{X})]^{\top}\delta_{\boldsymbol{\Theta}_{tj}}(\boldsymbol{X})}_{A_1} \\ &\quad + \underbrace{\left[L_{\boldsymbol{\Theta}_t,k}(\boldsymbol{X})\big(1-L_{\boldsymbol{\Theta}_t,k}(\boldsymbol{X})\big)\right]^2\cdot[\delta_{\boldsymbol{\Theta}_{tk}}(\boldsymbol{X})]^{\top}\delta_{\boldsymbol{\Theta}_{tk}}(\boldsymbol{X})}_{A_2}. \end{aligned}$$

For each $j = 1, \ldots, K$, we define

$$W_j := \sup_{t\in[0,1]} \mathbb{E}\left\{[\delta_{\mathbf{\Theta}_{tj}}(\boldsymbol{X})]^\top \delta_{\mathbf{\Theta}_{tj}}(\boldsymbol{X}) \cdot \|\mathbf{\Omega}_k^*(\boldsymbol{X}-\boldsymbol{\mu}_k^*)\|_2^2\right\}, \tag{S.4}$$

Then

$$\tau_1 \le \sup_{t\in[0,1]} \mathbb{E}\left[\|\mathbf{\Omega}_k^*(\boldsymbol{X}-\boldsymbol{\mu}_k^*)\|_2^2 (A_1+A_2)\right]. \tag{S.5}$$

Under Condition 6, it is sufficient to get an upper bound for τ_1,

$$\begin{aligned}\tau_1 &\le \sup_{t\in[0,1]} \mathbb{E}\left[\|\mathbf{\Omega}_k^*(\boldsymbol{X}-\boldsymbol{\mu}_k^*)\|_2^2 A_1\right] + \sup_{t\in[0,1]} \mathbb{E}\left[\|\mathbf{\Omega}_k^*(\boldsymbol{X}-\boldsymbol{\mu}_k^*)\|_2^2 A_2\right] \\ &\le \sum_{j\neq k} \frac{\gamma^2}{24^2(K-1)^2 M_j}\cdot W_j + \left(\frac{\gamma}{24(K-1)\sqrt{M_k}}(K-1)\right)^2 \cdot W_k.\end{aligned}$$

It implies that

$$\tau_1 \le \frac{\gamma^2}{288}. \tag{S.6}$$

Bounding II: We can apply similar trick above to bound II. By triangle inequality, we have

$$\begin{aligned}II &\le \underbrace{\left\|\frac{1}{2}\mathbb{E}\big[D_L(\mathbf{\Theta}^*,\mathbf{\Theta})\mathbf{\Omega}_k^{*-1}\big]\right\|_F^2}_{II_1} \\ &\quad + \underbrace{\left\|\frac{1}{2}\mathbb{E}\big[D_L(\mathbf{\Theta}^*,\mathbf{\Theta})(\boldsymbol{X}-\boldsymbol{\mu}_k^*)(\boldsymbol{X}-\boldsymbol{\mu}_k^*)^\top\big]\right\|_F^2}_{II_2}.\end{aligned}$$

Apply Taylor expansion in $(S.2)$, we obtain

$$\begin{aligned}II_1 &\le \underbrace{\frac{1}{2}\mathbb{E}\left[\|\nabla_{\mathbf{\Theta}} L_{\mathbf{\Theta}_t,k}(\boldsymbol{X})\|_2^2 \|\mathbf{\Omega}_k^{*-1}\|_F^2\right]}_{\gamma_{21}} \cdot \|\mathbf{\Theta}-\mathbf{\Theta}^*\|_2^2 \\ II_2 &\le \underbrace{\frac{1}{2}\mathbb{E}\left[\|\nabla_{\mathbf{\Theta}} L_{\mathbf{\Theta}_t,k}(\boldsymbol{X})\|_2^2 \left\|(\boldsymbol{X}-\boldsymbol{\mu}_k^*)(\boldsymbol{X}-\boldsymbol{\mu}_k^*)^\top\right\|_F^2\right]}_{\gamma_{22}} \cdot \|\mathbf{\Theta}-\mathbf{\Theta}^*\|_2^2.\end{aligned}$$

Analogously to (S.4), we define

$$W_j' := \sup_{t\in[0,1]} \mathbb{E}\left\{[\delta_{\mathbf{\Theta}_{tj}}(\boldsymbol{X})]^\top \delta_{\mathbf{\Theta}_{tj}}(\boldsymbol{X}) \left\|\mathbf{\Omega}_k^{*-1}\right\|_F^2\right\}, \tag{S.7}$$

$$W_j'' := \sup_{t\in[0,1]} \mathbb{E}\left\{[\delta_{\mathbf{\Theta}_{tj}}(\boldsymbol{X})]^\top \delta_{\mathbf{\Theta}_{tj}}(\boldsymbol{X}) \left\|(\boldsymbol{X}-\boldsymbol{\mu}_k^*)(\boldsymbol{X}-\boldsymbol{\mu}_k^*)^\top\right\|_F^2\right\}. \tag{S.8}$$

for each $j = 1, \ldots, K$. Under Condition 6, we have that,

$$\tau_{21} < \frac{\gamma^2}{576}, \quad \tau_{22} < \frac{\gamma^2}{576}, \text{ and hence } \tau_2 < \frac{\gamma^2}{288}.$$

This together with (S.6) implies that $\tau = \sqrt{\tau_1+\tau_2} < \gamma/12$, namely

$$\left\|\nabla_{\mathbf{\Theta}_k'} Q(\boldsymbol{\mu}_k^*, \mathbf{\Omega}_k^*|\mathbf{\Theta}^*) - \nabla_{\mathbf{\Theta}_k'} Q(\boldsymbol{\mu}_k^*, \mathbf{\Omega}_k^*|\mathbf{\Theta})\right\|_2 \le \frac{\gamma}{12}.$$

Now we take the summation

$$\sum_{k=1}^K \left\|\nabla_{\mathbf{\Theta}_k'} Q(\boldsymbol{\mu}_k^*, \mathbf{\Omega}_k^*|\mathbf{\Theta}^*) - \nabla_{\mathbf{\Theta}_k'} Q(\boldsymbol{\mu}_k^*, \mathbf{\Omega}_k^*|\mathbf{\Theta})\right\|_2^2 \le \frac{\gamma}{12}\|\mathbf{\Theta}-\mathbf{\Theta}^*\|_2, \tag{S.9}$$

for any $\mathbf{\Theta} \in \mathbb{B}_\alpha(\mathbf{\Theta}^*)$. This ends the proof of Lemma 7. ■

S.III Proof of Lemma 9

In order to compute γ, we consider each $\mathbf{\Theta}_k = \{\mu_k, \mathbf{\Omega}_k\}$ individually. That means we prove the following part first:

$$Q_n(\mathbf{\Theta}_k'|\mathbf{\Theta}) - Q_n(\mathbf{\Theta}_k^*|\mathbf{\Theta}) - \left\langle \nabla Q_n(\mathbf{\Theta}_k^*|\mathbf{\Theta}), \mathbf{\Theta}_k' - \mathbf{\Theta}_k^*\right\rangle \le -\frac{\gamma}{2}\left\|\mathbf{\Theta}_k' - \mathbf{\Theta}_k^*\right\|_2^2,$$

where $Q_n(\mathbf{\Theta}_k|\mathbf{\Theta})$ means we set $\mathbf{\Theta}_i$ $i \neq k$ to zero.

It is sufficient to compute γ_k in (22). Remind that $\mathbf{\Theta}_k' = \mathrm{vec}(\boldsymbol{\mu}_k, \mathbf{\Omega}_k) \in \mathbb{R}^{p^2+p}$. Therefore,

$$\nabla_{\mathbf{\Theta}_k'} Q_n(\mathbf{\Theta}_k'|\mathbf{\Theta}) = ([\nabla_{\boldsymbol{\mu}_k'} Q_n(\mathbf{\Theta}_k'|\mathbf{\Theta})]^\top, [\mathrm{vec}(\nabla_{\mathbf{\Omega}_k'} Q_n(\mathbf{\Theta}_k'|\mathbf{\Theta}))]^\top)^\top, \tag{S.10}$$

with

$$\begin{aligned}\nabla_{\boldsymbol{\mu}_k'} Q_n(\mathbf{\Theta}_k'|\mathbf{\Theta}) &= \frac{1}{n}\sum_{i=1}^n \left[L_{\mathbf{\Theta},k}(\boldsymbol{x}_i)\mathbf{\Omega}_k'(\boldsymbol{x}_i - \boldsymbol{\mu}_k')\right] \\ \nabla_{\mathbf{\Omega}_k'} Q_n(\mathbf{\Theta}_k'|\mathbf{\Theta}) &= \frac{1}{2n}\sum_{i=1}^n [L_{\mathbf{\Theta},k}(\boldsymbol{x}_i)]\mathbf{\Omega}_k'^{-1} \\ &\quad - \frac{1}{2n}\sum_{i=1}^n [L_{\mathbf{\Theta},k}(\boldsymbol{x}_i)(\boldsymbol{x}_i - \boldsymbol{\mu}_k')(\boldsymbol{x}_i - \boldsymbol{\mu}_k')^\top].\end{aligned}$$

Denote $h(\boldsymbol{\mu}, \mathbf{\Omega}) := \frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu})^\top \mathbf{\Omega}(\boldsymbol{x}_i - \boldsymbol{\mu})$. According to the definition in (9), we have

$$\begin{aligned}Q_n(\mathbf{\Theta}_k'|\mathbf{\Theta}) - Q_n(\mathbf{\Theta}_k^*|\mathbf{\Theta}) &= \frac{1}{n}\sum_{i=1}^n \Big[L_{\mathbf{\Theta},k}(\boldsymbol{x}_i)\big\{\frac{1}{2}\log\det(\mathbf{\Omega}_k') \\ &\quad -\frac{1}{2}\log\det(\mathbf{\Omega}_k^*) + h(\boldsymbol{\mu}_k^*, \mathbf{\Omega}_k^*) - h(\boldsymbol{\mu}_k', \mathbf{\Omega}_k')\big\}\Big].\end{aligned}$$

This together with $(S.10)$ implies that

$$Q_n(\mathbf{\Theta}_k'|\mathbf{\Theta}) - Q_n(\mathbf{\Theta}_k^*|\mathbf{\Theta}) - \left\langle \nabla_{\mathbf{\Theta}_k'} Q_n(\mathbf{\Theta}_k^*|\mathbf{\Theta}), \mathbf{\Theta}_k' - \mathbf{\Theta}_k^*\right\rangle = I + II,$$

where

$$I = \frac{1}{n}\sum_{i=1}^{n}\left[L_{\mathbf{\Theta},k}(\boldsymbol{x}_i)\left\{h(\boldsymbol{\mu}_k^*,\mathbf{\Omega}_k^*) - h(\boldsymbol{\mu}_k',\mathbf{\Omega}_k^*)\right\}\right]$$
$$-(\boldsymbol{\mu}_k' - \boldsymbol{\mu}_k^*)^\top \nabla_{\boldsymbol{\mu}_k'} Q_n(\mathbf{\Theta}_k^*|\mathbf{\Theta}^{(t)}),$$
$$II = \frac{1}{n}\sum_{i=1}^{n}\Big[L_{\mathbf{\Theta},k}(\boldsymbol{x}_i)\Big\{\frac{1}{2}\log\det(\mathbf{\Omega}_k') - \frac{1}{2}\log\det(\mathbf{\Omega}_k^*)$$
$$+h(\boldsymbol{\mu}_k',\mathbf{\Omega}_k^*) - h(\boldsymbol{\mu}_k',\mathbf{\Omega}_k')\Big\}\Big] - [\mathrm{vec}(\mathbf{\Omega}_k' - \mathbf{\Omega}_k^*)]^\top \nabla_{\mathbf{\Omega}_k'} Q_n(\mathbf{\Theta}_k^*|\mathbf{\Theta}^{(t)}).$$

By a little algebra, we can show that

$$I = -\frac{1}{2n}\sum_{i=1}^{n} L_{\mathbf{\Theta},k}(\boldsymbol{x}_i)(\boldsymbol{\mu}_k' - \boldsymbol{\mu}_k^*)^\top \mathbf{\Omega}_k^*(\boldsymbol{\mu}_k' - \boldsymbol{\mu}_k^*).$$

Due to the positive definiteness of $\mathbf{\Omega}_k^*$, it is shown the following inequality

$$(\boldsymbol{\mu}_k' - \boldsymbol{\mu}_k^*)^\top(\mathbf{\Omega}_k^* - \sigma_{\min}(\mathbf{\Omega}_k^*)I_p)(\boldsymbol{\mu}_k' - \boldsymbol{\mu}_k^*) \geq 0$$

$$(\boldsymbol{\mu}_k' - \boldsymbol{\mu}_k^*)^\top \mathbf{\Omega}_k^*(\boldsymbol{\mu}_k' - \boldsymbol{\mu}_k^*) \geq (\boldsymbol{\mu}_k' - \boldsymbol{\mu}_k^*)^\top \sigma_{\min}(\mathbf{\Omega}_k^*) I_p(\boldsymbol{\mu}_k' - \boldsymbol{\mu}_k^*) \geq \beta_1 \left\|\boldsymbol{\mu}_k' - \boldsymbol{\mu}_k^*\right\|_2^2.$$

Substituting the above bound, it is shown that

$$I \leq -\frac{\beta_1}{2n}\sum_{i=1}^{n} L_{\mathbf{\Theta},k}(\boldsymbol{x}_i)\|\boldsymbol{\mu}_k' - \boldsymbol{\mu}_k^*\|_2^2. \tag{S.11}$$

Therefore, it remains to show that

$$II \leq -\frac{1}{2n}\sum_{i=1}^{n}\frac{L_{\mathbf{\Theta},k}(\boldsymbol{x}_i)}{2(\beta_2+2\alpha)^2}\|\mathrm{vec}(\mathbf{\Omega}_k' - \mathbf{\Omega}_k^*)\|_2^2. \tag{S.12}$$

Note that, in order to show ($S.12$), it is equivalent to deriving the strong concavity parameter of $g(\mathbf{\Omega}_k)$, where

$$g(\mathbf{\Omega}_k) \coloneqq \frac{1}{n}\sum_{i=1}^{n}\left[L_{\mathbf{\Theta},k}(\boldsymbol{x}_i)\left\{\frac{1}{2}\log\det(\mathbf{\Omega}_k) - h(\boldsymbol{\mu}_k',\mathbf{\Omega}_k)\right\}\right].$$

To see it, finding the strong concavity parameter of $g(\mathbf{\Omega}_k)$ aims to compute ρ_k such that, for any $\mathbf{\Omega}_k', \mathbf{\Omega}_k^* \in \mathcal{B}_\alpha(\mathbf{\Omega}_k^*)$,

$$g(\mathbf{\Omega}_k') - g(\mathbf{\Omega}_k^*) - \left\langle \mathrm{vec}\left(\nabla g(\mathbf{\Omega}_k^*)\right), \mathrm{vec}(\mathbf{\Omega}_k' - \mathbf{\Omega}_k^*)\right\rangle \leq -\rho_k/2 \cdot \|\mathbf{\Omega}_k' - \mathbf{\Omega}_k^*\|_F^2,$$

where the left hand side is exactly II. According to Taylor expansion, we can expand $g(\mathbf{\Omega}_k')$ around $\mathbf{\Omega}_k^*$ and obtain

$$\begin{aligned} g(\mathbf{\Omega}_k') &= g(\mathbf{\Omega}_k^*) + \left\langle \mathrm{vec}(\nabla g(\mathbf{\Omega}_k^*), \mathrm{vec}(\mathbf{\Omega}_k' - \mathbf{\Omega}_k^*)\right\rangle \\ &\quad + \frac{1}{2}\left[\mathrm{vec}(\mathbf{\Omega}_k' - \mathbf{\Omega}_k^*)\right]^\top \nabla^2 g(\mathbf{Z})\left[\mathrm{vec}(\mathbf{\Omega}_k' - \mathbf{\Omega}_k^*)\right], \end{aligned}$$

where $\mathbf{Z} = t\mathbf{\Omega}_k' + (1-t)\mathbf{\Omega}_k^*$ with $t \in [0,1]$. For any two matrices $\mathbf{A}, \mathbf{B}$, we write $\mathbf{A} \succeq \mathbf{B}$ if $\mathbf{A} - \mathbf{B}$ is positive semi-definite. We denote 1_p as the identity matrix with dimension $p \times p$. And $\sigma_i(A)$ is the i-th eigenvalue of matrix A. Therefore, if we can show that $-\nabla^2 g(\mathbf{Z}) \succeq m\, 1_p$, i.e., the minimal eigenvalue value $\sigma_{\min}(-\nabla^2 g(\mathbf{Z})) \geq m$, for some positive $m \in \mathbb{R}$, then we have the strongly concavity parameter $\rho_k = m$. By the definition, we have $\nabla^2 g(\mathbf{\Omega}_k^*) = -\frac{1}{2n}\sum_{i=1}^n L_{\mathbf{\Theta},k}(\boldsymbol{x}_i)[\mathbf{\Omega}_k^*]^{-1} \otimes [\mathbf{\Omega}_k^*]^{-1}$. Denote $\widetilde{\Delta} = \mathbf{\Omega}_k' - \mathbf{\Omega}_k^*$. We obtain

$$-\nabla^2 g(\mathbf{Z}) = \frac{1}{2n}\sum_{i=1}^{n} L_{\mathbf{\Theta},k}(\boldsymbol{x}_i)\left(\mathbf{\Omega}_k^* + t\widetilde{\Delta}\right)^{-1} \otimes \left(\mathbf{\Omega}_k^* + t\widetilde{\Delta}\right)^{-1}.$$

According to Theorem 4.2.1 2 in Horn and Johnson (1988), for any two matrices $\mathbf{A}, \mathbf{B}$, the minimal eigenvalue value of $\mathbf{A} \otimes \mathbf{B}$ equals the products of the minimal eigenvalue values of $\mathbf{A}$ and $\mathbf{B}$. Therefore, we have $\sigma_{\min}\left(\mathbf{A}^{-1} \otimes \mathbf{A}^{-1}\right) = \left[\sigma_{\min}(\mathbf{A}^{-1})\right]^2 = [\sigma_{\max}(\mathbf{A})]^{-2} = \|\mathbf{A}\|_2^{-2}$, where $\|\mathbf{A}\|_2$ refers to the spectral norm of matrix $\mathbf{A}$. Hence,

$$\begin{aligned} \sigma_{\min}(-\nabla^2 g(\mathbf{Z})) &= \frac{1}{2n}\sum_{i=1}^{n} L_{\mathbf{\Theta},k}(\boldsymbol{x}_i)\|\mathbf{\Omega}_k^* + t\widetilde{\Delta}\|_2^{-2} \\ &\geq \frac{1}{2n}\sum_{i=1}^{n} L_{\mathbf{\Theta},k}(\boldsymbol{x}_i)\left[\|\mathbf{\Omega}_k^*\|_2 + \|t\widetilde{\Delta}\|_2\right]^{-2}. \end{aligned}$$

As $\|\mathbf{\Theta}' - \mathbf{\Theta}^*\| \leq 2\alpha$, $\|\mathbf{\Omega}_k' - \mathbf{\Omega}_k^*\|_2 \leq \|\mathbf{\Theta}' - \mathbf{\Theta}^*\|_2 \leq 2\alpha$. Therefore,

$$\begin{aligned} \sigma_{\min}(-\nabla^2 g(\mathbf{Z})) &\geq \frac{1}{2n}\sum_{i=1}^{n} L_{\mathbf{\Theta},k}(\boldsymbol{x}_i)\left[\|\mathbf{\Omega}_k^*\|_2 + 2\alpha\right]^{-2} \\ &\geq \frac{1}{2n}\sum_{i=1}^{n} L_{\mathbf{\Theta},k}(\boldsymbol{x}_i)\left(\beta_2 + 2\alpha\right)^{-2}, \end{aligned}$$

which implies ($S.12$). Putting the upper bound of I and II together,

$$I + II \leq \underbrace{-\frac{1}{2n}\sum_{i=1}^{n} L_{\mathbf{\Theta},k}(\boldsymbol{x}_i)}_{(a)} \cdot \min\left\{\beta_1, \frac{1}{2(\beta_2+2\alpha)^2}\right\}\|\mathbf{\Theta}_k' - \mathbf{\Theta}_k^*\|_2^2. \tag{S.13}$$

However, (a) is a random term but we require a non-random strong concavity parameter. Thus a concentration bound will be applied on it. $\{L_{\mathbf{\Theta},k}(\boldsymbol{x}_i), i = 1, \ldots, n\}$ are independent random variables with $0 \leq L_{\mathbf{\Theta},k}(\boldsymbol{x}_i) \leq 1$. After applying a basic Hoeffding's inequality, we have

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n} L_{\mathbf{\Theta},k}(\boldsymbol{x}_i) - \mathbb{E}[L_{\mathbf{\Theta},k}(\boldsymbol{X})]\right| \leq t\right) \geq 1 - 2e^{-2nt^2},$$

which implies

$$\left|\frac{1}{n}\sum_{i=1}^{n} L_{\mathbf{\Theta},k}(\boldsymbol{x}_i) - \mathbb{E}[L_{\mathbf{\Theta},k}(\boldsymbol{X})]\right| \leq \sqrt{\frac{1}{2}\log\frac{2K}{\delta}}\sqrt{\frac{1}{n}},$$

with probability at least $1-\delta/K$. As $\sqrt{\log(2K/\delta)/2n} = o(1)$, there exists some constant c such that

$$\sqrt{\frac{\log 2K}{2\delta n}} - \mathbb{E}[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})] \leq -c,$$

when n is large enough. Then plugging it into (S.13),

$$I + II \leq -\frac{1}{2}c \cdot \min\left\{\beta_1, \frac{1}{2(\beta_2+2\alpha)^2}\right\} \|\boldsymbol{\Theta}_k' - \boldsymbol{\Theta}_k^*\|_2^2,$$

with probability at least $1-\delta/K$, where

$$\gamma = c\min\left\{\beta_1, \frac{1}{2(\beta_2+2\alpha)^2}\right\}.$$

Once the individual strong concavity parameter is computed, we can simply take the summation from 1 to K:

$$\sum_{k=1}^{K} Q_n(\boldsymbol{\Theta}_k'|\boldsymbol{\Theta}) - Q_n(\boldsymbol{\Theta}_k^*|\boldsymbol{\Theta}) - \left\langle \nabla Q_n(\boldsymbol{\Theta}_k^*|\boldsymbol{\Theta}), \boldsymbol{\Theta}_k' - \boldsymbol{\Theta}_k^* \right\rangle \leq -\frac{1}{2}\sum_{k=1}^{K}\gamma \left\|\boldsymbol{\Theta}_k' - \boldsymbol{\Theta}_k^*\right\|_2^2$$

which implies

$$Q_n(\boldsymbol{\Theta}'|\boldsymbol{\Theta}) - Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) - \left\langle \nabla Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}), \boldsymbol{\Theta}' - \boldsymbol{\Theta}^* \right\rangle \leq -\frac{1}{2}\gamma \left\|\boldsymbol{\Theta}' - \boldsymbol{\Theta}^*\right\|_2^2$$

with probability at least $1-\delta$. This ends the proof of Lemma 9. ■

S.IV A Key Lemma for Proving Corollary 18

The next lemma computes the statistical errors in Condition 10 for our SCAN penalty and provides explicit forms of the corresponding $\varepsilon_1, \varepsilon_2$ and δ_1, δ_2.

Lemma S.1 *Suppose that Condition 16, 17 hold, then Condition 10 is satisfied for SCAN penalty with*

$$\varepsilon_1 = (CK\|\boldsymbol{\Omega}^*\|_\infty + C'K^{1.5})\sqrt{\frac{\log p + \log(e/\delta)}{n}}, \delta_1 = (18K^2+6K)\delta, \tag{S.14}$$

$$\varepsilon_2 = C''\sqrt{p}\sqrt{\frac{K^3\left(\log p + \log(e/\delta)\right)}{n}}, \delta_2 = (8K^2+2K)\delta, \tag{S.15}$$

for some absolute constant $C, C', C'' > 0$. Here $\|\boldsymbol{\Omega}^*\|_\infty$ is the overall max induced norm defined as $\|\boldsymbol{\Omega}^*\|_\infty = \max_{k\in[K]}\|\boldsymbol{\Omega}_k^*\|_\infty$.

In Lemma S.1, the number of clusters K is allowed to grow with the sample size n and the dimension p. The diverging rate of K controls the convergence probability at each iteration and is upper bounded to ensure that the statistical errors hold with a high probability tending to 1 with a proper choice of δ, e.g., $\delta = 1/p$.

Proof of Lemma S.1: For the first part of this proof, we focus on the upper bound of $\|\nabla Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) - \nabla Q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta})\|_{\mathcal{P}^*}$. Recall that

$$\begin{aligned}\nabla Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) - \nabla Q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) &= \begin{pmatrix} \nabla_{\boldsymbol{\Theta}_1^*} Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) - \nabla_{\boldsymbol{\Theta}_1^*} Q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) \\ \vdots \\ \nabla_{\boldsymbol{\Theta}_K^*} Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) - \nabla_{\boldsymbol{\Theta}_K^*} Q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) \end{pmatrix} \\ &= \begin{pmatrix} \nabla_{\boldsymbol{\mu}_1^*} Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) - \nabla_{\boldsymbol{\mu}_1^*} Q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) \\ \mathrm{vec}\left\{\nabla_{\boldsymbol{\Omega}_1^*} Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) - \nabla_{\boldsymbol{\Omega}_1^*} Q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta})\right\}^\top \\ \vdots \\ \nabla_{\boldsymbol{\mu}_K^*} Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) - \nabla_{\boldsymbol{\mu}_K^*} Q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) \\ \mathrm{vec}\left\{\nabla_{\boldsymbol{\Omega}_K^*} Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) - \nabla_{\boldsymbol{\Omega}_K^*} Q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta})\right\}^\top \end{pmatrix}. \end{aligned} \tag{S.16}$$

For simplicity, we define $h_{\mu_k}(\boldsymbol{\Theta}^*) = \nabla_{\boldsymbol{\mu}_k^*} Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) - \nabla_{\boldsymbol{\mu}_k^*} Q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta})$ and $h_{\boldsymbol{\Omega}_k^*}(\boldsymbol{\Theta}^*) = \nabla_{\boldsymbol{\Omega}_k^*} Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) - \nabla_{\boldsymbol{\Omega}_k^*} Q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta})$. Then from the definition of dual norm $\mathcal{P}^*$ (31), we can have

$$\begin{aligned}\|\nabla Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) - \nabla Q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta})\|_{\mathcal{P}^*} &\leq M_1 \max_{k\in[K]} \underbrace{\|h_{\mu_k}(\boldsymbol{\Theta}^*)\|_\infty}_{I} \\ +M_2 \max_{k\in[K]} \underbrace{\left\|h_{\boldsymbol{\Omega}_k^*}(\boldsymbol{\Theta}^*)\right\|_{\max}}_{II} &+ M_3 \max_{i,j} \underbrace{\left\|\left[h_{\boldsymbol{\Omega}_k^*}(\boldsymbol{\Theta}^*)\right]_{ij}, \ldots, \left[h_{\boldsymbol{\Omega}_k^*}(\boldsymbol{\Theta}^*)\right]_{ij}\right\|_2}_{III},\end{aligned}$$

which are corresponding to the penalty on element-wise cluster means, element-wise precision matrices and group structures of multiple precision matrices, respectively.

Bounding Statistical Error for k-th Cluster Mean: Referring to the proof in Lemma 5,

$$h_{\mu_k^*}(\boldsymbol{\Theta}^*) = \frac{1}{n}\sum_{i=1}^{n} L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)\boldsymbol{\Omega}_k^*(\boldsymbol{x}_i - \boldsymbol{\mu}_k^*) - \mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})\boldsymbol{\Omega}_k^*(\boldsymbol{X} - \boldsymbol{\mu}_k^*)\right].$$

Note that $\|\boldsymbol{\Omega}_k^*\|_\infty$ is a scalar. By using triangle inequality, we simplify I by two parts:

$$\begin{aligned} I &\leq \|\boldsymbol{\Omega}_k^*\|_\infty \left\|\frac{1}{n}\sum_{i=1}^{n} L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)(\boldsymbol{x}_i - \boldsymbol{\mu}_k^*) - \mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})(\boldsymbol{X} - \boldsymbol{\mu}_k^*)\right]\right\|_\infty \\ &\leq \|\boldsymbol{\Omega}_k^*\|_\infty \underbrace{\left\|\frac{1}{n}\sum_{i=1}^{n} L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)\boldsymbol{x}_i - \mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})\boldsymbol{X}\right]\right\|_\infty}_{I_1} \\ &\quad + \|\boldsymbol{\Omega}_k^*\|_\infty \underbrace{\left\|\left(\frac{1}{n}\sum_{i=1}^{n} L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i) - \mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})\right]\right)\boldsymbol{\mu}_k^*\right\|_\infty}_{I_2}. \end{aligned}$$

Bounding I_1: Denote

$$\zeta = \frac{1}{n}\sum_{i=1}^{n} L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)\boldsymbol{x}_i - \mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})\boldsymbol{X}\right]$$

For $\zeta \in \mathbb{R}^p$, we consider the j-th coordinate ζ_j of ζ

$$\zeta_j = \frac{1}{n}\sum_{i=1}^{n} L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)x_{ij} - \mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})X_j\right]. \tag{S.17}$$

We introduce a set of missing data $\{c_i, i = 1, \ldots, n\}$, which are independent copies of random variable c. The pair $(\boldsymbol{x}_i, c_i)$ are the independent copy of $(\boldsymbol{X}, c)$. Here c takes a value from the set $\{1, \ldots, K\}$, where $c = k'$ indicates that $\boldsymbol{X}$ was generated by the k'-th mixture component. In another word, the conditional distribution of $\boldsymbol{X}$ is defined below:

$$\boldsymbol{X}|c = k' \sim \mathcal{N}(\boldsymbol{\mu}^*_{k'}, \boldsymbol{\Sigma}^*_{k'})$$

$$\mathbb{P}(c = k') = \pi_{k'},\ \sum_{k'}^{K} \pi_{k'} = 1.$$

This is the usual choice of missing data in EM approaches to mixture modeling. The quantity $(\boldsymbol{x}_i, c_i)$ is referred to as the completed data. Now by the assumption, the j-th coordinate x_{ij} of $\boldsymbol{x}_i$ can be rewritten as the form below:

$$x_{ij} = \sum_{k'=1}^{K} I\{c_i = k'\}(\mu^*_{k'j} + V_{k'j}),\ j \in [p] \tag{S.18}$$

where $\mu^*_{k'j}$ is the j-th coordinate of the true cluster mean $\boldsymbol{\mu}^*_{k'}$ and $V_{k'j} \sim \mathcal{N}(0, \boldsymbol{\Sigma}^*_{k'jj})$. Plugging (S.18) into (S.17), it suffices to bound ζ_j.

$$\begin{aligned}
|\zeta_j| &\leq \left|\frac{1}{n}\sum_{i=1}^{n}\sum_{k'=1}^{K} L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)I\{c_i = k'\}\mu^*_{k'j} - \mathbb{E}\left[\sum_{k'=1}^{K} L_{\boldsymbol{\Theta},k}(\boldsymbol{X})I\{c = k'\}\mu^*_{k'j}\right]\right| \\
&+ \left|\frac{1}{n}\sum_{i=1}^{n}\sum_{k'=1}^{K} L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)I\{c_i = k'\}V^*_{k'j} - \mathbb{E}\left[\sum_{k'=1}^{K} L_{\boldsymbol{\Theta},k}(\boldsymbol{X})I\{c = k'\}V^*_{k'j}\right]\right| \\
&\leq \sum_{k'=1}^{K} \underbrace{\left|\frac{1}{n}\sum_{i=1}^{n} L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)I\{c_i = k'\}\mu^*_{k'j} - \mathbb{E}\Big[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})I\{c = k'\}\mu^*_{k'j}\Big]\right|}_{\zeta_{j_1}} \\
&+ \sum_{k'=1}^{K} \underbrace{\left|\frac{1}{n}\sum_{i=1}^{n} L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)I\{c_i = k'\}V^*_{k'j} - \mathbb{E}\Big[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})I\{c = k'\}V^*_{k'j}\Big]\right|}_{\zeta_{j_2}}.
\end{aligned}$$

We bound ζ_{j_1} first. Based on the fact that $|L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)I\{c_i = k'\}\mu^*_{k'j}| \leq |\mu^*_{k'j}| \leq \|\boldsymbol{\mu}^*_{k'}\|_\infty$ almost surely it can show that $L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)I\{c_i = k'\}\mu^*_{k'j}$ is a sub-gaussian random variable with norm $\|\boldsymbol{\mu}^*_{k'}\|_\infty$. Following the Example 5.8 in Vershynin (2012), $\|L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)I\{c_i = k'\}\mu^*_{k'j}\|_{\psi_2} \leq \|\boldsymbol{\mu}^*_{k'}\|_\infty$ where $\|\cdot\|_{\psi_2}$ is defined as sub-Gaussian norm. According to supporting Lemma S.5

$$\left\|L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)I\{c_i = k'\}\mu^*_{k'j} - \mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})I\{c = k'\}\mu^*_{k'j}\right]\right\|_{\psi_2} \leq 2\left\|\boldsymbol{\mu}^*_{k'}\right\|_\infty.$$

The standard concentration result in supporting Lemma S.6 yields that for every $t \geq 0$ and some constant D_1,

$$\mathbb{P}\left(|\zeta_{j_1}| \geq t\right) \leq e\exp\left(-\frac{D_1 n t^2}{4\|\boldsymbol{\mu}^*_{k'}\|^2_\infty}\right),$$

which implies that, with probability at least $1 - \delta$,

$$|\zeta_{j_1}| \leq \sqrt{\frac{4}{D_1}}\|\boldsymbol{\mu}^*_{k'}\|_\infty\sqrt{\frac{\log(e/\delta)}{n}}. \tag{S.19}$$

Now we start to bound ζ_{j_2}. The fact that $L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)I\{c_i = k'\} \leq 1$ shows that it is a sub-gaussian random variable with norm $\|L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)I\{c_i = k'\}\|_{\psi_2} \leq 1$. $V^*_{k'j}$ is a Gaussian random variable so that it is also a sub-gaussian random variable with norm $\|V^*_{k'j}\|_{\psi_2} \leq (\|\boldsymbol{\Sigma}^*_{k'}\|_{\max})^{1/2}$. Then using the result in supporting Lemma S.4, $L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)I\{c_i = k'\}V^*_{k'j}$ is sub-exponential random variable. Moreover, there exists constant D_2 such that

$$\left\|L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)I\{c_i = k'\}V^*_{k'j}\right\|_{\psi_1} \leq D_2\Big(\|\boldsymbol{\Sigma}^*_{k'}\|_{\max}\Big)^{1/2}.$$

Supporting lemma S.5 implies

$$\left\|L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)I\{c_i = k'\}V^*_{k'j} - \mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})I\{c = k'\}V^*_{k'j}\right]\right\|_{\psi_1} \leq 2D_2\Big(\|\boldsymbol{\Sigma}^*_{k'}\|_{\max}\Big)^{1/2}.$$

Following the concentration inequality of sub-exponential random variables in supporting Lemma S.7, there exists some constant D_3 such that the following inequality

$$\mathbb{P}\Big(|\zeta_{j_2}| \geq t\Big) \leq 2\exp\left(-D_3\min\left\{\frac{t^2}{4D_2^2\|\boldsymbol{\Sigma}^*_{k'}\|_{\max}}, \frac{t}{2D_2(\|\boldsymbol{\Sigma}^*_{k'}\|_{\max})^{1/2}}\right\}n\right),$$

holds every $t \geq 0$. For sufficient small t, it reduces to

$$\mathbb{P}\Big(|\zeta_{j_2}| \geq t\Big) \leq 2\exp\left(-D_3\frac{nt^2}{4D_2\|\boldsymbol{\Sigma}^*_{k'}\|_{\max}}\right),$$

which implies that

$$|\zeta_{j_2}| \leq \sqrt{\frac{4D_2}{D_3}}(\|\boldsymbol{\Sigma}^*_{k'}\|_{\max})^{1/2}\sqrt{\frac{\log(2/\delta)}{n}}, \tag{S.20}$$

with probability at least $1 - \delta$.

Adding (S.19) and (S.20) together, we have

$$\begin{aligned}
|\zeta_{j_1}| + |\zeta_{j_2}| &\leq \sqrt{\frac{4}{D_1}}\|\boldsymbol{\mu}^*_{k'}\|_\infty\sqrt{\frac{\log(e/\delta)}{n}} + \sqrt{\frac{4D_2}{D_3}}(\|\boldsymbol{\Sigma}^*_{k'}\|_{\max})^{1/2}\sqrt{\frac{\log(2/\delta)}{n}} \\
&\leq \sqrt{\frac{4}{D}}\left(\|\boldsymbol{\mu}^*_{k'}\|_\infty + (\|\boldsymbol{\Sigma}^*_{k'}\|_{\max})^{1/2}\right)\sqrt{\frac{\log(e/\delta)}{n}},
\end{aligned}$$

by taking $D = \min\{D_1, D_3/D_2\}$, with at least probability $1 - 2\delta$. Therefore, it's sufficient to bound $|\zeta_j|$ by

$$|\zeta_j| \leq \sqrt{\frac{4}{D}}\sum_{k'=1}^{K}\left(\|\boldsymbol{\mu}^*_{k'}\|_\infty + (\|\boldsymbol{\Sigma}^*_{k'}\|_{\max})^{1/2}\right)\sqrt{\frac{\log(e/\delta)}{n}},$$

with at least probability $1-2K\delta$. Taking the union bound over p coordinates, we obtain

$$I_1 \leq \sqrt{\frac{4}{D}} \sum_{k'=1}^{K} \left(\|\boldsymbol{\mu}^*_{k'}\|_\infty + (\|\boldsymbol{\Sigma}^*_{k'}\|_{\max})^{1/2} \right) \sqrt{\frac{\log(e/\delta) + \log p}{n}}, \tag{S.21}$$

with at least probability $1-2K\delta$.

Bounding I_2: Recall that

$$I_2 = \left\| \left(\frac{1}{n} \sum_{i=1}^{n} L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i) - \mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X}) \right] \right) \boldsymbol{\mu}^*_k \right\|_\infty \leq \left| \frac{1}{n} \sum_{i=1}^{n} L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i) - \mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X}) \right] \right| \|\boldsymbol{\mu}^*_k\|_\infty .$$

$\{L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i) | i = 1, \dots n\}$ are bounded independent random variables within interval between 0 and 1. Then it follows Hoeffding's inequality in supporting Lemma S.8 that

$$\mathbb{P}\left(\left| \frac{1}{n} \sum_{i=1}^{n} L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i) - \mathbb{E}[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})] \right| \leq t \right) \geq 1 - 2e^{-2nt^2},$$

which implies

$$\left| \frac{1}{n} \sum_{i=1}^{n} L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i) - \mathbb{E}[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})] \right| \leq \sqrt{\frac{1}{2} \log \frac{2}{\delta}} \cdot \sqrt{\frac{1}{n}}, \tag{S.22}$$

with probability at least $1-\delta$. Combining with the reminder term $\|\boldsymbol{\mu}^*_k\|$,

$$I_2 \leq \sqrt{\frac{1}{2} \log \frac{2}{\delta}} \cdot \sqrt{\frac{1}{n}} \|\boldsymbol{\mu}^*_k\|_\infty. \tag{S.23}$$

Note that the bound in (S.21) is $O_P((\log p/n)^{1/2})$ while the bound in (S.23) is $O_P((1/n)^{1/2})$, there exists some constant D_4 such that $I_2 \leq D_4 I_1$. Consequently, we conclude that I is upper bounded by

$$I \leq (1+D_4)\|\boldsymbol{\Omega}^*_k\|_\infty \sqrt{\frac{4}{D}} \sum_{k'=1}^{K} \left(\|\boldsymbol{\mu}^*_{k'}\|_\infty + (\|\boldsymbol{\Sigma}^*_{k'}\|_{\max})^{1/2} \right) \sqrt{\frac{\log(e/\delta) + \log p}{n}},$$

with probability at least $1-(2K+1)\delta$. For simplicity, let

$$\varphi_K = \sum_{k'=1}^{K} \left(\|\boldsymbol{\mu}^*_{k'}\|_\infty + (\|\boldsymbol{\Sigma}^*_{k'}\|_{\max})^{1/2} \right), \; C_1 = \sqrt{\frac{4(1+D_4)^2}{D}}. \tag{S.24}$$

Applying union bound,

$$\max_{k \in [K]} I \leq C_1 \|\boldsymbol{\Omega}^*\|_\infty \varphi_K \sqrt{\frac{\log p + \log(e/\delta)}{n}}, \tag{S.25}$$

with probability at least $1-K(2K+1)\delta$.

Bounding Statistical Error for k-th Precision Matrix: Referring to the proof in Lemma 5,

$$\begin{aligned} h_{\boldsymbol{\Omega}^*_k}(\boldsymbol{\Theta}^*) &= \frac{1}{2n} \sum_{i=1}^{n} L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i) \boldsymbol{\Sigma}^*_k - \frac{1}{2n} \sum_{i=1}^{n} L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)(\boldsymbol{x}_i - \boldsymbol{\mu}^*_k)(\boldsymbol{x}_i - \boldsymbol{\mu}^*_k)^\top \\ &- \frac{1}{2}\mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X}) \right] \boldsymbol{\Sigma}^*_k + \frac{1}{2}\mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})(\boldsymbol{X} - \boldsymbol{\mu}^*_k)(\boldsymbol{X} - \boldsymbol{\mu}^*_k)^\top \right]. \end{aligned}$$

Now we get an explicit from for $h_{\boldsymbol{\Omega}^*_k}(\boldsymbol{\Theta}^*)$. Then II is decomposed as below:

$$\begin{aligned} II \leq & \underbrace{\left\| \frac{1}{2} \left(\frac{1}{n} \sum_{i=1}^{n} L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i) \boldsymbol{\Sigma}^*_k - \mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X}) \boldsymbol{\Sigma}^*_k \right] \right) \right\|_{\max}}_{II_1} \\ + & \underbrace{\left\| \frac{1}{2} \left(\frac{1}{n} \sum_{i=1}^{n} L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)(\boldsymbol{x}_i - \boldsymbol{\mu}^*_k)(\boldsymbol{x}_i - \boldsymbol{\mu}^*_k)^\top - \mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})(\boldsymbol{X} - \boldsymbol{\mu}^*_k)(\boldsymbol{X} - \boldsymbol{\mu}^*_k)^\top \right] \right) \right\|_{\max}}_{II_2}. \end{aligned}$$

The first term is easy to deal with: since $\frac{1}{n}\sum_{i=1}^{n} L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i) - \mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X}) \right]$ is scalar by the definition of $L_{\boldsymbol{\Theta},k}(\boldsymbol{X})$ we can pull it out of the norm. Combining with the result in (S.22), the first term is upper bounded by

$$II_1 \leq \|\boldsymbol{\Sigma}^*_k\|_{\max} \sqrt{\frac{1}{2} \log \frac{2}{\delta}} \cdot \sqrt{\frac{1}{n}}, \tag{S.26}$$

with probability at least $1-\delta$.

For the second term II_2, it can be decomposed as four following terms:

$$\begin{aligned} II_2 \leq & \underbrace{\left\| \frac{1}{2} \left(\frac{1}{n} \sum_{i=1}^{n} L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i) \boldsymbol{x}_i \boldsymbol{x}_i^\top - \mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X}) \boldsymbol{X} \boldsymbol{X}^\top \right] \right) \right\|_{\max}}_{II_{21}} \\ + & \underbrace{\left\| \frac{1}{2} \left(\frac{1}{n} \sum_{i=1}^{n} L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i) \boldsymbol{x}_i \boldsymbol{\mu}^{*T}_k - \mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X}) \boldsymbol{X} \boldsymbol{\mu}^{*\top}_k \right] \right) \right\|_{\max}}_{II_{22}} \\ + & \underbrace{\left\| \frac{1}{2} \left(\frac{1}{n} \sum_{i=1}^{n} L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i) \boldsymbol{\mu}^*_k \boldsymbol{x}_i^\top - \mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X}) \boldsymbol{\mu}^*_k \boldsymbol{X}^\top \right] \right) \right\|_{\max}}_{II_{23}} \\ + & \underbrace{\left\| \frac{1}{2} \left(\frac{1}{n} \sum_{i=1}^{n} L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i) \boldsymbol{\mu}^*_k \boldsymbol{\mu}^{*\top}_k - \mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X}) \boldsymbol{\mu}^*_k \boldsymbol{\mu}^{*\top}_k \right] \right) \right\|_{\max}}_{II_{24}}. \end{aligned}$$

For the bound of II_{22} and II_{23}, we can just simply pull the $\boldsymbol{\mu}_k^*$ out, which implies

$$\begin{aligned} II_{22} &= \left\| \frac{1}{2}\left(\frac{1}{n}\sum_{i=1}^n L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)\boldsymbol{x}_i - \mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})\boldsymbol{X}\right]\right)\boldsymbol{\mu}_k^{*\top} \right\|_{\max} \\ &\leq \left\| \frac{1}{2}\left(\frac{1}{n}\sum_{i=1}^n L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)\boldsymbol{x}_i - \mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})\boldsymbol{X}\right]\right)\right\|_{\infty} \|\boldsymbol{\mu}_k^*\|_{\infty} \\ &\overset{(a)}{\leq} \sqrt{\frac{4}{D}}\|\boldsymbol{\mu}_k^*\|_{\infty}\varphi_K\sqrt{\frac{\log(e/\delta)+\log p}{n}}, \end{aligned} \tag{S.27}$$

with probability at least $1-2K\delta$, where (a) follows (S.21).

Next we turn to bound II_{21}. Expand $\boldsymbol{x}_i\boldsymbol{x}_i^\top$ to matrix form for convenient use

$$\boldsymbol{x}_i\boldsymbol{x}_i^\top = \begin{pmatrix} x_{i1}x_{i1} & \dots & x_{i1}x_{ip} \\ \vdots & \ddots & \vdots \\ x_{ip}x_{i1} & \dots & x_{ip}x_{ip} \end{pmatrix}.$$

Since we require a matrix max norm here, it suffices to bound II_{21} individually, namely

$$\zeta_{jj'} = \frac{1}{2}\left(\frac{1}{n}\sum_{i=1}^n L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)x_{ij}x_{ij'} - \mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})X_jX_{j'}\right]\right).$$

Recall in (S.18) the j-th coordinate of $\boldsymbol{x}_i$ could be expressed as

$$x_{ij} = \sum_{k'=1}^K I\{c_i = k'\}(\mu_{k'j}^* + V_{k'j}).$$

By straightforward algebra,

$$\begin{aligned} x_{ij}x_{ij'} &= \sum_{k'=1}^K I\{c_i=k'\}(\mu_{k'j}^*+V_{k'j})\cdot\sum_{k'=1}^K I\{c_i=k'\}(\mu_{k'j'}^*+V_{k'j'}) \\ &\overset{(a)}{=} \sum_{k'=1}^K I\{c_i=k'\}^2(\mu_{k'j}^*+V_{k'j})(\mu_{k'j'}^*+V_{k'j'}) \\ &= \sum_{k'=1}^K I\{c_i=k'\}\left(\mu_{k'j}^*\mu_{k'j'}^* + \mu_{k'j}^*V_{k'j'} + V_{k'j}\mu_{k'j'}^* + V_{k'j}V_{k'j'}\right), \end{aligned}$$

where (a) follows the fact that $I\{c_i=k\}I\{c_i=k'\}=0$ for any $k\neq k'$. Consequently, we divide $\zeta_{jj'}$ into four parts:

$$\zeta_{jj'} = \frac{1}{2}\sum_{k'=1}^K\left(\zeta_{jj'}(\mu_{k'j}^*\mu_{k'j'}^*) + \zeta_{jj'}(\mu_{k'j}^*V_{k'j'}) + \zeta_{jj'}(V_{k'j}\mu_{k'j'}^*) + \zeta_{jj'}(V_{k'j}V_{k'j'})\right),$$

where

$$\begin{aligned} \zeta_{jj'}(\mu_{k'j}^*\mu_{k'j'}^*) &= \frac{1}{n}\sum_{i=1}^n L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)I\{c_i=k'\}\mu_{k'j}^*\mu_{k'j'}^* \\ &\quad -\mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})I\{c=k'\}\mu_{k'j}^*\mu_{k'j'}^*\right]. \end{aligned}$$

Taking the supreme over set $[p]$ in terms of p,p',

$$\begin{aligned} \sup_{j,j'\in[p]}|\zeta_{jj'}| &\leq \sum_{k'=1}^K \underbrace{\left(\sup_{j,j'\in[p]}|\zeta_{jj'}(\mu_{k'j}^*\mu_{k'j'}^*)|\right)}_{(i)} + \sum_{k'=1}^K \underbrace{\left(\sup_{j,j'\in[p]}|\zeta_{jj'}(\mu_{k'j}^*V_{k'j'})|\right)}_{(ii)} \\ &\quad + \sum_{k'=1}^K \underbrace{\left(\sup_{j,j'\in[p]}|\zeta_{jj'}(V_{k'j}\mu_{k'j'}^*)|\right)}_{(iii)} + \sum_{k'=1}^K \underbrace{\left(\sup_{j,j'\in[p]}|\zeta_{jj'}(V_{k'j}V_{k'j'})|\right)}_{(iv)}. \end{aligned}$$

We will bound (i), (ii), (iii) and (iv) sequentially. $L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)I\{c_i=k'\}\mu_{k'j}^*\mu_{k'j'}^*$ is a sub-gaussian random variable with

$$\|L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)I\{c_i=k'\}\mu_{k'j}^*\mu_{k'j'}^*\|_{\psi_2} \leq \|\boldsymbol{\mu}_{k'}^*\|_\infty^2.$$

According to supporting Lemma S.5,

$$\left\|L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)I\{c_i=k'\}\mu_{k'j}^*\mu_{k'j'}^* - \mathbb{E}[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})I\{c=k'\}\mu_{k'j}^*\mu_{k'j'}^*]\right\|_{\psi_2} \leq 2\|\boldsymbol{\mu}_{k'}^*\|_\infty^2.$$

Applying concentration inequality in supporting Lemma S.6 yields that

$$\mathbb{P}\left(|\zeta_{jj'}(\mu_{k'j}^*\mu_{k'j'}^*)| \leq t\right) \geq 1 - e\exp\left(-\frac{D_4nt^2}{4\|\boldsymbol{\mu}_{k'}^*\|_\infty^4}\right), \tag{S.28}$$

for any $t>0$ and some constant D_4. After properly choosing t,

$$(i) \leq \sqrt{\frac{4}{D_4}}\|\boldsymbol{\mu}_{k'}^*\|_\infty^2\sqrt{\frac{\log p+\log(e/\delta)}{n}}, \tag{S.29}$$

with probability at least $1-\delta$. Note that both $L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)I\{c_i=k'\}\mu_{k'j}^*V_{k'j'}$ and $L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)I\{c_i=k'\}V_{k'j'}\mu_{k'j}^*$ are sub-exponential random variables with norm $\|\boldsymbol{\mu}_{k'}^*\|_\infty(\|\boldsymbol{\Sigma}_{k'}^*\|_{\max})^{1/2}$. Similar to the step in (S.20),

$$\left|\zeta_{jj'}(\mu_{k'j}^*V_{k'j'})\right| \leq \sqrt{\frac{4}{D_5}}\left(\|\boldsymbol{\mu}_{k'}^*\|_\infty(\|\boldsymbol{\Sigma}_{k'}^*\|_{\max})^{1/2}\right)\sqrt{\frac{\log(2/\delta)}{n}},$$

with at least probability $1-\delta$. Taking the union bound, it is shown that

$$(ii),(iii) \leq \sqrt{\frac{4}{D_5}}\left(\|\boldsymbol{\mu}_{k'}^*\|_\infty(\|\boldsymbol{\Sigma}_{k'}^*\|_{\max})^{1/2}\right)\sqrt{\frac{\log p+\log(2/\delta)}{n}}, \tag{S.30}$$

with probability at least $1-\delta$ for sufficient large n.

Lastly, the fact that both $L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)I\{c_i=k'\}V_{k'j}$ and $V_{k'j'}$ are sub-gaussian random variables implies $L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)I\{c_i=k'\}V_{k'j}V_{k'j'}$ is sub-exponential random variable with parameter $\|\boldsymbol{\Sigma}_{k'}^*\|_{\max}$. Applying concentration result, there exists some constant D_6 such that the following inequality

$$\mathbb{P}\left(|\zeta_{jj'}(V_{k'j}V_{k'j'})| \geq t\right) \leq 2\exp\left(-\frac{D_6nt^2}{4\|\boldsymbol{\Sigma}_{k'}^*\|_{\max}^2}\right),$$

holds for sufficiently small $t > 0$. Therefore,

$$\mathbb{P}\left(\sup_{j,j'\in[p]}|\zeta_{jj'}(V_{k'j}V_{k'j'})| \geq t\right) \leq 2p^2\exp\left(-\frac{D_6nt^2}{4\|\boldsymbol{\Sigma}^*_{k'}\|^2_{\max}}\right).$$

When n is sufficiently large, with probability at least $1-\delta$

$$(iv) \leq \sqrt{\frac{4}{D_6}}\|\boldsymbol{\Sigma}^*_{k'}\|_{\max}\sqrt{\frac{2\log p+\log(2/\delta)}{n}}. \tag{S.31}$$

Putting (S.29), (S.30) and (S.31) together and after some adjustments, II_{21} is upper bounded by

$$II_{21} \leq \sqrt{\frac{1}{D_7}}\sum_{k'=1}^{K}\left(\|\boldsymbol{\mu}^*_{k'}\|_\infty + (\|\boldsymbol{\Sigma}^*_{k'}\|_{\max})^{1/2}\right)^2\sqrt{\frac{2\log p+\log(e/\delta)}{n}},$$

with probability at least $1-4K\delta$. $D_7 = \min(D_4, D_5, D_6)$. For simplicity, we denote

$$\varphi'_K = \sum_{k'=1}^{K}\left(\|\boldsymbol{\mu}^*_{k'}\|_\infty + (\|\boldsymbol{\Sigma}^*_{k'}\|_{\max})^{1/2}\right)^2.$$

Therefore,

$$II_{21} \leq \sqrt{\frac{2}{D_7}}\varphi'_K\sqrt{\frac{\log p+\log(e/\delta)}{n}}, \tag{S.32}$$

with probability at least $1-4K\delta$.

For the last, it remains to bound II_{24}. Recall that

$$\begin{aligned} II_{24} &= \left\|\frac{1}{2}\left(\frac{1}{n}\sum_{i=1}^{n}L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i)\boldsymbol{\mu}^*_k\boldsymbol{\mu}^{*\top}_k - \mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})\boldsymbol{\mu}^*_k\boldsymbol{\mu}^{*\top}_k\right]\right)\right\|_{\max} \\ &\leq \left|\frac{1}{2}\left(\frac{1}{n}\sum_{i=1}^{n}L_{\boldsymbol{\Theta},k}(\boldsymbol{x}_i) - \mathbb{E}\left[L_{\boldsymbol{\Theta},k}(\boldsymbol{X})\right]\right)\right|\|\boldsymbol{\mu}^*_k\boldsymbol{\mu}^{*\top}_k\|_{\max}. \end{aligned}$$

Applying the result in (S.22), we have

$$II_{24} \leq \|\boldsymbol{\mu}^*_k\boldsymbol{\mu}^{*\top}_k\|_{\max}\sqrt{\frac{1}{2}\log\frac{2}{\delta}}\cdot\sqrt{\frac{1}{n}}, \tag{S.33}$$

with probability at least $1-\delta$.

Putting (S.27), (S.32) and (S.33) together, now we can have a upper bound for II_2.

$$II_2 \leq \sqrt{\frac{1}{D_7}}\left(2\|\boldsymbol{\mu}^*_k\|_\infty\varphi_K + \varphi'_K\right)\sqrt{\frac{\log p+\log(e/\delta)}{n}}, \tag{S.34}$$

for $D_7 < D/2$ with at least probability $1-(8K+1)\delta$. The upper bound in (S.26) is of order $O_P(n^{-1/2})$ while the upper bound in (S.34) is of order $O_P((\log p/n)^{1/2})$. Thus there exists

some constant D_8 such that $II_1 \leq D_8II_2$. Let $C_2 = ((1+D_8)^2/D_7)^{1/2}$. Applying union bound,

$$\max_{k\in[K]}II \leq C_2\left(2\|\boldsymbol{\mu}^*\|_\infty\varphi_K + \varphi'_K\right)\sqrt{\frac{\log p+\log(e/\delta)}{n}}, \tag{S.35}$$

with at least probability $1-K(8K+2)\delta$.

Bound the Group Structure Part of Precision Matrix:

Recall that

$$\begin{aligned} III &= \max_{i,j}\Big\|\left[\nabla_{\boldsymbol{\Omega}^*_1}Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) - \nabla_{\boldsymbol{\Omega}^*_1}Q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta})\right]_{ij}, \\ &\qquad \ldots, \left[\nabla_{\boldsymbol{\Omega}^*_K}Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) - \nabla_{\boldsymbol{\Omega}^*_K}Q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta})\right]_{ij}\Big\|_2 \\ &\leq \max_{i,j}\sqrt{K}\Big\|\left[\nabla_{\boldsymbol{\Omega}^*_1}Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) - \nabla_{\boldsymbol{\Omega}^*_1}Q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta})\right]_{ij}, \\ &\qquad \ldots, \left[\nabla_{\boldsymbol{\Omega}^*_K}Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) - \nabla_{\boldsymbol{\Omega}^*_K}Q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta})\right]_{ij}\Big\|_\infty \\ &\leq \sqrt{K}\max_{k\in[K]}\Big\|\Big[\nabla_{\boldsymbol{\Omega}^*_k}Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) - \nabla_{\boldsymbol{\Omega}^*_k}Q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta})\Big]\Big\|_{\max}. \end{aligned}$$

According to the result in (S.35) and applying union bound over $[K]$,

$$\mathbb{P}\left(III \geq C_2\sqrt{K}\left(2\|\boldsymbol{\mu}^*_k\|_\infty\varphi_K + \varphi'_K\right)\sqrt{\frac{\log p+\log(e/\delta)}{n}}\right) \leq K(8K+2)\delta.$$

Thus, III is upper bounded by

$$III \leq C_2\sqrt{K}\left(2\|\boldsymbol{\mu}^*\|_\infty\varphi_K + \varphi'_K\right)\sqrt{\frac{\log p+\log(e/\delta)}{n}}, \tag{S.36}$$

with at least probability $1-K(8K+2)\delta$.

Finally, putting the upper bound (S.25), (S.35) and (S.36) together, we have a upper bound for the following statistical error

$$\begin{aligned} &\left\|\nabla Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) - \nabla Q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta})\right\|_{\mathcal{P}^*} \\ \leq\ & C\left((\|\boldsymbol{\Omega}^*\|_\infty + (\sqrt{K}+1)\|\boldsymbol{\mu}^*\|_\infty)\varphi_K + 2(\sqrt{K}+1)\varphi'_K\right)\sqrt{\frac{\log p+\log(e/\delta)}{n}}, \end{aligned}$$

with probability at least $1-(18K+6)\delta$, where $C = \max(M_1C_1, M_2C_2, M_3C_3)$. Under regularity Condition 16, $\varphi_K \leq (c_1+c_2^{1/2})K$, $\varphi'_K \leq (c_1+c_2^{1/2})^2K$. Let $C = C(c_1+c_2^{1/2})$ and $C' = c_1^2 + c_1c_2^{1/2} + 2(c_1+c_2^{1/2})^2$. Consequently, the upper bound for statistical error can be written as:

$$\|\nabla Q_n(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}) - \nabla Q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta})\|_{\mathcal{P}^*} \leq \left(CK\|\boldsymbol{\Omega}^*\|_\infty + C'K^{1.5}\right)\sqrt{\frac{\log p+\log(e/\delta)}{n}},$$

with probability at least $1-(18K+6)\delta$. ■

For the second part of Lemma S.1, we are aiming to bound the statistical error arising from the estimation for diagonal term. The definition of $\mathcal{G}$ in (14) implies that $[\nabla Q_n(\mathbf{\Theta}^*|\mathbf{\Theta}) - \nabla Q(\mathbf{\Theta}^*|\mathbf{\Theta})]_{\mathcal{G}}$ is a Kp-dimensional vector. Following the same derivation before, it suffices to have:

$$\begin{aligned} & \left\|[\nabla Q_n(\mathbf{\Theta}^*|\mathbf{\Theta}) - \nabla Q(\mathbf{\Theta}^*|\mathbf{\Theta})]_{\mathcal{G}}\right\|_2 \\ \leq & \sqrt{Kp}\left\|[\nabla Q_n(\mathbf{\Theta}^*|\mathbf{\Theta}) - \nabla Q(\mathbf{\Theta}^*|\mathbf{\Theta})]_{\mathcal{G}}\right\|_{\max} \\ \overset{(a)}{\leq} & \sqrt{Kp}\cdot C_2\left(2\|\boldsymbol{\mu}^*\|_\infty\varphi_K + \varphi'_K\right)\sqrt{\frac{\log p + \log(e/\delta)}{n}} \\ = & \sqrt{K}\cdot C_2\left(2\|\boldsymbol{\mu}^*\|_\infty\varphi_K + \varphi'_K\right)\sqrt{\frac{p(\log p + \log(e/\delta))}{n}}, \end{aligned}$$

with probability at least $1-(8K^2+2K)\delta$ where (a) comes from (S.36). Now combining two parts together, we end the proof of Lemma S.1. ■

S.V Proof of Lemma 22

For any $\mathbf{\Theta} \in \mathcal{M}$,

$$\begin{aligned} \frac{\mathcal{P}(\mathbf{\Theta})}{\|\mathbf{\Theta}\|_2} &= \frac{\mathcal{P}_1(\mathbf{\Theta})}{\|\mathbf{\Theta}\|_2} + \frac{\mathcal{P}_2(\mathbf{\Theta})}{\|\mathbf{\Theta}\|_2} + \frac{\mathcal{P}_3(\mathbf{\Theta})}{\|\mathbf{\Theta}\|_2} \\ &\leq \frac{M_1\sum_{k=1}^K\sum_{j=1}^p|\mu_{kj}|}{\sqrt{\sum_{k=1}^K\|\boldsymbol{\mu}_k\|_2^2}} + \frac{M_2\sum_{k=1}^K\sum_{i\neq j}|\omega_{kij}|}{\sqrt{\sum_{k=1}^K\|\mathbf{\Omega}_k\|_F^2}} + \frac{\sum_{i\neq j}M_3(\sum_{k=1}^K\omega_{kij}^2)^{1/2}}{\sqrt{\sum_{k=1}^K\|\mathbf{\Omega}_k\|_F^2}}. \end{aligned}$$

By Cauchy's inequality, we can have

$$\frac{\mathcal{P}(\mathbf{\Theta}_{\mathcal{M}})}{\|\mathbf{\Theta}_{\mathcal{M}}\|_2} \leq M_1\sqrt{Kd} + M_2\sqrt{Ks} + M_3\sqrt{s}.$$

Recall that d and s are the sparse parameter for a single cluster mean and precision matrix, respectively. This ends the proof of Lemma 22. ■

S.VI Proof of Lemma 24

First we consider each $\mathbf{\Theta}_k = \{\mu_k, \mathbf{\Omega}_k\}$ individually. That means we prove the following part first:

$$Q_n(\mathbf{\Theta}_k^{(1)}|\mathbf{\Theta}^{(t-1)}) - Q_n(\mathbf{\Theta}_k^{(2)}|\mathbf{\Theta}^{(t-1)}) - \left\langle\nabla_{\mathbf{\Theta}_k}Q_n(\mathbf{\Theta}_k^{(2)}|\mathbf{\Theta}^{(t-1)}), \mathbf{\Theta}_k^{(1)} - \mathbf{\Theta}_k^{(2)}\right\rangle \leq 0,$$

where $Q_n(\mathbf{\Theta}_k|\mathbf{\Theta})$ means we set $\mathbf{\Theta}_i$ $i \neq k$ to zero.

Following the same technique we use in the proof of Lemma (9), the decomposition can be made as below:

$$Q_n(\mathbf{\Theta}_k^{(1)}|\mathbf{\Theta}^{(t-1)}) - Q_n(\mathbf{\Theta}_k^{(2)}|\mathbf{\Theta}^{(t-1)}) - \left\langle\nabla_{\mathbf{\Theta}_k}Q_n(\mathbf{\Theta}_k^{(2)}|\mathbf{\Theta}^{(t-1)}), \mathbf{\Theta}_k^{(1)} - \mathbf{\Theta}_k^{(2)}\right\rangle = I + II,$$

where

$$\begin{aligned} I &= \frac{1}{n}\sum_{i=1}^n\left[L_{\mathbf{\Theta},k}(\boldsymbol{x}_i)\left\{h(\boldsymbol{\mu}_k^{(2)}, \mathbf{\Omega}_k^{(2)}) - h(\boldsymbol{\mu}_k^{(1)}, \mathbf{\Omega}_k^{(2)})\right\}\right] \\ &-(\boldsymbol{\mu}_k^{(1)} - \boldsymbol{\mu}_k^{(2)})^\top\nabla_{\boldsymbol{\mu}_k}Q_n(\mathbf{\Theta}_k^{(2)}|\mathbf{\Theta}^{(t-1)}), \\ II &= \frac{1}{n}\sum_{i=1}^n\big[L_{\mathbf{\Theta},k}(\boldsymbol{x}_i)\{\frac{1}{2}\log\det(\mathbf{\Omega}_k^{(1)}) - \frac{1}{2}\log\det(\mathbf{\Omega}_k^{(2)}) \\ &+h(\boldsymbol{\mu}_k^{(1)}, \mathbf{\Omega}_k^{(2)}) - h(\boldsymbol{\mu}_k^{(1)}, \mathbf{\Omega}_k^{(1)})\}\big] - [\mathrm{vec}(\mathbf{\Omega}_k^{(1)} - \mathbf{\Omega}_k^{(2)})]^\top\nabla_{\mathbf{\Omega}_k}Q_n(\mathbf{\Theta}_k^{(2)}|\mathbf{\Theta}^{(t-1)}). \end{aligned}$$

Bounding I: By a little algebra, we can show that

$$I = -\frac{1}{2n}\sum_{i=1}^n L_{\mathbf{\Theta},k}(\boldsymbol{x}_i)(\boldsymbol{\mu}_k^{(1)} - \boldsymbol{\mu}_k^{(2)})^\top\mathbf{\Omega}_k^{(2)}(\boldsymbol{\mu}_k^{(1)} - \boldsymbol{\mu}_k^{(2)}).$$

Plugging in $(\mathbf{\Theta}^{(t)}, t^*\mathbf{\Theta}^{(t)} + (1-t^*)\mathbf{\Theta}^*)$, we have

$$I = -\frac{(1-t^*)^2}{2n}\sum_{i=1}^n L_{\mathbf{\Theta},k}(\boldsymbol{x}_i)(\boldsymbol{\mu}_k^{(t)} - \boldsymbol{\mu}_k^*)^\top\left(t^*\mathbf{\Omega}_k^{(t)} + (1-t^*)\mathbf{\Omega}_k^*\right)(\boldsymbol{\mu}_k^{(t)} - \boldsymbol{\mu}_k^*).$$

Recall that $\mathbf{\Theta}^{(t)}$ is the solution of the optimization problem (35). The algorithm guarantees that $\mathbf{\Omega}_k^{(t)}$ is positive definite. Thus, from the positive definiteness of $\mathbf{\Omega}_k^{(t)}$ and $\mathbf{\Omega}_k^*$, it is sufficient to show that

$$I \leq 0 \quad \text{holds a.s..} \tag{S.37}$$

When plugging in $(\mathbf{\Theta}^*, t^*\mathbf{\Theta}^{(t)} + (1-t^*)\mathbf{\Theta}^*)$, we have the same conclusion.

Bounding II: Define

$$g(\mathbf{\Omega}_k^{(2)}) := \frac{1}{n}\sum_{i=1}^n\left[L_{\mathbf{\Theta},k}(\boldsymbol{x}_i)\left\{\frac{1}{2}\log\det(\mathbf{\Omega}_k^{(2)}) - h\left(\boldsymbol{\mu}_k^{(1)}, \mathbf{\Omega}_k^{(2)}\right)\right\}\right].$$

We rewrite II as

$$g(\mathbf{\Omega}_k^{(1)}) - g(\mathbf{\Omega}_k^{(2)}) - \left\langle\mathrm{vec}\left(\nabla g(\mathbf{\Omega}_k^{(2)})\right), \mathrm{vec}\left(\mathbf{\Omega}_k^{(1)} - \mathbf{\Omega}_k^{(2)}\right)\right\rangle.$$

According to Taylor expansion, we can expand $g(\mathbf{\Omega}_k^{(1)})$ around $\mathbf{\Omega}_k^{(2)}$ and obtain

$$\begin{aligned} g(\mathbf{\Omega}_k^{(1)}) &= g(\mathbf{\Omega}_k^{(2)}) + \left\langle\mathrm{vec}(\nabla g(\mathbf{\Omega}_k^{(2)}), \mathrm{vec}(\mathbf{\Omega}_k^{(1)} - \mathbf{\Omega}_k^{(2)})\right\rangle \\ &+\frac{1}{2}\left[\mathrm{vec}(\mathbf{\Omega}_k^{(1)} - \mathbf{\Omega}_k^{(2)})\right]^\top\nabla^2 g(\mathbf{Z})\left[\mathrm{vec}(\mathbf{\Omega}_k^{(1)} - \mathbf{\Omega}_k^{(2)})\right], \end{aligned}$$

where $\mathbf{Z} = t\mathbf{\Omega}_k^{(1)} + (1-t)\mathbf{\Omega}_k^{(2)}$ with $t \in [0,1]$. So an equivalent expression for II is given below:

$$II = \frac{1}{2}\left[\mathrm{vec}(\mathbf{\Omega}_k^{(1)} - \mathbf{\Omega}_k^{(2)})\right]^\top\nabla^2 g(\mathbf{Z})\left[\mathrm{vec}(\mathbf{\Omega}_k^{(1)} - \mathbf{\Omega}_k^{(2)})\right].$$

By the definition of function g we construct, the negative Hessian matrix of function g is

$$-\nabla^2 g(\mathbf{Z}) = \frac{1}{2n}\sum_{i=1}^{n} L_{\mathbf{\Theta},k}(\boldsymbol{x}_i)\mathbf{Z}^{-1}\otimes\mathbf{Z}^{-1}.$$

According to the analysis in the proof of Lemma 9, $\sigma_{\min}\left(\mathbf{Z}^{-1}\otimes\mathbf{Z}^{-1}\right) = \left[\sigma_{\min}(\mathbf{Z}^{-1})\right]^2 \geq 0$. Therefore, $\nabla^2 g(\mathbf{Z})$ is a negative semi-definite matrix, which implies that $II \leq 0$ holds a.s. for any pair of points $(\mathbf{\Theta}^{(1)}, \mathbf{\Theta}^{(2)})$. Incorporating with the fact that $I < 0$, it implies that

$$Q_n(\mathbf{\Theta}_k^{(1)}|\mathbf{\Theta}^{(t-1)}) - Q_n(\mathbf{\Theta}_k^{(2)}|\mathbf{\Theta}^{(t-1)}) - \langle \nabla_{\mathbf{\Theta}_k} Q_n(\mathbf{\Theta}_k^{(2)}|\mathbf{\Theta}^{(t-1)}), \mathbf{\Theta}_k^{(1)} - \mathbf{\Theta}_k^{(2)}\rangle \leq 0,$$

holds a.s. for pair points $(\mathbf{\Theta}^{(t)}, t^*\mathbf{\Theta}^{(t)} + (1-t^*)\mathbf{\Theta}^*)$, $(\mathbf{\Theta}^{(t)}, t^*\mathbf{\Theta}^{(t)} + (1-t^*)\mathbf{\Theta}^*)$. After doing the summation from 1 to K, we finish the proof of Lemma 24. ■

S.VII Variable Selection Consistency

Theorem S.2 *Denote the final precision matrix estimator as $\widetilde{\mathbf{\Omega}}_k$ and the set of its nonzero off-diagonal elements as $\widetilde{\mathcal{V}}_k$. Under minimal signal condition, we have, with probability tending to 1, $\widetilde{\mathcal{V}}_k = \mathcal{V}_k$ for any $k = 1, \ldots, K$.*

Proof: We prove it in two steps. In Step 1, we show that $\widetilde{\mathcal{V}_k} \supset \mathcal{V}_k$, and in Step 2, we show that $\widetilde{\mathcal{V}_k} \subset \mathcal{V}_k$, both with high probability.

Step 1: In order to prove $\widetilde{\mathcal{V}_k} \supset \mathcal{V}_k$, it is sufficient to show that for any $(i,j) \in \mathcal{V}_k$ with any $k = 1, \ldots, K$, $\widetilde{\omega}_{kij} \neq 0$. Note that

$$|\omega_{kij}^{(T)}| \geq |\omega_{kij}^*| - |\omega_{kij}^{(T)} - \omega_{kij}^*| \geq |\omega_{kij}^*| - \sqrt{\sum_{i,j}(\omega_{kij}^{(T)} - \omega_{kij}^*)^2},$$

Moreover,

$$\sqrt{\sum_{i,j}(\omega_{kij}^{(T)} - \omega_{kij}^*)^2} \leq \|\mathbf{\Theta}^{(T)} - \mathbf{\Theta}^*\|_2. \tag{S.38}$$

According to Corollary 18 and minimal signal condition we have

$$|\omega_{kij}^{(T)}| > r_n.$$

Therefore, we see that $\widetilde{\omega}_{kij} \neq 0$, which implies $\widetilde{\mathcal{V}_k} \supset \mathcal{V}_k$.

Step 2: In order to show $\widetilde{\mathcal{V}_k} \subset \mathcal{V}_k$, we need to check that, for any $(i,j) \in \mathcal{V}_k^c$, the estimator $\widetilde{\omega}_{kij} = 0$. Note that, the estimator before the thresholding step satisfies,

$$|\omega_{kij}^{(T)}| = |\omega_{kij}^{(T)} - \omega_{kij}^*| \leq \sqrt{\sum_{i,j}(\omega_{kij}^{(T)} - \omega_{kij}^*)^2}.$$

From (S.38), it is known that $|\omega_{kij}^{(T)}| \leq r_n$. Therefore, the thresholding step will set $\widetilde{\omega}_{kij} = \omega_{kij}^{(T)} 1\{|\widehat{\omega}_{kij}| > r_n\} = 0$ with high probability. This ends the proof of Theorem S.2. ■

Appendix B. Updates steps of our SCAN algorithm

S.I Proof of Lemma 2:

The KKT conditions for μ_{kj} to be a maximizer of $Q(\mathbf{\Theta}|\mathbf{\Theta}^{(t-1)}) - \mathcal{R}(\mathbf{\Theta})$ are

$$\frac{1}{n}\sum_{i=1}^{n} L_{\mathbf{\Theta}^{(t-1)},k}\Big(\sum_{l=1}^{p}(x_{il}-\mu_{kl})\omega_{klj}\Big) = \lambda_1 \text{sign}(\mu_{kj}), \text{ when } \mu_{kj} \neq 0,$$
$$\left|\frac{1}{n}\sum_{i=1}^{n} L_{\mathbf{\Theta}^{(t-1)},k}\Big(\sum_{l=1,l\neq j}^{p}(x_{il}-\mu_{kl})\omega_{klj} + x_{ij}\omega_{kjj}\Big)\right| \leq \lambda_1, \text{ when } \mu_{kj} = 0.$$

Therefore, the update of $\mu_{kj}^{(t)}$ is given as:

$$\text{If } \left|\frac{1}{n}\sum_{i=1}^{n} L_{\mathbf{\Theta}^{(t-1)},k}(\boldsymbol{x}_i)\Big(\sum_{l=1,l\neq j}^{p}(x_{il}-\mu_{kl}^{(t-1)})\omega_{klj}^{(t-1)} + x_{ij}\omega_{kjj}^{(t-1)}\Big)\right| \leq \lambda_1,$$

then $\mu_{kj}^{(t)} = 0$; Else

$$\mu_{kj}^{(t)} = \Big(\omega_{kjj}^{(t-1)}\frac{1}{n}\sum_{i=1}^{n} L_{\mathbf{\Theta}^{(t-1)},k}(\boldsymbol{x}_i)\Big)^{-1}\Big\{\frac{1}{n}\sum_{i=1}^{n} L_{\mathbf{\Theta}^{(t-1)},k}(\boldsymbol{x}_i)\big(\sum_{l=1}^{p} x_{il}\omega_{klj}^{(t-1)}\big) -$$
$$\big(\frac{1}{n}\sum_{i=1}^{n} L_{\mathbf{\Theta}^{(t-1)},k}(\boldsymbol{x}_i)\big)\Big(\sum_{l=1}^{p}\mu_{kl}^{(t-1)}\omega_{klj}^{(t-1)} - \mu_{kj}^{(t-1)}\omega_{kjj}^{(t-1)}\Big) - \lambda_1 \text{sign}(\mu_{kj}^{(t-1)})\Big\}$$

Using the definitions of $g_{1,j}(\boldsymbol{x}; \mathbf{\Theta}_k^{(t-1)})$ and $g_{2,j}(\boldsymbol{x}_i; \mathbf{\Theta}_k^{(t-1)})$, we finish the proof of Lemma 2. ■

S.II Proof of Lemma 3:

Recall that in (8)

$$Q_n(\mathbf{\Theta}|\mathbf{\Theta}^{(t-1)}) := \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} L_{\mathbf{\Theta}^{(t-1)},k}(\boldsymbol{x}_i)[\log\pi_k + \log f_k(\boldsymbol{x}_i;\mathbf{\Theta}_k)] - \mathcal{R}(\mathbf{\Theta}),$$

Then,

$$\begin{aligned}
&\max_{\boldsymbol{\Omega}_1,\ldots,\boldsymbol{\Omega}_K} \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} L_{\boldsymbol{\Theta}^{(t-1)},k}(\boldsymbol{x}_i)[\log \pi_k + \log f_k(\boldsymbol{x}_i;\boldsymbol{\Theta}_k)] - \mathcal{R}(\boldsymbol{\Theta}) \\
= &\max_{\boldsymbol{\Omega}_1,\ldots,\boldsymbol{\Omega}_K} \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} L_{\boldsymbol{\Theta}^{(t-1)},k}(\boldsymbol{x}_i)[\log \pi_k - \frac{p}{2}\log(2\pi) + \frac{1}{2}\log\det(\boldsymbol{\Omega}_k) \\
&-\frac{1}{2}(\boldsymbol{x}_i-\boldsymbol{\mu}_k)^\top\boldsymbol{\Omega}_k(\boldsymbol{x}_i-\boldsymbol{\mu}_k)] - \frac{1}{2}\mathcal{R}(\boldsymbol{\Theta}) \\
= &\max_{\boldsymbol{\Omega}_1,\ldots,\boldsymbol{\Omega}_K} \frac{1}{n}\sum_{k=1}^{K}\left\{\frac{1}{n}\sum_{i=1}^{n} L_{\boldsymbol{\Theta}^{(t-1)},k}(\boldsymbol{x}_i)[\log\det(\boldsymbol{\Omega}_k) - (\boldsymbol{x}_i-\boldsymbol{\mu}_k)^\top\boldsymbol{\Omega}_k(\boldsymbol{x}_i-\boldsymbol{\mu}_k)]\right\} - \mathcal{R}(\boldsymbol{\Theta}) \\
= &\max_{\boldsymbol{\Omega}_1,\ldots,\boldsymbol{\Omega}_K} \frac{1}{n}\sum_{k=1}^{K} n_k[\log\det(\boldsymbol{\Omega}_k) - \mathrm{trace}(\widetilde{S}_k\boldsymbol{\Omega}_k)] - \mathcal{R}(\boldsymbol{\Theta}),
\end{aligned}$$

where the last equality is because

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n} L_{\boldsymbol{\Theta}^{(t-1)},k}(\boldsymbol{x}_i)(\boldsymbol{x}_i-\boldsymbol{\mu}_k)^\top\boldsymbol{\Omega}_k(\boldsymbol{x}_i-\boldsymbol{\mu}_k) \\
= &\frac{1}{n}\sum_{\boldsymbol{x}_i\in\mathcal{A}_k}\mathrm{trace}((\boldsymbol{x}_i-\boldsymbol{\mu}_k)(\boldsymbol{x}_i-\boldsymbol{\mu}_k)^\top\boldsymbol{\Omega}_k) \\
= &\frac{1}{n}\mathrm{trace}\Big(\sum_{\boldsymbol{x}_i\in\mathcal{A}_k}(\boldsymbol{x}_i-\boldsymbol{\mu}_k)(\boldsymbol{x}_i-\boldsymbol{\mu}_k)^\top\boldsymbol{\Omega}_k\Big).
\end{aligned}$$

Then plugging in the last update of $\boldsymbol{\mu}_k$ leads to the desirable result. ■

Appendix C. Supporting Lemma

Lemma S.3 *Consider a finite number of independent centered sub-gaussian random variables X_i. Then $\sum_i X_i$ is also a centered sub-gaussian random variable. Moreover,*

$$\left\|\sum_i X_i\right\|_{\psi_2}^2 \le C\sum_i \|X_i\|_{\psi_2}^2,$$

where C is an absolute constant.

Lemma S.4 *Let X, Y be two sub-Gaussian random variables. Then $Z = X\cdot Y$ is sub-exponential random variable. Moreover, there exits constant C such that*

$$\|Z\|_{\psi_1} \le C\|X\|_{\psi_2}\cdot\|Y\|_{\psi_2}. \tag{S.39}$$

Lemma S.5 *Let X be sub-Gaussian random variable and Y be sub-exponential random variables. Then $X-\mathbb{E}[X]$ is also sub-Gaussian; $Y-\mathbb{E}[Y]$ is also sub-exponential. Moreover, we have*

$$\|X-\mathbb{E}[X]\|_{\psi_2} \le 2\|X\|_{\psi_2},\ \ \|Y-\mathbb{E}[Y]\|_{\psi_1} \le 2\|Y\|_{\psi_1}.$$

Lemma S.6 *Suppose $X_1, X_2, \ldots, X_n$ are n iid centered sub-Gaussian random variables with $\|X_1\|_{\psi_2} \le K$. Then for every $t \ge 0$, we have*

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n} X_i\right| \ge t\right) \ge e\cdot\exp\left(-\frac{Cnt^2}{K^2}\right),$$

where C is an absolute constant.

Lemma S.7 *Suppose $X_1, X_2, \ldots, X_n$ are n iid centered sub-expoential random variables with $\|X_1\|_{\psi_1} \le K$. Then for every $t \ge 0$, we have*

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n} X_i\right| \ge t\right) \ge 2\cdot\exp\left(-C\min\left\{\frac{t^2}{K^2}, \frac{t}{K}\right\}n\right),$$

where C is an absolute constant.

Lemma S.8 *Hoeffding's inequality Suppose $X_1, X_2 \ldots X_n$ are independent random variable, $a_1 \le X_i \le b_i$, then we can have*

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n}(X_i-\mathbb{E}X_i)\right| > \varepsilon\right) \le 2\exp\left\{\frac{-2n\varepsilon^2}{\frac{1}{n}\sum_{i=1}^{n}(b_i-a_i)^2}\right\}.$$

Moreover, if $a_i = 0$ and $b_i = 1$, then we have

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n}(X_i-\mathbb{E}X_i)\right| > \varepsilon\right) > 1-2e^{-2n\varepsilon^2}.$$

Journal of Machine Learning Research 18 (2018) 1-50 Submitted 6/17; Published 5/18

Sketched Ridge Regression: Optimization Perspective, Statistical Perspective, and Model Averaging

Shusen Wang WSSATZJU@GMAIL.COM
International Computer Science Institute and Department of Statistics
University of California at Berkeley
Berkeley, CA 94720, USA

Alex Gittens GITTEA@RPI.EDU
Computer Science Department
Rensselaer Polytechnic Institute
Troy, NY 12180, USA

Michael W. Mahoney MMAHONEY@STAT.BERKELEY.EDU
International Computer Science Institute and Department of Statistics
University of California at Berkeley
Berkeley, CA 94720, USA

Editor: Mehryar Mohri

Abstract

We address the statistical and optimization impacts of the classical sketch and Hessian sketch used to approximately solve the Matrix Ridge Regression (MRR) problem. Prior research has quantified the effects of classical sketch on the strictly simpler least squares regression (LSR) problem. We establish that classical sketch has a similar effect upon the optimization properties of MRR as it does on those of LSR: namely, it recovers nearly optimal solutions. By contrast, Hessian sketch does not have this guarantee; instead, the approximation error is governed by a subtle interplay between the "mass" in the responses and the optimal objective value.

For both types of approximation, the regularization in the sketched MRR problem results in significantly different statistical properties from those of the sketched LSR problem. In particular, there is a bias-variance trade-off in sketched MRR that is not present in sketched LSR. We provide upper and lower bounds on the bias and variance of sketched MRR; these bounds show that classical sketch significantly increases the variance, while Hessian sketch significantly increases the bias. Empirically, sketched MRR solutions can have risks that are higher by an order-of-magnitude than those of the optimal MRR solutions.

We establish theoretically and empirically that model averaging greatly decreases the gap between the risks of the true and sketched solutions to the MRR problem. Thus, in parallel or distributed settings, sketching combined with model averaging is a powerful technique that quickly obtains near-optimal solutions to the MRR problem while greatly mitigating the increased statistical risk incurred by sketching.

Keywords: Randomized Linear Algebra, Matrix Sketching, Ridge Regression

1. Introduction

Regression is one of the most fundamental problems in machine learning. The simplest and most thoroughly studied regression model is least squares regression (LSR). Given features $\mathbf{X} = [\mathbf{x}_1^T; \ldots, \mathbf{x}_n^T] \in \mathbb{R}^{n \times d}$ and responses $\mathbf{y} = [y_1, \ldots, y_n]^T \in \mathbb{R}^n$, the LSR problem $\min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2$ can be solved in $\mathcal{O}(nd^2)$ time using the QR decomposition or in $\mathcal{O}(ndt)$ time using accelerated gradient descent algorithms. Here, t is the number of iterations, which depends on the initialization, the condition number of $\mathbf{X}^T\mathbf{X}$, and the stopping criterion.

This paper considers the $n \gg d$ problem, where there is much redundancy in $\mathbf{X}$. Matrix sketching, as used in the paradigm of Randomized Linear Algebra (RLA) (Mahoney, 2011; Woodruff, 2014; Drineas and Mahoney, 2016), aims to reduce the size of $\mathbf{X}$ while limiting information loss; the sketching operation can consist of sampling a subset of the rows of $\mathbf{X}$, or forming linear combinations of the rows of $\mathbf{X}$. Either operation is modeled mathematically by multiplication with a sketching matrix $\mathbf{S}$ to form the sketch $\mathbf{S}^T\mathbf{X}$. The sketching matrix $\mathbf{S} \in \mathbb{R}^{n \times s}$ satisfies $d < s \ll n$ so that $\mathbf{S}^T\mathbf{X}$ generically has the same rank but much fewer rows as $\mathbf{X}$. Sketching has been used to speed up LSR (Drineas et al., 2006b, 2011; Clarkson and Woodruff, 2013; Meng and Mahoney, 2013; Nelson and Nguyên, 2013) by solving the sketched LSR problem $\min_{\mathbf{w}} \|\mathbf{S}^T\mathbf{X}\mathbf{w} - \mathbf{S}^T\mathbf{y}\|_2^2$ instead of the original LSR problem. Solving sketched LSR costs either $\mathcal{O}(sd^2 + T_s)$ time using the QR decomposition or $\mathcal{O}(sdt + T_s)$ time using accelerated gradient descent algorithms, where t is as defined previously[1] and T_s is the time cost of sketching. For example, $T_s = \mathcal{O}(nd \log s)$ when $\mathbf{S}$ is the subsampled randomized Hadamard transform (Drineas et al., 2011), and $T_s = \mathcal{O}(nd)$ when $\mathbf{S}$ is a CountSketch matrix (Clarkson and Woodruff, 2013).

There has been much work in RLA on analyzing the quality of sketched LSR with different sketching methods and different objectives; see the reviews (Mahoney, 2011; Woodruff, 2014; Drineas and Mahoney, 2016) and the references therein. The concept of sketched LSR originated in the theoretical computer science literature, e.g., Drineas et al. (2006b, 2011), where the behavior of sketched LSR was first studied from an optimization perspective. Let $\mathbf{w}^\star$ be the optimal LSR solution and $\tilde{\mathbf{w}}$ be the solution to sketched LSR. This line of work established that if $s = \mathcal{O}(d/\epsilon + \text{poly}(d))$, then the objective value $\|\mathbf{X}\tilde{\mathbf{w}} - \mathbf{y}\|_2^2$ is at most $(1+\epsilon)$ times greater than $\|\mathbf{X}\mathbf{w}^\star - \mathbf{y}\|_2^2$. These works also bounded $\|\tilde{\mathbf{w}} - \mathbf{w}^\star\|_2^2$ in terms of the difference in the objective function values at $\tilde{\mathbf{w}}$ and $\mathbf{w}^\star$ and the condition number of $\mathbf{X}^T\mathbf{X}$.

A more recent line of work has studied sketched LSR from a statistical perspective: Ma et al. (2015); Raskutti and Mahoney (2016); Pilanci and Wainwright (2015); Wang et al. (2017c) considered statistical properties of sketched LSR such as the bias and variance. In particular, Pilanci and Wainwright (2015) showed that the solutions to sketched LSR have much higher variance than the optimal solutions.

Both of these perspectives are important and of practical interest. The optimization perspective is relevant when the approximate solution is used to initialize an (expensive) iterative optimization algorithm; the statistical perspective is relevant in machine learning and statistics applications where the approximate solution is directly used in lieu of the optimal solution.

1. The condition number of $\mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{X}$ is very close to that of $\mathbf{X}^T\mathbf{X}$, and thus the number of iterations t is almost unchanged.

In practice, regularized regression, e.g., ridge regression and LASSO, exhibit more attractive bias-variance trade-offs and generalization errors than vanilla LSR. Furthermore, the matrix generalization of LSR, where multiple responses are to be predicted, is often more useful than LSR. However, the properties of sketched regularized matrix regression are largely unknown. Hence, we consider the question: *how does our understanding of the optimization and statistical properties of sketched LSR generalize to sketched regularized regression problems?* We answer this question for the sketched matrix ridge regression (MRR) problem.

Recall that $\mathbf{X}$ is $n \times d$. Let $\mathbf{Y} \in \mathbb{R}^{n \times m}$ denote a matrix of corresponding responses. We study the MRR problem

$$\min_{\mathbf{W}} \Big\{ f(\mathbf{W}) \triangleq \frac{1}{n} \big\| \mathbf{X}\mathbf{W} - \mathbf{Y} \big\|_F^2 + \gamma \|\mathbf{W}\|_F^2 \Big\}, \tag{1}$$

which has optimal solution

$$\mathbf{W}^\star = (\mathbf{X}^T\mathbf{X} + n\gamma \mathbf{I}_d)^\dagger \mathbf{X}^T\mathbf{Y}. \tag{2}$$

Here, $(\cdot)^\dagger$ denotes the Moore-Penrose inversion operation. LSR is a special case of MRR, with $m = 1$ and $\gamma = 0$. The optimal solution $\mathbf{W}^\star$ can be obtained in $\mathcal{O}(nd^2 + nmd)$ time using a QR decomposition of $\mathbf{X}$. Sketching can be applied to MRR in two ways:

$$\mathbf{W}^{\mathrm{c}} = (\mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{X} + n\gamma \mathbf{I}_d)^\dagger (\mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{Y}), \tag{3}$$

$$\mathbf{W}^{\mathrm{h}} = (\mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{X} + n\gamma \mathbf{I}_d)^\dagger \mathbf{X}^T\mathbf{Y}. \tag{4}$$

Following the convention of Pilanci and Wainwright (2015); Wang et al. (2017a), we call $\mathbf{W}^{\mathrm{c}}$ the **classical sketch** and $\mathbf{W}^{\mathrm{h}}$ the **Hessian sketch**. Table 1 lists the time costs of the three solutions to MRR.

Table 1: The time cost of the solutions to MRR. Here $T_s(\mathbf{X})$ and $T_s(\mathbf{Y})$ denote the time cost of forming the sketches $\mathbf{S}^T\mathbf{X} \in \mathbb{R}^{s \times d}$ and $\mathbf{S}^T\mathbf{Y} \in \mathbb{R}^{s \times m}$.

Solution	Definition	Time Complexity
Optimal Solution	(2)	$\mathcal{O}(nd^2 + nmd)$
Classical Sketch	(3)	$\mathcal{O}(sd^2 + smd) + T_s(\mathbf{X}) + T_s(\mathbf{Y})$
Hessian Sketch	(4)	$\mathcal{O}(sd^2 + nmd) + T_s(\mathbf{X})$

1.1 Main Results and Contributions

We summarize all of our upper bounds in Table 2. Our optimization analysis bounds the gap between the objective function values at the sketched and optimal solutions, while our statistical analysis quantifies the behavior of the bias and variance of the sketched solutions relative to those of the true solutions.

We first study classical and Hessian sketches from the **optimization perspective**. Theorems 1 and 2 show:

Table 2: A summary of our main results. In the table, $\mathbf{W}$ is the solution of classical/Hessian sketch with or without model averaging (mod. avg.); $\mathbf{W}^\star$ is the optimal solution; g is the number of models used in model averaging; and $\beta = \frac{\|\mathbf{X}\|_2^2}{\|\mathbf{X}\|_2^2 + n\gamma} \leq 1$, where γ is the regularization parameter. For conciseness, we take the sketching matrix $\mathbf{S} \in \mathbb{R}^{n \times s}$ to correspond to Gaussian projection, SRHT, or shrinkage leverage score sampling. Similar but more complex expressions hold for uniform sampling (with or without model averaging) and CountSketch (only without model averaging.) All the bounds hold with constant probability. The notation $\tilde{\mathcal{O}}$ conceals logarithmic factors.

	Classical Sketch		**Hessian Sketch**	
	w/o mod. avg.	**w/ mod. avg.**	**w/o mod. avg.**	**w/ mod. avg.**
$s =$	$\tilde{\mathcal{O}}(d/\epsilon)$		$\tilde{\mathcal{O}}(d/\epsilon)$	
$f(\mathbf{W}) - f(\mathbf{W}^\star) \leq$	$\beta\epsilon f(\mathbf{W}^\star)$	$\beta(\frac{\epsilon}{g} + \beta^2\epsilon^2) f(\mathbf{W}^\star)$	$\beta^2\epsilon\big[\frac{\|\mathbf{Y}\|_F^2}{n} - f(\mathbf{W}^\star)\big]$	$\beta^2(\frac{\epsilon}{g} + \epsilon^2)\big[\frac{\|\mathbf{Y}\|_F^2}{n} - f(\mathbf{W}^\star)\big]$
Theorems	Theorem 1	Theorem 7	Theorem 2	Theorem 8
$s =$	$\tilde{\mathcal{O}}(d/\epsilon^2)$		$\tilde{\mathcal{O}}(d/\epsilon^2)$	
$\frac{\mathsf{bias}(\mathbf{W})}{\mathsf{bias}(\mathbf{W}^\star)} \leq$	$1 + \epsilon$	$1 + \epsilon$	$(1+\epsilon)(1 + \frac{\epsilon\|\mathbf{X}\|_2^2}{n\gamma})$	$1 + \epsilon + \big(\frac{\epsilon}{\sqrt{g}} + \epsilon^2\big)\frac{\|\mathbf{X}\|_2^2}{n\gamma}$
$\frac{\mathsf{var}(\mathbf{W})}{\mathsf{var}(\mathbf{W}^\star)} \leq$	$(1+\epsilon)\frac{n}{s}$	$\frac{n}{s}\Big(\sqrt{\frac{1+\epsilon/g}{g}} + \epsilon\Big)^2$	$1 + \epsilon$	$1 + \epsilon$
Theorems	Theorem 5	Theorem 10	Theorem 6	Theorem 11

- Classical sketch achieves relative error in the objective value. With sketch size $s = \tilde{\mathcal{O}}(d/\epsilon)$, the sketched solution satisfies $f(\mathbf{W}^{\mathrm{c}}) \leq (1+\epsilon) f(\mathbf{W}^\star)$.
- Hessian sketch does not achieve relative error in the objective value. In particular, if $\frac{1}{n}\|\mathbf{Y}\|_F^2$ is much larger than $f(\mathbf{W}^\star)$, then $f(\mathbf{W}^{\mathrm{h}})$ can be far larger than $f(\mathbf{W}^\star)$.
- For both classical and Hessian sketch, the relative quality of approximation often improves as the regularization parameter γ increases (because β decreases).

We then study classical and Hessian sketch from the **statistical perspective**, by modeling $\mathbf{Y} = \mathbf{X}\mathbf{W}_0 + \mathbf{\Xi}$ as the sum of a true linear model and random noise, decomposing the risk $R(\mathbf{W}) = \mathbb{E}\|\mathbf{X}\mathbf{W} - \mathbf{X}\mathbf{W}_0\|_F^2$ into bias and variance terms, and bounding these terms. We draw the following conclusions (see Theorems 4, 5, 6 for the details):

- The bias of classical sketch can be nearly as small as that of the optimal solution. The variance is $\Theta\big(\frac{n}{s}\big)$ times that of the optimal solution; this bound is optimal. Therefore over-regularization[2] should be used to supress the variance. (As γ increases, the bias increases, and the variance decreases.)

2. For example, using a larger value of the regularization parameter γ than one would optimally choose for the unsketched problem.

- Since Hessian sketch uses the whole of $\mathbf{Y}$, the variance of Hessian sketch can be close to that of the optimal solution. However, Hessian sketch incurs a high bias, especially when $n\gamma$ is small compared to $\|\mathbf{X}\|_2^2$. This indicates that over-regularization is necessary for Hessian sketch to deliver solutions with low bias.

Our empirical evaluations bear out these theoretical results. In particular, in Section 4, we show in Figure 3 that even when the regularization parameter γ is fine-tuned, the risks of classical and Hessian sketch are worse than that of the optimal solution by an order of magnitude. This is an empirical demonstration of the fact that the near-optimal properties of sketch from the optimization perspective are much less relevant in a statistical setting than its sub-optimal statistical properties.

We propose to use **model averaging**, which averages the solutions of g sketched MRR problems, to attain lower optimization and statistical errors. Without ambiguity, we denote model-averaged classical and Hessian sketches by $\mathbf{W}^{\mathrm{c}}$ and $\mathbf{W}^{\mathrm{h}}$, respectively. Theorems 7, 8, 10, 11 establish the following results:

- Classical Sketch. Model averaging decreases the objective function value and the variance and does not increase the bias. Specifically, with the same sketch size s, model averaging ensures $\frac{f(\mathbf{W}^{\mathrm{c}})-f(\mathbf{W}^\star)}{f(\mathbf{W}^\star)}$ and $\frac{\mathsf{var}(\mathbf{W}^{\mathrm{c}})}{\mathsf{var}(\mathbf{W}^\star)}$ respectively decrease to almost $\frac{1}{g}$ of those of classical sketch without model averaging, provided that $s \gg d$. See Table 2 for the details.

- Hessian Sketch. Model averaging decreases the objective function value and the bias and does not increase the variance.

In the distributed setting, the feature-response pairs $(\mathbf{x}_1, \mathbf{y}_1), \cdots, (\mathbf{x}_n, \mathbf{y}_n) \in \mathbb{R}^d \times \mathbb{R}^m$ are divided among g machines. Assuming that the data have been shuffled randomly, each machine contains a sketch of the MRR constructed by uniformly sampling rows from the data set without replacement. We illustrate this procedure in Figure 1. In this setting, the model averaging procedure communicates the g local models only once to return the final estimate; this process has very low communication and latency costs, and suggests two further applications of classical sketch with model averaging:

- Model Averaging for Machine Learning. When a low-precision solution is acceptable, model averaging can be used in lieu of distributed numerical optimization algorithms requiring multiple rounds of communication. If $\frac{n}{g}$ is large enough compared to d and the row coherence of $\mathbf{X}$ is small, then "one-shot" model averaging has bias and variance comparable to the optimal solution.

- Model Averaging for Optimization. If a high-precision solution to MRR is required, then an iterative numerical optimization algorithm must be used. The cost of such algorithms heavily depends on the quality of the initialization.[3] A good initialization reduces the number of iterations needed to reach convergence. The averaged model is provably close to the optimal solution, so model averaging provides a high-quality initialization for more expensive algorithms.

3. For example, the conjugate gradient method satisfies $\frac{\|\mathbf{W}^{(t)}-\mathbf{W}^\star\|_F^2}{\|\mathbf{W}^{(0)}-\mathbf{W}^\star\|_F^2} \le \theta_1^t$ and stochastic block coordinate descent (Tu et al., 2016) satisfies $\frac{\mathbb{E}f(\mathbf{W}^{(t)})-f(\mathbf{W}^\star)}{f(\mathbf{W}^{(0)})-f(\mathbf{W}^\star)} \le \theta_2^t$. Here $\mathbf{W}^{(t)}$ is the output of the t-th iteration; $\theta_1, \theta_2 \in (0,1)$ depend on the condition number of $\mathbf{X}^T\mathbf{X} + n\gamma\mathbf{I}_d$ and some other factors.

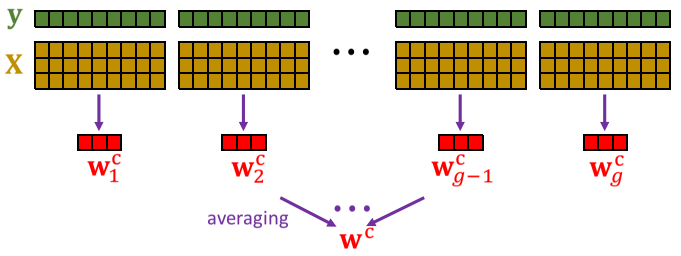


Figure 1: Using model averaging with the classical sketch in the distributed setting to approximately solve LSR.

1.2 Prior Work

The body of work on sketched LSR mentioned earlier (Drineas et al., 2006b, 2011; Clarkson and Woodruff, 2013; Meng and Mahoney, 2013; Nelson and Nguyên, 2013) shares many similarities with our results. However, the theories of sketched LSR developed from the optimization perspective do not obviously extend to MRR, and the statistical analysis of LSR and MRR differ: among other differences, LSR is unbiased while MRR is biased and therefore has a bias-variance tradeoff that must be considered.

Lu et al. (2013) has considered a different application of sketching to ridge regression: they assume $d \gg n$, reduce the number of features in $\mathbf{X}$ using sketching, and conduct statistical analysis. Our setting differs in that we consider $n \gg d$, reduce the number of samples by sketching, and allow for multiple responses.

The model averaging analyzed in this paper is similar in spirit to the AvgM algorithm of (Zhang et al., 2013). When classical sketch is used with uniform row sampling without replacement, our model averaging procedure is a special case of AvgM. However, our results do not follow from those of (Zhang et al., 2013). First, we make no assumption on the data, $\mathbf{X}$ and $\mathbf{Y}$, and the model (parameters), $\mathbf{W}$. Second, we study both the optimization objective, $\|\mathbf{X}\mathbf{W}^{\mathrm{c}} - \mathbf{X}\mathbf{W}^\star\|_F^2$, and the statistical objective, $\mathbb{E}\|\mathbf{X}\mathbf{W}^{\mathrm{c}} - \mathbf{X}\mathbf{W}_0\|_F^2$, where $\mathbf{W}^{\mathrm{c}}$ is the average of the approximate solutions obtained used classical sketch, $\mathbf{W}_0$ is the unknown ground truth, and $\mathbf{W}^\star$ is the optimal solution based on the observed data; they studied solely the optimization objective. Third, our results apply to many other sketching ensembles than uniform sampling without replacement. Our results clearly indicate that the performance critically depends on the row coherence of $\mathbf{X}$; this dependence has not been explicitly captured in (Zhang et al., 2013). Zhang et al. (2015) studied a different statistical objective and their resulting bound has a higher-order of dependence on d and other parameters.

Iterative Hessian sketch has been studied in Pilanci and Wainwright (2015); Wang et al. (2017a,b). By way of comparison, all the algorithms in this paper are "one-shot" rather than iterative. This work has connections to the contemporary works (Avron et al., 2017; Thanei et al., 2017; Derezinski and Warmuth, 2017, 2018). Avron et al. (2017) studied classical sketch from the optimization perspective; Thanei et al. (2017) studied LSR with model averaging; Derezinski and Warmuth (2017, 2018) studied linear regression with volume sampling for experimental design.

1.3 Paper Organization

Section 2 defines our notation and introduces the sketching schemes we consider. Section 3 presents our theoretical results. Sections 4 and 5 conduct experiments to verify our theories and demonstrates the efficacy of model averaging. Section 6 sketches the proofs of our main results. Complete proofs are provided in the appendix.

2. Preliminaries

Throughout, we take $\mathbf{I}_n$ to be the $n \times n$ identity matrix and $\mathbf{0}$ to be a vector or matrix of all zeroes of the appropriate size. Given a matrix $\mathbf{A} = [a_{ij}]$, the i-th row is denoted by $\mathbf{a}_{i:}$, and the j-th column is denoted by $\mathbf{a}_{:j}$. The Frobenius and spectral norms of $\mathbf{A}$ are written as, respectively, $\|\mathbf{A}\|_F$ and $\|\mathbf{A}\|_2$. The set $\{1, 2, \cdots, n\}$ is written $[n]$. Let $\mathcal{O}$, Ω, and Θ be the standard asymptotic notation, and let $\widetilde{\mathcal{O}}$ conceal logarithmic factors.

Throughout, we fix $\mathbf{X} \in \mathbb{R}^{n\times d}$ as our matrix of features. We set $\rho = \text{rank}(\mathbf{X})$ and write the SVD of $\mathbf{X}$ as $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U}$, $\mathbf{\Sigma}$, $\mathbf{V}$ are respectively $n \times \rho$, $\rho \times \rho$, and $d \times \rho$ matrices. We let $\sigma_1 \geq \cdots \geq \sigma_\rho > 0$ be the singular values of $\mathbf{X}$. The Moore-Penrose inverse of $\mathbf{X}$ is defined by $\mathbf{X}^\dagger = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^T$. The row leverage scores of $\mathbf{X}$ are $l_i = \|\mathbf{u}_{i:}\|_2^2$ for $i \in [n]$. The row coherence of $\mathbf{X}$ is $\mu(\mathbf{X}) = \frac{n}{\rho}\max_i \|\mathbf{u}_{i:}\|_2^2$. Throughout, we let μ be shorthand for $\mu(\mathbf{X})$. The notation defined in Table 3 is used throughout this paper.

Matrix sketching attempts to reduce the size of large matrices while minimizing the loss of spectral information that is useful in tasks like linear regression. We denote the process of sketching a matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ by $\mathbf{X}' = \mathbf{S}^T\mathbf{X}$. Here, $\mathbf{S} \in \mathbb{R}^{n\times s}$ is called a sketching matrix and $\mathbf{X}' \in \mathbb{R}^{s\times d}$ is called a sketch of $\mathbf{X}$. In practice, except for Gaussian projection (where the entries of $\mathbf{S}$ are i.i.d. sampled from $\mathcal{N}(0, 1/s)$), the sketching matrix $\mathbf{S}$ is not formed explicitly.

Matrix sketching can be accomplished by random sampling or random projection. **Random sampling** corresponds to sampling rows of $\mathbf{X}$ i.i.d. with replacement according to given row sampling probabilities $p_1, \cdots, p_m \in (0, 1)$. The corresponding (random) sketching matrix $\mathbf{S} \in \mathbb{R}^{n\times s}$ has exactly one non-zero entry, whose position indicates the index of the selected row in each column; in practice, this $\mathbf{S}$ is not explicitly formed. **Uniform sampling** fixes $p_1 = \cdots = p_n = \frac{1}{n}$. **Leverage score sampling** sets p_i proportional to the (exact or approximate (Drineas et al., 2012)) row leverage scores l_i of $\mathbf{X}$. In practice **shrinked leverage score sampling** can be a better choice than leverage score sampling (Ma et al., 2015). The sampling probabilities of shrinked leverage score sampling are defined by $p_i = \frac{1}{2}\left(\frac{l_i}{\sum_{j=1}^n l_j} + \frac{1}{n}\right)$.[4]

Table 3: The commonly used notation.

Notation	Definition
$\mathbf{X} \in \mathbb{R}^{n\times d}$	each row is a data sample (feature vector)
$\mathbf{Y} \in \mathbb{R}^{n\times m}$	each row contains the corresponding responses
$\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$	the SVD of $\mathbf{X}$
ρ	the rank of $\mathbf{X}$
μ	the row coherence of $\mathbf{X}$
σ_i	the i-th largest singular value of $\mathbf{X}$
γ	the regularization parameter
β	$\beta = \frac{\|\mathbf{X}\|_2^2}{\|\mathbf{X}\|_2^2 + n\gamma} \leq 1$
$\mathbf{S} \in \mathbb{R}^{n\times s}$	the sketching matrix
$\mathbf{W}^\star \in \mathbb{R}^{d\times m}$	the optimal solution (2)
$\mathbf{W}^{\text{c}} \in \mathbb{R}^{d\times m}$	approximate solution obtained using the classical sketch (3)
$\mathbf{W}^{\text{h}} \in \mathbb{R}^{d\times m}$	approximate solution obtained using the Hessian sketch (4)
$\mathbf{W}_0 \in \mathbb{R}^{d\times m}$	the unknown ground truth (in the statistical setting)

The exact leverage scores are unnecessary in practice; constant-factor approximation to the leverage scores is sufficient. Leverage scores can be efficiently approximated by the algorithms of (Drineas et al., 2012). Let $l_1, \cdots, l_n$ be the true leverage scores. We denote the approximate leverages by $\widetilde{l}_1, \cdots, \widetilde{l}_n$ and require that they satisfy

$$\tilde{l}_q \in [l_q, \tau l_q] \quad \text{for all} \quad q \in [n], \tag{5}$$

where $\tau \geq 1$ indicates the quality of approximation. We then use $p_q = \tilde{l}_q / \sum_j \tilde{l}_j$ as the sampling probabilities. One can obtain the same accuracies when using approximate leverage scores in place of the true leverage scores by increasing s by a factor of τ, so as long as τ is a small constant, the orders of the sketch sizes when using exact or approximate leverage score sampling are the same. Thus we do not distinguish between exact and approximate leverage scores in this paper. For shrinked leverage score sampling, we define the sampling probabilities

$$p_i = \frac{1}{2}\left(\frac{\tilde{l}_i}{\sum_{j=1}^n \tilde{l}_j} + \frac{1}{n}\right) \quad \text{for} \quad i = 1, \ldots, n. \tag{6}$$

Gaussian projection is also well-known as the prototypical Johnson-Lindenstrauss transform (Johnson and Lindenstrauss, 1984). Let $\mathbf{G} \in \mathbb{R}^{n\times s}$ be a standard Gaussian matrix, i.e., each entry is sampled independently from $\mathcal{N}(0, 1)$. The matrix $\mathbf{S} = \frac{1}{\sqrt{s}}\mathbf{G}$ is a Gaussian projection matrix. It takes $\mathcal{O}(nds)$ time to apply $\mathbf{S} \in \mathbb{R}^{n\times s}$ to any $n \times d$ dense matrix, which makes Gaussian projection computationally inefficient relative to other forms of sketching.

4. In fact, p_i can be any convex combination of $\frac{l_i}{\sum_{j=1}^n l_j}$ and $\frac{1}{n}$ (Ma et al., 2015). We use the weight $\frac{1}{2}$ for convenience; our conclusions extend in a straightforward manner to other weightings.

The **Subsampled randomized Hadamard transform (SRHT)** (Drineas et al., 2011; Lu et al., 2013; Tropp, 2011) is a more efficient alternative to Gaussian projection. Let $\mathbf{H}_n \in \mathbb{R}^{n\times n}$ be the Walsh-Hadamard matrix with $+1$ and -1 entries, $\mathbf{D} \in \mathbb{R}^{n\times n}$ be a diagonal matrix with diagonal entries sampled uniformly from $\{+1, -1\}$, and $\mathbf{P} \in \mathbb{R}^{n\times s}$ be the uniform row sampling matrix defined above. The matrix $\mathbf{S} = \frac{1}{\sqrt{n}}\mathbf{D}\mathbf{H}_n\mathbf{P} \in \mathbb{R}^{n\times s}$ is an SRHT matrix, and can be applied to any $n \times d$ matrix in $\mathcal{O}(nd \log s)$ time. In practice, the subsampled randomized Fourier transform (SRFT) (Woolfe et al., 2008) is often used in lieu of the SRHT, because the SRFT exists for all values of n, whereas $\mathbf{H}_n$ exists only for some values of n. Their performance and theoretical analyses are very similar.

CountSketch can be applied to any $\mathbf{X} \in \mathbb{R}^{n\times d}$ in $\mathcal{O}(nd)$ time (Charikar et al., 2004; Clarkson and Woodruff, 2013; Meng and Mahoney, 2013; Nelson and Nguyên, 2013; Pham and Pagh, 2013; Weinberger et al., 2009). Though more efficient to apply, CountSketch requires a larger sketch size than Gaussian projections, SRHT, and leverage score sampling to attain the same theoretical guarantees. Interested readers can refer to (Woodruff, 2014) for a detailed description of CountSketch. Unlike the other sketching methods mentioned here, model averaging with CountSketch may not be theoretically sound. See Remark 5 for further discussion.

3. Main Results

Sections 3.1 and 3.2 analyze sketched MRR from, respectively, the optimization and statistical perspectives. Sections 3.3 and 3.4 capture the impacts of model averaging on, respectively, the optimization and statistical properties of sketched MRR.

We described six sketching methods in Section 2. For simplicity, in this section, we refer to leverage score sampling, shrinked leverage score sampling, Gaussian projection, and SRHT as **the four sketching methods** while we refer to uniform sampling and CountSketch by name. Throughout, let μ be the row coherence of $\mathbf{X}$ and $\beta = \frac{\|\mathbf{X}\|_2^2}{\|\mathbf{X}\|_2^2+n\gamma} \leq 1$.

3.1 Sketched MRR: Optimization Perspective

Theorem 1 shows that $f(\mathbf{W}^{\mathrm{c}})$, the objective value of classical sketch, is close to the optimal objective value $f(\mathbf{W}^\star)$, and that the approximation quality improves as the regularization parameter γ increases.

Theorem 1 (Classical Sketch) *Let $\beta = \frac{\|\mathbf{X}\|_2^2}{\|\mathbf{X}\|_2^2+n\gamma} \leq 1$. For the four sketching methods with $s = \tilde{\mathcal{O}}\big(\frac{d}{\epsilon}\big)$, uniform sampling with $s = \mathcal{O}\big(\frac{\mu d \log d}{\epsilon}\big)$, and CountSketch with $s = \mathcal{O}\big(\frac{d^2}{\epsilon}\big)$, the inequality*

$$f(\mathbf{W}^c) - f(\mathbf{W}^\star) \quad \leq \quad \epsilon\beta\, f(\mathbf{W}^\star)$$

holds with probability at least 0.9. The uncertainty is with respect to the random choice of sketching matrix.

The corresponding guarantee for the performance of Hessian sketch is given in Theorem 2. It is weaker than the guarantee for classical sketch, especially when $\frac{1}{n}\|\mathbf{Y}\|_F^2$ is far larger than $f(\mathbf{W}^\star)$. If $\mathbf{Y}$ is nearly noiseless—$\mathbf{Y}$ is well-explained by a linear combination

of the columns of $\mathbf{X}$—and γ is small, then $f(\mathbf{W}^\star)$ is close to zero, and consequently $f(\mathbf{W}^\star)$ can be far smaller than $\frac{1}{n}\|\mathbf{Y}\|_F^2$. Therefore, in this case which is ideal for MRR, $f(\mathbf{W}^{\mathrm{h}})$ is not close to $f(\mathbf{W}^\star)$ and our theory suggests Hessian sketch does not perform as well as classical sketch. This is verified by our experiments (see Figure 2), which show that unless γ is large or a large portion of $\mathbf{Y}$ is outside the column space of $\mathbf{X}$, the ratio $\frac{f(\mathbf{W}^{\mathrm{h}})}{f(\mathbf{W}^\star)}$ can be large.

Theorem 2 (Hessian Sketch) *Let $\beta = \frac{\|\mathbf{X}\|_2^2}{\|\mathbf{X}\|_2^2+n\gamma} \leq 1$. For the four sketching methods with $s = \tilde{\mathcal{O}}\big(\frac{d}{\epsilon}\big)$, uniform sampling with $s = \mathcal{O}\big(\frac{\mu d \log d}{\epsilon}\big)$, and CountSketch with $s = \mathcal{O}\big(\frac{d^2}{\epsilon}\big)$, the inequality*

$$f(\mathbf{W}^h) - f(\mathbf{W}^\star) \quad \leq \quad \epsilon\beta^2 \left(\frac{\|\mathbf{Y}\|_F^2}{n} - f(\mathbf{W}^\star)\right).$$

holds with probability at least 0.9. The uncertainty is with respect to the random choice of sketching matrix.

These two results imply that $f(\mathbf{W}^{\mathrm{c}})$ and $f(\mathbf{W}^{\mathrm{h}})$ can be close to $f(\mathbf{W}^\star)$. When this is the case, curvature of the objective function ensures that the sketched solutions $\mathbf{W}^{\mathrm{c}}$ and $\mathbf{W}^{\mathrm{h}}$ are close to the optimal solution $\mathbf{W}^\star$. Lemma 3 bounds the Mahalanobis distance $\|\mathbf{M}(\mathbf{W} - \mathbf{W}^\star)\|_F^2$. Here $\mathbf{M}$ is any non-singular matrix; in particular, it can be the identity matrix or $(\mathbf{X}^T\mathbf{X})^{1/2}$. Lemma 3 is a consequence of Lemma 25.

Lemma 3 (Mahalanobis Distance) *Let f be the objective function of MRR defined in (1), $\mathbf{W} \in \mathbb{R}^{d\times m}$ be arbitrary, and $\mathbf{W}^\star$ be the optimal solution defined in (2). For any non-singular matrix $\mathbf{M}$, the Mahalanobis distance satisfies*

$$\frac{1}{n}\big\|\mathbf{M}(\mathbf{W} - \mathbf{W}^\star)\big\|_F^2 \;\leq\; \frac{f(\mathbf{W}) - f(\mathbf{W}^\star)}{\sigma_{\min}^2\big[(\mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{X} + n\gamma\mathbf{I}_d)^{1/2}\mathbf{M}^{-1}\big]}.$$

By choosing $\mathbf{M} = (\mathbf{X}^T\mathbf{X})^{1/2}$, we can bound $\frac{1}{n}\|\mathbf{X}\mathbf{W} - \mathbf{X}\mathbf{W}^\star\|_F^2$ in terms of the difference in the objective values:

$$\tfrac{1}{n}\big\|\mathbf{X}\mathbf{W} - \mathbf{X}\mathbf{W}^\star\big\|_F^2 \quad \leq \quad \beta\big[f(\mathbf{W}) - f(\mathbf{W}^\star)\big],$$

where $\beta = \frac{\|\mathbf{X}\|_2^2}{\|\mathbf{X}\|_2^2+n\gamma} \leq 1$. With Lemma 3, we can directly apply Theorems 1 or 2 to bound $\frac{1}{n}\|\mathbf{X}\mathbf{W}^{\mathrm{c}} - \mathbf{X}\mathbf{W}^\star\|_F^2$ or $\frac{1}{n}\|\mathbf{X}\mathbf{W}^{\mathrm{h}} - \mathbf{X}\mathbf{W}^\star\|_F^2$.

3.2 Sketched MRR: Statistical Perspective

We consider the following fixed design model. Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ be the observed feature matrix, $\mathbf{W}_0 \in \mathbb{R}^{d\times m}$ be the true and unknown model, $\mathbf{\Xi} \in \mathbb{R}^{n\times m}$ contain unknown random noise, and

$$\mathbf{Y} \;=\; \mathbf{X}\mathbf{W}_0 + \mathbf{\Xi} \tag{7}$$

be the observed responses. We make the following standard weak assumptions on the noise:

$$\mathbb{E}[\mathbf{\Xi}] = \mathbf{0} \quad \text{and} \quad \mathbb{E}[\mathbf{\Xi}\mathbf{\Xi}^T] = \xi^2\mathbf{I}_n.$$

We observe $\mathbf{X}$ and $\mathbf{Y}$ and seek to estimate $\mathbf{W}_0$.

We can evaluate the quality of the estimate by the risk:

$$R(\mathbf{W}) \;=\; \tfrac{1}{n}\mathbb{E}\big\|\mathbf{X}\mathbf{W}-\mathbf{X}\mathbf{W}_0\big\|_F^2, \tag{8}$$

where the expectation is taken w.r.t. the noise $\boldsymbol{\Xi}$. We study the risk functions $R(\mathbf{W}^\star)$, $R(\mathbf{W}^{\mathrm{c}})$, and $R(\mathbf{W}^{\mathrm{h}})$ in the following.

Theorem 4 (Bias-Variance Decomposition) *We consider the data model described in this subsection. Let* $\mathbf{W}$ *be* $\mathbf{W}^\star$, $\mathbf{W}^c$, *or* $\mathbf{W}^h$, *as defined in* (2), (3), *or* (4), *respectively; then the risk function can be decomposed as*

$$R(\mathbf{W}) \;=\; \mathsf{bias}^2(\mathbf{W})+\mathsf{var}(\mathbf{W}).$$

Recall the SVD of $\mathbf{X}$ *defined in Section 2:* $\mathbf{X}=\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$. *The bias and variance terms can be written as*

$$\begin{aligned}
\mathsf{bias}(\mathbf{W}^\star) &= \gamma\sqrt{n}\Big\|(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{W}_0\Big\|_F,\\
\mathsf{var}(\mathbf{W}^\star) &= \tfrac{\xi^2}{n}\Big\|\big(\mathbf{I}_\rho+n\gamma\boldsymbol{\Sigma}^{-2}\big)^{-1}\Big\|_F^2,\\
\mathsf{bias}(\mathbf{W}^c) &= \gamma\sqrt{n}\Big\|\big(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2}\big)^\dagger\boldsymbol{\Sigma}^{-1}\mathbf{V}^T\mathbf{W}_0\Big\|_F,\\
\mathsf{var}(\mathbf{W}^c) &= \tfrac{\xi^2}{n}\Big\|\big(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2}\big)^\dagger\mathbf{U}^T\mathbf{S}\mathbf{S}^T\Big\|_F^2,\\
\mathsf{bias}(\mathbf{W}^h) &= \gamma\sqrt{n}\Big\|\Big(\boldsymbol{\Sigma}^{-2}+\tfrac{\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}-\mathbf{I}_\rho}{n\gamma}\Big)\big(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2}\big)^\dagger\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{W}_0\Big\|_F,\\
\mathsf{var}(\mathbf{W}^h) &= \tfrac{\xi^2}{n}\Big\|\big(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2}\big)^\dagger\Big\|_F^2.
\end{aligned}$$

The functions $\mathsf{bias}(\mathbf{W}^\star)$ and $\mathsf{var}(\mathbf{W}^\star)$ are deterministic. The randomness in $\mathsf{bias}(\mathbf{W}^{\mathrm{c}})$, $\mathsf{var}(\mathbf{W}^{\mathrm{c}})$, $\mathsf{bias}(\mathbf{W}^{\mathrm{h}})$, and $\mathsf{var}(\mathbf{W}^{\mathrm{h}})$ all arises from the sketching matrix $\mathbf{S}$.

Throughout this paper, we compare the bias and variance of classical sketch and Hessian sketch to those of the optimal solution $\mathbf{W}^\star$. We first study the bias, variance, and risk of $\mathbf{W}^\star$, which will help us understand the subsequent comparisons. We can assume that $\boldsymbol{\Sigma}^2=\mathbf{V}^T\mathbf{X}^T\mathbf{X}\mathbf{V}$ is linear in n; this is reasonable because $\mathbf{X}^T\mathbf{X}=\sum_{i=1}^n\mathbf{x}_i\mathbf{x}_i^T$ and $\mathbf{V}$ is an orthogonal matrix.

- **Bias.** The bias of $\mathbf{W}^\star$ is independent of n and is increasing with γ. The bias is the price paid for using regularization to decrease the variance; for least squares regression, γ is zero, and the bias is zero.
- **Variance.** The variance of $\mathbf{W}^\star$ is inversely proportional to n. As n grows, the variance decreases to zero, and we must also decrease γ to ensure that the sum of the squared bias and variance decreases to zero.
- **Risk.** Note that $\mathbf{W}^\star$ is not the minimizer of $R(\cdot)$; $\mathbf{W}_0$ is the minimizer because $R(\mathbf{W}_0)=0$. Nevertheless, because $\mathbf{W}_0$ is unknown, $\mathbf{W}^\star$ for a carefully chosen γ is a standard proxy for the exact minimizer in practice. It is thus highly interesting to compare the risk of MRR solutions obtained using sketching to to $R(\mathbf{W}^\star)$.

Theorem 5 provides upper and lower bounds on the bias and variance of solutions obtained using classical sketch. In particular, we see that that $\mathsf{bias}(\mathbf{W}^{\mathrm{c}})$ is within a factor of $(1\pm\epsilon)$ of $\mathsf{bias}(\mathbf{W}^\star)$. However, $\mathsf{var}(\mathbf{W}^{\mathrm{c}})$ can be $\Theta(\frac{n}{s})$ times worse than $\mathsf{var}(\mathbf{W}^\star)$. The absolute value of $\mathsf{var}(\mathbf{W}^{\mathrm{c}})$ is inversely proportional to s, whereas the absolute value of $\mathsf{bias}(\mathbf{W}^{\mathrm{c}})$ is almost independent of s.

Theorem 5 (Classical Sketch) *Assume* $s\le n$. *For Gaussian projection and SRHT sketching with* $s=\tilde{\mathcal{O}}(\frac{d}{\epsilon^2})$, *uniform sampling with* $s=\mathcal{O}(\frac{\mu d\log d}{\epsilon^2})$, *or CountSketch with* $s=\mathcal{O}(\frac{d^2}{\epsilon^2})$, *the inequalities*

$$\begin{aligned}
1-\epsilon &\le \tfrac{\mathsf{bias}(\mathbf{W}^c)}{\mathsf{bias}(\mathbf{W}^\star)} \le 1+\epsilon,\\
(1-\epsilon)\tfrac{n}{s} &\le \tfrac{\mathsf{var}(\mathbf{W}^c)}{\mathsf{var}(\mathbf{W}^\star)} \le (1+\epsilon)\tfrac{n}{s}
\end{aligned}$$

hold with probability at least 0.9. For shrinked leverage score sampling with $s=\mathcal{O}(\frac{d\log d}{\epsilon^2})$, *these inequalities, except for the lower bound on the variance, hold with probability at least 0.9. Here the randomness comes from the sketching matrix* $\mathbf{S}$.

Remark 1 *To establish an upper (lower) bound on the variance, we need an upper (lower) bound on* $\|\mathbf{S}\|_2^2$. *There is no nontrivial upper nor lower bound on* $\|\mathbf{S}\|_2^2$ *for leverage score sampling, so the variance of leverage score sampling cannot be bounded. Shrinked leverage score sampling satisfies the upper bound* $\|\mathbf{S}\|_2^2\le\frac{2n}{s}$; *but* $\|\mathbf{S}\|_2^2$ *does not have a nontrivial lower bound, so there is no nontrivial lower bound on the variance of shrinked leverage score. Remark 4 explains the nonexistence of the relevant bounds on* $\|\mathbf{S}\|_2^2$ *for both variants of leverage score sampling.*

Theorem 6 establishes similar upper and lower bounds on the bias and variance of solutions obtained using Hessian sketch. The situation is the reverse of that with classical sketch: the variance of $\mathbf{W}^{\mathrm{h}}$ is close to that of $\mathbf{W}^\star$ if s is large enough, but as the regularization parameter γ goes to zero, $\mathsf{bias}(\mathbf{W}^{\mathrm{h}})$ becomes much larger than $\mathsf{bias}(\mathbf{W}^\star)$. The theory suggest that Hessian sketch should be preferred over classical sketch when $\mathbf{Y}$ is very noisy, because Hessian sketch does not magnify the variance.

Theorem 6 (Hessian Sketch) *For the four sketching methods with* $s=\tilde{\mathcal{O}}(\frac{d}{\epsilon^2})$, *uniform sampling with* $s=\mathcal{O}(\frac{\mu d\log d}{\epsilon^2})$, *and CountSketch with* $s=\mathcal{O}(\frac{d^2}{\epsilon^2})$, *the inequalities*

$$\begin{aligned}
\tfrac{\mathsf{bias}(\mathbf{W}^h)}{\mathsf{bias}(\mathbf{W}^\star)} &\le (1+\epsilon)\big(1+\tfrac{\epsilon\|\mathbf{X}\|_2^2}{n\gamma}\big),\\
1-\epsilon &\le \tfrac{\mathsf{var}(\mathbf{W}^h)}{\mathsf{var}(\mathbf{W}^\star)} \le 1+\epsilon
\end{aligned}$$

hold with probability at least 0.9. Further assume that the ρ*-th singular value of* $\mathbf{X}$ *satisfies* $\sigma_\rho^2\ge\frac{n\gamma}{\epsilon}$, *then*

$$\tfrac{\mathsf{bias}(\mathbf{W}^h)}{\mathsf{bias}(\mathbf{W}^\star)} \;\ge\; \tfrac{1}{1+\epsilon}\big(\tfrac{\epsilon\sigma_\rho^2}{n\gamma}-1\big)$$

with probability at least 0.9. Here the randomness is in the choice of sketching matrix $\mathbf{S}$.

The lower bound on the bias shows that the solution from Hessian sketch can exhibit a much higher bias than the optimal solution. The gap between $\mathsf{bias}(\mathbf{W}^{\mathrm{h}})$ and $\mathsf{bias}(\mathbf{W}^{\star})$ can be lessened by increasing the regularization parameter γ, but such over-regularization increases the baseline $\mathsf{bias}(\mathbf{W}^{\star})$ itself. It is also worth mentioning that unlike $\mathsf{bias}(\mathbf{W}^{\star})$ and $\mathsf{bias}(\mathbf{W}^{\mathrm{c}})$, $\mathsf{bias}(\mathbf{W}^{\mathrm{h}})$ is not monotonically increasing with γ, as is empirically verified in Figure 3.

In sum, our theory shows that the classical and Hessian sketches are not statistically comparable to the optimal solutions: classical sketch has too high a variance, and Hessian sketch has too high a bias for reasonable amounts of regularization. In practice, the regularization parameter γ should be tuned to optimize the prediction accuracy. Our experiments in Figure 3 show that even with carefully chosen γ, the risks of classical and Hessian sketch can be higher than the risk of the optimal solution by an order of magnitude. Formally speaking, $\min_\gamma R(\mathbf{W}^{\mathrm{c}}) \gg \min_\gamma R(\mathbf{W}^{\star})$ and $\min_\gamma R(\mathbf{W}^{\mathrm{h}}) \gg \min_\gamma R(\mathbf{W}^{\star})$ hold in practice.

Our empirical study in Figure 3 suggests classical and Hessian sketch both require over-regularization, i.e., setting γ larger than is best for the optimal solution $\mathbf{W}^{\star}$. Formally speaking, $\operatorname{argmin}_\gamma R(\mathbf{W}^{\mathrm{c}}) > \operatorname{argmin}_\gamma R(\mathbf{W}^{\star})$ and $\operatorname{argmin}_\gamma R(\mathbf{W}^{\mathrm{h}}) > \operatorname{argmin}_\gamma R(\mathbf{W}^{\star})$. Although this is the case for both types of sketching, the underlying explanations are different. Classical sketches have a high variance, so a large γ is required to supress their variance (the variance is non-increasing with γ). Hessian sketches magnify the bias when γ is small, so a reasonably large γ is necessary to lower their bias.

3.3 Model Averaging: Optimization Perspective

We consider model averaging as a method to increase the accuracy of sketched MRR solutions. The model averaging procedure is straightforward: one independently draws g sketching matrices $\mathbf{S}_1, \cdots, \mathbf{S}_g \in \mathbb{R}^{n\times s}$, uses these to form g sketched MRR solutions, denoted by $\{\mathbf{W}_i^{\mathrm{c}}\}_{i=1}^g$ or $\{\mathbf{W}_i^{\mathrm{h}}\}_{i=1}^g$, and averages these solutions to obtain the final estimate $\mathbf{W}^{\mathrm{c}} = \frac{1}{g}\sum_{i=1}^g \mathbf{W}_i^{\mathrm{c}}$ or $\mathbf{W}^{\mathrm{h}} = \frac{1}{g}\sum_{i=1}^g \mathbf{W}_i^{\mathrm{h}}$. Practical applications of model averaging are enumerated in Section 1.1.

Theorems 7 and 8 present guarantees on the optimization accuracy of using model averaging on classical/Hessian sketch solutions. We can contrast these with the guarantees provided for sketched MRR in Theorems 1 and 2. For classical sketch with model averaging, we see that when $\epsilon \le \frac{1}{g}$, the bound on $f(\mathbf{W}^{\mathrm{h}}) - f(\mathbf{W}^{\star})$ is proportional to ϵ/g. From Lemma 3 we see that the distance between $\mathbf{W}^{\mathrm{c}}$ and $\mathbf{W}^{\star}$ also decreases accordingly.

Theorem 7 (Classical Sketch with Model Averaging) *Let $\beta = \frac{\|\mathbf{X}\|_2^2}{\|\mathbf{X}\|_2^2+n\gamma} \le 1$. For the four methods, let $s = \tilde{\mathcal{O}}\big(\frac{d}{\epsilon}\big)$, and for uniform sampling, let $s = \mathcal{O}\big(\frac{\mu d\log d}{\epsilon}\big)$, then the inequality*

$$f(\mathbf{W}^{\mathrm{c}}) - f(\mathbf{W}^{\star}) \;\le\; \beta\Big(\frac{\epsilon}{g} + \beta^2\epsilon^2\Big) f(\mathbf{W}^{\star})$$

holds with probability at least 0.8. Here the randomness comes from the choice of sketching matrices.

For Hessian sketch with model averaging, if $\epsilon < \frac{1}{g}$, then the bound on $f(\mathbf{W}^{\mathrm{h}}) - f(\mathbf{W}^{\star})$ is proportional to $\frac{\epsilon}{g}$.

Theorem 8 (Hessian Sketch with Model Averaging) *Let $\beta = \frac{\|\mathbf{X}\|_2^2}{\|\mathbf{X}\|_2^2+n\gamma} \le 1$. For the four methods let $s = \tilde{\mathcal{O}}\big(\frac{d}{\epsilon}\big)$, and for uniform sampling let $s = \mathcal{O}\big(\frac{\mu d\log d}{\epsilon}\big)$, then the inequality*

$$f(\mathbf{W}^{h}) - f(\mathbf{W}^{\star}) \;\le\; \beta^2\Big(\frac{\epsilon}{g} + \epsilon^2\Big)\Big(\frac{\|\mathbf{Y}\|_F^2}{n} - f(\mathbf{W}^{\star})\Big)$$

holds with probability at least 0.8. Here the randomness comes from the choice of sketching matrices.

3.4 Model Averaging: Statistical Perspective

Model averaging has the salutatory property of reducing the risks of the classical and Hessian sketches. Our first result conducts a bias-variance decomposition for the averaged solution of the sketched MRR problem.

Theorem 9 (Bias-Variance Decomposition) *We consider the fixed design model* (7). *Decompose the risk function defined in* (8) *as*

$$R(\mathbf{W}) \;=\; \mathsf{bias}^2(\mathbf{W}) + \mathsf{var}(\mathbf{W}).$$

The bias and variance terms are

$$\begin{aligned}
\mathsf{bias}\big(\mathbf{W}^{c}\big) &= \gamma\sqrt{n}\Big\|\frac{1}{g}\sum_{i=1}^g \big(\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{U} + n\gamma\mathbf{\Sigma}^{-2}\big)^{\dagger}\mathbf{\Sigma}^{-1}\mathbf{V}^T\mathbf{W}_0\Big\|_F,\\
\mathsf{var}\big(\mathbf{W}^{c}\big) &= \frac{\xi^2}{n}\Big\|\frac{1}{g}\sum_{i=1}^g \big(\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{U} + n\gamma\mathbf{\Sigma}^{-2}\big)^{\dagger}\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\Big\|_F^2,\\
\mathsf{bias}\big(\mathbf{W}^{h}\big) &= \gamma\sqrt{n}\Big\|\frac{1}{g}\sum_{i=1}^g \big(\mathbf{\Sigma}^{-2} + \tfrac{\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{U}-\mathbf{I}_\rho}{n\gamma}\big)\big(\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{U} + n\gamma\mathbf{\Sigma}^{-2}\big)^{\dagger}\mathbf{\Sigma}\mathbf{V}^T\mathbf{W}_0\Big\|_F,\\
\mathsf{var}\big(\mathbf{W}^{h}\big) &= \frac{\xi^2}{n}\Big\|\frac{1}{g}\sum_{i=1}^g \big(\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{U} + n\gamma\mathbf{\Sigma}^{-2}\big)^{\dagger}\Big\|_F^2.
\end{aligned}$$

Theorems 10 and 11 provide upper bounds on the bias and variance of averaged sketched MRR solutions for, respectively, classical sketch and Hessian sketch. We can contrast them with Theorems 5 and 6 to see the statistical benefits of model averaging. Theorem 10 shows that when $g \approx \frac{n}{s}$, classical sketch with model averaging yields a solution with comparable bias and variance to the optimal solution.

Theorem 10 (Classical Sketch with Model Averaging) *For the four sketching methods with $s = \tilde{\mathcal{O}}\big(\frac{d}{\epsilon^2}\big)$, or uniform sampling with $s = \mathcal{O}\big(\frac{\mu d\log d}{\epsilon^2}\big)$, the inequalities*

$$\frac{\mathsf{bias}(\mathbf{W}^{c})}{\mathsf{bias}(\mathbf{W}^{\star})} \;\le\; 1+\epsilon \qquad \text{and} \qquad \frac{\mathsf{var}(\mathbf{W}^{c})}{\mathsf{var}(\mathbf{W}^{\star})} \;\le\; \frac{n}{s}\Big(\frac{\sqrt{1+\epsilon}}{\sqrt{h}} + \epsilon\Big)^2,$$

where $h = \min\{g, \Theta(\frac{n}{s})\}$, hold with probability at least 0.8. The randomness comes from the choice of sketching matrices.

Theorem 11 shows that model averaging decreases the bias of Hessian sketch without increasing the variance. For Hessian sketch without model averaging, recall that $\mathsf{bias}(\mathbf{W}^{\mathrm{h}})$ is larger than $\mathsf{bias}(\mathbf{W}^{\star})$ by a factor of $\mathcal{O}(\|\mathbf{X}\|_2^2/(n\gamma))$. Theorem 11 shows that model averaging significantly reduces the bias.

Theorem 11 (Hessian Sketch with Model Averaging) *For the four sketching methods with $s = \tilde{\mathcal{O}}\big(\frac{d}{\epsilon^2}\big)$, or uniform sampling with $s = \mathcal{O}\big(\frac{\mu d \log d}{\epsilon^2}\big)$, the inequalities*

$$\frac{\mathsf{bias}(\mathbf{W}^h)}{\mathsf{bias}(\mathbf{W}^\star)} \leq 1 + \epsilon + \Big(\frac{\epsilon}{\sqrt{g}} + \epsilon^2\Big)\frac{\|\mathbf{X}\|_2^2}{n\gamma} \qquad \textit{and} \qquad \frac{\mathsf{var}(\mathbf{W}^h)}{\mathsf{var}(\mathbf{W}^\star)} \leq 1 + \epsilon$$

hold with probability at least 0.8. Here the randomness comes from the choice of sketching matrices.

4. Experiments on Synthetic Data

We conduct experiments on synthetic data to verify our theory. Section 4.1 describes the data model and experiment settings. Sections 4.2 and 4.3 empirically study classical and Hessian sketch from the optimization and statistical perspectives, respectively, to verify Theorems 1, 2, 5, and 6. Sections 4.4 and 4.5 study model averaging from the optimization and statistical perspectives, respectively, to corroborate Theorems 7, 8, 10, and 11.

4.1 Settings

Following (Ma et al., 2015; Yang et al., 2016), we construct $\mathbf{X} = \mathbf{U}\mathsf{diag}(\boldsymbol{\sigma})\mathbf{V}^T \in \mathbb{R}^{n\times d}$ and $\mathbf{y} = \mathbf{X}\mathbf{w}_0 + \boldsymbol{\varepsilon} \in \mathbb{R}^n$ in the following way.

- We take $\mathbf{U}$ be the matrix of left singular vectors of $\mathbf{A} \in \mathbb{R}^{n\times d}$ which is constructed in the following way. (Note that $\mathbf{A}$ and $\mathbf{X}$ are different.) Let the rows of $\mathbf{A}$ be i.i.d. sampled from a multivariate t-distribution with covariance matrix $\mathbf{C}$ and $v = 2$ degree of freedom, where the (i,j)-th entry of $\mathbf{C} \in \mathbb{R}^{d\times d}$ is $2 \times 0.5^{|i-j|}$. Constructing $\mathbf{A}$ in this manner ensures that it has high row coherence.
- Let the entries of $\mathbf{b} \in \mathbb{R}^d$ be equally spaced between 0 and -6 and take $\sigma_i = 10^{b_i}$ for all $i \in [d]$.
- Let $\mathbf{V} \in \mathbb{R}^{d\times d}$ be an orthonormal basis for the column range of a $d \times d$ standard Gaussian matrix.
- Let $\mathbf{w}_0 = [\mathbf{1}_{0.2d}; 0.1\,\mathbf{1}_{0.6d}; \mathbf{1}_{0.2d}]$.
- Take the entries of $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ to be i.i.d. samples from the $\mathcal{N}(0, \xi^2)$ distribution.

This construction ensures that $\mathbf{X}$ has high row coherence, and its condition number is $\kappa(\mathbf{X}^T\mathbf{X}) = 10^{12}$. Let $\mathbf{S} \in \mathbb{R}^{n\times s}$ be any of the six sketching methods considered in this paper. We fix $n = 10^5$, $d = 500$, and $s = 5,000$. Since the sketching methods are randomized, we repeat each trial 10 times with idependent sketches and report averaged results.

4.2 Sketched MRR: Optimization Perspective

We seek to empirically verify Theorems 1 and 2 which study classical and Hessian sketches, respective, from the optimization perspective. In Figure 2, we plot the objective function value $f(\mathbf{w}) = \frac{1}{n}\|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \gamma\|\mathbf{w}\|_2^2$ against γ, under different settings of ξ (the standard deviation of the Gaussian noise added to the response). The black curves correspond to the optimal solution $\mathbf{w}^\star$; the color curves correspond to classical or Hessian sketch with different

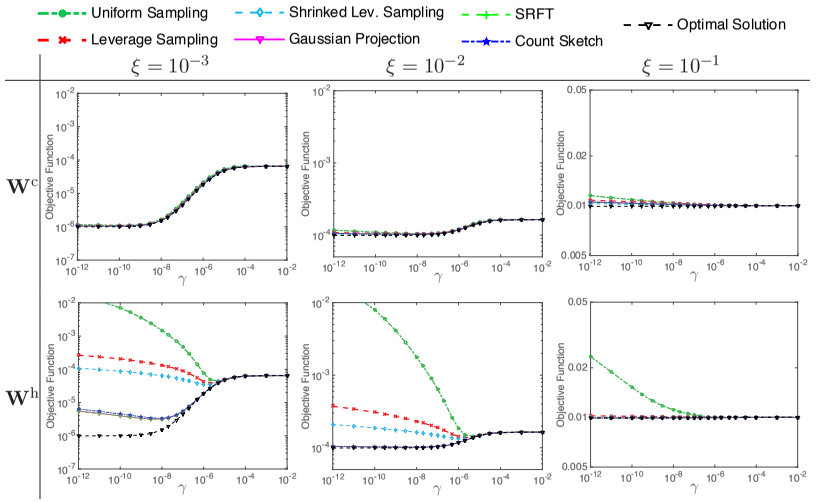


Figure 2: An empirical study of classical and Hessian sketch from the optimization perspective. The x-axis is the regularization parameter γ (log scale); the y-axis is the objective function values (log scale). Here ξ is the standard deviation of the Gaussian noise added to the response.

sketching methods. The results verify our theory: the objective value of the solution from the classical sketch, $\mathbf{w}^c$, is always close to optimal; and the objective value of the solution from the Hessian sketch, $\mathbf{w}^h$, is much worse than the optimal value when γ is small and $\mathbf{y}$ is mostly in the column space of $\mathbf{X}$.

4.3 Sketched MRR: Statistical Perspective

In Figure 3, we plot the analytical expressions for the squared bias, variance, and risk stated in Theorem 4 against the regularization parameter γ. Because these expressions involve the random sketching matrix $\mathbf{S}$, we randomly generate $\mathbf{S}$, repeat this procedure 10 times, and report the average of the computed squared biases, variances, and risks. We fix $\xi = 0.1$ (the standard deviation of the Gaussian noise). The results of this experiment match our theory: classical sketch magnified the variance, and Hessian sketch increased the bias. Even when γ is fine-tuned, the risks of classical and Hessian sketch can be much higher than those of the optimal solution. Our experiment also indicates that classical and Hessian sketch require setting γ larger than the best regularization parameter for the optimal solution $\mathbf{W}^\star$.

Classical and Hessian sketch do not outperform each other in terms of the risk. When variance dominates bias, Hessian sketch is better in terms of the risk; when bias dominates variance, classical sketch is preferable. In the experiment yielding Figure 3, Hessian sketch

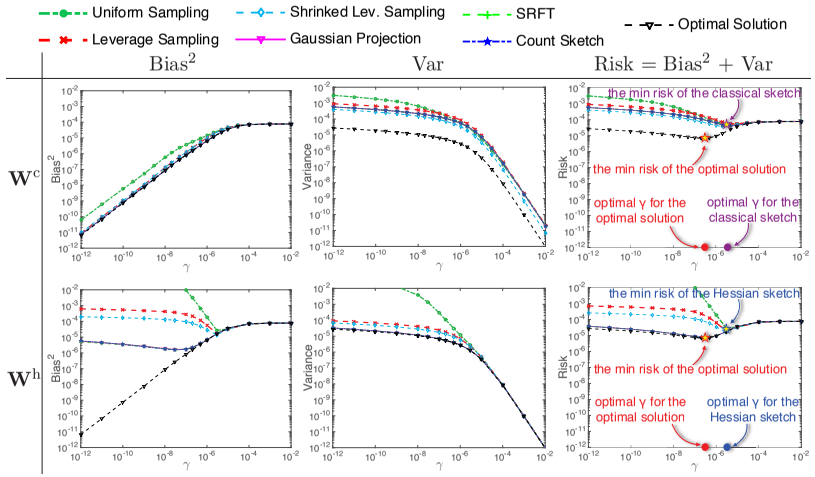


Figure 3: An empirical study of classical sketch and Hessian sketch from the statistical perspective. The x-axis is the regularization parameter γ (log-scale); the y-axes are respectively bias^2, variance, and risk (log-scale). We indicate the minimum risks and optimal choice of γ in the plots.

delivers lower risks than classical sketch. This is not generally true: if we use a smaller ξ (the standard deviation of the Gaussian noise), so that the variance is dominated by bias, then classical sketch results in lower risks than Hessian sketch.

4.4 Model Averaging: Optimization Objective

We consider different noise levels by setting $\xi = 10^{-2}$ or 10^{-1}, where ξ is defined in Section 4.1 as the standard deviation of the Gaussian noise in the response vector $\mathbf{y}$. We calculate the objective function values $f(\mathbf{w}^{\mathrm{c}}_{[g]})$ and $f(\mathbf{w}^{\mathrm{h}}_{[g]})$ for different settings of g, γ. We use different methods of sketching at the fixed sketch size $s = 5,000$.

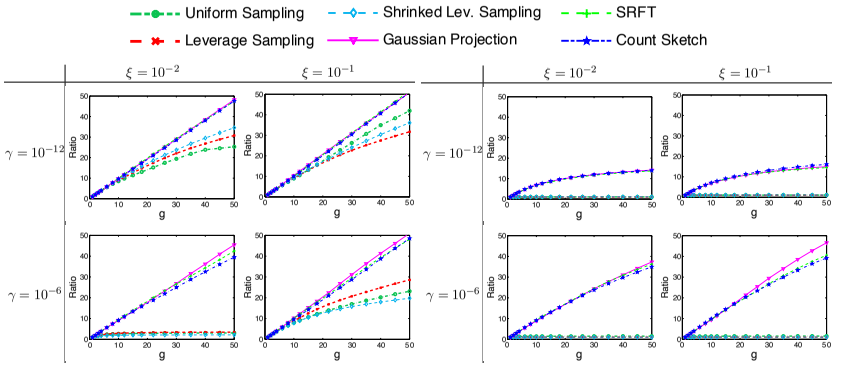
Theorem 7 indicates that for large s, e.g., Gaussian projection with $s = \tilde{\mathcal{O}}(\frac{d}{\epsilon})$,

$$f(\mathbf{w}^{\mathrm{c}}_{[g]}) - f(\mathbf{w}^\star) \quad \leq \quad \beta\left(\tfrac{\epsilon}{g} + \beta^2\epsilon^2\right) f(\mathbf{w}^\star), \tag{9}$$

where $\beta = \frac{\|\mathbf{X}\|_2^2}{\|\mathbf{X}\|_2^2 + n\gamma} \leq 1$. In Figure 4(a) we plot the ratio

$$\frac{f(\mathbf{w}^{\mathrm{c}}_{[1]}) - f(\mathbf{w}^\star)}{f(\mathbf{w}^{\mathrm{c}}_{[g]}) - f(\mathbf{w}^\star)} \tag{10}$$

against g. Rapid growth of this ratio indicates that model averaging is highly effective. The results in Figure 4(a) indicate that model averaging significantly improves the accuracy



(a) Classical sketch with model averaging. (b) Hessian sketch with model averaging.

Figure 4: An empirical study of model averaging from the optimization perspective. The x-axis is g, i.e., the number of models that are averaged. In 4(a), the y-axis is the ratio (log-scale) defined in (10). In 4(b), the y-axis is the ratio (log-scale) defined in (11). Here γ is the regularization parameter and ξ is the standard deviation of the Gaussian noise.

as measured by the objective function value. For the three random projection methods, the growth rate of this ratio is almost linear in g. In Figure 4(a), we observe that the regularization parameter γ affects the ratio (10). The ratio grows faster when $\gamma = 10^{-12}$ than when $\gamma = 10^{-6}$. This phenomenon is not explained by our theory.

Theorem 8 shows that for large sketch size s, e.g., Gaussian projection with $s = \tilde{\mathcal{O}}(\frac{d}{\epsilon})$,

$$f(\mathbf{w}^{\mathrm{h}}) - f(\mathbf{w}^\star) \quad \leq \quad \beta^2\left(\tfrac{\epsilon}{g} + \epsilon^2\right)\left(\tfrac{\|\mathbf{y}\|_2^2}{n} - f(\mathbf{w}^\star)\right),$$

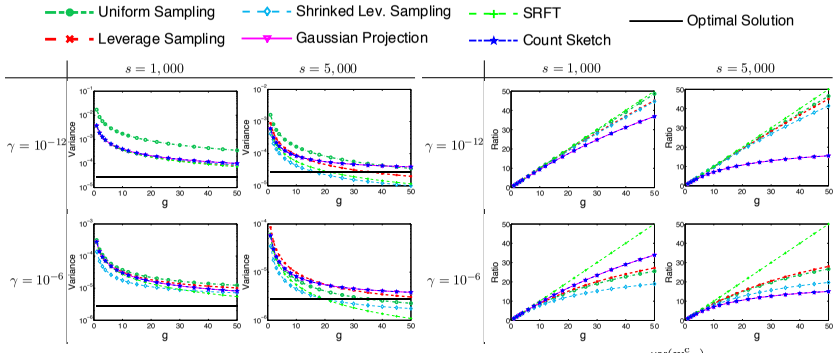
where $\beta = \frac{\|\mathbf{X}\|_2^2}{\|\mathbf{X}\|_2^2 + n\gamma} \leq 1$. In Figure 4(b), we plot the ratio

$$\frac{f(\mathbf{w}^{\mathrm{h}}_{[1]}) - f(\mathbf{w}^\star)}{f(\mathbf{w}^{\mathrm{h}}_{[g]}) - f(\mathbf{w}^\star)} \tag{11}$$

against g. Rapid growth of this ratio indicates that model averaging is highly effective. Our empirical results indicate that the growth rate of this ratio is moderately rapid for very small g and very slow for large g.

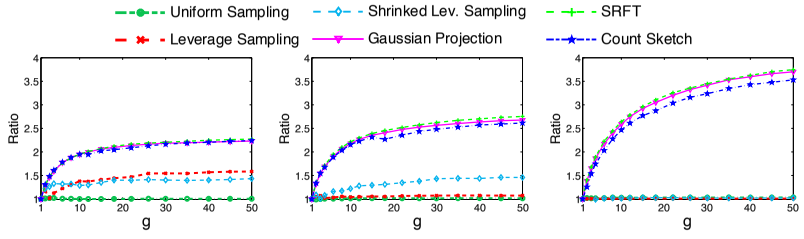
4.5 Model Averaging: Statistical Perspective

We empirically study model averaging from the statistical perspective. We calculate the bias and variance $\mathsf{bias}(\mathbf{w}^\star)$, $\mathsf{var}(\mathbf{w}^\star)$ of the optimal MRR solution according to Theorem 4



(a) The variance $\mathsf{var}(\mathbf{w}^{\mathrm{c}}_{[g]})$. (b) The ratio $\frac{\mathsf{var}(\mathbf{w}^{\mathrm{c}}_{[1]})}{\mathsf{var}(\mathbf{w}^{\mathrm{c}}_{[g]})}$.

Figure 5: An empirical study of the variance of classical sketch with model averaging. The x-axis is g, i.e., the number of models that are averaged. In 5(a), the y-axis is the variance $\mathsf{var}(\mathbf{w}^{\mathrm{c}}_{[g]})$ (log scale) defined in Theorem 9. In 5(b), the y-axis is the ratio $\frac{\mathsf{var}(\mathbf{w}^{\mathrm{c}}_{[1]})}{\mathsf{var}(\mathbf{w}^{\mathrm{c}}_{[g]})}$. Here γ is the regularization parameter and s is the sketch size.



(a) $s = 1,000$ (b) $s = 2,000$ (c) $s = 5,000$

Figure 6: An empirical study of the bias of Hessian sketch with model averaging. The x-axis is g, the number of models being averaged; the y-axis is the ratio (12).

and the bias and variance $\mathsf{bias}(\mathbf{w}^{\mathrm{c}}_{[g]})$, $\mathsf{var}(\mathbf{w}^{\mathrm{c}}_{[g]})$ and $\mathsf{bias}(\mathbf{w}^{\mathrm{h}}_{[g]})$, $\mathsf{var}(\mathbf{w}^{\mathrm{h}}_{[g]})$ of, respectively, the model averaged classical sketch solution and the model averaged Hessian sketch solution according to Theorem 9.

4.5.1 Classical Sketch

Theorem 10 indicates that for large enough s, e.g., Gaussian projection with $s = \tilde{\mathcal{O}}(\frac{d}{\epsilon^2})$, with high probability

$$\frac{\mathsf{bias}(\mathbf{w}^{\mathrm{c}}_{[g]})}{\mathsf{bias}(\mathbf{w}^\star)} \leq 1+\epsilon \qquad \text{and} \qquad \frac{\mathsf{var}(\mathbf{w}^{\mathrm{c}}_{[g]})}{\mathsf{var}(\mathbf{w}^\star)} \leq \frac{n}{s}\left(\sqrt{\frac{1+\epsilon}{h}}+\epsilon\right)^2,$$

where $h = \min\{g, \Theta(\frac{n}{s})\}$. This result implies that model averaging decreases the variance of classical sketch without significantly changing the bias. We conduct experiments to verify this point.

In Figure 5(a) we plot the variance $\mathsf{var}(\mathbf{w}^{\mathrm{c}}_{[g]})$ against g; the variance of the optimal solution $\mathbf{w}^\star$ is depicted for comparison. Clearly, the variance drops as g grows. In particular, when s is big ($s = 5,000$) and g exceeds $\frac{n}{s}$ ($= \frac{100,000}{5,000} = 20$), $\mathsf{var}(\mathbf{w}^{\mathrm{c}}_{[g]})$ can be even lower than $\mathsf{var}(\mathbf{w}^\star)$.

To more clearly decrease the impact of model averaging on the variance, in Figure 5(b) we plot the ratio $\frac{\mathsf{var}(\mathbf{w}^{\mathrm{c}}_{[1]})}{\mathsf{var}(\mathbf{w}^{\mathrm{c}}_{[g]})}$ against g. According to Theorem 10, this ratio grows linearly in g when s is at least $\tilde{\mathcal{O}}(dg)$, and otherwise is sublinear in g. This claim is verified by the empirical results in Figure 5(b).

When $\mathsf{bias}(\mathbf{w}^{\mathrm{c}}_{[g]})$ is plotted as a function of g, the curves are almost horizontal, indicating that, as expected, *the bias is insensitive to the number of models g*. We do not show such plots because these nearly horizontal curves are not interesting.

4.5.2 Hessian Sketch

Theorem 11 indicates that for large enough s, e.g., Gaussian projection with $s = \tilde{\mathcal{O}}(\frac{d}{\epsilon^2})$, the inequalities

$$\frac{\mathsf{bias}(\mathbf{w}^{\mathrm{h}}_{[g]})}{\mathsf{bias}(\mathbf{w}^\star)} \leq 1+\epsilon+\left(\frac{\epsilon}{\sqrt{g}}+\epsilon^2\right)\frac{\|\mathbf{X}\|_2^2}{n\gamma} \qquad \text{and} \qquad \frac{\mathsf{var}(\mathbf{w}^{\mathrm{h}}_{[g]})}{\mathsf{var}(\mathbf{w}^\star)} \leq 1+\epsilon$$

hold with high probability. That is, model averaging improves the bias without affecting the variance. The bound

$$\frac{\mathsf{bias}(\mathbf{w}^{\mathrm{h}}_{[g]})-\mathsf{bias}(\mathbf{w}^\star)}{\mathsf{bias}(\mathbf{w}^\star)} \leq \epsilon+\left(\frac{\epsilon}{\sqrt{g}}+\epsilon^2\right)\frac{\|\mathbf{X}\|_2^2}{n\gamma}$$

indicates that if $n\gamma$ is much smaller than $\|\mathbf{X}\|_2^2$ and $\epsilon \leq \frac{1}{\sqrt{g}}$, or equivalently, s is at least $\tilde{\mathcal{O}}(dg)$, then the ratio is proportional to $\frac{\epsilon}{\sqrt{g}}$.

To verify Theorem 11, we set γ very small—$\gamma = 10^{-12}$—and vary s and g. In Figure 6 we plot the ratio

$$\frac{\mathsf{bias}(\mathbf{w}^{\mathrm{h}}_{[1]})-\mathsf{bias}(\mathbf{w}^\star)}{\mathsf{bias}(\mathbf{w}^{\mathrm{h}}_{[g]})-\mathsf{bias}(\mathbf{w}^\star)}, \tag{12}$$

by fixing $\gamma = 10^{-12}$ and varying s and g. The theory indicates that for large sketch size $s = \tilde{\mathcal{O}}(dg^2)$, this ratio should grow nearly linearly in g. Figure 6 shows that only for large s and very small g, the growth is near linear in g; this verifies our theory.

When we similarly plot $\mathsf{var}\big(\mathbf{w}^{\mathrm{h}}_{[g]}\big)$ against g, we observe that $\mathsf{var}\big(\mathbf{w}^{\mathrm{h}}_{[g]}\big)$ remains nearly unaffected as g grows from 1 to 50. Since the curves of the variance against g are almost horizontal lines, we do not show this plot in the paper.

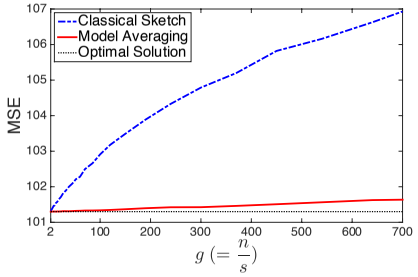


Figure 7: Prediction performance of classical sketch with and without model averaging on the Year Prediction data set. The x-axis is g, the number of data partitions, and the y-axis is the mean squared error (MSE) on the test set.

5. Model Averaging Experiments on Real-World Data

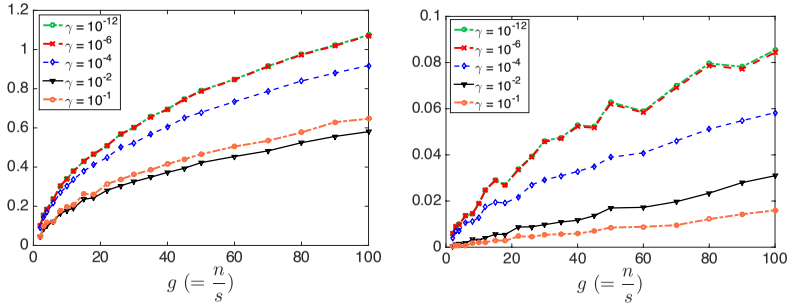
In Section 1 we mentioned that in the distributed setting where the feature-response pairs $(\mathbf{x}_1, \mathbf{y}_1), \cdots, (\mathbf{x}_n, \mathbf{y}_n) \in \mathbb{R}^{d \times m}$ are randomly and uniformly partitioned across g machines,[5] classical sketch with model averaging requires only one round of communication, and is therefore a communication-efficient algorithm that can be used to: (1) obtain an approximate solution of the MRR problem with risk comparable to a batch solution, and (2) obtain a low-precision solution of the MRR optimization problem that can be used as an initializer for more communication-intensive optimization algorithms. In this section, we demonstrate both applications.

We use the Million Song Year Prediction data set, which has $463,715$ training samples and $51,630$ test samples with 90 features and one response. We normalize the data by shifting the responses to have zero mean and scaling the range of each feature to $[-1, 1]$. We randomly partition the training data into g parts, which amounts to uniform row selection with sketch size $s = \frac{n}{g}$.

5.1 Prediction Error

We tested the prediction performance of sketched ridge regression by implementing classical sketch with model averaging in PySpark (Zaharia et al., 2010).[6] We ran our experiments using PySpark in local mode; the experiments proceeded in three steps: (1) use five-fold cross-validation to determine the regularization parameter γ; (2) learn the model $\mathbf{w}$ using the selected γ; and (3) use $\mathbf{w}$ to predict on the test set and record the mean squared errors (MSEs). These steps map cleanly onto the Map-Reduce programming model used by PySpark.

5. If the samples are i.i.d., then any deterministic partition is essentially a uniformly randomly distributed partition. Otherwise, we can invoke a Shuffle operation, which is supported by systems such as Apache Spark (Zaharia et al., 2010), to make the partitioning uniformly randomly distributed.
6. The code is available at https://github.com/wangshusen/SketchedRidgeRegression.git



(a) Classical sketch (b) Classical sketch with model averaging

Figure 8: Optimization performance of classical sketch with and without model averaging. The x-axis is g, the number of data partitions, and the y-axis is the ratio $\frac{\|\mathbf{w}-\mathbf{w}^\star\|_2}{\|\mathbf{w}^\star\|_2}$.

In Figure 7, we plot the test MSE against $g = \frac{n}{s}$. As g grows, the sketch size $s = \frac{n}{g}$ decreases, so the performance of classical sketch deteriorates. However classical sketch with model averaging always has test MSE comparable to the optimal solution.

5.2 Optimization Error

We mentioned earlier that classical sketch with or without model averaging can be used to initialize optimization algorithms for solving MRR problems. If $\mathbf{w}$ is initialized with zero-mean random variables or deterministically with zeros, then $\mathbb{E}\big[\|\mathbf{w}-\mathbf{w}^\star\|_2 / \|\mathbf{w}^\star\|_2\big] \geq 1$. Any $\mathbf{w}$ with the above ratio substantially smaller than 1 provides a better initialization. We implemented classical sketch with and without model averaging in Python and calculated the above ratio on the training set of the Year Prediction data set; to estimate the expectation, we repeated the procedure 100 times and report the average of the ratios.

In Figure 8, we plot the average of the ratio $\frac{\|\mathbf{w}-\mathbf{w}^\star\|_2}{\|\mathbf{w}^\star\|_2}$ against g for different settings of the regularization parameter γ. Clearly, classical sketch does not give a good initialization unless g is small (equivalently, the sketch size $s = \frac{n}{g}$ is large). In contrast, the averaged solution is always close to $\mathbf{w}^\star$.

6. Sketch of Proof

In this section, we outline the proofs of our main results. The complete details are provided in the appendix. Section 6.1 recaps several relevant properties of matrix sketching. Section 6.2 establishes certain properties of averages of sketches; these results are used to analyze the application of model averaging to the MRR problem. Sections 6.3 to 6.6 provide key structural results on sketched solutions to the MRR problem constructed with or without model averaging.

Our main results in Section 3 (Theorems 1, 2, 5, 6, 7, 8, 10, and 11) follow directly from the relevant properties of matrix sketching and the structural results for solutions to the

sketched MRR problem. Table 4 summarizes the dependency relationships among these theorems. For example, Theorem 1, which studies classical sketching from the optimization perspective, is one of our main theorems and is proven using Theorems 12 and 15.

Table 4: An overview of our results and their dependency relationships.

Main Theorems	Solution	Perspective	Prerequisites
Theorem 1	classical	optimization	Theorems 12 and 15
Theorem 2	Hessian	optimization	Theorems 12 and 16
Theorem 5	classical	statistical	Theorems 12, 13, 17, 18
Theorem 6	Hessian	statistical	Theorems 12 and 19
Theorem 7	classical, averaging	optimization	Theorems 14 and 20
Theorem 8	Hessian, averaging	optimization	Theorems 14 and 21
Theorem 10	classical, averaging	statistical	Theorems 14 and 22
Theorem 11	Hessian, averaging	statistical	Theorems 14 and 23

6.1 Properties of Matrix Sketching

Our analysis of the performance of solutions to the sketched MRR problem draws heavily on the three key properties defined in Assumption 1. Theorem 12 establishes that the six sketching methods considered in this paper indeed enjoy the three key properties under certain conditions. Finally, Theorem 13 establishes the lower bounds of $\|\mathbf{S}\|_2^2$ that are used to prove the lower bounds on the variance of sketched MRR solutions in Theorem 5.

Assumption 1 *Let $\eta, \epsilon \in (0,1)$ be fixed parameters. Let $\mathbf{B}$ be any fixed matrix of conformal shape, $\rho = \text{rank}(\mathbf{X})$, and $\mathbf{U} \in \mathbb{R}^{n\times\rho}$ be an orthonormal basis for the column span of $\mathbf{X}$. Let $\mathbf{S} \in \mathbb{R}^{n\times s}$ be a sketching matrix, where s depends on η and/or ϵ. Throughout this paper, we assume that $\mathbf{S}$ satisfies the following properties with a probability that depends on s:*

1.1 $\left\|\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U} - \mathbf{I}_\rho\right\|_2 \le \eta$ *(Subspace Embedding Property);*

1.2 $\left\|\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{B} - \mathbf{U}^T\mathbf{B}\right\|_F^2 \le \epsilon\|\mathbf{B}\|_F^2$ *(Matrix Multiplication Property);*

1.3 When $s < n$, $\|\mathbf{S}\|_2^2 \le \frac{\theta n}{s}$ for some constant θ (Bounded Spectral Norm Property).

The subspace embedding property requires that sketching preserves the inner products between the columns of a matrix with orthonormal columns. Equivalently, it ensures that the singular values of any sketched column-orthonormal matrix are all close to one. The subspace embedding property implies that, in particular, the squared norm of $\mathbf{Sx}$ is close to that of $\mathbf{x}$ for any n-dimensional vector in a fixed ρ-dimensional subspace. A dimension counting argument suggests that since $\mathbf{Sx}$ is an s-dimensional vector, its length must be scaled by a factor of $\sqrt{\frac{n}{s}}$ to ensure that this consequence of the subspace embedding property holds. The bounded spectral norm property requires that the spectral norm of $\mathbf{S}$ is not much larger than this rescaling factor of $\sqrt{\frac{n}{s}}$.

Remark 2 *The first two assumptions were identified in (Mahoney, 2011) and are the relevant structural conditions that allow strong results from the optimization perspective.*

Table 5: The two middle columns provide an upper bound on the sketch size s needed to satisfy the subspace embedding property and the matrix multiplication property, respectively, under the different sketching modalities considered; the right column lists the parameter θ with which the bounded spectral norm property holds. These properties hold with constant probability for the indicated values of s. Here τ is defined in (5) and reflects the quality of the approximation of the leverage scores of $\mathbf{U}$; μ is the row coherence of $\mathbf{U}$. For Gaussian projection and CountSketch, the small-o notation is a consequence of $s = o(n)$.

Sketching	Subspace Embedding	Matrix Multiplication	Spectral Norm
Leverage	$s = \mathcal{O}\left(\frac{\tau\rho}{\eta^2}\log\frac{\rho}{\delta_1}\right)$	$s = \mathcal{O}\left(\frac{\tau\rho}{\epsilon\delta_2}\right)$	$\theta = \infty$
Uniform	$s = \mathcal{O}\left(\frac{\mu\rho}{\eta^2}\log\frac{\rho}{\delta_1}\right)$	$s = \mathcal{O}\left(\frac{\mu\rho}{\epsilon\delta_2}\right)$	$\theta = 1$
Shrinked Leverage	$s = \mathcal{O}\left(\frac{\tau\rho}{\eta^2}\log\frac{\rho}{\delta_1}\right)$	$s = \mathcal{O}\left(\frac{\tau\rho}{\epsilon\delta_2}\right)$	$\theta = 2$
SRHT	$s = \mathcal{O}\left(\frac{\rho+\log n}{\eta^2}\log\frac{\rho}{\delta_1}\right)$	$s = \mathcal{O}\left(\frac{\rho+\log n}{\epsilon\delta_2}\right)$	$\theta = 1$
Gaussian Projection	$s = \mathcal{O}\left(\frac{\rho+\log(1/\delta_1)}{\eta^2}\right)$	$s = \mathcal{O}\left(\frac{\rho}{\epsilon\delta_2}\right)$	$\theta = 1 + o(1)$ w.h.p.
CountSketch	$s = \mathcal{O}\left(\frac{\rho^2}{\delta_1\eta^2}\right)$	$s = \mathcal{O}\left(\frac{\rho}{\epsilon\delta_2}\right)$	$\theta = 1 + o(1)$ w.h.p.

The third assumption is new, but Ma et al. (2015); Raskutti and Mahoney (2016) demonstrated that some sort of additional condition is necessary to obtain strong results from the statistical perspective.

Remark 3 *We note that $\mathbf{U}^T\mathbf{U} = \mathbf{I}_\rho$, and thus Assumption 1.1 can be expressed in the form of an approximate matrix multiplication bound (Drineas et al., 2006a). We call it the Subspace Embedding Property since, as first highlighted in Drineas et al. (2006b), this subspace embedding property is the key result necessary to obtain high-quality sketching algorithms for regression and related problems.*

Theorem 12 shows that the six sketching methods satisfy the three properties when s is sufficiently large. In particular, Theorem 12 shows that for all the sketching methods except leverage score sampling,[7] $\|\mathbf{S}\|_2^2$ has nontrivial upper bound. This is why Theorems 5 and 10 do not apply to leverage score sampling. This fact can also be viewed as a motivation to use shrinked leverage score sampling. We prove Theorem 12 in Appendix A.

Theorem 12 *Fix failure probability δ and error parameters η and ϵ; set the sketch size s as Table 5. Assumption 1.1 is satisfied with probability at least $1 - \delta_1$. Assumption 1.2 is satisfied with probability at least $1 - \delta_2$. Assumption 1.3 is satisfied either surely or with high probability (w.h.p.); the parameter θ is indicated in Table 5.*

Theorem 13 establishes lower bounds on $\|\mathbf{S}\|_2^2$, and will be applied to prove the lower bound on the variance of the classical sketch. From Table 6 we see that the lower bound for

7. If one leverage score approaches zero, then the corresponding sampling probability p_i goes to zero. By the definition of $\mathbf{S}$, the scale factor $\frac{1}{\sqrt{sp_i}}$ goes to infinity, which makes $\|\mathbf{S}\|_2^2$ unbounded. The shinked leverage score sampling avoids this problem and is thus a better choice than the leverage score sampling.

(shrinked) leverage score sampling is not interesting, because μ can be very large. This is why Theorem 5 does not provide a lower bound for shrinked leverage score sampling. We prove Theorem 13 in Appendix A.

Table 6: Lower bounds on ϑ for the sketching modalities (ϑ is defined in Theorem 13). The shrinked leverage score sampling is performed using the row leverage scores of a matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, and μ is the row coherence of $\mathbf{X}$.

Uniform	$\vartheta = 1$
Leverage	$\vartheta \geq \frac{1}{\mu}$
Shrinked Leverage	$\vartheta \geq \frac{2}{1+\mu}$
SRHT	$\vartheta = 1$
Gaussian Projection	$\vartheta \geq 1 - o(1)$ w.h.p.
CountSketch	$\vartheta \geq 1 - o(1)$ w.h.p.

Theorem 13 (Semidefinite Lower Bound on the Sketching Matrix) *When $s < n$, $\mathbf{S}^T\mathbf{S} \succeq \frac{\vartheta n}{s}\mathbf{I}_s$ holds either surely or with high probability (w.h.p.), where Table 6 provides the applicable ϑ for each sketching method.*

Remark 4 *Let $p_1, \cdots, p_n$ be an arbitrary set of sampling probabilities. By the definition of the associated sampling matrix $\mathbf{S} \in \mathbb{R}^{n\times s}$, the non-zero entries of $\mathbf{S}$ can be any of $\frac{1}{\sqrt{sp_i}}$, for $i \in [n]$.*

For leverage score sampling, since the smallest sampling probability can be zero or close, and the largest sampling probability can be close to one, $\|\mathbf{S}\|_2^2$ has no nontrivial upper or lower bound.[8] *It is because $\min_i p_i$ can be close to zero and $\max_i p_i$ can be large (close to one).*

For shrinked leverage score sampling, because $\min_i p_i$ is at least $\frac{1}{2n}$, $\|\mathbf{S}\|_2^2$ has a nontrivial upper bound; but as in the case of leverage score sampling, since $\max_i p_i$ can be large, there is no nontrivial lower bound on $\|\mathbf{S}\|_2^2$.

6.2 Matrix Sketching with Averaging

Assumptions 1.1 and 1.2 imply that sketching can be used to approximate certain matrix products, but what happens if we independently draw g sketches, use them to approximate the same matrix product, and then average the g results? Intuitively, averaging should lower the variance of the approximation without affecting its bias, and thus provide a better approximation of the true product.

To justify this intuition formally, let $\mathbf{S}_1, \cdots, \mathbf{S}_g \in \mathbb{R}^{n\times s}$ be sketching matrices and $\mathbf{A}$ and $\mathbf{B}$ be fixed conformal matrices. Then evidently

$$\frac{1}{g}\sum_{i=1}^{g} \mathbf{A}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{B} \;=\; \mathbf{A}^T\mathbf{S}\mathbf{S}^T\mathbf{B},$$

8. In our application, nontrivial bound means $\|\mathbf{S}\|_2^2$ is of order $\frac{n}{s}$.

where $\mathbf{S} = \frac{1}{\sqrt{g}}[\mathbf{S}_1, \cdots, \mathbf{S}_g] \in \mathbb{R}^{n\times gs}$ can be thought of as a sketching matrix formed by concatenating the g smaller sketching matrices. If $\mathbf{S}_1, \cdots, \mathbf{S}_g$ are all instance of column selection, SRHT, or Gaussian projection sketching matrices, then $\mathbf{S}$ is a larger instance of the same type of sketching matrix.[9]

To analyze the effect of model averaging on the solution to the sketched MRR problem, we make the following assumptions on the concatenated sketch matrix. Assumption 2.1 is the subspace embedding property, Assumption 2.2 is the matrix multiplication property, and Assumption 2.3 is the bounded spectral norm property.

Assumption 2 *Let $\eta, \epsilon \in (0,1)$ be fixed parameters. Let $\mathbf{B}$ be any fixed matrix of proper size, $\rho = \mathrm{rank}(\mathbf{X})$, and $\mathbf{U} \in \mathbb{R}^{n\times\rho}$ be an orthonormal basis for the column span of $\mathbf{X}$. Let $\mathbf{S}_1, \cdots, \mathbf{S}_g \in \mathbb{R}^{n\times s}$ be sketching matrices and $\mathbf{S} = \frac{1}{\sqrt{g}}[\mathbf{S}_1, \cdots, \mathbf{S}_g] \in \mathbb{R}^{n\times gs}$; here s depends on η and/or ϵ. Throughout this paper we assume that $\mathbf{S}$ and the $\mathbf{S}_i$ satisfy the following properties with a probability that depends on g and s:*

2.1 $\left\|\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{U} - \mathbf{I}_\rho\right\|_2 \leq \eta$ *for all* $i \in [g]$ *and* $\left\|\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U} - \mathbf{I}_\rho\right\|_2 \leq \frac{\eta}{\sqrt{g}}$;

2.2 $\left(\frac{1}{g}\sum_{i=1}^{g}\left\|\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{B} - \mathbf{U}^T\mathbf{B}\right\|_F\right)^2 \leq \epsilon\|\mathbf{B}\|_F^2$ *and* $\left\|\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{B} - \mathbf{U}^T\mathbf{B}\right\|_F^2 \leq \frac{\epsilon}{g}\|\mathbf{B}\|_F^2$;

2.3 For some constant θ, $\|\mathbf{S}_i\|_2^2 \leq \frac{\theta n}{s}$ *for all* $i \in [g]$, *and* $\|\mathbf{S}\|_2^2 \leq \frac{\theta n}{gs}$ *for* $gs < n$.

Except in the case of leverage score sampling, when gs is comparable to or larger than n, $\|\mathbf{S}\|_2^2 = \Theta(1)$.

Theorem 14 establishes that random column selection, SRHT, and Gaussian projection matrices satisfy Assumptions 2.1, 2.2, and 2.3. We prove Theorem 14 in Appendix A.

Theorem 14 *Let $\mathbf{S}_1, \cdots, \mathbf{S}_g \in \mathbb{R}^{n\times s}$ be independent and identically distributed random sketching matrices that are either column selection, SRHT, or Gaussian projection matrices. Fix a failure probability δ and error parameters η and ϵ, then set the sketch size s as Table 5.*

Assumption 2.1 holds with probability at least $1 - (g+1)\delta_1$. Assumption 2.2 holds with probability at least $1 - 2\delta_2$. Assumption 2.3 is satisfied either surely or with high probability, with the parameter θ specified in Table 5.

In Theorem 12, Assumption 1.1 fails with probability at most δ_1. In contrast, in Theorem 14, the counterpart assumption fails with probability at most $(g+1)\delta_1$. However, this makes little difference in practice, because the dependence of s on δ_1 is logarithmic, so δ_1 can be set very small (recall Table 5) without increasing s significantly.

Remark 5 *We do not know whether CountSketch enjoys the properties in Assumption 2. There are two difficulties in establishing this using the same route as is employed in our proof of Theorem 12 for other sketching methods. First, the concatenation of multiple CountSketch matrices is not a CountSketch matrix. Second, the probability that a CountSketch matrix does not have the subspace embedding property is constant, rather than exponentially small.*

9. CountSketch sketching matrices does not have this property. If $\mathbf{S}_i \in \mathbb{R}^{n\times s}$ is a CountSketch matrix, then it has only one non-zero entry in each row. In contrast, $\mathbf{S} \in \mathbb{R}^{n\times gs}$ has g non-zero entries in each row.

6.3 Sketched MRR: Optimization Perspective

The randomness in the performance of the classical and Hessian sketch is entirely due to the choice of random sketching matrix. We now assume that the randomly sampled sketching matrices are "nice" in that they satisfy the assumptions just introduced, and state deterministic results on the optimization performance of the classical and Hessian sketches.

Theorem 15 holds under the subspace embedding property and the matrix multiplication property (Assumptions 1.1 and 1.2), and quantifies the suboptimality of the classical sketch. We prove this result in Appendix B.

Theorem 15 (Classical Sketch) *Let Assumptions 1.1 and 1.2 hold for the sketching matrix* $\mathbf{S} \in \mathbb{R}^{n\times s}$. *Let* η *and* ϵ *be defined in Assumption 1, and let* $\alpha = \frac{2\max\{\epsilon,\eta^2\}}{1-\eta}$ *and* $\beta = \frac{\|\mathbf{X}\|_2^2}{\|\mathbf{X}\|_2^2+n\gamma}$, *then*

$$f(\mathbf{W}^c) - f(\mathbf{W}^\star) \;\; \leq \;\; \alpha\beta f(\mathbf{W}^\star).$$

Theorem 16 holds under the subspace embedding property (Assumption 1.1), and quantifies the suboptimality of the Hessian sketch. We prove this result in Appendix B.

Theorem 16 (Hessian Sketch) *Let Assumption 1.1 hold for the sketching matrix* $\mathbf{S} \in \mathbb{R}^{n\times s}$. *Let* η *be defined in Assumption 1 and* $\beta = \frac{\|\mathbf{X}\|_2^2}{\|\mathbf{X}\|_2^2+n\gamma}$, *then*

$$f(\mathbf{W}^h) - f(\mathbf{W}^\star) \;\; \leq \;\; \frac{\eta^2\beta^2}{(1-\eta)^2}\Big(\frac{\|\mathbf{Y}\|_F^2}{n} - f(\mathbf{W}^\star)\Big).$$

6.4 Sketched MRR: Statistical Perspective

Similarly, we assume that the randomly sampled sketching matrices are nice, and state deterministic results on the bias and variance of the classical and Hessian sketches.

Theorem 17 holds under the subspace embedding property (Assumption 1.1) and the bounded spectral norm property (Assumption 1.3), and bounds the bias and variance of the classical sketch. Specifically, it shows that the bias of the classical sketch is close to that of the optimal solution, but that the variance may be much larger. We prove this result in Appendix C.

Theorem 17 (Classical Sketch) *Let* η *and* θ *be defined in Assumption 1. Under Assumption 1.1, it holds that*

$$\tfrac{1}{1+\eta} \;\leq\; \tfrac{\mathsf{bias}(\mathbf{W}^c)}{\mathsf{bias}(\mathbf{W}^\star)} \;\leq\; \tfrac{1}{1-\eta}.$$

Further assume $s \leq n$; *under Assumptions 1.1 and 1.3, it holds that*

$$\tfrac{\mathsf{var}(\mathbf{W}^c)}{\mathsf{var}(\mathbf{W}^\star)} \;\leq\; \tfrac{(1+\eta)}{(1-\eta)^2}\tfrac{\theta n}{s}.$$

Theorem 18 establishes a lower bound on the variance of the classical sketch. We prove this result in Appendix C.

Theorem 18 (Lower Bound on the Variance) *Under Assumption 1.1 and the additional assumption that* $\mathbf{S}^T\mathbf{S} \succeq \frac{\vartheta n}{s}\mathbf{I}_s$, *it holds that*

$$\tfrac{\mathsf{var}(\mathbf{W}^c)}{\mathsf{var}(\mathbf{W}^\star)} \;\; \geq \;\; \tfrac{1-\eta}{(1+\eta)^2}\tfrac{\vartheta n}{s}.$$

Theorem 19 holds under the subspace embedding property (Assumption 1.1), and quantifies the bias and variance of the Hessian sketch. We prove this result in Appendix C.

Theorem 19 (Hessian Sketch) *Let* η *be defined in Assumption 1, take* $\rho = \mathrm{rank}(\mathbf{X})$, *and let* $\sigma_1 \geq \cdots \geq \sigma_\rho$ *be the singular values of* $\mathbf{X}$. *Under Assumption 1.1, it holds that*

$$\tfrac{\mathsf{bias}(\mathbf{W}^h)}{\mathsf{bias}(\mathbf{W}^\star)} \;\leq\; \tfrac{1}{1-\eta}\Big(1 + \tfrac{\eta\sigma_1^2}{n\gamma}\Big),$$
$$\tfrac{1}{1+\eta} \;\leq\; \tfrac{\mathsf{var}(\mathbf{W}^h)}{\mathsf{var}(\mathbf{W}^\star)} \;\leq\; \tfrac{1}{1-\eta}.$$

Further assume that $\sigma_\rho^2 \geq \frac{n\gamma}{\eta}$. *Then*

$$\tfrac{\mathsf{bias}(\mathbf{W}^h)}{\mathsf{bias}(\mathbf{W}^\star)} \;\geq\; \tfrac{1}{1+\eta}\Big(\tfrac{\eta\sigma_\rho^2}{n\gamma} - 1\Big).$$

6.5 Model Averaging: Optimization Perspective

Theorem 20 holds under the subspace embedding property (Assumption 2.1) and the matrix multiplication property (Assumption 2.2). We prove this result in Appendix D.

Theorem 20 (Classical Sketch with Model Averaging) *Let* η *and* ϵ *be defined in Assumption 2, and let* $\alpha = 2\big(\frac{1}{\sqrt{g}} + 2\beta\eta\big)^2\max\{\epsilon,\eta^2\}$ *and* $\beta = \frac{\|\mathbf{X}\|_2^2}{\|\mathbf{X}\|_2^2+n\gamma} \leq 1$. *Under Assumption 2.1 and 2.2, we have that*

$$f(\mathbf{W}^c) - f(\mathbf{W}^\star) \;\; \leq \;\; \alpha\beta f(\mathbf{W}^\star).$$

Theorem 21 holds under the subspace embedding property (Assumption 2.1), and is proven in Appendix D.

Theorem 21 (Hessian Sketch with Model Averaging) *Let* η *be defined in Assumption 2, and let* $\alpha = \big(\frac{\eta}{\sqrt{g}} + \frac{\eta^2}{1-\eta}\big)$ *and* $\beta = \frac{\|\mathbf{X}\|_2^2}{\|\mathbf{X}\|_2^2+n\gamma} \leq 1$. *Under Assumption 2.1, we have that*

$$f(\mathbf{W}^h) - f(\mathbf{W}^\star) \;\; \leq \;\; \alpha^2\beta^2\Big(\tfrac{1}{n}\|\mathbf{Y}\|_F^2 - f(\mathbf{W}^\star)\Big).$$

6.6 Model Averaging: Statistical Perspective

Theorem 22 requires the subspace embedding property (Assumption 2.1). In addition, to bound the variance, the spectral norms of $\mathbf{S}_1, \cdots, \mathbf{S}_g$ and $\mathbf{S} = \frac{1}{\sqrt{g}}[\mathbf{S}_1, \cdots, \mathbf{S}_g]$ must be bounded (Assumption 2.3). This result shows that model averaging decreases the variance of the classical sketch without increasing its bias. We prove this result in Appendix E.

Theorem 22 (Classical Sketch with Model Averaging) *Under Assumption 2.1, it holds that*

$$\frac{\mathsf{bias}(\mathbf{W}^c)}{\mathsf{bias}(\mathbf{W}^\star)} \;\leq\; \frac{1}{1-\eta}.$$

Under Assumptions 2.1 and 2.3, it holds that

$$\frac{\mathsf{var}(\mathbf{W}^c)}{\mathsf{var}(\mathbf{W}^\star)} \;\leq\; \frac{\theta n}{s}\left(\frac{\sqrt{1+\eta/\sqrt{g}}}{\sqrt{h}} + \frac{\eta\sqrt{1+\eta}}{1-\eta}\right)^2.$$

Here η and θ are defined in Assumption 2 and $h = \min\{g, \frac{n}{s}(1-o(1))\}$,

Theorem 23 requires the subspace embedding property (Assumption 2.1), and shows that model averaging decreases the bias of the Hessian sketch without increasing its variance. We prove this result in Appendix E.

Theorem 23 (Hessian Sketch with Model Averaging) *Under Assumption 2.1, it holds that:*

$$\frac{\mathsf{bias}(\mathbf{W}^h)}{\mathsf{bias}(\mathbf{W}^\star)} \;\leq\; \frac{1}{1-\eta} + \Big(\frac{\eta}{\sqrt{g}} + \frac{\eta^2}{1-\eta}\Big)\frac{\|\mathbf{X}\|_2^2}{n\gamma},$$
$$\frac{\mathsf{var}(\mathbf{W}^h)}{\mathsf{var}(\mathbf{W}^\star)} \;\leq\; \frac{1}{1-\eta}.$$

Here η is defined in Assumption 2.

7. Conclusions

We studied sketched matrix ridge regression (MRR) from the optimization and statistical perspectives. Using classical sketch, by taking a large enough sketch, one can obtain an ϵ-accurate approximate solution. Counterintuitively and in contrast to classical sketch, the relative error of Hessian sketch increases as the responses $\mathbf{Y}$ are better approximated by linear combinations of the columns of $\mathbf{X}$. Both classical and Hessian sketches can have statistical risks that are worse than the risk of the optimal solution by an order of magnitude.

We proposed the use of model averaging to attain better optimization and statistical properties. We have shown that model averaging leads to substantial improvements in the theoretical error bounds, suggesting applications in distributed optimization and machine learning. We also empirically verified its practical benefits.

Our fixed-design statistical analysis has limitations. We have shown that the classical sketch and Hessian sketch can significantly increase the in-sample statistical risk, which implies large training error, and that model averaging can alleviate such problems. However, our statistical results are not directly applicable to an unseen test sample. We conjecture that the generalization error can be bounded by following the random design analysis of Hsu et al. (2014), which is left as future work.

Acknowledgments

We thank the anonymous reviewers and Serena Ng for their helpful suggestions. We thank the Army Research Office and the Defense Advanced Research Projects Agency for partial support of this work.

Appendix A. Properties of Matrix Sketching: Proofs

In Section A.1 we prove Theorem 12. In Section A.2, we prove Theorem 13. In Section A.3 we prove Theorem 14.

A.1 Proof of Theorem 12

We prove that the six sketching methods considered in this paper satisfy the three key properties. In Section A.1.1 we show the six sketching methods satisfy Assumptions 1.1 and 1.2. In section A.1.2 we show the six sketching methods satisfy Assumption 1.3.

A.1.1 Proof of Assumptions 1.1 and 1.2

For uniform sampling, leverage score sampling, Gaussian projection, SRHT, and CountSketch, the subspace embedding property and matrix multiplication property have been established by the previous works (Drineas et al., 2008, 2011; Meng and Mahoney, 2013; Nelson and Nguyên, 2013; Tropp, 2011; Woodruff, 2014). See also (Wang et al., 2016b) for a summary.

In the following we prove only that **shrinked leverage score sampling** satisfies assumptions 1.1 and 1.2. We cite the following lemma from (Wang et al., 2016a); this lemma was first established in the works (Drineas et al., 2008; Gittens, 2011; Woodruff, 2014).

Lemma 24 (Wang et al. (2016a)) *Let $\mathbf{U} \in \mathbb{R}^{n\times\rho}$ be a fixed matrix with orthonormal columns. Let the column selection matrix $\mathbf{S} \in \mathbb{R}^{n\times s}$ sample s columns according to probabilities $p_1, p_2, \cdots, p_n$. Assume $\alpha \geq \rho$ and*

$$\max_{i\in[n]} \frac{\|\mathbf{u}_{i:}\|_2^2}{p_i} \;\leq\; \alpha.$$

When $s \geq \alpha\frac{6+2\eta}{3\eta^2}\log(\rho/\delta_1)$, it holds that

$$\mathbb{P}\Big\{\big\|\mathbf{I}_\rho - \mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}\big\|_2 \;\geq\; \eta\Big\} \;\leq\; \delta_1.$$

When $s \geq \frac{\alpha}{\epsilon\delta_2}$, it holds that

$$\mathbb{E}\big\|\mathbf{U}\mathbf{B} - \mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{B}\big\|_F^2 \;\leq\; \delta_2\epsilon\|\mathbf{B}\|_F^2;$$

as a consequence of Markov's inequality, it holds that

$$\mathbb{P}\Big\{\big\|\mathbf{U}\mathbf{B} - \mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{B}\big\|_F^2 \;\geq\; \epsilon\|\mathbf{B}\|_F^2\Big\} \;\leq\; \delta_2.$$

Here the expectation and probability are with respect to the randomness in $\mathbf{S}$.

Now we apply the above lemma to analyze **shrinked leverage score sampling**. Given the approximate shrinked leverage scores defined in (5), the sampling probabilities satisfy

$$p_i \;=\; \tfrac{1}{2}\big(\tfrac{1}{n} + \tfrac{\tilde{l}_i}{\sum_{q=1}^n \tilde{l}_q}\big) \;\geq\; \tfrac{\|\mathbf{u}_{i:}\|_2^2}{2\tau\rho}.$$

Here $\tilde{l}_i$ and τ are defined in (5). Thus for all $i \in [n]$, $\frac{\|\mathbf{u}_{i:}\|_2^2}{p_i} \leq 2\tau\rho$. We can then apply Lemma 24 to show that Assumption 1.1 holds with probability at least $1-\delta_1$ when $s \geq 2\tau\rho\frac{6+2\eta}{3\eta^2}\log\frac{\rho}{\delta_1}$ and that Assumption 1.2 holds with probability at least $1-\delta_2$ when $s \geq \frac{2\tau\rho}{\epsilon\delta_2}$.

A.1.2 Proof of Assumption 1.3

For uniform sampling (without replacement) and SRHT, when $s < n$, it is easy to show that $\mathbf{S}^T\mathbf{S} = \frac{n}{s}\mathbf{I}_s$, and thus $\|\mathbf{S}\|_2^2 = \frac{n}{s}$. Let $\{p_i^{\mathrm{s}}\}$ and $\{p_i^{\mathrm{u}}\}$ be the sampling probabilites of shrinked leverage score sampling and uniform sampling, respectively. Obviously $p_i^{\mathrm{s}} \geq \frac{1}{2}p_i^{\mathrm{u}}$. Thus for shrinked leverage score sampling, $\|\mathbf{S}\|_2^2 \leq \frac{2n}{s}$.

The greatest singular value of a standard Gaussian matrix $\mathbf{G} \in \mathbb{R}^{n\times s}$ is at most $\sqrt{n}+\sqrt{s}+t$ with probability at least $1-2e^{-t^2/2}$ (Vershynin, 2012). Thus a Gaussian projection matrix $\mathbf{S}$ satisfies

$$\|\mathbf{S}\|_2^2 \;=\; \frac{1}{s}\|\mathbf{G}\|_2^2 \;\leq\; \frac{(\sqrt{n}+\sqrt{s}+t)^2}{s}$$

with probability at least $1-2e^{-t^2/2}$.

If $\mathbf{S}$ is the CountSketch matrix, then each row of $\mathbf{S}$ has exactly one nonzero entry, either 1 or -1. Because the columns of $\mathbf{S}$ are orthogonal to each other, it holds that

$$\|\mathbf{S}\|_2^2 \;=\; \max_{i\in[s]}\|\mathbf{s}_{:i}\|_2^2 \;=\; \max_{i\in[s]}\mathrm{nnz}(\mathbf{s}_{:i}).$$

The problem of bounding $\mathrm{nnz}(\mathbf{s}_{:i})$ is equivalent to assigning n balls into s bins uniformly at random and bounding the number of balls in the bins. Patrascu and Thorup (2012) showed that for $s \ll n$, the maximal number of balls in any bin is at most $n/s + \mathcal{O}\big(\sqrt{n/s}\log^c n\big)$ with probability at least $1-\frac{1}{n}$, where $c = \mathcal{O}(1)$. Thus

$$\|\mathbf{S}\|_2^2 \;=\; \max_{i\in[s]}\mathrm{nnz}(\mathbf{s}_{:i}) \;\leq\; \frac{n}{s} + \mathcal{O}\bigg(\frac{\sqrt{n}\log^c n}{\sqrt{s}}\bigg) \;=\; \frac{n}{s}\big(1+o(1)\big)$$

holds with probability at least $1-\frac{1}{n}$.

A.2 Proof of Theorem 13

For uniform sampling (without replacement) and SRHT, it holds that $\mathbf{S}^T\mathbf{S} = \frac{n}{s}\mathbf{I}_s$.

For non-uniform sampling with probabilities $p_1, \cdots, p_n$, (with $\sum_i p_i = 1$), let $p_{\max} = \max_i p_i$. The smallest entry in $\mathbf{S}$ is $\frac{1}{\sqrt{sp_{\max}}}$, and thus $\mathbf{S}^T\mathbf{S} \succeq \frac{1}{sp_{\max}}\mathbf{I}_s$. For leverage score sampling, $p_{\max} = \frac{\mu}{n}$. For shrinked leverage score sampling, $p_{\max} = \frac{1+\mu}{2n}$. The lower bound on $\|\mathbf{S}\|_2^2$ is thus established.

The smallest singular value of any $n \times s$ standard Gaussian matrix $\mathbf{G}$ is at least $\sqrt{n}-\sqrt{s}-t$ with probability at least $1-2e^{-t^2/2}$ (Vershynin, 2012). Thus if $\mathbf{S} = \frac{1}{\sqrt{s}}\mathbf{G}$ is the

Gaussian projection matrix, the smallest eigenvalue of $\mathbf{S}^T\mathbf{S}$ is $(1-o(1))\frac{n}{s}$ with probability very close to one.

If $\mathbf{S}$ is the CountSketch matrix, then each row of $\mathbf{S}$ has exactly one nonzero entry, either 1 or -1. Because the columns of $\mathbf{S}$ are orthogonal to each other, it holds that

$$\sigma_{\min}^2(\mathbf{S}) \;=\; \min_{i\in[s]}\|\mathbf{s}_{:i}\|_2^2 \;=\; \min_{i\in[s]}\mathrm{nnz}(\mathbf{s}_{:i}).$$

The problem of bounding $\mathrm{nnz}(\mathbf{s}_{:i})$ is equivalent to assigning n balls into s bins uniformly at random and bounding the number of balls in the bins. Standard concentration arguments imply that each bin has at least $\frac{n}{s}(1-o(1))$ balls w.h.p., and hence $\sigma_{\min}^2(\mathbf{S}) \geq \frac{n}{s}(1-o(1))$ w.h.p.

A.3 Proof of Theorem 14

Assumption 2.1. By Theorem 12 and the union bound, we have that $\big\|\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T - \mathbf{I}_\rho\big\|_2 \leq \eta$ hold simultaneously for all $i \in [g]$ with probability at least $1-g\delta_1$. Because $\mathbf{S} \in \mathbb{R}^{n\times gs}$ is the same type of sketching matrix, it follows from Theorem 12 that $\big\|\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U} - \mathbf{I}_\rho\big\|_2 \leq \frac{\eta}{\sqrt{g}}$ holds with probability at least $1-\delta_1$.

Assumption 2.2. By the same proof of Theorem 12, we can easily show that

$$\mathbb{E}\big\|\mathbf{U}^T\mathbf{B} - \mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{B}\big\|_F^2 \;\leq\; \delta_2\epsilon\,\|\mathbf{B}\|_F^2,$$

where $\mathbf{B}$ is any fixed matrix and the expectation is taken w.r.t. $\mathbf{S}$. It follows from Jensen's inequality that

$$\Big(\mathbb{E}\big\|\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{B} - \mathbf{U}^T\mathbf{B}\big\|_F\Big)^2 \;\leq\; \mathbb{E}\big\|\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{B} - \mathbf{U}^T\mathbf{B}\big\|_F^2 \;\leq\; \delta_2\epsilon\big\|\mathbf{B}\big\|_F^2.$$

It follows that

$$\frac{1}{g}\sum_{i=1}^g \mathbb{E}\big\|\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{B} - \mathbf{U}^T\mathbf{B}\big\|_F \;\leq\; \sqrt{\delta_2\epsilon}\big\|\mathbf{B}\big\|_F,$$

and thus

$$\Big(\frac{1}{g}\sum_{i=1}^g \mathbb{E}\big\|\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{B} - \mathbf{U}^T\mathbf{B}\big\|_F\Big)^2 \;\leq\; \delta_2\epsilon\big\|\mathbf{B}\big\|_F^2.$$

It follows from Markov's bound that

$$\mathbb{P}\bigg\{\Big(\frac{1}{g}\sum_{i=1}^g \big\|\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{B} - \mathbf{U}^T\mathbf{B}\big\|_F\Big)^2 \;\leq\; \epsilon\big\|\mathbf{B}\big\|_F^2\bigg\} \;\geq\; 1-\delta_2.$$

Because $\mathbf{S} \in \mathbb{R}^{n\times gs}$ is the same type of sketching matrix, it follows from Theorem 12 that $\big\|\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{B} - \mathbf{U}^T\mathbf{B}\big\|_F^2 \leq \frac{\epsilon}{g}\|\mathbf{B}\|_F^2$ holds with probability at least $1-\delta_2$.

Assumption 2.3. Theorem 12 shows that $\|\mathbf{S}_i\|_2^2$ can be bounded either surely or w.h.p. (assuming n is large enough). Because $g \ll n$, $\|\mathbf{S}_i\|_2^2$ can be bounded simultaneously for all $i \in [g]$ either surely or w.h.p.

Suppose $sg < n$. Because $\mathbf{S} \in \mathbb{R}^{n \times gs}$ is the same type of sketching matrix, it follows from Theorem 12 that $\|\mathbf{S}\|_2^2 \leq \frac{\theta n}{gs}$ holds either surely or w.h.p.

Suppose $sg \geq n$. It is not hard to show that uniform sampling, shrinked leverage score sampling, and SRHT satisfy $\|\mathbf{S}\|_2 = \Theta(1)$ w.h.p. Previously we have shown that a random Gaussian projection matrix $\mathbf{S} \in \mathbb{R}^{n \times sg}$ satisfies

$$\|\mathbf{S}\|_2^2 \;\leq\; \big(1 + o(1)\big)\, \frac{(\sqrt{n} + \sqrt{gs})^2}{gs}$$

w.h.p. Hence for $sg \geq n$, $\|\mathbf{S}\|_2^2 \leq 4 + o(1)$ w.h.p.

Appendix B. Sketched MRR from the Optimization Perspective: Proofs

In Section B.1 we establish a key lemma. In Section B.2 we prove Theorem 15. In Section B.3 we prove Theorem 16.

B.1 Key Lemma

Recall that the objective function of the matrix ridge regression (MRR) problem is

$$f(\mathbf{W}) \;\triangleq\; \frac{1}{n}\big\|\mathbf{X}\mathbf{W} - \mathbf{Y}\big\|_F^2 + \gamma \|\mathbf{W}\|_F^2.$$

The optimal solution is $\mathbf{W}^\star = \operatorname{argmin}_{\mathbf{W}} f(\mathbf{W})$. The following is the key lemma for understanding the difference between the objective value at $\mathbf{W}^\star$ and any arbitrary $\mathbf{W}$.

Lemma 25 *For any matrix $\mathbf{W}$ and any nonsingular matrix $\mathbf{M}$ of proper size, it holds that*

$$\begin{aligned}
f(\mathbf{W}) &= \frac{1}{n}\mathrm{tr}\Big[\mathbf{Y}^T\mathbf{Y} - (2\mathbf{W}^\star - \mathbf{W})^T(\mathbf{X}^T\mathbf{X} + n\gamma\mathbf{I}_n)\mathbf{W}\Big],\\
f(\mathbf{W}^\star) &= \frac{1}{n}\Big[\big\|\mathbf{Y}^\perp\big\|_F^2 + n\gamma\big\|\big(\mathbf{\Sigma}^2 + n\gamma\mathbf{I}_\rho\big)^{-1/2}\mathbf{U}^T\mathbf{Y}\big\|_F^2\Big],\\
f(\mathbf{W}) - f(\mathbf{W}^\star) &= \frac{1}{n}\Big\|(\mathbf{X}^T\mathbf{X} + n\gamma\mathbf{I}_d)^{1/2}(\mathbf{W} - \mathbf{W}^\star)\Big\|_F^2,\\
\Big\|\mathbf{M}^{-1}(\mathbf{W} - \mathbf{W}^\star)\Big\|_F^2 &\leq \sigma_{\min}^{-2}\Big[(\mathbf{X}^T\mathbf{X} + n\gamma\mathbf{I}_d)^{1/2}\mathbf{M}\Big]\,\Big\|(\mathbf{X}^T\mathbf{X} + n\gamma\mathbf{I}_d)^{1/2}(\mathbf{W} - \mathbf{W}^\star)\Big\|_F^2.
\end{aligned}$$

Here $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is the SVD and $\mathbf{Y}^\perp = \mathbf{Y} - \mathbf{X}\mathbf{X}^\dagger\mathbf{Y}$.

Proof Let $\mathbf{U}$ be the left singular vectors of $\mathbf{X}$. The objective value $f(\mathbf{W})$ can be written as

$$\begin{aligned}
f(\mathbf{W}) &= \frac{1}{n}\big\|\mathbf{X}\mathbf{W} - \mathbf{Y}\big\|_F^2 + \gamma\big\|\mathbf{W}\big\|_F^2\\
&= \frac{1}{n}\mathrm{tr}\Big[\mathbf{Y}^T\mathbf{Y} - (2\mathbf{W}^\star - \mathbf{W})^T(\mathbf{X}^T\mathbf{X} + n\gamma\mathbf{I}_n)\mathbf{W}\Big],
\end{aligned}$$

so

$$\begin{aligned}
f(\mathbf{W}^\star) &= \frac{1}{n}\mathrm{tr}\Big[\mathbf{Y}^T\Big(\mathbf{I}_n - \mathbf{X}(\mathbf{X}^T\mathbf{X} + n\gamma\mathbf{I}_d)^{-1}\mathbf{X}^T\Big)\mathbf{Y}\Big]\\
&= \frac{1}{n}\mathrm{tr}\Big[\mathbf{Y}^T\Big(\mathbf{I}_n - \mathbf{U}(\mathbf{I}_\rho + n\gamma\mathbf{\Sigma}^{-2})^{-1}\mathbf{U}^T\Big)\mathbf{Y}\Big]\\
&= \frac{1}{n}\mathrm{tr}\Big[\mathbf{Y}^T\mathbf{Y} - \mathbf{Y}^T\mathbf{U}\mathbf{U}^T\mathbf{Y} + \mathbf{Y}^T\mathbf{U}\mathbf{U}^T\mathbf{Y} - \mathbf{Y}^T\mathbf{U}(\mathbf{I}_\rho + n\gamma\mathbf{\Sigma}^{-2})^{-1}\mathbf{U}^T\mathbf{Y}\Big]\\
&= \frac{1}{n}\Big\{\mathrm{tr}\Big[\mathbf{Y}^T(\mathbf{I}_n - \mathbf{U}\mathbf{U}^T)\mathbf{Y}\Big] + n\gamma\cdot\mathrm{tr}\Big[\mathbf{Y}^T\mathbf{U}\big(\mathbf{\Sigma}^2 + n\gamma\mathbf{I}_\rho\big)^{-1}\mathbf{U}^T\mathbf{Y}\Big]\Big\}\\
&= \frac{1}{n}\Big[\big\|\mathbf{Y}^\perp\big\|_F^2 + n\gamma\big\|\big(\mathbf{\Sigma}^2 + n\gamma\mathbf{I}_\rho\big)^{-1/2}\mathbf{U}^T\mathbf{Y}\big\|_F^2\Big].
\end{aligned}$$

The difference in the objective values is therefore

$$\begin{aligned}
f(\mathbf{W}) - f(\mathbf{W}^\star) &= \frac{1}{n}\mathrm{tr}\Big[(\mathbf{W} - \mathbf{W}^\star)^T(\mathbf{X}^T\mathbf{X} + n\gamma\mathbf{I}_d)(\mathbf{W} - \mathbf{W}^\star)\Big]\\
&= \frac{1}{n}\Big\|(\mathbf{X}^T\mathbf{X} + n\gamma\mathbf{I}_d)^{1/2}(\mathbf{W} - \mathbf{W}^\star)\Big\|_F^2.
\end{aligned}$$

Because $\sigma_{\min}(\mathbf{A})\|\mathbf{B}\|_F \leq \|\mathbf{A}\mathbf{B}\|_F$ holds for any nonsingular $\mathbf{A}$ and any $\mathbf{B}$, it holds for any nonsingular matrix $\mathbf{M}$ that

$$\begin{aligned}
\sigma_{\min}^2\Big[(\mathbf{X}^T\mathbf{X} + n\gamma\mathbf{I}_d)^{1/2}\mathbf{M}\Big]\Big\|\mathbf{M}^{-1}(\mathbf{W} - \mathbf{W}^\star)\Big\|_F^2 &\leq \Big\|(\mathbf{X}^T\mathbf{X} + n\gamma\mathbf{I}_d)^{1/2}\mathbf{M}\mathbf{M}^{-1}(\mathbf{W} - \mathbf{W}^\star)\Big\|_F^2\\
&= \Big\|(\mathbf{X}^T\mathbf{X} + n\gamma\mathbf{I}_d)^{1/2}(\mathbf{W} - \mathbf{W}^\star)\Big\|_F^2.
\end{aligned}$$

The last claim in the lemma follows from the above inequality. ■

B.2 Proof of Theorem 15

Proof Let $\rho = \mathrm{rank}(\mathbf{X})$, $\mathbf{U} \in \mathbb{R}^{n\times\rho}$ be the left singular vectors of $\mathbf{X}$, and $\mathbf{Y}^\perp = \mathbf{Y} - \mathbf{X}\mathbf{X}^\dagger\mathbf{Y} = \mathbf{Y} - \mathbf{U}\mathbf{U}^T\mathbf{Y}$. It follows from the definition of $\mathbf{W}^\star$ and $\mathbf{W}^\mathrm{c}$ that

$$\mathbf{W}^\mathrm{c} - \mathbf{W}^\star \;=\; (\mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{X} + n\gamma\mathbf{I}_d)^{-1}\mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{Y} - (\mathbf{X}^T\mathbf{X} + n\gamma\mathbf{I}_d)^{-1}\mathbf{X}^T\mathbf{Y}.$$

It follows that

$$\begin{aligned}
&(\mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{X} + n\gamma\mathbf{I}_d)(\mathbf{W}^\mathrm{c} - \mathbf{W}^\star)\\
&= \mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{Y}^\perp + \mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{X}\mathbf{X}^\dagger\mathbf{Y} - (\mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{X} + n\gamma\mathbf{I}_d)(\mathbf{X}^T\mathbf{X} + n\gamma\mathbf{I}_d)^{-1}\mathbf{X}^T\mathbf{Y}\\
&= \mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{Y}^\perp - n\gamma\mathbf{X}^\dagger\mathbf{Y} + (\mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{X} + n\gamma\mathbf{I}_d)\big[\mathbf{X}^\dagger - (\mathbf{X}^T\mathbf{X} + n\gamma\mathbf{I}_d)^{-1}\mathbf{X}^T\big]\mathbf{Y}\\
&= \mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{Y}^\perp - n\gamma\mathbf{X}^\dagger\mathbf{Y} + n\gamma(\mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{X} + n\gamma\mathbf{I}_d)(\mathbf{X}^T\mathbf{X} + n\gamma\mathbf{I}_d)^{-1}\mathbf{X}^\dagger\mathbf{Y}\\
&= \mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{Y}^\perp + n\gamma(\mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{X} - \mathbf{X}^T\mathbf{X})(\mathbf{X}^T\mathbf{X} + n\gamma\mathbf{I}_d)^{-1}\mathbf{X}^\dagger\mathbf{Y}.
\end{aligned}$$

It follows that

$$(\mathbf{X}^T\mathbf{X} + n\gamma\mathbf{I}_d)^{-1/2}(\mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{X} + n\gamma\mathbf{I}_d)(\mathbf{W}^\mathrm{c} - \mathbf{W}^\star) \;=\; \mathbf{A} + \mathbf{B}, \tag{13}$$

where

$$\begin{aligned}\mathbf{A} &= \big[(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{1/2}\big]^\dagger\mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{Y}^\perp \;=\; \mathbf{V}(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}\mathbf{\Sigma}\mathbf{U}\mathbf{S}\mathbf{S}^T\mathbf{Y}^\perp,\\ \mathbf{B} &= n\gamma\big[(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{1/2}\big]^\dagger(\mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{X}-\mathbf{X}^T\mathbf{X})(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^\dagger\mathbf{X}^\dagger\mathbf{Y}\\ &= n\gamma\mathbf{V}(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}\mathbf{\Sigma}(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}-\mathbf{I}_\rho)\mathbf{\Sigma}(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1}\mathbf{\Sigma}^{-1}\mathbf{U}^T\mathbf{Y}\\ &= n\gamma\mathbf{V}\mathbf{\Sigma}(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}-\mathbf{I}_\rho)(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1}\mathbf{U}^T\mathbf{Y}.\end{aligned}$$

It follows from (13) that

$$\begin{aligned}&(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{1/2}\big(\mathbf{W}^{\mathrm{c}}-\mathbf{W}^\star\big)\\ &= \big[(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{-1/2}(\mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{X}+n\gamma\mathbf{I}_d)(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{-1/2}\big]^\dagger(\mathbf{A}+\mathbf{B}).\end{aligned}$$

By Assumption 1.1, we have that

$$(1-\eta)(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d) \;\preceq\; (\mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{X}+n\gamma\mathbf{I}_d) \;\preceq\; (1+\eta)(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d).$$

It follows that

$$\Big\|\big[(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{-1/2}(\mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{X}+n\gamma\mathbf{I}_d)(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{-1/2}\big]^\dagger\Big\|_2 \;\le\; \frac{1}{1-\eta}.$$

Thus

$$\Big\|(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{1/2}\big(\mathbf{W}^{\mathrm{c}}-\mathbf{W}^\star\big)\Big\|_F^2 \;\le\; \frac{1}{1-\eta}\Big\|\mathbf{A}+\mathbf{B}\Big\|_F^2 \;\le\; \frac{2}{1-\eta}\Big(\|\mathbf{A}\|_F^2+\|\mathbf{B}\|_F^2\Big).$$

Lemma 25 shows

$$f\big(\mathbf{W}^{\mathrm{c}}\big)-f\big(\mathbf{W}^\star\big)=\frac{1}{n}\Big\|(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{1/2}(\mathbf{W}^{\mathrm{c}}-\mathbf{W}^\star)\Big\|_F^2\le\frac{2}{n(1-\eta)}\Big(\|\mathbf{A}\|_F^2+\|\mathbf{B}\|_F^2\Big). \quad (14)$$

We respectively bound $\|\mathbf{A}\|_F^2$ and $\|\mathbf{B}\|_F^2$ in the following. It follows from Assumption 1.2 and $\mathbf{U}^T\mathbf{Y}^\perp=\mathbf{0}$ that

$$\begin{aligned}\|\mathbf{A}\|_F^2 &= \Big\|\mathbf{V}(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}\mathbf{\Sigma}\mathbf{U}\mathbf{S}\mathbf{S}^T\mathbf{Y}^\perp\Big\|_F^2\\ &\le \big\|(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}\mathbf{\Sigma}\big\|_2^2\,\big\|\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{Y}^\perp-\mathbf{U}^T\mathbf{Y}^\perp\big\|_F^2\\ &\le \epsilon\big\|(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}\mathbf{\Sigma}\big\|_2^2\,\big\|\mathbf{Y}^\perp\big\|_F^2.\end{aligned}$$

By the definition of $\mathbf{B}$, we have

$$\begin{aligned}&\|\mathbf{B}\|_F^2 \;\le\; n^2\gamma^2\big\|\mathbf{\Sigma}(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}-\mathbf{I}_\rho)(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1}\mathbf{U}^T\mathbf{Y}\big\|_F^2\\ &\le\; n^2\gamma^2\big\|\mathbf{\Sigma}(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}-\mathbf{I}_\rho)(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}\big\|_2^2\big\|(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}\mathbf{U}^T\mathbf{Y}\big\|_F^2\\ &=\; n^2\gamma^2\big\|\mathbf{\Sigma}\mathbf{N}\big\|_2^2\big\|(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}\mathbf{U}^T\mathbf{Y}\big\|_F^2,\end{aligned}$$

where we define $\mathbf{N}=(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}-\mathbf{I}_\rho)(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}$. By Assumption 1.1, we have

$$-\eta(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1} \;\preceq\; \mathbf{N} \;\preceq\; \eta(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1}.$$

It follows that

$$\begin{aligned}\|\mathbf{B}\|_F^2 &\le\; n^2\gamma^2\big\|\mathbf{\Sigma}\mathbf{N}^2\mathbf{\Sigma}\big\|_2\big\|(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}\mathbf{U}^T\mathbf{Y}\big\|_F^2\\ &\le\; \eta^2n^2\gamma^2\big\|\mathbf{\Sigma}(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-2}\mathbf{\Sigma}\big\|_2\big\|(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}\mathbf{U}^T\mathbf{Y}\big\|_F^2\\ &=\; \eta^2n^2\gamma^2\big\|(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1}\mathbf{\Sigma}\big\|_2^2\big\|(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}\mathbf{U}^T\mathbf{Y}\big\|_F^2\\ &=\; \eta^2n\gamma\big\|(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}\mathbf{\Sigma}\big\|_2^2\big\|(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}\mathbf{U}^T\mathbf{Y}\big\|_F^2.\end{aligned}$$

The last equality follows from the fact that $\|(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}\|_2\le(n\gamma)^{-1/2}$. It follows that

$$\begin{aligned}\|\mathbf{A}\|_F^2+\|\mathbf{B}\|_F^2 &\le \max\big\{\epsilon,\eta^2\big\}\Big\|(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_d)^{-1}\mathbf{\Sigma}\Big\|_2\Big[\big\|\mathbf{Y}^\perp\big\|_F^2+n\gamma\big\|(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_d)^{-1/2}\mathbf{U}^T\mathbf{Y}\big\|_F^2\Big]\\ &\le \max\big\{\epsilon,\eta^2\big\}\frac{\sigma_{\max}^2}{\sigma_{\max}^2+n\gamma}\Big[\big\|\mathbf{Y}^\perp\big\|_F^2+n\gamma\big\|(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_d)^{-1/2}\mathbf{U}^T\mathbf{Y}\big\|_F^2\Big]\\ &\le \max\big\{\epsilon,\eta^2\big\}\beta nf(\mathbf{W}^\star). \quad (15)\end{aligned}$$

The last inequality follows from Lemma 25. The claimed result now follows from (15) and (14). ■

B.3 Proof of Theorem 16

Proof By the definition of $\mathbf{W}^{\mathrm{h}}$ and $\mathbf{W}^\star$, we have

$$\begin{aligned}&(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{1/2}\big(\mathbf{W}^{\mathrm{h}}-\mathbf{W}^\star\big)\\ &= (\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{1/2}\Big[(\mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{X}+n\gamma\mathbf{I}_d)^\dagger-(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^\dagger\Big]\mathbf{X}^T\mathbf{Y}\\ &= \mathbf{V}(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{1/2}\Big[(\mathbf{\Sigma}\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}\mathbf{\Sigma}+n\gamma\mathbf{I}_\rho)^\dagger-(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1}\Big]\mathbf{\Sigma}\mathbf{U}^T\mathbf{Y}.\end{aligned}$$

It follows from Assumption 1.1 that $\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}$ has full rank, and thus

$$\begin{aligned}&(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{1/2}\big(\mathbf{W}^{\mathrm{h}}-\mathbf{W}^\star\big)\\ &= \mathbf{V}(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{1/2}\Big[(\mathbf{\Sigma}\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}\mathbf{\Sigma}+n\gamma\mathbf{I}_\rho)^{-1}-(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1}\Big]\mathbf{\Sigma}\mathbf{U}^T\mathbf{Y}\\ &= \mathbf{V}(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{1/2}(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1}(\mathbf{\Sigma}^2-\mathbf{\Sigma}\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}\mathbf{\Sigma})(\mathbf{\Sigma}\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}\mathbf{\Sigma}+n\gamma\mathbf{I}_\rho)^{-1}\mathbf{\Sigma}\mathbf{U}^T\mathbf{Y}\\ &= \mathbf{V}(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}\mathbf{\Sigma}(\mathbf{I}_\rho-\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U})\mathbf{\Sigma}(\mathbf{\Sigma}\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}\mathbf{\Sigma}+n\gamma\mathbf{I}_\rho)^{-1}\mathbf{\Sigma}\mathbf{U}^T\mathbf{Y},\end{aligned}$$

where the second equality follow from $\mathbf{M}^{-1}-\mathbf{N}^{-1}=\mathbf{N}^{-1}(\mathbf{N}-\mathbf{M})\mathbf{M}^{-1}$. We define

$$(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{1/2}\big(\mathbf{W}^{\mathrm{h}}-\mathbf{W}^\star\big) \;=\; \mathbf{V}\mathbf{A}\mathbf{B}\mathbf{C},$$

where

$$\begin{aligned}\mathbf{A} &= (\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}\mathbf{\Sigma}(\mathbf{I}_\rho-\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U})\mathbf{\Sigma}(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2},\\ \mathbf{B} &= (\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{1/2}(\mathbf{\Sigma}\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}\mathbf{\Sigma}+n\gamma\mathbf{I}_\rho)^{-1}(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{1/2},\\ \mathbf{C} &= (\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}\mathbf{\Sigma}\mathbf{U}^T\mathbf{Y}.\end{aligned}$$

It follows from Assumption 1.1 that

$$\begin{aligned}\|\mathbf{A}\|_2 &\leq \eta\Big\|(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}\boldsymbol{\Sigma}^2(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}\Big\|_2 \leq \eta\beta,\\ \|\mathbf{B}\|_2 &\leq (1-\eta)^{-1}.\end{aligned}$$

It holds that

$$\begin{aligned}\big\|\mathbf{C}\big\|_F^2 &\leq \Big\|(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}\boldsymbol{\Sigma}\mathbf{U}^T\mathbf{Y}\Big\|_F^2\\ &= \Big[\operatorname{tr}\Big(\mathbf{Y}^T\mathbf{U}\mathbf{U}^T\mathbf{Y}\Big)-n\gamma\operatorname{tr}\Big(\mathbf{Y}^T\mathbf{U}(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_d)^{-1}\mathbf{U}^T\mathbf{Y}\Big)\Big]\\ &= \Big[-\operatorname{tr}\Big(\mathbf{Y}^T(\mathbf{I}_d-\mathbf{U}\mathbf{U}^T)\mathbf{Y}\Big)-n\gamma\operatorname{tr}\Big(\mathbf{Y}^T\mathbf{U}(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_d)^{\dagger}\mathbf{U}^T\mathbf{Y}\Big)+\operatorname{tr}\big(\mathbf{Y}^T\mathbf{Y}\big)\Big]\\ &= \Big(-nf(\mathbf{W}^\star)+\big\|\mathbf{Y}\big\|_F^2\Big),\end{aligned}$$

where the last equality follows from Lemma 25. It follows from Lemma 25 that

$$\begin{aligned}f(\mathbf{W}^{\mathrm{h}})-f(\mathbf{W}^\star) &= \frac{1}{n}\big\|(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{1/2}\big(\mathbf{W}^{\mathrm{h}}-\mathbf{W}^\star\big)\big\|_F^2\\ &= \frac{1}{n}\big\|\mathbf{A}\mathbf{B}\mathbf{C}\big\|_F^2 \leq \frac{\eta^2\beta^2}{(1-\eta)^2}\Big(\frac{1}{n}\big\|\mathbf{Y}\big\|_F^2-f(\mathbf{W}^\star)\Big).\end{aligned}$$

■

Appendix C. Sketched MRR from the Statistical Perspective: Proofs

In Section C.1 we prove Theorem 4. In Section C.2 we prove Theorem 17. In Section C.3 we prove Theorem 18. In Section A.2 we prove Theorem 13. In Section C.4 we prove Theorem 19. Recall that the fixed design model is $\mathbf{Y}=\mathbf{X}\mathbf{W}_0+\boldsymbol{\Xi}$ where $\boldsymbol{\Xi}$ is random, $\mathbb{E}\boldsymbol{\Xi}=0$, and $\mathbb{E}[\boldsymbol{\Xi}\boldsymbol{\Xi}^T]=\xi^2\mathbf{I}_n$.

C.1 Proofs of Theorem 4

We prove Theorem 4 in the following. In the proof we exploit several identities. The Frobenius norm and matrix trace satisfy

$$\|\mathbf{A}-\mathbf{B}\|_F^2 \;=\; \operatorname{tr}\big[(\mathbf{A}-\mathbf{B})(\mathbf{A}-\mathbf{B})^T)\big] \;=\; \operatorname{tr}(\mathbf{A}\mathbf{A}^T)+\operatorname{tr}(\mathbf{B}\mathbf{B}^T)-2\operatorname{tr}(\mathbf{A}\mathbf{B}^T)$$

for any conformal matrices $\mathbf{A}$ and $\mathbf{B}$. The trace is linear, and thus for any fixed $\mathbf{A}$ and $\mathbf{B}$ and conformal random matrix $\boldsymbol{\Psi}$,

$$\mathbb{E}\big[\operatorname{tr}(\mathbf{A}\boldsymbol{\Psi}\mathbf{B})\big] \;=\; \operatorname{tr}\big[\mathbf{A}(\mathbb{E}\boldsymbol{\Psi})\mathbf{B}\big],$$

where the expectation is taken with respect to $\boldsymbol{\Psi}$.

Proof It follows from the definition of the optimal solution $\mathbf{W}^\star$ in (2) that

$$\begin{aligned}\mathbf{X}\mathbf{W}^\star &= \mathbf{X}(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{\dagger}\mathbf{X}^T(\mathbf{X}\mathbf{W}_0+\boldsymbol{\Xi})\\ &= \mathbf{U}(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1}\boldsymbol{\Sigma}^3\mathbf{V}^T\mathbf{W}_0+\mathbf{U}(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1}\boldsymbol{\Sigma}^2\mathbf{U}^T\boldsymbol{\Xi}\\ &= \mathbf{U}\Big[\mathbf{I}_\rho-n\gamma(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1}\Big]\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{W}_0+\mathbf{U}(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1}\boldsymbol{\Sigma}^2\mathbf{U}^T\boldsymbol{\Xi}\\ &= \mathbf{X}\mathbf{W}_0-n\gamma\mathbf{U}(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{W}_0+\mathbf{U}(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1}\boldsymbol{\Sigma}^2\mathbf{U}^T\boldsymbol{\Xi}.\end{aligned}$$

Since $\mathbb{E}[\boldsymbol{\Xi}]=\mathbf{0}$ and $\mathbb{E}[\boldsymbol{\Xi}\boldsymbol{\Xi}^T]=\xi^2\mathbf{I}_n$, it holds that

$$\begin{aligned}R(\mathbf{W}^\star) &= \frac{1}{n}\mathbb{E}\big\|\mathbf{X}\mathbf{W}^\star-\mathbf{X}\mathbf{W}_0\big\|_F^2\\ &= \frac{1}{n}\Big\|-n\gamma(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{W}_0+(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1}\boldsymbol{\Sigma}^2\mathbf{U}^T\boldsymbol{\Xi}\Big\|_F^2\\ &= n\gamma^2\Big\|(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{W}_0\Big\|_F^2+\frac{\xi^2}{n}\Big\|(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1}\boldsymbol{\Sigma}^2\Big\|_F^2.\end{aligned}$$

This exposes expressions for the bias and variance of the optimal solution $\mathbf{W}^\star$.

We now decompose the risk function $R\big(\mathbf{W}^{\mathrm{c}}\big)$. It follows from the definition of $\mathbf{W}^{\mathrm{c}}$ in (3) that

$$\begin{aligned}\mathbf{X}\mathbf{W}^{\mathrm{c}} &= \mathbf{X}(\mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{\dagger}\mathbf{X}^T\mathbf{S}\mathbf{S}^T(\mathbf{X}\mathbf{W}_0+\boldsymbol{\Xi})\\ &= \mathbf{U}\boldsymbol{\Sigma}\big(\boldsymbol{\Sigma}\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}\boldsymbol{\Sigma}+n\gamma\mathbf{I}_d\big)^{\dagger}\boldsymbol{\Sigma}\big(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{W}_0+\mathbf{U}^T\mathbf{S}\mathbf{S}^T\boldsymbol{\Xi}\big)\\ &= \mathbf{U}(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2})^{-1}\Big[(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2})\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{W}_0-n\gamma\boldsymbol{\Sigma}^{-1}\mathbf{V}^T\mathbf{W}_0+\mathbf{U}^T\mathbf{S}\mathbf{S}^T\boldsymbol{\Xi}\Big]\\ &= \mathbf{X}\mathbf{W}_0+\mathbf{U}(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2})^{-1}\big(-n\gamma\boldsymbol{\Sigma}^{-1}\mathbf{V}^T\mathbf{W}_0+\mathbf{U}^T\mathbf{S}\mathbf{S}^T\boldsymbol{\Xi}\big).\end{aligned}$$

Since $\mathbb{E}[\boldsymbol{\Xi}]=\mathbf{0}$ and $\mathbb{E}[\boldsymbol{\Xi}\boldsymbol{\Xi}^T]=\xi^2\mathbf{I}_n$, it follows that

$$\begin{aligned}R\big(\mathbf{W}^{\mathrm{c}}\big) &= \frac{1}{n}\mathbb{E}\big\|\mathbf{X}\mathbf{W}^{\mathrm{c}}-\mathbf{X}\mathbf{W}_0\big\|_F^2\\ &= \frac{1}{n}\Big\|-n\gamma(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2})^{-1}\boldsymbol{\Sigma}^{-1}\mathbf{V}^T\mathbf{W}_0+(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2})^{-1}\mathbf{U}^T\mathbf{S}\mathbf{S}^T\boldsymbol{\Xi}\Big\|_F^2\\ &= n\gamma^2\Big\|(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2})^{-1}\boldsymbol{\Sigma}^{-1}\mathbf{V}^T\mathbf{W}_0\Big\|_F^2+\frac{\xi^2}{n}\Big\|(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2})^{-1}\mathbf{U}^T\mathbf{S}\mathbf{S}^T\Big\|_F^2.\end{aligned}$$

This exposes expressions for the bias and variance of the approximate solution $\mathbf{W}^{\mathrm{c}}$.

We now decompose the risk function $R\big(\mathbf{W}^{\mathrm{h}}\big)$. It follows from the definition of $\mathbf{W}^{\mathrm{h}}$ in (4) that

$$\begin{aligned}\mathbf{X}\mathbf{W}^{\mathrm{h}}-\mathbf{X}\mathbf{W}_0 &= \mathbf{X}(\mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{X}+n\gamma\mathbf{I}_n)^{\dagger}\mathbf{X}^T(\mathbf{X}\mathbf{W}_0+\boldsymbol{\Xi})-\mathbf{X}\mathbf{W}_0\\ &= \mathbf{X}(\mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{\dagger}\mathbf{X}^T\mathbf{X}\mathbf{W}_0-\mathbf{X}\mathbf{W}_0+\mathbf{X}(\mathbf{X}^T\mathbf{S}\mathbf{S}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{\dagger}\mathbf{X}^T\boldsymbol{\Xi}\\ &= \mathbf{U}\big[(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2})^{-1}-\mathbf{I}_\rho^{-1}\big]\mathbf{U}^T\mathbf{X}\mathbf{W}_0+\mathbf{U}(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}^T+n\gamma\boldsymbol{\Sigma}^{-2})^{\dagger}\mathbf{U}^T\boldsymbol{\Xi}\\ &= \mathbf{U}\big(\mathbf{I}_\rho-\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}-n\gamma\boldsymbol{\Sigma}^{-2}\big)\big(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2}\big)^{-1}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{W}_0\\ &\quad+\mathbf{U}(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}^T+n\gamma\boldsymbol{\Sigma}^{-2})^{\dagger}\mathbf{U}^T\boldsymbol{\Xi},\end{aligned}$$

where the last equality follows from the fact that $\mathbf{A}^{-1}-\mathbf{B}^{-1}=\mathbf{B}^{-1}(\mathbf{B}-\mathbf{A})\mathbf{A}^{-1}$ for any conformal nonsingular matrices $\mathbf{A}$ and $\mathbf{B}$. Since $\mathbb{E}[\boldsymbol{\Xi}]=\mathbf{0}$ and $\mathbb{E}[\boldsymbol{\Xi}\boldsymbol{\Xi}^T]=\xi^2\mathbf{I}_n$, it follows that

$$R\big(\mathbf{W}^{\mathrm{h}}\big) \;=\; \mathsf{bias}^2\big(\mathbf{W}^{\mathrm{h}}\big)+\mathsf{var}\big(\mathbf{W}^{\mathrm{h}}\big),$$

where

$$\begin{aligned}\mathsf{bias}^2\big(\mathbf{W}^{\mathrm{h}}\big) &= \frac{1}{n}\Big\|\big(n\gamma\boldsymbol{\Sigma}^{-2}+\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}-\mathbf{I}_\rho\big)\big(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2}\big)^{-1}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{W}_0\Big\|_F^2,\\ \mathsf{var}\big(\mathbf{W}^{\mathrm{h}}\big) &= \frac{\xi^2}{n}\Big\|\big(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2}\big)^{-1}\Big\|_F^2.\end{aligned}$$

This exposes expressions for the bias and variance of $\mathbf{W}^{\mathrm{h}}$. ■

C.2 Proof of Theorem 17

Proof Assumption 1.1 ensures that $(1-\eta)\mathbf{I}_\rho\preceq\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}\preceq(1+\eta)\mathbf{I}_\rho$. It follows that

$$(1-\eta)\big(\mathbf{I}_\rho+n\gamma\boldsymbol{\Sigma}^{-2}\big) \;\preceq\; \mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2} \;\preceq\; (1+\eta)\big(\mathbf{I}_\rho+n\gamma\boldsymbol{\Sigma}^{-2}\big).$$

The bias term can be written as

$$\begin{aligned}\mathsf{bias}^2\big(\mathbf{W}^{\mathrm{c}}\big) &= n\gamma^2\big\|\big(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2}\big)^{\dagger}\boldsymbol{\Sigma}^{-1}\mathbf{V}^T\mathbf{W}_0\big\|_F^2\\ &= n\gamma^2\,\mathrm{tr}\Big(\mathbf{W}_0^T\mathbf{V}\boldsymbol{\Sigma}^{-1}\big[(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2})^{\dagger}\big]^2\boldsymbol{\Sigma}^{-1}\mathbf{V}^T\mathbf{W}_0\Big)\\ &\le \tfrac{n\gamma^2}{(1-\eta)^2}\big\|\big(\mathbf{I}_\rho+n\gamma\boldsymbol{\Sigma}^{-2}\big)^{-1}\boldsymbol{\Sigma}^{-1}\mathbf{V}^T\mathbf{W}_0\big\|_F^2\\ &= \tfrac{n\gamma^2}{(1-\eta)^2}\big\|\big(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_\rho\big)^{-1}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{W}_0\big\|_F^2\\ &= \tfrac{1}{(1-\eta)^2}\,\mathsf{bias}^2(\mathbf{W}^\star).\end{aligned}$$

We can analogously show $\mathsf{bias}^2(\mathbf{W}^{\mathrm{c}})\ge\frac{1}{(1+\eta)^2}\mathsf{bias}^2(\mathbf{W}^\star)$.

Let $\mathbf{B}=\big(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2}\big)^{\dagger}\mathbf{U}^T\mathbf{S}\in\mathbb{R}^{\rho\times s}$. By Assumption 1.1, it holds that

$$(1-\eta)\big[\big(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2}\big)^2\big]^{\dagger} \;\preceq\; \mathbf{B}\mathbf{B}^T \;\preceq\; (1+\eta)\big[\big(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2}\big)^2\big]^{\dagger}.$$

Applying Assumption 1.1 again, we obtain

$$(1-\eta)^2\big(\mathbf{I}_\rho+n\gamma\boldsymbol{\Sigma}^{-2}\big)^2 \;\preceq\; \big(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2}\big)^2 \;\preceq\; (1+\eta)^2\big(\mathbf{I}_\rho+n\gamma\boldsymbol{\Sigma}^{-2}\big)^2.$$

Note that both sides are nonsingular. Combining the above two equations, we have

$$\tfrac{1-\eta}{(1+\eta)^2}\big(\mathbf{I}_\rho+n\gamma\boldsymbol{\Sigma}^{-2}\big)^{-2} \;\preceq\; \mathbf{B}\mathbf{B}^T \;\preceq\; \tfrac{1+\eta}{(1-\eta)^2}\big(\mathbf{I}_\rho+n\eta\boldsymbol{\Sigma}^{-2}\big)^{-2}.$$

Taking the trace of all the terms, we obtain

$$\tfrac{1-\eta}{(1+\eta)^2} \;\le\; \tfrac{\|\mathbf{B}\|_F^2}{\|(\mathbf{I}_\rho+n\gamma\boldsymbol{\Sigma}^{-2})^{-1}\|_F^2} \;\le\; \tfrac{1+\eta}{(1-\eta)^2}.$$

The variance term can be written as

$$\begin{aligned}\mathsf{var}\big(\mathbf{W}^{\mathrm{c}}\big) &= \tfrac{\xi^2}{n}\big\|\mathbf{B}\mathbf{S}^T\big\|_F^2 \;\le\; \tfrac{\xi^2}{n}\big\|\mathbf{B}\big\|_F^2\big\|\mathbf{S}\big\|_2^2\\ &\le \tfrac{\xi^2(1+\eta)}{n(1-\eta)^2}\big\|\big(\mathbf{I}_\rho+n\gamma\boldsymbol{\Sigma}^{-2}\big)^{-1}\big\|_F^2\big\|\mathbf{S}\big\|_2^2\\ &= \tfrac{(1+\eta)\|\mathbf{S}\|_2^2}{(1-\eta)^2}\mathsf{var}(\mathbf{W}^\star).\end{aligned}$$

The upper bound on the variance follows from Assumption 1.3. ■

C.3 Proof of Theorem 18

Proof Let $\mathbf{B}=\big(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2}\big)^{\dagger}\mathbf{U}^T\mathbf{S}\in\mathbb{R}^{\rho\times s}$. In the proof of Theorem 5 we show that

$$\mathsf{var}\big(\mathbf{W}^{\mathrm{c}}\big) \;=\; \tfrac{\xi^2}{n}\big\|\mathbf{B}\mathbf{S}^T\big\|_F^2.$$

If $\mathbf{S}^T\mathbf{S}\succeq\frac{\vartheta n}{s}\mathbf{I}_s$, then it holds that

$$\mathsf{var}\big(\mathbf{W}^{\mathrm{c}}\big) \;=\; \tfrac{\xi^2}{n}\big\|\mathbf{B}\mathbf{S}^T\big\|_F^2 \;\ge\; \tfrac{\vartheta n}{s}\tfrac{\xi^2}{n}\big\|\mathbf{B}\big\|_F^2 \;\ge\; \tfrac{\vartheta n}{s}\tfrac{1-\eta}{(1+\eta)^2}\mathsf{var}(\mathbf{W}^\star).$$

This establishes the lower bounds on the variance. ■

C.4 Proof of Theorem 19

Proof Theorem 4 shows that

$$\begin{aligned}\mathsf{bias}\big(\mathbf{W}^{\mathrm{h}}\big) &= \gamma\sqrt{n}\Big\|\Big(\boldsymbol{\Sigma}^{-2}+\tfrac{\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}-\mathbf{I}_\rho}{n\gamma}\Big)\big(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2}\big)^{\dagger}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{W}_0\Big\|_F\\ &= \gamma\sqrt{n}\big\|\mathbf{A}\boldsymbol{\Sigma}^2\mathbf{B}\big\|_F \;\le\; \gamma\sqrt{n}\big\|\mathbf{A}\boldsymbol{\Sigma}^2\big\|_2\big\|\mathbf{B}\big\|_F,\\ \mathsf{var}\big(\mathbf{W}^{\mathrm{h}}\big) &= \frac{\xi^2}{n}\Big\|\big(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2}\big)^{\dagger}\Big\|_F^2,\end{aligned}$$

where we define

$$\begin{aligned}\mathbf{A} &= \boldsymbol{\Sigma}^{-2}+\tfrac{\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}-\mathbf{I}_\rho}{n\gamma},\\ \mathbf{B} &= \boldsymbol{\Sigma}^{-2}\big(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\boldsymbol{\Sigma}^{-2}\big)^{\dagger}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{W}_0.\end{aligned}$$

We first analyze the bias. It follows from Assumption 1.1 that

$$\boldsymbol{\Sigma}^{-2}\big(\mathbf{I}_\rho-\tfrac{\eta}{n\gamma}\boldsymbol{\Sigma}^2\big) \;\preceq\; \mathbf{A} \;\preceq\; \boldsymbol{\Sigma}^{-2}\big(\mathbf{I}_\rho+\tfrac{\eta}{n\gamma}\boldsymbol{\Sigma}^2\big). \tag{16}$$

Since $\big(\mathbf{I}_\rho-\frac{\eta}{n\gamma}\boldsymbol{\Sigma}^2\big)^2\preceq\big(\mathbf{I}_\rho+\frac{\eta}{n\gamma}\boldsymbol{\Sigma}^2\big)^2\preceq\big(1+\frac{\eta\sigma_1^2}{n\gamma}\big)^2\mathbf{I}_\rho$, it follows that

$$\mathbf{A}^2 \;\preceq\; \boldsymbol{\Sigma}^{-4}\big(\mathbf{I}_\rho+\tfrac{\eta}{n\gamma}\boldsymbol{\Sigma}^2\big)^2 \;\preceq\; \big(1+\tfrac{\eta\sigma_1^2}{n\gamma}\big)^2\boldsymbol{\Sigma}^{-4}.$$

Thus

$$\left\|\mathbf{A}\mathbf{\Sigma}^2\right\|_2^2 \;=\; \left\|\mathbf{\Sigma}^2\mathbf{A}^2\mathbf{\Sigma}^2\right\|_2 \;\leq\; \left(1+\tfrac{\eta\sigma_1^2}{n\gamma}\right)^2.$$

It follows from Assumption 1.1 that

$$\begin{aligned}&(1+\eta)^{-1}\big(\mathbf{I}_\rho+n\gamma\mathbf{\Sigma}^{-2}\big)^{-1} \;\preceq\; \big((1+\eta)\mathbf{I}_\rho+n\gamma\mathbf{\Sigma}^{-2}\big)^{-1}\\ &\preceq\; \big(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\mathbf{\Sigma}^{-2}\big)^{\dagger} \;\preceq\; \big((1-\eta)\mathbf{I}_\rho+n\gamma\mathbf{\Sigma}^{-2}\big)^{-1} \;\preceq\; (1-\eta)^{-1}\big(\mathbf{I}_\rho+n\gamma\mathbf{\Sigma}^{-2}\big)^{-1}.\end{aligned}$$

Thus

$$\begin{aligned}\mathbf{B}^T\mathbf{B} \;&=\; \mathbf{W}_0^T\mathbf{V}\mathbf{\Sigma}^3\big(\mathbf{\Sigma}^{-2}(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\mathbf{\Sigma}^{-2})^{\dagger}\mathbf{\Sigma}^{-2}\big)^2\mathbf{\Sigma}^3\mathbf{V}^T\mathbf{W}_0\\ &\preceq\; (1-\eta)^{-2}\mathbf{W}_0^T\mathbf{V}\mathbf{\Sigma}^3\big(\mathbf{\Sigma}^{-2}(\mathbf{I}_\rho+n\gamma\mathbf{\Sigma}^{-2})^{-1}\mathbf{\Sigma}^{-2}\big)^2\mathbf{\Sigma}^3\mathbf{V}^T\mathbf{W}_0\\ &=\; (1-\eta)^{-2}\mathbf{W}_0^T\mathbf{V}\mathbf{\Sigma}\big(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho\big)^{-2}\mathbf{\Sigma}\mathbf{V}^T\mathbf{W}_0. \qquad (17)\end{aligned}$$

It follows that

$$\|\mathbf{B}\|_F^2 \;=\; \mathrm{tr}\big(\mathbf{B}^T\mathbf{B}\big) \;\leq\; (1-\eta)^{-2}\big\|\big(\mathbf{\Sigma}^{-2}+n\gamma\mathbf{I}_\rho\big)^{-1}\mathbf{\Sigma}\mathbf{V}^T\mathbf{W}_0\big\|_F^2 \;=\; \tfrac{\mathsf{bias}^2(\mathbf{W}^\star)}{n\gamma^2(1-\eta)^2},$$

where the last equality follows from the definition of $\mathsf{bias}(\mathbf{W}^\star)$. By the definition of $\mathbf{A}$ and $\mathbf{B}$, we have

$$\mathsf{bias}^2\big(\mathbf{W}^{\mathrm{h}}\big) \;\leq\; \gamma^2 n\left\|\mathbf{A}\mathbf{\Sigma}^2\right\|_2^2\|\mathbf{B}\|_F^2 \;=\; \tfrac{1}{(1-\eta)^2}\left(1+\tfrac{\eta\sigma_1^2}{n\gamma}\right)^2\mathsf{bias}^2\big(\mathbf{W}^\star\big).$$

Thus, the upper bound on $\mathsf{bias}\big(\mathbf{W}^{\mathrm{h}}\big)$ is established.

Using the same $\mathbf{A}$ and $\mathbf{B}$, we can also show that

$$\mathsf{bias}\big(\mathbf{W}^{\mathrm{h}}\big) \;=\; \gamma\sqrt{n}\left\|\mathbf{A}\mathbf{\Sigma}^2\mathbf{B}\right\|_F \;\geq\; \gamma\sqrt{n}\;\sigma_{\min}\big(\mathbf{A}\mathbf{\Sigma}^2\big)\left\|\mathbf{B}\right\|_F.$$

Assume that $\sigma_\rho^2 \geq \frac{n\gamma}{\eta}$. It follows from (16) that

$$\mathbf{A}^2 \;\succeq\; \big(\tfrac{\eta\sigma_\rho^2}{n\gamma}-1\big)^2\mathbf{\Sigma}^{-4}.$$

Thus

$$\sigma_{\min}^2\big(\mathbf{A}\mathbf{\Sigma}^2\big) \;=\; \sigma_{\min}\big(\mathbf{\Sigma}^2\mathbf{A}^2\mathbf{\Sigma}^2\big) \;\geq\; \big(\tfrac{\eta\sigma_\rho^2}{n\gamma}-1\big)^2.$$

It follows from (17) that

$$\mathbf{B}^T\mathbf{B} \;\succeq\; (1+\eta)^{-2}\mathbf{W}_0^T\mathbf{V}\mathbf{\Sigma}\big(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho\big)^{-2}\mathbf{\Sigma}\mathbf{V}^T\mathbf{W}_0.$$

Thus

$$\|\mathbf{B}\|_F^2 \;=\; \mathrm{tr}\big(\mathbf{B}^T\mathbf{B}\big) \;\geq\; (1+\eta)^{-2}\big\|\big(\mathbf{\Sigma}^{-2}+n\gamma\mathbf{I}_\rho\big)^{-1}\mathbf{\Sigma}\mathbf{V}^T\mathbf{W}_0\big\|_F^2 \;=\; \tfrac{1}{n\gamma^2(1+\eta)^2}\,\mathsf{bias}^2\big(\mathbf{W}^\star\big).$$

In sum, we obtain

$$\mathsf{bias}^2\big(\mathbf{W}^{\mathrm{h}}\big) \;\geq\; \gamma^2 n\;\sigma_{\min}^2\big(\mathbf{A}\mathbf{\Sigma}^2\big)\left\|\mathbf{B}\right\|_F^2 \;=\; (1+\eta)^{-2}\left(\tfrac{\eta\sigma_\rho^2}{n\gamma}-1\right)^2\mathsf{bias}^2\big(\mathbf{W}^\star\big).$$

Thus, the lower bound on $\mathsf{bias}\big(\mathbf{W}^{\mathrm{h}}\big)$ is established.

It follows from Assumption 1.1 that

$$(1+\eta)^{-1}\big(\mathbf{I}_\rho+n\gamma\mathbf{\Sigma}^{-2}\big)^{-1} \;\preceq\; \big(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\mathbf{\Sigma}^{-2}\big)^{-1} \;\preceq\; (1-\eta)^{-1}\big(\mathbf{I}_\rho+n\gamma\mathbf{\Sigma}^{-2}\big)^{-1}.$$

It follows from Theorem 4 that

$$\begin{aligned}\mathsf{var}\big(\mathbf{W}^{\mathrm{h}}\big) \;&=\; \tfrac{\xi^2}{n}\big\|\big(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}+n\gamma\mathbf{\Sigma}^{-2}\big)^{-1}\big\|_F^2\\ &\in\; \tfrac{1}{1\mp\eta}\tfrac{\xi^2}{n}\big\|\big(\mathbf{I}_\rho+n\gamma\mathbf{\Sigma}^{-2}\big)^{-1}\big\|_F^2\\ &=\; \tfrac{1}{1\mp\eta}\mathsf{var}\big(\mathbf{W}^\star\big).\end{aligned}$$

This concludes the proof. ■

Appendix D. Model Averaging from the Optimization Perspective: Proofs

In Section D.1 we prove Theorem 20. In Section D.2 we prove Theorem 21.

D.1 Proof of Theorem 20

Proof By Lemma 25, we only need to show that $\|(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{1/2}(\mathbf{W}^{\mathrm{c}}-\mathbf{W}^\star)\|_F^2 \leq n\alpha\beta f(\mathbf{W}^\star)$. In the proof, we define $\rho=\mathrm{rank}(\mathbf{X})$ and let $\sigma_1\geq\cdots\geq\sigma_\rho$ be the singular values of $\mathbf{X}$.

In the proof of Theorem 15 we show that

$$\begin{aligned}&(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{1/2}\big(\mathbf{W}_i^{\mathrm{c}}-\mathbf{W}^\star\big)\\ &=\; \big[(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{-1/2}(\mathbf{X}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{X}+n\gamma\mathbf{I}_d)(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{-1/2}\big]^{\dagger}\big(\mathbf{A}_i+\mathbf{B}_i\big)\\ &=\; \mathbf{C}_i^{\dagger}\big(\mathbf{A}_i+\mathbf{B}_i\big),\end{aligned}$$

where

$$\begin{aligned}\mathbf{A}_i \;&=\; \mathbf{V}(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}\mathbf{\Sigma}\mathbf{U}\mathbf{S}_i\mathbf{S}_i^T\mathbf{Y}^\perp,\\ \mathbf{B}_i \;&=\; n\gamma\mathbf{V}\mathbf{\Sigma}(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}(\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{U}-\mathbf{I}_\rho)(\mathbf{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1}\mathbf{U}^T\mathbf{Y}\\ \mathbf{C}_i \;&=\; \big[(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{1/2}\big]^{\dagger}\big(\mathbf{X}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{X}+n\gamma\mathbf{I}_d\big)\big[(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{1/2}\big]^{\dagger}\\ &=\; \mathbf{V}(\mathbf{I}_\rho+n\gamma\mathbf{\Sigma}^{-2})^{-1/2}(\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{U}+n\gamma\mathbf{\Sigma}^{-2})(\mathbf{I}_\rho+n\gamma\mathbf{\Sigma}^{-2})^{-1/2}\mathbf{V}^T\\ &=\; \mathbf{V}\mathbf{V}^T+\mathbf{V}(\mathbf{I}_\rho+n\gamma\mathbf{\Sigma}^{-2})^{-1/2}(\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{U}-\mathbf{I}_\rho)(\mathbf{I}_\rho+n\gamma\mathbf{\Sigma}^{-2})^{-1/2}\mathbf{V}^T.\end{aligned}$$

By Assumption 2.1, we have that $\mathbf{C}_i \succeq \big(1-\frac{\eta\,\sigma_{\max}^2}{\sigma_{\max}^2+n\gamma}\big)\mathbf{V}\mathbf{V}^T$. Since $\eta\leq 1/2$, it follows that $\mathbf{C}_i^{\dagger} \preceq \big(1+\frac{2\eta\,\sigma_{\max}^2}{\sigma_{\max}^2+n\gamma}\big)\mathbf{V}\mathbf{V}^T$. Let $\mathbf{C}_i^{\dagger}=\mathbf{V}\mathbf{V}^T+\mathbf{V}\mathbf{\Delta}_i\mathbf{V}^T$. It holds that $\mathbf{\Delta}_i \preceq \frac{2\eta\,\sigma_{\max}^2}{\sigma_{\max}^2+n\gamma}\mathbf{V}\mathbf{V}^T \preceq$

$2\eta\beta\mathbf{V}\mathbf{V}^T$. By definition, $\mathbf{W}^{\mathrm{c}} = \frac{1}{g}\sum_{i=1}^g \mathbf{W}_i^{\mathrm{c}}$. It follows that

$$\begin{aligned}
\Big\|(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{1/2}(\mathbf{W}_i^{\mathrm{c}}-\mathbf{W}^\star)\Big\|_F &= \Big\|\frac{1}{g}\sum_{i=1}^g \mathbf{C}_i^\dagger(\mathbf{A}_i+\mathbf{B}_i)\Big\|_F \\
&\le \Big\|\frac{1}{g}\sum_{i=1}^g(\mathbf{A}_i+\mathbf{B}_i)\Big\|_F + \Big\|\frac{1}{g}\sum_{i=1}^g \mathbf{V}\boldsymbol{\Delta}_i\mathbf{V}^T(\mathbf{A}_i+\mathbf{B}_i)\Big\|_F \\
&\le \Big\|\frac{1}{g}\sum_{i=1}^g \mathbf{A}_i\Big\|_F + \Big\|\frac{1}{g}\sum_{i=1}^g \mathbf{B}_i\Big\|_F + \frac{1}{g}\sum_{i=1}^g \|\boldsymbol{\Delta}_i\|_2\Big(\|\mathbf{A}_i\|_F+\|\mathbf{B}_i\|_F\Big) \\
&\le \Big\|\frac{1}{g}\sum_{i=1}^g \mathbf{A}_i\Big\|_F + \Big\|\frac{1}{g}\sum_{i=1}^g \mathbf{B}_i\Big\|_F + 2\eta\beta\frac{1}{g}\sum_{i=1}^g\Big(\|\mathbf{A}_i\|_F+\|\mathbf{B}_i\|_F\Big). \qquad (18)
\end{aligned}$$

By Assumption 2.3, we have that

$$\frac{1}{g}\sum_{i=1}^g \|\mathbf{A}_i\|_F = \|(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_d)^{-1/2}\boldsymbol{\Sigma}\|_2\cdot\frac{1}{g}\sum_{i=1}^g\|\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{Y}^\perp\|_F \le \sqrt{\tfrac{\epsilon\,\sigma_{\max}^2}{\sigma_{\max}^2+n\gamma}}\|\mathbf{Y}^\perp\|_F.$$

We apply Assumption 2.1 and follow the proof of Theorem 15 to show that

$$\|\mathbf{B}_i\|_F^2 \le \eta^2 n\gamma\tfrac{\sigma_{\max}^2}{\sigma_{\max}^2+n\gamma}\Big\|(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_d)^{-1/2}\mathbf{U}^T\mathbf{Y}\Big\|_F^2.$$

It follows that

$$\begin{aligned}
&\frac{1}{g}\sum_{i=1}^g\Big(\|\mathbf{A}_i\|_F+\|\mathbf{B}_i\|_F\Big) \\
&\le \max\Big\{\sqrt{\epsilon},\eta\Big\}\sqrt{\tfrac{\sigma_{\max}^2}{\sigma_{\max}^2+n\gamma}}\Big(\|\mathbf{Y}^\perp\|_F+\sqrt{n\gamma}\|(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_d)^{-1/2}\mathbf{U}^T\mathbf{Y}\|_F\Big) \\
&\le \max\Big\{\sqrt{\epsilon},\eta\Big\}\sqrt{\beta}\sqrt{2\|\mathbf{Y}^\perp\|_F^2+2n\gamma\|(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_d)^{-1/2}\mathbf{U}^T\mathbf{Y}\|_F^2} \\
&= \max\big\{\sqrt{\epsilon},\eta\big\}\sqrt{\beta}\,\sqrt{2n\,f(\mathbf{W}^\star)}. \qquad (19)
\end{aligned}$$

Here the equality follows from Lemma 25. Let $\mathbf{S} = \frac{1}{g}[\mathbf{S}_1,\cdots,\mathbf{S}_g]\in\mathbb{R}^{n\times sg}$. We have that

$$\begin{aligned}
\frac{1}{g}\sum_{i=1}^g\mathbf{A}_i &= \mathbf{V}(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_d)^{-1/2}\boldsymbol{\Sigma}\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{Y}^\perp, \\
\frac{1}{g}\sum_{i=1}^g\mathbf{B}_i &= n\gamma\mathbf{V}\boldsymbol{\Sigma}(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_d)^{-1/2}(\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}-\mathbf{I}_\rho)(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_d)^{-1}\mathbf{U}^T\mathbf{Y}.
\end{aligned}$$

Applying Assumptions 2.1 and 2.2, we use the same techniques as in the above to obtain

$$\begin{aligned}
&\Big\|\frac{1}{g}\sum_{i=1}^g\mathbf{A}_i\Big\|_F+\Big\|\frac{1}{g}\sum_{i=1}^g\mathbf{B}_i\Big\|_F \le \sqrt{2\Big\|\frac{1}{g}\sum_{i=1}^g\mathbf{A}_i\Big\|_F^2+2\Big\|\frac{1}{g}\sum_{i=1}^g\mathbf{B}_i\Big\|_F^2} \\
&\le \max\Big\{\tfrac{\sqrt{\epsilon}}{\sqrt{g}},\tfrac{\eta}{\sqrt{g}}\Big\}\sqrt{\tfrac{\sigma_{\max}^2}{\sigma_{\max}^2+n\gamma}}\sqrt{2n\,f(\mathbf{W}^\star)} = \max\big\{\sqrt{\epsilon},\eta\big\}\tfrac{\sqrt{\beta}}{\sqrt{g}}\sqrt{2n\,f(\mathbf{W}^\star)}. \qquad (20)
\end{aligned}$$

It follows from (18), (19), and (20) that

$$\begin{aligned}
&\Big\|(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{1/2}(\mathbf{W}_i^{\mathrm{c}}-\mathbf{W}^\star)\Big\|_F \\
&\le \Big\|\frac{1}{g}\sum_{i=1}^g\mathbf{A}_i\Big\|_F+\Big\|\frac{1}{g}\sum_{i=1}^g\mathbf{B}_i\Big\|_F+2\eta\beta\frac{1}{g}\sum_{i=1}^g\Big(\|\mathbf{A}_i\|_F+\|\mathbf{B}_i\|_F\Big) \\
&\le \Big[\tfrac{1}{\sqrt{g}}\max\big\{\sqrt{\epsilon},\eta\big\}+2\beta\eta\cdot\max\big\{\sqrt{\epsilon},\eta\big\}\Big]\sqrt{\beta}\sqrt{2n\,f(\mathbf{W}^\star)} \\
&= \max\big\{\sqrt{\epsilon},\eta\big\}\cdot\big(\tfrac{1}{\sqrt{g}}+2\beta\eta\big)\sqrt{\beta}\sqrt{2n\,f(\mathbf{W}^\star)} \\
&= \sqrt{\alpha\beta n\,f(\mathbf{W}^\star)}.
\end{aligned}$$

This concludes our proof. ■

D.2 Proof of Theorem 21

Proof By Lemma 25, we only need to show that $\big\|(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{1/2}\big(\mathbf{W}^{\mathrm{h}}-\mathbf{W}^\star\big)\big\|_F^2 \le \alpha^2\beta^2\big(-nf(\mathbf{W}^\star)+\|\mathbf{Y}\|_F^2\big)$.

In the proof of Theorem 2 we show that

$$(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{1/2}\big(\mathbf{W}_i^{\mathrm{h}}-\mathbf{W}^\star\big) = \mathbf{V}\mathbf{A}_i\mathbf{B}_i\mathbf{C},$$

where

$$\begin{aligned}
\mathbf{A}_i &= (\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}\boldsymbol{\Sigma}(\mathbf{I}_\rho-\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{U})\boldsymbol{\Sigma}(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}, \\
\mathbf{B}_i &= (\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{1/2}(\boldsymbol{\Sigma}\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{U}\boldsymbol{\Sigma}+n\gamma\mathbf{I}_\rho)^{-1}(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{1/2}, \\
\mathbf{C} &= (\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1/2}\boldsymbol{\Sigma}\mathbf{U}^T\mathbf{Y}.
\end{aligned}$$

It follows from Assumption 2.1 that for all $i\in[g]$,

$$\tfrac{1}{1+\eta}(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1} \preceq (\boldsymbol{\Sigma}\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{U}\boldsymbol{\Sigma}+n\gamma\mathbf{I}_\rho)^{-1} \preceq \tfrac{1}{1-\eta}(\boldsymbol{\Sigma}^2+n\gamma\mathbf{I}_\rho)^{-1}.$$

We let $\mathbf{B}_i = \mathbf{I}_\rho+\boldsymbol{\Delta}_i$. Thus $-\frac{\eta}{1+\eta}\mathbf{I}_\rho\preceq\boldsymbol{\Delta}_i\preceq\frac{\eta}{1-\eta}\mathbf{I}_\rho$. It follows that

$$\begin{aligned}
(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{1/2}\big(\mathbf{W}^{\mathrm{h}}-\mathbf{W}^\star\big) &= \frac{1}{g}\sum_{i=1}^g(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{1/2}\big(\mathbf{W}_i^{\mathrm{h}}-\mathbf{W}^\star\big) \\
&= \frac{1}{g}\sum_{i=1}^g\mathbf{V}\mathbf{A}_i(\mathbf{I}_\rho+\boldsymbol{\Delta}_i)\mathbf{C} = \frac{1}{g}\sum_{i=1}^g\mathbf{V}\mathbf{A}_i\mathbf{C}+\frac{1}{g}\sum_{i=1}^g\mathbf{V}\mathbf{A}_i\boldsymbol{\Delta}_i\mathbf{C}.
\end{aligned}$$

It follows that

$$\begin{aligned}
&\Big\|(\mathbf{X}^T\mathbf{X}+n\gamma\mathbf{I}_d)^{1/2}\big(\mathbf{W}^{\mathrm{h}}-\mathbf{W}^\star\big)\Big\|_F \le \Big\|\frac{1}{g}\sum_{i=1}^g\mathbf{A}_i\Big\|_2\Big\|\mathbf{C}\Big\|_F+\frac{1}{g}\sum_{i=1}^g\|\mathbf{A}_i\|_2\|\boldsymbol{\Delta}_i\|_2\|\mathbf{C}\|_F \\
&\le \Big\|\frac{1}{g}\sum_{i=1}^g\mathbf{A}_i\Big\|_2\Big\|\mathbf{C}\Big\|_F+\frac{\eta}{1-\eta}\Big(\frac{1}{g}\sum_{i=1}^g\|\mathbf{A}_i\|_2\Big)\|\mathbf{C}\|_F. \qquad (21)
\end{aligned}$$

Let $\mathbf{S} = \frac{1}{g}[\mathbf{S}_1, \cdots, \mathbf{S}_g] \in \mathbb{R}^{n \times gs}$. It follows from the definition of $\mathbf{A}_i$ that

$$\begin{aligned}
\big\|\mathbf{A}_i\big\|_2 &= \Big\|(\boldsymbol{\Sigma}^2 + n\gamma\mathbf{I}_\rho)^{-1/2}\boldsymbol{\Sigma}(\mathbf{I}_\rho - \mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{U})\boldsymbol{\Sigma}(\boldsymbol{\Sigma}^2 + n\gamma\mathbf{I}_\rho)^{-1/2}\Big\|_2 \\
&\leq \eta\Big\|(\boldsymbol{\Sigma}^2 + n\gamma\mathbf{I}_\rho)^{-1/2}\boldsymbol{\Sigma}\boldsymbol{\Sigma}(\boldsymbol{\Sigma}^2 + n\gamma\mathbf{I}_\rho)^{-1/2}\Big\|_2 = \eta\tfrac{\sigma_{\max}^2}{\sigma_{\max}^2 + n\gamma} = \eta\beta, \\
\Big\|\frac{1}{g}\sum_{i=1}^g \mathbf{A}_i\Big\|_2 &= \Big\|(\boldsymbol{\Sigma}^2 + n\gamma\mathbf{I}_\rho)^{-1/2}\boldsymbol{\Sigma}(\mathbf{I}_\rho - \mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U})\boldsymbol{\Sigma}(\boldsymbol{\Sigma}^2 + n\gamma\mathbf{I}_\rho)^{-1/2}\Big\|_2 \\
&\leq \tfrac{\eta}{\sqrt{g}}\big\|(\boldsymbol{\Sigma}^2 + n\gamma\mathbf{I}_\rho)^{-1/2}\boldsymbol{\Sigma}\boldsymbol{\Sigma}(\boldsymbol{\Sigma}^2 + n\gamma\mathbf{I}_\rho)^{-1/2}\big\|_2 = \tfrac{\eta}{\sqrt{g}}\tfrac{\sigma_{\max}^2}{\sigma_{\max}^2 + n\gamma} = \tfrac{\eta\beta}{\sqrt{g}}.
\end{aligned}$$

It follows from (21) that

$$\begin{aligned}
&\Big\|(\mathbf{X}^T\mathbf{X} + n\gamma\mathbf{I}_d)^{1/2}\big(\mathbf{W}^{\mathrm{h}} - \mathbf{W}^\star\big)\Big\|_F \\
&\leq \Big(\tfrac{\eta}{\sqrt{g}} + \tfrac{\eta^2}{1-\eta}\Big)\beta\big\|\mathbf{C}\big\|_F \\
&\leq \Big(\tfrac{\eta}{\sqrt{g}} + \tfrac{\eta^2}{1-\eta}\Big)\beta\sqrt{-nf(\mathbf{W}^\star) + \|\mathbf{Y}\|_F^2},
\end{aligned}$$

where the latter inequality follows from the proof of Theorem 16. This concludes the proof. ■

Appendix E. Model Averaging from the Statistical Perspective: Proofs

In Section E.1 we prove Theorem 22. In Section E.2 we prove Theorem 23.

E.1 Proof of Theorem 22

Proof The bound on $\mathsf{bias}\big(\mathbf{W}^{\mathrm{c}}\big)$ can be shown in the same way as the proof of Theorem 17.

We prove the bound on $\mathsf{var}\big(\mathbf{W}^{\mathrm{c}}\big)$ in the following. It follows from Assumption 2.1 that

$$(1+\eta)^{-1}(\mathbf{I}_\rho + n\gamma\boldsymbol{\Sigma}^{-2})^{-1} \preceq (\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{U} + n\gamma\boldsymbol{\Sigma}^{-2})^\dagger \preceq (1-\eta)^{-1}(\mathbf{I}_\rho + n\gamma\boldsymbol{\Sigma}^{-2})^{-1}.$$

Let

$$(\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{U} + n\gamma\boldsymbol{\Sigma}^{-2})^\dagger = (\mathbf{I}_\rho + n\gamma\boldsymbol{\Sigma}^{-2})^{-1/2}(\mathbf{I}_\rho + \boldsymbol{\Delta}_i)(\mathbf{I}_\rho + n\gamma\boldsymbol{\Sigma}^{-2})^{-1/2}.$$

It holds that

$$-\frac{\eta}{1+\eta}\mathbf{I}_\rho \preceq \boldsymbol{\Delta}_i \preceq \frac{\eta}{1-\eta}\mathbf{I}_\rho.$$

By the definition of $\mathsf{var}(\mathbf{W}^{\mathrm{c}})$ in Theorem 9, we have that

$$\begin{aligned}
&\sqrt{\mathsf{var}\big(\mathbf{W}^{\mathrm{c}}\big)} \\
&= \tfrac{\xi}{\sqrt{n}}\Big\|\frac{1}{g}\sum_{i=1}^g(\mathbf{I}_\rho + n\gamma\boldsymbol{\Sigma}^{-2})^{-1}\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T + \frac{1}{g}\sum_{i=1}^g(\mathbf{I}_\rho + n\gamma\boldsymbol{\Sigma}^{-2})^{-1/2}\boldsymbol{\Delta}_i(\mathbf{I}_\rho + n\gamma\boldsymbol{\Sigma}^{-2})^{-1/2}\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\Big\|_F \\
&\leq \tfrac{\xi}{\sqrt{n}}\Big(\Big\|(\mathbf{I}_\rho + n\gamma\boldsymbol{\Sigma}^{-2})^{-1}\mathbf{U}^T\mathbf{S}\mathbf{S}^T\Big\|_F + \frac{1}{g}\sum_{i=1}^g\Big\|(\mathbf{I}_\rho + n\gamma\boldsymbol{\Sigma}^{-2})^{-1/2}\boldsymbol{\Delta}_i(\mathbf{I}_\rho + n\gamma\boldsymbol{\Sigma}^{-2})^{-1/2}\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\Big\|_F\Big) \\
&\leq \tfrac{\xi}{\sqrt{n}}\big\|(\mathbf{I}_\rho + n\gamma\boldsymbol{\Sigma}^{-2})^{-1}\big\|_F\Big(\big\|\mathbf{U}^T\mathbf{S}\big\|_2\big\|\mathbf{S}\big\|_2 + \frac{\eta}{1-\eta}\frac{1}{g}\sum_{i=1}^g\big\|\mathbf{U}^T\mathbf{S}_i\big\|_2\big\|\mathbf{S}_i\big\|_2\Big) \\
&= \sqrt{\mathsf{var}\big(\mathbf{W}^\star\big)}\Big(\big\|\mathbf{U}^T\mathbf{S}\big\|_2\big\|\mathbf{S}\big\|_2 + \frac{\eta}{1-\eta}\frac{1}{g}\sum_{i=1}^g\big\|\mathbf{U}^T\mathbf{S}_i\big\|_2\big\|\mathbf{S}_i\big\|_2\Big).
\end{aligned}$$

Under Assumption 2.1, we have that $\|\mathbf{S}_i^T\mathbf{U}\|_2^2 \leq 1+\eta$ and $\|\mathbf{S}^T\mathbf{U}\|_2^2 \leq 1 + \frac{\eta}{\sqrt{g}}$. It follows that

$$\sqrt{\frac{\mathsf{var}\big(\mathbf{W}^{\mathrm{c}}\big)}{\mathsf{var}\big(\mathbf{W}^\star\big)}} \leq \sqrt{1 + \frac{\eta}{\sqrt{g}}}\big\|\mathbf{S}\big\|_2 + \frac{\eta\sqrt{1+\eta}}{1-\eta}\frac{1}{g}\sum_{i=1}^g\big\|\mathbf{S}_i\big\|_2.$$

Now the desired result follows from Assumption 2.3. ■

E.2 Proof of Theorem 23

Proof The bound on $\mathsf{var}\big(\mathbf{W}^{\mathrm{h}}\big)$ can be established in the same way as Theorem 19.

We prove the bound on $\mathsf{bias}\big(\mathbf{W}^{\mathrm{h}}\big)$ in the following. Let

$$(\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{U} + n\gamma\boldsymbol{\Sigma}^{-2})^\dagger = (\mathbf{I}_\rho + n\gamma\boldsymbol{\Sigma}^{-2})^{-1/2}(\mathbf{I}_\rho + \boldsymbol{\Delta}_i)(\mathbf{I}_\rho + n\gamma\boldsymbol{\Sigma}^{-2})^{-1/2}.$$

Under Assumption 2.1, we have that $\boldsymbol{\Delta}_i \preceq \frac{\eta}{1-\eta}\mathbf{I}_\rho$. It follows from Theorem 9 that

$$\begin{aligned}
\mathsf{bias}\big(\mathbf{W}^{\mathrm{h}}\big) &= \gamma\sqrt{n}\Big\|\frac{1}{g}\sum_{i=1}^g\Big(\boldsymbol{\Sigma}^{-2} + \tfrac{\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{U} - \mathbf{I}_\rho}{n\gamma}\Big)(\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{U} + n\gamma\boldsymbol{\Sigma}^{-2})^\dagger\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{W}_0\Big\|_F \\
&\leq \gamma\sqrt{n}\Big\|\frac{1}{g}\sum_{i=1}^g\Big(\boldsymbol{\Sigma}^{-2} + \tfrac{\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{U} - \mathbf{I}_\rho}{n\gamma}\Big)(\mathbf{I}_\rho + n\gamma\boldsymbol{\Sigma}^{-2})^{-1}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{W}_0\Big\|_F \\
&\quad + \gamma\sqrt{n}\Big\|\frac{1}{g}\sum_{i=1}^g\Big(\boldsymbol{\Sigma}^{-2} + \tfrac{\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{U} - \mathbf{I}_\rho}{n\gamma}\Big)(\mathbf{I}_\rho + n\gamma\boldsymbol{\Sigma}^{-2})^{-1/2}\boldsymbol{\Delta}_i(\mathbf{I}_\rho + n\gamma\boldsymbol{\Sigma}^{-2})^{-1/2}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{W}_0\Big\|_F \\
&\triangleq \gamma\sqrt{n}(A+B),
\end{aligned}$$

where

$$\begin{aligned} A &= \left\| \frac{1}{g}\sum_{i=1}^{g} \left(\boldsymbol{\Sigma}^{-2} + \tfrac{\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{U}-\mathbf{I}_\rho}{n\gamma}\right)\left(\mathbf{I}_\rho + n\gamma\boldsymbol{\Sigma}^{-2}\right)^{-1}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{W}_0 \right\|_F \\ &= \left\| \left(\boldsymbol{\Sigma}^{-2} + \tfrac{\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}-\mathbf{I}_\rho}{n\gamma}\right)\left(\mathbf{I}_\rho + n\gamma\boldsymbol{\Sigma}^{-2}\right)^{-1}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{W}_0 \right\|_F, \\ B &= \left\| \frac{1}{g}\sum_{i=1}^{g} \left(\boldsymbol{\Sigma}^{-2} + \tfrac{\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{U}-\mathbf{I}_\rho}{n\gamma}\right)\left(\mathbf{I}_\rho + n\gamma\boldsymbol{\Sigma}^{-2}\right)^{-1/2}\boldsymbol{\Delta}_i\left(\mathbf{I}_\rho + n\gamma\boldsymbol{\Sigma}^{-2}\right)^{-1/2}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{W}_0 \right\|_F \\ &\leq \frac{1}{g}\sum_{i=1}^{g} \left\| \left(\boldsymbol{\Sigma}^{-2} + \tfrac{\mathbf{U}^T\mathbf{S}_i\mathbf{S}_i^T\mathbf{U}-\mathbf{I}_\rho}{n\gamma}\right)\left(\mathbf{I}_\rho + n\gamma\boldsymbol{\Sigma}^{-2}\right)^{-1/2}\boldsymbol{\Delta}_i\left(\mathbf{I}_\rho + n\gamma\boldsymbol{\Sigma}^{-2}\right)^{-1/2}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{W}_0 \right\|_F. \end{aligned}$$

It follows from Assumption 2.1 that $\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}-\mathbf{I}_\rho$ is semidefinitely bounded between $\pm\frac{\eta}{\sqrt{g}}\mathbf{I}_\rho$. Thus

$$\left(1 - \tfrac{\eta\sigma_{\max}^2}{n\gamma\sqrt{g}}\right)\boldsymbol{\Sigma}^{-2} \preceq \boldsymbol{\Sigma}^{-2} + \tfrac{\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}-\mathbf{I}_\rho}{n\gamma} \preceq \left(1 + \tfrac{\eta\sigma_{\max}^2}{n\gamma\sqrt{g}}\right)\boldsymbol{\Sigma}^{-2}.$$

It follows that

$$\begin{aligned} A &= \left\| \left(\boldsymbol{\Sigma}^{-2} + \tfrac{\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}-\mathbf{I}_\rho}{n\gamma}\right)\left(\mathbf{I}_\rho + n\gamma\boldsymbol{\Sigma}^{-2}\right)^{-1}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{W}_0 \right\|_F \\ &\leq \left(1 + \tfrac{\eta\sigma_{\max}^2}{n\gamma\sqrt{g}}\right)\left\| \left(\boldsymbol{\Sigma}^2 + n\gamma\mathbf{I}_\rho\right)^{-1}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{W}_0 \right\|_F. \end{aligned}$$

Similar to the proof of Theorem 19, we can show that

$$\begin{aligned} B &\leq \left(1 + \frac{\eta\sigma_{\max}^2}{n\gamma}\right)\cdot\frac{1}{g}\sum_{i=1}^{g}\left\|\boldsymbol{\Sigma}^{-2}\left(\mathbf{I}_\rho + n\gamma\boldsymbol{\Sigma}^{-2}\right)^{-1/2}\boldsymbol{\Delta}_i\left(\mathbf{I}_\rho + n\gamma\boldsymbol{\Sigma}^{-2}\right)^{-1/2}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{W}_0\right\|_F \\ &\leq \frac{\eta}{1-\eta}\left(1 + \frac{\eta\sigma_{\max}^2}{n\gamma}\right)\cdot\left\|\left(\boldsymbol{\Sigma}^2 + n\gamma\mathbf{I}_\rho\right)^{-1}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{W}_0\right\|_F. \end{aligned}$$

Hence

$$\begin{aligned} \mathsf{bias}\left(\mathbf{W}^{\mathrm{h}}\right) &\leq \gamma\sqrt{n}(A+B) \\ &\leq \left[\tfrac{1}{1-\eta} + \left(\tfrac{\eta}{\sqrt{g}} + \tfrac{\eta^2}{1-\eta}\right)\tfrac{\sigma_{\max}^2}{n\gamma}\right]\gamma\sqrt{n}\left\|\left(\boldsymbol{\Sigma}^2 + n\gamma\mathbf{I}_\rho\right)^{-1}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{W}_0\right\|_F \\ &= \left[\tfrac{1}{1-\eta} + \left(\tfrac{\eta}{\sqrt{g}} + \tfrac{\eta^2}{1-\eta}\right)\tfrac{\sigma_{\max}^2}{n\gamma}\right]\mathsf{bias}\left(\mathbf{W}^\star\right). \end{aligned}$$

Here the equality follows from Theorem 4. ■

References

Haim Avron, Kenneth L. Clarkson, and David P. Woodruff. Sharper Bounds for Regularized Data Fitting. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 81, pages 27:1–27:22, Dagstuhl, Germany, 2017. Schloss Dagstuhl.

Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding Frequent Items in Data Streams. *Theoretical Computer Science*, 312(1):3–15, 2004.

Kenneth L. Clarkson and David P. Woodruff. Low Rank Approximation and Regression in Input Sparsity Time. In *Annual ACM Symposium on Theory of Computing (STOC)*, 2013.

Michal Derezinski and Manfred K. Warmuth. Unbiased estimates for linear regression via volume sampling. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

Michal Derezinski and Manfred K. Warmuth. Subsampling for ridge regression via regularized volume sampling. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

Petros Drineas and Michael W. Mahoney. RandNLA: Randomized Numerical Linear Algebra. *Communications of the ACM*, 59(6):80–90, 2016.

Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast Monte Carlo Algorithms for Matrices I: Approximating Matrix Multiplication. *SIAM Journal on Computing*, 36(1): 132–157, 2006a.

Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Sampling Algorithms for ℓ_2 Regression and Applications. In *Annual ACM-SIAM Symposium on Discrete Algorithm (SODA)*, 2006b.

Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Relative-Error CUR Matrix Decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2):844–881, September 2008.

Petros Drineas, Michael W. Mahoney, S. Muthukrishnan, and Tamás Sarlós. Faster Least Squares Approximation. *Numerische Mathematik*, 117(2):219–249, 2011.

Petros Drineas, Malik Magdon-Ismail, Michael W. Mahoney, and David P. Woodruff. Fast Approximation of Matrix Coherence and Statistical Leverage. *Journal of Machine Learning Research*, 13:3441–3472, 2012.

Alex Gittens. The Spectral Norm Error of the Naive Nyström Extension. *arXiv preprint arXiv:1110.5305*, 2011.

Daniel Hsu, Sham M. Kakade, and Tong Zhang. Random design analysis of ridge regression. *Foundations of Computational Mathematics*, 14(3):569–600, 2014.

William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26(189-206), 1984.

Yichao Lu, Paramveer Dhillon, Dean P. Foster, and Lyle Ungar. Faster Ridge Regression via the Subsampled Randomized Hadamard Transform. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

Ping Ma, Michael W. Mahoney, and Bin Yu. A Statistical Perspective on Algorithmic Leveraging. *Journal of Machine Learning Research*, 16(1):861–911, 2015.

Michael W. Mahoney. Randomized Algorithms for Matrices and Data. *Foundations and Trends in Machine Learning*, 3(2):123–224, 2011.

Xiangrui Meng and Michael W. Mahoney. Low-Distortion Subspace Embeddings in Input-Sparsity Time and Applications to Robust Linear Regression. In *Annual ACM Symposium on Theory of Computing (STOC)*, 2013.

John Nelson and Huy L. Nguyên. OSNAP: Faster Numerical Linear Algebra Algorithms via Sparser Subspace Embeddings. In *IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, 2013.

Mihai Patrascu and Mikkel Thorup. The Power of Simple Tabulation-Based Hashing. *Journal of the ACM*, 59(3), 2012.

Ninh Pham and Rasmus Pagh. Fast and Scalable Polynomial Kernels via Explicit Feature Maps. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2013.

Mert Pilanci and Martin J. Wainwright. Iterative Hessian Sketch: Fast and Accurate Solution Approximation for Constrained Least-Squares. *Journal of Machine Learning Research*, pages 1–33, 2015.

Garvesh Raskutti and Michael W. Mahoney. A Statistical Perspective on Randomized Sketching for Ordinary Least-Squares. *Journal of Machine Learning Research*, 17(214): 1–31, 2016.

Gian-Andrea Thanei, Christina Heinze, and Nicolai Meinshausen. Random Projections For Large-Scale Regression. In *Big and Complex Data Analysis*. Springer, 2017.

Joel A. Tropp. Improved Analysis of the Subsampled Randomized Hadamard Transform. *Advances in Adaptive Data Analysis*, 3(01n02):115–126, 2011.

Stephen Tu, Rebecca Roelofs, Shivaram Venkataraman, and Benjamin Recht. Large Scale Kernel Learning using Block Coordinate Descent. *arXiv preprint arXiv:1602.05310*, 2016.

Roman Vershynin. *Introduction to the Non-Asymptotic Analysis of Random Matrices*, pages 210–268. Cambridge University Press, 2012.

Jialei Wang, Jason D. Lee, Mehrdad Mahdavi, Mladen Kolar, and Nathan Srebro. Sketching Meets Random Projection in the Dual: a Provable Recovery Algorithm for Big and High-Dimensional Data. *Electronic Journal of Statistics*, 11(2):4896–4944, 2017a.

Shusen Wang, Luo Luo, and Zhihua Zhang. SPSD Matrix Approximation via Column Selection: Theories, Algorithms, and Extensions. *Journal of Machine Learning Research*, 17(49):1–49, 2016a.

Shusen Wang, Zhihua Zhang, and Tong Zhang. Towards More Efficient SPSD Matrix Approximation and CUR Matrix Decomposition. *Journal of Machine Learning Research*, 17(210):1–49, 2016b.

Shusen Wang, Farbod Roosta-Khorasani, Peng Xu, and Michael W. Mahoney. GIANT: Globally Improved Approximate Newton Method for Distributed Optimization. *arXiv preprint arXiv:1709.03528*, 2017b.

Yining Wang, Adams Wei Yu, and Aarti Singh. On computationally tractable selection of experiments in measurement-constrained regression models. *Journal of Machine Learning Research*, 18(143):1–41, 2017c.

Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. Feature Hashing for Large Scale Multitask Learning. In *International Conference on Machine Learning (ICML)*, 2009.

David P. Woodruff. Sketching as a Tool for Numerical Linear Algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1–2):1–157, 2014.

Franco Woolfe, Edo Liberty, Vladimir Rokhlin, and Mark Tygert. A Fast Randomized Algorithm for the Approximation of Matrices. *Applied and Computational Harmonic Analysis*, 25(3):335–366, 2008.

Jiyan Yang, Xiangrui Meng, and Michael W. Mahoney. Implementing Randomized Matrix Algorithms in Parallel and Distributed Environments. *Proceedings of the IEEE*, 104(1): 58–92, 2016.

Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster Computing with Working Sets. *HotCloud*, 10(10-10):95, 2010.

Yuchen Zhang, John C. Duchi, and Martin J. Wainwright. Communication-Efficient Algorithms for Statistical Optimization. *Journal of Machine Learning Research*, 14:3321–3363, 2013.

Yuchen Zhang, John C. Duchi, and Martin J. Wainwright. Divide and Conquer Kernel Ridge Regression: a Distributed Algorithm with Minimax Optimal Rates. *Journal of Machine Learning Research*, 16:3299–3340, 2015.

Journal of Machine Learning Research 18 (2018) 1-43 Submitted 3/16; Revised 1/18; Published 4/18

Compact Convex Projections

Steffen Grünewälder S.GRUNEWALDER@LANCASTER.AC.UK
Department of Mathematics and Statistics
Lancaster University
Lancaster, England

Editor: Mark Schmidt

Abstract

We study the usefulness of conditional gradient like methods for determining projections onto convex sets, in particular, projections onto naturally arising convex sets in reproducing kernel Hilbert spaces. Our work is motivated by the recently introduced kernel herding algorithm which is closely related to the Conditional Gradient Method (CGM). It is known that the herding algorithm converges with a rate of $1/t$, where t counts the number of iterations, when a point in the interior of a convex set is approximated. We generalize this result and we provide a necessary and sufficient condition for the algorithm to approximate projections with a rate of $1/t$. The CGM, which is in general vastly superior to the herding algorithm, achieves only an inferior rate of $1/\sqrt{t}$ in this setting. We study the usefulness of such projection algorithms further by exploring ways to use these for solving concrete machine learning problems. In particular, we derive non-parametric regression algorithms which use at their core a slightly modified kernel herding algorithm to determine projections. We derive bounds to control approximation errors of these methods and we demonstrate via experiments that the developed regressors are en-par with state-of-the-art regression algorithms for large scale problems.

1. Introduction

Convex sets and projections onto convex sets are omnipresent in Machine Learning and Statistics. Projections appear already in the most basic approaches like in ordinary least squares regression where the the estimate can be interpreted as the projection of some target vector onto a linear subspace. By adding constraints one arrives naturally at a convex projection problem. Similarly, the ridge regressor and Gaussian process regressor are closely related to projections onto balls and the lasso estimator (Tibshirani, 1996) corresponds to a projection onto a simplex (again a convex set). Regularization approaches with convex constraints are, in general, closely related to the problem of determining a projection. For example, sparsity constraints are often enforced by optimizing over the standard simplex and algorithms that determine projections onto the simplex have been studied extensively (Duchi, Shalev-Shwartz, Singer, and Chandra, 2008).

This paper is about exploring the usefulness of some closely related algorithms for determining projections onto convex sets in the large data context. Our inspiration for this line of research dates back to the paper by Welling (2009) in which an algorithm, called the herding algorithm, has been introduced which computes compact representations of probability measures. By a compact representation we mean here a representation that is supported on few points of the sample space and that can be used to calculate efficiently expectations of functions. The algorithm has garnered attention in recent years and various generalizations of the method have been explored. In one of these works it has been studied how the algorithm behaves in a non-parametric setting where a kernel function is defined on the sample space (Chen, Welling, and Smola, 2010). The authors analyzed the interplay between the algorithm and the reproducing kernel Hilbert space (RKHS, Aronszajn (1950)) associated with the kernel function. Their main finding has been that the algorithm has a guaranteed rate of convergence towards the given probability measure (or, more accurately, towards its representer in the RKHS) of $1/t$, where t is the number of iterations for which the algorithm is run. The rate of $1/t$ is an improvement over randomly selected support points, which would guarantee a rate of $1/\sqrt{t}$, but the rate is slow compared to what many other numerical algorithms yield. It is worth pointing out that we consider here convergence in $\|\cdot\|$, while in the literature results are often reported for quadratic objectives like $\|\cdot\|^2$. The virtue of this algorithm is that the computational costs per iteration are linear in the sample size. These low costs in dependence of the sample size make this algorithm, and related algorithms, attractive in the large data context where algorithms with faster rates of convergence are often prohibitively expensive to compute.

The herding algorithm is iterative and uses at its core an inner product between the approximation error $y - p_t$ at iteration t and a set of candidates S that can be chosen to reduce the error. Here, y lies inside a convex set C, p_t is the approximation of y at iteration t and S is a subset of C. The algorithm selects elements which maximize this inner product $\langle y - p_t, x\rangle$ over $x \in S$. We asked ourselves somewhat naïvely what would happen if y lies outside of C. One would hope that the algorithm converges in this case to the point in C that is closest to y, or in other words, that it would converge to the projection Py of y onto C. The first observation we had was that this will, in fact, be true if $y - Py$ stands orthogonal on the (affine) subspace spanned by C and if Py lies in the (affine) interior of C, because in this case the algorithm is completely unaffected by $y - Py$ and it behaves equivalently to the case where we would apply it directly to $Py \in C$. In this setting, the standard guarantees tell us that the algorithm converges with a rate of $1/t$ to the projection. Our initial observation was in a way naïve since the literature on herding provides a way to show that the algorithm with an added line search converges to the projection. The argument to verify the convergence is based on the observation from Bach, Lacoste-Julien, and Obozinski (2012) that the herding algorithm with an added line search is equivalent to a well known method called the conditional gradient method (CGM, Frank and Wolfe (1956)). Basic guarantees on the convergence of the CGM imply directly that the method converges with a rate of $1/\sqrt{t}$ to the projection (see our Literature Section below for more details). This convergence result is reaffirming, however, the rate of convergence the result guarantees is a slow rate of $1/\sqrt{t}$ and not $1/t$.

The rate of $1/\sqrt{t}$ is, in fact, the actual rate with which CGM converges in non-trivial projection problems. The reason for this is that, in interesting cases, projections lie in the boundary of the convex set and CGM itself achieves only in exceptional cases a rate that is better than $1/\sqrt{t}$ if the solution of an optimization problem lies in the boundary (Cannon and Cullum, 1968). In recent years extensions of the basic CGM algorithm have become popular and it was shown that a linear rate of convergence (a rate significantly better than $1/t$) is achieved by a particular extension, the CGM with away steps algorithm, even if the solution lies in the boundary (Lacoste-Julien and Jaggi, 2015). This is a significant result, but it comes with a caveat: we gain strong guarantees only if the convex set has a large pyramidal width. For instance, the d dimensional cube has a large pyramidal width and the authors provide an upper bound on the reduction of the approximation error in the order of $1/d^2$ per step. The convex sets we are mainly interested in are of the form $C = \{k(x,\cdot) : x \in X\} \subset \mathcal{H}$, where k is the reproducing kernel of the RKHS $\mathcal{H}$. Here, the dimension of the spanned subspace is often equal to the sample size (when using a Gaussian kernel,

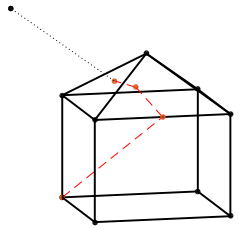


Figure 1: An illustration of the application of the herding algorithm to a projection problem. The aim is to find the projection of a ray starting at the black dot onto the house. The house is compact and convex. The herding algorithm finds for such sets approximations of the projection with an accuracy in the order of $1/t$, where t is the number of iterations. The red dots show the approximation after 0, 2, 5 and 100 steps with an initial estimate that equals the North-West corner on the bottom of the house.

for example). Furthermore, C is significantly more complicated than a cube and we expect that the upper bound gives us guarantees on the reduction that are significantly worse than $1/n^2$ per iteration, where n is the size of the sample. In the large data context, where $n \geq 10^5$, these reductions per step are tiny. Another recent approach to improve the convergence behavior of the CGM has been developed in Garber and Hazan (2013). Standard CGM is in this approach combined with what the authors call a Local Linear Optimization Oracle (LLOO) and it is shown that a linear rate of convergence is achieved by their method. Like in the away step approach the geometry of the convex set factors into the performance. For the method from Garber and Hazan (2013) the geometry affects the run-time of the LLOO and, hence, the run-time of the CGM with the added LLOO. The approach works well for simple shaped convex sets like the simplex but becomes difficult when C has no particular structure as in our RKHS setting. Understanding better the relevant geometric properties of C with respect to the reproducing kernel is an intriguing problem, but it is beyond the aims of this paper. We aim here for a deeper understanding of the core algorithms and their interplay with the projection problem in Hilbert space. The following figure visualizes the relation between the different CGM like methods. We focus in this paper on the methods in the shaded area.

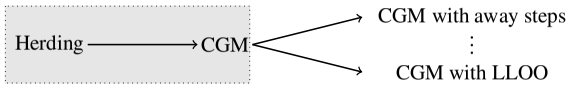


Figure 2: The relation between various CGM like methods.

CGM is known to converge with a linear rate if there exists a ball around the solution inside the convex set (this is also known as Slater's condition, Beck and Teboulle (2004)). Similarly, for the herding algorithm it has been shown that this very same condition guarantees a rate of $1/t$ (Chen et al., 2010). To the best of our knowledge, both convergence results have been derived

independently of each other and it is telling that in both approaches the same assumption plays a fundamental role (the existence of a ball around the optimum). The assumption appears naturally in the analyses since it allows a strong reduction of estimation errors per iteration independently of the direction in which the errors point. If this assumption does not hold then there are directions which pose problems, i.e. if an error builds up in such a direction then these errors will only decay slowly over time. This is why solutions that lie in the boundary pose problems: errors that point away from the convex set can not be easily reduced. We refine the convergence argument to be able to deal with such errors. On a high level our approach can be described in the following way: consider a convex set with finitely many extremes then each point in the boundary is in the relative interior of a face of the original convex set. For example, in Figure 1 our convex set is the house and we want to compute the projection of the black dot onto the house. The dotted line visualizes the process of projecting onto the house and the red dot at the end of the line is the projection that we want to determine algorithmically. The projection does not lie in the interior of the house but it lies inside a face of the convex set (the triangle that contains the projection). A face of a convex set is in turn a convex set and if we would apply the algorithm directly to this face then we could show fast convergence. In general, we can not identify the face that contains the projection without using significant computational resources. However, we can treat elements outside of this face, which are chosen by the algorithm, as perturbations that disturb the behavior of the algorithm. For instance, we applied the herding algorithm to the problem in Figure 1 and we used an initial approximation of the projection which corresponds to the North-West corner on the bottom of the house. The dashed line in red shows how the approximation evolves over multiple steps and how the error that is introduced by the initialization shrinks over time. Our line of attack in the theoretical part of this paper is to control such perturbations and to show that the rate of convergence of the herding algorithm is not negatively affected by these.

We were also interested in understanding how CGM like algorithms to determine projections in Hilbert spaces can be exploited in statistical problems. A blueprint is given in the paper by Chen et al. (2010) where the herding algorithm is applied in an RKHS $\mathcal{H}$ to approximate elements inside a compact and convex set $C \subset \mathcal{H}$. It is natural to consider extensions in which C is modified in a suitable way to represent a space of solutions of certain statistical problems. We showcase this approach in the regression problem. In this case, C is the space of regression functions. The projection approach itself is generic and one can gain with relative ease algorithms that infer kernel regressors from data. We call these CCP-regressors where the acronym CCP stands for compact convex projection. The advantage of this approach is that the corresponding algorithms are cheap to compute and they are applicable in the large data context. We find in experiments that the algorithms we derive are en-par with established kernel regression algorithms like the Gaussian process or the fast kernel ridge regressor (FastKRR, Zhang, Duchi, and Wainwright (2013)).

1.1 Literature

We aim in this section for a short overview summarizing key results that put our work in perspective. We start by stating the herding algorithm for approximating a point x^* inside a convex set C that is induced by the finite set of points S, i.e. $C = \operatorname{cch} S$, where cch denotes the closed convex hull operator. C is typically a subset of $\mathbb{R}^d$ equipped with the Euclidean inner product or some other Hilbert space. Given the starting value $w_0 = x^*$ and the initialization $t = 1$ the herding algorithm iterates

1. choose $x_t \in \arg\max_{x \in S} \langle w_{t-1}, x \rangle$,
2. set $w_t = w_{t-1} - (x_t - x^*)$,
3. set $t \leftarrow t + 1$.

This basic routine is combined with some termination criterion like running the algorithm for T iterations. The approximation of x^* is then $(x_1 + \ldots + x_T)/T$ where T denotes here the number of iterations the algorithm was run for.

The herding algorithm is a special case of what is called the *conditional gradient method (CGM)* or the *Frank-Wolfe algorithm* (Frank and Wolfe, 1956; Bach et al., 2012). The conditional gradient method is known since the fifties and various results about its convergence behavior have been derived throughout the years. The CGM method tries to approximate the minimal value of a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ on a set C by iterating two steps after an initialization with some value $x_0 \in C$:

1. Choose an $x^\star \in \arg\min_{x \in C} \langle x - x_{t-1}, \nabla f(x_{t-1}) \rangle$,
2. perform a line search over $\lambda \in [0, 1]$, i.e. choose a

$$\lambda^\star \in \arg\min_{\lambda \in [0,1]} f(x_{t-1} + \lambda(x^\star - x_{t-1}))$$

and update $x_t = x_{t-1} + \lambda^\star(x^\star - x_{t-1})$.

Standard results for the CGM method hold for continuously differentiable functions on convex compact subsets C of $\mathbb{R}^d$ (Beck and Teboulle, 2004). Let $f^* = \min_{x \in C} f(x)$ then it is known that a sequence $\{x_t\}_{t \geq 0}$ produced by the CGM attains the minimum eventually, i.e. $\lim_{t \to \infty} f(x_t) = f^*$ and there exists a constant b such that $f(x_t) - f^* \leq b/t$. A differentiable function on a compact set is Lipschitz continuous and the constant b depends on the Lipschitz constant and on the diameter of the set C, $\operatorname{diam}(C) = \sup_{x,y \in C} \|x - y\|$. We can use the CGM to find the projection of z onto C. Letting $f(x) = \|x - z\|^2$, with $\|\cdot\|$ the Euclidean norm, we can observe that f is continuously differentiable with derivative $2\langle x - z, \cdot \rangle$ and we are assured that $\|x_t - z\|^2 - \min_{x \in C} \|x - z\|^2 \leq b/t$ for some constant b and all $t \geq 0$. Observe that this rate is slower then what we aim for since we gain here a rate of about $t^{-1/2}$ for the convergence of $\|x_t - z\|$. The norm itself is not differentiable and we can not derive the faster rate of $1/t$ through these results.

It is also known that the rate of convergence can not be improved in general (Cannon and Cullum, 1968)[Thm. 2]. Hence, stronger assumptions are needed to gain faster rates of convergence. One typical assumption is that *Slater's condition* holds. Slater's condition is in our context the assumption that $z \in \operatorname{int} C$, where $\operatorname{int} C$ denotes the interior of C. Proposition 3.2 in Beck and Teboulle (2004) demonstrates that if Slater's condition is fulfilled then the CGM produces a sequence $\{x_t\}_{t \geq 0}$ which satisfies

$$\|x_t - z\| \leq be^{-tq/2}$$

for some positive constants b and q. The rate is significantly faster than what is known for the herding algorithm but the result is not applicable to the problem of computing projections since (non-trivial) projections lie in the boundary of C and not in the interior. These convergence results are nevertheless fundamental and they are a corner stone for various studies. For instance, in Jaggi (2013) these convergence results for the CGM are combined with a duality argument to give bounds

on the duality gap between the primal and dual solution of the optimization problem. Another line of research considers extensions of the CGM which are build to circumvent the sub-linear rate of convergence. We discussed already the most prominent approaches: the classical approach from Wolfe (1970) in which away steps are added to the CGM and the approach from Garber and Hazan (2013) which both guarantee a linear rate of convergence.

Much of the research on the CGM has been done in the general context where an arbitrary continuously differentiable function on a convex set is optimized. However, the more specialized problem of determining projections onto convex sets has also garnered attention. Especially, in the context of sparsity. The convex set C that is typically considered in this context is the standard simplex in $\mathbb{R}^d$ which is $\Delta^{d-1} = \operatorname{cch}\{e_n : 1 \leq n \leq d\}$, where $e_1, \ldots, e_d$ is the standard basis in $\mathbb{R}^d$. The simplex Δ^{d-1} is of particular importance because ℓ_1-constraints can be naturally expressed through it: $w \in \mathbb{R}^d_+$ with $\|w\|_{\ell_1} = 1$ implies that $w = \sum_{i=1}^d \langle w, e_i \rangle e_i \in \operatorname{cch}\{e_n : 1 \leq n \leq d\} = \Delta^{d-1}$ (and vice versa). The CGM acting on the simplex can be used to find sparse solutions for a variety of problems. Most notably the *lasso estimator* (Tibshirani, 1996) can be found through an application of the CGM to a simplex (Clarkson, 2010). In Duchi et al. (2008) a similar problem is studied. The motivation are there algorithms which require projections onto a simplex. The authors develop a projection algorithm that can be computed in the order of d operations on average (the algorithm is stochastic) where d is the dimension of the simplex. This resembles the order of run time of the CGM.

Another set of important results for the CGM rely on strong convexity assumptions about the function f and the set C, see for exampleGarber and Hazan (2015) and references therein. Under such assumptions one can obtain a rate of $1/t$ for finding the projection on C, i.e. the rate of convergence applies independently of where in C the solution lies. The boundary of a strongly convex set C bends significantly. In our paper we assume very different properties of C. We are interested in the convex hull C of a finite set S and faces of such sets C are flat, i.e. they do not bend.

Closely related to the closed convex hull of a set S is the minimum enclosing ball of S (MEB, Clarkson (2010)). The difference between the two is that an Euclidean ball bends (the Euclidean norm is uniformly convex) while the closed convex hull interpolates the boundaries through hyperplanes. For example, the MEB of $S = \{\pm e_1, \ldots, \pm e_d\}$ is the Euclidean unit ball while $\operatorname{cch} S$ is an ℓ_1-ball (a diamond). There exist efficient algorithms for determining the MEB under different conditions. Furthermore, a variety of machine learning problems can be rephrased as the problem of finding an MEB (support vector algorithms are approached in this way in Tsang et al. (2005a,b)).

1.2 Contributions

In this paper we study the usefulness of conditional gradient methods to determine projections onto compact convex sets in Hilbert spaces. The contributions split naturally into theoretical and applied contributions through which we explore this theme. The theoretical contributions are *two theorems*. Our *first theorem* shows that the herding algorithm retains its rate of convergence of $1/t$ if, and only if, the perturbations introduced through points outside the minimal face are coupled in a certain way. The approach we use (considering the minimal face to which well-known results can be easily adapted and controlling perturbations which are introduced by elements outside of the minimal face) is to the best of our knowledge new and we expect that this line of reasoning can be extended in the future to study more advanced methods like the CGM with away steps.

Our *second theorem* analyses why the standard CGM fares worse in this setting than the herding algorithm. On the applied side we show how projections onto compact convex sets in kernel space can be exploited to develop novel machine learning algorithms. We demonstrate this by developing a *new and fast kernel regressor* that is en-par with state-of-the-art non-parametric regressors. We provide *approximation error bounds* for the method by means of a dual approach and we study its performance in experiments. The approximation error bound is a novel modification of an approach from Wolfe (1976) which sacrifices tightness in favor of a significantly reduced computation time.

2. Compact Convex Projections

We use the following algorithm to find the projection of $y \in \mathcal{H}$ onto the compact convex set $C \subset \mathcal{H}$ in terms of a set of support points from the compact set $S \subseteq C$ which contains the extremes of C. In the following we denote the extremes of C with $\operatorname{ex} C$, i.e. $\operatorname{ex} C = \{x : x \in C,$ there exists no $y, z \in C, \alpha \in (0,1)$ such that $x = \alpha y + (1-\alpha)z\}$.

Algorithm 1. Input: $y \in \mathcal{H}$, $T \in \mathbb{N}$, and $S \subset \mathcal{H}$.

1. [Initialize] Set $w_0 = y, t = 1$.
2. [Optimization oracle] Choose $x_t \in \arg\max_{x \in S} \langle w_{t-1}, x \rangle$.
3. [Update weight] Set $w_t = w_{t-1} - (x_t - y)$.
4. [Iterate] If $t < T$ then increment t by 1 and go back to **2**.
5. [Terminate] Return the approximation $(x_1 + \ldots + x_T)/T$.

A maximizer exists in Step 2 because we search for a maximum of a continuous function on a compact set. If there are multiple maximizer then we can choose an arbitrary one, for instance, we can enumerate S and choose the maximizer with the lowest index. This is the herding algorithm as stated in Section 1.1 with the only modification being that it is applied to y which can lie outside of C. The weight at step t is essentially the approximation error scaled up by t because $w_t = (t+1)y - (x_1 + \ldots + x_t) \approx t(y - p_t)$, where $p_t = (x_1 + \ldots + x_t)/t$ is the approximation. We can also add a line search to the optimization:

Algorithm 1 (ls). Input: $y \in \mathcal{H}$, $T \in \mathbb{N}$, and $S \subset \mathcal{H}$.

1. [Initialize] Set $w_0 = y, t = 1$.
2. [Optimization oracle] Choose $x_t \in \arg\max_{x \in S} \langle w_{t-1}, x \rangle$.
3. [Line search] Calculate $\tilde{\alpha}_t = \langle w_{t-1}, w_{t-1} + (x_t - y) \rangle \,/\, \|w_{t-1} + (x_t - y)\|^2$.
4. [Line search] If $t = 1$ set $\alpha_t = 1$, otherwise set $\alpha_t = 1 \wedge \tilde{\alpha}_t$.
5. [Update weight] Set $w_t = (1 - \alpha_t) w_{t-1} - \alpha_t (x_t - y)$.
6. [Iterate] If $t < T$ then increment t by 1 and go back to **2**.
7. [Terminate] Set $\beta_i = \alpha_i \prod_{j=i+1}^{T} (1 - \alpha_j)$ for all $i \leq T$ and return the approximation $\beta_1 x_1 + \ldots + \beta_T x_T$.

We use here the convention that $0/0 = 1$ which implies that $\alpha_t = 1$ if $x_t = p_t$. The weight in the line search algorithm is similar to the weight of the herding algorithm scaled down by $1/t$, i.e. the weight in the line search algorithm is $w_t = y - (\beta_1^t x_1 + \ldots + \beta_t^t x_t)$ where $\beta_i^t = \alpha_i \prod_{j=i+1}^{t} (1 - \alpha_j)$. The choice $\tilde{\alpha}_t = \langle w_{t-1}, w_{t-1} + (x_t - y) \rangle \,/\, \|w_{t-1} + (x_t - y)\|^2$ minimizes $\|w_t\|$ over all choices of $\tilde{\alpha}_t \in \mathbb{R}$. We need to be assured that the scaling α_t lies in the interval $[0, 1]$ to guarantee that we have a convex combination of points of S as the approximation. $\tilde{\alpha}_t$ is always non-negative since for $t \geq 1$

$$\langle w_{t-1}, w_{t-1} + (x_t - y) \rangle = \langle w_{t-1}, x_t \rangle - \sum_{i=1}^{t-1} \beta_i^{t-1} \langle w_{t-1}, x_i \rangle \geq \langle w_{t-1}, x_t \rangle - \max_{x \in S} \langle w_{t-1}, x \rangle = 0,$$

where we define a sum from $i = 1$ to 0 to be 0. The inequality above holds because the β's are non-negative and they sum to 1. $\tilde{\alpha}_t$ can, however, be strictly larger than 1. Hence, to guarantee that our approximation is a convex combination of points from S we need to force the scaling factor back into the interval $[0, 1]$. We do this in the algorithm by assigning the value 1 to the scaling factor in this case. This choice minimizes $\|w_t\|$ because the derivative of $\|w_t\|^2$ with respect to α is non-positive for all $\alpha \leq \tilde{\alpha}$, that is

$$2(\alpha \|w_{t-1} + (x_t - y)\|^2 - \langle w_{t-1}, w_{t-1} + (x_t - y) \rangle) \leq 0, \quad \text{for all } \alpha \in (-\infty, \tilde{\alpha}_t].$$

Note that Algorithm 1 is the herding algorithm applied to a point outside the convex set. This corresponds to the CGM where the optimization over the step size is replaced by a step size of $1/t$. Algorithm 1 (ls) is the CGM algorithm applied to the problem of finding the projection of y on the convex set spanned by S.

2.1 Rate of Convergence

We take a closer look at the convergence behavior of the herding algorithm and the standard CGM when applied to a projection problem. It is known that the herding algorithm converges with a rate of $1/t$ to elements in the interior of compact convex sets in Hilbert space (this is shown in Chen et al. (2010) for what is called the marginal polytope. The same proof works for arbitrary compact convex sets in a Hilbert space). Hence, if we want to approximate the projection of an element y onto the convex set C and y is already in the interior of C (in essence a trivial projection problem) then standard proofs guarantee a rate of $1/t$. Obviously, the interesting case is one where the projection lies in the boundary. We provide a theorem below (with the proof being postponed to Section 6.1, p. 26) which extends current results to the case where y lies in the boundary or outside the convex set if an assumption on the perturbations is fulfilled. We also show that this assumption is both necessary and sufficient for the algorithm to converge with a rate of $1/t$. We conclude this section by presenting a number of settings in which this assumption is fulfilled.

It is instructive to go through the main ideas of our approach. Consider again Figure 1. We can observe that there is a minimal face of the convex set which contains the projection Py of y onto the convex set, i.e. there exists a face that contains Py and is a subset of any other face that contains Py. In the figure, the minimal face is the convex set which contains the projection, i.e. the red dot. The vector $y - Py$, with $Py \in C$ being the projection, stands orthogonal on this minimal face. Furthermore, this minimal face is either an extreme point or Py lies in the relative interior of it. In the latter case we have a ball around Py (relative to the affine subspace spanned by the minimal face) which is contained in the minimal face. This property of the existence of a ball

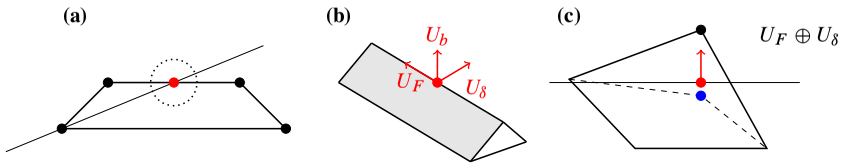


Figure 3: (a) A visualization of Lemma 1. (b) The figure depicts a triangular prism. The point of projection Py is marked by the red dot. The minimal face that contains Py is the top edge of the prism. The red arrows visualize the three subspaces U_F, U_b and U_δ. (c) The figure shows the projection of a set C_c onto the subspace $U_F \oplus U_\delta$ (the polytope enclosed by the thick line). The red dot marks the projection of Py onto the subspace and the red arrow visualizes the error at some stage t, i.e. w_t. The horizontal line represents $\{x : \langle w_t, x\rangle = 0\}$. The inner product between the black dot and w_t might be very different if evaluated over the whole space, i.e. if the component $P_b w_t$ is considered too. In particular, the polytope might appear to the algorithm like the modified polytope in which the black dot is replaced by the blue dot. In this modified polytope the best aligned element is the leftmost point.

around the element that we want to approximate is of crucial importance in all current approaches that demonstrate a fast reduction of error. We can therefore hope for a fast reduction of error in the affine subspace spanned by F. In fact, since $y - Py$ stands orthogonal on the minimal face, we can also observe that the algorithm applied to the minimal face behaves in exactly the same way as if we run it with Py instead of y and we can conclude that the algorithm converges with a rate of $1/t$ if it chooses only elements of F. The obvious problem is that we do not know the minimal face beforehand and we can not force the algorithm to choose only elements from it. At this point, a difficult task is to measure how much harm elements outside the minimal face can do when they are chosen by the algorithm. One important observation here is that for any element x outside the minimal face there exists no ball B around Py such that $\operatorname{aff}\{x, Py\} \cap B$ is contained in C (aff refers to the affine hull operator). Figure 3 (a) is a visualization of this property. The red dot marks Py. The line going through the bottom left corner and Py leaves the polytope at Py and there exists no ball in C such that the line segment in this ball lies fully in C. This property is also not difficult to prove formally. We summarize the result in the following simple lemma.

Lemma 1 *Given a compact convex set $C \subset \mathcal{H}$ and a $y \in \mathcal{H}$ there exists a minimal face F that contains Py and for every $x \in C \setminus F$ and any ball B centered at Py with radius greater than zero we have that* $\operatorname{aff}\{x, Py\} \cap B \not\subset C$.

The existence of the minimal face is shown in Step (ii) of the proof of Theorem 2. If the second part of the lemma would be false then we would have a point in $\operatorname{aff}\{x, Py\} \cap C$ such that Py is a convex combination of x and this point. This would imply $x \in F$ (see eq. 3 on p. 28) with a contradiction to the assumptions of the lemma.

The lemma implies that any element outside the minimal face which is chosen by the algorithm will introduce perturbations into the estimate that can not be easily removed: the line from x through

Py leaves the convex set at Py and there is no element $z \in C$ that is well aligned with $-(x - Py)$, i.e. a component of $-(x - Py)$ points outward of the convex set and there exists no z in C such that $z - Py$ has a positive inner product with this component. These perturbations get only smaller over time because of the down-scaling of the old approximation at steps t by $t/(t+1)$ and not because of future choices of the algorithm which are well aligned with the error. We are able to show that despite this property the herding algorithm converges with the same rate of $1/t$ that it would achieve if Py would lie inside C under an assumption on these perturbations.

Three subspaces. To state our assumption, it is convenient to introduce the centered versions $C_c = \{x - Py : x \in C\}$ of C and $F_c = \{x - Py : x \in F\}$ of F. We denote the linear subspace spanned by C_c with $\operatorname{span} C_c$. We can write $\operatorname{span} C_c$ as a direct sum $U_F \oplus U_\delta \oplus U_b$ of three orthogonal subspaces $U_F, U_\delta, U_b \subseteq \operatorname{span} C_c$ and we denote the orthogonal projections onto these subspaces and onto $\operatorname{span} C_c$ by P_F, P_δ, P_b and P_C. Here,

- $U_F = \operatorname{span} F_c$ ($U_F = \{0\}$ if $F_c = \{0\}$) is the (unique) subspace spanned by the minimal face,
- U_δ and U_b are any two orthogonal subspaces of $\operatorname{span} C_c$ that are also orthogonal to U_F and which fulfill
  1. $\operatorname{span} C_c = U_F \oplus U_\delta \oplus U_b$,
  2. there exists a ball (relative to the subspace U_δ) around 0 in $\{P_\delta x : x \in C_c\}$,
  3. there exists an orthonormal basis of U_b, say $e_1, \ldots, e_k$, $k = \dim U_b$, such that for all $i \leq k$, $\{0\} \neq \{\langle e_i, x\rangle : x \in C_c\} \subset (-\infty, 0]$,
  4. $P_C(y - Py) \in U_b$.

  $U_b = \{0\}$ and $U_\delta = \{0\}$ are possible.

Observe that this representation is not invariant under rotations. In particular, U_δ and U_b can change dimensions when C is rotated (see Figure 4 (a) for an example). The split into these three subspaces is helpful since it allows us to separate the errors that can be minimized by choosing particular elements $x \in S$ from the errors that cannot be reduced by choosing elements in S. Recall that the errors of our algorithms at step t are w_t. The perturbations at step t that cannot be minimized by suitable choices of elements in S are $P_b w_t$ and the remainder $P_\delta w_t + P_F w_t$ consists of the error that can be reduced by choosing appropriate elements in S. The split into these subspaces is shown in Figure 3 (b). U_F is here the subspace spanned by the line on top of the triangular prism, U_δ is the subspace orthogonal to it and aligned with the base of the prism. U_b is orthogonal to these two subspaces and the whole prism lies below Py (the red dot) in U_b.

A perturbed optimization problem. Apart from adding to the overall error, these perturbations introduce a subtle and more serious problem as follows. For any $(P_F + P_\delta)w_t$ we have an element s^* that maximizes $\langle s^*, (P_F + P_\delta)w_t\rangle$ and by choosing s^* we reduce the overall error (the reduction will be significant if $\|(P_F + P_\delta)w_t\|$ is large). Imagine there is a second element s which has a very small positive inner product with $(P_F + P_\delta)w_t$, i.e. $0 < \epsilon_t = \langle s, (P_F + P_\delta)w_t\rangle \ll \langle s^*, (P_F + P_\delta)w_t\rangle$. s will not be chosen if the algorithm optimizes over $U_F \oplus U_\delta$, however, since we optimize over all of $\operatorname{span} C_c$, it is possible that $\langle s, w_t\rangle > \langle s^*, w_t\rangle$. In particular, this may happen if $\langle P_b w_t, s^*\rangle / \langle P_b w_t, s\rangle$ is large and the perturbations $P_b w_t$ are effectively changing the geometry

of the optimization problem in $U_F \oplus U_\delta$ (see Figure 3 (c) for a visualization of this effect). The rate of convergence can be significantly reduced by this effect. In particular this happens if the perturbations make the algorithm choose a sequence of elements with inner products ϵ_t converging to 0 since the error in $U_F \oplus U_\delta$ will then not decrease over time. This problem can only occur if the $P_b w_t$ values affect the different elements in S in a very unbalanced way, i.e. there must be some elements $x, x' \in S$ such that $\langle P_b w_t, x\rangle \gg \langle P_b w_t, x'\rangle$. Our theorem below shows that this is the central problem that can hinder the rate of convergence. We demonstrate that Algorithm 1 converges with a rate of $1/t$ if, and only if, the perturbations affect the different elements in S in a sufficiently balanced way.

The main assumption and the theorems. Assumption 1 below formalizes the concept of a balanced influence of the perturbations on the elements of S. We need the following set of *critical points* for a given sequence of elements x_t chosen by the algorithm to state this assumption,

$$D(\{x_t\}) = \{x : x \in S \backslash F, x_t = x \text{ for infinitely many } t\}.$$

We use here $\{x_t\}$ to abbreviate $\{x_t\}_{t\geq 1}$. Only elements in $D(\{x_t\})$ can lead to a reduction in convergence rates and it suffices to check that none of the elements in $D(\{x_t\})$ lead to the above described effect to demonstrate fast convergence. Our main assumption is now the following:

Assumption 1: *Given sequences $\{x_t\}_{t\geq 1}, \{w_t\}_{t\geq 0}$ we assume that there exists a representation $U_F \oplus U_\delta \oplus U_b = \operatorname{span} C_c$ as described above, where for all $x, x' \in D(\{x_t\})$ there exists a $\Delta < \infty$ and a $t' < \infty$ such that $\langle -P_b w_t, x - Py\rangle \leq \Delta \langle -P_b w_t, x' - Py\rangle$ for all $t \geq t'$.*

We restrict our analysis to the case where there are only finitely many extremes of C. We thus assume that C is compact, convex and has only finitely many extremes. A set C with these properties is called a *convex polytope*. The following theorem demonstrates that for a convex polytope the above assumption is both necessary and sufficient for the fast rate of convergence of Algorithm 1.

Theorem 2 *Given a compact convex set $C \subset \mathcal{H}$ and a finite subset S of C with $\operatorname{ex} C \subseteq S$ there exists for $y \in \mathcal{H}$ a constant $b > 0$ such that Algorithm 1 has a worst-case approximation error of b/t for all $t \geq 1$ if, and only if, Assumption 1 is fulfilled for the sequences $\{x_t\}_{t\geq 1}$ and $\{w_t\}_{t\geq 0}$ generated by the algorithm.*

The reason why the herding algorithm retains its rate of convergence is that elements outside the minimal face will not be chosen anymore after some time $t_0 \in \mathbb{N}$ and the decay $t/(t+1)$ is high enough to remove the perturbations introduced in these first t_0 many steps fast enough so that they do not harm the overall rate of convergence, i.e. if we assume $Py = 0$ then

$$\|(I - P_F)w_t\| = \frac{t-1}{t}\|(I - P_F)w_{t-1}\| = \frac{t-2}{t}\|(I - P_F)w_{t-2}\| = \frac{t_0}{t}\|(I - P_F)w_{t_0}\|$$

and the error orthogonal to the minimal face decays with a rate of $1/t$. The situation is different for the CGM. While we can show that no elements outside the minimal face is chosen after some $t_0 \in \mathbb{N}$ the perturbations introduced in these initial steps dominate the rate of convergence of the algorithm even for steps $t \gg t_0$ (in the case of applying the CGM with away-steps it is known that the minimal face is identified after finite many steps under certain conditions and the algorithms optimizes after

this step over the minimal face (Wolfe, 1970; Guélat and Marcotte, 1986)). We decompose the error in the theorem below into two parts, the error in the affine subspace spanned by minimal face F and the error in the orthogonal complement of that space. This decomposition is natural in the way that we have a strong reduction of the error in the minimal face per step and only a weak reduction in the orthogonal complement of the minimal face. The first part of the theorem is standard and follows from Beck and Teboulle (2004). In the following, p_t denotes the approximation at time t. We use again Assumption 1, but be aware the the weight w_t is defined slightly differently for the CGM, i.e. the weight is scaled down by a factor of $1/t$ at step t.

Theorem 3 *Given a compact convex set $C \subset \mathcal{H}$, a finite subset S of C with $\operatorname{ex} C \subseteq S$ and an element $y \in \mathcal{H}$ the following holds:*

- *If only elements in $F \cap S$, where $F \subset C$ is the minimal face that contains Py, are chosen then the method converges linearly to the projection and there exist constants $b, \beta > 0$ with*

$$\|Py - p_t\| \leq be^{-\beta t}.$$

- *Under Assumption 1 if $\min_{x\in S}\|Py - x\| > 0$ and the approximation does not equal Py in finite many steps then the sequence $\{\|(I - P_F)(Py - p_t)\|\}_{t\geq 0}$ converges sub-linearly and there exists a constant $d > 0$ such that*

$$\|P_F(Py - p_t)\|^2 \leq (1 - \delta_F^2/\operatorname{diam}^2(F))\,\|P_F(Py - p_{t-1})\|^2 + d\,\|(I - P_F)(Py - p_{t-1})\|^2.$$

  Furthermore, there exists a time t_0 after which only elements in $F \cap S$ are chosen.

It is worth noting that the term $(1 - \delta_F^2/\operatorname{diam}^2(F))\,\|P_F(Py - p_{t-1})\|^2$ would guarantee a linear rate of convergence if the extra perturbation part would not be present.

The constant. It is of obvious interest to get insight into the size of the involved constants, in particular into the size of the constant b in Theorem 2 and its relation to quantities like δ_F and the dimension of $\operatorname{span} C_c$. The baseline is here the constant that we gain in the *trivial* case where $F = C$. Here, b can be taken to be $3r + 2r^2/\delta_F$, with $r = \sup_{x\in C}\|x\|$, and the constant does not depend on the affine dimension of C (see the proof of Theorem 2, part (ii.f)). In the case where $F \neq C$ we can first observe that b depends both on δ_m, the radius of the largest ball inside the projection of C_c onto $U_F \oplus U_\delta$, and on $\Delta' = \sup_{x,x'\in D(\{x_t\})} \Delta_{x,x'}$, where $\Delta_{x,x'}$ are the constants in Assumption 1, i.e. b depends on δ_m/Δ' (see the proof of Theorem 2, part (iv.b)). So the larger the ball around Py in $U_F \oplus U_\delta$ and the closer coupled the perturbations are the smaller the constant b. Referring these quantities back to geometric properties of C is non-trivial and in all likelihood providing a general characterization is at least as difficult as providing a general characterization of when Assumption 1 is fulfilled. However, in concrete settings it is often actually rather easy to provide bounds on b depending, for instance, on the affine dimension of C. We provide a number of examples in the next section. In particular, Corollary 6 - 8 contain concrete values for the constant. The dependence on d is here linear or sub-linear if we ignore the dependence of the dimension on the size of the set C, i.e. $r = \sup_{x\in C}\|x\|$ might also depend on the dimension. For instance, for the standard hypercube in d-dimensions $r = \sqrt{d}$.

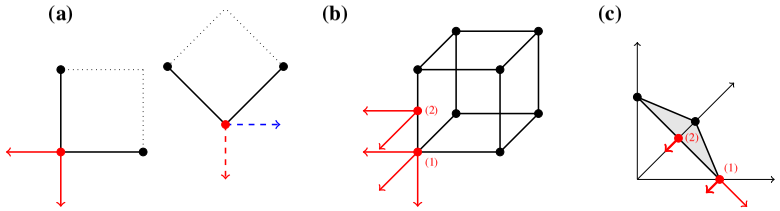


Figure 4: (a) The figure shows two possible splits into $U_b \oplus U_\delta$ depending on how the square is rotated. The first one is visualized by solid red arrows. Here $U_b = \mathbb{R}^2$ and $U_\delta = \{0\}$. The second split is visualized through the dashed arrows. The red dashed arrow is a basis vector of U_b and the blue dashed arrow is a basis vector of U_δ. (b) The figure shows two projection problems. The locations of Py are marked with the red dots and are annotated with (1) and (2). The red arrows visualize the space U_b for the two problems. For problem (1) this is a three dimensional space since U_F is here 0-dimensional and for (2) it is a 2 dimensional space. In both cases $U_\delta = \{0\}$. (c) The figure resambles (b). Instead of the hypercube it shows the simplex in three dimensions. The red arrows depict basis vectors for U_b. $U_\delta = \{0\}$ as for the hypercube.

2.1.1 EXAMPLES

We now take a look at a number of concrete projection problems to demonstrate how Assumption 1 can be used to prove fast convergence of Algorithm 1 in various situations. Our first result does not rely on assumptions about the shape of the convex polytope, but assumes that its span is a $d < \infty$ dimensional subspace of a Hilbert space and the projection Py lies in a $d-1$ dimensional face of the convex set (Corollary 4 and 5). In the applications we have in mind the point of projection is related to an estimator and the convex polytope enforces some form of sparsity. Typically, in such settings, the projection lies in some low dimensional face of the convex polytope.

We provide a rather general condition for this case in Corollary 6 and we use this corollary to demonstrate that Algorithm 1 converges with the fast rate independent of the location of Py if we are working, for instance, with the hypercube or the simplex (Corollary 7 and 8). In particular, Py can lie in a low dimensional face of the simplex or can be an extreme of it. There is certainly more to be understood here, but these cases demonstrate that the performance of Algorithm 1 is robust across a variety of settings and they demonstrate the uses of Assumption 1.

The proofs of the corollaries that follow are contained in the appendix on page 39 and onward. It seems also worth pointing out that we can rotate our coordinate system in an arbitrary way and translate the convex polytope and y without it affecting the algorithms or the rate of convergence. This holds because the algorithm uses only inner products and orthogonal operators (rotations) do not change these. Similarly, a translation does not affect the maximization step or the update. This allows us, for instance, to prove the fast rate of convergence for the standard hypercube $[0,1]^d$ and to generalize this result to arbitrary hypercubes in $\mathbb{R}^d$.

One-Dimensional U_b. If the space U_b is one-dimensional then the perturbations affect all elements in $S \backslash F$ equally (up to a finite multiplier) and Assumption 1 is always fulfilled. This is in particular the case if the minimal face is $(d-1)$-dimensional, but can also be fulfilled for lower dimensional faces if U_δ is not 0-dimensional.

Corollary 4 *Given a compact convex set $C \subset \mathcal{H}$ and a finite subset S of C with $\operatorname{ex} C \subseteq S$. If for $y \in \mathcal{H}$ there exists a decomposition $U_F \oplus U_\delta \oplus U_b$ of $\operatorname{span} C_c$ such that U_b is one dimensional then Assumption 1 is fulfilled. In particular, in this case there exists a constant $b > 0$ such that Algorithm 1 has a worst-case approximation error of b/t for all $t \geq 1$.*

One can relate the assumption that U_b is one-dimensional to a uniqueness assumption about the projection Py. If $P_C\, y \notin C$ and only elements in $A = \{z \in \mathcal{H} : z = \alpha P_C(y - Py) + P_C\, Py, \alpha \in [0,\infty)\}$ are projected onto Py then U_b is 1-dimensional and Assumption 1 is fulfilled. We summarize this result in another corollary.

Corollary 5 *Given a compact convex set $C \subset \mathcal{H}$ and a finite subset S of C with $\operatorname{ex} C \subseteq S$. Assumption 1 is fulfilled if $P_C\, y \in \mathcal{H} \backslash C$ and whenever $Pz = Py$ for some $z \in \mathcal{H}$ then $P_C z \in A$ holds. In particular, in this case there exists a constant $b > 0$ such that Algorithm 1 has a worst-case approximation error of b/t for all $t \geq 1$.*

This last assumption is not fulfilled in Figure 1 if the projection lies in the corners or edges, but it is fulfilled in all other cases. A quite different application of our theorem allows us to exploit the geometry of the convex set to derive the fast rate of convergence independent of the location of Py.

Zero-Dimensional U_δ. If we can rotate the convex set such that we get a split $U_F \oplus U_\delta \oplus U_b$ for which $U_\delta = \{0\}$ then our Assumption 1 is also fulfilled. The next corollary states this result. We use here, and in the following two corollaries, $d = \dim U_b$ and we let $e_1, \ldots, e_d$ be any basis of U_b. Furthermore, $\alpha = \min_{i \leq d} \min\{|\langle e_i, x - Py\rangle| : x \in S \backslash F, \langle e_i, x - Py\rangle \neq 0\} > 0$ and $r = \sup_{x \in C} \|x\|$.

Corollary 6 *Let C be a compact convex set in some Hilbert space $\mathcal{H}$, S a finite set with $\operatorname{cch} S = C$, and $y \in \mathcal{H}$ such that there exists a split into $U_F \oplus U_\delta \oplus U_b$ of $\operatorname{span} C_c$ with $U_\delta = \{0\}$. Assumption 1 is fulfilled and there exists a constant $b > 0$ such that Algorithm 1 has a worst-case approximation error of b/t for all $t \geq 1$. The constant b can be chosen as $\sqrt{d}\, 4r^3/(\alpha\delta_F) + 6r^2(1/\delta_F + 1/\alpha) + 5r$.*

This condition is somewhat abstract but leads, for example, to results that prove that the algorithm converges with the fast rate in classical sparsity settings independent of the location of Py (Corollary 7 and 8 below). Also, observe that Corollary 6 together with Corollary 4 imply that if Py lies in a face of dimension $(d-2)$ then Algorithm 1 will converge with the fast rate: if U_δ is 0-dimensional then this follows from Cor. 6 and if U_δ is 1-dimensional then U_F is also 1-dimensional and the result follows from Cor. 4 (U_δ cannot be 2-dimensional due to Lemma 1).

Hypercubes. Here, we consider the compact convex set $[0,1]^d = \operatorname{cch}\{0,1\}^d \subset \mathbb{R}^d$ and transformations of it (rotations, translations and changes of its size). The set of extremes of the standard hypercube is $\{0,1\}^d$ and the d-dimensional hypercube has $2^{d-m}\binom{d}{m}$ many m-dimensional faces, where $m \leq d$. We formulate the following corollary in terms of a rotation matrix Q, a translation by a vector z and a scaling by a scalar c. In the proof of the corollary we transform the hypercube to the standard hypercube and we show that $U_\delta = \{0\}$ independent of the location of y. This is visualized

in Figure 4 (b) for a 3-dimensional hypercube. (1) and (2) are here two projection problems and the red arrows depict two bases of U_b. Constructing such bases for which $U_\delta = \{0\}$ is always possible for the standard hypercube.

Corollary 7 *Let $y \in \mathbb{R}^d$, Q any orthogonal matrix, $c > 0$ any scaling, $z \in \mathbb{R}^d$, $S = \{0,1\}^d$ and $C = [0,1]^d$, $d \geq 1$. Assumption 1 is fulfilled for the set $\tilde{S} = cQ[S + z]$ and $\tilde{C} = cQ[C + z]$ (independently of the dimensionality of the face Py lies in). In particular, there exists a constant $b > 0$ such that Algorithm 1 has a worst-case approximation error of b/t for all $t \geq 1$. The constant b can be chosen as $\sqrt{d}4r^3/(c\delta_F) + 6r^2(1/\delta_F + 1/c) + 5r$.*

Standard Simplex. Algorithm 1 also attains a fast rate of convergence if the convex set we use is the standard simplex in $\mathbb{R}^d$ independently of the location of y. The standard simplex has $\binom{d}{m}$ many faces of dimension $m-1$, i.e. $1 \leq m \leq d$ and the dimension of the corresponding span of the centered face is $m-1$. The faces of such a simplex are again simplices. In particular, if we denote the standard simplex with $\Delta^{d-1} = \operatorname{cch}\{e_1, \ldots, e_d\}$, then the set of $m-1$ dimensional faces of Δ^{d-1} are $\{\operatorname{cch}\{e_{i_1}, \ldots, e_{i_m}\} : i_1 < i_2 \ldots < i_m, i_j \leq d\ \forall j \leq m\}$. As for the hypercube we can show that $U_\delta = \{0\}$. This is visualized in Figure 4 (c) for the standard simplex in $\mathbb{R}^3$. The figure shows two projection problems (1) and (2) and corresponding bases of U_b.

Corollary 8 *Let $y \in \mathbb{R}^d$, $S = \{e_i : i \leq d\}$ and $C = \Delta^{d-1} = \operatorname{cch} S$, $d \geq 1$. Assumption 1 is fulfilled and there exists a constant $b > 0$ such that Algorithm 1 has a worst-case approximation error of b/t for all $t \geq 1$. The constant b can be chosen as $4dr^3/\delta_F + 6r^2(1/\delta_F + \sqrt{d}) + 5r$.*

2.2 The Algorithm for Finite S

We are particularly interested in the case where $S = \{x_1, \ldots, x_n\}$ is finite. If S is finite we can change the algorithm to keep track of $\langle w_t, x_i\rangle =: a_{ti}$ instead of $w_t \in \mathcal{H}$. This reflects a change of representation from a basis representation of w_t to one based on S. The weight vector can, after this change of representation, be replaced by $a_t = (a_{t1}, \ldots, a_{tn})$ in the algorithm.

Algorithm 2. Input: $y \in \mathcal{H}$, $T \in \mathbb{N}$, and $S \subset \mathcal{H}$.

1. [Initialize] Set $a_0 = (\langle y, x_1\rangle, \ldots, \langle y, x_n\rangle)$ and $t = 1$.
2. [Optimization oracle] Choose $i_t \in \arg\max_{j \leq n} a_{(t-1)j}$.
3. [Update weight] Set $a_t = a_{t-1} - (\langle x_{i_t}, x_1\rangle, \ldots, \langle x_{i_t}, x_n\rangle) + a_0$.
4. [Iterate] If $t < T$ then increment t by 1 and go back to **2**.
5. [Terminate] Return the approximation $(x_{i_1} + \ldots + x_{i_T})/T$.

The computational costs per iteration are then determined by the number of samples n and the computational costs of calculating inner products in $\mathcal{H}$ (n inner products per iteration).

2.3 Line Search for Finite S

The line search in Algorithm 1 (ls) contains terms of the form $\|w_{t-1}\|^2$. The philosophy in Algorithm 2 is to avoid measuring objects in their norm and to work solely with relations between objects in the form of inner products $\langle y, x_i\rangle$ to reduce computational costs. To follow this philosophy we need a version of the line search that does not need access to $\|w_{t-1}\|$ or $\|y\|$. Achieving this goal is

easy enough: the line search aims for finding an α_t such that $p_t = (1-\alpha_t)p_{t-1} + \alpha_t x_{i_t}$, with p_t being our approximation at step t, achieves minimal distance to y. So we want that

$$\alpha_t \in \operatorname*{arg\,min}_{\alpha \in [0,1]} \|(1-\alpha)p_{t-1} + \alpha x_{i_t} - y\|^2 .$$

Setting the derivative of the squared norm to zero with respect to α we gain

$$\tilde{\alpha}_t = \frac{\langle y - p_{t-1}, x_{i_t} - p_{t-1}\rangle}{\|p_{t-1} - x_{i_t}\|^2}.$$

The norm in the denominator can be calculated recursively and calculating its value needs no significant computational resources. Because $\|w_t\| = \|p_t - y\|$ we gain the same $\tilde{\alpha}_t \in [0, \infty)$ as in Algorithm 1 (ls). Also, the minimizer over the interval $[0,1]$ is again $\alpha_t = 1 \wedge \tilde{\alpha}_t$.

2.4 Approximation Error Bounds through Duality

There exists a well established test for the approximation error (Wolfe, 1976). Let $S = \{x_1, \ldots, x_n\}$ still be finite. For any $p \in C$, $p \neq y$, we can write $\langle p, Py\rangle = \langle p, \alpha_1 x_1 + \ldots \alpha_n x_n\rangle \geq \min_{i \leq n}\langle p, x_i\rangle$ and

$$\|p - y\| \geq \|Py - y\| \geq \sum_{i=1}^n \alpha_i \langle x_i - y, \frac{p-y}{\|p-y\|}\rangle \geq \min_{x \in S}\langle x - y, \frac{p-y}{\|p-y\|}\rangle.$$

So

$$\|p - y\| - \|Py - y\| \leq \|p - y\| - \min_{x \in S}\langle x - y, \frac{p-y}{\|p-y\|}\rangle.$$

The last term tells us how far we are away from the optimum and it can be used to guarantee a specific approximation error at termination. In the settings we are interested in $\|p - y\|$ is, in fact, too expensive to compute and we need to modify the approach from Wolfe (1976). Observe that

$$\max_{x \in S}\langle p - x, \frac{p-y}{\|p-y\|}\rangle = \|p - y\| - \min_{x \in S}\langle x - y, \frac{p-y}{\|p-y\|}\rangle$$

gives us the easier to compute

$$\max_{x \in S}\langle p - x, p - y\rangle \geq \|p - y\|\,(\|p - y\| - \|Py - y\|)$$

The right side is directly an upper bound on $(\|p - y\| - \|Py - y\|)^2$, because $\|p - y\| \geq \|p - y\| - \|Py - y\| \geq 0$. However, this bound is overly conservative and we loose an order of magnitude since we lower bound $\|p - y\| \geq \|Py - y\|$ with a quantity that goes to 0. A better way seems to be to use linear functionals to approximate $p - y$. Natural choices are here $\langle x, \cdot\rangle$ for $x \in S$ or $\langle p, \cdot\rangle$ because we can often compute these functionals efficiently. The former choice can be exploited in the following way

$$\|y - p\| \geq \max_{x \in S}\left|\langle y - p, \frac{x}{\|x\|}\rangle\right|$$

and we can calculate a lower bound on $\|y - p\|$ in about $O(n)$ operations (depending on the representation of p) by calculating the right hand side inner product for every $x \in S$. Using this bound we get the following upper bound on the approximation error

$$\frac{\max_{x \in S}\langle p - x, p - y\rangle}{\max_{x \in S}|\langle y - p, x/\|x\|\rangle|} \geq \|p - y\| - \|Py - y\| .$$

The latter choice leads to

$$\|y-p\| \geq \langle y-p, \frac{p}{\|p\|}\rangle = \langle y, \frac{p}{\|p\|}\rangle - \|p\|\,.$$

The inequality also holds when taking the absolute value of the right side and

$$\max_{x\in S} \frac{\langle p-x, p-y\rangle}{|\|p\| - \langle y, p/\|p\|\rangle|} \geq \|p-y\| - \|Py-y\|\,.$$

If our convex set is a norm ball then as $p \to Py$ we also have that $y-p$ gets more and more aligned with $p/\|p\|$ and we expect this bound to be good.

2.5 Parallelization

Algorithm 2 is easy to parallelize since the bottleneck per iteration is the calculation of n inner products in the update equation of a_t. These inner products can be calculated independently of each other and by having c processes available we can distribute the computation such that each process has to calculate at most $\lceil n/c \rceil$ many inner products per iteration. Determining the arg max can then be achieved by a loop over the n entries of a_t. This operation is typically fast and no parallelization is needed. Though, it is easy to distribute this operation too to reduce the computation time to $\lceil n/c \rceil + c$ by first calculating c maxima over sets of size at most $\lceil n/c \rceil$ and by then calculating the maximum of these c maxima. Finally, the summation of the vector of inner products with $a_{t-1} - a_0$ can also be split such that each process has to perform at most $\lceil n/c \rceil$ many of these summations. This gives us in total a computation time in the order of $\lceil n/c \rceil$.

3. CCP-Kernel Regression

We now want to apply the algorithm to a challenging statistical problem. That is the problem of non-parametric regression. For this we use a reproducing kernel Hilbert space (RKHS, Aronszajn (1950)) which is in a certain sense natural given past herding applications, but it is also computationally efficient thanks to the reproducing property.

An RKHS $\mathcal{H}$ is implicitly defined through a kernel function $k(x, x')$. $\mathcal{H}$ is the completion of

$$L := \Big\{\sum_{i=1}^{n} \alpha_i k(x_i, \cdot) : n \in \mathbb{N}, x_i \in X, \alpha_i \in \mathbb{R}\Big\},$$

where X is the domain of the covariates (or inputs). In statistical applications we like to approximate $\mathcal{H}$ with a nested family of sets of functions that are restricted in a certain way in their size. One nested family is, for instance, the family of closed balls B_r of radius r centered at zero. For these we know that $\bigcup_{r\geq 0} B_r = \mathcal{H}$. The balls B_r are, however, not compact if $\mathcal{H}$ is infinite dimensional. Also, controlling the extremes of B_r is not necessarily easy since we usually only have access to the kernel function k and not a basis $\{e_n\}_{n\geq 1}$ of $\mathcal{H}$. So ideally we would like a family of sets that can approximate $\mathcal{H}$ like the balls B_r, but in contrast to B_r the different sets would be compact, convex and controllable through the kernel. There is a family of sets with these properties which arises naturally from the kernel function (cl denotes the closure operator):

$$C(r) = \operatorname{cch}\{k(x,\cdot) : x \in X\} = \operatorname{cl}\Big\{\sum_{i=1}^{n} \alpha_i k(x_i, \cdot) : n \in \mathbb{N}, x \in X^n, \alpha \in \mathbb{R}^n_+, \|\alpha\|_{\ell_1} \leq r\Big\} \subset \mathcal{H}.$$

$C(r)$ is compact and convex and the extremes of $C(r)$ are contained in the set $S = \{k(x,\cdot) : x \in X\}$. There is an obvious similarity to the definition of L and by suitably symmetrizing $C(r)$ we gain a family of sets which covers all of L and can approximate any element in $\mathcal{H}$ up to arbitrary precision.

3.1 Symmetric Closed-Convex Hull

The sets $C(r)$ might not contain 0 or any elements on the negative axes. This is obviously not ideal for representing functions. We overcome this shortcoming by symmetrising $C(r)$ by including the elements $-k(x,\cdot)$. In this way we get the family of sets

$$C_s(r) = \operatorname{cl}\Big\{\sum_{i=1}^{n} \alpha_i k(x_i, \cdot) : n \in \mathbb{N}, x \in X^n, \alpha \in \mathbb{R}^n, \|\alpha\|_{\ell_1} \leq r\Big\}.$$

This family of sets has some similarity to the closed balls in $\mathcal{H}$. In particular, if the kernel function is bounded then $C_s(r)$ is a subset of the closed ball of radius $\tilde{r} = r \sup_{x\in X} k(x,x)$ in $\mathcal{H}$ since

$$\Big\|\sum_{i=1}^{n} \alpha_i k(x_i,\cdot)\Big\| \leq \|\alpha\|_{\ell_1} \sup_{x\in X} k(x,x) \leq \tilde{r}$$

and the smallest closed set containing the elements $\sum_i \alpha_i k(x_i,\cdot)$ can not be larger than a closed ball of radius $\tilde{r}$.

Denseness and Universal Approximation. More important for us is the following property: the set $\bigcup_{r\geq 0} C_s(r)$ is *dense* in $\mathcal{H}$, since for any $f \in \mathcal{H}$ and $\epsilon > 0$ there exists an $n \in \mathbb{N}$, elements $x_1, \ldots, x_n \in X$ and coefficients $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$ such that $\big\|f - \sum_{i=1}^n \alpha_i k(x_i,\cdot)\big\| \leq \epsilon$. But $\sum_{i=1}^n \alpha_i k(x_i,\cdot)$ is an element of $C_s(r)$ if $r \geq \sum_{i=1}^n |\alpha_i|$. In other words, we can approximate any function in $\mathcal{H}$ arbitrary well by making r large enough. Furthermore, if our kernel function is a universal kernel then we can approximate any continuous function on X arbitrary well in the supremums norm. This property is sometimes called the universal approximation property.

3.2 Simplifying the Sets

Optimizing over $C_s(r)$ itself is difficult and from a practical point of view it makes sense to reduce the sets further to save computation time. In regression we have a number of covariates $x_1, \ldots, x_n$ given and we know that we can represent any RKHS function h exactly on these points $x_i, i \leq n$, by functions of the form $\sum_{i=1}^n \alpha_i k(x_i,\cdot)$, where $\alpha \in \mathbb{R}^n$ are suitable weights, if the kernel matrix $K = (k(x_i,x_j))_{i,j\leq n}$ is of full rank because

$$K\alpha = \begin{pmatrix} h(x_1) \\ \vdots \\ h(x_n) \end{pmatrix}$$

has then a (unique) solution. From this point of view it makes sense to use the family of sets

$$\mathcal{C}(r) := \Big\{\sum_{i=1}^{n} \alpha_i k(x_i,\cdot) : \alpha \in \mathbb{R}^n, \|\alpha\|_{\ell_1} \leq r\Big\}.$$

instead of $C_s(r)$. $\mathcal{C}(r)$ is fully characterized by $S = \{\pm rk(x_1,\cdot),\ldots,\pm rk(x_n,\cdot)\}$ in the sense that $\mathcal{C}(r) = \operatorname{cch} S$ and S contains the extremes of $\mathcal{C}(r)$. S itself is of size $2n$ and we can optimize efficiently over it.

3.3 Interpolation in $\mathcal{H}$

Before approaching the regression problem we start with a closely related interpolation problem. We aim for interpolating a function $g \in \mathcal{H}$ at n support points $x_1,\ldots,x_n$. Let $y_1 = g(x_1),\ldots,y_n = g(x_n)$. The interpolation algorithm is essentially an efficient version of Algorithm 2 applied to the elements $\pm rk(x_1,\cdot),\ldots,\pm rk(x_n,\cdot)$, where r is the scaling factor of $\mathcal{C}(r)$. So at the heart of the algorithm we have a vector $a \in \mathbb{R}^n$ which keeps track of the approximation error and the entries a_i are just $\langle w_t, k(x_i,\cdot)\rangle$, where w_t is the weight vector used in Algorithm 1. We formulate the algorithm in terms of "$\leftarrow$" assignments which denote the operation of overwriting the left hand side with the value on the right hand side.

Algorithm 3. Input: $y \in \mathbb{R}^n$, $T \in \mathbb{N}$, $r > 0$, a kernel function k and $x_1,\ldots,x_n \in X$.

1. [Initialize] Set $a = ry$ and $t = 1$.
2. [Optimization oracle] Choose $j' \in \arg\min_{i\le n} a_i$, $j'' \in \arg\max_{i\le n} a_i$.
3. If $-\min_{i\le n} a_i \ge \max_{i\le n} a_i$ then let $j = j', z_t = x_j$ and $s_t = 1$
4. else let $j = j'', z_t = x_j$ and $s_t = -1$.
5. [Update weight] Set $a_i \leftarrow a_i + ry_i - r^2 s_t k(x_i, z_t)$ for all $i \le n$.
6. [Iterate] If $t < T$ then increment t by 1 and go back to **2**.
7. [Terminate] Return the regressor $f(x) = r(s_1 k(z_1,x) + \ldots + s_T k(z_T,x))/T$.

The complexity of the algorithm is $O(Tn)$ since we have T iterations and we need to update in each iteration $a \in \mathbb{R}^n$. The bottleneck in this algorithm is the evaluation of the $k(z_i,x)$.

Line search. The line search version of this algorithm is based on Section 2.3. We update recursively the elements η,γ,π, where at step t we have that $\eta = \|p_{t-1}\|^2$, $\gamma = \langle p_{t-1}, y\rangle$ and $\pi = (\langle p_{t-1}, k(x_1,\cdot)\rangle,\ldots,\langle p_{t-1}, k(x_n,\cdot)\rangle)$, to keep the complexity of the algorithm at $O(Tn)$.

Algorithm 3 (ls). Input: $y \in \mathbb{R}^n$, $T \in \mathbb{N}$, $r > 0$, a kernel function k and $x_1,\ldots,x_n \in X$.

1. [Initialize] Set $a = ry, t = 1, \eta = 0, \gamma = 0$ and $\pi = (0,\ldots,0)$.
2. [Optimization oracle] Choose $j' \in \arg\min_{i\le n} a_i$, $j'' \in \arg\max_{i\le n} a_i$.
3. If $-\min_{i\le n} a_i \ge \max_{i\le n} a_i$ then let $j = j', z_t = x_j$ and $s_t = 1$
4. else let $j = j'', z_t = x_j$ and $s_t = -1$.

5. [Calculate step size] Let $v = r^2(k(x_1,z_t),\ldots,k(x_n,z_t))$ and
$$\tilde{\alpha}_t = \frac{rs_t y_j - rs_t\pi_j - \gamma + \eta}{\eta - 2rs_t\pi_j + v_j}, \quad \text{if } t = 1 \text{ let } \alpha_t = 1 \text{ and otherwise let } \alpha_t = 1 \wedge \tilde{\alpha}_t.$$
6. [Update weight] $a \leftarrow (1-\alpha_t)a + \alpha_t(ry - s_t v)$.
7. [Update line search variables] $\eta \leftarrow (1-\alpha_t)^2\eta + 2\alpha_t(1-\alpha_t)rs_t\pi_j + \alpha_t^2 v_j$, $\pi \leftarrow (1-\alpha_t)\pi + \alpha_t(s_t/r)v$ and $\gamma \leftarrow (1-\alpha_t)\gamma + \alpha_t s_t r y_j$.
8. [Iterate] If $t < T$ then increment t by 1 and go back to **2**.
9. [Terminate] Return the regressor $f(x) = r(s_1\beta_1 k(z_1,x) + \ldots + s_T\beta_T k(z_T,x))/T$, with $\beta_i = \alpha_i \prod_{n=i+1}^{T}(1-\alpha_n)$.

The expensive calculation is here the calculation of $v = (k(x_1,z_t),\ldots,k(x_n,z_t))$.

Approximation error stopping rule. We can also use an approximation error of below ϵ as the stopping criterion by using bounds from Section 2.4. The algorithm below achieves this in $O(Tn)$ for the bound that uses our approximation p_t to gauge the approximation error. The algorithm is very similar to the line search algorithm because we need to store similar quantities for calculating the bound as for the line search. We update recursively the elements η,γ,π, where at step t (before the update) we have that $\eta = \|p_{t-1}\|^2, \gamma = \langle p_{t-1}, y\rangle, \pi = (\langle p_{t-1}, k(x_1,\cdot)\rangle,\ldots,\langle p_{t-1}, k(x_n,\cdot)\rangle)$. The complexity of the algorithm is again $O(Tn)$. The version shown below is for the line search. By replacing α_t with $1/t$ one can gain a version for the standard algorithm. We state only the changes to Algorithm 3 (ls).

Algorithm 3 (ls, ae). *(replace 8. and 9. in Algorithm 3 (ls)).* Input: $y \in \mathbb{R}^n$, $\epsilon > 0$, $r > 0$, a kernel function k and $x_1,\ldots,x_n \in X$.

8. [Upper bound] Calculate the upper bound
$$b = \max_{i\le n} \frac{\eta - r\pi_i - \gamma + ry_i}{\left|\sqrt{\eta} - \gamma/\sqrt{\eta}\right|} \vee \max_{i\le n} \frac{\eta + r\pi_i - \gamma - ry_i}{\left|\sqrt{\eta} - \gamma/\sqrt{\eta}\right|}.$$
9. [Terminate] If $b \le \epsilon$ return the regressor $f(x) = r(s_1\beta_1 k(z_1,x) + \ldots + s_t\beta_t k(z_m,x))$ where $\beta_i = \alpha_i \prod_{n=i+1}^{t}(1-\alpha_n)$. Otherwise, go back to **2**.

As for the other two algorithms above the computationally most demanding operation in this algorithm is the calculation of $(k(x_1,z_t),\ldots,k(x_n,z_t))$.

3.4 Norm Minimizing Regressor

The interpolation algorithms can also be applied to the regression problem. Let the observations be $(x_1,y_1),\ldots,(x_n,y_n)$ and let K be the kernel matrix. If the kernel matrix is of full rank then with $\alpha = K^{-1}y$, where $y = (y_1,\ldots,y_n)^\top$, the function $h(x) = \sum_{i=1}^n \alpha_i k(x_i,\cdot) \in \mathcal{H}$ fulfills $(h(x_1),\ldots,h(x_n)) = K\alpha = y^\top$ and the interpolation algorithm applied to h will converge under

the usual conditions with a rate of $1/t$, that is $\|h - g_t\|^2 \leq b^2/t^2$ with $g_t = (1/t)\sum_{i=1}^{t} k(z_i, \cdot)$. There are two interesting observations here: first, the algorithm minimizes the distance to h implicitly without knowledge of h itself. In fact, determining h numerically is usually impossible due to ill-conditioning. Second, the algorithm minimizes the distance in the RKHS norm and not in a least-squares sense. Both, the RKHS norm and the least-squares criterion can be seen as distance measures that are in certain ways related. For instance, a norm-distance of zero between two functions implies that they have the same least-square error and, furthermore, the least-squares error of our approximation is bounded by

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - g_t(x_i))^2 = \frac{1}{n}\sum_{i=1}^{n}|\langle h - g_t, k(x_i, \cdot)\rangle|^2 \leq \left(\frac{1}{n}\sum_{i=1}^{n}k(x_i, x_i)\right)(\|h - Ph\|^2 + c/t^2)$$

where Ph denotes the projection of h onto $\mathcal{C}(r)$ the constant c is $(b^2/n)\sum_{i=1}^{n}k(x_i, x_i)$.

One can also formulate the norm minimization problem as a convex optimization problem

$$\min_{\alpha \in \mathbb{R}^n} \alpha^\top (K\alpha - 2y) \quad \text{s.t.} \sum_{j=1}^{n} |\alpha_j| = r.$$

and use convex programs to find the projection. This representation has the drawback that it makes explicit use of the $n \times n$ kernel matrix K and for large scale problems this formulation needs significant amounts of memory.

4. Experiments

We conducted a set of experiments to gauge the performance of the approach. The first set of experiments was constructed to demonstrate the behavior of the error bounds and to compare the optimization routine with some standard optimization procedures. The second set of experiments compared the regressor with well established regressors in a small scale setting. The advantage of the small scale setting is that we can compare to methods like the GP regressor. The final set of experiments focused on a large scale benchmark data set and compared our method to the Fast-KRR, which is the state-of-the-art regressor for large scale problems.

4.1 Experiment 1: Empirical Rate of Convergence

The first set of experiments were conducted to explore basic properties of the optimization approach: 'how does a typical regression curve look like in comparison to a GP regression curve?' 'How do the error bounds relate?' 'How does the optimization routine behave in comparison to a general purpose optimizer?' In our first experiment we generated 1000 data points from a Gaussian process (Gaussian covariance function) with normal distributed noise (the right plot in Figure 5). We fitted the maximum a posteriori estimator (MAP, known hyper-parameters) to it (red curve) and then did split the data into an 800 and 200 batch to run a cross validation loop over the hyper parameters (we also used a Gaussian covariance but with an unknown width parameter). We ran the CCP algorithm for 20 (yellow) and 100 (purple) iterations (without line search). The red bars show the points and weights of the solution when the algorithm is run for 100 iterations. The black bars show the solution found by the cvx Matlab toolbox for a precision of about 10^{-16}. The next experiment was about the bounds. We used again 1000 data points generated as above and a fixed hyper-parameter (no cv loop) to see how the bounds behave as the number of iterations increases. There

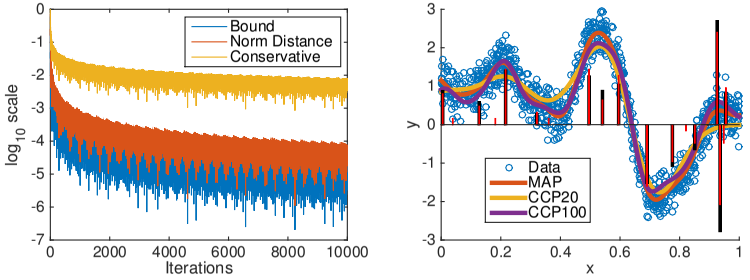


Figure 5: **Left:** The plot shows three different bounds on the approximation error. **Right:** The figure shows data from a Gaussian process with Gaussian likelihood function. The data is overlaid with three regression curves. MAP is here the optimal regressor. The vertical bars show the weight that is assigned by the CCP-regressors with 100 iterations (red) and the cvx matlab optimization toolbox (black) to observations at various locations x.

n	100	500	1000	2000	3000	4000	5000
CCP	0.03(0.03)	0.1(0.08)	0.2(0.09)	0.25(0.29)	0.3(0.26)	0.47(0.26)	0.57(0.61)
cvx	0.64(0.13)	1.3(0.08)	2.7(0.13)	8.16(0.12)	17.2(0.39)	31.3(1.68)	49.8(0.88)

Figure 6: Run time comparison between the CCP projection algorithm and a general purpose solver.

is one difficulty here, which is that we do not have the best fitting function $h \in \mathcal{H}$ and we can not calculate the exact distance to h. The three curves on the left side of the figure are our two bounds of the difference $\|h - g_t\| - \|h - Ph\|$, g_t is the regressor after t iterations and Ph the solution of the optimization problem which we try to find. *Conservative* refers to the bound where we use $\|g_t - h\| \geq (\|g_t - h\| - \|Ph - h\|)$ and *Bound* refers to our second bound) and the distance to Ph (*Norm Distance*), that is $\|g_t - Ph\|$ which is also an upper bound of $\|h - g_t\| - \|h - Ph\|$ (we determined Ph by using cvx with a precision of about 10^{-16}). The results are shown on the left in Figure 5. One can observe that all three bounds show similar behavior, however, the conservative bound is, as expected, very loose.

We were also interested in a run time comparison between the projection algorithm and a general purpose solver (the cvx toolbox) to see how much can be gained by using the specialized projection method. The cvx toolbox uses the SDPT3 package to solve semidefinite-quadratic-linear programming problems. The details of the SDPT3 package are described in Toh, Todd, and Tutuncu (1999); Tutuncu, Toh, and Todd (2003). We used as a stopping rule for both methods an error below 10^{-4}, that is CCP stopped when our bound guaranteed us that the error is below 10^{-4} and cvx stopped when its own error bound signaled an error below this threshold (details about how the precision is calculated in the SDPT3 package can be found in Toh et al. (1999)[Section 3]). The results of the run time comparison are shown in Figure 6 (mean(standard deviation) in seconds averaged over 10 runs; same experimental setup as for the bounds plot; n is the number of data points).

4.2 Experiment 2: Run time vs. Least-Squares Error on a Small Scale Sample

The second set of experiments tested on a small scale how the run time of the CCP-regressor measures against the statistical performance. We were interested in seeing how the method fares in comparison to standard Gaussian process/ridge regression and Fast-KRR. We reproduced the experiment from Zhang et al. (2013) which uses the million songs data set (about 450000 data points and a covariate dimension of 90). We normalized the data as in Zhang et al. (2013) by letting each covariate dimension have standard deviation 1. We also used the same kernel (Gaussian with $\sigma = 6$). Finally, we normalized the response variable (year when a song appeared) to lie in [0, 1] by subtracting the minimal year and dividing by (maximum - minimum). 450000 data points are too much to run the standard GP regressor/ridge regressor and we downsampled the data set to a small subset of 5000 training points and 1000 test points. We were interested in how the radius affects the performance of CCP and how different it is from how λ affects the GP. We were also interested in seeing how the error threshold translates into least-squares performance and run time. The results are shown in the table in Figure 7. The notation is (run time - least-squares error), r is the radius for CCP (with line-search), p the number of elements in each partition of Fast-KRR (we used for Fast-KRR the same regularization schedule as in Zhang et al. (2013)) and ϵ, in CCP-ϵ, refers to the stopping criterion. We used our upper bound to gauge the current error and we stopped when the error bound passed ϵr. The table below shows the results. We also ran the CCP-0.1 setting with $r = 5000$ to see how close we can get to the performance of the GP regressor. The estimator took 30.6 seconds to produce an estimate with an error of 0.124.

$r(p)/\lambda$	50/0.02	100/0.01	250/0.004	500/0.002	1000/0.001
GP	56 - 0.121	55 - 0.121	55 - 0.121	55 - 0.121	65 - 0.121
CCP-0.1	1.03 - 0.53	1.63 - 0.46	2.97 - 0.36	4.77 - 0.22	8.89 - 0.17
CCP-0.01	3.66 - 0.47	6.21 - 0.33	12.8 - 0.19	30.67 - 0.16	93.13 - 0.13
Fast-KRR		9.3 - 0.236	10.6 - 0.203	12.5 - 0.179	16.5 - 0.158

Figure 7: The table shows a comparison between the GP/ridge regressor, the Fast-KRR and the CCP-regressor. Each entry consists of the run time (seconds) and the least-squares error.

As one expects the GP run time is essentially independent of λ. A bit surprising is here that the regularization parameter seems to have hardly any effect on the least-squares error of the GP regressor. The run time of the CCP algorithm, however, is strongly affected by r which is consistent with our bound in which r influences the constant of the convergence rate. For both thresholds the computation time increases slightly non-linearly in r. The least-squares error depends on both ϵ and r. Increasing r results in this experiment in a higher reduction in least-squares error in dependence of the added computation time.

The lowest error is achieved by the GP-regressor (which corresponds to the Fast-KRR with one partition). A slightly suboptimal solution with a least-squares error of 0.124 is achieved by the CCP-regressor in about 25 seconds less than what the GP-regressor needs. One can also observe that Fast-KRR is fast in computing an estimate with a low least-squares errors which is en par with

the CCP-regressor in this experiment (in 10 seconds the Fast-KRR reaches, for instance, about 0.2 and the CCP-regressor 0.17). The interesting question is here how the performance of Fast-KRR and the CCP-regressor scale with the amount of data.

4.3 Experiment 3: Run time vs. Least-Squares Error on a Large Scale Sample

In the last experiment we compared Fast-KRR with the CCP-regressor (line search) on the full million songs data set. We used similar partition sizes as in Zhang et al. (2013) for Fast-KRR and we ran the CCP-regressor with $r = 100000$. The bottleneck in the CCP-regressor is the number of kernel evaluations that need to be performed which are tn for t iterations and n samples. The Fast-KRR regressor needs for m partitions (for simplicity let m be such that $0 \equiv n \mod m$) in the order of n^2/m many kernel evaluations and it needs to calculate m many inverses of matrices of size $(n/m) \times (n/m)$. So if $t = n/m$ then the CCP-regressor needs to perform exactly as many kernel evaluations as the Fast-KRR algorithm. Especially for large m this will leave us with few iterations for the CCP-regressor and one expects that if the kernel evaluation is expensive in comparison to the computational cost of an inverse then Fast-KRR will perform better. On the other hand if we have either small partitions or cheap to evaluate kernels then the CCP-regressor will excel compared to Fast-KRR. In terms of memory: the CCP-regressor needs a small multiple of the original data size while the Fast-KRR needs to store another $n/m \times n/m$ matrix. We made 25 GB of memory available to the Fast-KRR method, which allowed us to go down to 26 partitions on our cluster. The results of the comparison are shown in Figure 8 (mean over 10 runs with randomly chosen training and test sets).

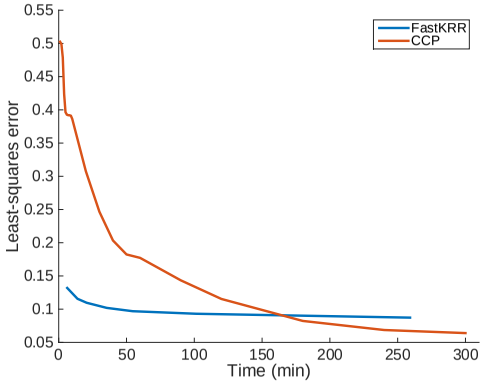


Figure 8: The plot shows the performance of the Fast-KRR and cpp-regressor in dependence of the run-time. We used partition sizes 26, 32, 38, 48, 64, 96, 128 and 256 for Fast-KRR. 26 was the limit we could achieve with 25 GB of memory.

We also determined the standard deviations. These were for both regressors marginal and we did not plot error-bars (maximal standard deviation for Fast-KRR: 0.16×10^{-4}; CCP: 0.02).

The plot shows us that the Fast-KRR achieves already a very good performance if the number of samples per partition is small. The CCP-regressor is significantly outperformed at that stage. With more run-time the CCP-regressor catches up and approaches the minimizer in the set $\mathcal{C}(100000)$. After about three hours of run-time the CCP-regressor overtakes the Fast-KRR and achieves the lowest least-squares error. Another difference one can observe is that there is, in principle, no limit of how long we can let the CCP-regressor optimize while we are limited with the Fast-KRR by the memory that we have available (the memory requirement increases linearly in the number of elements per partition and quadratically in the sample size).

5. Discussion and Open Problems

Motivated by the recent work on kernel herding we explored the uses of herding and the CGM algorithm for calculating projections onto convex sets. We derived a theorem that extends current convergence results for herding to the boundary if the perturbations introduced by our lack of knowledge of the minimal face that contains the projection are in a certain way well behaved. We also showed that our condition on the perturbations is both necessary and sufficient for the algorithm to converge with the fast rate. An important open question is if there exist convex sets for which this assumption can fail to hold. Furthermore, we demonstrated that the herding algorithm and the CGM will chose no elements outside of the minimal face after some finite time t_0 if this condition is fulfilled. Providing tight bounds on the size of t_0 and the number of elements outside the minimal face that are chosen in these first iterations in future research would help with improving the understanding of herding and the CGM further. Such bounds should also be useful to provide a better understanding of algorithms like the CGM with away steps: away steps are essentially trying to remove the perturbations introduced through elements outside of the minimal face. It is known that the CGM with away-steps successfully removes perturbations after finite many steps under certain conditions (e.g. Wolfe (1970); Guélat and Marcotte (1986)), though, as we discussed in the introduction, there is still much to be gained by refined analyses. Another interesting direction for future research concerns the geometry of the convex set in Hilbert space. The geometry factors into the rates of convergence of CGM like algorithms in various ways (Slater's condition, pyramidal width etc.). One might also aim for results that hold for convex sets with countably infinite or uncountable many extremes. We believe that one of the main difficulties will be here that there does not need to exist a gap between the minimal face and extremes that lie outside of the minimal face. We made use of such a gap to show that after some finite time only extremes of the minimal face will be chosen. Without such a gap one expects that there does not exist such a finite window of steps and perturbations will be continuously injected into the approximation.

On the practical side one can wonder how widely applicable the projection in kernel space approach, which we used to derive a novel kernel regressor, is. Projections onto the standard simplex have proven to be very valuable for a variety of statistical problems and the projection onto the set $\{\sum_{i=1}^{n} \alpha_i k(x_i, \cdot) : n \in \mathbb{N}, \|\alpha\|_{\ell_1} \leq 1, x_i \in X \text{ for } i \leq n\}$ is in a way the natural analogue of the standard simplex. There are some obvious differences. The set is, for example, from a geometrical point of view far more complicated since in kernel space the natural way to describe objects is through the kernel function and not through an orthonormal basis. Nevertheless, our results suggest that this approach has merit and it is worth to explore its uses for other non-parametric problems.

6. The Proofs of the Two Theorems and the Corollaries

This section contains the proof details of our two theorems and the various corollaries. We start with a technical lemma that is needed in the proof of the first theorem. Using this we prove that the basic algorithm without line-search achieves a convergence rate of $1/t$.

6.1 The First Theorem

We need a technical lemma that guarantees us that if we have two operators A, B that pull elements x towards the origin if x gets large (in a certain sense) then compositions of these operators (for instance $C = A^2BA^4B^3$) applied to x will be bounded, i.e. $\|Cx\| \leq b$ for some constant b. In fact, we need slightly more. We need a result which holds simultaneously for a family of operators $\mathfrak{A}$. We formulate the lemma in the way that some initial value x_0 and a particular infinite composition C of operators in $\mathfrak{A}$ and B is given. This is necessary since our operators in $\mathfrak{A}$ are not necessarily a contraction for arbitrary elements but only for the elements they are applied to. In the following, we denote with C_t the composition operator that consists of the first t operators in C (in the above example $C_3 = A^2B$) and we use the notation $C_{s,t}$ for the operator that fulfills $C_t = C_{s,t} \circ C_{s-1}$. We split the proof of the lemma and the following theorems into a number of separate claims and we use ❡ (proof) and ❑ (q.e.d.) brackets for the proofs of the claims.

Lemma 9 *Let $\mathcal{H}$ be a Hilbert space, K a closed subspace of $\mathcal{H}$, P the projection onto K, $B : \mathcal{H} \to \mathcal{H}$, $Bx = (I - P)x + PBPx$ an operator, $\mathfrak{A}$ a family of operators $A : \mathcal{H} \to \mathcal{H}$, C a finite composition of operators of $\mathfrak{A}$ and B, and $x_0 \in \mathcal{H}$ such that there exist constants $a, b, \xi > 0$ with*

(i) $\|Ax\|, \|Bx\| \leq \|x\| + b$ *for every* $A \in \mathfrak{A}$, $x \in \mathcal{H}$.

(ii) For any element $x \in \{C_t x_0\}_{t \geq 1}$ and for any $t \geq 1$ for which $C_{t,t} \in \mathfrak{A}$, i.e. an operator in $\mathfrak{A}$ is chosen at time t, we have with $A = C_{t,t} \in \mathfrak{A}$ that

$$\|Ax\|^2 \leq \|x\| (\|x\| - \xi) + b.$$

(iii) If $\|Px\| \geq a$ for an element $x \in \mathcal{H}$ then $\|Bx\| \leq \|x\|$.

Then for all $t \geq 1$

$$\|C_t x_0\|^2 \leq \|x\|^2 \vee (c + b)^2 + (a + b)^2 \quad \text{and} \quad \|C_t x_0\| \leq \|x\| \vee (c + b) + a + b,$$

with the constant $c := ((a + b)^2 + b)/\xi$.

Proof **(a)** For any $n \geq 0$, $x \in \mathcal{H}$, it holds that $\|B^n x\|^2 \leq \|x\|^2 + (a + b)^2$.

❡ Observe that $Bx = (I - P)x + BPx$. $\|Px\| < a$ implies $\|BPx\| \leq a + b$ and if $\|Px\| \geq a$ then

$$\|BPx\|^2 = \|Bx\|^2 - \|(I - P)x\|^2 \leq \|x\|^2 - \|(I - P)x\|^2 = \|Px\|^2$$

and $\|BPx\| \leq \|Px\| \vee (a + b)$. Hence,

$$\|B^n Px\| \leq \left\|PB^{n-1}x\right\| \vee (a + b) = \left\|B^{n-1}Px\right\| \vee (a + b) \leq \|Px\| \vee (a + b).$$

So for any $n \geq 0$ and $x \in \mathcal{H}$ we know that

$$\|B^n x\|^2 = \|(I - P)x\|^2 + \|B^n Px\|^2 \leq \|(I - P)x\|^2 + \|Px\|^2 \vee (a+b)^2 \leq \|x\|^2 + (a+b)^2. \quad ❑$$

(b) For any $n \geq 0$, $A \in \mathfrak{A}$, if $\|x\| \geq c + b$, $x = C_t x_0$ and $C_{t+1,t+n+1} = AB^n$ for some $t \geq 0$ then $\|AB^n x\| \leq \|x\|$ and $\|AB^n x\| \leq \|x\| \vee (c + b)$.

P If $\|x'\| \geq c$ then (ii) tells us that

$$\|Ax'\|^2 \leq \|x'\|^2 - \xi \|x'\| + b \leq \|x'\|^2 - (a + b)^2.$$

So, if $\|B^n x\| \geq c$ then (a) tells us that $\|AB^n x\|^2 \leq \|B^n x\|^2 - (a+b)^2 \leq \|x\|^2$. And, if $\|B^n x\| < c$ then $\|AB^n x\| \leq c + b$. Also, $\|AB^n x\| \leq \|x\|$ if $\|x\| \geq c + b$. **Q**

(c) For any $n \geq 0, m \geq 1$, $A_1, \ldots, A_m \in \mathfrak{A}$, if $\|x\| \geq c + b$, $x = C_t x_0$, $C_{t+1,t+n+m} = A_1 \ldots A_m B^n$ for some $t \geq 1$ then $\|A_1 \ldots A_m B^n x\| \leq \|x\|$ and it holds that $\|A_1 \ldots A_m B^n x\| \leq \|x\| \vee (c + b)$.

P If $\|z\| \geq b/\xi$ for a $z \in \mathcal{H}$ then $\|Az\| \leq \|z\|$ for all $A \in \mathfrak{A}$. If $\|x\| \geq c + b$ then (b) tells us that $\|AB^n x\| \leq \|x\|$ for all $A \in \mathfrak{A}$. Hence, for any $2 \leq l \leq m - 1$, $\|A_{m-l-1} \ldots A_m B^n x\| > \|A_{m-l} \ldots A_m B^n x\|$ implies $\|A_{m-l} \ldots A_m B^n x\| < b/\xi \leq c$. The maximal increase of operators $A \in \mathfrak{A}$ is bounded by b due to (i). Since we can only see an increase if $\|A_{m-l} \ldots A_m B^n x\| \leq c$ we gain $\|A_{m-l-1} \ldots A_m B^n x\| \leq (c + b) \vee \|x\| = \|x\|$. A slight modification of the argument yields the second case. **Q**

(d) It follows that compositions of such sequences, say $A_1 \ldots A_{m_1} B^{n_1} A_{m_1+1} \ldots A_{m_1+m_2} B^{n_2}$, $m_1, m_2 > 0$, $A_1, \ldots, A_{m_1+m_2} \in \mathfrak{A}$ do not increase the bound since with $x' = A_{m_1+1} \ldots A_{m_2} B^{n_2} x$,

$$\|x'\| = \|A_{m_1+1} \ldots A_{m_2} B^{n_2} x\| \leq \|x\| \vee (c + b)$$

we have

$$\|A_1 \ldots A_{m_1} B^{n_1} x'\| \leq \|x'\| \vee (c + b) \leq \|x\| \vee (c + b).$$

These cases cover all possible compositions with the only exception of sequences that start with some B^n, $n \geq 1$. But, if $\|x'\| \leq \|x\| \vee (c + b)$ then from (a) we know that

$$\|Cx\|^2 = \|B^n x'\|^2 \leq \|x'\|^2 + (a + b)^2 \leq \|x\|^2 \vee (c + b)^2 + (a + b)^2.$$

■

We need the definitions of the affine hull and the affine dimension. The *affine hull* of a set $A \subseteq \mathcal{H}$ is

$$\operatorname{aff} A = \left\{ \sum_{i=1}^{n} \alpha_i x_i : n \geq 1, x_i \in A, \sum_{i=1}^{n} \alpha_i = 1 \right\}.$$

The affine hull can be identified with a vector space by centering it around an arbitrary element $x_0 \in \operatorname{aff} A$, i.e. $V = \{x - x_0 : x \in \operatorname{aff} A\}$ is a vector space. The dimension of this vector space is called the *affine dimension* of A. If $A = \{x\}$ for some element $x \in \mathcal{H}$ then $\operatorname{aff} A$ is also just $\{x\}$ and we define its affine dimension to be 0.

We recall some basic properties of the projection Py of y onto a compact convex set C. The projection Py is characterized by $\|y - Py\| = \min_{x \in C} \|y - x\|$ and the geometric property $\langle y - Py, x - Py \rangle \leq 0$ for every $x \in C$. The latter property translates into a sort of orthogonal decomposition for convex sets, that is

$$\|y - x\|^2 = \|y - Py\|^2 - 2\langle y - Py, x - Py \rangle + \|Py - x\|^2 \geq \|y - Py\|^2 + \|Py - x\|^2. \quad (1)$$

We also need the definition of a face of a convex set C. A *face* F is a set which fulfills that whenever there are two points $a, b \in C$ and $\theta \in (0, 1)$ with $\theta a + (1 - \theta) b \in F$ then $a, b \in F$.

Theorem 2 *Given a compact convex set $C \subset \mathcal{H}$ and a finite subset S of C with $\operatorname{ex} C \subseteq S$ there exists for $y \in \mathcal{H}$ a constant $b > 0$ such that Algorithm 1 has a worst-case approximation error of b/t for all $t \geq 1$ if, and only if, Assumption 1 is fulfilled for the sequences $\{x_t\}_{t \geq 1}$ and $\{w_t\}_{t \geq 0}$ generated by the algorithm.*

Proof We start with some general observations before proving that the assumption is sufficient for the fast rate of convergence. That the condition is necessary is shown at the end (see (v)).

(i) The algorithm converges with the right rate to Py if the sequence $\{w_t - y - t(y - Py)\}_{t \geq 1}$ stays in a bounded norm ball of radius b since then

$$b \geq \|w_t - y - t(y - Py)\| = \left\| (t + 1)y - \sum_{i=1}^{t} x_i - y - t(y - Py) \right\| = \left\| tPy - \sum_{i=1}^{t} x_i \right\| \quad (2)$$

because $w_t = (t + 1)y - (x_1 + \ldots + x_t)$ and hence $\|Py - (x_1 + \ldots + x_t)/t\| \leq b/t$. Also observe that we can replace for any $z \in \mathcal{H}$

$$\arg\max_{x \in S} \langle z, x \rangle \quad \text{with} \quad \arg\max_{x \in S} \langle z, x - Py \rangle$$

in the maximization step.

(ii) (a) There exists a minimal face F that contains Py which is

$$F = \{Py\} \cup \{x \in C : \exists z \in C, \theta \in (0, 1) \text{ such that } \theta x + (1 - \theta) z = Py\}. \quad (3)$$

(b) F is a compact and convex set and the extremes of F are $\operatorname{ex} F = \operatorname{ex} C \cap F$.

Let $F_c = \{z - Py : z \in F\}$ be the face F centered around $Py \in F$. The set $\operatorname{aff} F_c = \operatorname{span} F_c$ is a closed subspace of $\mathcal{H}$ with orthogonal complement $(\operatorname{aff} F_c)^\perp = F_c^\perp$. So to any element $z \in \mathcal{H}$ we have a unique $z^{||} \in \operatorname{aff} F_c$ and $z^\perp \in (\operatorname{aff} F_c)^\perp$ with $z = z^{||} + z^\perp$. In the following, denote the projection onto $\operatorname{aff} F_c$ with P_F.

(c) $y - Py$ stands orthogonal on the centered face F_c, i.e. $y - Py \in F_c^\perp$. So the term $t(y - Py)$ contained in w_t does not influence the maximization over F.

(d) Either $F_c = \{0\}$ or there exists a $\delta_F > 0$ such that $B(0, \delta_F) \cap \operatorname{span} F_c = B(0, \delta_F) \cap F_c$, where $B(0, \delta_F)$ is a the closed ball of radius δ_F centered at the origin.

(e) For any w_t, $t \geq 0$, there exists an $x \in C$ such that $\langle x - Py, w_t \rangle \geq 0$. Furthermore, if $F_c \neq \{0\}$ and whenever $\|P_F w_t\| \geq 2r^2/\delta_F$, $r := \sup_{x \in C} \|x\| < \infty$, and an element in $S \cap F$ is chosen by the algorithm then $\|P_F w_{t+1}\| \leq \|P_F w_t\|$. So, if $\|P_F w_t\| \geq 2r^2/\delta_F$ and an element in $S \cap F$ is chosen then $\|w_{t+1} - y - (t + 1)(y - Py)\| \leq \|w_t - y - t(y - Py)\|$.

(f) If $F = C$ then $b = 2r^2/\delta_F + 3r$ satisfies eq. 2 and the error of the algorithm is bounded by $(2r^2/\delta_F + 3r)/t$ for any $t \geq 1$.

P (a) *F is the minimal face.* F is certainly a subset of any face that contains Py by the very definition of a face. It is also a face itself since if for any $x \in F$, $x \neq Py$, there exist two points $a, b \in C$ and a $\lambda \in (0, 1)$ such that $x = \lambda a + (1 - \lambda) b$ then there exists a $\theta \in (0, 1)$ and a $z \in C$ such that $Py = \theta x + (1 - \theta) z = \theta(\lambda a + (1 - \lambda) b) + (1 - \theta) z = \theta\lambda a + \theta(1 - \lambda) b + (1 - \theta) z = \theta\lambda a + (1 - \theta\lambda) \left(\frac{\theta(1-\lambda)}{1-\theta\lambda} b + \frac{1-\theta}{1-\theta\lambda} z \right)$. $c := \theta(1 - \lambda)/(1 - \theta\lambda) b + (1 - \theta)/(1 - \theta\lambda) z$ is an element of C since it is a convex combination of b and z and hence with $\xi := \theta\lambda \in (0, 1)$ we see that $Py = \xi a + (1 - \xi) c$ and $a \in F$. Applying the same argument to b shows us also that $b \in F$ and F is a face.

(b) *F is compact and convex and* $\operatorname{ex} F = \operatorname{ex} C \cap F$. F is a face and as such also convex. Furthermore, $F = C \cap \operatorname{aff} F$ where $\operatorname{aff} F$ is the affine hull of F. The affine hull has finite affine dimension since C has finite affine dimension. Affine hulls with finite affine dimension are closed. Hence, F is closed as the intersection of two closed sets. And since closed subsets of compact sets are compact we see that F is compact. Finally, $\operatorname{ex} F = F \cap \operatorname{ex} C$, that is the extremes of F are just the extremes of C which lie in F. The last property can be verified in the following way: the extremes of C are by definition not convex combinations of any two points $a, b \in C$, $a \neq b$, and hence of no two points in F. So $F \cap \operatorname{ex} C \subseteq \operatorname{ex} F$. On the other hand, if $c \in \operatorname{ex} F$ then there is no point $a \in C \setminus F$ for which a corresponding point $b \in C$ and $\theta \in]0,1[$ exists such that $c = \theta a + (1-\theta)b$ (a would otherwise be in the face F). However, since $c \in \operatorname{ex} F$ there exists also no point $a \in F$ with this property and $c \in \operatorname{ex} C$.

(c) *$y - Py$ is orthogonal to F_c.* If there would be an element $u \in \operatorname{aff} F_c$, $u = \sum_{i=1}^n \alpha_i v_i$ for some $n \geq 1$, $v_i \in F_c$, $\sum_{i=1}^n \alpha_i = 1$, such that $\langle u, y - Py\rangle \neq 0$ there would also be an element $v \in F_c$ with $\langle v, y - Py\rangle \neq 0$ (otherwise $\langle u, y - Py\rangle = \sum_{i=1}^n \alpha_i \langle v_i, y - Py\rangle = 0$). Furthermore, by the characterization of the minimal face in (3) and because $v + Py \in F$ there exists an element $z \in F_c$ such that $Py = \theta(v + Py) + (1-\theta)(z + Py)$ and $0 = \theta v + (1-\theta)z$. Hence, $0 = \langle y - Py, \theta v\rangle + \langle y - Py, (1-\theta)z\rangle$. Because $\langle y - Py, v\rangle \neq 0$ we know that $\langle y - Py, z\rangle \neq 0$ and both have opposing signs. So one of them, say without loss of generality $\langle y - Py, v\rangle$, is strictly greater zero. But this can not be since then $\tilde{v} = v + Py \in C$ would fulfill, in contradiction to the properties of Py, $\langle y - Py, \tilde{v} - Py\rangle > 0$.

(d) *$F_c = \{0\}$ or there exists a $B(0,\delta), \delta > 0$.* If the minimal face consists of a single element x then this element must be Py, since $Py \in F$, so in this case $F_c = \{0\}$. In the other case it suffices to consider a set of basis vectors which span the space $\operatorname{span} F_c$, say $e_1, \ldots, e_d$, where $d < \infty$ is the affine dimension of F. There exist $\alpha_1, \ldots, \alpha_d > 0$ such that $\pm\alpha_i e_i \in F_c$ for all $i \leq d$ if, and only if, $B(0,\delta) \cap \operatorname{span} F_c = B(0,\delta) \cap F_c$ for some $\delta > 0$.

Since, $e_i \in \operatorname{aff} F_c$ there exists a $n \geq 1$, $u_1, \ldots, u_n \in F_c$ and $\sum_{j=1}^n \beta_j = 1$ with $e_i = \sum_{j=1}^n \beta_j u_j$. We can also represent e_i as a sum with only non-negative coefficients, i.e. there exists $\tilde{\beta}_j \geq 0, \tilde{u}_j \in F_c$, $j \leq n$, such that $e_i = \sum_{j=1}^n \tilde{\beta}_j \tilde{u}_j$. This can be achieved by doing the following for each $j \leq n$: if $\beta_j \geq 0$, then let $\tilde{\beta}_j = \beta_j$ and $\tilde{u}_j = u_j$. If $\beta_j < 0$ and $u_j = 0$ then let $\tilde{\beta}_j = -\beta_j$ and $\tilde{u}_j = u_j$. Finally, if neither case applies ($\beta_j < 0$, $u_j \neq 0$) take a $\theta \in]0,1[$ and a $\tilde{u}_j \in F_c$ such that $0 = \theta u_j + (1-\theta)\tilde{u}_j$. Such a θ and $\tilde{u}_j$ exist due to the definition of F. With this choice $\beta_j u_j = -\tilde{u}_j \beta_j (1-\theta)/\theta = \tilde{\beta}_j \tilde{u}_j$, where $\tilde{\beta}_j = -\beta_j(1-\theta)/\theta > 0$.

So $e_i = \sum_{j=1}^n \tilde{\beta}_j \tilde{u}_j$ and $\tilde{\beta}_j \geq 0$. By normalizing the sum with $\xi = \sum_{j=1}^n \tilde{\beta}_j > 0$ (some β_j must be greater 0) we see that $\xi e_i \in F_c$. Furthermore, there exists a $\theta \in]0,1[$ and a $v \in F_c$ such that $-e_i \xi(1-\theta)/\theta = v$. And our claim is shown to be true by letting $\alpha_i = \min\{\xi(1-\theta)/\theta, \xi\}$.

(e) *Shrinking weight vector.* If F consists of more than one element then for any weight vector w_t there exists an element x in F with $\langle x - Py, P_F w_t\rangle = \delta_F \|P_F w_t\|$, since $\delta_F P_F w_t / \|P_F w_t\|$ is in $B(0,\delta_F)$ and $\langle(\delta_F P_F w_t/\|P_F w_t\| + Py) - Py, P_F w_t\rangle = \delta_F \|P_F w_t\|$. This implies directly that we have an element $z \in \operatorname{ex} F \subseteq S \cap F$ with $\langle z - Py, P_F w_t\rangle \geq \delta_F \|P_F w_t\|$, because $x = \sum_{j=1}^n \beta_j u_j$, for suitable n, $\beta_1, \ldots, \beta_n > 0$, $\sum_{j=1}^n \beta_j = 1$ and elements $u_j \in \operatorname{ex} F$ ($\operatorname{ex} C$ is finite and the convex hull C of $\operatorname{ex} C$ is in this case closed. That means any element in C can be represented exactly by a convex combination of finite many extremes), i.e. $\sum_{j=1}^n \beta_j \langle u_j - Py, P_F w_t\rangle = \delta \|P_F w_t\|$ and at least one term $\langle u_j - Py, P_F w_t\rangle$ must be as large as $\delta_F \|P_F w_t\|$.

Now, whenever $\|P_F w_t\| \geq 2r^2/\delta_F$ and an element in $S \cap F$ is chosen, then $\|P_F w_{t+1}\| \leq \|P_F w_t\|$. This can be verified in the following way: there exists an element $x' \in F \cap S$ such that $\langle x' - Py, P_F w_t\rangle \geq \delta_F \|P_F w_t\|$ and, hence, the algorithm will choose an element x with $\langle x - Py, P_F w_t\rangle \geq \langle x' - Py, P_F w_t\rangle \geq \delta_F \|P_F w_t\|$ (as said in (i) the maximization is unaffected by the translation $-Py$). So, since $x - Py \in F_c$ and P_F is linear we have that

$$\begin{aligned}\|P_F w_{t+1}\|^2 &= \|P_F(w_t - (x - Py))\|^2 = \|P_F w_t - (x - Py)\|^2 \\ &= \|P_F w_t\|^2 - 2\langle x - Py, P_F w_t\rangle + \|x - Py\|^2 .\end{aligned}$$

Furthermore, $\|x - Py\| \leq \|x\| + \|Py\|$ is bounded by $2r$ and

$$\|x - Py\|^2 - 2\langle x - Py, P_F w_t\rangle \leq 4r^2 - 2\delta_F \|P_F w_t\| \leq 0$$

by our assumption on $\|P_F w_t\|$.

For $x \in F \cap S$ we can also observe that for any w_t

$$\langle x - Py, w_t\rangle = \langle P_F(x - Py), w_t\rangle = \langle x - Py, P_F w_t\rangle$$

since $x - Py \in \operatorname{aff} F_c$ and P_F is self-adjoint. So there always exists an $x \in C$ such that $\langle x - Py, w_t\rangle \geq 0$. If $F_c = \{0\}$, that is $F = \{Py\}$, then $x = Py$ gives us $\langle x - Py, w_t\rangle = 0$.

(f) If $F = \{Py\}$ then $w_t - y - t(y - Py) = (x_1 - Py) + \ldots + (x_t - Py) = 0$ since all $x_i = Py$. If F contains more than one element and $C = F$ then (e) tells us that if $\|P_F w_t\| \geq 2r^2/\delta_F$ then

$$\|w_{t+1} - y - (t+1)(y - Py)\| \leq \|w_t - y - t(y - Py)\| .$$

Observe that if $F = C$ then the projection P_F is closely related to the projection $\tilde{P}$ onto the affine hull of C, that is $\tilde{P}x = P_F(x - \tilde{P}0) + \tilde{P}0$. Also, $\tilde{P}y = Py$ because $y - Py$ is orthogonal to $\operatorname{span} F_c$. We need to show that $\|P_F w_t\| \geq 2r^2/\delta_F$ under the stated condition. We have that

$$\|P_F w_t\| = \Big\|(t+1)P_F y - \sum_{i=1}^t P_F x_i\Big\| = \Big\|P_F y - \sum_{i=1}^t P_F(x_i - y)\Big\| = \Big\|P_F y - \sum_{i=1}^t (x_i - Py)\Big\|$$

because $x_i - Py$ lies in $\operatorname{span} F_c$ and $P_F y = P_F Py$. On the other hand

$$\|w_t - y - t(y - Py)\| = \Big\|\sum_{i=1}^t (x_i - Py)\Big\| \leq \|P_F w_t\| + \|P_F Py\| \leq \|P_F w_t\| + r.$$

Hence, if $\|w_t - y - t(y - Py)\| \geq r + 2r^2/\delta_F$ then $\|w_{t+1} - y - (t+1)(y - Py)\|$ is smaller than $\|w_t - y - t(y - Py)\|$. So we can only see an increase of the sequence $\{w_t - y - t(y - Py)\}_{t \geq 1}$ if at any given t it holds that $\|w_t - y - t(y - Py)\| < 2r^2/\delta_F + r$ and since the change between t and $t+1$ is just $Py - x$ we gain for any $t \geq 1$ the bound

$$\|w_t - y - t(y - Py)\| < 2r^2/\delta_F + r + \|Py - x\| \leq 3r + 2r^2/\delta_F . \qquad \blacksquare$$

(iii) The case $F = C$ is dealt with in (ii.f) so we may assume in the following that $C \setminus F \neq \emptyset$.

We assign to the three subspaces U_F, U_b and U_δ defined in Section 2.1 orthonormal bases. Take an arbitrary orthonormal basis of U_F and let E_F be the set of these basis elements. Similarly, chose

a basis of U_δ and let E_δ be the set of these basis elements. Finally, group the basis elements of U_b, guaranteed to us in our assumption, in E_b. Furthermore, let δ_m be the largest radius of an open ball around 0 in $U_F \oplus U_\delta$. δ_m is always strictly larger than 0. For any $e \in E_b$ and $t \geq 0$

$$\langle w_t, e\rangle = \langle Py, e\rangle + \langle (t+1)(y-Py), e\rangle - \sum_{n=1}^{t} \langle x_n - Py, e\rangle \geq \langle (t+1)(y-Py), e\rangle + \langle Py, e\rangle$$

and for any $e \in E_b$ and any $t \geq 0$ it holds that

$$\langle w_{t+1} - (y-Py), e\rangle - \langle w_t, e\rangle = -\langle x_{t+1} - Py, e\rangle \geq 0.$$

Hence $\langle w_{t+1} - (y-Py), e\rangle \geq \langle w_t, e\rangle$ for all $e \in E_b$.

Clearly, F_c lies in $U_F \oplus U_\delta$. More importantly, any element that is not in F_c lies at least partly in U_b, or in other words, if $x \in C\backslash F$ then $P_b(x-Py) \neq 0$.

P For any element $x \in C$ that is not in F there exists an $e \in E_b$ such that $\langle x - Py, e\rangle \neq 0$. Otherwise, $x - Py$ would lie in a subspace A for which there is a $\delta' > 0$ and $B(0,\delta') \cap A = B(0,\delta') \cap A \cap C_c$. The element $-(x-Py)$ would then also lie in A and $z = -(\delta'/2)(x - Py)/\|x - Py\| \in B(0,\delta') \cap A = B(0,\delta') \cap A \cap C_c$ ($x \neq Py$, since $Py \in F$ and the norm $\|x - Py\|$ is strictly positive). So with $\zeta = \delta'/(2\|x-Py\|)$ and $\xi = \zeta/(1+\zeta) \in]0,1[$ we would have that

$$\xi(x-Py) + (1-\xi)z = (x-Py)\zeta/(1+\zeta) - (x-Py)\zeta(1-\zeta/(1+\zeta)) = 0$$

and $\xi x + (1-\xi)(z+Py) = Py$. But, because x and $z + Py \in C$ this implies that $x \in F$ by the definition of the minimal face. The conclusion $x \in F$ is a contradiction to our original assumption and our claim holds. **Q**

(iv) (a) If the sequence $\{\|(P_F + P_\delta)(w_t - y - t(y-Py))\|\}_{t\geq 1}$ is bounded then the sequence $\{\|P_b(w_t - y - t(y-Py))\|\}_{t\geq 1}$ is also bounded and elements in $S\backslash F$ are chosen only finitely often.

(b) Under Assumption 1 the sequence $\{\|(P_F + P_\delta)(w_t - y - t(y-Py))\|\}_{t\geq 1}$ is bounded. Furthermore, under this assumption the sequence $\{\|w_t - y - t(y-Py)\|\}_{t\geq 1}$ is bounded and there exists a constant b such that the approximation error is bounded by b/t.

P (a) Let us assume that $\{\|P_b(w_t - y - t(y-Py)\|\}_{t\geq 1}$ is unbounded. If elements in F are chosen at any stage t then $P_b(w_t - y - t(y-Py)) = P_b(w_{t+1} - y - (t+1)(y-Py))$ and there is no increase of the normed sequence. Hence, there must exist an element $x^* \in S\backslash F$ which is selected infinitely often. Let $e = -P_b(x^* - Py)/\|P_b(x^* - Py)\|$ and observe that $\langle e, x - Py\rangle \leq 0$ for all $x \in C$ because $\langle P_b(x^* - Py), x - Py\rangle = \sum_{e' \in E_b} \langle e', x^* - Py\rangle \langle e', x - Py\rangle \geq 0$. If x^* is selected at any step t then from $\langle x^* - Py, w_t\rangle \geq 0$ it follows that

$$\langle x^* - Py, (P_F + P_\delta)w_t\rangle \geq -\langle x^* - Py, P_b w_t\rangle = \|P_b(x^* - Py)\| \langle e, w_t\rangle$$

and $\langle x^* - Py, y - Py\rangle \leq 0$ implies $\langle x^* - Py, (P_F + P_\delta)(y - Py)\rangle \leq -\langle x^* - Py, P_b(y - Py)\rangle$ which in turn implies

$$\begin{aligned}\langle x^* - Py, (P_F + P_\delta)(-y - t(y-Py))\rangle &\geq -\langle x^* - Py, P_b(-y - t(y-Py))\rangle - \langle x^* - Py, Py\rangle \\ &\geq -\langle P_b(x^* - Py), P_b(-y - t(y-Py))\rangle - 2r.\end{aligned}$$

Together, these inequalities give us

$$\langle x^* - Py, (P_F + P_\delta)(w_t - y - t(y-Py))\rangle \geq \|P_b(x^* - Py)\| \langle e, P_b(w_t - y - t(y-Py))\rangle - 2r.$$

Applying the Cauchy-Schwarz inequality yields then

$$\begin{aligned}&\frac{\|x^* - Py\|}{\|P_b(x^* - Py)\|} \|(P_F + P_\delta)(w_t - y - t(y-Py))\| + \frac{2r}{\|P_b(x^* - Py)\|} \geq \langle e, w_t - y - t(y-Py)\rangle \\ &\geq -\sum_{s=1}^{t} \langle e, x_s - Py\rangle \times \chi\{x_s = x^*\} = \sum_{s=1}^{t} \|P_b(x^* - Py)\| \times \chi\{x_s = x^*\}\end{aligned}$$

where χ is the characteristic function. Since $\|P_b(x^* - Py)\| > 0$ and x^* is selected infinitely often we conclude that $\|(P_F + P_\delta)(w_t - y - t(y-Py))\|$ diverges in t and the corresponding sequence $\{\|(P_F + P_\delta)(w_t - y - t(y-Py)\|\}_{t\geq 1}$ is unbounded.

(b) If $U_F = U_\delta = \{0\}$ then $(P_F + P_\delta)x = 0$ for all $x \in \mathcal{H}$ and (a) gives us

$$\|w_t - y - t(y-Py)\| = \Big\|\sum_{i=1}^{t}(x_i - Py)\Big\| = \Big\|\sum_{i=1}^{t} P_b(x_i - Py)\Big\| = \|P_b(w_t - y - t(y-Py)\|$$

and the sequence is bounded due to (a). Hence, the result follows.

Now let us assume that $U_F \oplus U_\delta \neq \{0\}$. We want to apply Lemma 9. Let $\tilde{\mathcal{H}} := U_F \oplus U_\delta$. $\tilde{\mathcal{H}}$ together with the inner product inherited from $\mathcal{H}$ is a Hilbert space. Let A be the operator defined for all $w \in \mathcal{H}$ through

$$x' \in \operatorname*{arg\,max}_{x \in S\backslash F} \langle x - Py, w\rangle \quad \text{and} \quad Aw = w + y - x'.$$

For any w for which there exist multiple maximizer assign to Aw one of these maximizer. Let B be defined in the same way with the only difference that $S\backslash F$ is replaced by $S \cap F$. We like to study the interaction between A and B on the subspace $U_F \oplus U_\delta$. The operator B optimizes over elements in $S \cap F$. For such elements $x \in S \cap F$ we know that $x - Py$ is orthogonal to $P_b w$ for all possible weights w and $P_b w$ does not influence the behavior of the operator B. Let $\tilde{B}$ be $(P_F + P_\delta)B$ restricted to $\tilde{\mathcal{H}}$ and let $K := U_F \subseteq \tilde{H}$. The operator $\tilde{B}$ fulfills the assumptions of the lemma:

- $\tilde{B}$ leaves the space orthogonal to K in $\tilde{H}$ unchanged and the maximization over $S \cap F$ does only depend on $P_F w$ for any $w \in \tilde{\mathcal{H}}$ so $\tilde{B}w = (I - P_F)w + P_F \tilde{B} P_F w$. This also holds (trivially) if $F_c = \{0\}$.
- Let $b := 2r$ then for $w \in \tilde{\mathcal{H}}$, $\|\tilde{B}w\| = \|(P_F + P_\delta)(w + (y - Py) - (x - Py))\| \leq \|w\| + \|x - Py\| \leq \|w\| + b$ where $x \in S \cap F$ and $(P_F + P_\delta)(y - Py) = 0$ by our choice of U_δ.
- If $F_c = \{0\}$ then point (iii) in Lemma 9 holds trivially. In all other cases let $a := 2r^2/\delta_F$, $\delta_F > 0$, then an argument as in (ii.e) tells us that if $\|P_F w\| \geq a$ for some $w \in \tilde{\mathcal{H}}$ then $\|\tilde{B}w\| \leq \|w\|$.

The operator A depends on $P_b w$ and we need to account for this error when studying the behavior of A acting on $U_F \oplus U_\delta$. We do so by working with a family of operators A which are parameterized by $P_b w$. Let

$$\mathfrak{A} = \{\tilde{A}_v : \tilde{\mathcal{H}} \to \tilde{\mathcal{H}} : \tilde{A}_v(w) = (P_F + P_\delta)A(w+v) \text{ for all } w \in \tilde{\mathcal{H}}, v \in U_b, \langle v, e\rangle \geq 0, \forall e \in E_b\}.$$

Observe that there is some time $t'' < \infty$ such that for all $t \geq t''$ no element in $S\backslash(F \cup D(\{x_t\}))$ is chosen and each element in $D(\{x_t\})$ has been chosen at least once. Write $U_b = U \oplus U'$ with $U' = \text{span } D(\{x_t\})$ and a suitable subspace U. Then $P_U w_t = P_U w_{\tilde{t}}$ for all $t \geq \tilde{t}$. Since each element in $D(\{x_t\})$ is chosen infinitely often and $\langle -P_b w_t, x - Py\rangle$ is non-decreasing for $x \in C$ there exists some element $t''' \geq t''$ such that for all $t \geq t'''$, $\langle P_U(x-Py), -P_b w_t\rangle \leq \langle x' - Py, -P_b w_t\rangle$ for all $x \in S\backslash D(\{x_t\})$ and $x' \in D(\{x_t\})$. Let $\tilde{t} < \infty$ be the maximum over all t' in Assumption 1 and t'''. The family of operators $\mathfrak{A}$ fulfills the assumptions of the lemma if Assumption 1 holds and if we use $x_{\tilde{t}}$ as the initial value.

- Given any $w \in \tilde{\mathcal{H}}$, $\tilde{A} \in \mathfrak{A}$ (with corresponding $v \in U_b$, $\langle v, e\rangle \geq 0$ for all $e \in E_b$), there exists a $x \in S\backslash F$ such that

$$\begin{aligned}\left\|\tilde{A}w\right\| &= \|(P_F + P_\delta)A(w+v)\| = \|(P_F + P_\delta)(w + v + (y - Py) - (x - Py))\| \\ &= \|w - (x - Py)\| \leq \|w\| + b.\end{aligned}$$

- If $D(\{x_t\}) = \emptyset$ then only elements in F are chosen after $\tilde{t}$ and the claim of the theorem follows by the arguments in (ii.e).

  In case that $D(\{x_t\}) \neq \emptyset$ consider the maximum over all Δ in Assumption 1 and call this maximum $\tilde{\Delta} < \infty$. We claim that there exists a constant $\Delta' < \infty$ such that with $\xi := 2\delta_m/\Delta'$ for all w_t, $t \geq \tilde{t}$, whenever $x_t \in S\backslash F$, then

$$\langle x_t - Py, (P_F + P_\delta)w_t\rangle \geq \|(P_F + P_\delta)w_t\|\, \delta_m/\Delta'$$

  where x_t is the element selected in the argmax step of A_v, $w_t = u + v, u \in U_F \oplus U_\delta, v \in U_b$. **P** ($\alpha$) Whenever $x_t \in S\backslash F, x_t \in D(\{x_t\})$, for some $t \geq \tilde{t}$ we know that $\langle P_b w_t, x_t - Py\rangle \neq 0$ because each element $x \in D(\{x_t\})$ has been chosen at least once and each element outside the minimal face lies partly in U_b. For any $x \in D(\{x_t\})$ we also have $\langle P_b w_t, x - Py\rangle \neq 0$ and at step $t \geq \tilde{t}$

$$\begin{aligned}&\langle x_t - Py, (P_\delta + P_F)w_t\rangle \geq \langle x - Py, (P_\delta + P_F)w_t\rangle - \langle (x-Py) - (x_t - Py), -P_b w_t\rangle \\ &\geq \langle x - Py, (P_F + P_\delta)w_t\rangle - \left(\frac{\langle x - Py, -P_b w_t\rangle}{\langle x_t - Py, -P_b w_t\rangle} - 1\right)\langle x_t - Py, -P_b w_t\rangle \\ &\geq \langle x - Py, (P_F + P_\delta)w_t\rangle - \left(\frac{\langle x - Py, -P_b w_t\rangle}{\langle x_t - Py, -P_b w_t\rangle} - 1\right)\langle x_t - Py, (P_F + P_\delta)w_t\rangle \\ &\geq \langle x - Py, (P_F + P_\delta)w_t\rangle - (\tilde{\Delta} - 1)\langle x_t - Py, (P_F + P_\delta)w_t\rangle\end{aligned}$$

  and

$$\langle x_t - Py, (P_\delta + P_F)w_t\rangle \geq \langle x - Py, (P_\delta + P_F)w_t\rangle / \tilde{\Delta}.$$

  (β) Consider a set of elements $z_1, \ldots, z_{\tilde{d}} \in D(\{x_t\})$ such that $z_1 - Py, \ldots z_{\tilde{d}} - Py$ are linearly independent and with $\dim U' = \tilde{d} \geq 1$. Apply the Gram-Schmidt method to transform these into an orthonormal basis $\tilde{z}_1, \ldots, \tilde{z}_{\tilde{d}}$ such that $\tilde{z}_i = \sum_{j=1}^{i} \beta_{ij}(z_j - Py)$ for some

scalars β_{ij}. Since the vectors are linearly independent we know that $\max_{i,j\leq d} \left|\beta_{ij}\right| < \infty$. For any $x \in S\backslash(F \cup D(\{x_t\}))$ if $P_b x \in U$ then, by the choice of $\tilde{t}$, for all $t \geq \tilde{t}$

$$\langle -P_b w_t, x - Py\rangle \leq \min_{x' \in D(\{x_t\})} \langle -P_b w_t, x' - Py\rangle.$$

If $P_b x \notin U$ then we can write $P_{U'}(x - Py) = \alpha_1 \tilde{z}_1, \ldots, \alpha_{\tilde{d}} \tilde{z}_{\tilde{d}}$ for suitable scalars $\alpha_1, \ldots, \alpha_{\tilde{d}}$. We claim that there exists a $\Delta_x < \infty$ such that

$$\langle -P_b w_t, P_{U'}(x - Py)\rangle \leq \Delta_x \max_{i \leq d} \langle -P_b w_t, z_i - Py\rangle$$

for all $t \geq \tilde{t}$. This holds since

$$\begin{aligned}\langle -P_b w_t, P_{U'}(x - Py)\rangle &\leq d \max_{i\leq d} |\alpha_i| \max_{i \leq d} \langle -P_b w_t, \tilde{z}_i\rangle \\ &\leq d^2 \max_{i\leq d} |\alpha_i| \max_{i,j\leq d} \left|\beta_{ij}\right| \max_{i\leq d} \langle -P_b w_t, z_i - Py\rangle\end{aligned}$$

and, using Parseval's identity, $|\alpha_i| \leq \|P_{U'}(x - Py)\| \leq 2r$. Hence, the claim follows with $\Delta_x = 2rd^2 \max_{i,j\leq d} \left|\beta_{ij}\right| < \infty$.

Furthermore, because for all $t \geq \tilde{t}$, $x \in S\backslash(F \cup D(\{x_t\}))$, $\langle P_U(x - Py), -P_b w_t\rangle \leq \min_{i\leq d}\langle z_i - Py, -P_b w_t\rangle$ by our choice of $\tilde{t}$ we can observe that for any $x \in S\backslash(F \cup D(\{x_t\}) =: M$

$$\begin{aligned}\langle -P_b w_t, x - Py\rangle &= \langle -P_b w_t, P_U(x - Py)\rangle + \langle -P_b w_t, P_{U'}(x - Py)\rangle \\ &\leq (\sup_{x\in M} \Delta_x + 1) \max_{i\leq d} \langle -P_b w_t, z_i - Py\rangle \\ &\leq \tilde{\Delta}(\sup_{x\in M} \Delta_x + 1)\langle -P_b w_t, x_t - Py\rangle.\end{aligned}$$

Repeating the argument in (α) this tells us that

$$\langle x_t - Py, (P_\delta + P_F)w_t\rangle \geq \langle x - Py, (P_\delta + P_F)w_t\rangle / (\tilde{\Delta}(\sup_{x\in M} \Delta_x + 1)).$$

(γ) The above argument also works for $x \in F$ without the need to adapt the multiplier $\Delta' := \tilde{\Delta}(\sup_{x\in M} \Delta_x + 1)$. Now, we know that there exists an element $x^* \in S$ such that

$$\langle x^* - Py, (P_F + P_\delta)w_t\rangle \geq \|(P_F + P_\delta)w_t\|\, \delta_m$$

and, hence,

$$\langle x_t - Py, (P_F + P_\delta)w_t\rangle \geq \langle x^* - Py, (P_F + P_\delta)w_t\rangle/\Delta' \geq \|(P_F + P_\delta)w_t\|\, \delta_m/\Delta'$$

which proves the claim. **Q**

Writing $u = w + v$, $w \in U_b \oplus U_\delta$, $v \in U_b$, this becomes $\langle x_t - Py, w\rangle \geq \|w\|\, \delta_m/\Delta'$ for all $t \geq \tilde{t}$. The usual argument gives us for the operator $\tilde{A}$ being used at time t that

$$\begin{aligned}\left\|\tilde{A}w\right\|^2 &= \left\|(P_F + P_\delta)(w + v + (y - Py) - (x' - Py))\right\|^2 \\ &= \|w\|^2 - 2\left\langle x' - Py, w\right\rangle + \left\|x' - Py\right\|^2 \\ &\leq \|w\|^2 - 2\|w\|\, \delta_m/\Delta' + b = \|w\|\,(\|w\| - 2\xi) + b.\end{aligned}$$

Hence, the lemma can be applied and the sequence of weights projected onto $U_F \oplus U_\delta$, i.e. $\{(P_F + P_\delta)w_t\}_{t\geq 0}$ is bounded in norm. Together with (a) this implies that the weight sequence stays bounded and the result follows. ◻

(v) The condition is also necessary for the fast rate of convergence. Observe that there always exists a decomposition $U_F \oplus U_\delta \oplus U_b$ of span C_c: let $U_F = \operatorname{span} F_c$. Let $\tilde{U}$ be the orthogonal complement of U_F in span C_c. If this is empty then we are done. Otherwise, chose an orthonormal basis $e_1, \ldots, e_k$ of $\tilde{U}, k = \dim \tilde{U}$ such that $e_1 = P_C(y - Py)/\|P_C(y - Py)\|$ if $P_C(y - Py) \neq 0$. Consider the set $E_b = \{e_i : i \leq k, P_{e_i}[C_c] \subset (-\infty, 0]$ or $P_{e_i}[C_c] \subset [0, \infty)\}$ and let $E_\delta = \{e_1, \ldots e_k\} \backslash E_b$. Then $U_b = \operatorname{span} E_b, U_\delta = \operatorname{span} E_\delta$ fulfill the assumptions. Since $P_\delta[C_c]$ is convex and we have an open interval around 0 in each direction $e \in E_\delta$ we have a ball around 0 in U_δ. Similarly, the assumption for U_b is fulfilled if we exchange $e \in E_b$ with $-e$ whenever $P_e[C_c] \subset [0, \infty)$.

If Assumption 1 is not fulfilled for this decomposition then there exists an element $x \in S \backslash F$ which is selected infinitely often by the algorithm. But, since $\|P_b(w_t - y - t(y - Py))\|$ is non-decreasing in t and because $x \in S \backslash F$ fulfills $\|P_b(x - Py)\| > 0$, we have $\|w_t - y - t(y - Py)\| \geq \|P_b(x - Py)\| \sum_{s=1}^{t} \chi\{x_s = x\}$ and the right side diverges in t. Therefore, we have an unbounded sequence $\{\|w_t - y - t(y - Py)\|\}_{t\geq 1}$. Observe that the algorithm does not converge with the rate $1/t$ if $\{\|w_t - y - t(y - Py)\|\}_{t\geq 1}$ is unbounded. ■

6.2 The Second Theorem

Theorem 3 *Given a compact convex set $C \subset \mathcal{H}$, a finite subset S of C with $\operatorname{ex} C \subseteq S$ and an element $y \in \mathcal{H}$ the following holds:*

- *If only elements in $F \cap S$, where $F \subset C$ is the minimal face that contains Py, are chosen then the method converges linearly to the projection and there exist constants $b, \beta > 0$ with*

$$\|Py - p_t\| \leq be^{-\beta t}.$$

- *Under Assumption 1 if $\min_{x\in S} \|Py - x\| > 0$ and the approximation does not equal Py in finite many steps then the sequence $\{\|(I - P_F)(Py - p_t)\|\}_{t\geq 0}$ converges sub-linearly and there exists a constant $d > 0$ such that*

$$\|P_F(Py - p_t)\|^2 \leq (1 - \delta_F^2/\operatorname{diam}^2(F))\, \|P_F(Py - p_{t-1})\|^2 + d\, \|(I - P_F)(Py - p_{t-1})\|^2 .$$

Furthermore, there exists a time t_0 after which only elements in $F \cap S$ are chosen.

Proof We aim for a similar argument as in the proof of the first theorem: (1) optimizing the approximation of Py by using y instead of Py does not hurt the rate of convergence. (2) This guarantees us that the rate of convergence is what we aim for if we work solely with the minimal face F that contains Py. Since, we do not know this minimal face we need to deal with perturbations that are introduced by elements in $S \backslash F$. These perturbations do slow down the rate of convergence. We adapt the proof from Beck and Teboulle (2004) to address (1). We make use of parts of the proof of Theorem 2 and we use Theorem 2 (ii.a) etc. to refer to it. In the following we will use p_t to denote the approximation at step t (with $p_0 = 0$), x_t as the element that is chosen by the algorithm and $w_t = y - p_t$.

(i) Let F be the minimal face which contains Py (Thm. 2 (ii.a)) and assume that either $C = F$ or only elements in F are chosen by the algorithm. (a) We claim that in this case $\tilde{\alpha}_t \in [0, 1]$ for all $t \geq 1$. ▶ In step 1 we have by definition that $\tilde{\alpha}_t = 1$. If C consists of a single element then $\tilde{\alpha}_t = 0/0$, which we define to be 0, for all $t \geq 2$. For the case that C consists of more than a single element and for any step $t \geq 2$ we have that

$$\tilde{\alpha}_t = \frac{\langle w_{t-1}, w_{t-1} + (x_t - y)\rangle}{\|w_{t-1} + x_t - y\|^2} = \frac{\langle y - p_{t-1}, x_t - p_{t-1}\rangle}{\|x_t - p_{t-1}\|^2} = \frac{\langle Py - p_{t-1}, x_t - p_{t-1}\rangle}{\|x_t - p_{t-1}\|^2}$$

where the last step holds because $x_t - p_{t-1}$ lies in the span of C_c and $y - Py$ stands orthogonal on span C_c (Thm. 2 (ii.c)), i.e. $\langle y - Py, x_t - p_{t-1}\rangle = 0$. We can also observe that

$$\langle Py - p_{t-1}, x_t - Py\rangle = \max_{x\in C} \langle Py - p_{t-1}, x - Py\rangle \geq \delta_F \|Py - p_{t-1}\| \geq 0 \tag{4}$$

holds: $y - Py$ stands orthogonal on span C_c and, hence,

$$\arg\max_{x\in S} \langle w_{t-1}, x\rangle = \arg\max_{x\in S} \langle Py - p_{t-1}, x\rangle \subseteq \arg\max_{x\in C} \langle Py - p_{t-1}, x\rangle .$$

Furthermore, $Py - p_{t-1}$ lies in the span of the centered minimal face, i.e. $Py - p_{t-1} \in F_c$, and Thm. 2 (ii.d) tells us that there exists a constant $\delta_F > 0$ which makes the above true. Following Beck and Teboulle (2004) we complete the square

$$\begin{aligned}\langle Py - p_{t-1}, Py - p_{t-1} - (Py - x_t)\rangle &\leq \|Py - p_{t-1}\|^2 - 2\langle Py - p_{t-1}, Py - x_t\rangle + \|Py - x_t\|^2 \\ &= \|p_{t-1} - x_t\|^2\end{aligned}$$

and observe that $\tilde{\alpha}_t \leq 1$. Hence, $\alpha_t = \tilde{\alpha}_t$. ◻

(b) If $F = C$ or only elements in F are chosen by the algorithm then for any $t \geq 2$ either $Py = p_t$ and for all $s \geq t$ we have $Py = p_s$ or $\|Py - p_t\|^2 \leq (1 - \delta_F^2/\operatorname{diam}^2(C))\, \|\tilde{w}_{t-1}\|^2$. Furthermore, there exist constants $b, \beta > 0$ such that $\|Py - p_t\| \leq be^{-\beta t}$. ▶ We aim at reproducing the argument from Beck and Teboulle (2004) by exploiting the orthogonality between $y - Py$ and span C_c. Let $\tilde{w}_t = Py - p_t = \tilde{w}_{t-1} - \alpha_t(x_t - Py + \tilde{w}_{t-1})$ and assume that $\tilde{w}_{t-1} \neq 0$. In short, if C consists of a single element then there is nothing to show and, otherwise, we have for $t \geq 2$ that $\alpha_t = \tilde{\alpha}_t$ and

$$\|\tilde{w}_t\|^2 = \|\tilde{w}_{t-1}\|^2 - 2\alpha_t \langle \tilde{w}_{t-1}, \tilde{w}_{t-1} + (x_t - Py)\rangle + \alpha_t^2 \|x_t - p_{t-1}\|^2 .$$

Observe that $\langle w_t, \tilde{w}_t\rangle = \langle y - p_t, Py - p_t\rangle = \|\tilde{w}_t\|^2$ because of the orthogonality and because $Py - p_t \in \operatorname{span} C_c$. This, together with the orthogonality between $y - Py$ and $x_t - Py$ yields

$$\langle w_{t-1}, w_{t-1} + (x_t - y)\rangle = \langle w_{t-1}, Py - p_{t-1} + x_t - Py\rangle = \langle \tilde{w}_{t-1}, \tilde{w}_{t-1} + x_t - Py\rangle .$$

Furthermore, $\|w_{t-1} + x_t - y\| = \|x_t - p_{t-1}\|$ and, hence,

$$\alpha_t = \frac{\langle \tilde{w}_{t-1}, \tilde{w}_{t-1} + (x_t - Py)\rangle}{\|x_t - p_{t-1}\|^2}.$$

By filling in this value of α_t we gain

$$\|\tilde{w}_t\|^2 = \|\tilde{w}_{t-1}\|^2 - 2\frac{|\langle \tilde{w}_{t-1}, \tilde{w}_{t-1} + x_t - Py\rangle|^2}{\|x_t - p_{t-1}\|^2} + \frac{|\langle \tilde{w}_{t-1}, \tilde{w}_{t-1} + x_t - Py\rangle|^2}{\|x_t - p_{t-1}\|^2}$$
$$= \frac{\|\tilde{w}_{t-1}\|^2 \|Py - x_t\|^2 - |\langle \tilde{w}_{t-1}, Py - x_t\rangle|^2}{\|x_t - p_{t-1}\|^2}.$$

Since $\arg\max_{x\in S} \langle w_{t-1}, x\rangle = \arg\max_{x\in S} \langle \tilde{w}_{t-1}, x\rangle$ we have that

$$\langle \tilde{w}_{t-1}, x_t - Py\rangle = \max_{x\in C} \langle \tilde{w}_{t-1}, x - Py\rangle \geq \delta_F \|\tilde{w}_{t-1}\|$$

and

$$\|\tilde{w}_{t-1}\|^2 \|Py - x_t\|^2 - |\langle \tilde{w}_{t-1}, Py - x_t\rangle|^2 \leq \|\tilde{w}_{t-1}\|^2 (\|Py - x_t\|^2 - \delta_F^2).$$

Eq. 4 tells us now that $\|x_t - Py + (Py - p_{t-1})\|^2 \geq \|x_t - Py\|^2$ and

$$\|\tilde{w}_t\|^2 \leq \frac{\|\tilde{w}_{t-1}\|^2 (\|x_t - Py\|^2 - \delta_F^2)}{\|x_t - Py\|^2} \leq (1 - \delta_F^2/\mathrm{diam}^2(C)) \|\tilde{w}_{t-1}\|^2.$$

This is the second part of our claim. The first part follows directly from the particular form that α_t attains. With $\tilde{w}_{t-1} = 0$ we have

$$\alpha_t = \frac{\langle \tilde{w}_{t-1}, \tilde{w}_{t-1} + (x_t - Py)\rangle}{\|x_t - p_{t-1}\|^2} = 0$$

and $\tilde{w}_t = \tilde{w}_{t-1} = 0$. Finally, both cases imply the fast rate of convergence: Let $\gamma = \delta_F^2/\mathrm{diam}^2(C) \in (0, 1)$ then in both cases $\|\tilde{w}_t\|^2 \leq (1 - \gamma)\|\tilde{w}_{t-1}\|^2$ for all $t \geq 2$ and Lemma A.1(ii) from Beck and Teboulle (2004) tells us that $\|\tilde{w}_t\|^2 \leq e^{-\gamma t}\|\tilde{w}_1\|^2$. But then with $\beta = \gamma/2$ we have $\|Py - p_t\| \leq e^{-\beta t}\|\tilde{w}_1\|$ and the result follows. ❶

(ii) We address now $F \neq C$. First, we can observe that in the general case either $\tilde{\alpha}_t \leq 1$ holds after at most finite many steps or there is one final step where $\tilde{\alpha}_t > 1$ and the approximation error becomes zero afterwards, i.e. there exists a time $t_0 < \infty$ such that $\tilde{\alpha}_t \leq 1$ for all $t \geq t_0$ or, if $\tilde{\alpha}_t > 1$ for some $t \geq t_0$, then for all $s > t$ we have that $p_s = Py$ and $\tilde{\alpha}_s \in [0, 1]$. t_0 depends here on y and S. ❶ We expand the argument of (i.a). In the case that S does not contain the element Py we can argue in the following way: If $\|y - Py\| > 0$ and $\|p_{t-1} - Py\| \leq (\|x_t - Py\|^2 - \langle p_{t-1} - Py, x_t - Py\rangle)/\|y - Py\|$ then

$$\langle y - p_{t-1}, x_t - p_{t-1}\rangle = \|y - p_{t-1}\|^2 + \langle y - p_{t-1}, x_t - y\rangle$$
$$= \|y - p_{t-1}\|^2 + \langle y - Py, x_t - y\rangle + \langle Py - p_{t-1}, Py - y\rangle + \langle Py - p_{t-1}, x_t - Py\rangle$$
$$= \|Py - p_{t-1}\|^2 - \langle p_{t-1} - Py, x_t - Py\rangle + \langle y - Py, Py - p_{t-1}\rangle + \langle y - Py, x_t - Py\rangle$$
$$\leq \|Py - p_{t-1}\|^2 - \langle p_{t-1} - Py, x_t - Py\rangle + \|y - Py\| \|p_{t-1} - Py\|$$
$$\leq \|x_t - p_{t-1}\|^2$$

and $\tilde{\alpha}_t \leq 1$. S consists of finite many elements and $\eta := \mathrm{dist}(S, Py) = \min_{x\in S} \|x - Py\| > 0$. Since $x_t \in S$ we know that $\|x_t - Py\| \geq \eta$. Furthermore,

$$\langle Py - p_{t-1}, x_t - Py\rangle = -\langle y - Py, x_t - Py\rangle + \langle y - p_{t-1}, x_t - Py\rangle \geq 0$$

because the first inner product is non-positive and the second term is non-negative (the same argument as in Theorem 2 (ii.e)). Hence,

$$\|x_t - Py\|^2 - \langle p_{t-1} - Py, x_t - Py\rangle \geq \eta^2 > 0.$$

Standard results for the CGM tell us that we have a constant $c > 0$ such that $\|p_t - y\|^2 - \|Py - y\|^2 \leq c/t$ and hence

$$\|p_t - Py\|^2 = \|p_t - y\|^2 + 2\langle p_t - y, y - Py\rangle + \|y - Py\|^2 \leq \|p_t - y\|^2 - \|y - Py\|^2 \leq c/t$$

where we used that

$$\langle p_t - y, y - Py\rangle = \langle p_t - Py, y - Py\rangle - \|y - Py\|^2 \leq -\|y - Py\|^2.$$

Hence, for all $t \in \mathbb{N}$ with $t \geq t_0$, where $t_0 = c\|y - Py\|^2/\eta^4$, we know that $\tilde{\alpha}_t \leq 1$. If $y = Py$ then the argument simplifies to

$$\langle y - p_{t-1}, x_t - p_{t-1}\rangle \leq \|Py - p_{t-1}\|^2 - \langle p_{t-1} - Py, x_t - Py\rangle \leq \|x_t - p_{t-1}\|^2.$$

The remaining case is the case where $Py \in S$. In fact, the only critical case is where $x_t = Py$. We can argue in the following way:

$$\tilde{\alpha}_t = \frac{\langle y - Py, Py - p_{t-1}\rangle + \langle Py - p_{t-1}, Py - p_{t-1}\rangle}{\|Py - p_{t-1}\|^2} \geq 1$$

and $\alpha_t = 1$. Hence, $p_t = Py$ and $x_{t+1} = \arg\max_{x\in S} \langle y - p_t, x\rangle = \arg\max_{x\in S} \langle y - Py, x - Py\rangle$. If $x_{t+1} = Py$ then $p_{t+1} = Py$ and $\tilde{\alpha}_{t+1} = 1$. Otherwise, x_{t+1} is an element of the minimal face. Hence, if $p_t = Py$ and $x_{t+1} \neq Py$ then

$$\tilde{\alpha}_{t+1} = \frac{\langle y - Py, x_{t+1} - p_t\rangle + \langle Py - p_t, x_{t+1} - p_t\rangle}{\|x_{t+1} - p_t\|^2} = 0 = \alpha_{t+1}$$

and $p_{t+1} = Py$. The same argument yields that $p_s = Py$ for all $s \geq t$ and $\tilde{\alpha}_s \in [0, 1]$. ❶

(iii) Consider now the split span $C_c = U_F \oplus U_\delta \oplus U_b$. We will use here the same notation δ_m, δ_F etc. as in Theorem 2.

(a) If Assumption 1 holds then there exists a $s_0 < \infty$ such that for all $t \geq s_0$ the algorithm chooses elements $x_t \in F$. ❶ Assumption 1 provides us with a time s_0 after which only elements in $D(\{x_t\}) \cup F$ are chosen. As in Theorem 2 (iv) we can observe that there exists a constant $c > 0$ such that for all $t \geq s_0$

$$\langle x_t - Py, (P_F + P_\delta)w_t\rangle \geq c\|(P_F + P_\delta)w_t\|.$$

$x_t \in S$ is here the element chosen at step t. Hence, the sequence $\{\|t(P_F + P_\delta)w_t\|\}_{t\geq 0}$ is bounded. Furthermore, because $0 \leq \langle x_t - Py, tw_t\rangle = \langle x_t - Py, t(P_F + P_\delta)w_t\rangle + \langle x_t - Py, tP_b w_t\rangle$ we can infer that

$$-\langle x_t - Py, tP_b w_t\rangle \leq \langle x_t - Py, t(P_F + P_\delta)w_t\rangle \leq \|x_t - Py\| \|t(P_F + P_\delta)w_t\|$$

and the sequence $\{-\langle x_t - Py, tP_b w_t\rangle\}_{t\geq 0}$ is bounded. But this implies that $x = x_t$ is either an element of F or it is selected only finitely often (and, hence, $x \notin D(\{x_t\})$). Therefore, $D(\{x_t\})$ is empty and the result follows. ❶

(b) There exists a constant $d > 0$ and a time u_0 after which for all $t \geq u_0$

$$\|P_F \tilde{w}_t\|^2 \leq \left(1 - \frac{\delta_F^2}{\operatorname{diam}^2(F)}\right) \|P_F \tilde{w}_{t-1}\|^2 + d \, \|(I - P_F)\tilde{w}_{t-1}\|^2 .$$

P Let us consider $t > t_0 \vee s_0$ with s_0 from (a) and t_0 from (ii). We know that only elements in F are chosen at t and hence

$$\begin{aligned}\|x_t - p_{t-1}\|^2 \tilde{\alpha}_t &= \langle w_{t-1}, \tilde{w}_{t-1} + x_t - Py\rangle = \langle \tilde{w}_{t-1}, \tilde{w}_{t-1} + x_t - Py\rangle + \langle y - Py, \tilde{w}_{t-1}\rangle \\ &= \langle P_F \tilde{w}_{t-1}, P_F \tilde{w}_{t-1} + x_t - Py\rangle + \|(I - P_F)\tilde{w}_{t-1}\|^2 + \langle y - Py, \tilde{w}_{t-1}\rangle .\end{aligned}$$

Also, $\tilde{\alpha}_t = \alpha_t$ and

$$\begin{aligned}\|P_F \tilde{w}_t\|^2 &= \|P_F \tilde{w}_{t-1}\|^2 - 2\alpha_t \langle P_F \tilde{w}_{t-1}, P_F \tilde{w}_{t-1} + x_t - Py\rangle + \alpha_t^2 \|x_t - P_F p_{t-1}\|^2 \\ &\leq \left(1 - \frac{\delta_F^2}{\operatorname{diam}^2(F)}\right) \|P_F \tilde{w}_{t-1}\|^2 + \frac{(\|(I - P_F)\tilde{w}_{t-1}\|^2 + \langle y - Py, \tilde{w}_{t-1}\rangle)^2}{\|x_t - p_{t-1}\|^2}.\end{aligned}$$

Now, since p_t converges to Py there exists a time $u_0 > t_0 \vee s_0$ after which $\|x_t - p_{t-1}\| \geq \|x_t - Py\|/2 \geq \min_{x \in S} \|x - Py\|/2$ and $\|(I - P_F)\tilde{w}_{t-1}\|^2 \leq \|(I - P_F)\tilde{w}_{t-1}\| \leq 1$. Hence, for all $t \geq u_0$ we have that

$$\|P_F \tilde{w}_t\|^2 \leq \left(1 - \frac{\delta_F^2}{\operatorname{diam}^2(F)}\right) \|P_F \tilde{w}_{t-1}\|^2 + \frac{4(1 + \|y - Py\|)}{\min_{x \in S} \|x - Py\|^2} \|(I - P_F)\tilde{w}_{t-1}\|^2 .$$

Choosing $d = 4(1 + \|y - Py\|)/(\min_{x \in S} \|x - Py\|^2)$ yields the result. **Q**

(c) If $\min_{x \in S} \|x - Py\| > 0$ and if $\|(I - P_F)\tilde{w}_{t_0 \vee s_0}\| > 0$ with s_0 from (a) and t_0 from (ii) then the sequence $\{\|(I - P_F)\tilde{w}_t\|\}_{t \geq 1}$ converges sub-linearly. **P** Let us consider $t \geq t_0 \vee s_0$. We know that only elements in F are chosen at t and that $\alpha_t = \tilde{\alpha}_t$. Therefore $\|(I - P_F)\tilde{w}_{t+1}\| = (1 - \alpha_{t+1}) \|(I - P_F)\tilde{w}_t\|$ and

$$\begin{aligned}\frac{\|(I - P_F)\tilde{w}_{t+1}\|}{\|(I - P_F)\tilde{w}_t\|} = 1 - \alpha_{t+1} &= \frac{\|x_{t+1} - Py + \tilde{w}_t\|^2 - \langle w_t, \tilde{w}_t + x_{t+1} - Py\rangle}{\|x_{t+1} - p_t\|^2} \\ &= \frac{\|x_{t+1} - Py\|^2 + \langle P_F \tilde{w}_t, x_{t+1} - Py\rangle - \langle y - Py, P_b \tilde{w}_t\rangle}{\|x_{t+1} - Py\|^2 + 2\langle x_{t+1} - Py, \tilde{w}_t\rangle + \|\tilde{w}_t\|^2}\end{aligned}$$

which converges to 1 since $\lim_{t \to \infty} \|\tilde{w}_t\| = 0$. This means that the sequence converges sub-linearly. **Q** ■

6.3 Corollaries

Corollary 4 *Given a compact convex set $C \subset \mathcal{H}$ and a finite subset S of C with $\operatorname{ex} C \subseteq S$. If for $y \in \mathcal{H}$ there exists a decomposition $U_F \oplus U_\delta \oplus U_b$ of $\operatorname{span} C_c$ such that U_b is one dimensional then Assumption 1 is fulfilled. In particular, in this case there exists a constant $b > 0$ such that Algorithm 1 has a worst-case approximation error of b/t for all $t \geq 1$.*

Proof Consider elements $x, x' \in S \backslash F$. Since $\|P_b(x - Py)\|, \|P_b(x' - Py)\| > 0$ we know that $\Delta = \|P_b(x - Py)\| / \|P_b(x' - Py)\| < \infty$ and since U_b is one-dimensional we have for any t

$$\langle -P_b w_t, x - Py\rangle = \|P_b w_t\| \, \|P_b(x - Py)\| \leq \Delta \, \|P_b w_t\| \, \|P_b(x' - Py)\| = \Delta \langle -P_b w_t, x' - Py\rangle$$

and Assumption 1 is fulfilled. ■

Let for the following corollary $A = \{z \in \mathcal{H} : z = \alpha P_C(y - Py) + P_C Py, \alpha \in [0, \infty)\}$.

Corollary 5 *Given a compact convex set $C \subset \mathcal{H}$ and a finite subset S of C with $\operatorname{ex} C \subseteq S$. Assumption 1 is fulfilled if $P_C y \in \mathcal{H} \backslash C$ and whenever $Pz = Py$ for some $z \in \mathcal{H}$ then $P_C z \in A$ holds. In particular, in this case there exists a constant $b > 0$ such that Algorithm 1 has a worst-case approximation error of b/t for all $t \geq 1$.*

Proof The assumption guarantees us that U_b is one-dimensional. Assume otherwise. By assumption $\|P_C(y - Py)\| > 0$ and we can chose E_b such that there exist two elements e_1, e_2 in E_b with $e_1 = P_C(y - Py)/\|P_C(y - Py)\|$ and $\langle e_1, e_2\rangle = 0, \|e_1\| = \|e_2\| = 1$. We claim that $z_1 = Py + e_1$ and $z_2 = Py + e_2$ are both projected onto Py, i.e. $Pz_1 = Py = Pz_2$: We know that $P_{e_1}[C_c] \subset (-\infty, 0]$ and $\langle e_1, x - Py\rangle \leq 0$ for all $x \in C$. Hence, for all $x \in C$ we know that $\langle z_1 - Py, x - Py\rangle = \langle e_1, x - Py\rangle \leq 0$. But, this means that $Pz_1 = Py$. The same argument applies to z_2 and Py is the projection of two elements for which $\langle z_1 - Py, z_2 - Py\rangle = 0$. Furthermore, $P_C z_1 = P_C Py + e_1$ and $P_C z_2 = P_C Py + e_2$. If $P_C z_2 \in A$ then there exist $\alpha, \tilde{\alpha} > 0$,

$$e_2 + P_C Py = P_C z_2 = \alpha P_C(y - Py) + P_C Py = \tilde{\alpha} e_1 + P_C Py$$

and $e_2 = \tilde{\alpha} e_1$ with a contradiction to orthogonality. ■

For the next corollaries let $d = \dim U_b$, let $e_1, \ldots, e_d$ be any basis of U_b and let $r = \sup_{x \in C} \|x\|$. Also introduce $\alpha = \min_{i \leq d} \min\{|\langle e_i, x - Py\rangle| : x \in S \backslash F, \langle e_i, x - Py\rangle \neq 0\} > 0$.

Corollary 6 *Let C be a compact convex set in some Hilbert space $\mathcal{H}$, S a finite set with $\operatorname{cch} S = C$, and $y \in \mathcal{H}$ such that there exists a split into $U_F \oplus U_\delta \oplus U_b$ of $\operatorname{span} C_c$ with $U_\delta = \{0\}$. Assumption 1 is fulfilled and there exists a constant $b > 0$ such that Algorithm 1 has a worst-case approximation error of b/t for all $t \geq 1$. The constant b can be chosen as $\sqrt{d}\, 4r^3/(\alpha\delta_F) + 6r^2(1/\delta_F + 1/\alpha) + 5r$.*

Proof If $F_c = \{0\}$ then each element $x \in S \backslash F$ can only be chosen once since $0 > \langle w_t, x - Py\rangle = \langle P_b w_t, x - Py\rangle$ if x has been chosen at least once. This implies that $D(\{x_t\})$ is, in fact, the empty set and Assumption 1 is fulfilled. It is also easy to see that $\langle w_t, e_i\rangle \leq \|x - Py\| \leq 2r$ for all $i \leq d$. This implies that the constant b can in this case be chosen as $2r\sqrt{d}$. Also, since δ_F and α are upper bounded by r this implies that $b \leq 2r^3\sqrt{d}/(\alpha\delta_F)$.

Assume now that $F_c \neq \{0\}$ and that there exists a $x \in D(\{x_t\})$. By definition $x \in S \backslash F$ and x is chosen infinitely often. Hence, $\{\langle P_b w_t, x - Py\rangle\}_{t \geq 1}$ is a non-increasing sequence that diverges to $-\infty$. There exists a $\delta_F > 0$ such that for any w_t, $\max_{x' \in S \cap F} \langle w_t, x' - Py\rangle \geq \delta_F \|P_F w_t\|$. Also, for any $x' \in S$, we have that $\langle w_t, x' - Py\rangle = \langle P_F w_t, x' - Py\rangle + \langle P_b w_t, x' - Py\rangle$ and, since the term $\langle P_b w_t, x' - Py\rangle$ is always non-positive, we know that if $P_F w_t \neq 0$ elements will be chosen such that $\langle P_F w_t, x' - Py\rangle \geq \delta_F \|P_F w_t\|$. Hence, whenever $\|P_F w_t\| \geq 2r^2/\delta_F$ for some t then $\|P_F w_{t+1}\| \leq \|P_F w_t\|$. This implies that $\|P_F w_t\| \leq 2r^2/\delta_F + 3r$ for all $t \geq 1$. Hence, for x to be chosen infinitely often we need for all time steps t where x is chosen that

$$0 \leq \langle w_t, x - Py\rangle \leq \langle P_F w_t, x - Py\rangle + \langle P_b w_t, x - Py\rangle \leq 2r(2r^2/\delta_F + 3r) + \langle P_b w_t, x - Py\rangle$$

since $\|x - Py\| \leq 2r$. Therefore, $-\langle P_b w_t, x - Py\rangle \leq r(4r^2/\delta_F + 6r)$ and $\{\langle P_b w_t, x - Py\rangle\}_{t\geq 1}$ cannot diverge to $-\infty$. In other words, x cannot be chosen infinitely often with a contradiction to the initial assumption. We can also control the constant in this case: $\|P_b w_t\|$ can only grow if either $P_F w_t = 0$ or there is an $x' \in S\backslash F$ such that $-\langle P_b w_t, x' - Py\rangle \leq r(4r^2/\delta_F + 6r)$. Then for w_t if for any $i \leq d$

$$\alpha \langle w_t, e_i\rangle > r(4r^2/\delta_F + 6r)$$

then no $x \in S\backslash F$ with $\langle e_i, x - Py\rangle \neq 0$ can be played since for such a x

$$-\langle P_b w_t, x - Py\rangle \geq \langle e_i, x - Py\rangle \langle e_i, w_t\rangle > r(4r^2/\delta_F + 6r).$$

Hence, $\langle e_i, w_t\rangle \leq r(4r^2/\delta_F + 6r)/\alpha + 2r$ since the increment of w_t in one step is bounded by $2r$. Since this holds for each $i \leq d$ we gain the bound $\|P_b w_t\| \leq \sqrt{d}4r^3/(\delta_F\alpha) + 6r^2/\alpha + 2r$ and

$$\|w_t\| \leq \|P_F w_t\| + \|P_b w_t\| \leq \sqrt{d}4r^3/(\delta_F\alpha) + 6r^2(1/\delta_F + 1/\alpha) + 5r.$$

■

Corollary 7 *Let $y \in \mathbb{R}^d$, Q any orthogonal matrix, $c > 0$ any scaling, $z \in \mathbb{R}^d$, $S = \{0,1\}^d$ and $C = [0,1]^d$, $d \geq 1$. Assumption 1 is fulfilled for the set $\tilde{S} = cQ[S + z]$ and $\tilde{C} = cQ[C + z]$ (independently of the dimensionality of the face Py lies in). In particular, there exists a constant $b > 0$ such that Algorithm 1 has a worst-case approximation error of b/t for all $t \geq 1$. The constant b can be chosen as $\sqrt{d}4r^3/(c\delta_F) + 6r^2(1/\delta_F + 1/c) + 5r$.*

Proof Because the algorithm does not change by translating and rotating S and C we can consider a hypercube anchored at the origin and aligned with the standard basis of $\mathbb{R}^d$. We therefore assume without loss of generality that we have to deal with a scaled standard hypercube in $\mathbb{R}^d$. Let us assume in the following that $\dim U_F = k \leq d$.

Observe that either $U_F = \{0\}$ or we can write $U_F = \text{span}\{e_{i_1}, \ldots, e_{i_k}\}$, where $e_1, \ldots, e_d$ is the standard basis in $\mathbb{R}^d$, $i_j \leq d$ for all $j \leq k$, and $k = \dim U_F$. This holds because a k-dimensional face of the hypercube is a translated k-dimensional hypercube in a subspace spanned by some basis vectors $e_{i_1}, \ldots, e_{i_k}$. Define the following index sets $\mathcal{I} = \{i_1, \ldots, i_k\}$, $\mathcal{I} = \emptyset$ if $U_F = \{0\}$, and $\mathcal{J} = \{1, \ldots, d\}\backslash\mathcal{I}$.

There are signs $s_j \in \{-1, 1\}$, $j \in \mathcal{J}$, such that $E_b = \{s_j e_j : j \in \mathcal{J}\}$ is a basis for U_b, and the split into $U_F \oplus U_\delta \oplus U_b$ is satisfying the conditions of Assumption 1. In particular, $U_\delta = \{0\}$, $P_C(y - Py) \in U_b$ and for all $e \in E_b$ we have $\{\langle x, e\rangle : x \in C_c\} \subset (-\infty, 0]$. ▶ $U_\delta = \{0\}$ because the minimal faces which are orthogonally projected onto a face that contains a ball around Py (relative to the projection) are translated versions of the minimal face and they induce the same subspace as the minimal face. That also implies directly that E_b can be chosen to fulfill Assumption 1. $P_C(y - Py)$ stands orthogonal on U_F and, hence, lies fully in U_b. Therefore, Corollary 6 applies. Observe that α can here be chosen as the scaling factor c and the corollary provides the stated result since $\dim U_b \leq d$. ■

Corollary 8 *Let $y \in \mathbb{R}^d$, $S = \{e_i : i \leq d\}$ and $C = \Delta^{d-1} = \text{cch}\, S$, $d \geq 1$. Assumption 1 is fulfilled and there exists a constant $b > 0$ such that Algorithm 1 has a worst-case approximation error of b/t for all $t \geq 1$. The constant b can be chosen as $4dr^3/\delta_F + 6r^2(1/\delta_F + \sqrt{d}) + 5r$.*

Proof Due to the rotation invariance it suffices to consider the case where $F = \text{cch}\{e_1, \ldots, e_k\}$ for some $k \leq d$. Let us consider first the case where $F = \{e_1\}$ and, hence, $F_c = \{0\}$. In this case, $C_c = \text{cch}\{0, e_2 - e_1, \ldots, e_d - e_1\}$. Observe that $\langle x + e_1, e_j\rangle \geq 0$ for any $x \in C_c$ and all $j \leq d$ since $x + e_1 \in C$ and, hence, $x + e_1 = \sum_{i\leq d} \alpha_i e_i$ for some non-negative α_i. In particular, for $j \geq 2$, $\langle x, e_j\rangle \geq 0$ for all $x \in C_c$. Furthermore, $\langle x, e_1\rangle \leq 0$ for all $x \in C_c$ because $x + e_1 = \sum_{i\leq d} \alpha_i e_i$ with non-negative scalars that fulfill $\sum_{i\leq d} \alpha_i = 1$. I.e. $\langle x, e_1\rangle = \langle x + e_1, e_1\rangle - 1 \leq \|e_1\|^2 - 1 = 0$. Therefore, we have an orthonormal basis $\{-e_1, e_2, \ldots, e_d\}$, of $\mathbb{R}^d$ such that $\langle x, e\rangle \geq 0$ for all basis elements e and all $x \in C_c$. This implies that there exists no basis element $\tilde{e} \in \mathbb{R}^d$ such that there exists elements $x, x' \in C_c$ with $\langle x, \tilde{e}\rangle > 0 > \langle x', \tilde{e}\rangle$ and, hence, $\{-e_1, e_2, \ldots, e_d\}$ spans the d-dimensional space U_b and $U_\delta = \{0\}$. The rate of convergence follows now from Corollary 6.

Let us consider the case $F = \text{cch}\{e_1, \ldots, e_k\}$ and $k \geq 2$. For any e_j, $j > k$, and any $x \in C_c$, $\langle x + Py, e_j\rangle \geq 0$ since $x + Py = \sum_{i\leq d} \alpha_i e_i$ for some non-negative α_i. But this implies $\langle x, e_j\rangle = \langle x + Py, e_j\rangle \geq 0$ because Py is a convex combination of $e_1, \ldots, e_k$. Also $e_{k+1}, \ldots, e_d \in F_c^\perp$ because span $F_c = \text{span}\{e_2 - e_1, \ldots, e_k - e_1\}$ and $\langle e_i - e_1, e_j\rangle = 0$ for any $i \leq k, j > k$. Finally, consider $e = \sum_{j=1}^k e_j/\sqrt{k}$, $\|e\| = 1$, and observe that $\langle e, e_i\rangle = 0$ for all $i > k$. Also, $\langle e, e_i - e_1\rangle = 0$ for $i \leq k$. This implies that e stands orthogonal on F_c because for $v \in \text{span}\, F_c$, $v = \sum_{i=2}^k \alpha_i(e_i - e_1), \alpha_i \in \mathbb{R}$, we have that $\langle e, v\rangle = \sum_{i=2}^k \alpha_i\langle e_i - e_1, e_i + e_1\rangle = 0$. Furthermore, for any $x \in C_c$, $x = \sum_{i=1}^d \alpha_i e_i - Py$, $\sum_{i=1}^d \alpha_i = 1$,

$$\langle e, x\rangle = \sum_{i=1}^d \alpha_i \langle e, e_i - Py\rangle = \sum_{i=1}^d \alpha_i \langle e, e_i - e_1\rangle - \langle e, Py - e_1\rangle = 0.$$

Hence, we have $d + 1 - k$ orthonormal vectors $e_{k+1}, \ldots, e_d, e$ that lie in the orthogonal complement of F_c and for any $x \in C_c$, $\langle x, e_i\rangle \geq 0$, $\langle x, e\rangle \geq 0$, for all $k + 1 \leq i \leq d$. Hence, $U_\delta = \{0\}$ and the result on the rate of convergence follows from Corollary 6.

The constant also follows directly from the corollary: in case that $F_c = \{0\}$ we have that $S\backslash F = \{e_2, \ldots, e_d\}$, the basis of U_b is $\{-e_1, e_2, \ldots, e_d\}$ and $Py = e_1$. α can be lower bounded by $1 : \langle -e_1, x - Py\rangle = 1$ for any $x \in S\backslash F$; $\langle e_i, e_j - Py\rangle$ attains value 1 if $i = j \geq 2$ and value 0 if $i \neq j, i, j \geq 2$.

Otherwise, $S\backslash F = \{e_{k+1}, \ldots, e_d\}$, the basis of U_b is $\{e_{k+1}, \ldots, e_d, \sum_{i=1}^k e_i/\sqrt{k}\}$ and $Py = \sum_{i=1}^k \alpha_i e_i$ for some $\alpha_i \geq 0, \sum_{i=1}^k \alpha_i = 1$. Hence, $\langle e_i, e_j - Py\rangle = 0$ if $i \neq j, i, j > k$; $\langle e_i, e_i - Py\rangle = 1$ if $i > k$; $\langle \sum_{i=1}^k e_i/\sqrt{k}, e_j - Py\rangle = 1/\sqrt{k}$, where $j \geq k$. Hence, $\alpha \geq 1/\sqrt{d}$ and the result follows. ■

References

N. Aronszajn. Theory of reproducing kernels. *Trans. of the American Mathematical Society*, 1950.

F. Bach, S. Lacoste-Julien, and G. Obozinski. On the equivalence between herding and conditional gradient algorithms. In *International Conference on Machine Learning*, 2012.

A. Beck and M. Teboulle. A conditional gradient method with linear rate of convergence for solving convex linear systems. *Math. Meth. Op. Res.*, 2004.

M.D. Cannon and C.D. Cullum. A tight upper bound on the rate of convergence of the frank-wolfe algorithm. *SIAM J. Control Optim.*, 1968.

Y. Chen, M. Welling, and A. Smola. Supersamples from kernel-herding. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2010.

K. L. Clarkson. Coresets, sparse greedy approximation,and the frank-wolfe algorithm. *ACM Transactions on Algorithms*, 2010.

J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *International Conference on Machine Learning*, 2008.

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Res. Logist. Quart.*, 1956.

D. Garber and E. Hazan. A polynomial time conditional gradient algorithm with applications to online and stochastic optimization. *CoRR*, abs/1301.4666, 2013.

D. Garber and E. Hazan. Faster rates for the frank-wolfe method over strongly-convex sets. In *International Conference on Machine Learning*, 2015.

J. Guélat and P. Marcotte. Some comments on wolfe's 'away step'. *Mathematical Programming*, 1986.

M. Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *International Conference on Machine Learning*, 2013.

S. Lacoste-Julien and M. Jaggi. On the global linear convergence of frank-wolfe optimization variants. In *Advances in Neural Information Processing*, 2015.

R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B.*, 1996.

K.C. Toh, M.J. Todd, and R.H. Tutuncu. Sdpt3 — a matlab software package for semidefinite programming. *Optimization Methods and Software*, 1999.

I. Tsang, J. Kwok, and P.-M. Cheung. Core vector machines: Fast svm training on very large data sets. *The Journal of Machine Learning Research*, 2005a.

I. Tsang, J. Kwok, and K. Lai. Core vector regression for very large regression problems. In *International Conference on Machine Learning*, 2005b.

R.H Tutuncu, K.C. Toh, and M.J. Todd. Solving semidefinite-quadratic-linear programs using sdpt3. *Mathematical Programming Ser. B*, 2003.

M. Welling. Herding dynamical weights to learn. In *International Conference on Machine Learning*, 2009.

P. Wolfe. *Convergence theory in nonlinear programming*, chapter 1. North-Holland Publishing Company, 1970.

P. Wolfe. Finding the nearest point in a polytope. *Mathematical Programming*, 1976.

Y. Zhang, J. Duchi, and M. Wainwright. Divide and conquer kernel ridge regression. In *Conference on Learning Theory*, 2013.

Journal of Machine Learning Research 18 (2018) 1-62 Submitted 6/16 Revised 4/18; Published 5/18

Complete Graphical Characterization and Construction of Adjustment Sets in Markov Equivalence Classes of Ancestral Graphs

Emilija Perković PERKOVIC@STAT.MATH.ETHZ.CH
Seminar for Statistics, ETH Zurich, Switzerland

Johannes Textor JOHANNES.TEXTOR@RADBOUDUMC.NL
Institute for Computing and Information Sciences and Department of Tumor Immunology, Radboud University Medical Center, Nijmegen, The Netherlands

Markus Kalisch KALISCH@STAT.MATH.ETHZ.CH
Marloes H. Maathuis MAATHUIS@STAT.MATH.ETHZ.CH
Seminar for Statistics, ETH Zurich, Switzerland

Editor: Christopher Meek

Abstract

We present a graphical criterion for covariate adjustment that is sound and complete for four different classes of causal graphical models: directed acyclic graphs (DAGs), maximal ancestral graphs (MAGs), completed partially directed acyclic graphs (CPDAGs), and partial ancestral graphs (PAGs). Our criterion unifies covariate adjustment for a large set of graph classes. Moreover, we define an explicit set that satisfies our criterion, if there is any set that satisfies our criterion. We also give efficient algorithms for constructing all sets that fulfill our criterion, implemented in the `R` package `dagitty`. Finally, we discuss the relationship between our criterion and other criteria for adjustment, and we provide new soundness and completeness proofs for the adjustment criterion for DAGs.

Keywords: causal effects, graphical models, covariate adjustment, latent variables, confounding

1. Introduction

Covariate adjustment is a well-known method to estimate causal effects from observational data. There are, however, still common misconceptions about what variables one should or should not adjust for. For example, it is sometimes thought that adjusting for more variables will lead to a more precise estimate, as long as the added variables are not affected by the exposure variable. While this is true for a randomized exposure, in observational data even adjustment for pre-exposure variables may lead to so-called collider bias as described in the "M-bias graph" (Shrier, 2008; Rubin, 2008). Another example is the "Table 2 fallacy" (Westreich and Greenland, 2013). In observational research papers, Table 1 often describes the data, and Table 2 shows a multiple regression analysis. By presenting all estimated coefficients in one table, it is implicitly suggested that all estimates can be interpreted similarly. This is usually not the case: some coefficients may be interpreted as a total causal effect, some may be interpreted as a direct causal effect, and some do not have any causal interpretation at all.

	DAG	MAG	CPDAG	PAG
Back-door criterion Pearl (1993)	$\Rightarrow$			
Adjustment criterion Shpitser et al. (2010), Shpitser (2012)	$\Leftrightarrow$			
Adjustment criterion van der Zander et al. (2014)	$\Leftrightarrow$	$\Leftrightarrow$		
Generalized back-door criterion Maathuis and Colombo (2015)	$\Rightarrow$	$\Rightarrow$	$\Rightarrow$	$\Rightarrow$
Generalized adjustment criterion Perković et al. (2015)	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$

Table 1: Graphical criteria for covariate adjustment: $\Rightarrow$ - sound, $\Leftrightarrow$ - sound and complete.

The practical importance of covariate adjustment has inspired a growing body of theoretical work on graphical criteria for adjustment. Pearl's back-door criterion (Pearl, 1993) is probably the most well-known, and is sound but not complete for adjustment in DAGs. Shpitser et al. (2010) and Shpitser (2012) refined the back-door criterion to a sound and complete graphical criterion for adjustment in DAGs. Others considered more general graph classes, which can represent structural uncertainty. Van der Zander et al. (2014, 2018) gave sound and complete graphical criteria for MAGs that allow for unobserved variables (latent confounding). Maathuis and Colombo (2015) presented a generalized back-door criterion for DAGs, CPDAGs, MAGs and PAGs, where CPDAGs and PAGs represent Markov equivalence classes of DAGs or MAGs, respectively, and can be inferred directly from data (see, for example, Spirtes et al., 2000; Chickering, 2002; Colombo et al., 2012; Claassen et al., 2013; Colombo and Maathuis, 2014; Nandy et al., 2018; Frot et al., 2018; Heinze-Deml et al., 2017). The generalized back-door criterion is sound but not complete for adjustment. Another line of work explores data driven covariate adjustment that does not require knowing the graph (VanderWeele and Shpitser, 2011; De Luna et al., 2011; Entner et al., 2013). Some of these data driven results are sound and complete for adjustment, but they all rely on some additional assumptions. We will not explore this direction in our paper.

In Perković et al. (2015), the preliminary conference version of the present paper, we extended the results of Shpitser et al. (2010); Shpitser (2012), van der Zander et al. (2014) and Maathuis and Colombo (2015) to derive a single sound and complete adjustment criterion for DAGs, CPDAGs, MAGs and PAGs. The different adjustment criteria are summarized in Table 1. Additionally, we note that van der Zander and Liśkiewicz (2016) showed that the generalized adjustment criterion can also be applied to the more general class of restricted chain graphs (representing a subset of a Markov equivalence class of DAGs). Furthermore, in Perković et al. (2017), we extend the generalized adjustment criterion to maximally oriented partially directed acyclic graphs (PDAGs), which represent CPDAGs with added background knowledge.

To illustrate the use of our generalized adjustment criterion, suppose we are given the CPDAG in Figure 1a and we want to estimate the total causal effect of X on Y. Our

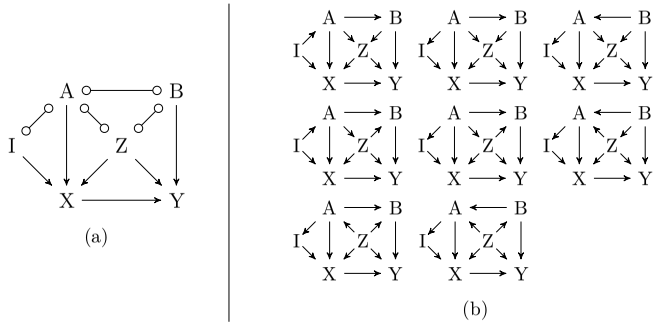


Figure 1: (a) A CPDAG in which, according to our criterion, $\{A, Z\}$ is an adjustment set for the total causal effect of X on Y. (b) The Markov equivalence class of DAGs represented by the CPDAG. An adjustment set for a CPDAG (PAG) is one that is valid for all DAGs (MAGs) in the Markov equivalence class.

criterion will inform us that the set $\{A, Z\}$ is an adjustment set for this CPDAG, meaning that it is an adjustment set in every DAG that the CPDAG represents (Figure 1b). Hence, we can estimate the causal effect without knowledge of the full causal structure. In a similar manner, by applying our criterion to a MAG or a PAG, we find adjustment sets that are valid for all DAGs represented by this MAG or PAG. Our criterion finds such adjustment sets whenever they exist; else, the causal effect is not identifiable by covariate adjustment. We hope that this ability to allow for incomplete structural knowledge, latent confounding, or both will help address concerns that graphical causal modelling "assumes that all [...] DAGs have been properly specified" (West and Koch, 2014). Moreover, our criterion for CPDAGs and PAGs can be combined with causal structure learning algorithms.

In the current paper, we give full proofs of the results in Perković et al. (2015). In addition, we provide several new results that allow us to construct sets $\mathbf{Z}$ that fulfill the generalized adjustment criterion for given sets $\mathbf{X}$ of exposures and $\mathbf{Y}$ of response variables in a DAG, CPDAG, MAG or PAG $\mathcal{G}$. In Corollary 15 we define a specific set that satisfies our criterion, if any set does. We refer to this set as a "constructive set". In Theorem 7, we show how one can express adjustment sets in terms of m-separating sets in a certain subgraph of $\mathcal{G}$. This theorem reduces the problem of finding adjustment sets to the problem of finding m-separating sets, which has been studied in detail by van der Zander et al. (2014). In Lemma 10, we prove that all adjustment sets for a CPDAG (PAG) $\mathcal{G}$ can be found in an arbitrary orientation of $\mathcal{G}$ into a valid DAG (MAG). This allows us to leverage existing implementations (van der Zander et al., 2014). We implemented the criterion itself and the construction of all adjustment sets in the software `dagitty` (Textor et al., 2016), available as a web-based GUI and an R package, and in the R package `pcalg` (Kalisch et al., 2012).

Furthermore, we explore the relationships between our generalized adjustment criterion and the previously suggested generalized back-door criterion and Pearl's back-door criterion. For both Pearl's back-door criterion and the generalized back-door criterion, a constructive

set was given only in the case when the number of exposures is limited to one ($|\mathbf{X}| = 1$). We give constructive sets for each of these criteria for general $\mathbf{X}$ in Corollary 22 and Corollary 24. Moreover, in Theorem 26 we identify cases in which there exist sets satisfying two, or all three of these criteria, as well as cases in which there are only sets satisfying the generalized adjustment criterion.

Another important contribution, included in the appendix, are new soundness and completeness proofs of the adjustment criterion for DAGs as defined in Shpitser et al. (2010) and in the unpublished addendum Shpitser (2012), where the adjustment criterion in Shpitser (2012) is a revised version of the criterion in Shpitser et al. (2010) (see Definition 55 in Appendix E). Since there are no published soundness and completeness proofs for the revised criterion and since we build on this work, we felt it was important to provide these proofs. The proofs are non-trivial, but rely only on elementary concepts.

We note that, although we can find all causal effects that are identifiable by covariate adjustment, we generally do not find all identifiable causal effects, since some effects may be identifiable only by other means, using for example IDA approaches (Maathuis et al., 2009, 2010; Nandy et al., 2017; Malinsky and Spirtes, 2017), Pearl's front-door criterion (Pearl, 2009, Section 3.3.2) or the ID algorithm (Tian and Pearl, 2002; Shpitser and Pearl, 2006).

We also point out that MAGs and PAGs are in principle not only able to represent unobserved confounding, but can also account for unobserved selection variables. In this paper, however, we assume that there are no unobserved selection variables, since selection bias often rules out causal effect identification using just covariate adjustment. Bareinboim et al. (2014) discuss these problems and present creative approaches to work around them, for example by combining data from different sources. The question whether our adjustment criterion could be combined with such auxiliary methods is left for future research.

2. Preliminaries

Throughout the paper we denote sets in bold (for example $\mathbf{X}$), graphs in calligraphic font (for example $\mathcal{G}$) and nodes in a graph in uppercase letters (for example X). All omitted proofs are given in the appendix.

Nodes and edges. A graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ consists of a set of nodes (variables) $\mathbf{V} = \{X_1, \ldots, X_p\}$ and a set of edges $\mathbf{E}$. We consider simple graphs, meaning that there is at most one edge between any pair of nodes. Two nodes are called *adjacent* if they are connected by an edge. Every edge has two edge marks that can be arrowheads, tails or circles. Edges can be *directed* $\rightarrow$, *bi-directed* $\leftrightarrow$, *non-directed* $\circ\!\!-\!\!\circ$, or *partially directed* $\circ\!\!\rightarrow$. We use $\bullet$ as a stand in for any of the allowed edge marks. An edge is *into* (*out of*) a node X if the edge has an arrowhead (tail) at X. A *directed graph* contains only directed edges. A *mixed graph* may contain directed and bi-directed edges. A *partial mixed graph* may contain any of the described edges. Unless stated otherwise, definitions apply to partial mixed graphs.

Paths. A *path* p from X to Y in $\mathcal{G}$ is a sequence of distinct nodes $\langle X, \ldots, Y \rangle$ in which every pair of successive nodes is adjacent in $\mathcal{G}$. If $p = \langle X_1, X_2, \ldots, X_k, \rangle, k \geq 2$, then with $-p$ we denote the path $\langle X_k, \ldots, X_2, X_1 \rangle$. A node V *lies on a path* p if V occurs in the sequence of nodes. If $p = \langle X_1, X_2, \ldots, X_k, \rangle, k \geq 2$, then X_1 and X_k are *endpoints* of p, and any other node $X_i, 1 < i < k$, is a *non-endpoint* node on p. The *length* of a path equals the number of edges on the path. A *directed path* from X to Y is a path from X to Y in which

all edges are directed towards Y, that is, $X \to \cdots \to Y$. We also refer to this as a *causal path*. A *possibly directed path* or *possibly causal path* from X to Y is a path from X to Y that does not contain an arrowhead pointing in the direction of X. A path from X to Y that is not possibly causal is called a *non-causal path* from X to Y. A directed path from X to Y together with $Y \to X$ forms a *directed cycle*. A directed path from X to Y together with $Y \leftrightarrow X$ forms an *almost directed cycle*. For two disjoint subsets $\mathbf{X}$ and $\mathbf{Y}$ of $\mathbf{V}$, a path from $\mathbf{X}$ to $\mathbf{Y}$ is a path from some $X \in \mathbf{X}$ to some $Y \in \mathbf{Y}$. A path from $\mathbf{X}$ to $\mathbf{Y}$ is *proper* (wrt $\mathbf{X}$) if only its first node is in $\mathbf{X}$. If $\mathcal{G}$ and $\mathcal{G}^*$ are two graphs with identical adjacencies and p is a path in $\mathcal{G}$, then the *corresponding path* p^* is the path in $\mathcal{G}^*$ constituted by the same sequence of nodes as p.

Subsequences, subpaths and concatenation. A *subsequence* of a path p is a sequence of nodes obtained by deleting some nodes from p without changing the order of the remaining nodes. A subsequence of a path is not necessarily a path. For a path $p = \langle X_1, X_2, \ldots, X_m \rangle$, the *subpath* from X_i to X_k $(1 \leq i \leq k \leq m)$ is the path $p(X_i, X_k) = \langle X_i, X_{i+1}, \ldots, X_k \rangle$. We denote the concatenation of paths by $\oplus$, so that for example $p = p(X_1, X_k) \oplus p(X_k, X_m)$. In this paper, we only concatenate paths if the result of the concatenation is again a path.

Ancestral relationships. If $X \to Y$, then X is a *parent* of Y. If there is a directed (possibly directed) path from X to Y, then X is a *ancestor* (*possible ancestor*) of Y, and Y is a *descendant* (*possible descendant*) of X. We also use the convention that every node is a descendant, possible descendant, ancestor and possible ancestor of itself. The sets of parents, descendants and ancestors of X in $\mathcal{G}$ are denoted by $\mathrm{Pa}(X, \mathcal{G})$, $\mathrm{De}(X, \mathcal{G})$ and $\mathrm{An}(X, \mathcal{G})$ respectively. The sets of possible descendants and possible ancestors of X in $\mathcal{G}$ are denoted by $\mathrm{PossDe}(X, \mathcal{G})$ and $\mathrm{PossAn}(X, \mathcal{G})$ respectively. For a set of nodes $\mathbf{X} \subseteq \mathbf{V}$, we let $\mathrm{Pa}(\mathbf{X}, \mathcal{G}) = \cup_{X \in \mathbf{X}} \mathrm{Pa}(X, \mathcal{G})$, with analogous definitions for $\mathrm{De}(\mathbf{X}, \mathcal{G})$, $\mathrm{An}(\mathbf{X}, \mathcal{G})$, $\mathrm{PossDe}(\mathbf{X}, \mathcal{G})$ and $\mathrm{PossAn}(\mathbf{X}, \mathcal{G})$.

Colliders, shields and definite status paths. If a path p contains $X_i \bullet\!\!\to X_j \leftarrow\!\!\bullet X_k$ as a subpath, then X_j is a *collider* on p. A *collider path* is a path on which every non-endpoint node is a collider. A path of length one is a trivial collider path. A path $\langle X_i, X_j, X_k \rangle$ is an *(un)shielded triple* if X_i and X_k are (not) adjacent. A path is *unshielded* if all successive triples on the path are unshielded. A node X_j is a *definite non-collider* (Zhang, 2008a) on a path p if there is at least one edge out of X_j on p, or if $X_i \bullet\!\!-\!\!\circ X_j \circ\!\!-\!\!\bullet X_k$ is a subpath of p and $\langle X_i, X_j, X_k \rangle$ is an unshielded triple. Any collider on a path is always of definite status and hence, a *definite collider*. In a DAG (MAG) we refer to definite non-colliders as *non-colliders*. A node is of *definite status* on a path if it is a collider or a definite non-collider on the path. A path p is of definite status if every non-endpoint node on p is of definite status.

m-separation and m-connection. A definite status path p between nodes X and Y is *m-connecting* given a set of nodes $\mathbf{Z}$ $(X, Y \notin \mathbf{Z})$ if every definite non-collider on p is not in $\mathbf{Z}$, and every collider on p has a descendant in $\mathbf{Z}$ (Richardson, 2003). Otherwise $\mathbf{Z}$ blocks p. If $\mathcal{G}$ is a DAG or MAG (defined later) and if $\mathbf{Z}$ blocks all paths between X and Y, we say that X and Y are m-separated given $\mathbf{Z}$ in $\mathcal{G}$. Otherwise, X and Y are m-connected given $\mathbf{Z}$ in $\mathcal{G}$. For pairwise disjoint subsets $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ of $\mathbf{V}$ in $\mathcal{G}$, we say that $\mathbf{X}$ and $\mathbf{Y}$ are m-separated given $\mathbf{Z}$ in $\mathcal{G}$ if X and Y are m-separated given $\mathbf{Z}$ in $\mathcal{G}$ for any $X \in \mathbf{X}$ and

$Y \in \mathbf{Y}$. Otherwise, $\mathbf{X}$ and $\mathbf{Y}$ are m-connected given $\mathbf{Z}$ in $\mathcal{G}$. In a DAG, m-separation and m-connection simplify to d-separation and d-connection (Pearl, 2009).

Causal Bayesian networks. A directed graph without directed cycles is a *directed acyclic graph* (DAG). A Bayesian network for a set of variables $\mathbf{V} = \{X_1, \ldots, X_p\}$ is a pair $(\mathcal{G}, f)$, where $\mathcal{G}$ is a DAG, and f is a joint density for $\mathbf{V}$ that factorizes as $f(\mathbf{V}) = \prod_{i=1}^{p} f(X_i | \mathrm{Pa}(X_i, \mathcal{G}))$ (Pearl, 2009). We call a DAG *causal* if every edge $X_i \to X_j$ in $\mathcal{G}$ represents a direct causal effect of X_i on X_j. A Bayesian network $(\mathcal{G}, f)$ is a *causal Bayesian network* if $\mathcal{G}$ is a causal DAG. If a causal Bayesian network is given and all variables are observed, one can easily derive post-intervention densities. In particular, we consider interventions $do(\mathbf{X} = \mathbf{x})$, or shorthand $do(\mathbf{x})$, $(\mathbf{X} \subseteq \mathbf{V})$, which represent outside interventions that set $\mathbf{X}$ to $\mathbf{x}$, uniformly in the population (see Pearl, 2009):

$$f(\mathbf{v} \mid do(\mathbf{x})) = \begin{cases} \prod_{\{i | X_i \in \mathbf{V} \setminus \mathbf{X}\}} f(x_i \mid \mathrm{Pa}(x_i, \mathcal{G})), & \text{if } \mathbf{v} \text{ is consistent with } \mathbf{x}, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Equation (1) is known as the truncated factorization formula (Pearl, 2009), the g-formula (Robins, 1986) or the manipulated density (Spirtes et al., 2000).

Maximal ancestral graphs. A mixed graph $\mathcal{G}$ without directed cycles and almost directed cycles is called *ancestral*. A *maximal ancestral graph* (MAG) is an ancestral graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ where every pair of non-adjacent nodes X and Y in $\mathcal{G}$ can be m-separated by a set $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\}$. A DAG with unobserved variables can be uniquely represented by a MAG on the observed variables that preserves the ancestral and m-separation relationships among the observed variables (page 981 in Richardson and Spirtes, 2002). Since we consider MAGs that do not encode selection bias, the MAGs in this paper can only contain directed $(\to)$ and bi-directed $(\leftrightarrow)$ edges. The MAG of a causal DAG is a *causal* MAG.

Markov equivalence. Several DAGs can encode the same conditional independencies via d-separation. Such DAGs form a *Markov equivalence class* which can be described uniquely by a *completed partially directed acyclic graph* (CPDAG) (Meek, 1995). A CPDAG $\mathcal{C}$ has the same adjacencies as any DAG in the Markov equivalence class described by $\mathcal{C}$. A directed edge $X \to Y$ in a CPDAG $\mathcal{C}$ corresponds to a directed edge $X \to Y$ in every DAG in the Markov equivalence class described by $\mathcal{C}$. For any non-directed edge $X \circ\!\!-\!\!\circ Y$ in a CPDAG $\mathcal{C}$, the Markov equivalence class described by $\mathcal{C}$ contains a DAG with $X \to Y$ and a DAG with $X \leftarrow Y$[1]. Thus, CPDAGs only contain directed $(\to)$ and non-directed $(\circ\!\!-\!\!\circ)$ edges.

Several MAGs can also encode the same conditional independencies via m-separation. Such MAGs form a Markov equivalence class which can be described uniquely by a *partial ancestral graph* (PAG) (Richardson and Spirtes, 2002; Ali et al., 2009). A PAG $\mathcal{P}$ has the same adjacencies as any MAG in the Markov equivalence class described by $\mathcal{P}$. Any non-circle edge-mark in a PAG $\mathcal{P}$ corresponds to that same non-circle edge-mark in every MAG in the Markov equivalence class described by $\mathcal{P}$. We only consider maximally informative PAGs (Zhang, 2008b), that is, for any circle mark $X \circ\!\!\bullet$ in a PAG $\mathcal{P}$, the Markov equivalence class described by $\mathcal{P}$ contains a MAG with $X \leftarrow\!\!\bullet Y$ and a MAG with $X \bullet\!\!-\!\!\bullet Y$. We denote all

1. The non-directed edges in a CPDAG, which we denote as $\circ\!\!-\!\!\circ$, are often denoted as $-$ in the relevant literature, see for example (Meek, 1995). We use $\circ\!\!-\!\!\circ$ instead of $-$ for the sake of consistency among different graph classes.

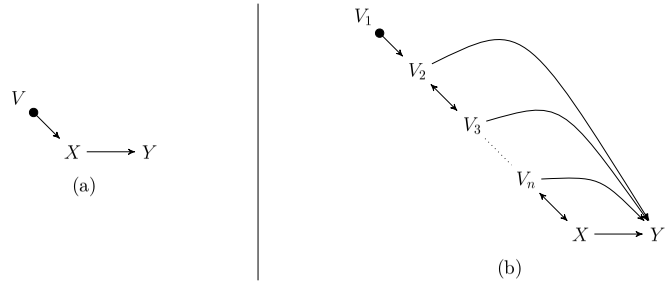


Figure 2: Two configurations where the edge $X \to Y$ is visible. Nodes V and Y must be nonadjacent in 2a, and V_1 and Y must be nonadjacent in 2b.

DAGs (MAGs) in the Markov equivalence class described by a CPDAG (PAG) $\mathcal{G}$ by $[\mathcal{G}]$. The CPDAG (PAG) of a causal DAG (MAG) is a *causal* CPDAG (PAG).

Consistent densities. A density f is *consistent* with a causal DAG $\mathcal{D}$ if the pair $(\mathcal{D}, f)$ forms a causal Bayesian network. A density f is consistent with a causal MAG $\mathcal{M}$ if there exists a causal Bayesian network $(\mathcal{D}', f')$ such that $\mathcal{M}$ represents $\mathcal{D}'$ and f is the observed marginal of f'. A density f is consistent with a causal CPDAG (PAG) $\mathcal{G}$ if it is consistent with a causal DAG (MAG) in $[\mathcal{G}]$.

Visible and invisible edges. All directed edges in DAGs and CPDAGs are said to be visible. Given a MAG $\mathcal{M}$ or a PAG $\mathcal{G}$, a directed edge $X \to Y$ is *visible* if there is a node V not adjacent to Y such that there is an edge $V \bullet\!\!\to X$, or if there is a collider path between V and X that is into X and every non-endpoint node on the path is a parent of Y, see Figure 2 (Zhang, 2006). A visible edge $X \to Y$ means that there are no latent confounders between X and Y. A directed edge $X \to Y$ that is not visible in a MAG $\mathcal{M}$ or a PAG $\mathcal{G}$ is said to be *invisible*. In the FCI algorithm, invisible edges can occur due to orientation rules **R5** - **R10** of Zhang (2008b). When considering MAGs and PAGs that do not encode selection bias, invisible edges occur as a consequence of the orientation rules **R8** - **R10** of Zhang (2008b).

3. The Generalized Adjustment Criterion

Throughout, let $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ represent a DAG, CPDAG, MAG or PAG, and let $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ be pairwise disjoint subsets of $\mathbf{V}$, with $\mathbf{X} \neq \emptyset$ and $\mathbf{Y} \neq \emptyset$. Here, $\mathbf{X}$ represents the set of exposures and $\mathbf{Y}$ represents the set of response variables.

We will define sound and complete graphical conditions for adjustment sets relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$. Thus, if a set $\mathbf{Z}$ satisfies our conditions relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$ (see Definition 4), then it is a valid adjustment set for calculating the causal effect of $\mathbf{X}$ on $\mathbf{Y}$ (see Definition 1), and every existing valid adjustment set satisfies our conditions (see Theorem 5). First, we define what we mean by an adjustment set.

Definition 1 ***(Adjustment set;*** *Maathuis and Colombo, 2015) Let* $\mathbf{X}, \mathbf{Y}$ *and* $\mathbf{Z}$ *be pairwise disjoint node sets in a causal* DAG, CPDAG, MAG *or* PAG $\mathcal{G}$. *Then* $\mathbf{Z}$ *is an adjustment set relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{G}$ *if for any density*[2] f *consistent with* $\mathcal{G}$ *we have*

$$f(\mathbf{y} \mid do(\mathbf{x})) = \begin{cases} f(\mathbf{y} \mid \mathbf{x}) & \text{if } \mathbf{Z} = \emptyset, \\ \int_{\mathbf{z}} f(\mathbf{y} \mid \mathbf{x}, \mathbf{z}) f(\mathbf{z}) d\mathbf{z} & \text{otherwise.} \end{cases} \tag{2}$$

Thus, adjustment sets allow post-intervention densities involving the do-operator (left-hand side of Equation 2) to be identified as specific functions of conditional densities (right-hand side of Equation 2). The latter can be estimated from observational data. As a result, adjustment sets are important for the computation of causal effects. This is illustrated in Example 1 for the special case of multivariate Gaussian densities.

Example 1 *Suppose* f *is a multivariate Gaussian density that is consistent with a causal* DAG $\mathcal{D}$. *Let* $\mathbf{Z} \neq \emptyset$ *be an adjustment set relative to two distinct variables* X *and* Y *in* $\mathcal{D}$ *such that* $\mathbf{Z} \cap \{X \cup Y\} = \emptyset$. *Then*

$$\begin{aligned} E(Y \mid do(x)) &= \int_y y f(y \mid do(x)) dy = \int_y y \int_{\mathbf{z}} f(y \mid x, \mathbf{z}) f(\mathbf{z}) d\mathbf{z} dy \\ &= \int_{\mathbf{z}} \int_y y f(y \mid x, \mathbf{z}) dy f(\mathbf{z}) d\mathbf{z} = \int_{\mathbf{z}} E(Y \mid x, \mathbf{z}) f(\mathbf{z}) d\mathbf{z} \\ &= \int_{\mathbf{z}} (\alpha + \gamma x + \beta^T \mathbf{z}) f(\mathbf{z}) d\mathbf{z} = \alpha + \gamma x + \beta^T E(\mathbf{Z}), \end{aligned}$$

where we use the fact that all conditional expectations in a multivariate Gaussian distribution are linear, so that $E(Y \mid x, \mathbf{z}) = \alpha + \gamma x + \beta^T \mathbf{z}$, *for some* $\alpha, \gamma \in \mathbb{R}$ *and* $\beta \in \mathbb{R}^{|\mathbf{z}|}$. *Defining the total causal effect of* X *on* Y *as* $\frac{\partial}{\partial x} E(Y \mid do(x))$, *we obtain that the total causal effect of* X *on* Y *is* γ, *that is, the regression coefficient of* X *in the regression of* Y *on* X *and* $\mathbf{Z}$.

Our first goal in this paper is to give a graphical criterion (see Definition 4) that is equivalent to Definition 1. To this end, we introduce some additional terminology.

Definition 2 ***(Amenability)*** *Let* $\mathbf{X}$ *and* $\mathbf{Y}$ *be disjoint node sets in a* DAG, CPDAG, MAG *or* PAG $\mathcal{G}$. *Then* $\mathcal{G}$ *is said to be amenable relative to* $(\mathbf{X}, \mathbf{Y})$ *if every proper possibly directed path from* $\mathbf{X}$ *to* $\mathbf{Y}$ *in* $\mathcal{G}$ *starts with a visible edge out of* $\mathbf{X}$.

If $\mathcal{G}$ is a MAG, then Definition 2 reduces to the notion of amenability as introduced in van der Zander et al. (2014). The intuition behind the concept of amenability is the following: In MAGs and PAGs, directed edges $X \to Y$ can represent causal effects, but also mixtures of causal effects and latent confounding. For instance, when the graph $X \to Y$ is interpreted as a DAG, the empty set is a valid adjustment set with respect to (X, Y). When the same graph is interpreted as a MAG, it can still represent the DAG $X \to Y$, but also the DAG $X \to Y$ with an additional non-causal path $X \leftarrow L \to Y$ where L is latent.

2. We use the notation for continuous random variables throughout. The discrete analogues should be obvious.

In CPDAGs and PAGs, there are edges with unknown direction. This complicates adjustment because paths containing such edges can correspond to causal paths in some represented DAGs and to non-causal paths in others. For example, the CPDAG $X \circ\!\!-\!\!\circ Y$ represents the DAGs $X \to Y$ and $X \leftarrow Y$. Amenable graphs are graphs where these problems do not occur.

Definition 3 *(**Forbidden set;** $\mathrm{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G})$) Let $\mathbf{X}$ and $\mathbf{Y}$ be disjoint node sets in a* DAG, CPDAG, MAG *or* PAG $\mathcal{G}$. *Then the forbidden set relative to $(\mathbf{X},\mathbf{Y})$ is defined as*

$$\mathrm{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G}) = \{W' \in \mathbf{V} : W' \in \mathrm{PossDe}(W,\mathcal{G}), \text{ for some } W \notin \mathbf{X} \text{ which lies on a proper possibly directed path from } \mathbf{X} \text{ to } \mathbf{Y} \text{ in } \mathcal{G}\}.$$

Definition 4 *(**Generalized adjustment criterion**) Let $\mathbf{X},\mathbf{Y}$ and $\mathbf{Z}$ be pairwise disjoint node sets in a* DAG, CPDAG, MAG *or* PAG $\mathcal{G}$. *Then $\mathbf{Z}$ satisfies the generalized adjustment criterion relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{G}$ if the following three conditions hold:*

*(**Amenability**) $\mathcal{G}$ is adjustment amenable relative to $(\mathbf{X},\mathbf{Y})$, and*

*(**Forbidden set**) $\mathbf{Z} \cap \mathrm{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G}) = \emptyset$, and*

*(**Blocking**) all proper definite status non-causal paths from $\mathbf{X}$ to $\mathbf{Y}$ are blocked by $\mathbf{Z}$ in $\mathcal{G}$.*

If $\mathcal{G}$ is a DAG (MAG), our criterion reduces to the adjustment criterion of Shpitser (2012), (van der Zander et al., 2014) (see Definition 55 in Appendix E). For consistency, however, we will refer to the generalized adjustment criterion for all graph types.

We note that the amenability condition does not depend on $\mathbf{Z}$. In other words, if the amenability condition is violated, then no set satisfies the generalized adjustment criterion relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{G}$. The forbidden set contains nodes that cannot be used for adjustment. We will try to give some intuition. For simplicity, we consider $\mathbf{X} = \{X\}$ and $\mathbf{Y} = \{Y\}$ in a DAG $\mathcal{D}$, and we are interested in estimating the total causal effect of X on Y in $\mathcal{D}$. It is clear that nodes on any causal path from X to Y in $\mathcal{D}$ should not be included in the set used for adjustment, since including such nodes would block the causal path.

To understand why we cannot include descendants of nodes on a causal path from X to Y in $\mathcal{D}$ (except for descendants of X), it is useful to consider *walks* from X to Y in $\mathcal{D}$, that is, sequences of nodes $\langle X,\dots,Y\rangle$ in which every pair of successive nodes is adjacent in $\mathcal{D}$ (but the nodes are not necessarily distinct). A walk $r = \langle X = V_0, V_1, \dots, V_k = Y\rangle$ is non-causal if $V_i \leftarrow\!\!\circ V_{i+1}$ for at least one $i \in \{0,\dots,k-1\}$. A walk r from X to Y in $\mathcal{D}$ is *connecting* given a set of nodes $\mathbf{Z}$ if $\mathbf{Z}$ contains all colliders on r and no non-collider on r is in $\mathbf{Z}$. If a walk r is not connecting given $\mathbf{Z}$, then r is *blocked* by $\mathbf{Z}$. Koster (2002) proved that there is a walk from X to Y that is connecting given $\mathbf{Z}$ in $\mathcal{D}$ if and only if there is path from X to Y that is d-connecting given $\mathbf{Z}$ in $\mathcal{D}$.

Intuitively, all non-causal walks from X to Y should be blocked in order to estimate the total causal effect of X on Y. Now, consider a path p of the form $X \to V_1 \to \cdots \to V_k \to Y$ in $\mathcal{G}$. Assume $V_i \notin \mathbf{Z}$, for all $i \in \{1,\dots,k\}$. Including a descendant A of V_i in the set $\mathbf{Z}$ leads to walk of the form $X \to \cdots \to V_i \to \cdots \to A \leftarrow \cdots \leftarrow V_i \to \cdots \to V_k \to Y$ being connecting given $\mathbf{Z}$ in $\mathcal{G}$. Hence, including A in the adjustment set opens a non-causal walk from X to Y in $\mathcal{G}$.

We now give the main theorem of this section. Corresponding examples can be found in Section 3.1 and the proof of the theorem is given in Section 3.2.

Theorem 5 *Let $\mathbf{X},\mathbf{Y}$ and $\mathbf{Z}$ be pairwise disjoint node sets in a causal* DAG, CPDAG, MAG *or* PAG $\mathcal{G}$. *Then $\mathbf{Z}$ is an adjustment set relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{G}$ (see Definition 1) if and only if $\mathbf{Z}$ satisfies the generalized adjustment criterion relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{G}$ (see Definition 4).*

Verifying the blocking condition by checking all paths requires keeping track of which paths are non-causal and hence, scales poorly to larger graphs. We therefore give an alternative definition of this condition which relies on m-separation in a so-called proper back-door graph.

Definition 6 *(**Proper back-door graph**; $\mathcal{G}^{pbd}_{\mathbf{XY}}$) Let $\mathbf{X}$ and $\mathbf{Y}$ be disjoint node sets in a* DAG, CPDAG, MAG *or* PAG $\mathcal{G}$. *The proper back-door graph $\mathcal{G}^{pbd}_{\mathbf{XY}}$ is obtained from $\mathcal{G}$ by removing all visible edges out of $\mathbf{X}$ that are on proper possibly directed paths from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}$.*

If $\mathcal{G}$ is a DAG or MAG, then Definition 6 reduces to the definition of proper back-door graphs as introduced in van der Zander et al. (2014).

Theorem 7 *Replacing the Blocking condition in Definition 4 with:*

*(**Separation**) $\mathbf{Z}$ m-separates $\mathbf{X}$ and $\mathbf{Y}$ in $\mathcal{G}^{pbd}_{\mathbf{XY}}$,*

results in a criterion that is equivalent to the generalized adjustment criterion.

If $\mathcal{G}$ is a DAG or MAG, then Theorem 7 reduces to Theorem 4.6 in van der Zander et al. (2014).

3.1 Examples

We now provide some examples that illustrate how the generalized adjustment criterion can be applied.

Example 2 *We return to the* CPDAG $\mathcal{C}$ *in Figure 1(a). $\mathcal{C}$ is amenable relative to (X,Y) and $\mathrm{Forb}(X,Y,\mathcal{C}) = \{Y\}$. One can easily verify that any superset of $\{Z,A\}$ or of $\{Z,B\}$ that does not contain X or Y satisfies the generalized adjustment criterion relative to (X,Y) in $\mathcal{C}$.*

Example 3 *To illustrate the concept of amenability, consider Figure 3 with a* PAG $\mathcal{P}$ *in (a), and two* MAGs $\mathcal{M}_1$ *and* $\mathcal{M}_2$ *in $[\mathcal{P}]$ in (b) and (c). The graphs $\mathcal{P}$ and $\mathcal{M}_1$ are not amenable relative to (X,Y). For $\mathcal{P}$ this is due to the path $X \circ\!\!-\!\!\circ Y$, and for $\mathcal{M}_1$ this is due*

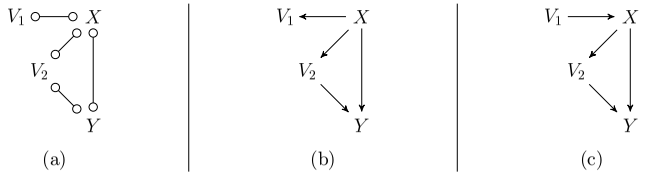


Figure 3: (a) PAG $\mathcal{P}$, (b) MAG $\mathcal{M}_1$, (c) MAG $\mathcal{M}_2$ used in Example 3.

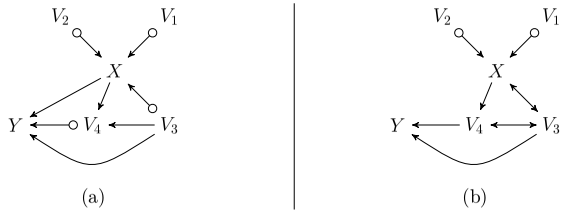


Figure 4: (a) PAG $\mathcal{P}_1$, (b) PAG $\mathcal{P}_2$ used in Example 4 and Example 6.

to the invisible edge $X \to Y$ (which implies that $\mathcal{M}_1$ also represents a DAG that contains a hidden confounder that is an ancestor of both X and Y). On the other hand, $\mathcal{M}_2$ is amenable relative to (X, Y), since the edges $X \to Y$ and $X \to V_2$ are visible due to the edge $V_1 \to X$, with V_1 not adjacent to Y or V_2. Since there are no proper definite status non-causal paths from X to Y in $\mathcal{M}_2$, it follows that the empty set satisfies the generalized adjustment criterion relative to (X, Y) in $\mathcal{M}_2$. Finally, note that $\mathcal{M}_1$ could also be interpreted as a DAG*. In that case, it would be amenable relative to (X, Y). This shows that amenability depends crucially on the interpretation of the graph.*

Example 4 *Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be the* PAG*s in Figure 4(a) and Figure 4(b), respectively. Both* PAG*s are amenable relative to (X, Y). We will show that there is an adjustment set relative to (X, Y) in $\mathcal{P}_1$ but not in $\mathcal{P}_2$. This illustrates that amenability is not a sufficient criterion for the existence of an adjustment set.*

We first consider $\mathcal{P}_1$. Note that $\mathrm{Forb}(X, Y, \mathcal{P}_1) = \{V_4, Y\}$ *and that there are two proper definite status non-causal paths from X to Y: $X \leftrightarrow\circ V_3 \to Y$ and $X \to V_4 \leftarrow V_3 \to Y$. Path $X \leftrightarrow\circ V_3 \to V_4 \circ\!\!\to Y$ is not of definite status, as node V_4 is not of definite status on this path. Both proper definite status non-causal paths from X to Y are blocked by any set containing V_3. Hence, all sets satisfying the generalized adjustment criterion relative to (X, Y) in $\mathcal{P}_1$ are: $\{V_3\}$, $\{V_1, V_3\}$, $\{V_2, V_3\}$ and $\{V_1, V_2, V_3\}$.*

In $\mathcal{P}_2$, we have $\mathrm{Forb}(X, Y, \mathcal{P}_2) = \mathrm{Forb}(X, Y, \mathcal{P}_1) = \{V_4, Y\}$*, and there are three proper definite status non-causal paths from X to Y in $\mathcal{P}_2$: p_1 of the form $X \leftrightarrow V_3 \to Y$, p_2 of the form $X \leftrightarrow V_3 \leftrightarrow V_4 \to Y$ and p_3 of the form $X \to V_4 \leftrightarrow V_3 \to Y$. To block p_1, we must use V_3, and this implies that we must use V_4 to block p_2. But $V_4 \in \mathrm{Forb}(X, Y, \mathcal{P}_2)$. Hence, no set satisfies the generalized adjustment criterion $\mathbf{Z}$ relative to (X, Y) in $\mathcal{P}_2$.*

3.2 Proof of Theorem 5

To prove that the generalized adjustment criterion is sound and complete for adjustment (Theorem 5), we build on the fact that the adjustment criterion for DAGs and MAGs is sound and complete for adjustment. The adjustment criterion for DAGs was first presented in Shpitser et al. (2010) and was modified in the unpublished addendum (Shpitser, 2012). In Appendix E we give the revised version of the criterion, as well as new soundness and completeness proofs, relying only on basic probability calculus, linear algebra and the do-calculus rules. The adjustment criterion for MAGs was presented and proved to be sound and complete for adjustment in van der Zander et al. (2014, see Theorem 5.8). However, van der Zander et al. (2014)'s proof assumed the soundness and completeness of the adjustment criterion for DAGs given in Shpitser (2012). This latter claim is proved here in Theorem 56 in Appendix E.

Lastly, our proof of Theorem 5 heavily relies on the three lemmas given below. Their proofs can be found in Appendix B.

Lemma 8 *Let $\mathbf{X}$ and $\mathbf{Y}$ be disjoint node sets in a* CPDAG (PAG) *$\mathcal{G}$. If $\mathcal{G}$ is amenable (see Definition 4) relative to $(\mathbf{X}, \mathbf{Y})$, then every* DAG (MAG) *in $[\mathcal{G}]$ is amenable relative to $(\mathbf{X}, \mathbf{Y})$. On the other hand, if $\mathcal{G}$ violates the amenability condition relative to $(\mathbf{X}, \mathbf{Y})$, then there is no adjustment set relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$ (see Definition 1).*

Lemma 9 *Let $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$ be pairwise disjoint node sets in a* CPDAG (PAG) *$\mathcal{G}$. If $\mathcal{G}$ is amenable relative to $(\mathbf{X}, \mathbf{Y})$, then the following statements are equivalent:*

(i) $\mathbf{Z}$ *satisfies the forbidden set condition (see Definition 4) relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$.*

(ii) $\mathbf{Z}$ *satisfies the forbidden set condition relative to $(\mathbf{X}, \mathbf{Y})$ in every* DAG (MAG) *in $[\mathcal{G}]$.*

Lemma 10 *Let $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$ be pairwise disjoint node sets in a* CPDAG (PAG) *$\mathcal{G}$. If $\mathcal{G}$ is amenable relative to $(\mathbf{X}, \mathbf{Y})$, and $\mathbf{Z}$ satisfies the forbidden set condition relative to $(\mathbf{X}, \mathbf{Y})$, then the following statements are equivalent:*

(i) $\mathbf{Z}$ *satisfies the blocking condition (see Definition 4) relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$.*

(ii) $\mathbf{Z}$ *satisfies the blocking condition relative to $(\mathbf{X}, \mathbf{Y})$ in every* DAG (MAG) *in $[\mathcal{G}]$.*

(iii) $\mathbf{Z}$ *satisfies the blocking condition relative to $(\mathbf{X}, \mathbf{Y})$ in a* DAG *$\mathcal{D}$ (*MAG *$\mathcal{M}$) in $[\mathcal{G}]$.*

Proof of Theorem 5. If $\mathcal{G}$ is a DAG (MAG), then our criterion reduces to the adjustment criterion from Shpitser (2012) (van der Zander et al., 2014) which is sound and complete for adjustment (Theorem 56 in Appendix E, Theorem 5.8 in van der Zander et al., 2014). Hence, we only consider the case that $\mathcal{G}$ is a CPDAG (PAG).

Suppose first that $\mathbf{Z}$ satisfies the generalized adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in the CPDAG (PAG) $\mathcal{G}$. We need to show that $\mathbf{Z}$ is an adjustment set (see Definition 1) relative to $(\mathbf{X}, \mathbf{Y})$ in every DAG $\mathcal{D}$ (MAG $\mathcal{M}$) in $[\mathcal{G}]$. By applying Lemmas 8, 9 and 10 in turn, it directly follows that $\mathbf{Z}$ satisfies the generalized adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in every DAG $\mathcal{D}$ (MAG $\mathcal{M}$) in $[\mathcal{G}]$. Since the generalized adjustment criterion is

sound for adjustment in DAGs (MAGs) (see Theorem 58 in Appendix E and Theorem 5.8 in van der Zander et al., 2014), $\mathbf{Z}$ is an adjustment set relative to $(\mathbf{X}, \mathbf{Y})$ in every $\mathcal{D}$ ($\mathcal{M}$) in $[\mathcal{G}]$.

To prove the other direction, suppose that $\mathbf{Z}$ does not satisfy the generalized adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$. First, suppose that $\mathcal{G}$ violates the amenability condition relative to $(\mathbf{X}, \mathbf{Y})$. Then by Lemma 8, there is no adjustment set relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$. Otherwise, $\mathcal{G}$ is amenable relative to $(\mathbf{X}, \mathbf{Y})$, but $\mathbf{Z}$ violates the forbidden set condition or the blocking condition. We need to show $\mathbf{Z}$ is not an adjustment set in at least one DAG $\mathcal{D}$ (MAG $\mathcal{M}$) in $[\mathcal{G}]$. Suppose $\mathbf{Z}$ violates the forbidden set condition. Then by Lemma 9, it follows that there exists a DAG $\mathcal{D}$ (MAG $\mathcal{M}$) in $[\mathcal{G}]$ such that $\mathbf{Z}$ does not satisfy the generalized adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$ ($\mathcal{M}$). Since the generalized adjustment criterion is complete for adjustment in DAGs (MAGs) (see Theorem 57 in Appendix E and Theorem 5.8 in van der Zander et al., 2014), it follows that $\mathbf{Z}$ is not an adjustment set relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$ ($\mathcal{M}$). Otherwise, suppose $\mathbf{Z}$ satisfies the forbidden set condition, but violates the blocking condition. Then by Lemma 10, it follows that there is a DAG $\mathcal{D}$ (MAG $\mathcal{M}$) in $[\mathcal{G}]$ such that $\mathbf{Z}$ does not satisfy the generalized adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$ ($\mathcal{M}$). Since the generalized adjustment criterion is complete for adjustment in DAGs (MAGs), it follows that $\mathbf{Z}$ is not an adjustment set relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$ ($\mathcal{M}$). ■

4. Constructing Adjustment Sets

We now present approaches to construct adjustment sets. First, in Theorem 11, we discuss a pre-processing of the node set $\mathbf{X}$ that in conjunction with our generalized adjustment criterion can help identify $f(\mathbf{y}|do(\mathbf{x}))$ in DAG, CPDAG, MAG or PAG $\mathcal{G}$ via adjustment.

As mentioned before, if $f(\mathbf{y}|do(\mathbf{x}))$ is not identifiable via adjustment in $\mathcal{G}$, it may be identifiable through other means. In particular, if $f(\mathbf{y}|do(\mathbf{x}))$ is not identifiable via adjustment in $\mathcal{G}$, there may be a set $\mathbf{X}' \subseteq \mathbf{X}$ such that $f(\mathbf{y}|do(\mathbf{x})) = f(\mathbf{y}|do(\mathbf{x}'))$, and $f(\mathbf{y}|do(\mathbf{x}'))$ is identifiable via adjustment in $\mathcal{G}$. One such example is given in Theorem 11.

Next, we introduce Theorem 14 that will allow us to easily construct adjustment sets that do not contain certain nodes, if any such adjustment set exists. We illustrate the results of Theorem 14 with examples in Section 4.1 and give the proof of this theorem in Section 4.2. In Section 4.3 we explain how to leverage previous results of van der Zander et al. (2014) to enumerate all (minimal) adjustment sets, and discuss how to implement this procedure efficiently.

Theorem 11 *Let $\mathbf{X}$ and $\mathbf{Y}$ be disjoint node sets in a causal DAG, CPDAG, MAG or PAG $\mathcal{G}$. Let $\mathbf{X}' \subseteq \mathbf{X}$ such that there is no possibly directed path from $\mathbf{X} \setminus \mathbf{X}'$ to $\mathbf{Y}$ that is proper with respect to $\mathbf{X}$. Then*

$$f(\mathbf{y}|do(\mathbf{x})) = \begin{cases} f(\mathbf{y}) & \text{if } \mathbf{X}' = \emptyset, \\ f(\mathbf{y}|do(\mathbf{x}')) & \text{otherwise.} \end{cases}$$

Furthermore, if $\mathbf{X}' \neq \emptyset$ and if $\mathbf{Z}$ is an adjustment set relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$, then $\mathbf{Z}$ is an adjustment set relative to $(\mathbf{X}', \mathbf{Y})$ in $\mathcal{G}$.

Following Theorem 11, we recommend pre-processing the set $\mathbf{X}$ as follows: remove all nodes $X \in \mathbf{X}$ that do not have a possibly directed path to $\mathbf{Y}$ which is proper with respect to $\mathbf{X}$

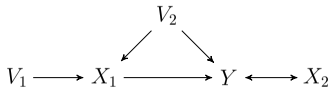


Figure 5: MAG $\mathcal{M}$.

in $\mathcal{G}$. In other words, if $\mathbf{W}$ is the set of all nodes that have a possibly directed path to $\mathbf{Y}$ which is proper with respect to $\mathbf{X}$ in $\mathcal{G}$, then $\mathbf{X}' = \mathbf{X} \cap \mathbf{W}$. Then by choice of $\mathbf{W}$ and the proof of Theorem 11, all nodes in $\mathbf{X}'$ have a possibly directed path to $\mathbf{Y}$ that is proper with respect to $\mathbf{X}'$.

By Theorem 11, this pre-processing of $\mathbf{X}$ cannot hurt in identifying $f(\mathbf{y}|do(\mathbf{x}))$ via adjustment. Moreover, there are cases when this pre-processing helps to identify $f(\mathbf{y}|do(\mathbf{x}))$ via adjustment. For example, in the MAG $\mathcal{M}$ in Figure 5, there is no adjustment set relative to $(\{X_1, X_2\}, Y)$. However, there is no possibly directed path from X_2 to Y that is proper with respect to $\{X_1, X_2\}$. Furthermore, $\{V_2\}$ is an adjustment set relative to (X_1, Y). Hence, by Theorem 11 and Theorem 5, $f(y|do(x_1, x_2)) = f(y|do(x_1)) = \int_{v_2} f(y|x_1, v_2) f(v_2) dv_2$.

We now introduce two definitions that will be used in Theorem 14. First, we define the set $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$ relative to disjoint node sets $\mathbf{X}$ and $\mathbf{Y}$ in a DAG, CPDAG, MAG or PAG $\mathcal{G}$.

Definition 12 *(Adjust$(\mathbf{X}, \mathbf{Y}, \mathcal{G})$) Let $\mathbf{X}$ and $\mathbf{Y}$ be disjoint node sets in a* DAG, CPDAG, MAG *or* PAG *$\mathcal{G}$. We define*

$$\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) = \text{PossAn}(\mathbf{X} \cup \mathbf{Y}, \mathcal{G}) \setminus (\mathbf{X} \cup \mathbf{Y} \cup \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G})). \tag{3}$$

If $\mathcal{G}$ is a DAG or MAG, then Definition 12 reduces to the definition of $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$ in van der Zander et al. (2014), that is, $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) = \text{An}(\mathbf{X} \cup \mathbf{Y}, \mathcal{G}) \setminus (\mathbf{X} \cup \mathbf{Y} \cup \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G}))$.

Definition 13 ***(Descendral set)*** *Let $\mathbf{I}$ be a node set in a* DAG, CPDAG, MAG *or* PAG *$\mathcal{G}$. Then $\mathbf{I}$ is called descendral in $\mathcal{G}$ if $\mathbf{I} = \text{PossDe}(\mathbf{I}, \mathcal{G})$.*

A descendral set is in a sense analogous to an ancestral set, which is a set containing all ancestors of itself. Note that $\text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$ and $\text{PossDe}(\mathbf{X}, \mathcal{G})$ are both descendral sets. This property will be used throughout the proofs. Note that if $\mathbf{A}$ is an ancestral set and $\mathbf{B}$ is a descendral set, then $\mathbf{A} \setminus \mathbf{B}$ is an ancestral set and $\mathbf{B} \setminus \mathbf{A}$ is a descendral set.

Theorem 14 ***(Constructive set)*** *Let $\mathbf{X}$ and $\mathbf{Y}$ be disjoint node sets in a* DAG, CPDAG, MAG *or* PAG *$\mathcal{G}$. Let $\mathbf{I} \supseteq \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$ be a descendral set in $\mathcal{G}$. Then there exists a set $\mathbf{Z}$ that satisfies the generalized adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$ such that $\mathbf{Z} \cap \mathbf{I} = \emptyset$ if and only if $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \setminus \mathbf{I}$ satisfies the generalized adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$.*

The smallest set we can take for $\mathbf{I}$ in Theorem 14 is $\text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$. This leads to Corollary 15. Pearl's back-door criterion does not allow using descendants of $\mathbf{X}$ in a DAG $\mathcal{D}$. Moreover, the generalized back-door criterion does not allow using possible descendants of $\mathbf{X}$ in a DAG, CPDAG, MAG or PAG $\mathcal{G}$. Thus, another natural set to consider for $\mathbf{I}$ is $\text{PossDe}(\mathbf{X}, \mathcal{G})$. We will use $\mathbf{I} = \text{PossDe}(\mathbf{X}, \mathcal{G})$ and Theorem 14 in Section 5 to define sets that satisfy generalized back-door criterion and Pearl's back-door criterion.

Corollary 15 *Let $\mathbf{X}$ and $\mathbf{Y}$ be disjoint node sets in a* DAG, CPDAG, MAG *or* PAG $\mathcal{G}$. *The following statements are equivalent:*

(i) There exists an adjustment set relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$.

(ii) $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$ *satisfies the generalized adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$.*

(iii) $\mathcal{G}$ is amenable relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$ and $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$ *satisfies the blocking condition relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$.*

4.1 Examples

We now provide some examples that illustrate the construction of adjustment sets.

Example 5 *Consider again the* CPDAG $\mathcal{C}$ *in Figure 1(a). As previously discussed in Example 2, $\mathcal{C}$ is amenable relative to (X, Y). The set* $\text{Adjust}(X, Y, \mathcal{C}) = \{X, Y, I, A, Z, B\} \setminus \{X, Y\} = \{I, A, Z, B\}$ *satisfies the blocking condition relative to (X, Y) in $\mathcal{C}$. Hence, by Corollary 15, $\{I, A, Z, B\}$ satisfies the generalized adjustment criterion relative to (X, Y) in $\mathcal{C}$.*

Example 6 *Consider again the* PAG*s $\mathcal{P}_1$ and $\mathcal{P}_2$ in Figure 4(a) and Figure 4(b), respectively. As previously discussed in Example 4, both $\mathcal{P}_1$ and $\mathcal{P}_2$ are amenable relative to (X, Y).*

In $\mathcal{P}_1$, $\text{Forb}(X, Y, \mathcal{P}_1) = \{V_4, Y\}$, *so* $\text{Adjust}(X, Y, \mathcal{P}_1) = \{X, Y, V_1, V_2, V_3, V_4\} \setminus \{X, Y, V_4\} = \{V_1, V_2, V_3\}$. *Since $\{V_1, V_2, V_3\}$ satisfies the blocking condition relative to (X, Y) in $\mathcal{G}$, it follows that $\{V_1, V_2, V_3\}$ satisfies the generalized adjustment criterion relative to (X, Y) in $\mathcal{G}$.*

In $\mathcal{P}_2$, again $\text{Forb}(X, Y, \mathcal{P}_2) = \{V_4, Y\}$, *so* $\text{Adjust}(X, Y, \mathcal{P}_2) = \{X, Y, V_1, V_2, V_3, V_4\} \setminus \{X, Y, V_4\} = \{V_1, V_2, V_3\}$. *Since $\{V_1, V_2, V_3\}$ does not block the path $X \leftrightarrow V_3 \leftrightarrow V_4 \rightarrow Y$ it does not satisfy the blocking condition relative to (X, Y) in $\mathcal{G}$. Hence, Corollary 15 implies that there is no adjustment set relative to (X, Y) in $\mathcal{P}_2$.*

4.2 Proof of Theorem 14

To prove Theorem 14 we heavily rely on Lemma 16 and Lemma 17 given below. Their proofs are given in Appendix C. Lemma 17 is related to Lemma 1 from Richardson (2003) (see Lemma 37 in Appendix A).

Lemma 16 *Let $\mathbf{X}$ and $\mathbf{Y}$ be disjoint node sets in a* DAG, CPDAG, MAG *or* PAG $\mathcal{G}$. *Let $\mathbf{I} \supseteq \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$ be a descendral set in $\mathcal{G}$ (see Definition 13). If there is a proper definite status non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}$ that is m-connecting given* $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \setminus \mathbf{I}$, *then there is a path p from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}$ such that:*

(i) p is a proper definite status non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}$, and

(ii) all colliders on p are in $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \setminus \mathbf{I}$, *and*

(iii) all definite non-colliders on p are in $\mathbf{I}$, and

(iv) for any collider C on p, there is an unshielded possibly directed path from C to $\mathbf{X} \cup \mathbf{Y}$, that starts with $\circ\!\!\rightarrow$ or $\rightarrow$.

Lemma 17 *Let $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$ be pairwise disjoint node sets in a* DAG, CPDAG, MAG *or* PAG $\mathcal{G}$. *Let $\mathbf{I} \supseteq \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$ be a node set in $\mathcal{G}$ such that $\mathbf{Z} \cap \mathbf{I} = \emptyset$. Let p be a path from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}$ such that:*

(i) p is a proper definite status non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}$, and

(ii) all colliders on p are in $\text{An}(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}, \mathcal{G}) \setminus \mathbf{I}$, *and*

(iii) no definite non-collider on p is in $\mathbf{Z}$.

Then there is a proper definite status non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ that is m-connecting given $\mathbf{Z}$ in $\mathcal{G}$.

Proof of Theorem 14. We only prove the non-trivial direction. Thus, assume there is a set $\mathbf{Z}$ satisfying the generalized adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$ such that $\mathbf{Z} \cap \mathbf{I} = \emptyset$. We will prove that $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \setminus \mathbf{I}$ satisfies the generalized adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$.

Since $\mathbf{Z}$ satisfies the generalized adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$, $\mathcal{G}$ is amenable relative to $(\mathbf{X}, \mathbf{Y})$. Additionally, since $\text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \subseteq \mathbf{I}$, $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \setminus \mathbf{I}$ satisfies the forbidden set condition relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$. It is only left to prove that $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \setminus \mathbf{I}$ satisfies the blocking condition relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$.

Suppose for a contradiction that there is a proper definite status non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ that is m-connecting given $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \setminus \mathbf{I}$. Then we can choose a path p^* in $\mathcal{G}$ that satisfies (i)−(iv) in Lemma 16. Then p^* also satisfies (i) in Lemma 17. By (iii) in Lemma 16, every definite non-collider on p^* is in $\mathbf{I}$. Since $\mathbf{Z} \cap \mathbf{I} = \emptyset$, no definite non-collider on p^* is in $\mathbf{Z}$. So p^* satisfies (iii) in Lemma 17. Also, since by (iv) in Lemma 16 there is a possibly directed unshielded path q^* from every collider C on p^* to $\mathbf{X} \cup \mathbf{Y}$ that starts with $C \circ\!\!\rightarrow$, Lemma 42 implies that any other edge on q^* (if there is any) is directed in $\mathcal{G}$.

Then if $\mathcal{G}$ is a DAG, CPDAG or MAG, it follows from (iv) in Lemma 16 that all colliders on p^* are in $\text{An}(\mathbf{X} \cup \mathbf{Y}, \mathcal{G})$. Combining this with (ii) in Lemma 16 implies that all colliders on p^* are in $\text{An}(\mathbf{X} \cup \mathbf{Y}, \mathcal{G}) \setminus \mathbf{I}$, so that p^* satisfies (ii) in Lemma 17. Hence, if $\mathcal{G}$ is a DAG, CPDAG or MAG, all conditions of Lemma 17 are satisfied, which implies that $\mathbf{Z}$ does not satisfy the blocking condition relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$.

Thus, assume $\mathcal{G}$ is a PAG. Let $\mathcal{M} \in [\mathcal{G}]$ be a MAG obtained from $\mathcal{G}$ by first replacing all partially directed edges $\circ\!\!\rightarrow$ by directed edges $\rightarrow$, and then orienting all non-directed edges $\circ\!\!-\!\!\circ$ as a DAG without unshielded colliders (see Lemma 43 in Appendix A). Let p be the path in $\mathcal{M}$ corresponding to p^* in $\mathcal{G}$. Then p satisfies (i) and (iii) in Lemma 17. By the choice of $\mathcal{M}$ and Lemma 42, any possibly directed unshielded path q^* in $\mathcal{G}$ that starts with a partially

directed edge $\circ\!\!\rightarrow$, corresponds to a directed path q in $\mathcal{M}$. Hence, by (iv) in Lemma 16, every collider on p is in $\mathrm{An}(\mathbf{X} \cup \mathbf{Y}, \mathcal{M})$. Since p^* satisfies (ii) in Lemma 16 in $\mathcal{G}$, no collider on p^* is in $\mathbf{I}$. Hence, also no collider on p is in $\mathbf{I}$. Then all colliders on p are in $\mathrm{An}(\mathbf{X} \cup \mathbf{Y}, \mathcal{M}) \setminus \mathbf{I}$. Additionally, since $\mathbf{I} \supseteq \mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$, and $\mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \supseteq \mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{M})$, it follows that p satisfies (ii) in Lemma 17. Thus, all conditions of Lemma 17 are satisfied, which implies that $\mathbf{Z}$ does not satisfy the blocking condition relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{M}$. This contradicts Lemma 10. ■

4.3 Implementation

We now discuss how one can implement the generalized adjustment criterion in an algorithmically efficient manner, and describe our implementation in the software `dagitty` and the R package `pcalg`.

Verification of the criterion. Given a DAG, CPDAG, MAG or PAG $\mathcal{G}$ and three disjoint node sets $\mathbf{X}, \mathbf{Y}$, and $\mathbf{Z}$, we wish to test whether $\mathbf{Z}$ fulfills the generalized adjustment criterion with respect to $(\mathbf{X}, \mathbf{Y})$. Of course, we could do this simply by verifying the three conditions of the generalized adjustment criterion (see Definition 4). However, the blocking condition is a statement about individual paths, which can pose problems for large graphs.

As a worst-case example, consider a DAG with X, Y and p remaining variables. Let every pair of variables be connected by an edge. Then the DAG contains $\sum_{i=0}^{p} p!/i! \approx p!\, e$ paths from X to Y. Thus, a direct implementation of the generalized adjustment criterion has an exponential runtime in p. Still, a verbatim implementation of the criterion can be useful for verification and didactic purposes, as well as for sparse graphs with few paths, and we therefore provide one in the function `gac` of the R package `pcalg`.

The key result for implementing the criterion in an efficient manner is Theorem 7, which replaces the path blocking condition by an m-separation condition in a subgraph of $\mathcal{G}$, the proper back-door graph $\mathcal{G}_{\mathbf{XY}}^{pbd}$. This condition can be checked efficiently by a simple depth-first or breadth-first graph traversal, known as the "Bayes-Ball algorithm" (Shachter, 1998). Specifically, for graphs represented as adjacency lists, the runtime is $O(|p| + |\mathbf{E}|)$ where $|\mathbf{E}|$ is the number of edges. Our implementation of this method can be accessed via the function `isAdjustmentSet` of the R package `dagitty`.

Constructing adjustment sets. Given a DAG, CPDAG, MAG or PAG $\mathcal{G}$ and two disjoint variable sets $\mathbf{X}, \mathbf{Y}$, we wish to find one or several sets $\mathbf{Z}$ that fulfill the generalized adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$. If a single set is sufficient, we can directly apply the main result of Section 4 and construct $\mathrm{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$ (see Definition 12) and verify whether it satisfies the generalized adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$. Since this set is defined in terms of (possible) ancestors of $\mathbf{X}$ and $\mathbf{Y}$, it can be constructed by graph traversal in linear time. However, $\mathrm{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$ can be a large set, since it contains all (possible) ancestors of $\mathbf{X}$ and $\mathbf{Y}$ except the forbidden nodes. This may result in a loss of statistical precision.

To avoid this, it is of interest to construct all possible adjustment sets. Again, Theorem 7 is key to achieving this: it allows us to use the algorithmic framework developed by van der Zander et al. (2014) for constructing and enumerating m-separating sets in DAGs and MAGs. For DAGs and MAGs, this can be directly applied. We propose the following procedure for CPDAGs and PAGs:

For a given CPDAG or PAG $\mathcal{G}$ and disjoint node sets $\mathbf{X}$ and $\mathbf{Y}$,

(1) Check if $\mathcal{G}$ is amenable relative to $(\mathbf{X}, \mathbf{Y})$. If not, stop because there is no adjustment set relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$. Otherwise, continue.

(2) Find $\mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$.

(3) If $\mathcal{G}$ is a CPDAG, orient $\mathcal{G}$ into a DAG $\mathcal{D}$ in $[\mathcal{G}]$. If $\mathcal{G}$ is a PAG, orient $\mathcal{G}$ into a MAG $\mathcal{M}$ in $[\mathcal{G}]$ according to Theorem 2 from Zhang (2008b) (see Lemma 43 in Appendix A).

(4) By Lemma 10, finding all sets satisfying the generalized adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$ is equivalent to finding all sets $\mathbf{Z}$ satisfying the separation condition relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$ ($\mathcal{M}$) such that $\mathbf{Z} \cap (\mathbf{X} \cup \mathbf{Y} \cup \mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G})) = \emptyset$. Thus, we can apply the algorithms from van der Zander et al. (2014) on $\mathcal{D}$ ($\mathcal{M}$). These algorithms are able to deal with the additional restriction that the resulting set must not contain nodes in $\mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$.

Thus, through this simple procedure we gain complete access to all functions in the algorithmic framework by van der Zander et al. (2014). These include listing all adjustment sets and all minimal adjustment sets (in polynomial time per set that is listed). We have implemented these features in the function `adjustmentSets` of the R package `dagitty`.

5. Relationship to (Generalized) Back-door Criteria

We now discuss the relationship between our generalized adjustment criterion and some other existing graphical criteria for covariate adjustment. In particular, we discuss Pearl's back-door criterion (see Definition 18) and the generalized back-door criterion (see Definition 20) and give constructive sets for both in Section 5.1. We use the results from Section 5.1 to precisely characterize the differences between our generalized adjustment criterion, Pearl's back-door criterion and the generalized back-door criterion in Theorem 26 of Section 5.2. We illustrate the results of Sections 5.1 and 5.2 with examples in Section 5.3.

Definition 18 *(**Back-door criterion**; Pearl, 1993) Let X and Y be distinct nodes in a* DAG *$\mathcal{D}$. A set of nodes $\mathbf{Z}$ not containing X or Y satisfies the back-door criterion relative to (X, Y) in $\mathcal{D}$ if:*

(i) no node in $\mathbf{Z}$ is a descendant of X, and

(ii) $\mathbf{Z}$ blocks every path between X and Y that contains an arrow into X.

If $\mathbf{X}$ and $\mathbf{Y}$ are two disjoint sets of nodes in $\mathcal{D}$, then $\mathbf{Z}$ is said to satisfy the back-door criterion relative to $(\mathbf{X}, \mathbf{Y})$ if it satisfies the criterion relative to any pair (X, Y) such that $X \in \mathbf{X}$, and $Y \in \mathbf{Y}$. A set $\mathbf{Z}$ that satisfies the back-door criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$ is called a back-door set relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$.

Definition 19 *(**Back-door path**; Maathuis and Colombo, 2015) Let X and Y be distinct nodes in a* DAG*,* CPDAG*,* MAG *or* PAG *$\mathcal{G}$. A path from X to Y in $\mathcal{G}$ is a back-door path if it does not start with a visible edge out of X.*

Definition 20 *(**Generalized back-door criterion**; Maathuis and Colombo, 2015) Let* $\mathbf{X}, \mathbf{Y}$ *and* $\mathbf{Z}$ *be pairwise disjoint node sets in a* DAG, CPDAG, MAG *or* PAG $\mathcal{G}$. *Then* $\mathbf{Z}$ *satisfies the generalized back-door criterion relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{G}$ *if:*

(i) $\mathbf{Z}$ *does not contain possible descendants of* $\mathbf{X}$ *in* $\mathcal{G}$, *and*

(ii) for every $X \in \mathbf{X}$, *the set* $\mathbf{Z} \cup \mathbf{X} \setminus \{X\}$ *blocks every definite status back-door path from* X *to any member of* $\mathbf{Y}$, *if any, in* $\mathcal{G}$.

A set $\mathbf{Z}$ *that satisfies the generalized back-door criterion relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{G}$ *is called a generalized back-door set relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{G}$.

5.1 Constructing (Generalized) Back-door Sets

We first focus on Pearl's back-door criterion. If $|\mathbf{X}| = 1$, the existence and construction of a back-door set in a causal DAG $\mathcal{D}$ is well understood. If $Y \in \mathrm{Pa}(X, \mathcal{D})$, then there is no back-door set relative to (X, Y) in $\mathcal{D}$, but it is obvious that $f(y \mid \mathrm{do}(x)) = f(y)$. If $Y \notin \mathrm{Pa}(X, \mathcal{D})$, then $\mathrm{Pa}(X, \mathcal{D})$ is a back-door set relative to (X, Y) in $\mathcal{D}$.

If $|\mathbf{X}| \geq 1$, the construction of a back-door set relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$ is less obvious. One could perhaps think that any set $\mathbf{Z}$ that satisfies our generalized adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$ such that $\mathbf{Z} \cap \mathrm{De}(\mathbf{X}, \mathcal{D}) = \emptyset$ satisfies Pearl's back-door criterion. This is not true, as shown in Lemma 21 that describes the graphical pattern that appears when there is an adjustment set but no back-door set. An example of a DAG that satisfies (i)-(iv) in Lemma 21 is given in Figure 8. Using this result and Theorem 14 we are able to define a specific set that satisfies Pearl's back-door criterion, when such a set exists. This result is given in Corollary 22.

Lemma 21 *Let* $\mathbf{X}$ *and* $\mathbf{Y}$ *be disjoint node sets in a* DAG $\mathcal{D}$. *Assume there is a set satisfying the generalized adjustment criterion* $\mathbf{Z}$ *relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{D}$ *such that* $\mathbf{Z} \cap \mathrm{De}(\mathbf{X}, \mathcal{D}) = \emptyset$. *Then there is no back-door set relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{D}$ *if and only if there is a path* p *from* $\mathbf{X}$ *to* $\mathbf{Y}$ *such that:*

(i) p *is a back-door path, and*

(ii) a proper subpath of p *is a causal path, and*

(iii) there are no colliders on p, *and*

(iv) all nodes on p *are in* $\mathrm{De}(\mathbf{X}, \mathcal{D})$.

Corollary 22 *(**Constructive back-door set**) Let* $\mathbf{X}$ *and* $\mathbf{Y}$ *be disjoint node sets in a* DAG $\mathcal{D}$. *The following statements are equivalent:*

(i) There exists a set that satisfies Pearl's back-door criterion relative to $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{D}$.

(ii) $\mathrm{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{D}) \setminus \mathrm{De}(\mathbf{X}, \mathcal{D})$ *satisfies Pearl's back-door criterion relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{D}$.

(iii) For all $X \in \mathbf{X}$, $Y \in \mathbf{Y}$, $\mathrm{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{D}) \setminus \mathrm{De}(\mathbf{X}, \mathcal{D})$ *satisfies condition (ii) of Pearl's back-door criterion.*

Maathuis and Colombo (2015) presented a constructive generalized back-door set for a DAG, CPDAG, MAG or PAG $\mathcal{G}$ when $|\mathbf{X}| = 1$. In Lemma 23, we show that any set $\mathbf{Z}$ that satisfies our generalized adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$ such that $\mathbf{Z} \cap \mathrm{PossDe}(\mathbf{X}, \mathcal{G}) = \emptyset$ is a generalized back-door set relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$. Using this result and Theorem 14, we give a constructive set that satisfies the generalized back-door criterion, when such a set exists. This set is given in Corollary 24. If $|\mathbf{X}| = 1$, our constructive set for the generalized back-door criterion is a superset of the set presented in Maathuis and Colombo (2015) (Corollary 53 in Appendix D).

Lemma 23 *Let* $\mathbf{X}, \mathbf{Y}$ *and* $\mathbf{Z}$ *be pairwise disjoint node sets in a* DAG, CPDAG, MAG *or* PAG $\mathcal{G}$. *If* $\mathcal{G}$ *is amenable relative to* $(\mathbf{X}, \mathbf{Y})$ *and* $\mathbf{Z}$ *satisfies the blocking condition relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{G}$, *then* $\mathbf{Z}$ *satisfies condition (ii) of the generalized back-door criterion relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{G}$.

Corollary 24 *(**Constructive generalized back-door set**) Let* $\mathbf{X}$ *and* $\mathbf{Y}$ *be disjoint node sets in a* DAG, CPDAG, MAG *or* PAG $\mathcal{G}$. *The following statements are equivalent:*

(i) There exists a set that satisfies the generalized back-door criterion relative to $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{G}$.

(ii) $\mathrm{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \setminus \mathrm{PossDe}(\mathbf{X}, \mathcal{G})$ *satisfies the generalized back-door criterion relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{G}$.

(iii) $\mathcal{G}$ *is amenable and* $\mathrm{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \setminus \mathrm{PossDe}(\mathbf{X}, \mathcal{G})$ *satisfies condition (ii) of the generalized back-door criterion relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{G}$.

5.2 Graphs for Which the Criteria Differ

We now define graphical conditions for the existence of a set satisfying one, two, or all three of the mentioned criteria. The main result of this section is given in Theorem 26, which describes the 4 graphical patterns that can appear when there is no set satisfying at least one of the criteria.

Previously, in Lemma 8 and Lemma 16, we described such patterns for our generalized adjustment criterion. In Section 5.1, we showed that any set $\mathbf{Z}$ that satisfies our generalized adjustment criterion such that $\mathbf{Z} \cap \mathrm{PossDe}(\mathbf{X}, \mathcal{G}) = \emptyset$ satisfies the generalized back-door criterion. Thus, to describe an additional pattern that appears when there is no set that satisfies the generalized back-door criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$ we give Lemma 25. Lastly, to complete Theorem 26 we add the pattern described in Lemma 21 that additionally appears when there is no set that satisfies Pearl's back-door criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$. Thus, Theorem 26 summarizes and subsumes the results of Lemmas 8, 16, 21 and 25.

Lemma 25 *Let* $\mathbf{X}$ *and* $\mathbf{Y}$ *be disjoint node sets in a* DAG, CPDAG, MAG *or* PAG $\mathcal{G}$ *such that there exists an adjustment set relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{G}$. *There is no set that satisfies the generalized back-door criterion relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{G}$ *if and only if there is a path* p *from* $X \in \mathbf{X}$ *to* $Y \in \mathbf{Y}$ *in* $\mathcal{G}$ *and a node* V *on* p *such that:*

(i) p *is a proper definite status non-causal path from* $\mathbf{X}$ *to* $\mathbf{Y}$ *in* $\mathcal{G}$, *and*

(ii) $V \in \text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \cap \text{PossDe}(\mathbf{X}, \mathcal{G})$ *and is a definite non-collider on* p*, and*

(iii) any collider on p *is in* $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \setminus \text{PossDe}(\mathbf{X}, \mathcal{G})$ *and any definite non-collider on* $p(V, Y)$ *is in* $\text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$*, and*

(iv) path p *is of the form* $X \leftarrow \cdots \leftarrow V \leftarrow\!\bullet W \ldots Y$*, where* $W = Y$ *is possible and if* $W \neq Y$ *then* $V \leftrightarrow W$ *is on* p*.*

Theorem 26 *Let* $\mathbf{X}$ *and* $\mathbf{Y}$ *be disjoint node sets in a* DAG*,* CPDAG*,* MAG *or* PAG $\mathcal{G}$*. Consider the following criteria:*

(1) $\mathcal{G}$ *violates the amenability condition relative to* $(\mathbf{X}, \mathbf{Y})$*.*

(2) There is a proper definite status non-causal path p *from* $\mathbf{X}$ *to* $\mathbf{Y}$ *in* $\mathcal{G}$ *such that every collider on* p *is in* $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$ *and every definite non-collider on* p *is in* $\text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$*.*

(3) There is a path p *from* $\mathbf{X}$ *to* $\mathbf{Y}$ *in* $\mathcal{G}$ *that satisfies (i)–(iv) in Lemma 25.*

(4) There is a back-door path p *from* $\mathbf{X}$ *to* $\mathbf{Y}$ *in* $\mathcal{G}$ *that satisfies (i)–(iv) in Lemma 21.*

The following hold:

(i) There is no set that satisfies the generalized adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{G}$ *if and only if (1) or (2) are satisfied.*

(ii) There is no generalized back-door set relative to $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{G}$ *if and only if (1), (2) or (3) are satisfied.*

(iii) If $\mathcal{G}$ *is a* DAG*, then there is no back-door set relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{G}$ *if and only if (1), (2), (3) or (4) are satisfied.*

We now further explore condition (2) in Theorem 26 under the assumption that the DAG, CPDAG, MAG or PAG $\mathcal{G}$ is amenable relative to disjoint node sets $(\mathbf{X}, \mathbf{Y})$ (that is, (1) in Theorem 26 is violated). Condition (2) in Theorem 26 is satisfied relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$ if and only if there is no adjustment set relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$ (Theorem 5). Corollary 27 provides a simple sufficient condition for condition (2) in Theorem 26 to be satisfied in DAGs, CPDAGs, MAGs and PAGs, as well as a necessary and sufficient condition for condition (2) in Theorem 26 in certain DAGs and CPDAGs.

Corollary 27 *Let* $\mathbf{X}$ *and* $\mathbf{Y}$ *be disjoint node sets in a DAG, CPDAG, MAG or PAG* $\mathcal{G}$ *such that* $\mathcal{G}$ *is amenable relative to* $(\mathbf{X}, \mathbf{Y})$*. The following statements hold:*

(i) If $\mathbf{X} \cap \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \neq \emptyset$*, then there is no adjustment set relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{G}$*.*

(ii) Let $\mathcal{G}$ *be a DAG or CPDAG and* $\mathbf{Y} \subseteq \text{PossDe}(\mathbf{X}, \mathcal{G})$*. Then* $\mathbf{X} \cap \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \neq \emptyset$ *if and only if there is no adjustment set relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{G}$*.*

A necessary condition for both (3) and (4) in Theorem 26 is that $\mathcal{G}$ contains a (possibly) directed path from one node in $\mathbf{X}$ to another node in $\mathbf{X}$. Thus, if $|\mathbf{X}| = 1$, both (3) and (4) in Theorem 26 are violated. Hence, if $|\mathbf{X}| = 1$ there is a (generalized) back-door set relative to $(\mathbf{X}, \mathbf{Y})$ in a DAG (or a CPDAG, MAG or PAG) $\mathcal{G}$, if and only if there is a set satisfying the generalized adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$.

We finish this section by giving two corollaries that describe some additional simple conditions under which there exists a set satisfying two or all three of the discussed adjustment criteria. The intuition behind these results is as follows. A necessary condition for the existence of a path p from $\mathbf{X}$ to $\mathbf{Y}$ in a DAG $\mathcal{D}$ that satisfies Lemma 21 is the existence of a causal path from one node in $\mathbf{X}$ to another node in $\mathbf{X}$ in $\mathcal{D}$. This gives us Corollary 28. Similarly, a necessary condition for the existence of a path p from $\mathbf{X}$ to $\mathbf{Y}$ in a DAG, CPDAG, MAG or PAG $\mathcal{G}$ that satisfies Lemma 25 is the existence of a causal path from one node in $\mathbf{X}$ to another node in $\mathbf{X}$ that contains at least one node not in $\mathbf{X}$ in $\mathcal{G}$. Or, in the case when $\mathcal{G}$ is a DAG or CPDAG, another necessary condition for the existence of a path p from $\mathbf{X}$ to $\mathbf{Y}$ that satisfies Lemma 25 in $\mathcal{G}$ is that $\mathbf{Y} \not\subseteq \text{PossDe}(\mathbf{X}, \mathcal{G})$. This gives us Corollary 29.

Corollary 28 *Let* $\mathbf{X}$ *and* $\mathbf{Y}$ *be disjoint node sets in a* DAG $\mathcal{D}$*. If there is no directed path from one node in* $\mathbf{X}$ *to another node in* $\mathbf{X}$ *in* $\mathcal{D}$*, then the following statements are equivalent:*

(i) There exists a set that satisfies the generalized adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{D}$*.*

(ii) There exists a back-door set relative to $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{D}$*.*

Corollary 29 *Let* $\mathbf{X}$ *and* $\mathbf{Y}$ *be disjoint node sets in a* DAG*,* CPDAG*,* MAG *or* PAG $\mathcal{G}$*. If* $\mathcal{G}$ *contains no possibly directed path* $p = \langle V_1, \ldots, V_k \rangle$ *with* $k \geq 3$ *such that* $\{V_1, V_k\} \subseteq \mathbf{X}$ *and* $\{V_2, \ldots, V_{k-1}\} \cap \mathbf{X} = \emptyset$*, or if* $\mathcal{G}$ *is a* DAG *or* CPDAG *and* $\mathbf{Y} \subseteq \text{PossDe}(\mathbf{X}, \mathcal{G})$*, then the following statements are equivalent:*

(i) There exists a set that satisfies the generalized adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{G}$*.*

(ii) There exists a generalized back-door set relative to $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{G}$*.*

5.3 Examples

Figure 3 (see Example 3) in Section 3.1 shows a non-amenable graph (that is, condition (1) in Theorem 26 is satisfied). Figure 4(b) (see Example 4) in Section 3.1 shows an amenable graph for which there is no set that satisfies the generalized adjustment criterion (that is, condition (1) in Theorem 26 is violated, but condition (2) is satisfied).

We now give three additional examples. Figure 6(b) (see Example 7) shows an amenable graph for which there is no set that satisfies the generalized adjustment criterion (that is, (2) in Theorem 26 is satisfied). This example illustrates the result of Corollary 27. Figure 6(a) (see Example 7) and Figure 7 (see Example 8) show cases where there is a set that satisfies the generalized adjustment criterion, but there is no generalized back-door set (that is, conditions (1) and (2) in Theorem 26 are violated, but condition (3) is satisfied).

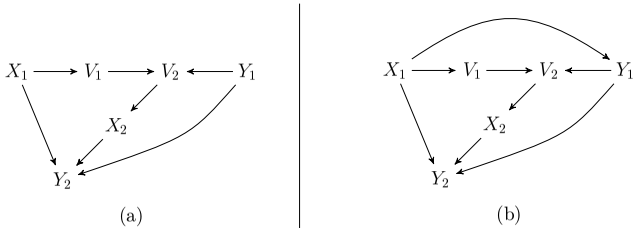


Figure 6: (a) DAG $\mathcal{D}_1$, (b) DAG $\mathcal{D}_2$ used in Example 7.

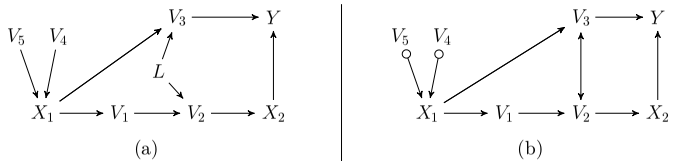


Figure 7: (a) DAG $\mathcal{D}$, (b) PAG $\mathcal{P}$ used in Example 8.

Figure 8 (see Example 9) shows an example of a DAG in which there are sets that satisfy the generalized adjustment criterion and the generalized back-door criterion, but no set satisfies Pearl's back-door criterion (that is, conditions (1), (2) and (3) in Theorem 26 are violated, but condition (4) is satisfied).

Example 7 *Let* $\mathbf{X} = \{X_1, X_2\}$ *and* $\mathbf{Y} = \{Y_1, Y_2\}$ *and consider the* DAG*s* $\mathcal{D}_1$ *and* $\mathcal{D}_2$ *in Figure 6(a) and 6(b) respectively. We first consider* DAG $\mathcal{D}_1$*. The proper non-causal path* $X_2 \leftarrow V_2 \leftarrow Y_1$ *satisfies (i)–(iv) in Lemma 25. Hence, there is no generalized back-door set relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{D}_1$*. However,* $\{V_1, V_2\}$ *satisfies the generalized adjustment criterion relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{D}_1$*.*

We now consider DAG $\mathcal{D}_2$*. Note that the only difference between* $\mathcal{D}_1$ *and* $\mathcal{D}_2$ *is the additional edge* $X_1 \to Y_1$ *in* $\mathcal{D}_2$*. This edge implies that* $Y_1 \in \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D}_2)$*. Hence, the proper non-causal path* $X_2 \leftarrow V_2 \leftarrow Y_1$ *satisfies (2) in Theorem 26 and thus, there is no set that satisfies the generalized adjustment criterion relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{D}_2$*. Then by Theorem 5 there is no adjustment set relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{D}_2$*. Since* $\mathcal{D}_2$ *is a DAG such that* $\mathbf{Y} \subseteq \text{De}(\mathbf{X}, \mathcal{D}_2)$*, (ii) in Corollary 27 implies that* $\mathbf{X} \cap \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D}_2) \neq \emptyset$*. This is indeed true, since* $\text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D}_2) = \{X_2, V_2, Y_1, Y_2\}$*.*

Example 8 *Let* $\mathbf{X} = \{X_1, X_2\}$ *and* $\mathbf{Y} = \{Y\}$ *and consider* DAG $\mathcal{D}$ *and* PAG $\mathcal{P}$ *in Figures 7(a) and 7(b). We first consider* DAG $\mathcal{D}$*. Any generalized back-door set relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{D}$ *must contain* L*. However, the same is not true for the generalized adjustment criterion. For example,* $\{V_1, V_2\}$ *satisfies the generalized adjustment criterion relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{D}$*.*

We now consider $\mathcal{P}$*. Note that* $\mathcal{P}$ *is the PAG of* $\mathcal{D}$ *when* L *is unobserved. The proper non-causal path* $X_2 \leftarrow V_2 \leftrightarrow V_3 \to Y$ *satisfies (i)–(iv) in Lemma 25. Hence, there is no*

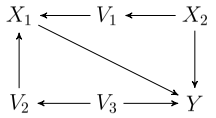


Figure 8: DAG $\mathcal{D}$ used in Example 9.

generalized back-door set relative to $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{P}$*. However, the sets* $\{V_1, V_2\}$*,* $\{V_1, V_2, V_4\}$*,* $\{V_1, V_2, V_5\}$*,* $\{V_1, V_2, V_4, V_5\}$ *all satisfy the generalized adjustment criterion relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{P}$*.*

Example 9 *Let* $\mathbf{X} = \{X_1, X_2\}$ *and* $\mathbf{Y} = \{Y\}$ *and consider* DAG $\mathcal{D}$ *in Figure 8. The non-causal path* $X_1 \leftarrow V_1 \to X_2 \to Y$ *satisfies (i)–(iv) in Lemma 21. Hence, no set can satisfy the back-door criterion relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{D}$*. However,* $\{V_2\}$*,* $\{V_3\}$*,* $\{V_2, V_3\}$*,* $\{V_1, V_2\}$*,* $\{V_1, V_3\}$*, and* $\{V_1, V_2, V_3\}$ *all satisfy the generalized adjustment criterion and* $\{V_2\}$*,* $\{V_3\}$ *and* $\{V_2, V_3\}$ *all satisfy the generalized back-door criterion relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{D}$*.*

6. Discussion

We have derived a generalized adjustment criterion that is sound and complete for adjustment in DAGs, MAGs, CPDAGs and PAGs (see Definition 4, Theorem 5). This is relevant in practice, in particular in combination with algorithms that can learn CPDAGs or PAGs from observational data.

In addition to the criterion itself, we have also given all necessary ingredients for implementing efficient algorithms to test the criterion for a given set and to construct all sets that fulfill it, or to learn that no set fulfilling the criterion exists. Thus, we obtain a complete generalization of the algorithmic framework for DAGs and MAGs by van der Zander et al. (2014) to CPDAGs or PAGs. In this sense, our work presented in this paper is a theoretical contribution that closes the chapter on covariate adjustment for DAGs, CPDAGs, MAGs and PAGs without selection variables.

Correa and Bareinboim (2017) define necessary and sufficient graphical conditions for covariate adjustment in latent projection graphs in the presence of selection variables. Future work may explore how to extend their results to MAGs and PAGs with selection variables. Other future work may study the estimation accuracy of estimators based on different adjustment sets. Any valid adjustment set can be used to produce an unbiased estimator of the total causal effect. However, the efficiency of the estimators induced by distinct adjustment sets varies (Greenland et al., 1999; Kuroki and Miyakawa, 2003; Kuroki and Cai, 2004; Hahn, 2004, 1998; Guo and Dawid, 2010; De Luna et al., 2011). Moreover, Kuroki and Miyakawa (2003) and Kuroki and Cai (2004) indicate that a minimal adjustment set does not necessarily lead to the most efficient estimator. Defining a best adjustment set in terms of efficiency is still an open question.

Acknowledgments

This work was supported in part by Swiss NSF Grants 200021_149760 and 200021_172603.

Appendix A. Preliminaries

We first state various existing results and definitions.

Adjacencies, discriminating paths and d-separations. The set of all nodes adjacent to X in a graph $\mathcal{G}$ is denoted by $\mathrm{Adj}(X, \mathcal{G})$. A path $p = \langle X, \dots, Z, V, Y \rangle$ is a *discriminating path* from X to Y for V in graph $\mathcal{G}$, if it consists of at least four nodes, X is not adjacent to Y in $\mathcal{G}$ and every non-endpoint node on $p(X, V)$ is a collider on p and a parent of Y. If $\mathbf{X}$ and $\mathbf{Y}$ are d-separated given $\mathbf{Z}$ in a DAG $\mathcal{D}$, we write $\mathbf{X} \perp_d \mathbf{Y} \mid \mathbf{Z}$.

Definition 30 ***(Distance-from-*$\mathbf{Z}$*)*** *Let $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$ be pairwise disjoint node sets in a* DAG*,* CPDAG*,* MAG *or* PAG *$\mathcal{G}$. Let p be a path between $\mathbf{X}$ and $\mathbf{Y}$ in $\mathcal{G}$ such that every collider C on p has a possibly directed path (possibly of length 0) to $\mathbf{Z}$. Define the* distance-from-$\mathbf{Z}$ *of C to be the length of a shortest possibly directed path (possibly of length 0) from C to $\mathbf{Z}$, and define the* distance-from-$\mathbf{Z}$ *of p to be the sum of the distances from $\mathbf{Z}$ of the colliders on p.*

If $\mathcal{G}$ is a MAG and p is a path from $\mathbf{X}$ to $\mathbf{Y}$ that is m-connecting given $\mathbf{Z}$ in $\mathcal{G}$, then Definition 30 reduces to the notion of *distance-from-*$\mathbf{Z}$ in Zhang (2006, p213).

Theorem 31 ***(Wright's rule** cf. Wright, 1921) Let $\mathbf{X} = \mathbf{A}\mathbf{X} + \epsilon$, where $\mathbf{A} \in \mathbb{R}^{k \times k}$, $\mathbf{X} = (X_1, \dots, X_k)^T$ and $\epsilon = (\epsilon_1, \dots, \epsilon_k)^T$ is a vector of mutually independent errors with means zero. Moreover, let $\mathrm{Var}(\mathbf{X}) = \mathbf{I}$. Let $\mathcal{D} = (\mathbf{X}, \mathbf{E})$, be the corresponding* DAG *such that $X_i \to X_j$ in $\mathcal{D}$ if and only if $A_{ji} \neq 0$. A nonzero entry A_{ji} is called the edge coefficient of $X_i \to X_j$. For two distinct nodes $X_i, X_j \in \mathbf{X}$, let $p_1, \dots, p_r$ be all paths between X_i and X_j in $\mathcal{D}$ that do not contain a collider. Then $\mathrm{Cov}(X_i, X_j) = \sum_{s=1}^{r} \pi_s$, where π_s is the product of all edge coefficients along path p_s, $s \in \{1, \dots, r\}$.*

Theorem 32 *(cf. Theorem 3.2.4 Mardia et al., 1980, p63) Let $\mathbf{X} = (\mathbf{X_1}^T, \mathbf{X_2}^T)^T$ be a p-dimensional multivariate Gaussian random vector with mean vector $\mu = (\mu_\mathbf{1}^T, \mu_\mathbf{2}^T)^T$ and covariance matrix $\mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma_{11}} & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma_{21}} & \mathbf{\Sigma}_{22} \end{bmatrix}$, so that $\mathbf{X_1}$ is a q-dimensional multivariate Gaussian random vector with mean vector $\mu_\mathbf{1}$ and covariance matrix $\mathbf{\Sigma}_{11}$ and $\mathbf{X_2}$ is a $(p-q)$-dimensional multivariate Gaussian random vector with mean vector $\mu_\mathbf{2}$ and covariance matrix $\mathbf{\Sigma}_{22}$. Then $E[\mathbf{X_2} \mid \mathbf{X_1} = \mathbf{x_1}] = \mu_\mathbf{2} + \mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1}(\mathbf{x_1} - \mu_\mathbf{1})$.*

Definition 33 ***(Moralization**; Lauritzen and Spiegelhalter, 1988) Let $\mathcal{D}$ be a* DAG*. The moral graph $\mathcal{D}^m$ is formed by adding the edge $A - B$ to any structure of the form $A \to C \leftarrow B$ with $A \notin \mathrm{Adj}(B, \mathcal{D})$ (marrying unmarried parents) and subsequently making all edges in the resulting graph undirected.*

Definition 34 ***(Induced subgraph)*** *Let $\mathbf{X} \subseteq \mathbf{V}$ be a node set in a* DAG *$\mathcal{D} = (\mathbf{V}, \mathbf{E})$. Then $\mathcal{D}_\mathbf{X} = (\mathbf{X}, \mathbf{E_X})$, where $\mathbf{E_X}$ consists of all edges in $\mathbf{E}$ for which both endpoints are in $\mathbf{X}$, is the induced subgraph of $\mathcal{D}$ on $\mathbf{X}$.*

Theorem 35 ***(Reduction of d-separation to node cuts**; cf. Proposition 3 in Lauritzen et al., 1990, cf. Corollary 2 in Richardson, 2003) Let $\mathcal{D}$ be a* DAG *and let $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$ be pairwise disjoint node sets in $\mathcal{D}$. Then $\mathbf{Z}$ d-separates $\mathbf{X}$ and $\mathbf{Y}$ in $\mathcal{D}$ if and only if all paths between $\mathbf{X}$ and $\mathbf{Y}$ in $(\mathcal{D}_{\mathrm{An}(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}, \mathcal{D})})^m$ contain at least one node in $\mathbf{Z}$.*

Lemma 36 ***(Basic property of** CPDAG**s and** PAG**s**; cf. Lemma 1 in Meek, 1995, cf. Lemma 3.3.1 in Zhang, 2006) Let X, Y and Z be distinct nodes in a* CPDAG *or* PAG *$\mathcal{G}$. If $X \bullet\!\!\to Y \circ\!\!-\!\!\bullet Z$, then there is an edge between X and Z with an arrowhead at Z. Furthermore, if the edge between X and Y is $X \to Y$, then the edge between X and Z is either $X \circ\!\!\to Z$ or $X \to Z$ (that is, not $X \leftrightarrow Z$).*

Lemma 37 *(cf. Lemma 1 in Richardson, 2003) Let $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$ be pairwise disjoint node sets in a* DAG *or* MAG *$\mathcal{G}$. If there is a path p from $X \in \mathbf{X}$ to $Y \in \mathbf{Y}$, on which no non-collider is in $\mathbf{Z}$ and every collider on p is in $\mathrm{An}(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}, \mathcal{G})$, then there exists a path q from $X' \in \mathbf{X}$ to $Y' \in \mathbf{Y}$ that is m-connecting given $\mathbf{Z}$ in $\mathcal{G}$.*

Lemma 38 *(Lemma 0 in Zhang, 2006, p208) Let X and Y be distinct nodes in a* MAG *$\mathcal{M}$. If $p = \langle X, \dots, Z, V, Y \rangle$ is a discriminating path from X to Y for V in a* MAG *$\mathcal{M}$, and the corresponding path to $p(X, V)$ in the* PAG *$\mathcal{P}$ of $\mathcal{M}$ is (also) a collider path, then the corresponding path to p in $\mathcal{P}$ is also a discriminating path from X to Y for V.*

Lemma 39 *(cf. Lemma 1 in Zhang, 2006, p208) Let X and Y be distinct nodes and let $\mathbf{Z}$ be a node set that does not contain X and Y in a* MAG *$\mathcal{M}$ (*DAG *$\mathcal{D}$). Let p be a shortest path from X to Y that is m-connecting given $\mathbf{Z}$ in $\mathcal{M}$ ($\mathcal{D}$). Let $\mathcal{P}$ be the* PAG *of $\mathcal{M}$ (*CPDAG *of $\mathcal{D}$) and let p^* in $\mathcal{P}$ be the corresponding path to p in $\mathcal{M}$ ($\mathcal{D}$). Then p^* is a definite status path in $\mathcal{P}$.*

Lemma 40 *(cf. Lemma 2 in Zhang, 2006, p213) Let X and Y be distinct nodes and let $\mathbf{Z}$ be a node set that does not contain X and Y in a* MAG *$\mathcal{M}$ (*DAG *$\mathcal{D}$). Let p be a shortest path from X to Y that is m-connecting given $\mathbf{Z}$ in $\mathcal{M}$ ($\mathcal{D}$) such that no equally short m-connecting path has a shorter distance-from-$\mathbf{Z}$ (see Definition 30) than p does. Let $\mathcal{P}$ be the* PAG *of $\mathcal{M}$ (*CPDAG *of $\mathcal{D}$) and let p^* in $\mathcal{P}$ be the corresponding path to p in $\mathcal{M}$ ($\mathcal{D}$). Then p^* is a definite status path from X to Y that is m-connecting given $\mathbf{Z}$ in $\mathcal{P}$.*

Lemma 41 *(cf. Lemma B.1 in Zhang, 2008b) Let X and Y be distinct nodes in a* CPDAG *or* PAG *$\mathcal{G}$. If $p = \langle X, \dots, Y \rangle$ is a possibly directed path from X to Y in $\mathcal{G}$, then some subsequence of p forms an unshielded possibly directed path from X to Y in $\mathcal{G}$.*

Lemma 42 *(cf. Lemma B.2 in Zhang, 2008b, Lemma 7.2 in Maathuis and Colombo, 2015) Let X and Y be distinct nodes in a* CPDAG *or* PAG *$\mathcal{G}$. If $p = \langle X = V_0, \dots, V_k = Y \rangle$, $k \geq 2$ is an unshielded possibly directed path from X to Y in $\mathcal{G}$, and $V_{i-1} \bullet\!\!\to V_i$ for some $i \in \{1, \dots, k\}$, then $V_{j-1} \to V_j$ for all $j \in \{i+1, \dots, k\}$.*

Lemma 43 *(cf. Theorem 2 in Zhang, 2008b, Lemma 7.6 in Maathuis and Colombo, 2015) Let $\mathcal{G}$ be a* PAG (CPDAG)*. Let $\mathcal{M}$ ($\mathcal{D}$) be the graph resulting from the following procedure applied to a $\mathcal{G}$:*

(1) replace all partially directed edges $\circ\!\!\rightarrow$ in $\mathcal{G}$ with directed edges $\rightarrow$, and

(2) orient the subgraph of $\mathcal{G}$ consisting of all non-directed edges $\circ\!\!-\!\!\circ$ into a DAG *with no unshielded colliders.*

Then $\mathcal{M}$ ($\mathcal{D}$) is in $[\mathcal{G}]$. Moreover, if X is a node in $\mathcal{G}$, then one can always find an orientation of (2) that does not create any new edges into X.

Lemma 44 *(Lemma 7.5 in Maathuis and Colombo, 2015) Let X and Y be two distinct nodes in a* DAG, CPDAG, MAG *or* PAG *$\mathcal{G}$. Then $\mathcal{G}$ cannot have both a possibly directed path from X to Y and an edge of the form $Y \bullet\!\!\rightarrow X$.*

Definition 45 *($DSEP(X,Y,\mathcal{G})$; Spirtes et al., 2000, p136) Let X and Y be two distinct nodes in a* DAG *or* MAG *$\mathcal{G}$. We say that $V \in DSEP(X,Y,\mathcal{G})$ if $V \neq X$ and there is a collider path between X and V in $\mathcal{G}$ such that every node on this path is an ancestor of X or Y in $\mathcal{G}$.*

Definition 46 *($\mathcal{R}$ and $\mathcal{R}_{\underline{X}}$; Maathuis and Colombo, 2015) Let X be a node in a* DAG, CPDAG, MAG *or* PAG *$\mathcal{G}$. Let $\mathcal{R}$ be a* DAG *or* MAG *represented by $\mathcal{G}$, in the following sense. If $\mathcal{G}$ is a* DAG *or* MAG*, let $\mathcal{R} = \mathcal{G}$. If $\mathcal{G}$ is a* CPDAG (PAG)*, let $\mathcal{R}$ be a* DAG (MAG) *in $[\mathcal{G}]$ as defined in Lemma 43, so that $\mathcal{R}$ has the same number of edges into X as $\mathcal{G}$. Let $\mathcal{R}_{\underline{X}}$ be the graph obtained from $\mathcal{R}$ by removing all directed edges out of X that are visible in $\mathcal{G}$.*

Theorem 47 *(Theorem 4.1 in Maathuis and Colombo, 2015) Let X and Y be distinct nodes in a* DAG, CPDAG, MAG *or* PAG *$\mathcal{G}$. Let $\mathcal{R}$ and $\mathcal{R}_{\underline{X}}$ be defined as in Definition 46. Then there exists a generalized back-door set relative to (X,Y) in $\mathcal{G}$ if and only if $Y \notin \mathrm{Adj}(X, \mathcal{R}_{\underline{X}})$ and $DSEP(X,Y,\mathcal{R}_{\underline{X}}) \cap \mathrm{PossDe}(X,\mathcal{G}) = \emptyset$. Moreover, if such a set exists, then $DSEP(X,Y,\mathcal{R}_{\underline{X}})$ is a generalized back-door set relative to (X,Y) in $\mathcal{G}$.*

A.1 Rules of the Do-calculus (Pearl, 2009, Chapter 3.4)

Let $\mathbf{X}', \mathbf{Y}', \mathbf{Z}', \mathbf{W}'$ be pairwise disjoint (possibly empty) sets of nodes in a causal DAG $\mathcal{D}$. Let $\mathcal{D}_{\overline{\mathbf{X}'}}$ denote the graph obtained by deleting all edges into $\mathbf{X}'$ from $\mathcal{D}$. Similarly, let $\mathcal{D}_{\underline{\mathbf{X}'}}$ denote the graph obtained by deleting all edges out of $\mathbf{X}'$ in $\mathcal{D}$ and let $\mathcal{D}_{\overline{\mathbf{X}'}\underline{\mathbf{Z}'}}$ denote the graph obtained by deleting all edges into $\mathbf{X}'$ and all edges out of $\mathbf{Z}'$ in $\mathcal{D}$. Then the following three rules are valid for every density function consistent with $\mathcal{D}$.

Rule 1 (Insertion/deletion of observations) If $\mathbf{Y}' \perp_d \mathbf{Z}' \mid \mathbf{X}' \cup \mathbf{W}'$ in $\mathcal{D}_{\overline{\mathbf{X}'}}$, then

$$f(\mathbf{y}' \mid \mathrm{do}(\mathbf{x}'), \mathbf{w}') = f(\mathbf{y}' \mid \mathrm{do}(\mathbf{x}'), \mathbf{z}', \mathbf{w}'). \tag{4}$$

Rule 2 (Action/observation exchange) If $\mathbf{Y}' \perp_d \mathbf{Z}' \mid \mathbf{X}' \cup \mathbf{W}'$ in $\mathcal{D}_{\overline{\mathbf{X}'}\underline{\mathbf{Z}'}}$, then

$$f(\mathbf{y}' \mid \mathrm{do}(\mathbf{x}'), \mathrm{do}(\mathbf{z}'), \mathbf{w}') = f(\mathbf{y}' \mid \mathrm{do}(\mathbf{x}'), \mathbf{z}', \mathbf{w}'). \tag{5}$$

Rule 3 (Insertion/deletion of actions)If $\mathbf{Y}' \perp_d \mathbf{Z}' \mid \mathbf{X}' \cup \mathbf{W}'$ in $\mathcal{D}_{\overline{\mathbf{X}' \cup \mathbf{Z}'(\mathbf{W}')}}$, then

$$f(\mathbf{y}' \mid \mathrm{do}(\mathbf{x}'), \mathbf{w}') = f(\mathbf{y}' \mid \mathrm{do}(\mathbf{x}'), \mathrm{do}(\mathbf{z}'), \mathbf{w}'), \tag{6}$$

where $\mathbf{Z}'(\mathbf{W}')$ denotes the set of $\mathbf{Z}'$-nodes that are not ancestors of any $\mathbf{W}'$ node in $\mathcal{D}_{\overline{\mathbf{X}'}}$. If $\mathbf{W}' = \emptyset$, then $\mathbf{Z}'(\mathbf{W}') = \mathbf{Z}'$.

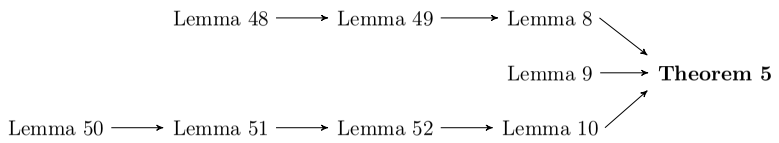


Figure 9: Proof structure of Theorem 5.

A.2 FCI Orientation Rules (Spirtes et al., 2000, p183)

Let A, B, C and D be distinct nodes in a PAG $\mathcal{P}$. Below, we give the first 4 orientation rules of the FCI algorithm defined in Spirtes et al. (2000).
R1 If $A \bullet\!\!\rightarrow B \circ\!\!-\!\!\bullet C$, and A and C are not adjacent, then orient the triple $\langle A, B, C\rangle$ as $A \bullet\!\!\rightarrow B \rightarrow C$.
R2 If $A \rightarrow B \bullet\!\!\rightarrow C$ or $A \bullet\!\!\rightarrow B \rightarrow C$ and $A \bullet\!\!-\!\!\circ C$, then orient $A \bullet\!\!-\!\!\circ C$ as $A \bullet\!\!\rightarrow C$.
R3 If $A \bullet\!\!\rightarrow B \leftarrow\!\!\bullet C$, $A \bullet\!\!-\!\!\circ D \circ\!\!-\!\!\bullet C$, A and C are not adjacent, and $D \bullet\!\!-\!\!\circ B$, then orient $D \bullet\!\!-\!\!\circ B$ as $D \bullet\!\!\rightarrow B$.
R4 If $\langle D, \ldots, A, B, C\rangle$ is a discriminating path from D to C for B and $B \circ\!\!-\!\!\bullet C$, then orient $B \circ\!\!-\!\!\bullet C$ as $B \rightarrow C$ if B is in the separation set of D and C; otherwise orient the triple $\langle A, B, C\rangle$ as $A \leftrightarrow B \leftrightarrow C$.
These four rules were proven to be sound in Spirtes et al. (2000), meaning that edge marks oriented by these rules correspond to invariant edge marks in the maximally informative PAG for the true causal MAG. Six additional orientation rules for the FCI algorithm were defined in Zhang (2008b). The augmented FCI algorithm, including all ten orientation rules was proven to be sound and complete in Zhang (2008b).

Appendix B. Proofs for Section 3

Figure 9 shows how all lemmas in this section fit together to prove Theorem 5.

Lemma 48 *Let X be a node in a* PAG *$\mathcal{P}$. Let $\mathcal{M}$ be a* MAG *$\mathcal{M}$ in $[\mathcal{P}]$ that satisfies Lemma 43. Then any edge that is either $X \circ\!\!-\!\!\circ Y$, $X \circ\!\!\rightarrow Y$ or invisible $X \rightarrow Y$ in $\mathcal{P}$ is invisible $X \rightarrow Y$ in $\mathcal{M}$.*

Proof of Lemma 48. Let $\mathcal{M}$ be a MAG in $[\mathcal{P}]$ that satisfies Lemma 43. Then the edge $X \circ\!\!-\!\!\circ Y$, $X \circ\!\!\rightarrow Y$, or invisible $X \rightarrow Y$ in $\mathcal{P}$ corresponds to edge $X \rightarrow Y$ in $\mathcal{M}$. It is left to prove that $X \rightarrow Y$ is invisible in $\mathcal{M}$ in all these cases.

Suppose for a contradiction that $X \rightarrow Y$ is visible in $\mathcal{M}$. Then there is a node $D \notin \mathrm{Adj}(Y, \mathcal{M})$ such that (1) $D \bullet\!\!\rightarrow X$ is in $\mathcal{M}$, or (2) there is a collider path $\langle D, D_1, \ldots, D_k, X\rangle$, $k \geq 1$, into X such that every $D_i, 1 \leq i \leq k$ is a parent of X in $\mathcal{M}$. We consider these cases separately and show that we arrive at a contradiction.

(1) Since $D \bullet\!\!\rightarrow X$ is in $\mathcal{M}$, the choice of $\mathcal{M}$ implies that $D \bullet\!\!\rightarrow X$ is in $\mathcal{P}$. Since $D \notin \mathrm{Adj}(Y, \mathcal{P})$, $X \rightarrow Y$ must be in $\mathcal{P}$, since otherwise rule **R1** of the FCI algorithm from Spirtes et al. (2000) (see Appendix A) would have been applied in $\mathcal{P}$. But then $X \rightarrow Y$ is visible in $\mathcal{P}$, which is a contradiction.

(2) Path $p = \langle D, D_1, \dots D_k, X, Y\rangle$ is a discriminating path from D to Y for X, that is into X in $\mathcal{M}$. Let p^* be the path in $\mathcal{P}$ corresponding to p in $\mathcal{M}$. Then since $p(D_1, X)$ contains only bi-directed edges in $\mathcal{M}$, the choice of $\mathcal{M}$ implies that $p^*(D_1, X)$ also contains only bi-directed edges in $\mathcal{P}$. Since $D{\bullet\!\!\rightarrow}D_1$ is in $\mathcal{M}$, $D{\circ\!\!-\!\!\circ}D_1$ or $D{\bullet\!\!\rightarrow}D_1$ is in $\mathcal{P}$.

Suppose first that $D{\bullet\!\!\rightarrow}D_1$ is in $\mathcal{P}$, then by Lemma 0 from Zhang (2006) (see Lemma 38), p^* is a discriminating path from D to Y for X, that is into X in $\mathcal{P}$. Hence, $X \to Y$ is in $\mathcal{P}$, since otherwise rule **R4** in Appendix A would have been applied. But then $X \to Y$ is a visible edge in $\mathcal{P}$, which is a contradiction.

Next, suppose that $D{\circ\!\!-\!\!\circ}D_1$ is in $\mathcal{P}$. Since $D_1 \leftrightarrow D_2$ is in $\mathcal{P}$, by Lemma 36 $D{\leftarrow\!\!\bullet}D_2$ is in $\mathcal{P}$. This edge cannot be $D{\leftarrow\!\!\circ}D_2$ or $D \leftarrow D_2$, otherwise $D{\leftarrow\!\!\circ}D_1$ would be in $\mathcal{P}$ (Lemma 36, or **R2** of the FCI orientation rules in Appendix A), contrary to our assumption. Hence, the edge $D \leftrightarrow D_2$ is in $\mathcal{P}$. Then $D \leftrightarrow D_2$ is also in $\mathcal{M}$ and $p_1 = \langle D, D_2, \dots D_k, X, Y\rangle$ is a discriminating path from D to Y for X, that is into X in $\mathcal{M}$. Additionally, since $D \leftrightarrow D_2 \leftrightarrow \cdots \leftrightarrow D_k \leftrightarrow X$ is in $\mathcal{P}$, by Lemma 38 the path ${p_1}^* = \langle D, D_2, \dots D_k, X, Y\rangle$ is a discriminating path from D to Y for X, that is into X. But then $X \to Y$ is a visible edge in $\mathcal{P}$, which is a contradiction. ■

Lemma 49 *Let X and Y be distinct nodes in a* PAG $\mathcal{P}$ *such that there is a possibly directed path p^* from X to Y in $\mathcal{P}$ that does not start with a visible edge out of X. Then there is a* MAG $\mathcal{M}$ *in $[\mathcal{P}]$ such that the path p in $\mathcal{M}$, consisting of the same sequence of nodes as p^* in $\mathcal{P}$, contains a subsequence p' that is a directed path from X to Y starting with an invisible edge in $\mathcal{M}$. In other words, $\mathcal{M}$ violates the amenability condition relative to (X, Y).*

Proof of Lemma 49. If $\mathcal{P}$ violates the amenability condition relative to (X, Y), then there is a possibly directed path p^* from X to Y in $\mathcal{P}$ that does not start with a visible edge out of X. Let $\mathcal{M}$ be a MAG in $[\mathcal{P}]$ that satisfies Lemma 43. We will show that $\mathcal{M}$ violates the amenability condition relative to (X, Y).

Let ${p'}^*$ be a shortest subsequence of p^* such that ${p'}^*$ is also a possibly directed path from X to Y in $\mathcal{P}$ that does not start with a visible edge out of X. We write ${p'}^* = \langle X = V_0, V_1, \dots, V_r = Y\rangle, r \geq 1$. Let p' in $\mathcal{M}$ be the path corresponding to ${p'}^*$ in $\mathcal{P}$. Then by Lemma 48, $X \to V_1$ is an invisible edge in $\mathcal{M}$. It is left to show that p' is a directed path from X to Y in $\mathcal{M}$.

Suppose first that ${p'}^*$ is a definite status path in $\mathcal{P}$. Then all non-endpoint nodes on ${p'}^*$ are definite non-colliders. Hence, $X \to V_1$ in $\mathcal{M}$ implies that all the remaining edges on p' are oriented towards Y.

Else, ${p'}^*$ is not of definite status in $\mathcal{P}$ and $r \geq 2$. Since $X \to V_1$ is in $\mathcal{M}$, it is sufficient to show that $p'(V_1, Y)$ is a directed path from V_1 to Y. Note that by the choice of ${p'}^*$, ${p'}^*(V_1, Y)$ is a shortest possibly directed path from V_1 to Y in $\mathcal{P}$. Hence, it is unshielded (Lemma B.1 from Zhang (2008b), see Lemma 41 in Appendix A). If $V_1{\circ\!\!\rightarrow}V_2$ or $V_1 \to V_2$ is in $\mathcal{P}$, then by the choice of $\mathcal{M}$ (Lemma 43), $V_1 \to V_2$ is in $\mathcal{M}$. Additionally, since $p'(V_1, Y)$ is a possibly directed definite status path, $V_1 \to V_2$ in $\mathcal{M}$ implies that all the remaining edges on $p'(V_1, Y)$ are oriented towards Y.

Otherwise, $V_1{\circ\!\!-\!\!\circ}V_2$ is in $\mathcal{P}$. Path ${p'}^*$ is not of definite status, whereas ${p'}^*(V_1, Y)$ is of definite status, as it is unshielded. Thus, node V_1 is not of definite status on ${p'}^*$ and $X \in \mathrm{Adj}(V_2, \mathcal{P})$. The edge $X{\leftarrow\!\!\bullet}V_2$ is not in $\mathcal{P}$ since ${p'}^*(X, V_2)$ is a possibly directed path from X to V_2 in $\mathcal{P}$ (Lemma 7.5 from Maathuis and Colombo (2015), see Lemma 44 in

Appendix A). Since ${p'}^*$ is a shortest possibly directed path from X to Y in $\mathcal{P}$ that does not start with a visible edge out of X, and $X{\leftarrow\!\!\bullet}V_2$ is not in $\mathcal{P}$, it follows that $X \to V_2$ is visible in $\mathcal{P}$. Since $X \to V_2$ is visible, there is a node $D \notin \mathrm{Adj}(V_2, \mathcal{P})$ such that (1) $D{\bullet\!\!\rightarrow}X$ is in $\mathcal{P}$, or (2) there is a collider path $\langle D, D_1, \dots, D_k, X\rangle, k \geq 1$, that is into X in $\mathcal{P}$ such that every $D_i, 1 \leq i \leq k$ is a parent of V_2 in $\mathcal{P}$. We consider these cases separately and show that we arrive at a contradiction, implying that ${p'}^*(V_1, Y)$ cannot start with $V_1{\circ\!\!-\!\!\circ}V_2$.

(1) A node $D \notin \mathrm{Adj}(V_2, \mathcal{P})$ such that $D{\bullet\!\!\rightarrow}X$ is in $\mathcal{P}$. Since $D{\bullet\!\!\rightarrow}X$ and $X{\circ\!\!-\!\!\circ}V_1$, $X{\circ\!\!\rightarrow}V_1$ or $X \to V_1$ is invisible in $\mathcal{P}$, by Lemma 36 and the definition of visibility, an edge between D and V_1 is in $\mathcal{P}$. This edge is of type $D{\bullet\!\!\rightarrow}V_1$, since otherwise both a possibly directed path $\langle X, V_1, D\rangle$ and $D{\bullet\!\!\rightarrow}X$ are in $\mathcal{P}$ (contrary to Lemma 44). Then $D{\bullet\!\!\rightarrow}V_1{\circ\!\!-\!\!\circ}V_2$ is in $\mathcal{P}$ and Lemma 36 implies that $D \in \mathrm{Adj}(V_2, \mathcal{P})$, a contradiction.

(2) There is a node $D \notin \mathrm{Adj}(V_2, \mathcal{P})$ and a collider path $\langle D, D_1, \dots, D_k, X\rangle, k \geq 1$, into X such that every $D_i, 1 \leq i \leq k$ is a parent of V_2 in $\mathcal{P}$. Paths $D_i \to V_2{\circ\!\!-\!\!\circ}V_1$, $i = 1, \dots, k$ are in $\mathcal{P}$, so by Lemma 36 either $D_i{\circ\!\!\rightarrow}V_1$ or $D_i \to V_1$ is in $\mathcal{P}$, for $i = 1, \dots, k$. If $D_1{\circ\!\!\rightarrow}V_1$ is in $\mathcal{P}$, then $D{\bullet\!\!\rightarrow}D_1{\circ\!\!\rightarrow}V_1$ implies $D{\bullet\!\!\rightarrow}V_1$ is also in $\mathcal{P}$ (Lemma 36). But, then $D{\bullet\!\!\rightarrow}V_1{\circ\!\!-\!\!\circ}V_2$ implies $D \in \mathrm{Adj}(V_2, \mathcal{P})$ (Lemma 36), a contradiction. Hence, $D_1 \to V_1$ is in $\mathcal{P}$.

This allows us to deduce that $D \notin \mathrm{Adj}(V_1, \mathcal{P})$, otherwise $D{\bullet\!\!\rightarrow}D_1 \to V_1$ would imply $D{\bullet\!\!\rightarrow}V_1$ (Lemma 44) and we arrive at the contradiction $D \in \mathrm{Adj}(V_2, \mathcal{P})$, as above. Hence, $\langle D, D_1, D_2, V_1\rangle$ is a discriminating path from D to V_1 for D_2, implying that D_2 is of definite status on this path (**R4** of the FCI orientation rules in Appendix A). Thus, $D_2{\circ\!\!\rightarrow}V_1$ is not possible, and since $D_2 \leftrightarrow V_1$ is already ruled out by Lemma 36, $D_2 \to V_1$ is in $\mathcal{P}$. By the same reasoning, $D_i \to V_1$ is in $\mathcal{P}$, for $i = 3, \dots k$. It then follows that $\langle D, D_1, \dots, D_k, X, V_1\rangle$ is a discriminating path from D to V_1 for X in $\mathcal{P}$, so $X \to V_1$ is in $\mathcal{P}$ (**R4** in Appendix A and the fact that $X{\circ\!\!-\!\!\circ}V_1$, $X{\circ\!\!\rightarrow}V_1$ or invisible $X \to V_1$ is in $\mathcal{P}$) and $X \to V_1$ is visible. This contradicts the fact $X{\circ\!\!-\!\!\circ}V_1$, $X{\circ\!\!\rightarrow}V_1$ or invisible $X \to V_1$ is in $\mathcal{P}$. ■

Proof of Lemma 8. First suppose that $\mathcal{G}$ is amenable relative to $(\mathbf{X}, \mathbf{Y})$, meaning that every proper possibly directed path from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}$ starts with a visible edge out of $\mathbf{X}$. Any visible edge in $\mathcal{G}$ is visible in all DAGs (MAGs) in $[\mathcal{G}]$, and any proper directed path in a DAG (MAG) in $[\mathcal{G}]$ corresponds to a proper possibly directed path in $\mathcal{G}$. Hence, any proper directed path from $\mathbf{X}$ to $\mathbf{Y}$ in any DAG (MAG) in $[\mathcal{G}]$ starts with a visible edge out of $\mathbf{X}$.

Next, suppose that $\mathcal{G}$ violates the amenability condition relative to $(\mathbf{X}, \mathbf{Y})$. We will show that this implies that there is no adjustment set relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$. Since every directed edge in a CPDAG is visible and since the same does not hold true in a PAG, we give separate proofs for CPDAGs and PAGs.

Suppose that $\mathcal{G}$ is a PAG. Since $\mathcal{G}$ violates the amenability condition relative to $(\mathbf{X}, \mathbf{Y})$, there exists a proper possibly directed path p^* from some $X \in \mathbf{X}$ to some $Y \in \mathbf{Y}$ in $\mathcal{G}$ that does not start with visible edge out of X in $\mathcal{G}$. Then by Lemma 49 there is a MAG $\mathcal{M}$ in $[\mathcal{G}]$ such that the path p in $\mathcal{M}$, consisting of the same sequence of nodes as p^* in $\mathcal{P}$, contains a subsequence p' that is a proper directed path from X to Y starting with an invisible edge in $\mathcal{M}$. Hence, $\mathcal{M}$ violates the amenability condition relative to $(\mathbf{X}, \mathbf{Y})$. Since the generalized adjustment criterion is complete for MAGs (Theorem 5.8 in van der Zander et al., 2014) this means that there is no adjustment set relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{M}$. Hence, there is no adjustment set relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$.

Next, suppose $\mathcal{G}$ is a CPDAG. We now show how to find DAGs $\mathcal{D}_1$ and $\mathcal{D}_2$ in $[\mathcal{G}]$, such that a proper causal path q'_1 from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{D}_1$ corresponds to a proper non-causal path q'_2 from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{D}_2$ that does not contain colliders. Since $\mathcal{G}$ is not amenable relative to $(\mathbf{X}, \mathbf{Y})$, there is a proper possibly directed path q^* from a node $X \in \mathbf{X}$ to a node $Y \in \mathbf{Y}$ that starts with a non-directed edge ($\circ\!\!-\!\!\circ$).

Let $q'^* = \langle X = V_0, V_1, \ldots, V_k = Y \rangle, k \geq 1$, be a shortest subsequence of q^* such that q'^* is also a proper possibly directed path from X to Y starting with a non-directed edge in $\mathcal{G}$. Since we chose q'^* using the additional constraint that it must start with a non-directed edge in $\mathcal{G}$, we cannot use Lemma 41 to guarantee that q'^* is of definite status. Hence, we first show that q'^* is a definite status path, by contradiction. Thus, suppose that q'^* is not a definite status path. Then $k \geq 2$. Since the subpath $q'^*(V_1, Y)$ is a definite status path (otherwise, by Lemma 41 we can choose a shorter path), this means that V_1 is not of definite status on q'^*. This implies $X \in \mathrm{Adj}(V_2, \mathcal{G})$. Moreover, we must have $X \to V_2$, since $X \circ\!\!-\!\!\circ V_2$ contradicts the choice of q'^*, and $X \leftarrow V_2$ together with the possibly directed path $q'^*(X, V_2)$ contradicts Lemma 44. Then $X \to V_2$ implies $V_1 \to V_2$, otherwise $X \to V_2$ and a possibly directed path $-q'^*(V_2, X)$ are in $\mathcal{G}$, which contradicts Lemma 44. But then V_1 is a definite non-collider on q'^*, which contradicts that V_1 is not of definite status.

Hence, q'^* is a proper definite status possibly directed path from X to Y. By Lemma 43, there is a DAG $\mathcal{D}_1$ in $[\mathcal{G}]$ such that there are no additional arrowheads into X, as well as a DAG $\mathcal{D}_2$ in $[\mathcal{G}]$ such that there are no additional arrowheads into V_1. Let q'_1 in $\mathcal{D}_1$ (q'_2 in $\mathcal{D}_2$) be the path corresponding to q'^* in $\mathcal{G}$. Then q'_1 is of the form $= X \to V_1 \to \cdots \to Y$ and q'_2 is of the form $X \leftarrow V_1 \to \cdots \to Y$. Since q'_1 is a proper causal path from X to Y and q'_2 is a proper non-causal path from X to Y, $f(\mathbf{y} \mid do(\mathbf{x}))$ generally differs in $\mathcal{D}_1$ and $\mathcal{D}_2$. Since $\mathcal{D}_1$ and $\mathcal{D}_2$ are both represented by $\mathcal{C}$, this implies that $f(\mathbf{y} \mid do(\mathbf{x}))$ is not identifiable in $\mathcal{C}$. ■

Proof of Lemma 9. We first prove (i)⇒(ii). Suppose that $\mathbf{Z} \cap \mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) = \emptyset$. Since $\mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D}) \subseteq \mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$ ($\mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{M}) \subseteq \mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$) for any DAG $\mathcal{D}$ (MAG $\mathcal{M}$) in $[\mathcal{G}]$, it follows directly that $\mathbf{Z}$ satisfies the forbidden set condition relative to $(\mathbf{X}, \mathbf{Y})$ in all DAGs (MAGs) in $[\mathcal{G}]$.

Next, we prove ¬ (i)⇒ ¬ (ii). Suppose that $\mathcal{G}$ is amenable relative to $(\mathbf{X}, \mathbf{Y})$, but there is a node $V \in (\mathbf{Z} \cap \mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G}))$. Then $V \in \mathrm{PossDe}(W, \mathcal{G})$ for some $W = V_i, 1 \leq i \leq k$ on a proper possibly directed path $p = \langle X = V_0, V_1, \ldots, V_k = Y \rangle, k \geq 1$. Then $q = p(X, W)$ is proper and $r = p(W, Y)$, where r is allowed to be of zero length (if $W = Y$), does not contain a node in $\mathbf{X}$. Moreover, there is a possibly directed path s from W to V, where this path is allowed to be of zero length. We will show that this implies that there is a DAG $\mathcal{D}$ (MAG $\mathcal{M}$) in $[\mathcal{G}]$ such that $\mathbf{Z} \cap \mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D}) \neq \emptyset$ ($\mathbf{Z} \cap \mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{M}) \neq \emptyset$).

By Lemma 41, there are subsequences q', r' and s' of q, r and s that are unshielded possibly directed paths (again r' and s' are allowed to be paths of zero length). Moreover, q' is proper and must start with a directed (visible) edge, since otherwise the concatenated path $q' \oplus r'$, which is a proper possibly directed path from X to Y, would violate the amenability condition. Lemma 42 then implies that q' is a directed path from X to W in $\mathcal{G}$.

By Lemma 43, there is at least one DAG $\mathcal{D}$ (MAG $\mathcal{M}$) in $[\mathcal{G}]$ that has no additional arrowheads into W. In this graph $\mathcal{D}$ ($\mathcal{M}$), the first edge on the path corresponding to r' is oriented out of W and since r' is an unshielded possibly directed path in $\mathcal{P}$, by Lemma 42 the path in $\mathcal{D}$ ($\mathcal{M}$) corresponding to r' is a directed path from W to Y. By the same

reasoning, the path corresponding to s' in $\mathcal{D}$ ($\mathcal{M}$) is a directed path from W to V. Hence, $V \in \mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D})$ ($V \in \mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{M})$), so that $\mathbf{Z}$ does not satisfy the forbidden set condition relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$ ($\mathcal{M}$). ■

We now start the path of proving Lemma 10. The most involved part is proving the implication ¬ (ii)⇒ ¬ (i), that is, if there is a proper non-causal path p from $\mathbf{X}$ to $\mathbf{Y}$ that is m-connecting given $\mathbf{Z}$ in a DAG $\mathcal{D}$ (MAG $\mathcal{M}$) in $[\mathcal{G}]$, then there must be a proper non-causal definite status path p^* from $\mathbf{X}$ to $\mathbf{Y}$ that is m-connecting given $\mathbf{Z}$ in $\mathcal{G}$ as well. We proceed in three steps. First, we show that proper non-causal paths from $\mathbf{X}$ to $\mathbf{Y}$ that are m-connecting given $\mathbf{Z}$ in $\mathcal{D}$ ($\mathcal{M}$) correspond to proper non-causal paths in $\mathcal{G}$ (Lemma 50). Second, we show that a certain shortest proper non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ that is m-connecting given $\mathbf{Z}$ in $\mathcal{D}$ ($\mathcal{M}$) corresponds to a proper definite status non-causal path p^* from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}$ (Lemma 51). Lastly, we show that p^* is also m-connecting given $\mathbf{Z}$ in $\mathcal{G}$ (Lemma 52).

Lemma 50 *Let $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$ be pairwise disjoint node sets in a* PAG (CPDAG) *$\mathcal{P}$. Let $\mathcal{P}$ be amenable relative to $(\mathbf{X}, \mathbf{Y})$ and let $\mathbf{Z}$ satisfy the forbidden set condition relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{P}$. Let $\mathcal{M}$ be a* MAG (DAG) *in $[\mathcal{P}]$ and let $p = \langle X = V_0, V_1, \ldots, V_n = Y \rangle, n \geq 2$, be a proper non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ that is m-connecting given $\mathbf{Z}$ in $\mathcal{M}$. Let p^* in $\mathcal{P}$ denote the path corresponding to p in $\mathcal{M}$. Then:*

(i) Let $i, j \in \mathbb{N}, 0 < i < j \leq n$ such that there is an edge $\langle V_i, V_j \rangle$ in $\mathcal{P}$. The path $p^*(X, V_i) \oplus \langle V_i, V_j \rangle \oplus p^*(V_j, Y)$ ($p^*(V_j, Y)$ is possibly of zero length) is a proper non-causal path in $\mathcal{P}$. For $j = i + 1$, this implies that p^* is a proper non-causal path.

(ii) If V_1 is not of definite status on p^*, then $\langle X, V_2 \rangle \oplus p^*(V_2, Y)$, ($p^*(V_2, Y)$ is possibly of zero length) exists and is a proper non-causal path in $\mathcal{P}$.

(iii) If $n \geq 3$ and $V_k, 2 \leq k < n$ is not of definite status on p^*, and every non-endpoint node on $p^*(X, V_k)$ is a collider on p^* and a parent of V_{k+1} in $\mathcal{M}$, then $\langle X, V_{k+1} \rangle \oplus p^*(V_{k+1}, Y)$, ($p^*(V_{k+1}, Y)$ is possibly of zero length) exists and is a proper non-causal path in $\mathcal{P}$.

.

Proof of Lemma 50. All paths considered are proper as they are subsequences of p^*, which consists of the same sequence of nodes as the proper path p.

(i) Suppose for a contradiction that $q^* = p^*(X, V_i) \oplus \langle V_i, V_j \rangle \oplus p^*(V_j, Y)$ is possibly directed in $\mathcal{P}$. All nodes on q^* except X are in $\mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{P})$. Since $\mathcal{P}$ is amenable relative to $(\mathbf{X}, \mathbf{Y})$, q^* starts with a visible edge $X \to V_1$ in $\mathcal{P}$. Edge $X \to V_1$ is then also in $\mathcal{M}$ and since p is a non-causal path in $\mathcal{M}$, there is at least one collider on p. Let $V_r, r \geq 1$, be the collider closest to X on p. Then $V_r \in \mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{P})$. Since p is m-connecting given $\mathbf{Z}$, a descendant of V_r is in $\mathbf{Z}$. This contradicts $\mathbf{Z} \cap \mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{P}) = \emptyset$.

(ii) Since V_1 is not of definite status on p^* there is an edge between X and V_2 in $\mathcal{P}$, so path $q^* = \langle X, V_2 \rangle \oplus p^*(V_2, Y)$ exists in $\mathcal{P}$. Suppose for a contradiction that q^* is a possibly directed path from X to Y in $\mathcal{P}$. Then $X \to V_2$ is in $\mathcal{P}$, since $\mathcal{P}$ is amenable relative to

$(\mathbf{X}, \mathbf{Y})$ and every node on q^* except X is in $\text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{P})$. From (i) above we know that p^* is non-causal, so since q^* is possibly directed, there is an arrowhead towards X on $p^*(X, V_2)$.

First, suppose $X \bullet\!\!\bullet V_1 \leftarrow\!\bullet V_2$. Then $X \rightarrow V_2 \bullet\!\!\rightarrow V_1$ implies $X \bullet\!\!\rightarrow V_1$ is in $\mathcal{P}$, since $\mathcal{P}$ is ancestral. This contradicts that V_1 is not of definite status on p^*.

Next, suppose $X \leftarrow\!\bullet V_1 \bullet\!\!\bullet V_2$ is in $\mathcal{P}$. If $\mathcal{P}$ is a CPDAG then V_1 is a definite non-collider on p^*, which contradicts that V_1 is not of definite status. If $\mathcal{P}$ is a PAG, then since $X \rightarrow V_2$ is a visible edge in $\mathcal{P}$ there is a node $D \notin \text{Adj}(V_2, \mathcal{P})$ such that $D \bullet\!\!\rightarrow X$ in $\mathcal{P}$ (there is a collider path $D \bullet\!\!\rightarrow D_1 \leftrightarrow \cdots \leftrightarrow D_s \leftrightarrow X, s \geq 1$, where every node $D_1, \ldots, D_s$ is a parent of V_2 in $\mathcal{P}$). The path $\langle D, X, V_1, V_2\rangle$ ($\langle D, D_1, \ldots, D_s, X, V_1, V_2\rangle$) is a discriminating path from D to V_2 for V_1 in $\mathcal{P}$ so V_1 is of definite status on p^*, contrary to the assumption.

(iii) If $\mathcal{P}$ is a CPDAG and $p(X, V_k), k \geq 2$ is a collider path, then it must be that $k = 2$ and V_k is of definite status on p. Hence, let $\mathcal{P}$ be a PAG. There is an edge between X and V_{k+1} in $\mathcal{P}$, otherwise by Lemma 38 the subpath $p^*(X, V_{k+1})$ is a discriminating path for V_k in $\mathcal{P}$, so V_k would be of definite status on p^*. Then path $q^* = \langle X, V_{k+1}\rangle \oplus p^*(V_{k+1}, Y)$ exists in $\mathcal{P}$. Suppose for a contradiction that q^* is a possibly directed path from X to Y in $\mathcal{P}$. Because $\mathcal{P}$ is amenable relative to $(\mathbf{X}, \mathbf{Y})$ the edge $X \rightarrow V_{k+1}$ is visible in $\mathcal{P}$. Also, since $V_1 \rightarrow V_{k+1}$ is in $\mathcal{M}$, edge $\langle V_1, V_{k+1}\rangle$ is possibly directed towards V_{k+1} in $\mathcal{P}$.

Consider the edge $X \bullet\!\!\rightarrow V_1$ in $\mathcal{P}$. If $X \bullet\!\!\rightarrow V_1$ is not into X in $\mathcal{P}$ then $p^*(X, V_1) \oplus \langle V_1, V_{k+1}\rangle \oplus p^*(V_{k+1}, Y)$ is a proper possibly directed path from X to Y in $\mathcal{P}$ so $V_1 \in \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{P})$. By assumption V_1 is a collider on p^* in $\mathcal{P}$ and hence also on p in $\mathcal{M}$. Since p is m-connecting given $\mathbf{Z}$, there is a node $Z \in \mathbf{Z}$ such that $Z \in \text{De}(V_1, \mathcal{M})$. Since $\text{De}(V_1, \mathcal{M}) \subseteq \text{PossDe}(V_1, \mathcal{P}) \subseteq \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{P})$, node $Z \in \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{P})$. This contradicts the forbidden set condition.

So $X \leftrightarrow V_1$ must be in $\mathcal{P}$. Since $X \rightarrow V_{k+1}$ is a visible edge in $\mathcal{P}$ there is a node $D \notin \text{Adj}(V_{k+1}, \mathcal{P})$ such that the edge $D \bullet\!\!\rightarrow X$ is in $\mathcal{P}$ (there is a collider path $D \bullet\!\!\rightarrow D_1 \leftrightarrow \cdots \leftrightarrow D_s \leftrightarrow X, s \geq 1$, and every node $D_1, \ldots, D_s$ is a parent of V_{k+1} in $\mathcal{P}$). By Lemma 38, path $\langle D, X, V_1, \ldots, V_k, V_{k+1}\rangle$ ($\langle D, D_1, \ldots, D_s, X, V_1, \ldots, V_k, V_{k+1}\rangle$) is then a discriminating path from D to V_{k+1} for V_k in $\mathcal{P}$. Hence, V_k is of definite status on p^*, contrary to the original assumption. ■

Lemma 51 *Let $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$ be pairwise disjoint node sets in a* PAG (CPDAG) *$\mathcal{P}$. Let $\mathcal{P}$ be amenable relative to $(\mathbf{X}, \mathbf{Y})$ and let $\mathbf{Z}$ satisfy the forbidden set condition relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{P}$. Let $\mathcal{M}$ be a* MAG (DAG) *in $[\mathcal{P}]$ and let p be a shortest proper non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ that is m-connecting given $\mathbf{Z}$ in $\mathcal{M}$. Let p^* in $\mathcal{P}$ be the corresponding path to p in $\mathcal{M}$. Then p^* is a proper definite status non-causal path in $\mathcal{P}$.*

Proof of Lemma 51. Let $p = \langle X = V_0, V_1, \ldots, V_k = Y\rangle, k \geq 1$, such that $X \in \mathbf{X}, Y \in \mathbf{Y}$. It follows directly that p^* is proper and by (i) in Lemma 50, it is also non-causal in $\mathcal{P}$.

It is left to prove that p^* is of definite status in $\mathcal{P}$. For this we rely on the proof of Lemma 1 from Zhang (2006) (see Lemma 39 in Appendix A) and prove the following claims for p^*.

Claim 1 *If $V_r, 1 \leq r \leq k-1$ is not of definite status on p^*, then V_{r+1} is a parent of V_{r-1} in $\mathcal{M}$.*

Claim 2 *If $V_r, 1 \leq r \leq k-1$ is not of definite status on p^*, then V_{r-1} is a parent of V_{r+1} in $\mathcal{M}$.*

These claims contradict each other, so every node on p^* must be of definite status. Zhang (2006) proved these claims for a path q^* in $\mathcal{P}$, which is the path corresponding to a shortest path q from $\mathbf{X}$ to $\mathbf{Y}$ that is m-connecting given $\mathbf{Z}$ in $\mathcal{M}$. These claims are proven using the following argument:

If a node on q^* is not of definite status, then a subsequence q'^* that is formed by "jumping" over one, or a sequence of nodes on q^* one of which is not of definite status, constitutes a path in $\mathcal{P}$. Let q' in $\mathcal{M}$ be the path corresponding to q'^* in $\mathcal{P}$. Then q' is a path from $\mathbf{X}$ to $\mathbf{Y}$ that is shorter than q, so it is blocked by $\mathbf{Z}$ in $\mathcal{M}$. By the choice of q', the collider/definite non-collider status of all nodes on q', except two, is the same as on q. Therefore, one of these two nodes must block q' in $\mathcal{M}$. All possible cases for the status of these two nodes are considered in Zhang (2006) and a contradiction is reached in every case that does not support the claim being proven.

Almost exactly the same argument as in Zhang (2006) can be carried out to prove that Claim 1 and 2 hold for p^*. The only difference is in considering the paths that are subsequences of p. Since these paths are shorter than p and proper they are either blocked by $\mathbf{Z}$ or causal in $\mathcal{M}$. However, the subsequences of p considered in the proof of Lemma 39 are either immediately non-causal or they are constructed as in (i)-(iii) in Lemma 50 and thus non-causal by Lemma 50. The argument from Zhang (2006) then still holds for Claims 1 and 2 for p^*. ■

Lemma 52 *Let $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$ be pairwise disjoint node sets in a* PAG (CPDAG) *$\mathcal{P}$. Let $\mathcal{P}$ be amenable relative to $(\mathbf{X}, \mathbf{Y})$ and let $\mathbf{Z}$ satisfy the forbidden set condition relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{P}$. Let $\mathcal{M}$ be a* MAG (DAG) *in $[\mathcal{P}]$ and let p be a path with minimal* distance-from-$\mathbf{Z}$ *among the shortest proper non-causal paths from $\mathbf{X}$ to $\mathbf{Y}$ that are m-connecting given $\mathbf{Z}$ in $\mathcal{M}$. Let p^* in $\mathcal{P}$ be the corresponding path to p in $\mathcal{M}$. Then p^* is a proper definite status non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ that is m-connecting given $\mathbf{Z}$ in $\mathcal{P}$.*

Proof of Lemma 52. By Lemma 51, p^* is a proper definite status non-causal path in $\mathcal{P}$. It is only left to prove that p^* is m-connecting given $\mathbf{Z}$ in $\mathcal{P}$.

Every definite non-collider on p^* in $\mathcal{P}$ corresponds to a non-collider on p in $\mathcal{M}$, and every collider on p^* is also a collider on p. Since p is m-connecting given $\mathbf{Z}$, no non-collider is in $\mathbf{Z}$ and every collider has a descendant in $\mathbf{Z}$. Let Q be an arbitrary collider (if there is one) on p. Then there is a directed path (possibly of zero length) from Q to a node in $\mathbf{Z}$ in $\mathcal{M}$. Let d be a shortest such path from Q to a node in $\mathbf{Z}$. Let d^* in $\mathcal{P}$ denote the corresponding path to p in $\mathcal{M}$. Then d^* is a possibly directed path from Q to Z in $\mathcal{P}$. It is only left to prove that d^* is a directed path. If d^* is of zero length, this is trivially true. Otherwise, suppose for a contradiction that there is a circle mark on d^*. Then d^* must start with a circle mark at Q (Lemma 7.2 from Maathuis and Colombo (2015) and Lemma 42, see Appendix A).

We first prove that d^* is unshielded in $\mathcal{P}$. Suppose for a contradiction that d^* is shielded. Then there exists a subpath $\langle A, B, C\rangle$ of d^* such that the edge $\langle A, C\rangle$ is in $\mathcal{P}$. The path corresponding to $d^*(Q, A) \oplus \langle A, C\rangle \oplus d^*(C, Z)$ must be a non-causal path from Q to Z in $\mathcal{M}$, otherwise we could have chosen a shorter path d. Hence, the edge $A \leftarrow\!\bullet C$ is in $\mathcal{M}$. But path d is directed from Q to Z in $\mathcal{M}$ so $A \rightarrow B \rightarrow C$ is also in $\mathcal{M}$. This contradicts that $\mathcal{M}$ is ancestral.

Let S be the first node on d after Q. If S is not a node on p, then following the proof of Lemma 2 from Zhang (2006) (Lemma 40 in Appendix 2) there exist nodes W on $p(X, Q)$ and

V on $p(Q,Y)$, distinct from Q, such that the path $W \bullet\!\!\rightarrow S \leftarrow\!\!\bullet V$ is in $\mathcal{M}$ and both W and V have the same colliders/non-collider status of both p and $p' = p(X,W) \oplus \langle W,S,V\rangle \oplus p(V,Y)$. Then p' is m-connecting given $\mathbf{Z}$. Since p' is non-causal and shorter than p, or as long as p but with a shorter distance-from-$\mathbf{Z}$ than p, p' must be non-proper, that is, $S \in \mathbf{X}$. But then $\langle S,V\rangle \oplus p(V,Y)$ is a proper non-causal m-connecting path from $\mathbf{X}$ to $\mathbf{Y}$ given $\mathbf{Z}$ that is shorter than p in $\mathcal{M}$. This contradicts our assumption about p.

If S is a node on p, then it lies either on $p(X,Q)$ or $p(Q,Y)$. Assume without loss of generality that S is on $p(Q,Y)$. Following the proof of Lemma 2 from Zhang (2006), there exists a node $W, W \neq Q$, on $p(X,Q)$ such that $W \bullet\!\!\rightarrow S$ is in $\mathcal{M}$ and W has the same collider/non-collider status on both p and $p' = p(X,W) \oplus \langle W,S\rangle \oplus p(S,Y)$. Then p' is m-connecting given $\mathbf{Z}$. Since p' is proper, and shorter than p, or as long as p but with a shorter distance-from-$\mathbf{Z}$ than p, p' must be causal in $\mathcal{M}$. Let p'^* in $\mathcal{P}$ denote the corresponding path to p in $\mathcal{M}$. Then p'^* is a possibly directed path from $\mathbf{X}$ to $\mathbf{Y}$, S is on p'^* and $Z \in \text{PossDe}(S,\mathcal{P})$, so $Z \in \text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{P}) \cap \mathbf{Z}$. This is a contradiction with $\mathbf{Z} \cap \text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{P}) = \emptyset$. ■

Proof of Lemma 10. Let the CPDAG (PAG) $\mathcal{G}$ be amenable relative to $(\mathbf{X},\mathbf{Y})$, and let $\mathbf{Z} \cap \text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G}) = \emptyset$.

We first prove ¬ (i)⇒ ¬ (iii). Assume $\mathbf{Z}$ does not satisfy the blocking condition relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{G}$. Thus, there is a proper definite status non-causal path p from $X \in \mathbf{X}$ to $Y \in \mathbf{Y}$ that is m-connecting given $\mathbf{Z}$ in $\mathcal{G}$. Consider any DAG $\mathcal{D}$ (MAG $\mathcal{M}$) in $[\mathcal{G}]$. The path corresponding to p in $\mathcal{D}$ ($\mathcal{M}$) is a proper non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ that is m-connecting given $\mathbf{Z}$. Hence, $\mathbf{Z}$ does not satisfy the blocking condition relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{D}$ ($\mathcal{M}$) for all $\mathcal{D}$ ($\mathcal{M}$) in $[\mathcal{G}]$.

The implication ¬ (iii)⇒ ¬ (ii) trivially holds, so it is only left to prove that ¬ (ii)⇒ ¬ (i). Thus, assume that there is a DAG $\mathcal{D}$ (MAG $\mathcal{M}$) in $[\mathcal{G}]$ such that there is a proper non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{D}$ ($\mathcal{M}$) that is m-connecting given $\mathbf{Z}$. We choose a path p with minimal distance-from-$\mathbf{Z}$ among the shortest proper non-causal paths from $\mathbf{X}$ to $\mathbf{Y}$ that are m-connecting given $\mathbf{Z}$ in $\mathcal{D}$ ($\mathcal{M}$). By Lemma 52, the corresponding path p^* in $\mathcal{G}$ is a proper definite status non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ that is m-connecting given $\mathbf{Z}$. ■

Proof of Theorem 7. Assume that $\mathcal{G}$ is amenable relative to $(\mathbf{X},\mathbf{Y})$ and $\mathbf{Z}$ satisfies the blocking condition relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{G}$. Then it is left to prove that $\mathbf{Z}$ satisfies the blocking condition relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{G}$ if and only if it satisfies the separation condition relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{G}^{pbd}_{\mathbf{XY}}$.

We first prove that if $\mathbf{Z}$ satisfies the blocking condition, then $\mathbf{Z}$ m-separates $\mathbf{X}$ and $\mathbf{Y}$ in $\mathcal{G}^{pbd}_{\mathbf{XY}}$, that is, $\mathbf{Z}$ blocks all definite status paths from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}^{pbd}_{\mathbf{XY}}$. Since every definite status path from $\mathbf{X}$ to $\mathbf{Y}$ has a proper definite status path as a subpath, it is enough to show that $\mathbf{Z}$ blocks all proper definite status paths from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}^{pbd}_{\mathbf{XY}}$. Let p be a proper definite status path from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}^{pbd}_{\mathbf{XY}}$. Then p must be non-causal, since otherwise $\mathcal{G}$ is not amenable. Let p^* in $\mathcal{G}$ be the path corresponding to p in $\mathcal{G}^{pbd}_{\mathbf{XY}}$, consisting of the same sequence of nodes as p. Then p^* is a proper definite status non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}$. Thus, p^* is blocked by $\mathbf{Z}$. Since removing edges cannot m-connect a previously blocked path, p is blocked by $\mathbf{Z}$ in $\mathcal{G}^{pbd}_{\mathbf{XY}}$.

Next, we prove that if $\mathbf{Z}$ m-separates $\mathbf{X}$ and $\mathbf{Y}$ in $\mathcal{G}^{pbd}_{\mathbf{XY}}$, then $\mathbf{Z}$ satisfies the blocking condition. Suppose for a contradiction that there exists a proper definite status non-causal path p^* from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}$ that is m-connecting given $\mathbf{Z}$. Let $\tilde{p}^*$ in $\mathcal{G}^{pbd}_{\mathbf{XY}}$ be the path corresponding to p^* in $\mathcal{G}$, constituted by the same sequence of nodes as p^*, if such a path exists in $\mathcal{G}^{pbd}_{\mathbf{XY}}$.

First, suppose $\tilde{p}^*$ does not exist in $\mathcal{G}^{pbd}_{\mathbf{XY}}$. Then since p^* is proper, it must start with a visible edge $X \rightarrow D$ in $\mathcal{G}$ such that D lies on a proper causal path from X to $\mathbf{Y}$, that is, $D \in \text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G})$. Since p^* is non-causal and of definite status it must contain a collider $C \in \text{PossDe}(D,\mathcal{G})$. Since $\text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G})$ is descendral (see Definition 13), $C \in \text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G})$ and similarly all descendants of C are in $\text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G})$. Considering that p^* is m-connecting given $\mathbf{Z}$ and C is a collider on p^*, it follows that $C \in \text{An}(\mathbf{Z},\mathcal{G})$. This contradicts $\mathbf{Z} \cap \text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G}) = \emptyset$.

Otherwise, $\tilde{p}^*$ is a path in $\mathcal{G}^{pbd}_{\mathbf{XY}}$. Then $\tilde{p}^*$ is a proper definite status non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}^{pbd}_{\mathbf{XY}}$ that is blocked by $\mathbf{Z}$. Since p^* is m-connecting given $\mathbf{Z}$ in $\mathcal{G}$, all colliders on p^* are in $\text{An}(\mathbf{Z},\mathcal{G})$. Since no definite non-collider on p^* is in $\mathbf{Z}$, no definite non-collider on $\tilde{p}^*$ is in $\mathbf{Z}$. However, $\tilde{p}^*$ is blocked by $\mathbf{Z}$ in $\mathcal{G}^{pbd}_{\mathbf{XY}}$, so at least one collider C on $\tilde{p}^*$ (and therefore p^* as well) is not in $\text{An}(\mathbf{Z},\mathcal{G}^{pbd}_{\mathbf{XY}})$. Thus, any directed path from C to $\mathbf{Z}$ must contain a visible edge $X \rightarrow D$, where $D \in \text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G})$. This implies $D \in \text{An}(\mathbf{Z},\mathcal{G})$, which contradicts $\mathbf{Z} \cap \text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G}) = \emptyset$. ■

Appendix C. Proofs for Section 4

Proof of Theorem 11. The first part of the theorem follows from from Rule 3 of the do-calculus for DAGs from Pearl (2009), Rule 3 of the do-calculus for MAGs and PAGs from Zhang (2006) and the properties of the CPDAG.

Since the generalized adjustment criterion is sound and complete with respect to adjustment in DAGs, CPDAGs, MAGs and PAGs (Theorem 5), we will prove the second part of this theorem by proving that if $\mathbf{Z}$ satisfies the generalized adjustment criterion relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{G}$, then $\mathbf{Z}$ satisfies the generalized adjustment criterion relative to $(\mathbf{X}',\mathbf{Y})$ in $\mathcal{G}$. We prove this by showing that $\mathbf{Z}$ satisfies the three conditions of Definition 4 relative to $(\mathbf{X}',\mathbf{Y})$ in $\mathcal{G}$.

We first show that a possibly directed path from $\mathbf{X}'$ to $\mathbf{Y}$ is proper with respect to $\mathbf{X}'$ if and only if it is proper with respect to $\mathbf{X}$. Since $\mathbf{X}' \subseteq \mathbf{X}$ any path that is proper with respect to $\mathbf{X}$ will also be proper with respect to $\mathbf{X}'$. Hence, we only need to show that if p is a possibly directed path from $X' \in \mathbf{X}'$ to $Y \in \mathbf{Y}$ that is proper with respect to $\mathbf{X}'$, then p is also proper with respect to $\mathbf{X}$. Suppose for a contradiction that p is not proper with respect to $\mathbf{X}$. Let $p(X,Y)$ be the subpath of p that is proper with respect to $\mathbf{X}$. Since $X \notin \mathbf{X}'$, X must be in $\mathbf{X} \setminus \mathbf{X}'$, which contradicts the assumption that there are no possibly directed paths from $\mathbf{X} \setminus \mathbf{X}'$ to $\mathbf{Y}$ that are proper with respect to $\mathbf{X}$.

Then $\text{Forb}(\mathbf{X}',\mathbf{Y},\mathcal{G}) \subseteq \text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G})$, so $\mathbf{Z}$ must satisfy the forbidden set condition relative to $(\mathbf{X}',\mathbf{Y})$ in $\mathcal{G}$. Additionally, since $\mathbf{Z}$ satisfies the generalized adjustment criterion relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{G}$, $\mathcal{G}$ must be amenable relative to $(\mathbf{X},\mathbf{Y})$. This means that every possibly directed path from $\mathbf{X}$ to $\mathbf{Y}$ that is proper with respect to $\mathbf{X}$ starts with a visible edge out of $\mathbf{X}$. Since $\mathbf{X}' \subseteq \mathbf{X}$ and since possibly directed paths from $\mathbf{X}'$ to $\mathbf{Y}$ that are proper with respect to $\mathbf{X}'$ are proper with respect to $\mathbf{X}$, it follows that the amenability condition is satisfied relative to $(\mathbf{X}',\mathbf{Y})$ and $\mathcal{G}$.

It is only left to show that $\mathbf{Z}$ satisfies the blocking condition relative to $(\mathbf{X}', \mathbf{Y})$ in $\mathcal{G}$. Since $\mathbf{Z}$ satisfies the blocking condition relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$ and since $\mathbf{X}' \subseteq \mathbf{X}$, $\mathbf{Z}$ blocks every definite status non-causal path from $\mathbf{X}'$ to $\mathbf{Y}$ that is proper with respect to $\mathbf{X}$. Hence, we only need to show that $\mathbf{Z}$ also blocks every definite status non-causal path from $\mathbf{X}'$ to $\mathbf{Y}$ that is proper with respect to $\mathbf{X}'$, but not proper with respect to $\mathbf{X}$. Let p be one such path from $X' \in \mathbf{X}'$ to $Y \in \mathbf{Y}$. Since p is proper with respect to $\mathbf{X}'$, but not with respect to $\mathbf{X}$, let $X \in \mathbf{X} \setminus \mathbf{X}'$ be the node on p such that $p(X, Y)$ is proper with respect to $\mathbf{X}$. Since there is no possibly directed path from $\mathbf{X} \setminus \mathbf{X}'$ to $\mathbf{Y}$ that is proper with respect to $\mathbf{X}$, $p(X, Y)$ must be a non-causal path from X to Y. Additionally, since $p(X, Y)$ is a subpath of p, $p(X, Y)$ is of definite status. Then $p(X, Y)$ is a non-causal definite status path from $\mathbf{X}$ to $\mathbf{Y}$ that is proper with respect to $\mathbf{X}$, so $p(X, Y)$ is blocked by $\mathbf{Z}$. Thus, p is also blocked by $\mathbf{Z}$. ■

Proof of Lemma 16. There is a proper definite status non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ that is m-connecting given $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \setminus \mathbf{I}$. Among all such paths consider the ones with minimal length and among those let $p = \langle X, \dots, Y \rangle, X \in \mathbf{X}, Y \in \mathbf{Y}$ be the path with a shortest distance-from-$(\mathbf{X} \cup \mathbf{Y})$ in $\mathcal{G}$. By the choice of p, (i) holds. It is left to prove that (ii)−(iv) also hold for p.

(ii) Since p is m-connecting given $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \setminus \mathbf{I}$ and since $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \setminus \mathbf{I} = \text{PossAn}(\mathbf{X} \cup \mathbf{Y}, \mathcal{G}) \setminus (\mathbf{X} \cup \mathbf{Y} \cup \mathbf{I})$, any collider on p is in $\text{PossAn}(\mathbf{X} \cup \mathbf{Y}, \mathcal{G})$. Since p is proper, no collider on p is in $\mathbf{X}$. Additionally, no collider C on p is in $\mathbf{Y} \setminus \mathbf{I}$, otherwise $p(X, C)$ is a non-causal path and we could have chosen a shorter p. It is only left to show that no collider on p is in $\mathbf{I}$. Suppose for a contradiction that a collider on p is in $\mathbf{I}$. Since $\mathbf{I}$ is a descendral set, all (possible) descendants of this collider are also in $\mathbf{I}$. But then p is not m-connecting given $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \setminus \mathbf{I}$, a contradiction.

(iii) Any definite non-collider on p is a possible ancestor of an endpoint node of p or of a collider on p. Then it follows from (ii) that any definite non-collider on p is in $\text{PossAn}(\mathbf{X} \cup \mathbf{Y}, \mathcal{G})$. Furthermore, p is m-connecting given $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \setminus \mathbf{I}$, so no definite non-collider on p is in $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \setminus \mathbf{I}$. Since $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \setminus \mathbf{I} = \text{PossAn}(\mathbf{X} \cup \mathbf{Y}, \mathcal{G}) \setminus (\mathbf{X} \cup \mathbf{Y} \cup \mathbf{I})$, it follows that all definite non-colliders on p are in $\text{PossAn}(\mathbf{X} \cup \mathbf{Y}, \mathcal{G}) \cap (\mathbf{X} \cup \mathbf{Y} \cup \mathbf{I})$. Since p is proper, no definite non-collider on p is in $\mathbf{X}$. Additionally, no definite non-collider C on p is in $\mathbf{Y} \setminus \mathbf{I}$, otherwise we could have chosen path $p(X, C)$ instead of p. Thus, all definite non-colliders on p are in $\text{PossAn}(\mathbf{X} \cup \mathbf{Y}, \mathcal{G}) \cap \mathbf{I} \subseteq \mathbf{I}$.

(iv) Let C be a collider on p. From (ii), it follows that $C \notin \mathbf{X} \cup \mathbf{Y}$ and that there is an unshielded possibly directed path from C to a node $V \in \mathbf{X} \cup \mathbf{Y}$. Suppose that this path starts with an edge of type $C \circ\!\!-\!\!\circ Q$ (possibly $Q = V$). We derive a contradiction by constructing a proper definite status non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ that is m-connecting given $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \setminus \mathbf{I}$ and shorter than p, or of the same length as p, but with a shorter distance-from-$(\mathbf{X} \cup \mathbf{Y})$.

Let A and B be nodes on p such that $A \bullet\!\!\rightarrow C \leftarrow\!\!\bullet B$ is a subpath of p (possibly $A = X$, $B = Y$). Then paths $A \bullet\!\!\rightarrow C \circ\!\!-\!\!\circ Q$ and $B \bullet\!\!\rightarrow C \circ\!\!-\!\!\circ Q$ together with Lemma 36 imply that $A \bullet\!\!\rightarrow Q \leftarrow\!\!\bullet B$ is in $\mathcal{G}$.

Since all colliders on p are not in $\mathbf{I}$, and all definite non-colliders on p are in $\mathbf{I}$, and since $\mathbf{I}$ is a descendral set, it follows that no collider on p is a possible descendant of a definite non-collider on p. Thus, if $A \neq X$ $(B \neq Y)$, then $A \leftrightarrow C$ $(C \leftrightarrow B)$ is in $\mathcal{G}$. Moreover, if $A \leftrightarrow C$ $(C \leftrightarrow B)$ is in $\mathcal{G}$, then $A \leftrightarrow Q$ $(Q \leftrightarrow B)$ is in $\mathcal{G}$, otherwise a possibly directed path

$\langle A, Q, C \rangle$ $(\langle B, Q, C \rangle)$ and $C \leftrightarrow A$ $(C \leftrightarrow B)$ are in $\mathcal{G}$, which contradicts Lemma 44. Hence, if $A \neq X$ $(B \neq Y)$, the collider/definite non-collider status of A (B) is the same on p and on $p(X, A) \oplus \langle A, Q \rangle$ $(\langle Q, B \rangle \oplus p(B, Y))$.

Suppose first that $Q \in \mathbf{X} \cup \mathbf{Y}$. If $Q \in \mathbf{X}$, then $\langle Q, B \rangle \oplus p(B, Y)$ is a proper definite status non-causal path that is m-connecting given $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \setminus \mathbf{I}$ in $\mathcal{G}$ and shorter than p. Otherwise, $Q \in \mathbf{Y}$. If $Q \in \mathbf{Y} \cap \mathbf{I}$, this would imply that $C \in \mathbf{I}$, which contradicts (ii). So Q must be in $\mathbf{Y} \setminus \mathbf{I}$. Then $p(X, A) \oplus \langle A, Q \rangle$ is a non-causal path. Hence, we found a proper definite status non-causal path that is m-connecting given $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \setminus \mathbf{I}$ in $\mathcal{G}$ and shorter than p, which is a contradiction.

Next, suppose that $Q \notin \mathbf{X} \cup \mathbf{Y}$. Then if Q is not on p, $p(X, A) \oplus \langle A, Q, B \rangle \oplus p(B, Y)$ is a proper definite status non-causal path that is m-connecting given $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \setminus \mathbf{I}$ in $\mathcal{G}$ and of the same length as p, but with a shorter distance-from-$(\mathbf{X} \cup \mathbf{Y})$. Otherwise, Q is on p. Then Q is a collider on p, otherwise $Q \in \mathbf{I}$ and $C \in \mathbf{I}$. Suppose first that Q is on $p(C, Y)$. Then $p(X, A) \oplus \langle A, Q \rangle \oplus p(Q, Y)$ is a non-causal path because $p(Q, Y)$ is into Q. Hence, there is a proper definite status non-causal path that is m-connecting given $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \setminus \mathbf{I}$ in $\mathcal{G}$ and shorter than p, which is a contradiction. Next, suppose that Q is on $p(X, C)$. Then $p(X, Q) \oplus \langle Q, B \rangle \oplus p(B, Y)$ is a non-causal path since it contains $Q \leftarrow B$. This path is a proper definite status non-causal path that is m-connecting given $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \setminus \mathbf{I}$ in $\mathcal{G}$ and shorter than p, which is a contradiction. ■

The following proof is similar to the proof of Lemma 1 from Richardson (2003) (see Lemma 37 in Appendix A). The difference lies in the fact that Lemma 17 additionally considers CPDAGs and PAGs $\mathcal{G}$ in which we define m-separation (m-connection) only for paths of definite status, as well as the fact that we require the resulting path p to be proper and non-causal from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}$.

Proof of Lemma 17. Let p be a path in $\mathcal{G}$ satisfying (i)−(ii). If there is no collider on p, or all colliders on p are in $\text{An}(\mathbf{Z}, \mathcal{G})$, then p is a proper definite status non-causal path that is m-connecting given $\mathbf{Z}$ in $\mathcal{G}$ and the lemma holds.

Hence, assume there is at least one collider C on p that is not in $\text{An}(\mathbf{Z}, \mathcal{G})$. By (ii), $C \in \text{An}(\mathbf{X} \cup \mathbf{Y}, \mathcal{G}) \setminus \text{An}(\mathbf{Z}, \mathcal{G})$. We now construct a path q from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}$ that is m-connecting given $\mathbf{Z}$ and prove it is proper, of definite status and non-causal.

Let D be the node closest to Y on p such that $D \in \text{An}(\mathbf{X}, \mathcal{G}) \setminus \text{An}(\mathbf{Z}, \mathcal{G})$ if such a node exists, otherwise let $D = X$. Let E be the node closest to D on $p(D, Y)$ such that $E \in \text{An}(\mathbf{Y}, \mathcal{G}) \setminus \text{An}(\mathbf{Z}, \mathcal{G})$ if such a node exists, otherwise let $E = Y$. Since at least one collider on p is in $\text{An}(\mathbf{X} \cup \mathbf{Y}) \setminus \text{An}(\mathbf{Z}, \mathcal{G})$, either $D \neq X$ or $E \neq Y$ must hold. Moreover, if $D = X$, then since there is at least one collider C on p that is in $\text{An}(\mathbf{X} \cup \mathbf{Y}, \mathcal{G}) \setminus \text{An}(\mathbf{Z}, \mathcal{G})$, it follows that $E \neq D$. However, if $D \neq X$, then $D = E$ is possible.

Since $D \in \text{An}(\mathbf{X}, \mathcal{G}) \setminus \text{An}(\mathbf{Z}, \mathcal{G})$, let v_D be a shortest directed path from D to a node in $\mathbf{X}$ (possibly of length zero). Since $E \in \text{An}(\mathbf{Y}, \mathcal{G}) \setminus \text{An}(\mathbf{Z}, \mathcal{G})$, let v_E be a shortest directed path from E to a node in $\mathbf{Y}$ (possibly of length zero). Thus, all non-endpoint nodes on v_D and v_E are in $\text{An}(\mathbf{X} \cup \mathbf{Y}, \mathcal{G}) \setminus \text{An}(\mathbf{Z}, \mathcal{G})$. Also, by the choice of D and E, no non-endpoint node on $p(D, E)$ is in $\text{An}(\mathbf{X} \cup \mathbf{Y}, \mathcal{G}) \setminus \text{An}(\mathbf{Z}, \mathcal{G})$. Hence, no non-endpoint node on either v_D or v_E is also on $p(D, E)$.

Let $q = (-v_D) \oplus p(D, E) \oplus v_E$. We prove that q is a proper definite status non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}$ that is m-connecting given $\mathbf{Z}$. Path q is of definite status by construction.

To prove that q is proper we must show that no non-endpoint node on v_D or v_E is in $\mathbf{X}$. No non-endpoint node on v_D is in $\mathbf{X}$, otherwise we could have chosen a shorter v_D. Similarly, no non-endpoint node on v_E is in $\mathbf{X}$, as this would contradict the choice of D. It is left to show that q is non-causal from $\mathbf{X}$ to $\mathbf{Y}$ and m-connecting given $\mathbf{Z}$.

We first show that q is m-connecting given $\mathbf{Z}$. By assumption, no definite non-collider on p is in $\mathbf{Z}$. Additionally, if D and E are non-endpoints on p, then all nodes on v_D and v_E are in $\mathrm{An}(\mathbf{X}\cup\mathbf{Y},\mathcal{G})\setminus\mathrm{An}(\mathbf{Z},\mathcal{G})$, that is, no node on either v_D or v_E is in $\mathbf{Z}$. Hence, no definite non-collider on q is in $\mathbf{Z}$. For q to be m-connecting given $\mathbf{Z}$ we still have to show that all colliders on q are in $\mathrm{An}(\mathbf{Z},\mathcal{G})$. Any collider on q is a non-endpoint node on $p(D,E)$. Since no non-endpoint node on $p(D,E)$ is in $\mathrm{An}(\mathbf{X}\cup\mathbf{Y},\mathcal{G})\setminus\mathrm{An}(\mathbf{Z},\mathcal{G})$, by choice of D and E, and all colliders on p are in $\mathrm{An}(\mathbf{X}\cup\mathbf{Y}\cup\mathbf{Z},\mathcal{G})$, by assumption, it follows that any collider on $p(D,E)$ must be in $\mathrm{An}(\mathbf{Z},\mathcal{G})$.

It is only left to show that q is a non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}$. If v_D is not of zero length, this is obviously true. If v_D is of zero length, then $q=p(X,E)\oplus v_E$. Since v_E is a directed path from E to Y, we need to show that $p(X,E)$ is non-causal from X to E. Suppose for a contradiction that $p(X,E)$ is possibly directed from X to E. Then q is a proper possibly directed path from $\mathbf{X}$ to $\mathbf{Y}$, so $E\in\mathrm{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G})\subseteq\mathbf{I}$. By assumption, there is at least one collider on p, and in particular on $p(E,Y)$. Let C be the collider on $p(E,Y)$ closest to E. Then $C\in\mathrm{De}(E,\mathcal{G})$. Since $\mathrm{De}(E,\mathcal{G})\subseteq\mathrm{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G})$, it follows that $C\in\mathrm{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G})$. Since $\mathrm{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G})\subseteq\mathbf{I}$, this is in contradiction with (ii). ■

Appendix D. Proofs for Section 5

Proof of Lemma 21. If there is a path p from $\mathbf{X}$ to $\mathbf{Y}$ for which the listed statements hold, then no set $\mathbf{Z}$ such that $\mathbf{Z}\cap\mathrm{De}(\mathbf{X},\mathcal{D})=\emptyset$ can block p.

Conversely, since $\mathbf{Z}\cap\mathrm{De}(\mathbf{X},\mathcal{D})=\emptyset$ and $\mathbf{Z}$ satisfies the generalized adjustment criterion relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{D}$, then $\mathbf{Z}'=\mathrm{Adjust}(\mathbf{X},\mathbf{Y},\mathcal{D})\setminus\mathrm{De}(\mathbf{X},\mathcal{D})$ satisfies the generalized adjustment criterion relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{D}$ (Theorem 14). Suppose there is no back-door set relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{D}$. Then $\mathbf{Z}'$ violates condition (ii) of Pearl's back-door criterion relative to at least one $X\in\mathbf{X}$ and $Y\in\mathbf{Y}$ in $\mathcal{D}$. Hence, we can choose p to be a shortest back-door path from a node $X\in\mathbf{X}$ to a node $Y\in\mathbf{Y}$ that is d-connecting given $\mathbf{Z}'$ in $\mathcal{D}$. Then the statement in (i) holds for p. We prove that the statements (ii)−(iv) also hold for p.

(ii) Since $\mathbf{Z}'$ satisfies the generalized adjustment criterion and $\mathbf{Z}'$ does not block the back-door path p, it follows that p is not proper. Since p is not proper, a subpath of p forms a proper path q from $\mathbf{X}$ to $\mathbf{Y}$. If q is non-causal, then it is blocked by $\mathbf{Z}'$. But in this case $\mathbf{Z}'$ would block p as well. Hence, q is causal.

(iii) Any non-collider on p is an ancestor of an endpoint or a collider on p. Since p is d-connecting given $\mathbf{Z}'$, all colliders on p are in $\mathrm{An}(\mathbf{Z}',\mathcal{D})$. By our choice of $\mathbf{Z}'$, $\mathrm{An}(\mathbf{Z}',\mathcal{D})\subseteq\mathrm{An}(\mathbf{X}\cup\mathbf{Y},\mathcal{D})$, so all colliders on p are in $\mathrm{An}(\mathbf{X}\cup\mathbf{Y},\mathcal{D})$. Since any non-collider on p is an ancestor of an endpoint or a collider on p, it follows that all non-colliders on p are also in $\mathrm{An}(\mathbf{X}\cup\mathbf{Y},\mathcal{D})$. Moreover, since p is d-connecting given $\mathbf{Z}'$, no non-collider on p is in $\mathbf{Z}'$. Thus, since $\mathbf{Z}'=\mathrm{An}(\mathbf{X}\cup\mathbf{Y},\mathcal{D})\setminus(\mathrm{De}(\mathbf{X},\mathcal{D})\cup\mathbf{Y})$, any non-collider on p must be in $\mathrm{An}(\mathbf{X}\cup\mathbf{Y},\mathcal{D})\cap(\mathrm{De}(\mathbf{X},\mathcal{D})\cup\mathbf{Y})$. Path p is a shortest back-door path from $\mathbf{X}$ to $\mathbf{Y}$, so no non-collider on p is in $\mathbf{Y}$. Hence, all non-colliders on p are in $\mathrm{De}(\mathbf{X},\mathcal{D})$.

Now, assume that there is a collider C on p. This collider is a descendant of X or a non-collider on p, so $C\in\mathrm{De}(\mathbf{X},\mathcal{D})$ as well. However, since $\mathrm{De}(\mathbf{X},\mathcal{D})\cap\mathbf{Z}=\emptyset$ and $\mathrm{De}(C,\mathcal{D})\subseteq\mathrm{De}(\mathbf{X},\mathcal{D})$ it follows that p is blocked by $\mathbf{Z}'$. This contradicts that p is d-connecting given $\mathbf{Z}'$.

(iv) From (ii), it follows that $Y\in\mathrm{De}(\mathbf{X},\mathcal{D})$. Additionally, we've shown in (iii) that there is no collider on p and that all non-collider on p are in $\mathrm{De}(\mathbf{X},\mathcal{D})$. Thus, all nodes on p are in $\mathrm{De}(\mathbf{X},\mathcal{D})$. ■

Proof of Lemma 23. Let p be a definite status back-door path from a node $X\in\mathbf{X}$ to a node $Y\in\mathbf{Y}$ in $\mathcal{G}$. Then there exists a node $X'\in\mathbf{X}$ (possibly $X'=X$) on p such that the subpath $p(X',Y)$ is a proper path from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}$. Since $p(X',Y)$ is a subpath of p, $p(X',Y)$ is of definite status.

If $X'\neq X$ and X' is a definite non-collider on p, then since $X'\in\mathbf{Z}\cup\mathbf{X}\setminus\{X\}$, p is blocked by $\mathbf{Z}\cup\mathbf{X}\setminus\{X\}$. Else, $X'\neq X$ and X' is a definite collider on p, or $X'=X$ and p is a proper back-door path in $\mathcal{G}$. Since $\mathcal{G}$ is amenable, all proper back-door paths from $\mathbf{X}$ to $\mathbf{Y}$ are also proper non-causal paths from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}$. We prove that $\mathbf{Z}\cup\mathbf{X}\setminus\{X\}$ blocks p, by proving that it blocks $p(X',Y)$.

Suppose for a contradiction that $p(X',Y)$ is m-connecting given $\mathbf{Z}\cup\mathbf{X}\setminus\{X\}$ in $\mathcal{G}$. We show that it is then possible to construct a proper definite status non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}$, that is m-connecting given $\mathbf{Z}$, which contradicts that $\mathbf{Z}$ satisfies the blocking condition relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{G}$. Since $p(X',Y)$ is a proper back-door path and since $\mathcal{G}$ is amenable relative to $(\mathbf{X},\mathbf{Y})$, $p(X',Y)$ is a non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}$. Then since $p(X',Y)$ also of definite status, it must be blocked by $\mathbf{Z}$ in $\mathcal{G}$. Since $p(X',Y)$ is m-connecting given $\mathbf{Z}\cup\mathbf{X}\setminus\{X\}$ and blocked by $\mathbf{Z}$, it follows that no definite non-collider on $p(X',Y)$ is in $\mathbf{Z}$ and there is at least one collider on $p(X',Y)$ that is in $\mathrm{An}(\mathbf{X}\setminus\{X\},\mathcal{G})\setminus\mathrm{An}(\mathbf{Z},\mathcal{G})$. Let C be the collider closest to Y on $p(X',Y)$ such that $C\in\mathrm{An}(\mathbf{X},\mathcal{G})\setminus\mathrm{An}(\mathbf{Z},\mathcal{G})$. Let q be of the form $C\to\cdots\to X''$, $X''\in\mathbf{X}$ be the shortest causal path from C to $\mathbf{X}$. Since $p(X',Y)$ is proper, it follows that $C\neq X''$.

Let D be the node closest to X'' on $-q$ such that D is also on $p(C,Y)$ (possibly $D=C$) and $r=-q(X'',D)\oplus p(D,Y)$. It is left to show that r is a proper definite status non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ that is m-connecting given $\mathbf{Z}$ in $\mathcal{G}$. Since $p(C,Y)$ does not contain a node in $\mathbf{X}$, it follows that $-q(X'',D)$ is of non-zero length. So r is a definite status non-causal path. Additionally, since q was chosen as the shortest path from C to $\mathbf{X}$, it follows that r is proper. Lastly, since both q and $p(C,Y)$ are m-connecting given $\mathbf{Z}$ and $D\notin\mathbf{Z}$ and D is a definite non-collider on r, it follows that r is m-connecting given Z. ■

Corollary 53 *Let X and Y be distinct nodes in a* DAG, CPDAG, MAG *or* PAG *$\mathcal{G}$ and let $\mathcal{R}_{\underline{X}}$ be a graph as defined in Definition 46. If there exists a generalized back-door set relative to (X,Y) in $\mathcal{G}$, then $DSEP(X,Y,\mathcal{R}_{\underline{X}})\subseteq\mathrm{Adjust}(X,Y,\mathcal{G})\setminus\mathrm{PossDe}(X,\mathcal{G})$.*

Proof of Corollary 53. Since there exists a generalized back-door set relative to (X,Y) in $\mathcal{G}$, by Theorem 47 $DSEP(X,Y,\mathcal{R}_{\underline{X}})\subseteq\mathrm{PossAn}(X\cup Y,\mathcal{G})\setminus(\mathrm{PossDe}(X,\mathcal{G})\cup Y)$ is a generalized back-door set relative to (X,Y) in $\mathcal{G}$. Additionally, by Corollary 24 $\mathrm{Adjust}(X,Y,\mathcal{G})\setminus\mathrm{PossDe}(X,\mathcal{G})$ is also generalized back-door set relative to (X,Y) in $\mathcal{G}$ and $\mathrm{Adjust}(X,Y,\mathcal{G})\setminus\mathrm{PossDe}(X,\mathcal{G})=\mathrm{PossAn}(X\cup Y,\mathcal{G})\setminus(\mathrm{PossDe}(X,\mathcal{G})\cup Y)$. ■

Proof of Lemma 25. Let there be a p from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}$ for which (i)−(iv) hold. Then p is a proper definite status non-causal path that is m-connecting given $\text{Adjust}(\mathbf{X},\mathbf{Y},\mathcal{G}) \setminus \text{PossDe}(\mathbf{X},\mathcal{G})$. Thus, $\text{Adjust}(\mathbf{X},\mathbf{Y},\mathcal{G}) \setminus \text{PossDe}(\mathbf{X},\mathcal{G})$ violates the blocking condition relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{G}$ and Theorem 14 implies that there is no adjustment set $\mathbf{Z}$ relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{G}$ such that $\mathbf{Z} \cap \text{PossDe}(\mathbf{X},\mathcal{G}) = \emptyset$.

Conversely, assume there is no adjustment set $\mathbf{Z}$ relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{G}$ such that $\mathbf{Z} \cap \text{PossDe}(\mathbf{X},\mathcal{G}) = \emptyset$. Since there exists an adjustment set relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{G}$, $\mathcal{G}$ is amenable relative to $(\mathbf{X},\mathbf{Y})$. Then Theorem 14 implies that $\text{Adjust}(\mathbf{X},\mathbf{Y},\mathcal{G}) \setminus \text{PossDe}(\mathbf{X},\mathcal{G})$ violates the blocking condition relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{G}$. Hence, there is a proper definite status non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}$ that is m-connecting given $\text{Adjust}(\mathbf{X},\mathbf{Y},\mathcal{G}) \setminus \text{PossDe}(\mathbf{X},\mathcal{G})$. Then we can use Lemma 16 with $\mathbf{I} = \text{PossDe}(\mathbf{X},\mathcal{G})$ to choose a shortest path p from $X \in \mathbf{X}$ to $Y \in \mathbf{Y}$ for which (i)−(iv) in Lemma 16 hold.

We now show that (i)−(iv) in Lemma 25 hold for p.

(i) Follows immediately from (i) in Lemma 16.

(ii) Any definite non-collider on p is in $\text{PossDe}(\mathbf{X},\mathcal{G})$ ((iii) in Lemma 16). Since there is an adjustment set relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{G}$, it follows from Corollary 15 that $\text{Adjust}(\mathbf{X},\mathbf{Y},\mathcal{G})$ satisfies the generalized adjustment criterion. Thus, p is blocked by $\text{Adjust}(\mathbf{X},\mathbf{Y},\mathcal{G})$ and m-connecting given $\text{Adjust}(\mathbf{X},\mathbf{Y},\mathcal{G}) \setminus \text{PossDe}(\mathbf{X},\mathcal{G})$. This implies that at least one definite non-collider on p must be in $\text{Adjust}(\mathbf{X},\mathbf{Y},\mathcal{G}) \cap \text{PossDe}(\mathbf{X},\mathcal{G})$.

For the remainder of the proof let V be the definite non-collider that is closest to Y on p, among all definite non-colliders on p in $\text{Adjust}(\mathbf{X},\mathbf{Y},\mathcal{G}) \cap \text{PossDe}(\mathbf{X},\mathcal{G})$.

(iii) By (ii) in Lemma 16, all colliders on p are in $\text{Adjust}(\mathbf{X},\mathbf{Y},\mathcal{G}) \setminus \text{PossDe}(\mathbf{X},\mathcal{G})$. It is left to show that all definite non-colliders on $p(V,Y)$ are in $\text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G})$.

All definite non-colliders on p are possible ancestors of an endpoint node of p or a collider on p. Hence, all definite non-colliders on p are in $\text{PossAn}(\mathbf{X}\cup\mathbf{Y},\mathcal{G})$. By (iii) in Lemma 16, all definite non-colliders are also in $\text{PossDe}(\mathbf{X},\mathcal{G})$. Hence, all definite non-colliders on p are in $\text{PossAn}(\mathbf{X}\cup\mathbf{Y},\mathcal{G}) \cap \text{PossDe}(\mathbf{X},\mathcal{G})$. Additionally, by the choice of V, no definite non-collider on $p(V,Y)$ is in $\text{Adjust}(\mathbf{X},\mathbf{Y},\mathcal{G}) \cap \text{PossDe}(\mathbf{X},\mathcal{G})$. Since $\text{Adjust}(\mathbf{X},\mathbf{Y},\mathcal{G}) \cap \text{PossDe}(\mathbf{X},\mathcal{G}) = (\text{PossAn}(\mathbf{X}\cup\mathbf{Y},\mathcal{G}) \cap \text{PossDe}(\mathbf{X},\mathcal{G})) \setminus (\mathbf{X}\cup\mathbf{Y}\cup\text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G}))$, it follows that all definite non-colliders on $p(V,Y)$ are in $\mathbf{X}\cup\mathbf{Y}\cup\text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G})$. It is only left to show that no definite non-collider on $p(V,Y)$ is in $\mathbf{X}$ or $\mathbf{Y} \setminus \text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G})$. Since p is proper, no definite non-collider on p is in $\mathbf{X}$. Also, no non-endpoint node C on p is in $\mathbf{Y} \setminus \text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G})$, otherwise $p(X,C)$ is a non-causal path and we could have chosen a shorter p. Hence, any definite non-collider on $p(V,Y)$ is in $\text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G})$.

(iv) Let V_2 be the node closest to Y on p such that $p(V,V_2)$ is a possibly directed path from V to V_2 (possibly of zero length). We will show that $V_2 = V$. Note that V_2 is either Y, V or a collider on p. Since $V_2 \in \text{PossDe}(V,\mathcal{G})$ and $\text{PossDe}(V,\mathcal{G}) \subseteq \text{PossDe}(\mathbf{X},\mathcal{G})$, by (iii) V_2 cannot be a collider on p. Hence, V_2 is either Y or V.

Let V_1 be the node closest to X on p such that $-p(V,V_1)$ is an possibly directed path from V to V_1 (possibly of zero length). We will show that $V_1 = X$. Note that V_1 is either X, V or a collider on p. Using the same reasoning as for V_2, V_1 cannot be a collider on p. So V_1 is either X or V. As V is a definite non-collider on p, either $V_1 \neq V$ or $V_2 \neq V$.

Suppose that $V_2 \neq V$. Then $V_2 = Y$ and $p(V,Y)$ is a possibly directed path from V to Y. Since $V \in \text{PossDe}(\mathbf{X},\mathcal{G})$, let q be a proper possibly directed path from $X' \in \mathbf{X}$ (possibly $X = X'$) to V in $\mathcal{G}$. Let W' (possibly $W' = V$) be the node closest to X' on q that is also on

$p(V,Y)$. Then $q(X',W') \oplus p(W',Y)$ is a proper possibly directed path from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}$, so $W' \in \text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G})$. Since $V \in \text{PossDe}(W',\mathcal{G})$, it follows that $V \in \text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G})$. This contradicts (ii).

Hence, $V_2 = V$. Then $V_1 = X$ and p is of the form $X \cdots \leftarrow V \leftarrow\!\bullet W \ldots Y$, possibly $W = Y$ is in $\mathcal{G}$. Since p is of definite status, $p(X,V)$ must be of the form $X \leftarrow \cdots \leftarrow V$. If $W \neq Y$, then if W is a definite non-collider on $p(V,Y)$, (iii) implies that $W \in \text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G})$. Hence, $V \leftrightarrow W$ is in $\mathcal{G}$, otherwise $V \in \text{PossDe}(W,\mathcal{G}) \subseteq \text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G})$, which contradicts (ii). Else, W is a collider on $p(V,Y)$, so $V \leftrightarrow W$ must be on p. ■

Proof of Corollary 27. (i) Let $\mathbf{X}' = \mathbf{X} \cap \text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G})$. Since $\mathbf{X}' \neq \emptyset$, there exists a proper possibly directed path $p = \langle X = V_1, \ldots, V_k = Y\rangle, k > 1$, from $X \in \mathbf{X}$ to $Y \in \mathbf{Y}$ in $\mathcal{G}$ such that for some V_i, $i \in \{2,\ldots,k\}$, $\text{PossDe}(V_i,\mathcal{G}) \cap \mathbf{X}' \neq \emptyset$. Let $V_j, j \in \{2,\ldots,k\}$, be the node closest to Y on p such that $\text{PossDe}(V_j,\mathcal{G}) \cap \mathbf{X}' \neq \emptyset$. Let q be a shortest possibly directed path from V_j to a node X' in $\mathbf{X}'$. Since p was chosen to be proper with respect to $\mathbf{X}$, we know that $V_j \neq X'$.

By the choice of V_j on p, no other node from $p(V_j,Y)$ is on q. By Lemma 41, let $\overline{p(V_j,Y)}$ be a subsequence of $p(V_j,Y)$ that forms a possibly directed definite status path from V_j to Y. Now, we can concatenate $-q$ and $\overline{p(V_j,Y)}$ to form the path $r = (-q) \oplus \overline{p(V_j,Y)}$. We will show that r is a proper definite status non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ that does not contain a collider and consists of nodes in $\text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G})$. This means that r satisfies condition (2) in Theorem 26, so there is no set that satisfies the generalized adjustment criterion relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{G}$. By completeness of the generalized adjustment criterion (Theorem 5), there is no adjustment set relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{G}$.

We first show that r is proper with respect to $\mathbf{X}$. Since p is proper with respect to $\mathbf{X}$ and since $V_j \neq X$, $\overline{p(V_j,Y)}$ does not contain a node in $\mathbf{X}$. Additionally, by choice of q, X' is the only node from $\mathbf{X}$ on q. Hence, r is proper with respect to $\mathbf{X}$.

Since q and $\overline{p(V_j,Y)}$ are possibly directed paths from V_j to X' and from V_j to Y, there is no collider on r. Additionally, since $V_j \in \text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G})$ and since $\text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G})$ is a descendral set in $\mathcal{G}$, all nodes on r are in $\text{Forb}(\mathbf{X},\mathbf{Y},\mathcal{G})$.

Next, we show that r is of definite status. First, $\overline{p(V_j,Y)}$ is of definite status. Since q is a shortest possibly directed path from V_j to X', it is of definite status by Lemma 41. Hence, if $V_j = Y$, then $r = (-q)$ is of definite status. If $V_j \neq Y$, it is left to show that V_j is of definite status on r.

We prove this by contradiction. Thus, suppose that V_j is not of definite status on r. Let $\langle A,V_j,B\rangle$ be a subpath of r, so that A is the node adjacent to V_j on q and B is the node adjacent to V_j on $\overline{p(V_j,Y)}$. Then $A\bullet\!\!\rightarrow V_j \circ\!\!-\!\!\bullet B$, $A\bullet\!\!-\!\!\circ V_j \leftarrow\!\!\bullet B$, or $A\bullet\!\!-\!\!\circ V_j \circ\!\!-\!\!\bullet B$ is in $\mathcal{G}$ and there is an edge $\langle A,B\rangle$ in $\mathcal{G}$. Since q and $\overline{p(V_j,Y)}$ are possibly directed paths from V_j to X' and from V_j to Y, $A\bullet\!\!\rightarrow V_j \circ\!\!-\!\!\bullet B$ and $A\bullet\!\!-\!\!\circ V_j \leftarrow\!\!\bullet B$ cannot be in $\mathcal{G}$. Hence, $A\bullet\!\!-\!\!\circ V_j \circ\!\!-\!\!\bullet B$ is a subpath of r.

Since $A\bullet\!\!-\!\!\circ V_j \circ\!\!-\!\!\bullet B$ is in $\mathcal{G}$, neither $A\bullet\!\!\rightarrow B$ nor $A \leftarrow\!\!\bullet B$ can be in $\mathcal{G}$ (Lemma 36). This implies that $A \circ\!\!-\!\!\circ B$ is in $\mathcal{G}$ and so, $\langle B,A\rangle \oplus q(A,X')$ is a possibly directed path from B to X', which contradicts the choice of V_j (since B is closer to Y on p).

It is only left to show that r is a non-causal path from X' to Y. We again use a proof by contradiction. Thus, suppose that r is a possibly directed path from X' to Y. Then, since $r(X',V_j) = (-q)$ and q are both possibly directed paths in $\mathcal{G}$, r must start with a

non-directed edge. Hence, r is a proper possibly directed path from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}$ that starts with a non-directed edge, which contradicts that $\mathcal{G}$ is amenable relative to $(\mathbf{X}, \mathbf{Y})$.

(ii) Let $\mathcal{G}$ be a DAG or CPDAG that is amenable relative to $(\mathbf{X}, \mathbf{Y})$ and let $\mathbf{Y} \subseteq \text{PossDe}(\mathbf{X}, \mathcal{G})$. By (i), it follows that if $\mathbf{X} \cap \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G}) \neq \emptyset$, then there is no adjustment set relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$. Hence, we only prove the converse statement.

If there is no adjustment set relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$, then by the soundness of the generalized adjustment criterion (Theorem 5) there is no set that satisfies the generalized adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$. Since $\mathcal{G}$ is amenable relative to $(\mathbf{X}, \mathbf{Y})$, this means that there is a path p from $X \in \mathbf{X}$ to $Y \in \mathbf{Y}$ in $\mathcal{G}$ that satisfies condition (2) in Theorem 26 i.e., p is a proper definite status non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{G}$ such that every collider on p is in $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$ and every definite non-collider on p is in $\text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$.

We first note that, since $\mathbf{Y} \subseteq \text{PossDe}(\mathbf{X}, \mathcal{G})$, $\mathbf{Y} \subseteq \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$. We now show, by contradiction, that there is no collider on p. Thus, suppose that there is a collider C on p. Then C must be either a descendant of a non-collider on p or a descendant of both X and Y. Since $C \in \text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$, $C \notin \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$. Moreover, every non-collider on p is in $\text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$ and since $\text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$ is a descendral set, C cannot be a descendant of a non-collider on p. Then p must be of the form $X \to C \leftarrow Y$. Since $Y \in \text{PossDe}(\mathbf{X}, \mathcal{G})$ and $C \in \text{De}(Y, \mathcal{G})$, this contradicts that $C \notin \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$.

Hence, p does not contain a collider. Additionally, p is a non-causal path from X to Y. This implies that there is a node A on p, $A \neq X$, such that $-p(A, X)$ is a directed path from A to X. Since A is either a non-collider on p, or $A = Y$, $A \in \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$. Hence, $X \in \mathbf{X} \cap \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$. ■

Proof of Corollary 28. Since there is no directed path from one node in $\mathbf{X}$ to another node in $\mathbf{X}$ in $\mathcal{D}$, it is easy to see that a path of the form $X \leftarrow V \dots Y$, where $X \in \mathbf{X}, Y \in \mathbf{Y}$ and $V \in \text{De}(\mathbf{X}, \mathcal{D})$ cannot occur in $\mathcal{D}$. Then there can be no path p from $\mathbf{X}$ to $\mathbf{Y}$ that satisfies condition (i) in Lemma 21 (condition (iv) in Lemma 25) in $\mathcal{D}$. Hence, conditions (3) and (4) in Theorem 26 are violated relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$. Then by (i) and (iii) in Theorem 26 it follows that there exists a set that satisfies the generalized adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$ if and only if there exists a back-door set relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$. ■

Proof of Corollary 29. It is enough to prove that if there exists an adjustment set relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$ and the assumptions of the corollary hold, then there is no proper definite status non-causal path p_1 that satisfies (i)−(iv) in Lemma 25.

If $\mathcal{G}$ contains no possibly directed path $p = \langle V_1, \dots, V_k \rangle$, with $k \geq 3, \{V_1, V_k\} \subseteq \mathbf{X}$ and $\{V_2, \dots, V_{k-1}\} \cap \mathbf{X} = \emptyset$, it follows that there cannot be a node $V \in \text{PossDe}(\mathbf{X}, \mathcal{G})$ and the path $X \leftarrow \cdots \leftarrow V$ in $\mathcal{G}$. So there cannot be a proper definite status non-causal path p_1 that satisfies (i)−(iv) in Lemma 25.

Next, suppose that $\mathcal{G}$ is a DAG or CPDAG and that $\mathbf{Y} \subseteq \text{PossDe}(\mathbf{X}, \mathcal{G})$ and that there exists an adjustment set relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$. By Corollary 15, it follows that $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$ then satisfies the blocking condition relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$. Suppose for a contradiction that there is a proper definite status non-causal path p_1 that satisfies (i)−(iv) in Lemma 25. Then p_1 is of the form $X \leftarrow \cdots \leftarrow V \leftarrow Y$. By assumption $Y \in \text{PossDe}(\mathbf{X}, \mathcal{G})$, so it follows that $Y \in \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$. By the definition of the forbidden set, every other node on p is also in $\text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$. But then $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$ does cannot block p. This is

in contradiction with $\text{Adjust}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$ satisfying the blocking condition relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$. ■

Appendix E. Adjustment Criterion for DAGs

In this section we provide the soundness and completeness proof for the adjustment criterion from Shpitser (2012); van der Zander et al. (2014) (see Definition 55). The main result is given in Theorem 56.

This section can be read independently from the rest of the paper. Since we restrict our proof to DAGs, we first define adjustment sets (see Definition 54) and the adjustment criterion (see Definition 55) in DAGs. Thus, Definition 54 and Definition 55 are special cases of Definition 1 and Definition 4 for DAGs.

Definition 54 *(**Adjustment set**; Pearl, 2009, Chapter 3.3.1) Let $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$ be pairwise disjoint node sets in a causal* DAG *$\mathcal{D}$. Then $\mathbf{Z}$ is an adjustment set relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$ if for any density f consistent with $\mathcal{D}$:*

$$f(\mathbf{y} \mid do(\mathbf{x})) = \begin{cases} f(\mathbf{y} \mid \mathbf{x}) & \text{if } \mathbf{Z} = \emptyset, \\ \int_{\mathbf{z}} f(\mathbf{y} \mid \mathbf{x}, \mathbf{z}) f(\mathbf{z}) d\mathbf{z} & \text{otherwise.} \end{cases}$$

Definition 55 *(**Adjustment criterion**; cf. Shpitser, 2012, van der Zander et al., 2014) Let $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$ be pairwise disjoint node sets in a* DAG *$\mathcal{D}$. Let $\text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D})$ denote the set of all descendants in $\mathcal{D}$ of any $W \notin \mathbf{X}$ which lies on a proper causal path from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{D}$. Then $\mathbf{Z}$ satisfies the adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$ if the following two conditions hold:*

*(**Forbidden set**) $\mathbf{Z} \cap \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D}) = \emptyset$, and*

*(**Blocking**) all proper non-causal paths from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{D}$ are blocked by $\mathbf{Z}$.*

Definition 55 was introduced in van der Zander et al. (2014) and differs from the definition of the adjustment criterion in Shpitser (2012) in that it uses $\mathcal{D}$ in the forbidden set condition, as opposed to $\mathcal{D}_{\overline{\mathbf{X}}}$, where $\mathcal{D}_{\overline{\mathbf{X}}}$ is the graph obtained by removing all edges into $\mathbf{X}$ from $\mathcal{D}$. These two formulations of the adjustment criterion are equivalent (Remark 4.3 in van der Zander et al., 2014). We now give the main result in Theorem 56, which follows directly from Theorem 57 and Theorem 58. To prove Theorem 58 we rely on Lemma 59 and Lemma 60, which are given later in this section.

Theorem 56 *Let $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$ be pairwise disjoint node sets in a causal* DAG *$\mathcal{D} = (\mathbf{V}, \mathbf{E})$. Then $\mathbf{Z}$ satisfies the adjustment criterion (see Definition 55) if and only if $\mathbf{Z}$ is an adjustment set (see Definition 54).*

Theorem 57 *(**Completeness of the adjustment criterion for** DAG**s**) Let $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$ be pairwise disjoint node sets in a causal* DAG *$\mathcal{D}$. If $\mathbf{Z}$ does not satisfy the adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$, then there exists a density f consistent with $\mathcal{D}$ such that $f(\mathbf{y} \mid do(\mathbf{x})) \neq \int_{\mathbf{z}} f(\mathbf{y} \mid \mathbf{x}, \mathbf{z}) f(\mathbf{z}) d\mathbf{z}$.*

Proof of Theorem 57. Suppose that $\mathbf{Z}$ does not satisfy the adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D} = (\mathbf{V}, \mathbf{E})$. It suffices to show that there is a density consistent with $\mathcal{D}$ such that $E[Y \mid do(\mathbf{X} = \mathbf{1})] \neq \int_z E[Y \mid \mathbf{X} = \mathbf{1}, \mathbf{z}] f(\mathbf{z}) d\mathbf{z}$ for at least one node $Y \in \mathbf{Y}$.

We consider multivariate Gaussian densities with mean vector zero, constructed using linear structural equation models (SEMs) with Gaussian noise. In particular, we let each random variable $A \in \mathbf{V}$ be a linear combination of its parents in $\mathcal{D}$ and a designated Gaussian noise variable ϵ_A with zero mean and a fixed variance. We also assume that the Gaussian noise variables $\{\epsilon_A : A \in \mathbf{V}\}$, are mutually independent. Thus, this model can be parameterized using one number per node (the residual variance) and one number per edge (the edge coefficient).

Since $\mathbf{Z}$ does not satisfy the adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$, $\mathbf{Z}$ violates the forbidden set condition or the blocking condition.

1 If $\mathbf{Z}$ violates the forbidden set condition, then there is a proper causal path $\langle X, V_1, \ldots, V_k = Y \rangle$, $k \geq 1$, from $X \in \mathbf{X}$ to $Y \in \mathbf{Y}$ in $\mathcal{D}$ and a node $Z \in \mathbf{Z}$ such that

   (a) $Z = V_i$, for some $i \in \{1, \ldots k-1\}$, or

   (b) $Z \in \mathrm{De}(\mathbf{Y}, \mathcal{D})$, or

   (c) $Z \in \mathrm{De}(V_i, \mathcal{D}) \setminus \{V_1, \ldots, V_{k-1}\}$ for some $i \in \{1, \ldots k-1\}$.

2 If $\mathbf{Z}$ violates the blocking condition, then there exists a proper non-causal path from $X \in \mathbf{X}$ to $Y \in \mathbf{Y}$ that is d-connecting given $\mathbf{Z}$ in $\mathcal{D}$ such that:

   (a) the path does not contain any colliders, or

   (b) the path contains at least one collider.

We now discuss these cases systematically.

(i) Suppose there is a path p from $X \in \mathbf{X}$ to $Y \in \mathbf{Y}$ that satisfies 2(a) in $\mathcal{D}$. Since p is a proper non-causal path that does not contain colliders, p starts with an edge into X, that is, p is of the form $X \leftarrow \ldots Y$.

We define our SEM so that all edge coefficients except for the ones on p are 0, and all edge coefficients on p are in $(0, 1)$ and small enough so that we can choose the residual variances so that the variance of every random variable in $\mathbf{V}$ is 1. Then the density f generated by this SEM is consistent with the causal DAG $\mathcal{D}$. Moreover, f is also consistent with the causal DAG $\mathcal{D}'$ that is obtained from $\mathcal{D}$ by removing all edges except for the ones on p.

Since $Y \perp_d \mathbf{X}$ in $\mathcal{D}'_{\underline{\mathbf{X}}}$, we use Rule 3 of the do-calculus (see Equation 6 in Appendix A), with $\mathbf{X}' = \emptyset, \mathbf{W}' = \emptyset$, $\mathbf{Z}' = \mathbf{X}$ and $\mathbf{Y}' = \{Y\}$, so that $E[Y \mid do(\mathbf{X} = \mathbf{1})] = E[Y] = 0$.

Since p is proper, no node in $\mathbf{X} \setminus \{X\}$ is on p. Additionally, since p is d-connecting given $\mathbf{Z}$ and p does not contain colliders, no node in $\mathbf{Z}$ is on p. This implies $Y \perp_d \mathbf{Z} \cup \mathbf{X} \setminus \{X\}$ in $\mathcal{D}'$. Furthermore, $Y \perp_d \mathbf{Z} \cup \mathbf{X} \mid \mathbf{S}$ in $\mathcal{D}'$ for any subset $\mathbf{S}$ of the remaining nodes. In particular, we have $Y \perp_d \mathbf{Z} \cup \mathbf{X} \setminus \{X\} \mid X$ in $\mathcal{D}'$, so that $\int_{\mathbf{z}} E[Y \mid \mathbf{X} = \mathbf{1}, \mathbf{z}] f(\mathbf{z}) d\mathbf{z} = E[Y \mid X = 1]$. By Theorem 32, $E[Y \mid X = 1] = \mathrm{Cov}(X, Y)$. By Wright's rule (Theorem 31), $\mathrm{Cov}(X, Y) = a$, where a is the product of all edge coefficients on p. Since $\int_{\mathbf{z}} E[Y \mid \mathbf{X} = \mathbf{1}, \mathbf{z}] f(\mathbf{z}) d\mathbf{z} = a \neq 0$, this case is completed.

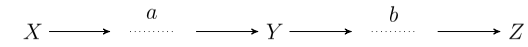


Figure 10: An example of path $p \oplus q$ in $\mathcal{D}$ corresponding to (iii), where $Z \in \mathbf{Z}$.

(ii) Suppose no path satisfies 2(a) in $\mathcal{D}$, but there is a path p from $X \in \mathbf{X}$ to $Y \in \mathbf{Y}$ that satisfies 1(a) in $\mathcal{D}$. Let $\tilde{\mathbf{Z}}$ be the set of all nodes in $\mathbf{Z}$ that are on p and let $Z \in \tilde{\mathbf{Z}}$.

We define our SEM so that all edge coefficients except the ones on p are 0, and all edge coefficients on p are in $(0, 1)$ and small enough so that we can choose the residual variances such that the variance of every random variable in $\mathbf{V}$ is 1. Then the density f generated by this SEM is consistent with the causal DAG $\mathcal{D}$, and also with the causal DAG $\mathcal{D}'$ that is obtained from $\mathcal{D}$ by removing all edges except for the ones on p.

Since no node from $\mathbf{X} \setminus \{X\}$ is on p, it follows that $Y \perp_d \mathbf{X} \setminus \{X\}$ in $\mathcal{D}'$. Furthermore, $Y \perp_d \mathbf{X} \setminus \{X\} \mid X$ in $\mathcal{D}'_{\underline{X}}$. We use Rule 3 of the do-calculus, with $\mathbf{X}' = \{X\}, \mathbf{W}' = \emptyset$, $\mathbf{Z}' = \mathbf{X} \setminus \{X\}$ and $\mathbf{Y}' = \{Y\}$, so that $E[Y \mid do(\mathbf{X} = \mathbf{1})] = E[Y \mid do(X = 1)]$. This means that we have reduced a joint intervention to a single intervention.

Additionally, since $Y \perp_d X$ in $\mathcal{D}'_{\underline{X}}$, we use Rule 2 of the do-calculus, with $\mathbf{X}' = \emptyset, \mathbf{W}' = \emptyset$, $\mathbf{Z}' = X$ and $\mathbf{Y}' = \{Y\}$, so that $E[Y \mid do(X = 1)] = E[Y \mid X = 1]$. By Theorem 32, $E[Y \mid X = 1] = \mathrm{Cov}(X, Y)$. Let a be the product of all edge coefficients on p. By Wright's rule (Theorem 31), we have that $\mathrm{Cov}(X, Y) = a \neq 0$.

We complete this case by showing that $\int_{\mathbf{z}} E[Y \mid \mathbf{X} = \mathbf{1}, \mathbf{z}] f(\mathbf{z}) d\mathbf{z} = 0$. Since p is proper, no node in $\mathbf{X} \setminus \{X\}$ is on p. Additionally, by our choice of $\tilde{\mathbf{Z}}$, no node in $\mathbf{Z} \setminus \tilde{\mathbf{Z}}$ is on p. Thus, $Y \perp_d \mathbf{X} \cup (\mathbf{Z} \setminus \tilde{\mathbf{Z}}) \mid \tilde{\mathbf{Z}}$ in $\mathcal{D}'$. Then $\int_{\mathbf{z}} E[Y \mid \mathbf{X} = \mathbf{1}, \mathbf{z}] f(\mathbf{z}) d\mathbf{z} = \int_{\tilde{\mathbf{z}}} E[Y \mid \tilde{\mathbf{z}}] f(\tilde{\mathbf{z}}) d\tilde{\mathbf{z}} = E[Y] = 0$.

(iii) Suppose no path satisfies 2(a) or 1(a), but there is a path p from $X \in \mathbf{X}$ to $Y \in \mathbf{Y}$ that satisfies 1(b) in $\mathcal{D}$. Choose $Z \in \mathbf{Z}$ such that the causal path q from Y to Z is a shortest causal path from Y to a node in $\mathbf{Z}$. Then Y is the only node that is on both p and q, otherwise there is a cycle in $\mathcal{D}$. Hence, $p \oplus q$ is a causal path from X to Z that contains Y (see Figure 10).

We define our SEM so that all edge coefficients except the ones on $p \oplus q$ are 0, and all edge coefficients on $p \oplus q$ are in $(0, 1)$ and small enough so that we can choose the residual variances such that the variance of every random variable in $\mathbf{V}$ is 1. Then the density f generated by this SEM is consistent with the causal DAG $\mathcal{D}$, as well as with the causal DAG $\mathcal{D}'$ that is obtained from $\mathcal{D}$ by removing all edges except the ones on $p \oplus q$.

Since there is no path that satisfies 2(a) in $\mathcal{D}$, no node from $\mathbf{X}$ is on q. Additionally, since p is proper, no node in $\mathbf{X} \setminus \{X\}$ is on $p \oplus q$. Thus, $Y \perp_d \mathbf{X} \setminus \{X\} \mid X$ in $\mathcal{D}'$. Furthermore, $Y \perp_d \mathbf{X} \setminus \{X\} \mid X$ in $\mathcal{D}'_{\underline{X}}$. Hence, we use Rule 3 of the do-calculus, with $\mathbf{X}' = \{X\}, \mathbf{W}' = \emptyset$, $\mathbf{Z}' = \mathbf{X} \setminus \{X\}$ and $\mathbf{Y}' = \{Y\}$, so that $E[Y \mid do(\mathbf{X} = \mathbf{1})] = E[Y \mid do(X = 1)]$.

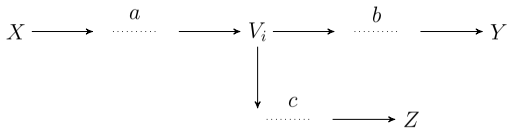


Figure 11: An example of paths p and q_i in $\mathcal{D}$ corresponding to (iv), where $Z \in \mathbf{Z}$.

Moreover, $Y \perp_d X$ in $\mathcal{D}'_{\underline{X}}$. Hence, we use Rule 2 of the do-calculus, with $\mathbf{X}' = \emptyset, \mathbf{W}' = \emptyset$, $\mathbf{Z}' = X$ and $\mathbf{Y}' = \{Y\}$, so that $E[Y \mid do(X = 1)] = E[Y \mid X = 1]$. Lastly, using Theorem 32 and Wright's rule (Theorem 31), we have that $E[Y \mid X = 1] = \mathrm{Cov}(X, Y) = a$, where a is the product of all edge coefficients on p .

Next, we show that $\int_{\mathbf{z}} E[Y \mid \mathbf{X} = \mathbf{1}, \mathbf{z}] f(\mathbf{z}) d\mathbf{z} \neq a$. Since no path satisfies 1(a), no node from $\mathbf{Z}$ is on p. Furthermore, by the choice of q, no node from $\mathbf{Z} \setminus \{Z\}$ is on q. Hence, Z is the only node from $\mathbf{Z}$ that is on $p \oplus q$. From the above, we also know that X is the only node from $\mathbf{X}$ that is on $p \oplus q$. Hence, $Y \perp_d (\mathbf{X} \cup \mathbf{Z}) \setminus \{X, Z\} \mid \{X, Z\}$ in $\mathcal{D}'$ and we have that $\int_{\mathbf{z}} E[Y \mid \mathbf{X} = \mathbf{1}, \mathbf{z}] f(\mathbf{z}) d\mathbf{z} = \int_z E[Y \mid X = 1, z] f(z) dz$.

Let b be the product of all edge coefficients on q. By Wright's rule (Theorem 31), we have that $\mathrm{Cov}(X, Y) = a, \mathrm{Cov}(Y, Z) = b$ and $\mathrm{Cov}(X, Z) = ab$. Now, we use Theorem 32 to calculate $E[Y \mid X = 1, z]$:

$$E[Y \mid X = 1, z] = \begin{bmatrix} a & b \end{bmatrix} \begin{bmatrix} 1 & ab \\ ab & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ z \end{bmatrix} = \frac{a(1-b^2)}{1-(ab)^2} + \frac{b(1-a^2)}{1-(ab)^2} z.$$

$$\int_z E[Y \mid X = 1, z] f(z) dz = a \frac{1-b^2}{1-(ab)^2} + \frac{b(1-a^2)}{1-(ab)^2} E[Z] = a \frac{1-b^2}{1-(ab)^2}. \tag{7}$$

Since $0 < a < 1$ and $0 < b < 1$, right-hand side of Equation (7) is strictly smaller than $a = E[Y \mid do(\mathbf{X} = \mathbf{1})]$.

(iv) Suppose no path satisfies 1(a), 1(b), or 2(a), but there is a path p from $X \in \mathbf{X}$ to $Y \in \mathbf{Y}$ that satisfies 1(c) in $\mathcal{D}$. Let V_i, $i \in \{1, \ldots, k-1\}$, be a node on p that has a shortest causal path to a node in $\mathbf{Z}$. Let q_i be such a shortest causal path from V_i to $\mathbf{Z}$. Then no node except V_i is on both p and q_i, otherwise we would have chosen a different V_i (see Figure 11).

We define our SEM so that all edge coefficients which are not on p or q_i are 0, and all edge coefficients on p and q_i are in $(0, 1)$ and small enough so that we can choose the residual variances so that the variance of every random variable in $\mathbf{V}$ is 1. Then the density f generated by this SEM is consistent with the causal DAG $\mathcal{D}$, as well as with the causal DAG $\mathcal{D}'$ that is obtained from $\mathcal{D}$ by removing all edges except for the ones on p and q_i.

Since there is no path that satisfies 2(a), no node from $\mathbf{X}$ is on q_i. Additionally, since p is proper, no node from $\mathbf{X} \setminus \{X\}$ is on p. Thus, $Y \perp_d \mathbf{X} \setminus \{X\} \mid X$ in $\mathcal{D}'_{\underline{X}}$ and we

use Rule 3 of the do-calculus, with $\mathbf{X}' = \{X\}, \mathbf{W}' = \emptyset$, $\mathbf{Z}' = \mathbf{X} \setminus \{X\}$ and $\mathbf{Y}' = \{Y\}$, so that $E[Y \mid do(\mathbf{X} = \mathbf{1})] = E[Y \mid do(X = 1)]$.

Additionally, $Y \perp_d X$ in $\mathcal{D}'_{\underline{X}}$. Hence, we use Rule 2 of the do-calculus with $\mathbf{X}' = \emptyset, \mathbf{W}' = \emptyset$, $\mathbf{Z}' = \{X\}$ and $\mathbf{Y}' = \{Y\}$, so that $E[Y \mid do(X = 1)] = E[Y \mid X = 1]$. Let a be the product of all edge coefficients on $p(X, V_i)$ and let b be the product of all edge coefficients on $p(V_i, Y)$. Then, using Theorem 32 and Wright's rule (Theorem 31), we have that $E[Y \mid X = 1] = \mathrm{Cov}(X, Y) = ab$.

Since there is no path that satisfies 1(a), no node from $\mathbf{Z}$ is on p. By the choice of q_i, no node from $\mathbf{Z} \setminus \{Z\}$ is on q_i. Hence, no node from $(\mathbf{X} \cup \mathbf{Z}) \setminus \{X, Z\}$ is on p nor q_i. Then $Y \perp_d (\mathbf{X} \cup \mathbf{Z}) \setminus \{X, Z\} \mid \{X, Z\}$ in $\mathcal{D}'$ and it follows that $\int_{\mathbf{z}} E[Y \mid \mathbf{X} = \mathbf{1}, \mathbf{z}] f(\mathbf{z}) d\mathbf{z} = \int_z E[Y \mid X = 1, z] f(z) dz$.

Let c be the product of all edge coefficients on q_i. By Wright's rule (Theorem 31), we have that $\mathrm{Cov}(X, Y) = ab, \mathrm{Cov}(Y, Z) = bc$ and $\mathrm{Cov}(X, Z) = ac$. We can now use Theorem 32 to calculate $E[Y \mid X = 1, z]$:

$$E[Y \mid X = 1, z] = \begin{bmatrix} ab & bc \end{bmatrix} \begin{bmatrix} 1 & ac \\ ac & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ z \end{bmatrix} = \frac{ab(1-c^2)}{1-(ac)^2} + \frac{bc(1-a^2)}{1-(ac)^2} z.$$

Hence,

$$\int_z E[Y \mid X = 1, z] f(z) dz = \frac{ab(1-c^2)}{1-(ac)^2} + \frac{bc(1-a^2)}{1-(ac)^2} E[Z] = ab \frac{1-c^2}{1-(ac)^2}. \tag{8}$$

Since $0 < a < 1$, $0 < b < 1$ and $0 < c < 1$, right-hand side of Equation (8) is strictly smaller than $ab = E[Y \mid do(\mathbf{X} = \mathbf{1})]$.

(v) Suppose there is no path that satisfies 1(a), 1(b), 1(c), or 2(a), but there is a path that satisfies 2(b) in $\mathcal{D}$. Let p be such a path from $X \in \mathbf{X}$ to $Y \in \mathbf{Y}$ in $\mathcal{D}$ that contains the smallest number of colliders among all such paths. Since no path satisfies 2(a), there is at least one collider on p. Let $C_1, \ldots, C_r$, $r \geq 1$, be all colliders on p ordered from the collider closest to X on p, which is C_1, to the collider closest to Y on p, which is C_r. Since p is d-connecting given $\mathbf{Z}$, we have $C_i \in \mathrm{An}(\mathbf{Z}, \mathcal{D})$ for all $i = 1, \ldots, r$. . For each $i = 1, \ldots, r$, let q_i be a shortest path (possibly of length zero) from C_i to $\mathbf{Z}$. Let $\tilde{\mathbf{Z}}$ be the collection of all nodes in $\mathbf{Z}$ that are enpoints of $q_1, \ldots, q_r$.

We define our SEM so that all edge coefficients which are not on $p, q_1, \ldots, q_r$ are 0, and all edge coefficients which are on $p, q_1, \ldots, q_r$ are in $(0, 1)$ and are small enough so that we can choose the residual variances such that the variance of every random variable in $\mathbf{V}$ is 1. Then the density f generated by this SEM is consistent with the causal DAG $\mathcal{D}$, as well as with the causal DAG $\mathcal{D}'$ which is obtained from $\mathcal{D}$ by removing all edges except the ones on p, $q_1, \ldots, q_r$. Moreover, f is a non-degenerate multivariate Gaussian density on $\mathbf{V}$.

Since there is no path that satisfies 2(a) in $\mathcal{D}$, no node from $\mathbf{X}$ is on q_r. Additionally, no node from $\mathbf{X}$ is on q_i, for any $i \in \{1, \ldots, r-1\}$, when $r > 1$, since otherwise there is a proper non-causal path p' from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{D}$ that is d-connecting given $\mathbf{Z}$ and

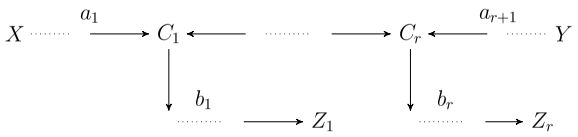


Figure 12: An example of paths p and $q_1, \ldots, q_r$ in $\mathcal{D}$ corresponding to (v).

that contains a smaller number of colliders than p. Hence, X is the only node from $\mathbf{X}$ that is on $p, q_1, \ldots, q_r$.

Since X is the only node from $\mathbf{X}$ that is on $p, q_1, \ldots, q_r$, $Y \perp_d \mathbf{X} \setminus \{X\}$ in $\mathcal{D}'$. By assumption there is at least one collider on p. Since there is no path that satisfies 1(b) in $\mathcal{D}$, Y is not on q_1. Additionally, Y is not on q_i, for any $i \in \{2, \ldots, r\}$, when $r > 1$, otherwise there is a proper non-causal path p' from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{D}$ that is d-connecting given $\mathbf{Z}$ and that contains a smaller number of colliders than p. Hence, $Y \perp_d \mathbf{X}$ in $\mathcal{D}'$. Furthermore, $Y \perp_d \mathbf{X}$ in $\mathcal{D}'_{\overline{X}}$, so we use Rule 3 of the do-calculus, with $\mathbf{X}' = \emptyset, \mathbf{W}' = \emptyset$, $\mathbf{Z}' = \mathbf{X}$ and $\mathbf{Y}' = \{Y\}$, so that $E[Y \mid do(\mathbf{X} = \mathbf{1})] = E[Y] = 0$.

By the choice of $p, q_1, \ldots, q_r$, $\tilde{\mathbf{Z}}$ are the only nodes from $\mathbf{Z}$ that are on $p, q_1, \ldots, q_r$. Then $\{X\} \cup \tilde{\mathbf{Z}}$ are the only nodes from $\mathbf{X} \cup \mathbf{Z}$ that are on $p, q_1, \ldots, q_r$. Hence, $Y \perp_d (\mathbf{X} \cup \mathbf{Z}) \setminus (\{X\} \cup \tilde{\mathbf{Z}})$ in $\mathcal{D}'$. Furthermore, $Y \perp_d (\mathbf{X} \cup \mathbf{Z}) \setminus (\{X\} \cup \tilde{\mathbf{Z}}) \mid \{X\} \cup \tilde{\mathbf{Z}}$ in $\mathcal{D}'$. Hence, $\int_{\mathbf{z}} E[Y \mid \mathbf{X} = \mathbf{1}, \mathbf{z}] f(\mathbf{z}) d\mathbf{z} = \int_{\tilde{\mathbf{z}}} E[Y \mid X = 1, \tilde{\mathbf{z}}] f(\tilde{\mathbf{z}}) d\tilde{\mathbf{z}}$.

We now show $\int_{\tilde{\mathbf{z}}} E[Y \mid X = 1, \tilde{\mathbf{z}}] f(\tilde{\mathbf{z}}) d\tilde{\mathbf{z}} \neq 0$. For this we need the covariance matrix of $(X, \tilde{\mathbf{Z}}, Y)^T$, which we will compute by applying Wright's rule to $\mathcal{D}'$ (see Figure 12). In order to do this, we first need to show that no node on q_i is on q_j, for all $i, j \in \{1, \ldots, r\}$ with $i \neq j$ and that no node on q_i except C_i is on p, for all $i \in \{1, \ldots, r\}$. From this it will follow that each path q_i ends in a different node in $\tilde{\mathbf{Z}}$. We label these nodes as $\tilde{\mathbf{Z}} = (Z_1, \ldots, Z_r)^T$ (see Figure 12).

We start by showing that no node on q_i, except C_i, is on p, for any $i \in \{1, \ldots, r\}$. Suppose for a contradiction that for some $i \in \{1, \ldots, r\}$ a node on q_i, other than C_i, is on p. Then q_i is at least of length 1, that is, $C_i \notin \mathbf{Z}$. Let D be the node closest to C_i on q_i that is also on p. Then D is either on $p(X, C_i)$ or on $p(C_i, Y)$.

Suppose first that D is on $p(X, C_i)$. Let $p' = p(X, D) \oplus (-q_i)(D, C_i) \oplus p(C_i, Y)$. From the above, we know that p' is a proper path from $\mathbf{X}$ to $\mathbf{Y}$. Since $(-q_i)(D, C_i)$ is of the form $D \leftarrow \cdots \leftarrow C_i$, p' is a non-causal path from X to Y. By construction p' will have fewer colliders than p. Hence, p' is a proper non-causal path from X to Y with fewer colliders than p, since C_i is a non-collider on p'. Hence, in order to reach a contradiction, we only need to show that p' is d-connecting given $\mathbf{Z}$.

Since $p, q_1, \ldots, q_r$ are d-connecting given $\mathbf{Z}$, we only need to discuss the collider/non-collider status of D and C_i on p'. Since $C_i \notin \mathbf{Z}$ and since C_i is a non-collider on p', we have that $p'(D, Y) = (-q_i)(D, C_i) \oplus p(C_i, Y)$ is d-connecting given $\mathbf{Z}$. Since D is on q_i, $D \in \mathrm{An}(\mathbf{Z}, \mathcal{D})$. So if D is a collider on p', p' is d-connecting given $\mathbf{Z}$. If D is a non-collider on p', then $(-p)(D, X)$ is out of D, so D is also a non-collider on p. Thus, in this case $D \notin \mathbf{Z}$ and hence, p' is d-connecting given $\mathbf{Z}$.

Next, suppose that D is on $p(C_i, Y)$. Let $p' = p(X, C_i) \oplus q_i(C_i, D) \oplus p(D, Y)$. From the above, we know that p' is a proper path from $\mathbf{X}$ to $\mathbf{Y}$. Since $D \in \mathrm{An}(\mathbf{Z}, \mathcal{D})$ and since there is no path that satisfies 1(a), 1(b), or 1(c), p' cannot be a causal path. Thus, p' is a proper non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ that by construction has fewer colliders than p, since C_i is a non-collider on p'. Hence, in order to reach a contradiction, we only need to show that p' is d-connecting given $\mathbf{Z}$.

Since $p, q_1, \ldots, q_r$ are d-connecting given $\mathbf{Z}$, we only need to discuss the collider/non-collider status of D and C_i on p'. Since $C_i \notin \mathbf{Z}$ and since C_i is a non-collider on p', we have that $p'(X, D) = p(X, C_i) \oplus q_i(C_i, D)$ is d-connecting given $\mathbf{Z}$. Similarly as above, $D \in \mathrm{An}(\mathbf{Z}, \mathcal{D})$, so if D is a collider on p', p' is d-connecting given $\mathbf{Z}$. If D is a non-collider on p', then $p(D, Y)$ starts with an edge out of D, so D is also a non-collider on p. Thus, in this case $D \notin \mathbf{Z}$ and hence, p' is d-connecting given $\mathbf{Z}$.

Thus, we have shown that no node on q_i, other than C_i, is on p, for all $i \in \{1, \ldots, r\}$. Next, we consider the case $r > 1$ and show, by contradiction, that no node on q_i is on q_j, for any $i, j \in \{1, \ldots, r\}$ with $i \neq j$. Hence, suppose that there are q_i and q_j such that a node on q_i is also on q_j, for some $i < j$. Let D be the node closest to C_i on q_i, that is also on q_j. Note that from the above, $D \neq C_j$ and $D \neq C_i$, so that q_i and q_j are at least of length 1. Then let $p' = p(X, C_i) \oplus q_i(C_i, D) \oplus (-q_j)(D, C_j) \oplus p(C_j, Y)$. As discussed above, no node on q_i (or q_j) is in $\mathbf{X}$. Hence, p' is a proper path from $\mathbf{X}$ to $\mathbf{Y}$. Since $(-q_j)(D, C_j)$ is of the form $D \leftarrow \cdots \leftarrow C_j$, p' is also a non-causal path from X to Y. Thus p' is a proper non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ that by construction has fewer colliders than p, since C_i and C_j are non-colliders on p'. Hence, in order to reach a contradiction we only need to show that it is d-connecting given $\mathbf{Z}$.

Since $p, q_1, \ldots, q_r$ are d-connecting given $\mathbf{Z}$, we only need to discuss the collider/non-collider status of D, C_i and C_j on p'. Since C_i and C_j are non-colliders on p', we have that $p'(X, D)$ and $p'(D, Y)$ are both d-connecting given $\mathbf{Z}$. Since D is on q_i, $D \in \mathrm{An}(\mathbf{Z}, \mathcal{D})$. Since D is a collider on p', p' is d-connecting given $\mathbf{Z}$.

We have now established that $\mathcal{D}'$ looks like Figure 12, where none of the paths intersect, and we can compute the covariance matrix of $(X, \tilde{\mathbf{Z}}^T, Y)^T$ using Wright's rule (Theorem 31) on $\mathcal{D}'$. For this purpose, let a_1 and a_{r+1} be the products of all edge coefficients on $p(X, C_1)$ and $p(C_r, Y)$, respectively. Let b_i, $i \in \{1, \ldots, r\}$ be product of all edge coefficients on q_i. If $r > 1$, let a_j, $j \in \{2, \ldots, r\}$ be the product of all edge coefficients on $p(C_{j-1}, C_j)$. Let $\mathbf{\Sigma}$ be the covariance matrix of $(X, \tilde{\mathbf{Z}}^T, Y)^T$. Then using Wright's rule (Theorem 31) on $\mathcal{D}'$ yields:

$$\mathbf{\Sigma} = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & \cdots & r+1 & r+2 \end{matrix} \\ \begin{bmatrix} 1 & a_1 b_1 & & & & \\ a_1 b_1 & 1 & b_1 a_2 b_2 & & & \\ & b_1 a_2 b_2 & 1 & \ddots & & 0 \\ & & \ddots & \ddots & \ddots & \\ & 0 & & 1 & b_{r-1} a_r b_r & \\ & & & b_{r-1} a_r b_r & 1 & b_r a_{r+1} \\ & & & & b_r a_{r+1} & 1 \end{bmatrix} \end{array} = \begin{bmatrix} \mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{22} \end{bmatrix},$$

where $\mathbf{\Sigma}_{11}$ is the covariance matrix of $(X, \tilde{\mathbf{Z}}^T)^T$, $\mathbf{\Sigma}_{22} = 1$, $\mathbf{\Sigma}_{21} = \mathbf{\Sigma}_{12}^T$ and $\mathbf{\Sigma}_{21} = \begin{bmatrix} 0 & \cdots & 0 & b_r a_{r+1} \end{bmatrix}$, $\mathbf{\Sigma}_{21} \in \mathbb{R}^{1\times(r+1)}$. Then using Theorem 32 we have that

$$E[Y \mid X = 1, \tilde{\mathbf{z}}] = \begin{bmatrix} 0 & \dots & 0 & b_r a_{r+1} \end{bmatrix} \mathbf{\Sigma}_{11}^{-1} \begin{bmatrix} 1 \\ \tilde{\mathbf{z}} \end{bmatrix}.$$

Since our multivariate Gaussian distribution is non-degenerate, $\mathrm{Det}(\mathbf{\Sigma}_{11}) > 0$. Let $t_{i,j}$ denote $(i,j)^{th}$ element of $\mathbf{\Sigma}_{11}^{-1}$, $\mathbf{\Sigma}_{11}^{-1} \in \mathbb{R}^{(r+1)\times(r+1)}$. Putting everything together, we have that

$$\begin{aligned}
\int_{\mathbf{z}} E[Y|\mathbf{X} = \mathbf{1}, \mathbf{z}] f(\mathbf{z}) d\mathbf{z} &= \int_{\tilde{\mathbf{z}}} E[Y|X = 1, \tilde{\mathbf{z}}] f(\tilde{\mathbf{z}}) d\tilde{\mathbf{z}} \\
&= \int_{\tilde{\mathbf{z}}} \begin{bmatrix} 0 & \dots & 0 & b_r a_{r+1} \end{bmatrix} \begin{bmatrix} t_{1,1} & \dots & t_{1,r+1} \\ \vdots & \ddots & \vdots \\ t_{r+1,1} & \dots & t_{r+1,r+1} \end{bmatrix} \begin{bmatrix} 1 \\ \tilde{\mathbf{z}} \end{bmatrix} f(\tilde{\mathbf{z}}) d\tilde{\mathbf{z}} \\
&= b_r a_{r+1} t_{r+1,1} + \sum_{i=1}^{r} b_r a_{r+1} t_{r+1,i+1} E[Z_i] = b_r a_{r+1} t_{r+1,1}.
\end{aligned}$$

Since $b_r \neq 0$ and $a_{r+1} \neq 0$, it is only left to show that $t_{r+1,1} \neq 0$. Using standard linear algebra, we find

$$\begin{aligned}
t_{r+1,1} &= \frac{(-1)^r}{\mathrm{Det}(\mathbf{\Sigma}_{11})} \begin{vmatrix} a_1 b_1 & 1 & b_1 a_2 b_2 & & & \\ & b_1 a_2 b_2 & 1 & b_2 a_3 b_3 & & 0 \\ & & \ddots & \ddots & \ddots & \\ & 0 & & b_{r-3} a_{r-2} b_{r-2} & 1 & b_{r-2} a_{r-1} b_{r-1} \\ & & & & b_{r-2} a_{r-1} b_{r-1} & 1 \\ & & & & & b_{r-1} a_r b_r \end{vmatrix} \\
&= \frac{(-1)^r a_r b_r}{\mathrm{Det}(\mathbf{\Sigma}_{11})} \prod_{i=1}^{r-1} a_i b_i^2.
\end{aligned}$$

Since $a_i \neq 0$ and $b_i \neq 0$ for all $i \in \{1, \dots, r\}$, the proof is completed.

■

Theorem 58 ***(Soundness of the adjustment criterion for*** DAG***s)*** *Let $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z_0}$ be pairwise disjoint node sets in a causal* DAG *$\mathcal{D}$. If $\mathbf{Z_0}$ satisfies the adjustment criterion (see Definition 55), then $\mathbf{Z_0}$ is an adjustment set (see Definition 54).*

Proof of Theorem 58. Assume that $\mathbf{Z_0}$ satisfies the adjustment criterion (see Definition 55) relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$ and let f be a density consistent with $\mathcal{D}$. We need to show that

$$f(\mathbf{y} \mid \mathrm{do}(\mathbf{x})) = \int_{\mathbf{z_0}} f(\mathbf{y} \mid \mathbf{x}, \mathbf{z_0}) f(\mathbf{z_0}) d\mathbf{z_0}. \tag{9}$$

Let $\mathbf{Y_D} = \mathbf{Y} \cap \mathrm{De}(\mathbf{X}, \mathcal{D})$ and $\mathbf{Y_N} = \mathbf{Y} \setminus \mathrm{De}(\mathbf{X}, \mathcal{D})$. Then $\mathbf{Y_N} \perp_d \mathbf{X}$ in $\mathcal{D}_{\overline{\mathbf{X}}}$, since $\mathcal{D}_{\overline{\mathbf{X}}}$ does not contain paths into $\mathbf{X}$ and all paths from $\mathbf{X}$ to $\mathbf{Y_N}$ that are out of $\mathbf{X}$ in $\mathcal{D}_{\overline{\mathbf{X}}}$ must contain a collider by definition of $\mathbf{Y_N}$. Hence, using Rule 3 of the do-calculus (see Equation 6 in Appendix A), with $\mathbf{X}' = \emptyset, \mathbf{W}' = \emptyset$, $\mathbf{Z}' = \mathbf{X}$ and $\mathbf{Y}' = \mathbf{Y_N}$ we have

$$f(\mathbf{y_N}) = f(\mathbf{y_N} \mid \mathrm{do}(\mathbf{x})). \tag{10}$$

Now, assume that $\mathbf{Y_D} = \emptyset$, so that $\mathbf{Y} = \mathbf{Y_N}$. Then all paths from $\mathbf{X}$ to $\mathbf{Y}$ are non-causal in $\mathcal{D}$. Since $\mathbf{Z_0}$ satisfies the adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$ it blocks all proper non-causal paths from $\mathbf{X}$ to $\mathbf{Y}$. Thus, $\mathbf{Z_0}$ also blocks all non-causal paths from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{D}$, so $\mathbf{X} \perp_d \mathbf{Y} \mid \mathbf{Z_0}$ in $\mathcal{D}$. Combining the probabilistic implications of d-separation and Equation (10) with the right-hand side of Equation (9) we obtain

$$\begin{aligned}
\int_{\mathbf{z_0}} f(\mathbf{y} \mid \mathbf{x}, \mathbf{z_0}) f(\mathbf{z_0}) d\mathbf{z_0} &= \int_{\mathbf{z_0}} f(\mathbf{y} \mid \mathbf{z_0}) f(\mathbf{z_0}) d\mathbf{z_0} \\
&= f(\mathbf{y}) = f(\mathbf{y} \mid \mathrm{do}(\mathbf{x})).
\end{aligned} \tag{11}$$

For the remainder of the proof we assume $\mathbf{Y_D} \neq \emptyset$. We enlarge the set $\mathbf{Z_0}$ to $\mathbf{Z} = \mathbf{Z_0} \cup \mathrm{An}(\mathbf{X} \cup \mathbf{Y}, \mathcal{D}) \setminus (\mathrm{De}(\mathbf{X}, \mathcal{D}) \cup \mathbf{Y})$. Then applying (iii) in Lemma 59 to the right-hand side of Equation (9) we get

$$\int_{\mathbf{z_0}} f(\mathbf{y} \mid \mathbf{x}, \mathbf{z_0}) f(\mathbf{z_0}) d\mathbf{z_0} = \int_{\mathbf{z}} f(\mathbf{y} \mid \mathbf{x}, \mathbf{z}) f(\mathbf{z}) d\mathbf{z}. \tag{12}$$

Let $\mathbf{Z_D} = \mathbf{Z} \cap \mathrm{De}(\mathbf{X}, \mathcal{D})$ and $\mathbf{Z_N} = \mathbf{Z} \setminus \mathrm{De}(\mathbf{X}, \mathcal{D})$. Now suppose that $\mathbf{Y} = \mathbf{Y_D}$, so that $\mathbf{Y_N} = \emptyset$. From (iii) in Lemma 60, we have $\mathbf{Y_D} \perp_d \mathbf{Z_D} \mid \mathbf{X} \cup \mathbf{Z_N}$. Using the probabilistic implications of d-separation, the right-hand side of Equation (12) equals

$$\begin{aligned}
\int_{\mathbf{z_D}, \mathbf{z_N}} f(\mathbf{y} \mid \mathbf{x}, \mathbf{z_D}, \mathbf{z_N}) f(\mathbf{z_D}, \mathbf{z_N}) d\mathbf{z_D} d\mathbf{z_N} &= \int_{\mathbf{z_D}, \mathbf{z_N}} f(\mathbf{y} \mid \mathbf{x}, \mathbf{z_N}) f(\mathbf{z_D}, \mathbf{z_N}) d\mathbf{z_D} d\mathbf{z_N} \\
&= \int_{\mathbf{z_N}} f(\mathbf{y} \mid \mathbf{x}, \mathbf{z_N}) f(\mathbf{z_N}) d\mathbf{z_N}.
\end{aligned} \tag{13}$$

Since $\mathbf{Z_N}$ satisfies the generalized back-door criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$ (see (iv) in Lemma 60) and since the generalized back-door criterion is sound by Theorem 3.1 in Maathuis and Colombo (2015), we have

$$\int_{\mathbf{z_N}} f(\mathbf{y} \mid \mathbf{x}, \mathbf{z_N}) f(\mathbf{z_N}) d\mathbf{z_N} = f(\mathbf{y} \mid \mathrm{do}(\mathbf{x})). \tag{14}$$

Combining Equations (12), (13) and (14) completes the proof when $\mathbf{Y_N} = \emptyset$. In the remainder of the proof we assume $\mathbf{Y_D} \neq \emptyset$ and $\mathbf{Y_N} \neq \emptyset$. From (iii) in Lemma 60,

$\mathbf{Y_D} \perp_d \mathbf{Z_D} \mid \mathbf{Y_N} \cup \mathbf{X} \cup \mathbf{Z_N}$. Using the probabilistic implications of d-separation, the right-hand side of Equation (12) equals

$$\begin{aligned}
&\int_{\mathbf{z_D},\mathbf{z_N}} f(\mathbf{y_D},\mathbf{y_N} \mid \mathbf{x},\mathbf{z_D},\mathbf{z_N}) f(\mathbf{z_D},\mathbf{z_N}) d\mathbf{z_D} d\mathbf{z_N} \\
&= \int_{\mathbf{z_D},\mathbf{z_N}} f(\mathbf{y_D} \mid \mathbf{y_N},\mathbf{x},\mathbf{z_D},\mathbf{z_N}) f(\mathbf{y_N} \mid \mathbf{x},\mathbf{z_D},\mathbf{z_N}) f(\mathbf{z_D},\mathbf{z_N}) d\mathbf{z_D} d\mathbf{z_N} \\
&= \int_{\mathbf{z_N}} f(\mathbf{y_D} \mid \mathbf{y_N},\mathbf{x},\mathbf{z_N}) \int_{\mathbf{z_D}} f(\mathbf{y_N} \mid \mathbf{x},\mathbf{z_D},\mathbf{z_N}) f(\mathbf{z_D},\mathbf{z_N}) d\mathbf{z_D} d\mathbf{z_N}. \qquad (15)
\end{aligned}$$

Since $\mathbf{Z}$ satisfies the adjustment criterion relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{D}$ ((ii) in Lemma 59), $\mathbf{Z}$ blocks all proper non-causal paths from $\mathbf{X}$ to $\mathbf{Y}$. Hence, $\mathbf{Z}$ blocks all proper paths from $\mathbf{X}$ to $\mathbf{Y_N}$ in $\mathcal{D}$, so $\mathbf{X} \perp_d \mathbf{Y_N} \mid \mathbf{Z}$ in $\mathcal{D}$. Additionally, the empty set satisfies the generalized back-door criterion relative to $((\mathbf{Y_N} \cup \mathbf{X} \cup \mathbf{Z_N}), \mathbf{Y_D})$ ((v) in Lemma 60). Since the generalized back-door criterion is sound by Theorem 3.1 in Maathuis and Colombo (2015), we apply this to the right-hand side of Equation (15)

$$\begin{aligned}
&\int_{\mathbf{z_N}} f(\mathbf{y_D} \mid \mathbf{y_N},\mathbf{x},\mathbf{z_N}) \int_{\mathbf{z_D}} f(\mathbf{y_N} \mid \mathbf{x},\mathbf{z_D},\mathbf{z_N}) f(\mathbf{z_D},\mathbf{z_N}) d\mathbf{z_D} d\mathbf{z_N} \\
&= \int_{\mathbf{z_N}} f(\mathbf{y_D} \mid \mathrm{do}(\mathbf{x},\mathbf{y_N},\mathbf{z_N})) \int_{\mathbf{z_D}} f(\mathbf{y_N} \mid \mathbf{z_N},\mathbf{z_D}) f(\mathbf{z_D},\mathbf{z_N}) d\mathbf{z_D} d\mathbf{z_N} \\
&= \int_{\mathbf{z_N}} f(\mathbf{y_D} \mid \mathrm{do}(\mathbf{x},\mathbf{y_N},\mathbf{z_N})) f(\mathbf{z_N} \mid \mathbf{y_N}) f(\mathbf{y_N}) d\mathbf{z_N}. \qquad (16)
\end{aligned}$$

To finish the proof we rely on the do-calculus rules. By (vi) in Lemma 60, it follows that $\mathbf{Y_N} \cup \mathbf{Z_N} \perp_d \mathbf{Y_D} \mid \mathbf{X}$ in $\mathcal{D}_{\overline{\mathbf{X}\mathbf{Y_N}\cup\mathbf{Z_N}}}$. Using Rule 2 of the do-calculus (see Equation 5 in Appendix A) with $\mathbf{X'} = \mathbf{X}, \mathbf{W'} = \emptyset, \mathbf{Z'} = \mathbf{Y_N} \cup \mathbf{Z_N}$ and $\mathbf{Y'} = \mathbf{Y_D}$:

$$f(\mathbf{y_D} \mid \mathrm{do}(\mathbf{x},\mathbf{y_N},\mathbf{z_N})) = f(\mathbf{y_D} \mid \mathrm{do}(\mathbf{x}),\mathbf{z_N},\mathbf{y_N}). \qquad (17)$$

Additionally, by (vii) in Lemma 60, $\mathbf{Z_N} \perp_d \mathbf{X} \mid \mathbf{Y_N}$ in $\mathcal{D}_{\overline{\mathbf{X}}}$. Using Rule 3 of the do-calculus (see Equation 6 in Appendix A) with $\mathbf{X'} = \emptyset, \mathbf{W'} = \mathbf{Y_N}, \mathbf{Z'} = \mathbf{X}$ and $\mathbf{Y'} = \mathbf{Z_N}$:

$$f(\mathbf{z_N} \mid \mathbf{y_N}) = f(\mathbf{z_N} \mid \mathrm{do}(\mathbf{x}),\mathbf{y_N}). \qquad (18)$$

Finally, we combine Equations (17),(18) and (10) with the right-hand side of Equation (16):

$$\begin{aligned}
&\int_{\mathbf{z_N}} f(\mathbf{y_D} \mid \mathrm{do}(\mathbf{y_N},\mathbf{x},\mathbf{z_N})) f(\mathbf{z_N} \mid \mathbf{y_N}) f(\mathbf{y_N}) d\mathbf{z_N} \\
&= \int_{\mathbf{z_N}} f(\mathbf{y_D} \mid \mathbf{z_N},\mathbf{y_N},\mathrm{do}(\mathbf{x})) f(\mathbf{z_N} \mid \mathbf{y_N},\mathrm{do}(\mathbf{x})) f(\mathbf{y_N} \mid \mathrm{do}(\mathbf{x})) d\mathbf{z_N} \\
&= \int_{\mathbf{z_N}} f(\mathbf{y_D},\mathbf{z_N} \mid \mathbf{y_N},\mathrm{do}(\mathbf{x})) f(\mathbf{y_N} \mid \mathrm{do}(\mathbf{x})) d\mathbf{z_N} \\
&= f(\mathbf{y_D} \mid \mathbf{y_N},\mathrm{do}(\mathbf{x})) f(\mathbf{y_N} \mid \mathrm{do}(\mathbf{x})) = f(\mathbf{y_D},\mathbf{y_N} \mid \mathrm{do}(\mathbf{x})) = f(\mathbf{y} \mid \mathrm{do}(\mathbf{x})).
\end{aligned}$$

■

Lemma 59 *Let $\mathbf{X},\mathbf{Y}$ and $\mathbf{Z_0}$ be pairwise disjoint node sets in a* DAG *$\mathcal{D}$ such that $\mathbf{Z_0}$ satisfies the adjustment criterion relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{D}$. Let $\mathbf{Z_1} \subseteq \mathrm{An}(\mathbf{X}\cup\mathbf{Y},\mathcal{D})\backslash(\mathrm{De}(\mathbf{X},\mathcal{D})\cup\mathbf{Y})$ and $\mathbf{Z} = \mathbf{Z_0} \cup \mathbf{Z_1}$. Then:*

(i) $\mathbf{X},\mathbf{Y}$ and $\mathbf{Z}$ are pairwise disjoint, and

(ii) $\mathbf{Z}$ satisfies the adjustment criterion relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{D}$, and

(iii) $\int_{\mathbf{z_0}} f(\mathbf{y} \mid \mathbf{x},\mathbf{z_0}) f(\mathbf{z_0}) d\mathbf{z_0} = \int_{\mathbf{z_0},\mathbf{z_1}} f(\mathbf{y} \mid \mathbf{x},\mathbf{z_0},\mathbf{z_1}) f(\mathbf{z_0},\mathbf{z_1}) d\mathbf{z_0} d\mathbf{z_1}$, for any density f consistent with $\mathcal{D}$.

Proof of Lemma 59. (i) Since $\mathbf{X},\mathbf{Y}$ and $\mathbf{Z_0}$ are pairwise disjoint, and $\mathbf{Z} = \mathbf{Z_0} \cup \mathbf{Z_1}$, where $\mathbf{Z_1} \cap (\mathbf{X} \cup \mathbf{Y}) = \emptyset$, it follows that $\mathbf{X},\mathbf{Y}$ and $\mathbf{Z}$ are also pairwise disjoint.

(ii) A set that satisfies the forbidden set condition of the adjustment criterion relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{D}$ satisfies the blocking condition if and only if it d-separates $\mathbf{X}$ and $\mathbf{Y}$ in the proper back-door graph $\mathcal{D}^{pbd}_{\mathbf{XY}}$ (van der Zander et al. (2014, Theorem 4.6); see Theorem 7 in Section 3). Thus, $\mathbf{Z_0}$ d-separates $\mathbf{X}$ and $\mathbf{Y}$ in $\mathcal{D}^{pbd}_{\mathbf{XY}}$. Then by Theorem 35, any path between $\mathbf{X}$ and $\mathbf{Y}$ in $((\mathcal{D}^{pbd}_{\mathbf{XY}})_{\mathrm{An}(\mathbf{X}\cup\mathbf{Y}\cup\mathbf{Z_0},\mathcal{D}^{pbd}_{\mathbf{XY}})})^m$ contains a node in $\mathbf{Z_0}$.

Since $\mathbf{Z_1} \cap \mathrm{De}(\mathbf{X},\mathcal{D}) = \emptyset$, $\mathrm{Forb}(\mathbf{X},\mathbf{Y},\mathcal{D}) \subseteq \mathrm{De}(\mathbf{X},\mathcal{D})$ and $\mathbf{Z_0} \cap \mathrm{Forb}(\mathbf{X},\mathbf{Y},\mathcal{D}) = \emptyset$, $\mathbf{Z} \cap \mathrm{Forb}(\mathbf{X},\mathbf{Y},\mathcal{D}) = \emptyset$. Additionally, since $\mathbf{Z} \supseteq \mathbf{Z_0}$, all paths between $\mathbf{X}$ and $\mathbf{Y}$ in $((\mathcal{D}^{pbd}_{\mathbf{XY}})_{\mathrm{An}(\mathbf{X}\cup\mathbf{Y}\cup\mathbf{Z_0},\mathcal{D}^{pbd}_{\mathbf{XY}})})^m$ contain a node in $\mathbf{Z}$. Furthermore, $\mathbf{Z_1} \subseteq \mathrm{An}(\mathbf{X}\cup\mathbf{Y},\mathcal{D})$ implies that $\mathrm{An}(\mathbf{X}\cup\mathbf{Y}\cup\mathbf{Z},\mathcal{D}) = \mathrm{An}(\mathbf{X}\cup\mathbf{Y}\cup\mathbf{Z_0},\mathcal{D})$. Thus, all paths between $\mathbf{X}$ and $\mathbf{Y}$ in $((\mathcal{D}^{pbd}_{\mathbf{XY}})_{\mathrm{An}(\mathbf{X}\cup\mathbf{Y}\cup\mathbf{Z},\mathcal{D}^{pbd}_{\mathbf{XY}})})^m$ contain a node in $\mathbf{Z}$. Hence, $\mathbf{Z}$ satisfies the blocking condition relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{D}$ (Theorem 35, Theorem 7).

(iii) We prove this statement by induction on the number of nodes in $\mathbf{Z_1}$. Below, we prove the base case: $|\mathbf{Z_1}| = 1$. We then assume that the result holds for $|\mathbf{Z_1}| = k$, and show that it holds for $|\mathbf{Z_1}| = k+1$. Thus, let $|\mathbf{Z_1}| = k+1$ and take an arbitrary $Z_1 \in \mathbf{Z_1}$. Let $\mathbf{Z_0'} = \mathbf{Z_0} \cup \{Z_1\}$ and $\mathbf{Z_1'} = \mathbf{Z_1} \setminus \{Z_1\}$. The base case then implies

$$\begin{aligned}
\int_{\mathbf{z_0}} f(\mathbf{y} \mid \mathbf{x},\mathbf{z_0}) f(\mathbf{z_0}) d\mathbf{z_0} &= \int_{\mathbf{z_0}} f(\mathbf{y} \mid \mathbf{x},\mathbf{z_0}) \int_{z_1} f(\mathbf{z_0},z_1) dz_1 d\mathbf{z_0} \\
&= \int_{\mathbf{z_0},z_1} f(\mathbf{y} \mid \mathbf{x},\mathbf{z_0},z_1) f(\mathbf{z_0},z_1) d\mathbf{z_0} dz_1 \\
&= \int_{\mathbf{z_0'}} f(\mathbf{y} \mid \mathbf{x},\mathbf{z_0'}) f(\mathbf{z_0'}) d\mathbf{z_0'}. \qquad (19)
\end{aligned}$$

By (ii) above $\mathbf{Z_0'}$ satisfies the adjustment criterion relative to $(\mathbf{X},\mathbf{Y})$ in $\mathcal{D}$ and $\mathbf{Z_1'} \subseteq \mathrm{An}(\mathbf{X}\cup\mathbf{Y},\mathcal{D})$. Then since $|\mathbf{Z_1'}| = k$, by the induction hypothesis

$$\int_{\mathbf{z_0'}} f(\mathbf{y} \mid \mathbf{x},\mathbf{z_0'}) f(\mathbf{z_0'}) d\mathbf{z_0'} = \int_{\mathbf{z_0'},\mathbf{z_1'}} f(\mathbf{y} \mid \mathbf{x},\mathbf{z_0'},\mathbf{z_1'}) f(\mathbf{z_0'},\mathbf{z_1'}) d\mathbf{z_0'} d\mathbf{z_1'}. \qquad (20)$$

Combining 19 and 20 yields

$$\int_{\mathbf{z_0}} f(\mathbf{y} \mid \mathbf{x},\mathbf{z_0}) f(\mathbf{z_0}) d\mathbf{z_0} = \int_{\mathbf{z_0},\mathbf{z_1}} f(\mathbf{y} \mid \mathbf{x},\mathbf{z_0},\mathbf{z_1}) f(\mathbf{z_0},\mathbf{z_1}) d\mathbf{z_0} d\mathbf{z_1}.$$

It is left to prove the base case of the induction. Hence, suppose $\mathbf{Z_1} = \{Z_1\}$. We show below that either (a) $\mathbf{Y} \perp_d Z_1 \mid \mathbf{X} \cup \mathbf{Z_0}$ or (b) $\mathbf{X} \perp_d Z_1 \mid \mathbf{Z_0}$ are satisfied in $\mathcal{D}$. (Note that (a) and (b) are very similar to the conditions U1 and U2, as well as U1* and U2*, from Greenland et al., 1999. These conditions are also used in Theorem 5 from Kuroki and Miyakawa, 2003, Lemma 3 from Kuroki and Cai, 2004 and are the foundation for the results in De Luna et al., 2011. Additionally, Pearl and Paz, 2014 give a discussion of these conditions, which they refer to as c-equivalence conditions, in their Theorem 1.)

If (a) $\mathbf{Y} \perp_d Z_1 \mid \mathbf{X} \cup \mathbf{Z_0}$ in $\mathcal{D}$, then by the probabilistic implications of d-separation we have that for any density f consistent with $\mathcal{D}$

$$\begin{aligned}\int_{\mathbf{z_0}} f(\mathbf{y} \mid \mathbf{x}, \mathbf{z_0}) f(\mathbf{z_0}) d\mathbf{z_0} &= \int_{\mathbf{z_0}} f(\mathbf{y} \mid \mathbf{x}, \mathbf{z_0}) \int_{z_1} f(\mathbf{z_0}, z_1) dz_1 d\mathbf{z_0} \\ &= \int_{\mathbf{z_0}, z_1} f(\mathbf{y} \mid \mathbf{x}, \mathbf{z_0}) f(\mathbf{z_0}, z_1) d\mathbf{z_0} dz_1 \\ &= \int_{\mathbf{z_0}, z_1} f(\mathbf{y} \mid \mathbf{x}, \mathbf{z_0}, z_1) f(\mathbf{z_0}, z_1) d\mathbf{z_0} dz_1.\end{aligned}$$

If (b) $\mathbf{X} \perp_d Z_1 \mid \mathbf{Z_0}$ in $\mathcal{D}$, then similarly

$$\begin{aligned}\int_{\mathbf{z_0}} f(\mathbf{y} \mid \mathbf{x}, \mathbf{z_0}) f(\mathbf{z_0}) d\mathbf{z_0} &= \int_{\mathbf{z_0}} f(\mathbf{z_0}) \int_{z_1} f(\mathbf{y}, z_1 \mid \mathbf{x}, \mathbf{z_0}) dz_1 d\mathbf{z_0} \\ &= \int_{\mathbf{z_0}, z_1} f(\mathbf{y}, z_1 \mid \mathbf{x}, \mathbf{z_0}) f(\mathbf{z_0}) d\mathbf{z_0} dz_1 \\ &= \int_{\mathbf{z_0}, z_1} f(\mathbf{y} \mid \mathbf{x}, \mathbf{z_0}, z_1) f(z_1 \mid \mathbf{x}, \mathbf{z_0}) f(\mathbf{z_0}) d\mathbf{z_0} dz_1 \\ &= \int_{\mathbf{z_0}, z_1} f(\mathbf{y} \mid \mathbf{x}, \mathbf{z_0}, z_1) f(z_1 \mid \mathbf{z_0}) f(\mathbf{z_0}) d\mathbf{z_0} dz_1 \\ &= \int_{\mathbf{z_0}, z_1} f(\mathbf{y} \mid \mathbf{x}, \mathbf{z_0}, z_1) f(z_1, \mathbf{z_0}) d\mathbf{z_0} dz_1.\end{aligned}$$

We complete the proof by showing that (a) or (b) must hold. Suppose for a contradiction that both (a) and (b) are violated. Then there is a path from $\mathbf{X}$ to Z_1 that is d-connecting given $\mathbf{Z_0}$ and a path from $\mathbf{Y}$ to Z_1 that is d-connecting given $\mathbf{X} \cup \mathbf{Z_0}$. Let p be a proper path from $X \in \mathbf{X}$ to Z_1 that is d-connecting given $\mathbf{Z_0}$ in $\mathcal{D}$ and let q be a path from Z_1 to $Y \in \mathbf{Y}$ that is d-connecting given $\mathbf{X} \cup \mathbf{Z_0}$ in $\mathcal{D}$. We will show that this contradicts that $\mathbf{Z_0}$ satisfies the adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$.

We first show, by contradiction, that q also is d-connecting given $\mathbf{Z_0}$. Thus, assume that $q = \langle Z_1, \dots, Y \rangle$ is blocked by $\mathbf{Z_0}$. Since q is d-connecting given $\mathbf{X} \cup \mathbf{Z_0}$ and blocked by $\mathbf{Z_0}$, it must contain at least one collider in $\mathrm{An}(\mathbf{X}, \mathcal{D}) \setminus \mathrm{An}(\mathbf{Z_0}, \mathcal{D})$. Let C be the collider closest to Y on q such that $C \in \mathrm{An}(\mathbf{X}, \mathcal{D}) \setminus \mathrm{An}(\mathbf{Z_0}, \mathcal{D})$. Let $r = \langle C, \dots, X' \rangle, X' \in \mathbf{X}$ be a shortest directed path from C to $\mathbf{X}$ (possibly of zero length). Then no node on $q(C, Y)$ or r, except possibly C, is in $\mathbf{X}$. We now concatenate the paths $-r(X', C)$ and $q(C, Y)$, while taking out possible loops. Hence, let V be the node closest to X' on r that is also on $q(C, Y)$. Then $-r(X', V) \oplus q(V, Y)$ is non-causal since either $-r(X', V)$ is of non-zero length, or $X' = V = C$ and $q(C, Y)$ is a path into C, because C is a collider on q. By the

choice of C and r, $-r(X', V) \oplus q(V, Y)$ is a proper path that is d-connecting given $\mathbf{Z_0}$. This contradicts that $\mathbf{Z_0}$ satisfies the adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$. Thus, q is also d-connecting given $\mathbf{Z_0}$.

Let $\tilde{p}$ and $\tilde{q}$ be paths in the proper back-door graph $\mathcal{D}^{pbd}_{\mathbf{XY}}$ constituted by the same sequences of nodes as p and q in $\mathcal{D}$ respectively. We first prove that the paths $\tilde{p}$ and $\tilde{q}$ exist in $\mathcal{D}^{pbd}_{\mathbf{XY}}$. Path q is d-connecting given $\mathbf{X} \cup \mathbf{Z_0}$, so any node in $\mathbf{X}$ on q must be a collider on q. Since $\mathcal{D}^{pbd}_{\mathbf{XY}}$ is obtained from $\mathcal{D}$ by removing certain edges out of $\mathbf{X}$, no edges from q are removed and $\tilde{q}$ exists in $\mathcal{D}^{pbd}_{\mathbf{XY}}$.

Since p is proper, for $\tilde{p}$ to exist in $\mathcal{D}^{pbd}_{\mathbf{XY}}$, it is enough to show that p does not start with an edge of type $X \to W$ in $\mathcal{D}$ where W lies on a proper causal path from X to $\mathbf{Y}$ in $\mathcal{D}$. Suppose for a contradiction that p does start with $X \to W$. Then $W \in \mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D})$. Then either p is a directed path from X to Z_1 so that $Z_1 \in \mathrm{De}(W, \mathcal{D})$, or p is non-causal and there is a collider C' on p such that $C' \in \mathrm{De}(W, \mathcal{D})$. In the former case, since $\mathrm{De}(W, \mathcal{D}) \subseteq \mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D})$, it follows that $Z_1 \in \mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D})$, which contradicts the choice of Z_1 because $Z_1 \notin \mathrm{De}(\mathbf{X}, \mathcal{D})$. In the latter case, since p is d-connecting given $\mathbf{Z_0}$, we have $\mathrm{De}(C', \mathcal{D}) \cap \mathbf{Z_0} \neq \emptyset$. Combining this with $\mathrm{De}(C', \mathcal{D}) \subseteq \mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D})$, it follows that $\mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D}) \cap \mathbf{Z_0} \neq \emptyset$ which contradicts that $\mathbf{Z_0}$ satisfies the forbidden set condition relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$. Thus, $\tilde{p}$ exists in $\mathcal{D}^{pbd}_{\mathbf{XY}}$.

We now show that $\tilde{p}$ and $\tilde{q}$ are d-connecting given $\mathbf{Z_0}$ in $\mathcal{D}^{pbd}_{\mathbf{XY}}$. The collider/non-collider status of any node on $\tilde{p}$ and $\tilde{q}$ in $\mathcal{D}^{pbd}_{\mathbf{XY}}$ is the same as the collider/non-collider status of that same node on p and q in $\mathcal{D}$ respectively. So $\tilde{p}$ and $\tilde{q}$ are both d-connecting given $\mathbf{Z_0}$, unless every causal path from a collider on either p or q to $\mathbf{Z_0}$ contains a first edge on a proper causal path from $\mathbf{X}$ to $\mathbf{Y}$ in $\mathcal{D}$. Any such causal path also contains a node in $\mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D})$, so a node in $\mathbf{Z_0}$ would be a descendant of $\mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D})$ in $\mathcal{D}$. Since $\mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D})$ is descendral, it follows that $\mathbf{Z_0} \cap \mathrm{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D}) \neq \emptyset$. This contradicts that $\mathbf{Z_0}$ satisfies the adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$, specifically the forbidden set condition.

Since $\tilde{p}$ is d-connecting given $\mathbf{Z_0}$ in $\mathcal{D}^{pbd}_{\mathbf{XY}}$, by Theorem 35 it follows that there is a path a from X to Z_1 that does not contain a node in $\mathbf{Z_0}$ in the moral induced subgraph of $\mathcal{D}^{pbd}_{\mathbf{XY}}$ on nodes $\mathrm{An}(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z_0} \cup \{Z_1\}, \mathcal{D}^{pbd}_{\mathbf{XY}})$. Similarly, $\tilde{q}$ is a d-connecting path from Z_1 to Y given $\mathbf{Z_0}$ in $\mathcal{D}^{pbd}_{\mathbf{XY}}$, so there is a path b from Z_1 to Y that does not contain a node in $\mathbf{Z_0}$ in the moral induced subgraph of $\mathcal{D}^{pbd}_{\mathbf{XY}}$ on nodes $\mathrm{An}(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z_0} \cup \{Z_1\}, \mathcal{D}^{pbd}_{\mathbf{XY}})$. By combining paths a and b we get a path c from $\mathbf{X}$ to $\mathbf{Y}$ that does not contain a node in $\mathbf{Z_0}$ in the moral induced subgraph of $\mathcal{D}^{pbd}_{\mathbf{XY}}$ on nodes $\mathrm{An}(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z_0} \cup \{Z_1\}, \mathcal{D}^{pbd}_{\mathbf{XY}})$. Since $Z_1 \in \mathrm{An}(\mathbf{X} \cup \mathbf{Y}, \mathcal{D})$ and $\mathcal{D}^{pbd}_{\mathbf{XY}}$ is obtained from $\mathcal{D}$ by removing certain edges out of $\mathbf{X}$, it follows that $Z_1 \in \mathrm{An}(\mathbf{X} \cup \mathbf{Y}, \mathcal{D}^{pbd}_{\mathbf{XY}})$. Then $\mathrm{An}(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z_0} \cup \{Z_1\}, \mathcal{D}^{pbd}_{\mathbf{XY}}) = \mathrm{An}(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z_0}, \mathcal{D}^{pbd}_{\mathbf{XY}})$. Hence, c is a path from $\mathbf{X}$ to $\mathbf{Y}$ that does not contain a node in $\mathbf{Z_0}$ in the moral induced subgraph of $\mathcal{D}^{pbd}_{\mathbf{XY}}$ on nodes $\mathrm{An}(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z_0}, \mathcal{D}^{pbd}_{\mathbf{XY}})$. Thus, by Theorem 35, $\mathbf{X}$ and $\mathbf{Y}$ are d-connected given $\mathbf{Z_0}$ in $\mathcal{D}^{pbd}_{\mathbf{XY}}$. By Theorem 7, this contradicts that $\mathbf{Z_0}$ satisfies the adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$. ■

Lemma 60 *Let $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z_0}$ be pairwise disjoint node sets in a* DAG *$\mathcal{D}$ such that $\mathbf{Z_0}$ satisfies the adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$. Let $\mathbf{Z} = \mathbf{Z_0} \cup \mathrm{An}(\mathbf{X} \cup \mathbf{Y}, \mathcal{D}) \setminus (\mathrm{De}(\mathbf{X}, \mathcal{D}) \cup \mathbf{Y})$. Additionally, let $\mathbf{Z_D} = \mathbf{Z} \cap \mathrm{De}(\mathbf{X}, \mathcal{D})$, $\mathbf{Z_N} = \mathbf{Z} \setminus \mathrm{De}(\mathbf{X}, \mathcal{D})$, $\mathbf{Y_D} = \mathbf{Y} \cap \mathrm{De}(\mathbf{X}, \mathcal{D})$ and $\mathbf{Y_N} = \mathbf{Y} \setminus \mathrm{De}(\mathbf{X}, \mathcal{D})$. Then the following statements hold:*

(i) $(\mathbf{X} \cup \mathbf{Y_N} \cup \mathbf{Z}) \cap \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D}) = \emptyset$, *and*

(ii) *if* $p = \langle A, \dots, Y_d \rangle$ *is a non-causal path from a node* $A \in \mathbf{X} \cup \mathbf{Y_N} \cup \mathbf{Z}$ *to a node* $Y_d \in \mathbf{Y_D}$, *then* p *is blocked by* $(\mathbf{X} \cup \mathbf{Y_N} \cup \mathbf{Z_N}) \setminus \{A\}$ *in* $\mathcal{D}$, *and*

(iii) $\mathbf{Y_D} \perp_d \mathbf{Z_D} \mid \mathbf{Y_N} \cup \mathbf{X} \cup \mathbf{Z_N}$ *in* $\mathcal{D}$, *where* $\mathbf{Y_N} = \emptyset$ *is allowed, and*

(iv) *if* $\mathbf{Y_N} = \emptyset$ *then* $\mathbf{Z_N}$ *is a generalized back-door set relative to* $(\mathbf{X}, \mathbf{Y})$ *in* $\mathcal{D}$, *and*

(v) *the empty set is a generalized back-door set relative to* $((\mathbf{X} \cup \mathbf{Y_N} \cup \mathbf{Z_N}), \mathbf{Y_D})$ *in* $\mathcal{D}$, *and*

(vi) $\mathbf{Y_N} \cup \mathbf{Z_N} \perp_d \mathbf{Y_D} \mid \mathbf{X}$ *in* $\mathcal{D}_{\overline{\mathbf{X}}\underline{\mathbf{Y_N} \cup \mathbf{Z_N}}}$, *and*

(vii) $\mathbf{X} \perp_d \mathbf{Z_N} \mid \mathbf{Y_N}$ *in* $\mathcal{D}_{\overline{\mathbf{X}}}$.

Proof of Lemma 60. (i) By Lemma 59, $\mathbf{Z}$ satisfies the adjustment criterion relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{D}$, implying that $\mathbf{Z} \cap \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D}) = \emptyset$. By definition $\mathbf{Y_N} \cap \text{De}(\mathbf{X}, \mathcal{D}) = \emptyset$ and $\text{De}(\mathbf{X}, \mathcal{D}) \supseteq \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{G})$. Hence, $\mathbf{Y_N} \cap \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D}) = \emptyset$. It is only left to show that $\mathbf{X} \cap \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D}) = \emptyset$.

Suppose for a contradiction that $\mathbf{X} \cap \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D}) \neq \emptyset$. Let $V \notin \mathbf{X}$ be a node on a proper causal path p from $\mathbf{X}$ to $\mathbf{Y}$ (possibly $V \in \mathbf{Y}$) such that $V \in \text{An}(\mathbf{X}, \mathcal{D})$. Let $q = \langle V, \dots, X \rangle$, $X \in \mathbf{X}$, be a shortest causal path from V to $\mathbf{X}$. All nodes on $p(V, Y)$ and q are in $\text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D})$. We now concatenate $-q$ and $p(V, Y)$, while taking out loops. Hence, let W be the node closest to X on q that is also on $p(V, Y)$. Then $r = -q(X, W) \oplus p(W, Y)$ is a proper non-causal path from $\mathbf{X}$ to $\mathbf{Y}$ that cannot be blocked by $\mathbf{Z}$ since $\mathbf{Z} \cap \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D}) = \emptyset$, which contradicts Lemma 59.

(ii) We distinguish the cases that p is (a) out of, or (b) into Y_d.

(a) Let p be out of Y_d. Since $Y_d \in \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D})$ and since by (i) node $A \notin \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D})$, there is at least one collider on p. The collider closest to Y_d on p and all of its descendants are also in $\text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D})$. It then follows from (i) that p is blocked by $(\mathbf{X} \cup \mathbf{Y_N} \cup \mathbf{Z_N}) \setminus \{A\}$.

(b) Let p be into Y_d. Since p is a non-causal path from A to Y_d, there is at least one node on p that has two edges out of it. Let B be the closest such node to Y_d on p. Then $B \in \text{An}(Y_d, \mathcal{D})$. If any node on $p(B, Y_d)$ is in $\mathbf{X} \cup \mathbf{Y_N} \cup \mathbf{Z_N}$, then (ii) holds.

Hence, assume no node on $p(B, Y_d)$ is in $\mathbf{X} \cup \mathbf{Y_N} \cup \mathbf{Z_N}$. Then $B \notin \mathbf{X} \cup \mathbf{Y_N} \cup \mathbf{Z_N}$. Since $B \notin \mathbf{Z_N}$, $\mathbf{Z_N} \supseteq \text{An}(\mathbf{X} \cup \mathbf{Y}, \mathcal{D}) \setminus (\text{De}(\mathbf{X}, \mathcal{D}) \cup \mathbf{Y})$, it follows that $B \notin \text{An}(\mathbf{X} \cup \mathbf{Y}, \mathcal{D}) \setminus (\text{De}(\mathbf{X}, \mathcal{D}) \cup \mathbf{Y})$. Additionally, since $B \in \text{An}(Y_d, \mathcal{D})$, $B \in \text{An}(\mathbf{X} \cup \mathbf{Y}, \mathcal{D})$. Combining $B \notin \text{An}(\mathbf{X} \cup \mathbf{Y}, \mathcal{D}) \setminus (\text{De}(\mathbf{X}, \mathcal{D}) \cup \mathbf{Y})$ and $B \in \text{An}(\mathbf{X} \cup \mathbf{Y}, \mathcal{D})$ implies $B \in \text{De}(\mathbf{X}, \mathcal{D}) \cup \mathbf{Y}$. Furthermore, $B \notin \mathbf{X} \cup \mathbf{Y_N}$ and $\mathbf{Y_D} \subseteq \text{De}(\mathbf{X}, \mathcal{D}) \setminus \mathbf{X}$, so $B \in \text{De}(\mathbf{X}, \mathcal{D}) \setminus \mathbf{X}$. But $B \in \text{An}(Y_d, \mathcal{D})$ through $p(B, Y_d)$ on which no other node is in $\mathbf{X}$, so $B \in \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D})$. Now, $-p(B, A)$ is a path out of B, where $B \in \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D})$ and $A \notin \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D})$ by (i). Using the same reasoning as in (a), $p(A, B)$ is blocked by $(\mathbf{X} \cup \mathbf{Y_N} \cup \mathbf{Z_N}) \setminus \{A\}$. Hence, p is also blocked by $(\mathbf{X} \cup \mathbf{Y_N} \cup \mathbf{Z_N}) \setminus \{A\}$.

(iii) By (ii) every non-causal path from $\mathbf{Z_D}$ to $\mathbf{Y_D}$ is blocked by $\mathbf{X} \cup \mathbf{Y_N} \cup \mathbf{Z_N}$. Additionally, $\mathbf{Z_D} \cap \text{Forb}(\mathbf{X}, \mathbf{Y}, \mathcal{D}) = \emptyset$ and $\mathbf{Z_D} \subseteq \text{De}(\mathbf{X}, \mathcal{D})$, so no node in $\mathbf{Z_D}$ can have a proper causal path to $\mathbf{Y_D}$ in $\mathcal{D}$. Hence, any causal path p from $\mathbf{Z_D}$ to $\mathbf{Y_D}$ in $\mathcal{D}$, must be non-proper with respect to $\mathbf{X}$, that is, p has to contain a node in $\mathbf{X}$ as a non-collider. Hence, every causal path from $\mathbf{Z_D}$ to $\mathbf{Y_D}$ is also blocked by $\mathbf{X} \cup \mathbf{Y_N} \cup \mathbf{Z_N}$ in $\mathcal{D}$.

(iv) Follows directly from $\mathbf{Z_N} \cap \text{De}(\mathbf{X}, \mathcal{D}) = \emptyset$ and (ii) for $\mathbf{Y_N} = \emptyset$.

(v) Follows directly from (ii).

(vi) Since $\mathcal{D}_{\overline{\mathbf{X}}\underline{\mathbf{Y_N} \cup \mathbf{Z_N}}}$ does not contain edges into $\mathbf{X}$, all paths from $\mathbf{Y_N} \cup \mathbf{Z_N}$ to $\mathbf{Y_D}$ in $\mathcal{D}_{\overline{\mathbf{X}}\underline{\mathbf{Y_N} \cup \mathbf{Z_N}}}$ that contain a collider are blocked by $\mathbf{X}$. Hence, it is enough to prove that $\mathbf{X}$ blocks any path from $\mathbf{Y_N} \cup \mathbf{Z_N}$ to $\mathbf{Y_D}$ in $\mathcal{D}_{\overline{\mathbf{X}}\underline{\mathbf{Y_N} \cup \mathbf{Z_N}}}$ that does not contain a collider. Let $r = \langle A, \dots, Y_d \rangle$, $A \in \mathbf{Y_N} \cup \mathbf{Z_N}$, $Y_d \in \mathbf{Y_D}$ be an arbitrary path from $\mathbf{Y_N} \cup \mathbf{Z_N}$ to $\mathbf{Y_D}$ in $\mathcal{D}_{\overline{\mathbf{X}}\underline{\mathbf{Y_N} \cup \mathbf{Z_N}}}$ that does not contain a collider. Since there are no edges out of $\mathbf{Y_N} \cup \mathbf{Z_N}$ in $\mathcal{D}_{\overline{\mathbf{X}}\underline{\mathbf{Y_N} \cup \mathbf{Z_N}}}$, it follows that r is into A and that r does not contain non-colliders that are in $\mathbf{Y_N} \cup \mathbf{Z_N}$. Let r' in $\mathcal{D}$ be the path constituted by the same sequence of nodes as r in $\mathcal{D}_{\overline{\mathbf{X}}\underline{\mathbf{Y_N} \cup \mathbf{Z_N}}}$. Since removing edges cannot d-connect blocked paths, it is enough to prove that r' is blocked by $\mathbf{X}$ in $\mathcal{D}$. Since r' is into A, r' is a non-causal path from $\mathbf{Y_N} \cup \mathbf{Z_N}$ to $\mathbf{Y_D}$, so by (ii), r' is blocked by $(\mathbf{X} \cup \mathbf{Y_N} \cup \mathbf{Z_N}) \setminus \{A\}$ in $\mathcal{D}$. Since $A \in \mathbf{Y_N} \cup \mathbf{Z_N}$ and $(\mathbf{Y_N} \cup \mathbf{Z_N}) \cap \mathbf{X} = \emptyset$, it follows that $A \notin \mathbf{X}$. Then r' is blocked by $\mathbf{X} \cup ((\mathbf{Y_N} \cup \mathbf{Z_N}) \setminus \{A\})$. Furthermore, since r does not contain non-colliders in $\mathbf{Y_N} \cup \mathbf{Z_N}$, the same is true for r', so r' is blocked by $\mathbf{X}$.

(vii) Since $\mathcal{D}_{\overline{\mathbf{X}}}$ does not contain edges into $\mathbf{X}$, all paths from $\mathbf{X}$ to $\mathbf{Z_N}$ in $\mathcal{D}_{\overline{\mathbf{X}}}$ are out of $\mathbf{X}$. Since $\mathbf{Z_N} \cap \text{De}(\mathbf{X}, \mathcal{D}) = \emptyset$ and $\text{De}(\mathbf{X}, \mathcal{D}) \supseteq \text{De}(\mathbf{X}, \mathcal{D}_{\overline{\mathbf{X}}})$, all paths from $\mathbf{X}$ to $\mathbf{Z_N}$ in $\mathcal{D}_{\overline{\mathbf{X}}}$ contain at least one collider. The closest collider to $\mathbf{X}$ on any such path is then in $\text{De}(\mathbf{X}, \mathcal{D}_{\overline{\mathbf{X}}})$. Since $\mathbf{Y_N} \cap \text{De}(\mathbf{X}, \mathcal{D}) = \emptyset$ and $\text{De}(\mathbf{X}, \mathcal{D}) \supseteq \text{De}(\mathbf{X}, \mathcal{D}_{\overline{\mathbf{X}}})$, this path is blocked by $\mathbf{Y_N}$ in $\mathcal{D}_{\overline{\mathbf{X}}}$. ■

References

Ayesha R. Ali, Thomas S. Richardson, and Peter Spirtes. Markov equivalence for ancestral graphs. *Ann. Stat.*, 37:2808–2837, 2009.

Elias Bareinboim, Jin Tian, and Judea Pearl. Recovering from selection bias in causal and statistical inference. In *Proceedings of AAAI 2014*, pages 2410–2416, 2014.

David M. Chickering. Optimal structure identification with greedy search. *J. Mach. Learn. Res.*, 3:507–554, 2002.

Tom Claassen, Joris Mooij, and Tom Heskes. Learning sparse causal models is not NP-hard. In *Proceedings of UAI 2013*, pages 172–181, 2013.

Diego Colombo and Marloes H. Maathuis. Order-independent constraint-based causal structure learning. *J. Mach. Learn. Res.*, 15:3741–3782, 2014.

Diego Colombo, Marloes H. Maathuis, Markus Kalisch, and Thomas S. Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *Ann. Stat.*, 40:294–321, 2012.

Juan D. Correa and Elias Bareinboim. Causal effect identification by adjustment under confounding and selection biases. In *Proceedings of AAAI 2017*, pages 3740–3746, 2017.

Xavier De Luna, Ingeborg Waernbaum, and Thomas S. Richardson. Covariate selection for the nonparametric estimation of an average treatment effect. *Biometrika*, 98(4):861–875, 2011.

Doris Entner, Patrik O. Hoyer, and Peter Spirtes. Data-driven covariate selection for nonparametric estimation of causal effects. In *Proceedings of AISTATS 2013*, pages 256–264, 2013.

Benjamin Frot, Preetam Nandy, and Marloes H. Maathuis. Learning directed acyclic graphs with hidden variables via latent Gaussian graphical model selection. arXiv preprint arXiv:1708.01151, 2018.

Sander Greenland, Judea Pearl, and James M. Robins. Causal diagrams for epidemiologic research. *Epidemiology*, 10(1):37–48, 1999.

Hui Guo and Philip A. Dawid. Sufficient covariates and linear propensity analysis. In *Proceedings of AISTATS 2010*, pages 281–288, 2010.

Jinyong Hahn. On the role of the propensity score in efficient semiparametric estimation of average treatment effects. *Econometrica*, pages 315–331, 1998.

Jinyong Hahn. Functional restriction and efficiency in causal inference. *Rev. Econ. Stat.*, 86 (1):73–76, 2004.

Christina Heinze-Deml, Marloes H. Maathuis, and Nicolai Meinshausen. Causal structure learning. *Annu. Rev. Stat. Appl.*, 5:371–391, 2017.

Markus Kalisch, Martin Mächler, Diego Colombo, Marloes H. Maathuis, and Peter Bühlmann. Causal inference using graphical models with the R package pcalg. *J. Stat. Softw.*, 47(11):1–26, 2012.

Jan TA. Koster. Marginalizing and conditioning in graphical models. *Bernoulli*, 8(6):817–840, 2002.

Manabu Kuroki and Zhihong Cai. Selection of identifiability criteria for total effects by using path diagrams. In *Proceedings of UAI 2004*, pages 333–340, 2004.

Manabu Kuroki and Masami Miyakawa. Covariate selection for estimating the causal effect of control plans by using causal diagrams. *J. Roy. Stat. Soc. B*, 65(1):209–222, 2003.

Steffen L. Lauritzen and David J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *J. Roy. Stat. Soc. B*, 50(2): 157–224, 1988.

Steffen L. Lauritzen, Philip A. Dawid, Birgitte N. Larsen, and Hans-Georg Leimer. Independence properties of directed Markov fields. *Networks*, 20(5):491–505, 1990.

Marloes H. Maathuis and Diego Colombo. A generalized back-door criterion. *Ann. Stat.*, 43:1060–1088, 2015.

Marloes H. Maathuis, Markus Kalisch, and Peter Bühlmann. Estimating high-dimensional intervention effects from observational data. *Ann. Stat.*, 37:3133–3164, 2009.

Marloes H. Maathuis, Diego Colombo, Markus Kalisch, and Peter Bühlmann. Predicting causal effects in large-scale systems from observational data. *Nat. Methods*, 7:247–248, 2010.

Daniel Malinsky and Peter Spirtes. Estimating bounds on causal effects in high-dimensional and possibly confounded systems. *Int. J. of Approx. Reason.*, 88:371–384, 2017.

Kanti V. Mardia, John T. Kent, and John M. Bibby. *Multivariate Analysis (Probability and Mathematical Statistics)*. Academic Press London, 1980.

Christopher Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of UAI 1995*, pages 403–410, 1995.

Preetam Nandy, Marloes H. Maathuis, and Thomas S. Richardson. Estimating the effect of joint interventions from observational data in sparse high-dimensional settings. *Ann. Stat.*, 45(2):647–674, 2017.

Preetam Nandy, Alain Hauser, and Marloes H. Maathuis. High-dimensional consistency in score-based and hybrid structure learning. *Ann. Stat.*, 2018. To appear.

Judea Pearl. Comment: Graphical models, causality and intervention. *Stat. Sci.*, 8:266–269, 1993.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, New York, NY, second edition, 2009.

Judea Pearl and Azaria Paz. Confounding equivalence in causal inference. *J. Causal Infer.*, 2(1):75–93, 2014.

Emilija Perković, Johannes Textor, Markus Kalisch, and Marloes H. Maathuis. A complete generalized adjustment criterion. In *Proceedings of UAI 2015*, pages 682–691, 2015.

Emilija Perković, Markus Kalisch, and Marloes H. Maathuis. Interpreting and using CPDAGs with background knowledge. In *Proceedings of UAI 2017*, 2017.

Thomas S. Richardson. Markov properties for acyclic directed mixed graphs. *Scand. J. Statist.*, 30:145–157, 2003.

Thomas S. Richardson and Peter Spirtes. Ancestral graph Markov models. *Ann. Stat.*, 30: 962–1030, 2002.

James M. Robins. A new approach to causal inference in mortality studies with a sustained exposure period-application to control of the healthy worker survivor effect. *Math. Mod.*, 7:1393–1512, 1986.

Donald Rubin. Author's reply. *Stat. Med.*, 27:2741–2742, 2008.

Ross D. Shachter. Bayes-ball: The rational pastime. In *Proceedings of UAI 1998*, pages 480–487, 1998.

Ilya Shpitser. Appendum to "On the validity of covariate adjustment for estimating causal effects". Personal communication, 2012.

Ilya Shpitser and Judea Pearl. Identification of joint interventional distributions in recursive semi-Markovian causal models. In *Proceedings of AAAI 2006*, pages 1219–1226, 2006.

Ilya Shpitser, Tyler VanderWeele, and James M. Robins. On the validity of covariate adjustment for estimating causal effects. In *Proceedings of UAI 2010*, pages 527–536, 2010.

Ian Shrier. Letter to the editor. *Stat. Med.*, 27:2740–2741, 2008.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT Press, Cambridge, MA, second edition, 2000.

Johannes Textor, Benito van der Zander, Mark S. Gilthorpe, Maciej Liśkiewicz, and George TH Ellison. Robust causal inference using directed acyclic graphs: the R package 'dagitty'. *Int. J. Epidemiol.*, 45(6):1887–1894, 2016.

Jin Tian and Judea Pearl. A general identification condition for causal effects. In *Proceedings of AAAI 2002*, pages 567–573, 2002.

Benito van der Zander and Maciej Liśkiewicz. Separators and adjustment sets in Markov equivalent DAGs. In *Proceedings of AAAI 2016*, pages 3315–3321, 2016.

Benito van der Zander, Maciej Liśkiewicz, and Johannes Textor. Constructing separators and adjustment sets in ancestral graphs. In *Proceedings of UAI 2014*, pages 907–916, 2014.

Benito van der Zander, Maciej Liśkiewicz, and Johannes Textor. Separators and adjustment sets in causal graphs: Complete criteria and an algorithmic framework. arXiv preprint arXiv:1803.00116, 2018.

Tyler J. VanderWeele and Ilya Shpitser. A new criterion for confounder selection. *Biometrics*, 67(4):1406–1413, 2011.

Stephen G. West and Tobias Koch. Restoring causal analysis to structural equation modeling. *Struct. Equ. Modeling*, 21:161–166, 2014.

Daniel Westreich and Sander Greenland. The table 2 fallacy: Presenting and interpreting confounder and modifier coefficients. *Am. J. Epidemiol.*, 177:292–298, 2013.

Sewall Wright. Correlation and causation. *J. Agric. Res.*, 20(7):557–585, 1921.

Jiji Zhang. *Causal Inference and Reasoning in Causally Insufficient Systems*. PhD thesis, Carnegie Mellon University, 2006.

Jiji Zhang. Causal reasoning with ancestral graphs. *J. Mach. Learn. Res.*, 9:1437–1474, 2008a.

Jiji Zhang. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artif. Intell.*, 172:1873–1896, 2008b.

Journal of Machine Learning Research 18 (2018) 1-51 Submitted 8/16; Revised 5/17; Published 6/18

Katyusha: The First Direct Acceleration of Stochastic Gradient Methods

Zeyuan Allen-Zhu* zeyuan@csail.mit.edu
Microsoft Research AI
Redmond, WA 98052, USA

Editor: Leon Bottou

Abstract

Nesterov's momentum trick is famously known for accelerating gradient descent, and has been proven useful in building fast iterative algorithms. However, in the stochastic setting, counterexamples exist and prevent Nesterov's momentum from providing similar acceleration, even if the underlying problem is convex and finite-sum.

We introduce `Katyusha`, a direct, primal-only stochastic gradient method to fix this issue. In convex finite-sum stochastic optimization, `Katyusha` has an optimal accelerated convergence rate, and enjoys an optimal parallel linear speedup in the mini-batch setting.

The main ingredient is *Katyusha momentum*, a novel "negative momentum" on top of Nesterov's momentum. It can be incorporated into a variance-reduction based algorithm and speed it up, both in terms of *sequential and parallel* performance. Since variance reduction has been successfully applied to a growing list of practical problems, our paper suggests that in each of such cases, one could potentially try to give Katyusha a hug.

1. Introduction

In large-scale machine learning, the number of data examples is usually very large. To search for the optimal solution, one often uses *stochastic gradient methods* which only require one (or a small batch of) random example(s) per iteration in order to form an *estimator* of the full gradient.

While full-gradient based methods can enjoy an *accelerated* (and optimal) convergence rate if Nesterov's momentum trick is used (Nesterov, 1983, 2004, 2005), theory for stochastic gradient methods are generally lagging behind and less is known for their acceleration.

At a high level, momentum is *dangerous* if stochastic gradients are present. If some gradient estimator is very inaccurate, then adding it to the momentum and moving further in this direction (for every future iteration) may hurt the convergence performance. In other words, when naively equipped with momentum, stochastic gradient methods are "very prune to error accumulation" (Konečnỳ et al., 2016) and do *not* yield accelerated convergence rates in general.[1]

In this paper, we show that at least for convex optimization purposes, such an issue can be solved with a novel "negative momentum" that can be added on top of momentum. We obtain accelerated and the first optimal convergence rates for stochastic gradient methods. As one of our side results, under this "negative momenmtum," our new method enjoys a linear speedup in the parallel (i.e., mini-match) setting. We hope our new insight could potentially deepen our understanding to the theory of accelerated methods.

Problem Definition. Consider the following composite convex minimization problem

$$\min_{x\in\mathbb{R}^d}\left\{F(x)\stackrel{\text{def}}{=}f(x)+\psi(x)\stackrel{\text{def}}{=}\frac{1}{n}\sum_{i=1}^n f_i(x)+\psi(x)\right\}\ . \tag{1.1}$$

Here, $f(x)=\frac{1}{n}\sum_{i=1}^n f_i(x)$ is a convex function that is a finite average of n convex, smooth functions $f_i(x)$, and $\psi(x)$ is convex, lower semicontinuous (but possibly non-differentiable) function, sometimes referred to as the *proximal* function. We mostly focus on the case when $\psi(x)$ is σ-strongly convex and each $f_i(x)$ is L-smooth. (Both these assumptions can be removed and we shall discuss that later.) We look for approximate minimizers $x\in\mathbb{R}^d$ satisfying $F(x)\le F(x^*)+\varepsilon$, where $x^*\in\arg\min_x\{F(x)\}$.

Problem (1.1) arises in many places in machine learning, statistics, and operations research. All convex *regularized empirical risk minimization (ERM)* problems such as Lasso, SVM, Logistic Regression, fall into this category (see Section 1.3). Efficient stochastic methods for Problem (1.1) have also inspired stochastic algorithms for neural nets (Johnson and Zhang, 2013; Allen-Zhu and Hazan, 2016a; Lei et al., 2017) as well as SVD, PCA, and CCA (Garber et al., 2016; Allen-Zhu and Li, 2016, 2017b).

We summarize the history of stochastic gradient methods for Problem (1.1) in three eras.

The First Era: Stochastic Gradient Descent (SGD).

Recall that stochastic gradient methods iteratively perform the following update

$$\text{stochastic gradient iteration:}\qquad x_{k+1}\leftarrow\arg\min_{y\in\mathbb{R}^d}\left\{\frac{1}{2\eta}\|y-x_k\|_2^2+\langle\widetilde{\nabla}_k,y\rangle+\psi(y)\right\}\ ,$$

where η is the step length and $\widetilde{\nabla}_k$ is a random vector satisfying $\mathbb{E}[\widetilde{\nabla}_k]=\nabla f(x_k)$ and is referred to as the *gradient estimator*. If the proximal function $\psi(y)$ equals zero, the update reduces to $x_{k+1}\leftarrow x_k-\eta\widetilde{\nabla}_k$. A popular choice for the gradient estimator is to set $\widetilde{\nabla}_k=\nabla f_i(x_k)$ for some random index $i\in[n]$ per iteration, and methods based on this choice are known as *stochastic gradient descent (SGD)* (Zhang, 2004; Bottou). Since computing $\nabla f_i(x)$ is usually n times faster than that of $\nabla f(x)$, SGD enjoys a low per-iteration cost as compared to full-gradient methods; however, SGD cannot converge at a rate faster than $1/\varepsilon$ even if $F(\cdot)$ is strongly convex and smooth.

∗. The arXiv version of this paper can be found at `http://arxiv.org/abs/1603.05953`, and may include future revisions.

1. In practice, experimentalists have observed that momentums could sometimes help if stochastic gradient iterations are used. However, the so-obtained methods (1) sometimes fail to converge in an accelerated rate, (2) become unstable and hard to tune, and (3) have no support theory behind them. See Section 7.1 for an experiment illustrating that, even for convex stochastic optimization.

The Second Era: Variance Reduction Gives Faster Convergence.

The convergence rate of SGD can be further improved with the so-called *variance-reduction* technique, first proposed by Schmidt et al. (2013) (solving a sub-case of Problem (1.1)) and then followed by many others (Zhang et al., 2013; Mahdavi et al., 2013; Johnson and Zhang, 2013; Shalev-Shwartz and Zhang, 2013; Shalev-Shwartz, 2016; Shalev-Shwartz and Zhang, 2012; Xiao and Zhang, 2014; Defazio et al., 2014; Mairal, 2015; Allen-Zhu and Yuan, 2016). In these cited results, the authors have shown that SGD converges much faster if one makes a better choice of the gradient estimator $\widetilde{\nabla}_k$ so that its variance reduces as k increases. One way to choose this estimator can be described as follows (Johnson and Zhang, 2013; Zhang et al., 2013). Keep a *snapshot* vector $\widetilde{x} = x_k$ that is updated once every m iterations (where m is some parameter usually around $2n$), and compute the full gradient $\nabla f(\widetilde{x})$ only for such snapshots. Then, set

$$\widetilde{\nabla}_k = \nabla f_i(x_k) - \nabla f_i(\widetilde{x}) + \nabla f(\widetilde{x}) \ . \tag{1.2}$$

This choice of gradient estimator ensures that its variance approaches to zero as k grows. Furthermore, the number of stochastic gradients (i.e., the number of computations of $\nabla f_i(x)$ for some i) required to reach an ε-approximate minimizer of Problem (1.1) is only $O\big(\big(n+\frac{L}{\sigma}\big)\log\frac{1}{\varepsilon}\big)$. Since it is often denoted by $\kappa \stackrel{\text{def}}{=} L/\sigma$ the condition number of the problem, we rewrite the above iteration complexity as $O\big((n+\kappa)\log\frac{1}{\varepsilon}\big)$.

Unfortunately, the iteration complexities of all known variance-reduction based methods have a linear dependence on κ. It was an open question regarding how to obtain an *accelerated* stochastic gradient method with an optimal $\sqrt{\kappa}$ dependency.

The Third Era: Acceleration Gives Fastest Convergence.

This open question was partially solved recently by the APPA (Frostig et al., 2015) and Catalyst (Lin et al., 2015) reductions, both based on an outer-inner loop structure first proposed by Shalev-Shwartz and Zhang (2014). We refer to both of them as Catalyst in this paper. Catalyst solves Problem (1.1) using $O\big(\big(n+\sqrt{n\kappa}\big)\log\kappa\log\frac{1}{\varepsilon}\big)$ stochastic gradient iterations, through a logarithmic number of calls to a variance-reduction method.[2] However, Catalyst is still imperfect for the following reasons:

- Optimality. Catalyst does not match the optimal $\sqrt{\kappa}$ dependence (Woodworth and Srebro, 2016) and has an extra $\log\kappa$ factor. It yields suboptimal rate $\frac{\log^4 T}{T^2}$ if the objective is not strongly convex or is non-smooth; and it yields suboptimal rate $\frac{\log^4 T}{T}$ if the objective is both non-strongly convex and non-smooth.[3]
- Practicality. To the best of our knowledge, Catalyst is not very practical since each of its inner iterations needs to be very accurately executed. This makes the stopping criterion hard to be tuned, and makes Catalyst sometimes run slower than non-accelerated variance-reduction methods. We have also confirmed this in our experiments.
- Parallelism. To the best of our knowledge, Catalyst does not give competent parallel performance (see Section 1.2). If $b \in \{1,\dots,n\}$ stochastic gradients (instead of one) are computed in each iteration, the number of iterations of Catalyst reduces by $O(\sqrt{b})$. In contrast, the best parallel speedup one can hope for is "linear speedup": that is, to reduce the number of iterations by a factor of $O(b)$ for $b \le \sqrt{n}$.
- Generality. To the best of our knowledge, being a reduction-based method, Catalyst does not seem to support non-Euclidean norm smoothness (see Section 1.2).

Another acceleration method by Lan and Zhou (2015) is based on a primal-dual analysis that also has suboptimal convergence rates like Catalyst. Their method requires n times more storage compared with Catalyst for solving Problem (1.1).

In sum, it is desirable and also an open question to develop a *direct*, *primal-only*, and *optimal* accelerated stochastic gradient method without using reductions. This could have both theoretical and practical impacts to the problems that fall into the general framework of (1.1), and potentially deepen our understanding to acceleration in stochastic settings.

1.1 Our Main Results and High-Level Ideas

We develop a direct, accelerated stochastic gradient method `Katyusha` for Problem (1.1) in

$$\boxed{O\big(\big(n+\sqrt{n\kappa}\big)\log(1/\varepsilon)\big) \text{ stochastic gradient iterations (see Theorem 2.1).}}$$

This gives both optimal dependency on κ and on ε which was not obtained before for stochastic gradient methods. In addition, if $F(\cdot)$ is non-strongly convex (non-SC), `Katyusha` converges to an ε-minimizer in

$$\boxed{O\big(n\log(1/\varepsilon)+\sqrt{nL/\varepsilon}\,\big) \text{ stochastic gradient iterations (see Corollary 3.7).}}$$

This gives an optimal $\varepsilon \propto \frac{n}{T^2}$ rate where in contrast Catalyst has rate $\varepsilon \propto \frac{n\log^4 T}{T^2}$. The lower bound from Woodworth and Srebro (2016) is $\Omega\big(n+\sqrt{nL/\varepsilon}\big)$.

Our Algorithm. If ignoring the proximal term $\psi(\cdot)$ and viewing it as zero, our `Katyusha` method iteratively perform the following updates for $k = 0, 1, \dots$:

- $x_{k+1} \leftarrow \tau_1 z_k + \tau_2 \widetilde{x} + (1-\tau_1-\tau_2) y_k$; (so $x_{k+1} = y_k + \tau_1(z_k - y_k) + \tau_2(\widetilde{x} - y_k)$)
- $\widetilde{\nabla}_{k+1} \leftarrow \nabla f(\widetilde{x}) + \nabla f_i(x_{k+1}) - \nabla f_i(\widetilde{x})$ where i is a random index in $[n]$;
- $y_{k+1} \leftarrow x_{k+1} - \frac{1}{3L}\widetilde{\nabla}_{k+1}$, and
- $z_{k+1} \leftarrow z_k - \alpha\widetilde{\nabla}_{k+1}$.

Above, $\widetilde{x}$ is a snapshot point which is updated every m iterations, $\widetilde{\nabla}_{k+1}$ is the gradient estimator defined in the same way as (1.2), $\tau_1, \tau_2 \in [0,1]$ are two momentum parameters, and α is a parameter that is equal to $\frac{1}{3\tau_1 L}$. The reason for keeping three vector sequences (x_k, y_k, z_k) is a common ingredient that can be found in all existing accelerated methods.[4]

Our New Technique – Katyusha Momentum. The most interesting ingredient of `Katyusha` is the novel choice of x_{k+1} which is a convex combination of y_k, z_k, and $\widetilde{x}$. Our theory suggests the parameter choices $\tau_2 = 0.5$ and $\tau_1 = \min\{\sqrt{n\sigma/L}, 0.5\}$ and they work well in practice too. To explain this novel combination, let us recall the classical "momentum" view of accelerated methods.

2. Note that $n+\sqrt{n\kappa}$ is always less than $O(n+\kappa)$.
3. Obtaining *optimal* rates is one of the main goals in optimization and machine learning. For instance, obtaining the optimal $1/T$ rate for online learning was a very meaningful result, even though the $\log T/T$ rate was known (Hazan and Kale, 2014; Rakhlin et al., 2012).
4. One can of course rewrite the algorithm and keep track of only two vectors per iteration during implementation. This will make the algorithm statement less clean so we refrain from doing so in this paper.

In a classical accelerated gradient method, x_{k+1} is only a convex combination of y_k and z_k (or equivalently, $\tau_2 = 0$ in our formulation). At a high level, z_k plays the role of "momentum" which adds a weighted sum of the gradient history into y_{k+1}. As an illustrative example, suppose $\tau_2 = 0$, $\tau_1 = \tau$, and $x_0 = y_0 = z_0$. Then, one can compute that

$$y_k = \begin{cases} x_0 - \frac{1}{3L}\widetilde{\nabla}_1, & k=1; \\ x_0 - \frac{1}{3L}\widetilde{\nabla}_2 - \big((1-\tau)\frac{1}{3L}+\tau\alpha\big)\widetilde{\nabla}_1, & k=2; \\ x_0 - \frac{1}{3L}\widetilde{\nabla}_3 - \big((1-\tau)\frac{1}{3L}+\tau\alpha\big)\widetilde{\nabla}_2 - \big((1-\tau)^2\frac{1}{3L}+(1-(1-\tau)^2)\alpha\big)\widetilde{\nabla}_1, & k=3. \end{cases}$$

Since α is usually much larger than $1/3L$, the above recursion suggests that the contribution of a fixed gradient $\widetilde{\nabla}_t$ gradually increases as time goes. For instance, the weight on $\widetilde{\nabla}_1$ is increasing because $\frac{1}{3L} < \big((1-\tau)\frac{1}{3L}+\tau\alpha\big) < \big((1-\tau)^2\frac{1}{3L}+(1-(1-\tau)^2)\alpha\big)$. This is known as "momentum" which is at the heart of all accelerated first-order methods.

In `Katyusha`, we put a "magnet" around $\widetilde{x}$, where we choose $\widetilde{x}$ to be essentially "the average x_t of the most recent n iterations". Whenever we compute the next x_{k+1}, it will be attracted by the magnet $\widetilde{x}$ with weight $\tau_2 = 0.5$. This is a strong magnet: it ensures that x_{k+1} is not too far away from $\widetilde{x}$ so the gradient estimator remains "accurate enough". This can be viewed as a "negative momentum" component, because the magnet retracts x_{k+1} back to $\widetilde{x}$ and this can be understood as "counteracting a fraction of the positive momentum incurred from earlier iterations."

We call it the Katyusha momentum.

This summarizes the high-level idea behind our `Katyusha` method. We remark here if $\tau_1 = \tau_2 = 0$, `Katyusha` becomes almost identical to SVRG (Johnson and Zhang, 2013; Zhang et al., 2013) which is a variance-reduction based method.

1.2 Our Side Results

Parallelism / Mini-batch. Instead of using a single $\nabla f_i(\cdot)$ per iteration, for any stochastic gradient method, one can replace it with the average of b stochastic gradients $\frac{1}{b}\sum_{i\in S}\nabla f_i(\cdot)$, where S is a random subset of $[n]$ with cardinality b. This is known as the *mini-batch* technique and it allows the stochastic gradients to be computed in a distributed manner, using up to b processors.

Our `Katyusha` method trivially extends to this mini-batch setting. For instance, at least for $b \in \{1, 2, \ldots, \lceil\sqrt{n}\rceil\}$, `Katyusha` enjoys a *linear speed-up* in the parallel running time. In other words, if ignoring communication overhead,

> `Katyusha` can be distributed to $b \le \sqrt{n}$ machines with a parallel speed-up factor b.

In contrast, to the best of our knowledge, without any additional assumption, (1) non-accelerated methods such as SVRG or SAGA are not known to enjoy any parallel speed-up; (2) Catalyst enjoys a parallel speed-up factor of only $\sqrt{b}$. Details are in Section 5.

Non-Uniform Smoothness. If each $f_i(\cdot)$ has a possibly different smooth parameter L_i and $\overline{L} = \frac{1}{n}\sum_{i=1}^n L_i$, then an naive implementation of `Katyusha` only gives a complexity that depends on $\max_i L_i$ but not $\overline{L}$. In such a case, we can select the random index $i \in [n]$ with probability proportional to L_i per iteration to slightly improve the total running time.

Furthermore, suppose $f(x) = \frac{1}{n}\sum_{i=1}^n f_i(x)$ is smooth with parameter L, it satisfies $\overline{L} \in [L, nL]$. One can ask whether or not L influences the performance of `Katyusha`. We show that, in the mini-batch setting when b is large, the total complexity becomes a function on L as opposed to $\overline{L}$. The details are in Section 5.

A Precise Statement. Taking into account both the mini-batch parameter b and the non-uniform smoothness parameters L and $\overline{L}$, we show `Katyusha` solves Problem (1.1) in

$$\boxed{O\Big(\big(n + b\sqrt{L/\sigma} + \sqrt{n\overline{L}/\sigma}\big)\cdot\log\frac{1}{\varepsilon}\Big) \text{ stochastic gradient computations (see Theorem 5.2)}}$$

Non-Euclidean Norms. If the smoothness of each $f_i(x)$ is with respect to a non-Euclidean norm (such as the well known ℓ_1 norm case over the simplex), our main result still holds. Our update on the y_{k+1} side becomes the non-Euclidean norm gradient descent, and our update on the z_{k+1} side becomes the non-Euclidean norm mirror descent. We include such details in Section 6. To the best of our knowledge, most known accelerated methods (including Catalyst, AccSDCA and APCG) do not work with non-Euclidean norms. SPDC can be revised to work with non-Euclidean norms, see (Allen-Zhu et al., 2016b).

Remark on Katyusha Momentum Weight τ_2. To provide the simplest proof, we choose $\tau_2 = 1/2$ which also works well in practice. Our proof trivially generalizes to all constant values $\tau_2 \in (0, 1)$, and it could be beneficial to tune τ_2 for different datasets. However, for a stronger comparison, in our experiments we refrain from tuning τ_2: by fixing $\tau_2 = 1/2$ and without increasing parameter tuning difficulties, `Katyusha` already outperforms most of the state-of-the-arts.

In the mini-batch setting, it turns out the best theoretical choice is essentially $\tau_2 = \frac{1}{2b}$, where b is the size of the mini-batch. In other words, the larger the mini-batch size, the smaller weight we want to give to Katyusha momentum. This should be intuitive, because when $b = n$ we are almost in the full-gradient setting and do not need Katyusha momentum.

1.3 Applications: Optimal Rates for Empirical Risk Minimization

Suppose we are given n feature vectors $a_1, \ldots, a_n \in \mathbb{R}^d$ corresponding to n data samples. Then, the *empirical risk minimization (ERM)* problem is to study Problem (1.1) when each $f_i(x)$ is "rank-one" structured: $f_i(x) \stackrel{\text{def}}{=} g_i(\langle a_i, x\rangle)$ for some loss function $g_i : \mathbb{R} \to \mathbb{R}$. Slightly abusing notation, we write $f_i(x) = f_i(\langle a_i, x\rangle)$.[5] In such a case, Problem (1.1) becomes

$$\text{ERM:} \quad \min_{x\in\mathbb{R}^d}\Big\{F(x) \stackrel{\text{def}}{=} f(x) + \psi(x) \stackrel{\text{def}}{=} \tfrac{1}{n}\textstyle\sum_{i=1}^n f_i(\langle a_i, x\rangle) + \psi(x)\Big\} \ . \tag{1.3}$$

Without loss of generality, we assume each a_i has norm 1 because otherwise one can scale $f_i(\cdot)$ accordingly. As summarized for instance in Allen-Zhu and Hazan (2016b), there are four interesting cases of ERM problems, all can be written in the form of (1.3):

Case 1: $\psi(x)$ is σ-SC and $f_i(x)$ is L-smooth. Examples: ridge regression, elastic net;

Case 2: $\psi(x)$ is non-SC and $f_i(x)$ is L-smooth. Examples: Lasso, logistic regression;

Case 3: $\psi(x)$ is σ-SC and $f_i(x)$ is non-smooth. Examples: support vector machine;

5. Assuming "rank-one" simplifies the notations; all of the results stated in this subsection generalize to constant-rank structured functions $f_i(x)$.

Case 4: $\psi(x)$ is non-SC and $f_i(x)$ is non-smooth. Examples: ℓ_1-SVM.

Known Results. For all of the four ERM cases above, accelerated stochastic methods were introduced in the literature, most notably AccSDCA (Shalev-Shwartz and Zhang, 2014), APCG (Lin et al., 2014), SPDC (Zhang and Xiao, 2015). These methods have suboptimal convergence rates for Cases 2, 3 and 4. (In fact, they also have the suboptimal dependence on the condition number L/σ for Case 1.) The best known rate was $\frac{\log(1/\varepsilon)}{\sqrt{\varepsilon}}$, $\frac{\log(1/\varepsilon)}{\sqrt{\varepsilon}}$, or $\frac{\log(1/\varepsilon)}{\varepsilon}$ respectively for Cases 2, 3, or 4, and is a factor $\log(1/\varepsilon)$ worse than optimal (Woodworth and Srebro, 2016).

It is an open question to design a stochastic gradient method with optimal convergence for such problems. In particular, Dang and Lan (2014) provided an interesting attempt to remove such log factors but using a non-classical notion of convergence.[6]

Besides the log factor loss in the running time,[7] the aforementioned methods suffer from several other issues that most dual-based methods also suffer. First, they only apply to ERM problems but not to the more general Problem (1.1). Second, they require proximal updates with respect to the Fenchel conjugate $f_i^*(\cdot)$ which is sometimes unpleasant to work with. Third, their performances cannot benefit from the implicit strong convexity in $f(\cdot)$. All of these issues together make these methods sometimes even outperformed by primal-only non-accelerated ones, such as SAGA (Defazio et al., 2014) or SVRG (Johnson and Zhang, 2013; Zhang et al., 2013).

Our Results. `Katyusha` simultaneously closes the gap for all of the three classes of problems with the help from the optimal reductions developed in Allen-Zhu and Hazan (2016b). We obtain an ε-approximate minimizer for Case 2 in $O\big(n\log\frac{1}{\varepsilon}+\frac{\sqrt{nL}}{\sqrt{\varepsilon}}\big)$ iterations, for Case 3 in $O\big(n\log\frac{1}{\varepsilon}+\frac{\sqrt{n}}{\sqrt{\sigma\varepsilon}}\big)$ iterations, and for Case 4 in $O\big(n\log\frac{1}{\varepsilon}+\frac{\sqrt{n}}{\varepsilon}\big)$ iterations. None of the existing accelerated methods can lead to such optimal rates even if the optimal reductions are used.

Woodworth and Srebro (2016) proved the tightness of our results. They showed lower bounds $\Omega\big(n+\frac{\sqrt{nL}}{\sqrt{\varepsilon}}\big)$, $\Omega\big(n+\frac{\sqrt{n}}{\sqrt{\sigma\varepsilon}}\big)$, and $\Omega\big(n+\frac{\sqrt{n}}{\varepsilon}\big)$ for Cases 2, 3, and 4 respectively.[8]

1.4 Roadmap

- In Section 2, we state and prove our theorem on `Katyusha` for the strongly convex case.
- In Section 3, we apply `Katyusha` to non-strongly convex or non-smooth cases by reductions.
- In Section 4, we provide a *direct* algorithm `Katyusha`$^{\mathtt{ns}}$ for the non-strongly case.
- In Section 5, we generalize `Katyusha` to mini-batch and non-uniform smoothness.
- In Section 6, we generalize `Katyusha` to the non-Euclidean norm setting.
- In Section 7, we provide an empirical evaluation to illustrate the necessity of Katyusha momentum, and the practical performance of `Katyusha`.

1.5 Notations

Throughout this paper (except Section 6), we denote by $\|\cdot\|$ the Euclidean norm. We denote by $\nabla f(x)$ the full gradient of function f if it is differentiable, or any of its subgradients if f is only Lipschitz continuous. Recall some classical definitions on strong convexity (SC) and smoothness.

Definition 1.1 *For a convex function $f\colon \mathbb{R}^n \to \mathbb{R}$,*

- *f is σ-strongly convex if $\forall x,y\in\mathbb{R}^n$, it satisfies $f(y)\geq f(x)+\langle\nabla f(x),y-x\rangle+\frac{\sigma}{2}\|x-y\|^2$.*
- *f is L-smooth if $\forall x,y\in\mathbb{R}^n$, it satisfies $\|\nabla f(x)-\nabla f(y)\|\leq L\|x-y\|$.*

2. Katyusha in the Strongly Convex Setting

We formally introduce our `Katyusha` algorithm in Algorithm 1. It follows from our high-level description in Section 1.1, and we make several remarks here behind our specific design.

- `Katyusha` *is divided into epochs each consisting of m iterations. In theory, m can be anything linear in n. We let snapshot $\widetilde{x}$ be a weighted average of y_k in the most recent epoch.*

  $\widetilde{x}$ and $\widetilde{\nabla}_k$ correspond to a standard design on variance-reduced gradient estimators, called SVRG (Johnson and Zhang, 2013; Zhang et al., 2013). The practical recommendation is $m=2n$ (Johnson and Zhang, 2013). Our choice $\widetilde{\nabla}_k$ is independent from our acceleration techniques, and we expect our result continues to apply to other choices of gradient estimators. We choose $\widetilde{x}$ to be a weighted average, rather than the last or the uniform average, because it yields the tightest possible result.[9]

- *τ_1 and α are standard parameters already present in Nesterov's full-gradient method (Allen-Zhu and Orecchia, 2017).*

  We choose $\alpha=1/3\tau_1 L$ to present the simplest proof, and recall it was $\alpha=1/\tau_1 L$ in the original Nesterov's full-gradient method. (Any α that is constant factor smaller than $1/\tau_1 L$ works in theory, and we use $1/3$ to provide the simplest proof.) In practice, like other accelerated methods, it suffices to fix $\alpha=1/3\tau_1 L$ and only tune τ_1 and thus τ_1 is viewed as the learning rate.

6. Dang and Lan (2014) work in a primal-dual $\phi(x,y)$ formulation of Problem (1.1), and produce a primal-dual pair (x,y) so that for every fixed (u,v), the expectation $\mathbb{E}[\phi(x,v)-\phi(u,y)]\leq\varepsilon$. Unfortunately, to ensure x is an ε-approximate minimizer of Problem (1.1), one needs the stronger $\mathbb{E}[\max_{(u,v)}\phi(x,v)-\phi(u,y)]\leq\varepsilon$ to hold.
7. In fact, dual-based methods have to suffer from a log factor loss in the convergence rate. This is so because even for Case 1 of Problem (1.3), converting an ε-maximizer for the dual objective to the primal, one only obtains an $n\kappa\varepsilon$-minimizer on the primal objective. As a result, algorithms like APCG who directly work on the dual, algorithms like SPDC who maintain both primal and dual variables, and algorithms like RPDG (Lan and Zhou, 2015) that are primal-like but still use dual analysis, have to suffer from a log loss in the convergence rates.
8. More precisely, their lower bounds for Cases 3 and 4 are $\Omega\big(\min\big\{\frac{1}{\sigma\varepsilon},n+\frac{\sqrt{n}}{\sqrt{\sigma\varepsilon}}\big\}\big)$ and $\Omega\big(\min\big\{\frac{1}{\varepsilon^2},n+\frac{\sqrt{n}}{\varepsilon}\big\}\big)$. However, since the vanilla SGD requires $O(\frac{1}{\sigma\varepsilon})$ and $O(\frac{1}{\varepsilon^2})$ iterations for Cases 3 and 4, such lower bounds are matched by combining the best between `Katyusha` and SGD.
9. If one uses the uniform average, in theory, the algorithm needs to restart every a number of epochs (that is, by resetting $k=0$, $s=0$, and $x_0=y_0=z_0$); we refrain from doing so because we wish to provide a simple and direct algorithm. We can also use the last iterate, then the total complexity loses a factor $\log(L/\sigma)$. In practice, it was reported that even for SVRG, choosing average works better than choosing the last iterate (Allen-Zhu and Yuan, 2016).

Algorithm 1 `Katyusha`(x_0, S, σ, L)

$m \leftarrow 2n$; ◇ *epoch length*
$\tau_2 \leftarrow \frac{1}{2}$, $\tau_1 \leftarrow \min\{\frac{\sqrt{m\sigma}}{\sqrt{3L}}, \frac{1}{2}\}$, $\alpha \leftarrow \frac{1}{3\tau_1 L}$; ◇ *parameters*
$y_0 = z_0 = \widetilde{x}^0 \leftarrow x_0$; ◇ *initial vectors*
for $s \leftarrow 0$ **to** $S-1$ **do**
  $\mu^s \leftarrow \nabla f(\widetilde{x}^s)$; ◇ *compute the full gradient once every m iterations*
  for $j \leftarrow 0$ **to** $m-1$ **do**
    $k \leftarrow (sm)+j$;
    $x_{k+1} \leftarrow \tau_1 z_k + \tau_2 \widetilde{x}^s + (1-\tau_1-\tau_2) y_k$;
    $\widetilde{\nabla}_{k+1} \leftarrow \mu^s + \nabla f_i(x_{k+1}) - \nabla f_i(\widetilde{x}^s)$ where i is random from $\{1, 2, \ldots, n\}$;
    $z_{k+1} = \arg\min_z \{\frac{1}{2\alpha}\|z - z_k\|^2 + \langle \widetilde{\nabla}_{k+1}, z\rangle + \psi(z)\}$;
    Option I: $y_{k+1} \leftarrow \arg\min_y \{\frac{3L}{2}\|y - x_{k+1}\|^2 + \langle \widetilde{\nabla}_{k+1}, y\rangle + \psi(y)\}$;
    Option II: $y_{k+1} \leftarrow x_{k+1} + \tau_1(z_{k+1} - z_k)$ ◇ *we analyze only I but II also works*
  end for
  $\widetilde{x}^{s+1} \leftarrow \big(\sum_{j=0}^{m-1}(1+\alpha\sigma)^j\big)^{-1} \cdot \big(\sum_{j=0}^{m-1}(1+\alpha\sigma)^j \cdot y_{sm+j+1}\big)$; ◇ *compute snapshot $\widetilde{x}$*
end for
return $\widetilde{x}^S$.

- *The parameter τ_2 is our novel weight for the Katyusha momentum. Any constant in $(0,1)$ works for τ_2, and we simply choose $\tau_2 = 1/2$ for our theoretical and experimental results.*

We state our main theorem for `Katyusha` as follows:

Theorem 2.1 *If each $f_i(x)$ is convex, L-smooth, and $\psi(x)$ is σ-strongly convex in the above Problem (1.1), then* `Katyusha`(x_0, S, σ, L) *satisfies*

$$\mathbb{E}\big[F(\widetilde{x}^S)\big] - F(x^*) \le \begin{cases} O\Big(\big(1+\sqrt{\sigma/(3Lm)}\big)^{-Sm}\Big) \cdot \big(F(x_0) - F(x^*)\big), & \text{if } \frac{m\sigma}{L} \le \frac{3}{4}; \\ O\big(1.5^{-S}\big) \cdot \big(F(x_0) - F(x^*)\big), & \text{if } \frac{m\sigma}{L} > \frac{3}{4}. \end{cases}$$

In other words, choosing $m = \Theta(n)$, `Katyusha` *achieves an ε-additive error (i.e., $\mathbb{E}\big[F(\widetilde{x}^S)\big] - F(x^*) \le \varepsilon$) using at most $O\big(\big(n+\sqrt{nL/\sigma}\big) \cdot \log\frac{F(x_0)-F(x^*)}{\varepsilon}\big)$ iterations.*[10]

The proof of Theorem 2.1 is included in Section 2.1 and 2.2. As discussed in Section 1.1, the main idea behind our theorem is the negative momentum that helps reduce the error occurred from the stochastic gradient estimator.

Remark 2.2 *Because $m = 2n$, each iteration of* `Katyusha` *computes only 1.5 stochastic gradients $\nabla f_i(\cdot)$ in the amortized sense, the same as non-accelerated methods such as SVRG (Johnson and Zhang, 2013).*[11] *Therefore, the per-iteration cost of* `Katyusha` *is dominated by the computation of $\nabla f_i(\cdot)$, the proximal update in Line 10 of Algorithm 1, plus an overhead $O(d)$. If $\nabla f_i(\cdot)$ has at most $d' \le d$ non-zero entries, this overhead $O(d)$ is improvable to $O(d')$ using a sparse implementation of* `Katyusha`.[12]

For ERM problems defined in Problem (1.3), the amortized per-iteration complexity of `Katyusha` *is $O(d')$ where d' is the sparsity of feature vectors, the same as the per-iteration complexity of SGD.*

11. The claim "SVRG or `Katyusha` computes 1.5 stochastic gradients" requires one to store $\nabla_i f(\widetilde{x})$ in the memory for each $i \in [n]$, and this costs $O(dn)$ space in the most general setting. If one does not store $\nabla_i f(\widetilde{x})$ in the memory, then each iteration of SVRG or `Katyusha` computes 2.5 stochastic gradients for the choice $m = 2n$.

2.1 One-Iteration Analysis

In this subsection, we first analyze the behavior of `Katyusha` in a single iteration (i.e., for a fixed k). We view y_k, z_k and x_{k+1} as fixed in this section so the only randomness comes from the choice of i in iteration k. We abbreviate in this subsection by $\widetilde{x} = \widetilde{x}^s$ where s is the epoch that iteration k belongs to, and denote by $\sigma_{k+1}^2 \stackrel{\text{def}}{=} \|\nabla f(x_{k+1}) - \widetilde{\nabla}_{k+1}\|^2$ so $\mathbb{E}[\sigma_{k+1}^2]$ is the variance of the gradient estimator $\widetilde{\nabla}_{k+1}$ in this iteration.

Our first lemma lower bounds the expected objective decrease $F(x_{k+1}) - \mathbb{E}[F(y_{k+1})]$. Our $\mathsf{Prog}(x_{k+1})$ defined below is a non-negative, classical quantity that would be a lower bound on the amount of objective decrease if $\widetilde{\nabla}_{k+1}$ were equal to $\nabla f(x_{k+1})$ (Allen-Zhu and Orecchia, 2017). However, since the variance σ_{k+1}^2 is non-zero, this lower bound must be compensated by a negative term that depends on $\mathbb{E}[\sigma_{k+1}^2]$.

Lemma 2.3 (proximal gradient descent) *If*

$$y_{k+1} = \arg\min_y \Big\{\frac{3L}{2}\|y - x_{k+1}\|^2 + \langle \widetilde{\nabla}_{k+1}, y - x_{k+1}\rangle + \psi(y) - \psi(x_{k+1})\Big\} \ , \qquad \text{and}$$

$$\mathsf{Prog}(x_{k+1}) \stackrel{\text{def}}{=} -\min_y \Big\{\frac{3L}{2}\|y - x_{k+1}\|^2 + \langle \widetilde{\nabla}_{k+1}, y - x_{k+1}\rangle + \psi(y) - \psi(x_{k+1})\Big\} \ge 0 \ ,$$

we have (where the expectation is only over the randomness of $\widetilde{\nabla}_{k+1}$)

$$F(x_{k+1}) - \mathbb{E}\big[F(y_{k+1})\big] \ge \mathbb{E}\big[\mathsf{Prog}(x_{k+1})\big] - \frac{1}{4L}\mathbb{E}[\sigma_{k+1}^2] \ .$$

Proof

$$\begin{aligned} \mathsf{Prog}(x_{k+1}) &= -\min_y \Big\{\frac{3L}{2}\|y - x_{k+1}\|^2 + \langle \widetilde{\nabla}_{k+1}, y - x_{k+1}\rangle + \psi(y) - \psi(x_{k+1})\Big\} \\ &\stackrel{①}{=} -\Big(\frac{3L}{2}\|y_{k+1} - x_{k+1}\|^2 + \langle \widetilde{\nabla}_{k+1}, y_{k+1} - x_{k+1}\rangle + \psi(y_{k+1}) - \psi(x_{k+1})\Big) \\ &= -\Big(\frac{L}{2}\|y_{k+1} - x_{k+1}\|^2 + \langle \nabla f(x_{k+1}), y_{k+1} - x_{k+1}\rangle + \psi(y_{k+1}) - \psi(x_{k+1})\Big) \\ &\quad + \Big(\langle \nabla f(x_{k+1}) - \widetilde{\nabla}_{k+1}, y_{k+1} - x_{k+1}\rangle - L\|y_{k+1} - x_{k+1}\|^2\Big) \\ &\stackrel{②}{\le} -\Big(f(y_{k+1}) - f(x_{k+1}) + \psi(y_{k+1}) - \psi(x_{k+1})\Big) + \frac{1}{4L}\|\nabla f(x_{k+1}) - \widetilde{\nabla}_{k+1}\|^2 \ . \end{aligned}$$

Above, ① is by the definition of y_{k+1}, and ② uses the smoothness of function $f(\cdot)$, as well as Young's inequality $\langle a, b\rangle - \frac{1}{2}\|b\|^2 \le \frac{1}{2}\|a\|^2$. Taking expectation on both sides we arrive at the desired result. ■

12. This requires to defer a coordinate update to the moment it is accessed. Update deferral is a standard technique used in sparse implementations of all stochastic gradient methods, including SVRG, SAGA, APCG (Johnson and Zhang, 2013; Defazio et al., 2014; Lin et al., 2014).

The following lemma provides a novel upper bound on the expected variance of the gradient estimator. Note that all known variance reduction analysis for convex optimization, in one way or another, upper bounds this variance essentially by $4L\cdot(f(\widetilde{x})-f(x^*))$, the objective distance to the minimizer (c.f. Johnson and Zhang (2013); Defazio et al. (2014)). The recent result of Allen-Zhu and Hazan (2016b) upper bounds it by the point distance $\|x_{k+1}-\widetilde{x}\|^2$ for non-convex objectives, which is tighter if $\widetilde{x}$ is close to x_{k+1} but unfortunately not enough for the purpose of this paper.

In this paper, we upper bound it by the tightest possible quantity which is essentially $2L\cdot\big(f(\widetilde{x})-f(x_{k+1})\big)\ll 4L\cdot\big(f(\widetilde{x})-f(x^*)\big)$. Unfortunately, this upper bound needs to be compensated by an additional term $\langle\nabla f(x_{k+1}),\widetilde{x}-x_{k+1}\rangle$, which could be positive but we shall cancel it using the introduced Katyusha momentum.

Lemma 2.4 (variance upper bound)

$$\mathbb{E}\big[\|\widetilde{\nabla}_{k+1}-\nabla f(x_{k+1})\|^2\big]\le 2L\cdot\big(f(\widetilde{x})-f(x_{k+1})-\langle\nabla f(x_{k+1}),\widetilde{x}-x_{k+1}\rangle\big)\ .$$

Proof Each $f_i(x)$, being convex and L-smooth, implies the following inequality which is classical in convex optimization and can be found for instance in Theorem 2.1.5 of the textbook of Nesterov (2004).

$$\|\nabla f_i(x_{k+1})-\nabla f_i(\widetilde{x})\|^2\le 2L\cdot\big(f_i(\widetilde{x})-f_i(x_{k+1})-\langle\nabla f_i(x_{k+1}),\widetilde{x}-x_{k+1}\rangle\big)$$

Therefore, taking expectation over the random choice of i, we have

$$\begin{aligned}\mathbb{E}\big[\|\widetilde{\nabla}_{k+1}-\nabla f(x_{k+1})\|^2\big]&=\mathbb{E}\big[\big\|\big(\nabla f_i(x_{k+1})-\nabla f_i(\widetilde{x})\big)-\big(\nabla f(x_{k+1})-\nabla f(\widetilde{x})\big)\big\|^2\big]\\&\overset{①}{\le}\mathbb{E}\big[\big\|\nabla f_i(x_{k+1})-\nabla f_i(\widetilde{x})\big\|^2\big]\\&\overset{②}{\le}2L\cdot\mathbb{E}\big[f_i(\widetilde{x})-f_i(x_{k+1})-\langle\nabla f_i(x_{k+1}),\widetilde{x}-x_{k+1}\rangle\big]\\&=2L\cdot\big(f(\widetilde{x})-f(x_{k+1})-\langle\nabla f(x_{k+1}),\widetilde{x}-x_{k+1}\rangle\big)\ .\end{aligned}$$

Above, ① is because for any random vector $\zeta\in\mathbb{R}^d$, it holds that $\mathbb{E}\|\zeta-\mathbb{E}\zeta\|^2=\mathbb{E}\|\zeta\|^2-\|\mathbb{E}\zeta\|^2$; ② follows from the first inequality in this proof. ■

The next lemma is a classical one for proximal mirror descent.

Lemma 2.5 (proximal mirror descent) *Suppose $\psi(\cdot)$ is σ-SC. Then, fixing $\widetilde{\nabla}_{k+1}$ and letting*

$$z_{k+1}=\arg\min_z\big\{\frac{1}{2}\|z-z_k\|^2+\alpha\langle\widetilde{\nabla}_{k+1},z-z_k\rangle+\alpha\psi(z)-\alpha\psi(z_k)\big\}\ ,$$

it satisfies for all $u\in\mathbb{R}^d$,

$$\alpha\langle\widetilde{\nabla}_{k+1},z_{k+1}-u\rangle+\alpha\psi(z_{k+1})-\alpha\psi(u)\le-\frac{1}{2}\|z_k-z_{k+1}\|^2+\frac{1}{2}\|z_k-u\|^2-\frac{1+\alpha\sigma}{2}\|z_{k+1}-u\|^2\ .$$

Proof By the minimality definition of z_{k+1}, we have that

$$z_{k+1}-z_k+\alpha\widetilde{\nabla}_{k+1}+\alpha g=0$$

where g is *some* subgradient of $\psi(z)$ at point $z=z_{k+1}$. This implies that for every u it satisfies

$$0=\big\langle z_{k+1}-z_k+\alpha\widetilde{\nabla}_{k+1}+\alpha g,z_{k+1}-u\big\rangle\ .$$

At this point, using the equality $\langle z_{k+1}-z_k,z_{k+1}-u\rangle=\frac{1}{2}\|z_k-z_{k+1}\|^2-\frac{1}{2}\|z_k-u\|^2+\frac{1}{2}\|z_{k+1}-u\|^2$, as well as the inequality $\langle g,z_{k+1}-u\rangle\ge\psi(z_{k+1})-\psi(u)+\frac{\sigma}{2}\|z_{k+1}-u\|^2$ which comes from the strong convexity of $\psi(\cdot)$, we can write

$$\begin{aligned}&\alpha\langle\widetilde{\nabla}_{k+1},z_{k+1}-u\rangle+\alpha\psi(z_{k+1})-\alpha\psi(u)\\&=-\langle z_{k+1}-z_k,z_{k+1}-u\rangle-\langle\alpha g,z_{k+1}-u\rangle+\alpha\psi(z_{k+1})-\alpha\psi(u)\\&\le-\frac{1}{2}\|z_k-z_{k+1}\|^2+\frac{1}{2}\|z_k-u\|^2-\frac{1+\alpha\sigma}{2}\|z_{k+1}-u\|^2\ .\end{aligned}$$

■

The following lemma combines Lemma 2.3, Lemma 2.4 and Lemma 2.5 all together, using the special choice of x_{k+1} which is a convex combination of y_k, z_k and $\widetilde{x}$:

Lemma 2.6 (coupling step 1) *If $x_{k+1}=\tau_1z_k+\tau_2\widetilde{x}+(1-\tau_1-\tau_2)y_k$, where $\tau_1\le\frac{1}{3\alpha L}$ and $\tau_2=\frac{1}{2}$,*

$$\begin{aligned}&\alpha\langle\nabla f(x_{k+1}),z_k-u\rangle-\alpha\psi(u)\\&\le\frac{\alpha}{\tau_1}\Big(F(x_{k+1})-\mathbb{E}\big[F(y_{k+1})\big]+\tau_2F(\widetilde{x})-\tau_2f(x_{k+1})-\tau_2\langle\nabla f(x_{k+1}),\widetilde{x}-x_{k+1}\rangle\Big)\\&\quad+\frac{1}{2}\|z_k-u\|^2-\frac{1+\alpha\sigma}{2}\mathbb{E}\big[\|z_{k+1}-u\|^2\big]+\frac{\alpha(1-\tau_1-\tau_2)}{\tau_1}\psi(y_k)-\frac{\alpha}{\tau_1}\psi(x_{k+1})\ .\end{aligned}$$

Proof We first apply Lemma 2.5 and get

$$\begin{aligned}&\alpha\langle\widetilde{\nabla}_{k+1},z_k-u\rangle+\alpha\psi(z_{k+1})-\alpha\psi(u)\\&=\alpha\langle\widetilde{\nabla}_{k+1},z_k-z_{k+1}\rangle+\alpha\langle\widetilde{\nabla}_{k+1},z_{k+1}-u\rangle+\alpha\psi(z_{k+1})-\alpha\psi(u)\\&\le\alpha\langle\widetilde{\nabla}_{k+1},z_k-z_{k+1}\rangle-\frac{1}{2}\|z_k-z_{k+1}\|^2+\frac{1}{2}\|z_k-u\|^2-\frac{1+\alpha\sigma}{2}\|z_{k+1}-u\|^2\ .\end{aligned}\tag{2.1}$$

By defining $v\overset{\text{def}}{=}\tau_1z_{k+1}+\tau_2\widetilde{x}+(1-\tau_1-\tau_2)y_k$, we have $x_{k+1}-v=\tau_1(z_k-z_{k+1})$ and therefore

$$\begin{aligned}&\mathbb{E}\Big[\alpha\langle\widetilde{\nabla}_{k+1},z_k-z_{k+1}\rangle-\frac{1}{2}\|z_k-z_{k+1}\|^2\Big]=\mathbb{E}\Big[\frac{\alpha}{\tau_1}\langle\widetilde{\nabla}_{k+1},x_{k+1}-v\rangle-\frac{1}{2\tau_1^2}\|x_{k+1}-v\|^2\Big]\\&=\mathbb{E}\Big[\frac{\alpha}{\tau_1}\Big(\langle\widetilde{\nabla}_{k+1},x_{k+1}-v\rangle-\frac{1}{2\alpha\tau_1}\|x_{k+1}-v\|^2-\psi(v)+\psi(x_{k+1})\Big)+\frac{\alpha}{\tau_1}\Big(\psi(v)-\psi(x_{k+1})\Big)\Big]\\&\overset{①}{\le}\mathbb{E}\Big[\frac{\alpha}{\tau_1}\Big(\langle\widetilde{\nabla}_{k+1},x_{k+1}-v\rangle-\frac{3L}{2}\|x_{k+1}-v\|^2-\psi(v)+\psi(x_{k+1})\Big)+\frac{\alpha}{\tau_1}\Big(\psi(v)-\psi(x_{k+1})\Big)\Big]\\&\overset{②}{\le}\mathbb{E}\Big[\frac{\alpha}{\tau_1}\Big(F(x_{k+1})-F(y_{k+1})+\frac{1}{4L}\sigma_{k+1}^2\Big)+\frac{\alpha}{\tau_1}\Big(\psi(v)-\psi(x_{k+1})\Big)\Big]\\&\overset{③}{\le}\mathbb{E}\Big[\frac{\alpha}{\tau_1}\Big(F(x_{k+1})-F(y_{k+1})+\frac{1}{2}\big(f(\widetilde{x})-f(x_{k+1})-\langle\nabla f(x_{k+1}),\widetilde{x}-x_{k+1}\rangle\big)\Big)\\&\quad+\frac{\alpha}{\tau_1}\Big(\tau_1\psi(z_{k+1})+\tau_2\psi(\widetilde{x})+(1-\tau_1-\tau_2)\psi(y_k)-\psi(x_{k+1})\Big)\Big]\ .\end{aligned}\tag{2.2}$$

Above, ① uses our choice $\tau_1 \le \frac{1}{3\alpha L}$, ② uses Lemma 2.3, ③ uses Lemma 2.4 together with the convexity of $\psi(\cdot)$ and the definition of v. Finally, noticing that $\mathbb{E}[\langle \widetilde{\nabla}_{k+1}, z_k - u\rangle] = \langle \nabla f(x_{k+1}), z_k - u\rangle$ and $\tau_2 = \frac{1}{2}$, we obtain the desired inequality by combining (2.1) and (2.2). ■

The next lemma simplifies the left hand side of Lemma 2.6 using the convexity of $f(\cdot)$, and gives an inequality that relates the objective-distance-to-minimizer quantities $F(y_k) - F(x^*)$, $F(y_{k+1}) - F(x^*)$, and $F(\widetilde{x}) - F(x^*)$ to the point-distance-to-minimizer quantities $\|z_k - x^*\|^2$ and $\|z_{k+1} - x^*\|^2$.

Lemma 2.7 (coupling step 2) *Under the same choices of τ_1, τ_2 as in Lemma 2.6, we have*

$$0 \le \frac{\alpha(1-\tau_1-\tau_2)}{\tau_1}\big(F(y_k) - F(x^*)\big) - \frac{\alpha}{\tau_1}\big(\mathbb{E}\big[F(y_{k+1})\big] - F(x^*)\big) + \frac{\alpha\tau_2}{\tau_1}\big(F(\widetilde{x}) - F(x^*)\big) + \frac{1}{2}\|z_k - x^*\|^2 - \frac{1+\alpha\sigma}{2}\mathbb{E}\big[\|z_{k+1} - x^*\|^2\big] \ .$$

Proof We first compute that

$$\begin{aligned}
&\alpha\big(f(x_{k+1}) - f(u)\big) \overset{①}{\le} \alpha\langle \nabla f(x_{k+1}), x_{k+1} - u\rangle \\
&= \alpha\langle \nabla f(x_{k+1}), x_{k+1} - z_k\rangle + \alpha\langle \nabla f(x_{k+1}), z_k - u\rangle \\
&\overset{②}{=} \frac{\alpha\tau_2}{\tau_1}\langle \nabla f(x_{k+1}), \widetilde{x} - x_{k+1}\rangle + \frac{\alpha(1-\tau_1-\tau_2)}{\tau_1}\langle \nabla f(x_{k+1}), y_k - x_{k+1}\rangle + \alpha\langle \nabla f(x_{k+1}), z_k - u\rangle \\
&\overset{③}{\le} \frac{\alpha\tau_2}{\tau_1}\langle \nabla f(x_{k+1}), \widetilde{x} - x_{k+1}\rangle + \frac{\alpha(1-\tau_1-\tau_2)}{\tau_1}(f(y_k) - f(x_{k+1})) + \alpha\langle \nabla f(x_{k+1}), z_k - u\rangle \ .
\end{aligned}$$

Above, ① uses the convexity of $f(\cdot)$, ② uses the choice that $x_{k+1} = \tau_1 z_k + \tau_2 \widetilde{x} + (1-\tau_1-\tau_2)y_k$, and ③ uses the convexity of $f(\cdot)$ again. By applying Lemma 2.6 to the above inequality, we have

$$\begin{aligned}
&\alpha\big(f(x_{k+1}) - F(u)\big) \le \frac{\alpha(1-\tau_1-\tau_2)}{\tau_1}(F(y_k) - f(x_{k+1})) \\
&+\frac{\alpha}{\tau_1}\Big(F(x_{k+1}) - \mathbb{E}\big[F(y_{k+1})\big] + \tau_2 F(\widetilde{x}) - \tau_2 f(x_{k+1})\Big) + \frac{1}{2}\|z_k - u\|^2 - \frac{1+\alpha\sigma}{2}\mathbb{E}\big[\|z_{k+1} - u\|^2\big] - \frac{\alpha}{\tau_1}\psi(x_{k+1})
\end{aligned}$$

which implies

$$\begin{aligned}
&\alpha\big(F(x_{k+1}) - F(u)\big) \le \frac{\alpha(1-\tau_1-\tau_2)}{\tau_1}(F(y_k) - F(x_{k+1})) \\
&+\frac{\alpha}{\tau_1}\Big(F(x_{k+1}) - \mathbb{E}\big[F(y_{k+1})\big] + \tau_2 F(\widetilde{x}) - \tau_2 F(x_{k+1})\Big) + \frac{1}{2}\|z_k - u\|^2 - \frac{1+\alpha\sigma}{2}\mathbb{E}\big[\|z_{k+1} - u\|^2\big] \ .
\end{aligned}$$

After rearranging and setting $u = x^*$, the above inequality yields

$$0 \le \frac{\alpha(1-\tau_1-\tau_2)}{\tau_1}\big(F(y_k) - F(x^*)\big) - \frac{\alpha}{\tau_1}\big(\mathbb{E}\big[F(y_{k+1}) - F(x^*)\big]\big) + \frac{\alpha\tau_2}{\tau_1}\big(F(\widetilde{x}) - F(x^*)\big) + \frac{1}{2}\|z_k - x^*\|^2 - \frac{1+\alpha\sigma}{2}\mathbb{E}\big[\|z_{k+1} - x^*\|^2\big] \ .$$

■

2.2 Proof of Theorem 2.1

We are now ready to combine the analyses across iterations, and derive our final Theorem 2.1. Our proof next requires a careful telescoping of Lemma 2.7 together with our specific parameter choices.

Proof [Proof of Theorem 2.1] Define $D_k \overset{\text{def}}{=} F(y_k) - F(x^*)$, $\widetilde{D}^s \overset{\text{def}}{=} F(\widetilde{x}^s) - F(x^*)$, and rewrite Lemma 2.7:

$$0 \le \frac{(1-\tau_1-\tau_2)}{\tau_1}D_k - \frac{1}{\tau_1}D_{k+1} + \frac{\tau_2}{\tau_1}\mathbb{E}\big[\widetilde{D}^s\big] + \frac{1}{2\alpha}\|z_k - x^*\|^2 - \frac{1+\alpha\sigma}{2\alpha}\mathbb{E}\big[\|z_{k+1} - x^*\|^2\big] \ .$$

At this point, let us define $\theta = 1 + \alpha\sigma$ and multiply the above inequality by θ^j for each $k = sm + j$. Then, we sum up the resulting m inequalities for all $j = 0, 1, \ldots, m-1$:

$$\begin{aligned}
0 \le \mathbb{E}\Big[\frac{(1-\tau_1-\tau_2)}{\tau_1}\sum_{j=0}^{m-1} D_{sm+j}\cdot\theta^j - \frac{1}{\tau_1}\sum_{j=0}^{m-1} D_{sm+j+1}\cdot\theta^j\Big] + \frac{\tau_2}{\tau_1}\widetilde{D}^s\cdot\sum_{j=0}^{m-1}\theta^j \\
+ \frac{1}{2\alpha}\|z_{sm} - x^*\|^2 - \frac{\theta^m}{2\alpha}\big[\|z_{(s+1)m} - x^*\|^2\big] \ .
\end{aligned}$$

Note that in the above inequality we have assumed all the randomness in the first $s-1$ epochs are fixed and the only source of randomness comes from epoch s. We can rearrange the terms in the above inequality and get

$$\begin{aligned}
\mathbb{E}\Big[\frac{\tau_1+\tau_2-(1-1/\theta)}{\tau_1}\sum_{j=1}^{m} D_{sm+j}\cdot\theta^j\Big] \le \frac{(1-\tau_1-\tau_2)}{\tau_1}\Big(D_{sm} - \theta^m\mathbb{E}\big[D_{(s+1)m}\big]\Big) \\
+ \frac{\tau_2}{\tau_1}\widetilde{D}^s\cdot\sum_{j=0}^{m-1}\theta^j + \frac{1}{2\alpha}\|z_{sm} - x^*\|^2 - \frac{\theta^m}{2\alpha}\mathbb{E}\big[\|z_{(s+1)m} - x^*\|^2\big] \ .
\end{aligned}$$

Using the special choice that $\widetilde{x}^{s+1} = \big(\sum_{j=0}^{m-1}\theta^j\big)^{-1}\cdot\sum_{j=0}^{m-1} y_{sm+j+1}\cdot\theta^j$ and the convexity of $F(\cdot)$, we derive that $\widetilde{D}^{s+1} \le \big(\sum_{j=0}^{m-1}\theta^j\big)^{-1}\cdot\sum_{j=0}^{m-1} D_{sm+j+1}\cdot\theta^j$. Substituting this into the above inequality, we get

$$\begin{aligned}
\frac{\tau_1+\tau_2-(1-1/\theta)}{\tau_1}\theta\mathbb{E}\big[\widetilde{D}^{s+1}\big]\cdot\sum_{j=0}^{m-1}\theta^j \le \frac{(1-\tau_1-\tau_2)}{\tau_1}\Big(D_{sm} - \theta^m\mathbb{E}\big[D_{(s+1)m}\big]\Big) \\
+ \frac{\tau_2}{\tau_1}\widetilde{D}^s\cdot\sum_{j=0}^{m-1}\theta^j + \frac{1}{2\alpha}\|z_{sm} - x^*\|^2 - \frac{\theta^m}{2\alpha}\mathbb{E}\big[\|z_{(s+1)m} - x^*\|^2\big] \ . \quad (2.3)
\end{aligned}$$

We consider two cases next.

Case 1. Suppose $\frac{m\sigma}{L} \le \frac{3}{4}$. In this case, we choose $\alpha = \frac{1}{\sqrt{3m\sigma L}}$ and $\tau_1 = \frac{1}{3\alpha L} = m\alpha\sigma = \frac{\sqrt{m\sigma}}{\sqrt{3L}} \in [0, \frac{1}{2}]$ for `Katyusha`. It implies $\alpha\sigma \le 1/2m$ and therefore the following inequality holds:

$$\tau_2(\theta^{m-1} - 1) + (1 - 1/\theta) = \frac{1}{2}((1+\alpha\sigma)^{m-1} - 1) + (1 - \frac{1}{1+\alpha\sigma}) \le (m-1)\alpha\sigma + \alpha\sigma = m\alpha\sigma = \tau_1 \ .$$

In other words, we have $\tau_1+\tau_2-(1-1/\theta)\geq\tau_2\theta^{m-1}$ and thus (2.3) implies that

$$\mathbb{E}\Big[\frac{\tau_2}{\tau_1}\widetilde{D}^{s+1}\cdot\sum_{j=0}^{m-1}\theta^j+\frac{1-\tau_1-\tau_2}{\tau_1}D_{(s+1)m}+\frac{1}{2\alpha}\|z_{(s+1)m}-x^*\|^2\Big]$$
$$\leq\theta^{-m}\cdot\Big(\frac{\tau_2}{\tau_1}\widetilde{D}^{s}\cdot\sum_{j=0}^{m-1}\theta^j+\frac{1-\tau_1-\tau_2}{\tau_1}D_{sm}+\frac{1}{2\alpha}\|z_{sm}-x^*\|^2\Big)\ .$$

If we telescope the above inequality over all epochs $s=0,1,\ldots,S-1$, we obtain

$$\mathbb{E}\big[F(\widetilde{x}^S)-F(x^*)\big]=\mathbb{E}\big[\widetilde{D}^S\big]\overset{①}{\leq}\theta^{-Sm}\cdot O\Big(\widetilde{D}^0+D_0+\frac{\tau_1}{\alpha m}\|x_0-x^*\|^2\Big)$$
$$\overset{②}{\leq}\theta^{-Sm}\cdot O\Big(1+\frac{\tau_1}{\alpha m\sigma}\Big)\cdot\big(F(x_0)-F(x^*)\big)$$
$$\overset{③}{=}O((1+\alpha\sigma)^{-Sm})\cdot\big(F(x_0)-F(x^*)\big)\ . \qquad (2.4)$$

Above, ① uses the fact that $\sum_{j=0}^{m-1}\theta^j\geq m$ and $\tau_2=\frac{1}{2}$; ② uses the strong convexity of $F(\cdot)$ which implies $F(x_0)-F(x^*)\geq\frac{\sigma}{2}\|x_0-x^*\|^2$; and ③ uses our choice of τ_1.

Case 2. Suppose $\frac{m\sigma}{L}>\frac{3}{4}$. In this case, we choose $\tau_1=\frac{1}{2}$ and $\alpha=\frac{1}{3\tau_1 L}=\frac{2}{3L}$ as in `Katyusha`. Our parameter choices help us simplify (2.3) as (noting $(\tau_1+\tau_2-(1-1/\theta))\theta=1$)

$$2\mathbb{E}\big[\widetilde{D}^{s+1}\big]\cdot\sum_{j=0}^{m-1}\theta^j\leq\widetilde{D}^s\cdot\sum_{j=0}^{m-1}\theta^j+\frac{1}{2\alpha}\|z_{sm}-x^*\|^2-\frac{\theta^m}{2\alpha}\mathbb{E}\big[\|z_{(s+1)m}-x^*\|^2\big]\ .$$

Since $\theta^m=(1+\alpha\sigma)^m\geq1+\alpha\sigma m=1+\frac{2\sigma m}{3L}\geq\frac{3}{2}$, the above inequality implies

$$\frac{3}{2}\mathbb{E}\big[\widetilde{D}^{s+1}\big]\cdot\sum_{j=0}^{m-1}\theta^j+\frac{9L}{8}\mathbb{E}\big[\|z_{(s+1)m}-x^*\|^2\big]\leq\widetilde{D}^s\cdot\sum_{j=0}^{m-1}\theta^j+\frac{3L}{4}\|z_{sm}-x^*\|^2\ .$$

If we telescope this inequality over all the epochs $s=0,1,\ldots,S-1$, we immediately have

$$\mathbb{E}\Big[\widetilde{D}^S\cdot\sum_{j=0}^{m-1}\theta^j+\frac{3L}{4}\|z_{Sm}-x^*\|^2\Big]\leq\big(\frac{2}{3}\big)^S\cdot\Big(\widetilde{D}^0\cdot\sum_{j=0}^{m-1}\theta^j+\frac{3L}{4}\|z_0-x^*\|^2\Big)\ .$$

Finally, since $\sum_{j=0}^{m-1}\theta^j\geq m$ and $\frac{\sigma}{2}\|z_0-x^*\|^2\leq F(x_0)-F(x^*)$ owing to the strong convexity of $F(\cdot)$, we conclude that

$$\mathbb{E}\big[F(\widetilde{x}^S)-F(x^*)\big]\leq O\big(1.5^{-S}\big)\cdot\big(F(x_0)-F(x^*)\big)\ . \qquad (2.5)$$

Combining (2.4) and (2.5) we finish the proof of Theorem 2.1. ■

3. Corollaries on Non-Smooth or Non-SC Problems

In this section we apply reductions to translate our Theorem 2.1 into optimal algorithms also for non-strongly convex objectives and/or non-smooth objectives.

To begin with, recall the following definition of the HOOD property:

Definition 3.1 (Allen-Zhu and Hazan (2016b)) *An algorithm solving the strongly convex case of Problem (1.1) satisfies the* homogenous objective decrease (HOOD) *property with $T(L,\sigma)$, if for every starting point x_0, it produces an output x' satisfying $\mathbb{E}\big[F(x')\big]-F(x^*)\leq\frac{F(x_0)-F(x^*)}{4}$ in at most $T(L,\sigma)$ stochastic gradient iterations.*

Theorem 2.1 shows that `Katyusha` satisfies the HOOD property:

Corollary 3.2 `Katyusha` *satisfies the HOOD property with $T(L,\sigma)=O\big(n+\frac{\sqrt{nL}}{\sqrt{\sigma}}\big)$.*

Remark 3.3 *Existing accelerated stochastic methods before this work (even for simpler Problem (1.3)) either do not satisfy HOOD or satisfy HOOD with an additional factor $\log(L/\sigma)$ in the number of iterations.*

Allen-Zhu and Hazan (2016b) designed three reductions algorithms to convert an algorithm satisfying the HOOD property to solve the following three cases:

Theorem 3.4 *Given algorithm $\mathcal{A}$ satisfying HOOD with $T(L,\sigma)$ and a starting vector x_0.*

- NonSC+Smooth. *For Problem (1.1) where $f(\cdot)$ is L-smooth,* AdaptReg($\mathcal{A}$) *outputs x satisfying $\mathbb{E}\big[F(x)\big]-F(x^*)\leq O(\varepsilon)$ in T stochastic gradient iterations where*

$$T=\sum_{s=0}^{S-1}T\Big(L,\frac{\sigma_0}{2^s}\Big)\ \text{where}\ \sigma_0=\frac{F(x_0)-F(x^*)}{\|x_0-x^*\|^2}\ \text{and}\ S=\log_2\frac{F(x_0)-F(x^*)}{\varepsilon}.$$

- SC+NonSmooth. *For Problem (1.3) where $\psi(\cdot)$ is σ-SC and each $f_i(\cdot)$ is $\sqrt{G}$-Lipschitz continuous,* AdaptSmooth($\mathcal{A}$) *outputs x satisfying $\mathbb{E}\big[F(x)\big]-F(x^*)\leq O(\varepsilon)$ in*

$$T=\sum_{s=0}^{S-1}T\Big(\frac{2^s}{\lambda_0},\sigma\Big)\ \text{where}\ \lambda_0=\frac{F(x_0)-F(x^*)}{G}\ \text{and}\ S=\log_2\frac{F(x_0)-F(x^*)}{\varepsilon}.$$

- NonSC+NonSmooth. *For Problem (1.3) where each $f_i(\cdot)$ is $\sqrt{G}$-Lipschitz continuous, then* JointAdaptRegSmooth($\mathcal{A}$) *outputs x satisfying $\mathbb{E}\big[F(x)\big]-F(x^*)\leq O(\varepsilon)$ in*

$$T=\sum_{s=0}^{S-1}T\Big(\frac{2^s}{\lambda_0},\frac{\sigma_0}{2^s}\Big)$$
$$\text{where}\ \lambda_0=\frac{F(x_0)-F(x^*)}{G},\ \sigma_0=\frac{F(x_0)-F(x^*)}{\|x_0-x^*\|^2}\ \text{and}\ S=\log_2\frac{F(x_0)-F(x^*)}{\|x_0-x^*\|^2}.$$

Combining Corollary 3.2 with Theorem 3.4, we have the following corollaries:

Corollary 3.5 *If each $f_i(x)$ is convex, L-smooth and $\psi(\cdot)$ is not necessarily strongly convex in Problem (1.1), then by applying* AdaptReg *on* `Katyusha` *with a starting vector x_0, we obtain an output x satisfying $\mathbb{E}[F(x)]-F(x^*)\leq\varepsilon$ in*

$$T=O\Big(n\log\tfrac{F(x_0)-F(x^*)}{\varepsilon}+\tfrac{\sqrt{nL}\cdot\|x_0-x^*\|}{\sqrt{\varepsilon}}\Big)\propto\tfrac{1}{\sqrt{\varepsilon}}\ \textit{iterations. (Or equivalently}\ \varepsilon\propto\tfrac{1}{T^2}.\textit{)}$$

In contrast, the best known convergence rate was $\varepsilon\propto\frac{\log^4 T}{T^2}$ or more precisely

Catalyst: $T=O\Big(\Big(n+\frac{\sqrt{nL}\cdot\|x_0-x^*\|}{\sqrt{\varepsilon}}\Big)\log\frac{F(x_0)-F(x^*)}{\varepsilon}\log\frac{L\|x_0-x^*\|^2}{\varepsilon}\Big)\propto\frac{\log^2(1/\varepsilon)}{\sqrt{\varepsilon}}$ iterations.

Algorithm 2 $\texttt{Katyusha}^{\texttt{ns}}(x_0, S, \sigma, L)$

$m \leftarrow 2n$; ◇ *epoch length*
$\tau_2 \leftarrow \frac{1}{2}$;
$y_0 = z_0 = \widetilde{x}^0 \leftarrow x_0$; ◇ *initial vectors*
for $s \leftarrow 0$ **to** $S-1$ **do**
  $\tau_{1,s} \leftarrow \frac{2}{s+4}$, $\alpha_s \leftarrow \frac{1}{3\tau_{1,s}L}$ ◇ *different parameter choices comparing to* Katyusha
  $\mu^s \leftarrow \nabla f(\widetilde{x}^s)$; ◇ *compute the full gradient only once every m iterations*
  for $j \leftarrow 0$ **to** $m-1$ **do**
    $k \leftarrow (sm)+j$;
    $x_{k+1} \leftarrow \tau_{1,s} z_k + \tau_2 \widetilde{x}^s + (1-\tau_{1,s}-\tau_2) y_k$;
    $\widetilde{\nabla}_{k+1} \leftarrow \mu^s + \nabla f_i(x_{k+1}) - \nabla f_i(\widetilde{x}^s)$ where i is randomly chosen from $\{1,2,\dots,n\}$;
    $z_{k+1} = \arg\min_z \left\{ \frac{1}{2\alpha_s}\|z - z_k\|^2 + \langle \widetilde{\nabla}_{k+1}, z\rangle + \psi(z) \right\}$;
    Option I: $y_{k+1} \leftarrow \arg\min_y \left\{ \frac{3L}{2}\|y - x_{k+1}\|^2 + \langle \widetilde{\nabla}_{k+1}, y\rangle + \psi(y) \right\}$;
    Option II: $y_{k+1} \leftarrow x_{k+1} + \tau_{1,s}(z_{k+1} - z_k)$ ◇ *we analyze only I but II also works*
  end for
  $\widetilde{x}^{s+1} \leftarrow \frac{1}{m}\sum_{j=1}^m y_{sm+j}$; ◇ *compute snapshot $\widetilde{x}$*
end for
return $\widetilde{x}^S$.

Corollary 3.6 *If each $f_i(x)$ is $\sqrt{G}$-Lipschitz continuous and $\psi(x)$ is σ-SC in Problem (1.3), then by applying* AdaptSmooth *on* Katyusha *with a starting vector x_0, we obtain an output x satisfying $\mathbb{E}[F(x)] - F(x^*) \le \varepsilon$ in*

$$T = O\Big(n \log \tfrac{F(x_0)-F(x^*)}{\varepsilon} + \tfrac{\sqrt{nG}}{\sqrt{\sigma\varepsilon}}\Big) \propto \tfrac{1}{\sqrt{\varepsilon}} \text{ iterations. } (\text{Or equivalently } \varepsilon \propto \tfrac{1}{T^2}.)$$

In contrast, the best known convergence rate was $\varepsilon \propto \frac{\log^2 T}{T^2}$, or more precisely

$$\text{APCG/SPDC:} \quad T = O\Big(\Big(n + \tfrac{\sqrt{nG}}{\sqrt{\sigma\varepsilon}}\Big) \log \tfrac{nG(F(x_0)-F(x^*))}{\sigma\varepsilon}\Big) \propto \tfrac{\log(1/\varepsilon)}{\sqrt{\varepsilon}} \text{ iterations.}$$

Corollary 3.7 *If each $f_i(x)$ is $\sqrt{G}$-Lipschitz continuous and $\psi(x)$ is not necessarily strongly convex in Problem (1.3), then by applying* JointAdaptRegSmooth *on* Katyusha *with a starting vector x_0, we obtain an output x satisfying $\mathbb{E}[F(x)] - F(x^*) \le \varepsilon$ in*

$$T = O\Big(n \log \tfrac{F(x_0)-F(x^*)}{\varepsilon} + \tfrac{\sqrt{nG}\|x_0 - x^*\|}{\varepsilon}\Big) \propto \tfrac{1}{\varepsilon} \text{ iterations. } (\text{Or equivalently } \varepsilon \propto \tfrac{1}{T}.)$$

In contrast, the best known convergence rate was $\varepsilon \propto \frac{\log T}{T}$, or more precisely

$$\text{APCG/SPDC:} \quad T = O\Big(\Big(n + \tfrac{\sqrt{nG}\|x_0 - x^*\|}{\varepsilon}\Big) \log \tfrac{nG\|x_0 - x^*\|^2 (F(x_0)-F(x^*))}{\varepsilon^2}\Big) \propto \tfrac{\log(1/\varepsilon)}{\varepsilon} \text{ iterations.}$$

4. Katyusha in the Non-Strongly Convex Setting

Due to the increasing popularity of *non-strongly convex* minimization tasks (most notably ℓ_1-regularized problems), researchers often make additional efforts to design separate methods for minimizing the non-strongly convex variant of Problem (1.1) that are *direct*, meaning without restarting and in particular without using any reductions such as Theorem 3.4 (Defazio et al., 2014; Allen-Zhu and Yuan, 2016).

In this section, we also develop our *direct and accelerated* method for the non-strongly convex variant of Problem (1.1). We call it $\texttt{Katyusha}^{\texttt{ns}}$ and state it in Algorithm 2.

The only difference between $\texttt{Katyusha}^{\texttt{ns}}$ and Katyusha is that we choose $\tau_1 = \tau_{1,s} = \frac{2}{s+4}$ to be a parameter that depends on the epoch index s, and accordingly $\alpha = \alpha_s = \frac{1}{3L\tau_{1,s}}$. This should not be a big surprise because in accelerated full-gradient methods, the values τ_1 and α also decrease (although with respect to k rather than s) when there is no strong convexity (Allen-Zhu and Orecchia, 2017). We note that τ_1 and τ_2 remain constant throughout an epoch, and this could simplify the implementations.

We state the following convergence theorem for $\texttt{Katyusha}^{\texttt{ns}}$ and defer its proof to Appendix C.1. The proof also relies on the one-iteration inequality in Lemma 2.7, but requires telescoping such inequalities in a different manner as compared with Theorem 2.1.

Theorem 4.1 *If each $f_i(x)$ is convex, L-smooth in Problem (1.1) and $\psi(\cdot)$ is not necessarily strongly convex, then $\texttt{Katyusha}^{\texttt{ns}}(x_0, S, L)$ satisfies*

$$\mathbb{E}\big[F(\widetilde{x}^S)\big] - F(x^*) \le O\Big(\frac{F(x_0)-F(x^*)}{S^2} + \frac{L\|x_0 - x^*\|^2}{mS^2}\Big)$$

In other words, choosing $m = \Theta(n)$, $\texttt{Katyusha}^{\texttt{ns}}$ achieves an ε-additive error (i.e., $\mathbb{E}\big[F(\widetilde{x}^S)\big] - F(x^*) \le \varepsilon$) using at most $O\Big(\frac{n\sqrt{F(x_0)-F(x^*)}}{\sqrt{\varepsilon}} + \frac{\sqrt{nL}\|x_0-x^*\|}{\sqrt{\varepsilon}}\Big)$ iterations.

Remark 4.2 $\texttt{Katyusha}^{\texttt{ns}}$ *is a* direct, accelerated *solver for the non-SC case of Problem (1.1). It is illustrative to compare it with the convergence theorem of a* direct, non-accelerated *solver of the same setting. Below is the convergence theorem of SAGA after translating to our notations:*

$$\textit{SAGA:} \quad \mathbb{E}\big[F(x)\big] - F(x^*) \le O\Big(\frac{F(x_0)-F(x^*)}{S} + \frac{L\|x_0 - x^*\|^2}{nS}\Big) \ .$$

It is clear from this comparison that $\texttt{Katyusha}^{\texttt{ns}}$ is a factor S faster than non-accelerated methods such as SAGA, where $S = T/n$ if T is the total number of stochastic iterations. This convergence can also be written in terms of the number of iterations which is $O\big(\frac{n(F(x_0)-F(x^*))}{\varepsilon} + \frac{L\|x_0-x^*\|^2}{\varepsilon}\big)$.

Remark 4.3 *Theorem 4.1 appears worse than the reduction-based complexity in Corollary 3.7. This can be fixed by setting either the parameters τ_1 or the epoch length m in a more sophisticated way. Since it complicates the proofs and the notations we refrain from doing so in this version of the paper.*[13] *In practice, being a direct method, $\texttt{Katyusha}^{\texttt{ns}}$ enjoys satisfactory performance.*

5. Katyusha in the Mini-Batch Setting

We mentioned in earlier versions of this paper that our Katyusha method naturally generalizes to mini-batch (parallel) settings and non-uniform smoothness settings, but did not include a full proof. In this section, we carefully deal with both generalizations together.

13. Recall that a similar issue has also happened in the non-accelerated world: the iteration complexity $O(\frac{n+L}{\varepsilon})$ in SAGA can be improved to $O(n \log \frac{1}{\varepsilon} + \frac{L}{\varepsilon})$ by doubling the epoch length across epochs (Allen-Zhu and Yuan, 2016). Similar techniques can also be used to improve our result above.

Mini-batch. In each iteration k, instead of using a single $\nabla f_i(x_{k+1})$, one can

$$\text{use the average of } b \text{ stochastic gradients } \frac{1}{b}\sum_{i\in S_k}\nabla f_i(x_{k+1})$$

where S_k is a random subset of $[n]$ with cardinality b. This average can be computed in a distributed manner using up to b processors. This idea is known as *mini-batch* for stochastic gradient methods.

Non-Uniform Smoothness. Suppose in Problem (1.1),

$$\text{each } f_i(x) \text{ is } L_i\text{-smooth and } f(x)=\frac{1}{n}\sum_{i=1}^n f_i(x) \text{ is } L\text{-smooth.}$$

We denote by $\overline{L}=\frac{1}{n}\sum_{i=1}^n L_i$, and assume without loss of generality $L\le\overline{L}\le nL$. [14] We note that $\overline{L}$ can sometimes be indeed much greater than L, see Remark 5.3.

Remark 5.1 *L_i and L only need to be upper bounds to the minimum smoothness parameters of $f_i(\cdot)$ and $f(\cdot)$ respectively. In practice, sometimes the minimum smoothness parameters for $f_i(x)$ is efficiently computable (such as for ERM problems).*

5.1 Algorithmic Changes and Theorem Restatement

To simultaneously deal with mini-batch and non-uniform smoothness, we propose the following changes to `Katyusha`:

- *Change the epoch length from $m=\Theta(n)$ to $m=\lceil\frac{n}{b}\rceil$.*

  This is standard. In each iteration we need to compute $O(b)$ stochastic gradients; therefore every $\lceil\frac{n}{b}\rceil$ iterations, we can compute the full gradient once without hurting the total performance.

- *Define distribution $\mathcal{D}$ over $[n]$ to be choosing $i\in[n]$ with probability $p_i\stackrel{\text{def}}{=}L_i/n\overline{L}$, and define gradient estimator $\widetilde{\nabla}_{k+1}\stackrel{\text{def}}{=}\nabla f(\widetilde{x})+\frac{1}{b}\sum_{i\in S_k}\frac{1}{np_i}\big(\nabla f_i(x_{k+1})-\nabla f_i(\widetilde{x})\big)$, where $S_k\subseteq[n]$ is a multiset with b elements each i.i.d. generated from $\mathcal{D}$.*

  This is standard, see for instance Prox-SVRG (Xiao and Zhang, 2014), and it is easy to verify $\mathbb{E}[\widetilde{\nabla}_{k+1}]=\nabla f(x_{k+1})$.

- *Change τ_2 from $\frac{1}{2}$ to $\min\{\frac{\overline{L}}{2Lb},\frac{1}{2}\}$.*

  Note that if $\overline{L}=L$ then we have $\tau_2=\frac{1}{2b}$. In other words, the larger the mini-batch size, the smaller weight we want to give to Katyusha momentum. This should be intuitive. The reason τ_2 has a more involved form when $L\neq\overline{L}$ is explained in Remark 5.4 later.

- *Change L in gradient descent step (Line 19) to some other $L_\diamond\ge L$, and define $\alpha=\frac{1}{3\tau_1 L_\diamond}$ instead.*

  In most cases (e.g., when $\overline{L}=L$ or $L\ge\overline{L}m/b$) we choose $L_\diamond=L$. Otherwise, we let $L_\diamond=\frac{\overline{L}}{2b\tau_2}\ge L$. The reason $L_\diamond$ has a more involved form is explained in Remark 5.4 later.

14. It is easy to verify (using triangle inequality) that $f(x)=\frac{1}{n}\sum_{i\in[n]}f_i(x)$ must be $\overline{L}$ smooth. Also, if $f(x)$ is L-smooth then each $f_i(x)$ must be nL smooth (this can be checked via Hessian $\nabla^2 f_i(x)\preceq n\nabla^2 f(x)$ or similarly if f is not twice-differentiable).

Algorithm 3 `Katyusha1`$(x_0,S,\sigma,L,(L_1,\dots,L_n),b)$

1: $m\leftarrow\lceil n/b\rceil$ and $\overline{L}\leftarrow\frac{1}{n}(L_1+\cdots+L_n)$; ◇ *$m$ is epoch length*
2: $\tau_2\leftarrow\min\{\frac{\overline{L}}{2Lb},\frac{1}{2}\}$; ◇ *if $\overline{L}=L$ then $\tau_2=\frac{1}{2b}$ and $L_\diamond=L$*
3: **if** $L\le\frac{\overline{L}m}{b}$ **then**
4: $\quad\tau_1\leftarrow\min\{\frac{\sqrt{8bm\sigma}}{\sqrt{3\overline{L}}}\tau_2,\tau_2\}$ and $L_\diamond\leftarrow\frac{\overline{L}}{2b\tau_2}$;
5: **else**
6: $\quad\tau_1\leftarrow\min\{\frac{\sqrt{2\sigma}}{\sqrt{3L}},\frac{1}{2m}\}$ and $L_\diamond\leftarrow L$;
7: **end if**
8: $\alpha\leftarrow\frac{1}{3\tau_1 L_\diamond}$; ◇ *parameters*
9: Let distribution $\mathcal{D}$ be to output $i\in[n]$ with probability $p_i\stackrel{\text{def}}{=}L_i/(n\overline{L})$.
10: $y_0=z_0=\widetilde{x}^0\leftarrow x_0$; ◇ *initial vectors*
11: **for** $s\leftarrow0$ **to** $S-1$ **do**
12: $\quad\mu^s\leftarrow\nabla f(\widetilde{x}^s)$; ◇ *compute the full gradient once every m iterations*
13: $\quad$**for** $j\leftarrow0$ **to** $m-1$ **do**
14: $\quad\quad k\leftarrow(sm)+j$;
15: $\quad\quad x_{k+1}\leftarrow\tau_1 z_k+\tau_2\widetilde{x}^s+(1-\tau_1-\tau_2)y_k$;
16: $\quad\quad S_k\leftarrow b$ independent copies of i from $\mathcal{D}$ with replacement.
17: $\quad\quad\widetilde{\nabla}_{k+1}\leftarrow\mu^s+\frac{1}{b}\sum_{i\in S_k}\frac{1}{np_i}\big(\nabla f_i(x_{k+1})-\nabla f_i(\widetilde{x}^s)\big)$;
18: $\quad\quad z_{k+1}=\arg\min_z\{\frac{1}{2\alpha}\|z-z_k\|^2+\langle\widetilde{\nabla}_{k+1},z\rangle+\psi(z)\}$;
19: $\quad\quad$Option I: $y_{k+1}\leftarrow\arg\min_y\{\frac{3L_\diamond}{2}\|y-x_{k+1}\|^2+\langle\widetilde{\nabla}_{k+1},y\rangle+\psi(y)\}$;
20: $\quad\quad$Option II: $y_{k+1}\leftarrow x_{k+1}+\tau_1(z_{k+1}-z_k)$ ◇ *we analyze only I but II also works*
21: $\quad$**end for**
22: $\quad\widetilde{x}^{s+1}\leftarrow\big(\sum_{j=0}^{m-1}\theta^j\big)^{-1}\cdot\big(\sum_{j=0}^{m-1}\theta^j\cdot y_{sm+j+1}\big)$; ◇ *where $\theta=1+\min\{\alpha\sigma,\frac{1}{4m}\}$*
23: **end for**
24: **return** $x^{\mathsf{out}}\leftarrow\frac{\tau_2 m\widetilde{x}^S+(1-\tau_1-\tau_2)y_{Sm}}{\tau_2 m+(1-\tau_1-\tau_2)}$.

- *Change τ_1 to be $\tau_1=\min\{\frac{\sqrt{8bm\sigma}}{\sqrt{3\overline{L}}}\tau_2,\tau_2\}$ if $L\le\overline{L}m/b$ or $\tau_1=\min\{\frac{\sqrt{2\sigma}}{\sqrt{3L}},\frac{1}{2m}\}$ if $L>\overline{L}m/b$.*

  This corresponds to a phase-transition behavior of `Katyusha1` (see Remark 5.5 later). Intuitively, when $L\le\overline{L}m/b$ then we are in a mini-batch phase; when $L>\overline{L}m/b$ we are in a full-batch phase.

- *Due to technical reasons, we define $\widetilde{x}^s$ as a slightly different weighted average (Line 22) and output x^{out} which is a weighted combination of $\widetilde{x}^S$ and y_{Sm} as opposed to simply $\widetilde{x}^S$ (Line 24).*

We emphasize here that some of these changes are not necessary for instance in the special case of $\overline{L}=L$, but to state the strongest theorem, we have to include all such changes. It is a simple exercise to verify that, if $\overline{L}=L$ and $b=1$, then up to only constant factors in the parameters, `Katyusha1` is exactly identical to `Katyusha`. We have the following main theorem for `Katyusha1`:

Theorem 5.2 *If each $f_i(x)$ is convex and L_i-smooth, $f(x)$ is L-smooth, $\psi(x)$ is σ-strongly convex in Problem (1.1), then for any $b \in [n]$,*

$$x^{\mathsf{out}} = \texttt{Katyusha1}(x_0, S, \sigma, L, (L_1, \dots, L_n), b)$$

satisfies $\mathbb{E}\big[F(x^{\mathsf{out}})\big] - F(x^*)$

$$\leq \begin{cases} O\Big(\Big(1+\sqrt{b\sigma/(6\overline{L}m)}\Big)^{-Sm}\Big) \cdot \big(F(x_0)-F(x^*)\big), & \text{if } \frac{m\alpha b}{\overline{L}} \leq \frac{3}{8} \text{ and } L \leq \frac{\overline{L}m}{b}; \\ O\Big(\big(1+\sqrt{\sigma/(6L)}\big)^{-Sm}\Big) \cdot \big(F(x_0)-F(x^*)\big), & \text{if } \frac{m^2\sigma}{L} \leq \frac{3}{8} \text{ and } L > \frac{\overline{L}m}{b}; \\ O\big(1.25^{-S}\big) \cdot \big(F(x_0)-F(x^*)\big), & \text{otherwise.} \end{cases}$$

In other words, choosing $m = \lceil n/b \rceil$, Katyusha achieves an ε-additive error (that is, $\mathbb{E}\big[F(x^{\mathsf{out}})\big] - F(x^*) \leq \varepsilon$) using at most

$$S \cdot n = O\Big(\big(n + b\sqrt{L/\sigma} + \sqrt{n\overline{L}/\sigma}\big) \cdot \log \frac{F(x_0)-F(x^*)}{\varepsilon}\Big)$$

stochastic gradient computations.

5.2 Observations and Remarks

We explain the significance of Theorem 5.2 below. We use *total work* to refer to the total number of stochastic gradient computations, and *iteration complexity* (also known as parallel depth) to refer to the total number of iterations.

Parallel Performance. The total work of Katyusha1 stays the same when $b \leq (n\overline{L}/L)^{1/2} \in \big[\sqrt{n}, n\big]$. This means, at least for all values $b \in \{1, 2, \dots, \lceil\sqrt{n}\rceil\}$, our Katyusha1 achieves the same total work and thus

Katyusha1 can be distributed to $b \leq \sqrt{n}$ machines with a parallel speed-up factor b
(known as linear speed-up if ignoring communication overhead.)

In contrast, even in the special case of $\overline{L} = L$ and if no additional assumption is made, to the best of our knowledge:

- Mini-batch SVRG requires $\widetilde{O}\big(n + \frac{bL}{\sigma}\big)$ total work.

  Therefore, if SVRG is distributed to b machines, the total work is increased by a factor of b, and the parallel speed-up factor is 1 (i.e., no speed up).

- Catalyst on top of mini-batch SVRG requires $\widetilde{O}\big(n + \frac{\sqrt{bLn}}{\sqrt{\sigma}}\big)$ total work.

  Therefore, if Catalyst is distributed to b machines, the total work is increased by a factor $\sqrt{b}$, and the parallel speed-up factor is $\sqrt{b}$ only.

When preparing the journal revision (i.e., version 5), we found out at least in the case $\overline{L} = L$, some other groups of researchers very recently obtained similar results for the ERM Problem (1.3) using SPDC (Shibagaki and Takeuchi, 2017), and for the general Problem (1.1) (Murata and Suzuki, 2017).[15] These results together with Theorem 5.2 confirm the power of acceleration in the parallel regime for stochastic gradient methods.

15. These two papers claimed that Katyusha does not enjoy linear speed-up for $b \leq \sqrt{n}$, based on an earlier version of the paper (where we did not include the mini-batch theorem). As evidenced by Theorem 5.2, such claims are false.

Outperforming Full-Gradient Method. If $b = n$, the total work of Katyusha1 becomes $\widetilde{O}\big((L/\sigma)^{1/2}n\big)$. This matches the total work of Nesterov's accelerated gradient method (Nesterov, 1983, 2004; Allen-Zhu and Orecchia, 2017), and does not depend on the possibly larger parameter $\overline{L}$.

More interestingly, to achieve the *same* iteration complexity $\widetilde{O}\big((L/\sigma)^{1/2}\big)$ as Nesterov's method, our Katyusha1 only needs to compute $b = (n\overline{L}/L)^{1/2}$ stochastic gradients $\nabla f_i(\cdot)$ per iteration (in the amortized sense). This can be much faster than computing $\nabla f(\cdot)$.

Remark 5.3 *Recall $\overline{L}$ is in the range $[L, nL]$ so indeed $\overline{L}$ can be much larger than L. For instance in linear regression we have $f_i(x) = \frac{1}{2}(\langle a_i, x\rangle - b_i)^2$. Denoting by $A = [a_1, \dots, a_n] \in \mathbb{R}^{d\times n}$, we have $L = \frac{1}{n}\lambda_{\max}(A^\top A)$ and $\overline{L} = \frac{1}{n}\|A\|_F^2$. If each entry of each a_i is a random Gaussian $N(0,1)$, then $\overline{L}$ is around d and $\overline{L}$ is around only $\Theta(1+\frac{d}{n})$ (using the Wishart random matrix theory).*

Remark 5.4 *The parameter specifications in* Katyusha1 *look intimidating partially because we have tried to obtain the strongest statement and match the full-gradient descent performance when $b = n$. If $\overline{L}$ is equal to L, then one can simply set $\tau_2 = \frac{1}{2b}$ and $L_\diamond = L$ in* Katyusha1.

Phase Transition between Mini-Batch and Full-Batch. Theorem 5.2 indicates a phase transition of Katyusha1 at the point $b_0 = (n\overline{L}/L)^{1/2}$.

- If $b \leq b_0$, we say Katyusha1 is in the *mini-batch phase* and the total work is $\widetilde{O}\big(n + \sqrt{n\overline{L}/\sigma}\big)$, independent of b.
- If $b > b_0$, we say Katyusha1 is in the *full-batch phase*, and the total work is $\widetilde{O}\big(n+b\sqrt{L/\sigma}\big)$, so essentially linearly-scales with b and matches that of Nesterov's method when $b = n$.

Remark 5.5 *We set different values for τ_1 and $L_\diamond$ in the mini-batch phase and full-batch phase respectively (see Line 3). From the final complexities above, it should not be surprising that τ_1 depends on $\overline{L}$ but not L in the mini-batch phase, and depends on L but not $\overline{L}$ in the full-batch phase. In addition, one can even tune the parameters so that it suffices for* Katyusha *to output $\widetilde{x}^S$ in the mini-batch phase and y_{Sm} in the full-batch phase; we did not do so and simply choose to output x^{out} which is a convex combination of $\widetilde{x}^S$ and y_{Sm}.*

Remark 5.6 *In the simple case $\overline{L} = L$, Nitanda (2014) obtained a total work $\widetilde{O}\big(n+\frac{n-b}{n-1}\frac{L}{\sigma} + b\sqrt{L/\sigma}\big)$, which also implies a phase transition for b. However, this result is no better than ours for all b, and in addition, in terms of total work, it is no faster than SVRG when $b \leq n/2$, and no faster than accelerated full-gradient descent when $b > n/2$.*

5.3 Corollaries on Non-Smooth or Non-SC Problems

In the same way as Section 3, we can apply the reductions from Allen-Zhu and Hazan (2016b) to convert the performance of Theorem 5.2 to non-smooth or non-strongly convex settings. We state the corollaries below:

Corollary 5.7 *If each $f_i(x)$ is convex and L_i-smooth, $f(x)$ is L-smooth, $\psi(\cdot)$ is not necessarily strongly convex in Problem (1.1), then for any $b \in [n]$, by applying* AdaptReg *on* `Katyusha1` *with a starting vector x_0, we obtain an output x satisfying $\mathbb{E}[F(x)] - F(x^*) \le \varepsilon$ in at most*

$$O\Big(n \log \tfrac{F(x_0)-F(x^*)}{\varepsilon} + \tfrac{b\sqrt{L}\cdot\|x_0-x^*\|}{\sqrt{\varepsilon}} + \tfrac{\sqrt{nL}\cdot\|x_0-x^*\|}{\sqrt{\varepsilon}}\Big) \text{ stochastic gradient computations.}$$

Corollary 5.8 *If each $f_i(x)$ is $\sqrt{G_i}$-Lipschitz continuous and $\psi(x)$ is σ-SC in Problem (1.3), then for any $b \in [n]$, by applying* AdaptSmooth *on* `Katyusha1` *with a starting vector x_0, we obtain an output x satisfying $\mathbb{E}[F(x)] - F(x^*) \le \varepsilon$ in at most*

$$O\Big(n \log \tfrac{F(x_0)-F(x^*)}{\varepsilon} + \tfrac{b\sqrt{\overline{G}}}{\sqrt{\sigma\varepsilon}} + \tfrac{\sqrt{n\overline{G}}}{\sqrt{\sigma\varepsilon}}\Big) \text{ stochastic gradient computations.}$$

Corollary 5.9 *If each $f_i(x)$ is $\sqrt{G_i}$-Lipschitz continuous and $\psi(x)$ is not necessarily strongly convex in Problem (1.3), then for any $b \in [n]$, by applying* JointAdaptRegSmooth *on* `Katyusha1` *with a starting vector x_0, we obtain an output x satisfying $\mathbb{E}[F(x)] - F(x^*) \le \varepsilon$ in at most*

$$O\Big(n \log \tfrac{F(x_0)-F(x^*)}{\varepsilon} + \tfrac{b\overline{G}\|x_0-x^*\|}{\varepsilon} + \tfrac{\sqrt{n\,\overline{G}}\|x_0-x^*\|}{\varepsilon}\Big) \text{ stochastic gradient computations.}$$

6. Katyusha in the Non-Euclidean Norm Setting

In this section, we show that `Katyusha` and `Katyusha`$^{\mathsf{ns}}$ naturally extend to settings where the smoothness definition is with respect to a non-Euclidean norm.

Non-Euclidean Norm Smoothness. We consider smoothness (and strongly convexity) with respect to an arbitrary norm $\|\cdot\|$ in domain $Q \stackrel{\text{def}}{=} \{x \in \mathbb{R}^d : \psi(x) < +\infty\}$. Symbolically, we say

- f is σ-strongly convex w.r.t. $\|\cdot\|$ if $\forall x, y \in Q$, it satisfies $f(y) \ge f(x) + \langle \nabla f(x), y - x\rangle + \frac{\sigma}{2}\|x-y\|^2$;
- f is L-smooth w.r.t. $\|\cdot\|$ if $\forall x, y \in Q$, it satisfies $\|\nabla f(x) - \nabla f(y)\|_* \le L\|x-y\|$.[16]

Above, $\|\cdot\|_* \stackrel{\text{def}}{=} \max\{\langle \xi, x\rangle : \|x\| \le 1\}$ is the dual norm of $\|\cdot\|$. For instance, ℓ_p norm is dual to ℓ_q norm if $\frac{1}{p} + \frac{1}{q} = 1$. Some famous problems have better smoothness parameters when non-Euclidean norms are adopted, see the discussions in Allen-Zhu and Orecchia (2017).

Bregman Divergence. Following the traditions in the non-Euclidean norm setting (Allen-Zhu and Orecchia, 2017), we

- select a *distance generating function* $w(\cdot)$ that is 1-strongly convex w.r.t. $\|\cdot\|$, and[17]
- define the *Bregman divergence function* $V_x(y) \stackrel{\text{def}}{=} w(y) - w(x) - \langle \nabla w(x), y - x\rangle$.

The final algorithms and proofs will be described using $V_x(y)$ and $w(x)$.

16. This definition has another equivalent form: $\forall x, y \in Q$, it satisfies $f(y) \le f(x) + \langle \nabla f(x), y-x\rangle + \frac{L}{2}\|y-x\|^2$.
17. For instance, if $Q = \mathbb{R}^d$ and $\|\cdot\|_p$ is the ℓ_p norm for some $p \in (1, 2]$, one can choose $w(x) = \frac{1}{2(p-1)}\|x\|_p^2$; if $Q = \{x \in \mathbb{R}^d : \sum_i x_i = 1\}$ is the probability space and $\|\cdot\|_1$ is the ℓ_1 norm, one can choose $w(x) = \sum_i x_i \log x_i$.

Generalized Strong Convexity of $\psi(\cdot)$. We require $\psi(\cdot)$ to be σ-strongly convexity with respect to function $V_x(y)$ rather than the $\|\cdot\|$ norm; or symbolically,

$$\psi(y) \ge \psi(x) + \langle \nabla\psi(x), y - x\rangle + \sigma V_x(y) \ .$$

(For instance, this is satisfied if $\omega(y) \stackrel{\text{def}}{=} \frac{1}{\sigma}\psi(y)$.) This is known as the "generalized strong convexity" (Shalev-Shwartz, 2007) and is necessary for any linear-convergence result in the SC setting. Of course, in the non-SC setting, we do not require any (general or not) strong convexity for $\psi(\cdot)$.

6.1 Algorithm Changes and Theorem Restatements

Suppose each $f_i(x)$ is L_i-smooth with respect to norm $\|\cdot\|$, and a Bregman divergence function $V_x(y)$ is given. We perform the following changes to the algorithms:

- In Line 9 of `Katyusha` (resp. Line 10 of `Katyusha`$^{\mathsf{ns}}$), we choose i with probability proportional to L_i instead of uniformly at random.
- In Line 10 of `Katyusha` (resp. Line 11 of `Katyusha`$^{\mathsf{ns}}$), we change the arg min to be its non-Euclidean norm variant (Allen-Zhu and Orecchia, 2017): $z_{k+1} = \arg\min_z \big\{\frac{1}{\alpha}V_{z_k}(z) + \langle \widetilde{\nabla}_{k+1}, z\rangle + \psi(z)\big\}$
- We forbidden Option II and use Option I only (but *without* replacing $\|y - x_{k+1}\|^2$ with $V_{x_{k+1}}(y)$).

Interested readers can find discussions regarding why such changes are natural in Allen-Zhu and Orecchia (2017). We call the resulting algorithms `Katyusha2` and `Katyusha2`$^{\mathsf{ns}}$, and include them in Appendix E for completeness' sake. We state our final theorems below (recall $\overline{L} = \frac{1}{n}\sum_{i=1}^n L_i$).

Theorem 6.1 (ext. of Theorem 2.1) *If each $f_i(x)$ is convex and L_i-smooth with respect to some norm $\|\cdot\|$, $V_x(y)$ is a Bregman divergence function for $\|\cdot\|$, and $\psi(x)$ is σ-strongly convex with respect to $V_x(y)$, then* `Katyusha2`$(x_0, S, \sigma, (L_1, \dots, L_n))$ *satisfies*

$$\mathbb{E}\big[F(\widetilde{x}^S)\big] - F(x^*) \le \begin{cases} O\Big(\Big(1 + \sqrt{\sigma/(9\overline{L}m)}\Big)^{-Sm}\Big) \cdot \big(F(x_0) - F(x^*)\big), & \text{if } m\sigma/\overline{L} \le \frac{9}{4}; \\ O\big(1.5^{-S}\big) \cdot \big(F(x_0) - F(x^*)\big), & \text{if } m\sigma/\overline{L} > \frac{9}{4}. \end{cases}$$

In other words, choosing $m = \Theta(n)$, `Katyusha2` *achieves an ε-additive error (i.e., $\mathbb{E}\big[F(\widetilde{x}^S)\big] - F(x^*) \le \varepsilon$) using at most $O\big(\big(n + \sqrt{n\overline{L}/\sigma}\big) \cdot \log \frac{F(x_0)-F(x^*)}{\varepsilon}\big)$ iterations.*

Theorem 6.2 (ext. of Theorem 4.1) *If each $f_i(x)$ is convex and L_i-smooth with respect to some norm $\|\cdot\|$, $V_x(y)$ is a Bregman divergence function for $\|\cdot\|$, and $\psi(\cdot)$ is not necessarily strongly convex, then* `Katyusha2`$^{\mathsf{ns}}(x_0, S, (L_1, \dots, L_n))$ *satisfies*

$$\mathbb{E}\big[F(\widetilde{x}^S)\big] - F(x^*) \le O\Big(\frac{F(x_0) - F(x^*)}{S^2} + \frac{\overline{L}V_{x_0}(x^*)}{nS^2}\Big) \ .$$

In other words, `Katyusha2`$^{\mathsf{ns}}$ *achieves an ε-additive error (i.e., $\mathbb{E}\big[F(\widetilde{x}^S)\big] - F(x^*) \le \varepsilon$) using at most $O\Big(\frac{n\sqrt{F(x_0)-F(x^*)}}{\sqrt{\varepsilon}} + \frac{\sqrt{nLV_{x_0}(x^*)}}{\sqrt{\varepsilon}}\Big)$ iterations.*

The proofs of Theorem 6.1 and Theorem 6.2 follow exactly the same proof structures of Theorem 2.1 and Theorem 4.1, so we include them only in Appendix E.

6.2 Remarks

We highlight one main difference between the proof of `Katyusha2` and that of `Katyusha`: if ξ is a random vector and $\|\cdot\|$ is an arbitrary norm, we do not necessarily have $\mathbb{E}[\|\xi-\mathbb{E}[\xi]\|_*^2] \le \mathbb{E}[\|\xi\|_*^2]$. Therefore, we only used $\mathbb{E}[\|\xi-\mathbb{E}[\xi]\|_*^2] \le 2\mathbb{E}[\|\xi\|_*^2]+2\|\mathbb{E}[\xi]\|_*^2$ (see Lemma E.2) and this loses a constant factor in some parameters. (For instance, α now becomes $\frac{1}{9\tau_1\overline{L}}$ as opposed to $\frac{1}{3\tau_1 L}$).

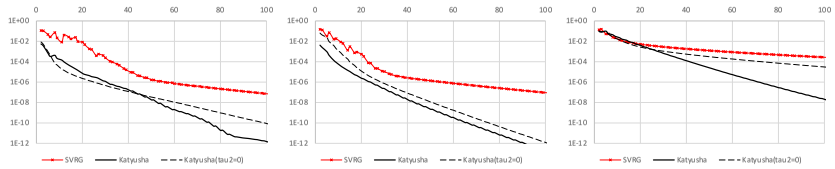
More interestingly, one may ask how our revised algorithms `Katyusha2` or `Katyusha2`$^{\mathsf{ns}}$ perform in the mini-batch setting (just like we have studied in Section 5 for the Euclidean case). We are optimistic here, but unfortunately do not have a clean worst-case statement for how much speed-up we can get. The underlying reason is that, if $\mathcal{D}$ is a distribution for vectors, $\mu = \mathbb{E}_{\xi\sim\mathcal{D}}[\xi]$ is its expectation, and $\xi_1,\dots,\xi_b$ are b i.i.d. samples from $\mathcal{D}$, then letting $\overline{\xi} = \frac{1}{b}(\xi_1+\cdots+\xi_b)$, we do not necessarily have $\mathbb{E}[\|\overline{\xi}-\mu\|_*^2] \le \frac{1}{b}\mathbb{E}_{\xi\sim\mathcal{D}}[\|\xi-\mu\|_*^2]$. In other words, using a mini-batch version of the gradient estimator, the "variance" with respect to an arbitrary norm may not necessarily go down by a factor of b. For such reason, in the mini-batch setting, the best total work we can cleanly state, say for `Katyusha2` in the SC setting, is only $O\Big(\big(n+\sqrt{bn\overline{L}/\sigma}\big)\cdot\log\frac{F(x_0)-F(x^*)}{\varepsilon}\Big)$.

7. Empirical Evaluations

We conclude this paper with empirical evaluations to our theoretical speed-ups. We work on Lasso and ridge regressions (with regularizer $\frac{\lambda}{2}\|x\|^2$ for ridge and regularizer $\lambda\|x\|_1$ for Lasso) on the following six datasets: adult, web, mnist, rcv1, covtype, sensit. We defer dataset and implementation details to Appendix B.

Algorithms and Parameter Tuning. We have implemented the following algorithms, all with mini-batch size 1 for this version of the paper:

- SVRG (Johnson and Zhang, 2013) with default epoch length $m = 2n$. We tune only *one parameter*: the learning rate.
- `Katyusha` for ridge and `Katyusha`$^{\mathsf{ns}}$ for Lasso. We tune only *one parameter*: the learning rate.
- SAGA (Defazio et al., 2014). We tune only *one parameter*: the learning rate.
- Catalyst (Lin et al., 2015) on top of SVRG. We tune *three parameters*: SVRG's learning rate, Catalyst's learning rate, as well as the regularizer weight in the Catalyst reduction.
- APCG (Lin et al., 2014). We tune the learning rate. For Lasso, we also tune the ℓ_2 regularizer weight.
- APCG+AdaptReg (Lasso only). Since APCG intrinsically require an ℓ_2 regularizer to be added on Lasso, we apply AdaptReg from Allen-Zhu and Hazan (2016b) to adaptively learn this regularizer and improve APCG's performance. Two parameters to be tuned: APCG's learning rate and σ_0 in AdaptReg.



(a) web, ridge $\lambda = 10^{-6}$ (b) mnist, ridge $\lambda = 10^{-6}$ (c) rcv1, ridge $\lambda = 10^{-6}$

Figure 1: Comparing SVRG vs. `Katyusha` vs. `Katyusha` with $\tau_2 = 0$.

All of the parameters were equally, fairly, and automatically tuned by our code base. For interested readers, we discuss more details in Appendix B.

We emphasize that `Katyusha` is *as simple as SAGA or SVRG in terms of parameter tuning*. In contrast, APCG for Lasso requires two parameters to be tuned, and Catalyst requires three. (Lin, 2016)

Performance Plots. Following the tradition of ERM experiments, we use the number of "passes" of the dataset as the x-axis. Letting n be the number of feature vectors, each new stochastic gradient computation $\nabla f_i(\cdot)$ counts as $1/n$ pass, and a full gradient computation $\nabla f(\cdot)$ counts as 1 pass.

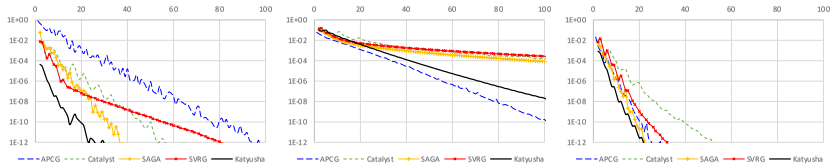
The y-axis in our plots represents the training objective distance to the minimum. Since we aim to evaluate our theoretical finding, we did not include the test error. We emphasize that it is practically also crucial to study high-accuracy regimes (such as objective distance $\le 10^{-7}$). This is because nowadays there is an increasing number of methods that reduce large-scale machine learning tasks to multiple black-box calls to ERM solvers (Allen-Zhu and Li, 2017b,a; Frostig et al., 2016). In all such applications, due to error blowups between oracle calls, the ERM solver is required to be *very accurate in training error*.

7.1 Effectiveness of Katyusha Momentum

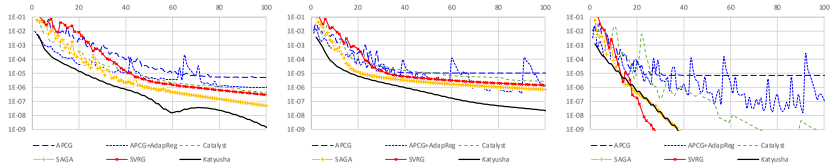
In our `Katyusha` method, τ_1 controls to the classical Nesterov's momentum and τ_2 controls our newly introduced Katyusha momentum. We find in our theory that setting $\tau_2 = 1/2$ is a good choice so we universally set it to be $1/2$ without tuning in all our experiments. (Of course, if time permits, tuning τ_2 could only help in performance.)

Before this paper, researchers have tried heuristics that is to add Nesterov's momentums directly to stochastic gradient methods (Nitanda, 2014), and this corresponds to setting $\tau_2 = 0$ in `Katyusha`. In Figure 1, we compare `Katyusha` with $\tau_2 = 1/2$ and $\tau_2 = 0$ in order to illustrate the importance and effectiveness of our Katyusha momentum.

We conclude that the old heuristics (i.e., $\tau_2 = 0$) sometimes indeed make the method faster after careful parameter tuning. However, for certain tasks such as Figure 1(c), without Katyusha momentum the algorithm does not even enjoy an accelerated convergence rate.



(a) covtype, ridge $\lambda = 10^{-6}$ (small) (b) rcv1, ridge $\lambda = 10^{-7}$ (small) (c) adult, ridge $\lambda = 10^{-4}$ (large)



(d) web, ridge $\lambda = 10^{-6}$ (small) (e) mnist, ridge $\lambda = 10^{-6}$ (small) (f) sensit, ridge $\lambda = 10^{-4}$ (large)

Figure 2: Some representative performance charts where λ is the regularizer weight. See Figure 3 and Figure 4 in the appendix for the full plots.

7.2 Performance Comparison Across Algorithms

For each of the six datasets and each objective (ridge or lasso), we experiment on three different magnitudes of regularizer weights.[18] This totals 36 performance charts, and we include them in full at the end of this paper. For the sake of cleanness, in Figure 2 we select 6 representative charts for ridge regression and make the following observations.

- Accelerated methods are more powerful when the regularizer weights are small (cf. Shalev-Shwartz and Zhang (2014); Allen-Zhu et al. (2016c); Lin et al. (2014)). For instance, Figure 2(c) and 2(f) are for large values of λ and `Katyusha` performs relatively the same as compared with SVRG / SAGA; however, `Katyusha` significantly outperforms SVRG / SAGA for small values of λ, see for instance Figure 2(b) and 2(e).
- `Katyusha` almost always either outperform or equal-perform its competitors. The only notable place it gets outperformed is by SVRG (see Figure 2(f)); however, this performance gap cannot be large because `Katyusha` is capable of recovering SVRG if $\tau_1 = \tau_2 = 0$.[19]
- Catalyst does not work as beautiful as its theory in high-accuracy regimes, even though we have carefully tuned parameters α_0 and κ in Catalyst in addition to its learning rate. Indeed, in Figure 2(a), 2(c) and 2(f) Catalyst (which is a reduction on SVRG) is outperformed by SVRG.
- APCG performs poorly on all Lasso tasks (cf. Figure 2(d), 2(e), 2(f)) because it is not designed for non-SC objectives. The reduction in Allen-Zhu and Hazan (2016b) helps to fix this issue, but not by a lot.
- APCG can sometimes be largely dominated by SVRG or SAGA (cf. Figure 2(f)): this is because for datasets such as sensit, dual-based methods (such as APCG) cannot make use of the implicity local strong convexity in the objective. In such cases, `Katyusha` is not lost to SVRG or SAGA.

18. We choose three values λ that are powers of 10 and around $10/n, 1/n, 1/10n$. This range can be verified to contain the best regularization weights using cross validation.
19. The only reason `Katyusha` does not match the performance of SVRG in Figure 2(f) is because we have not tuned parameter τ_2. If we also tune τ_2 for the best performance, `Katyusha` shall no longer be outperformed by SVRG. In any case, it is not really necessary to tune τ_2 because the performance of `Katyusha` is already superb.

8. Conclusion

The Katyusha momentum technique introduced in this paper gives rise to accelerated convergence rates even in the stochastic setting. For many classes of the problems, such convergence rates are the first to match the theoretical lower bounds (Woodworth and Srebro, 2016). The algorithms generated by Katyusha momentum are simple yet highly practical and parallelizable.

More importantly, this new technique has the potential to enrich our understanding of accelerated methods in a broader sense. Currently, although acceleration methods are becoming more and more important to the field of computer science, they are still often regarded as "analytical tricks" (Juditsky, 2013; Bubeck et al., 2015) and lacking complete theoretical understanding. The Katyusha momentum presented in this paper, however, adds a new level of decoration on top of the classical Nesterov momentum. This decoration is shown valuable for stochastic problems in this paper, but may also lead to future applications as well. In general, the author hopes that the technique and analysis in this paper could facilitate more studies in this field and thus become a stepping stone towards the ultimate goal of unveiling the mystery of acceleration.

Acknowledgements

We would like to specially thank Shai Shalev-Shwartz for useful feedbacks and suggestions on this paper, thank Blake Woodworth and Nati Srebro for pointer to their paper (Woodworth and Srebro, 2016), thank Guanghui Lan for correcting our citation of (Dang and Lan, 2014), thank Weston Jackson, Xu Chen and Zhe Li for verifying the proofs and correcting typos, thank Hongzhou Lin for discussing the experiments, thank Robert Hannah for discussing lower bounds, and thank anonymous reviewers for a number of writing suggestions. This paper is partially supported by an NSF Grant, no. CCF-1412958, and a Microsoft Research Grant, no. 0518584. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF or Microsoft.

Appendix

Appendix A. Other Related Works

For smooth convex minimization problems, (full) gradient descent converges at a rate $\frac{L}{\varepsilon}$ —or $\frac{L}{\sigma}\log\frac{1}{\varepsilon}$ if the objective is σ-strongly convex. This is not optimal among the class of first-order methods. Nesterov showed that the optimal rate should be $\frac{\sqrt{L}}{\sqrt{\varepsilon}}$ —or $\frac{\sqrt{L}}{\sqrt{\sigma}}\log\frac{1}{\varepsilon}$ if the objective is σ-strongly convex— and this was achieved by his celebrated accelerated (full) gradient descent method (Nesterov, 1983).

Sum-of-Nonconvex Optimization. One important generalization of Problem (1.1) is the case when the functions $f_i(x)$ are non-convex but their average $f(x)=\frac{1}{n}\sum_{i=1}^n f_i(x)$ is convex. Solvers for this "sum-of-nonconvex" setting can be applied to PCA/SVD, online eigenvector, general non-convex optimization, and more. See (Allen-Zhu, 2018a) and the references therein.

Variance reduction was first introduced to solve this problem by Shalev-Shwartz (2016), and APPA/Catalyst also accelerates SVRG for this problem (Garber et al., 2016). One can similarly ask whether one can design a *directly accelerated* method for this more general problem, and this was achieved by the `KatyushaX` method in (Allen-Zhu, 2018a). It is a sister paper to us but uses very different sets of techniques.

Online Stochastic Optimization. Some literatures also focus on the (more general) online variant of Problem (1.1), that is for n being sufficiently large so that the algorithm cannot access $f(x)$ or compute its full gradient $\nabla f(x)$. In this regime, without additional assumption, the optimal convergence rate is $1/\varepsilon^2$ (or $1/\varepsilon$ if the function is strongly convex). This is obtained by SGD and its hybrid variants (Lan, 2011; Hu et al., 2009).

Coordinate Descent. Another way to define gradient estimator is to set $\widetilde{\nabla}_k = d\nabla_j f(x_k)$ where j is a random coordinate. This is *(randomized) coordinate descent* as opposed to stochastic gradient descent. Designing accelerated methods for coordinate descent is significantly easier than designing that for stochastic gradient descent, and has indeed been done in many previous results including (Nesterov, 2012; Lin et al., 2014; Lu and Xiao, 2013; Allen-Zhu et al., 2016c).[20] The fastest rate is $O\big(\sum_i \sqrt{L_i/\varepsilon}\big)$ where parameters L_i correspond to the coordinate smoothness of $f(x)$ (Allen-Zhu et al., 2016c). Coordinate descent *cannot* be applied to solve Problem (1.1) because in our *stochastic* setting, only one copy $\nabla f_i(\cdot)$ is computed in every iteration.

Hybrid Stochastic Methods. Several recent results study hybrid methods with convergence rates that are generally *non-accelerated* and only accelerated in *extreme cases*.

- Lan (2011); Hu et al. (2009) obtained iteration complexity of the form $O(L/\sqrt{\varepsilon}+\sigma/\varepsilon^2)$ in the presence of stochastic gradient with variance σ. These results can be interpreted as follows, if σ is very small, then one can directly apply Nesterov's accelerated gradient method and achieve $O(L/\sqrt{\varepsilon})$; or if σ is large then they match the SGD iteration complexity $O(\sigma/\varepsilon^2)$. For Problem (1.1), these algorithms do not give faster running time than `Katyusha` unless σ is very small.[21]
- Nitanda (2014) adds momentum to the non-accelerated variance-reduction method in a naive manner. It corresponds to this paper but *without* Katyusha momentum (i.e., $\tau_2=0$). The theoretical running time of Nitanda (2014) is always slower than this paper and cannot even outperform SVRG (Johnson and Zhang, 2013; Zhang et al., 2013) unless $\kappa > n^2$ —which is usually false in practice (see page 7 of Nitanda (2014)).[22] We have included an experiment in Section 7.1 to illustrate why Katyusha momentum is necessary.

Linear Coupling. Allen-Zhu and Orecchia (2017) proposed a framework called *linear coupling* that facilitates the design of accelerated gradient methods. The simplest use of linear coupling can reconstruct Nesterov's accelerated full-gradient method (Allen-Zhu and Orecchia, 2017), or to provide faster coordinate descent (Allen-Zhu et al., 2016c). More careful use of linear coupling can also give accelerated methods for non-smooth problems (such as positive LP (Allen-Zhu and Orecchia, 2015b,a), positive SDP (Allen-Zhu et al., 2016a), matrix scaling (Allen-Zhu et al., 2017)) or for general non-convex problems (Allen-Zhu and Hazan, 2016a). This present paper falls into this linear-coupling framework, but our Katyusha momentum technique was not present in any of these cited results.

Acceleration in Nonconvex Optimization. One can also ask how does acceleration help in non-convex optimization? This is a new area with active research going on.

In the deterministic setting, under standard Lipschitz smoothness, gradient descent finds a point x with $\|\nabla f(x)\|\le\varepsilon$ in $O(\varepsilon^{-2})$ iterations (Nesterov, 2004), and acceleration is not known to help. If second-order Lipschitz smoothness is added, then one can use momentum to non-trivially improve the rate to $O(\varepsilon^{-1.75})$ (Carmon et al., 2016; Agarwal et al., 2017).

In the finite-sum stochastic setting, gradient descent finds a point x with $\|\nabla f(x)\|\le\varepsilon$ in $T=O(n\varepsilon^{-2})$ stochastic gradient computations under standard Lipschitz smoothness. If second-order Lipschitz smoothness is added, then one can use momentum to non-trivially improve the complexity $T=O(n\varepsilon^{-1.5}+n^{0.75}\varepsilon^{-1.75})$ (Agarwal et al., 2017).

In the online stochastic setting, SGD finds a point x with $\|\nabla f(x)\|\le\varepsilon$ in $T=O(\varepsilon^{-4})$ stochastic gradient iterations under the standard Lipschitz smoothness assumption. Perhaps surprisingly, without using momentum, one can already improve this rate to $T=O(\varepsilon^{-3.5})$ (using SGD) (Allen-Zhu, 2018b), $T=O(\varepsilon^{-3.333})$ (Lei et al., 2017), or even to $T=O(\varepsilon^{-3.25})$ (Allen-Zhu, 2017) if second-order Lipschitz smoothness is added. It is unclear whether such rates can be improved using momentum. We stress that, even if "improved rates" can be obtained using momentum, one also needs to prove from the lower-bound side that such "improved rates" cannot be obtained by any momentum-free method.

20. The reason behind it can be understood as follows. If a function $f(\cdot)$ is L smooth with respect to coordinate j, then a coordinate descent step $x'\leftarrow x-\frac{1}{L}\nabla_j f(x)\mathbf{e}_j$ always decreases the objective, i.e., $f(x+\frac{1}{L}\nabla_j f(x)\mathbf{e}_j)<f(x)$. In contrast, this is *false* for stochastic gradient descent, because $f(x_k-\eta\widetilde{\nabla}_k)$ may be even larger than $f(x_k)$.

21. When σ is large, even if n is large, the iteration complexity of (Lan, 2011; Hu et al., 2009) becomes $O(\sigma/\varepsilon^2)$. In this regime, almost all variance-reduction methods, including SVRG and `Katyusha`, can be shown to satisfy $\varepsilon\le O(\frac{\sqrt{\sigma}}{\sqrt{T}})$ within the first epoch, if the learning rates are appropriately chosen. Therefore, `Katyusha` and SVRG are no slower than Lan (2011); Hu et al. (2009).

22. Nitanda's method is usually not considered as an accelerated method, since it requires mini-batch size to be very large in order to be accelerated. If mini-batch is large then one can use full-gradient method directly and acceleration is trivial. This is confirmed by (Konečnỳ et al., 2016, Section IV.F). In contrast, our acceleration holds even if mini-batch size is 1.

Appendix B. Experiment Details

The datasets we used in this paper are downloaded from the LibSVM website (Fan and Lin):

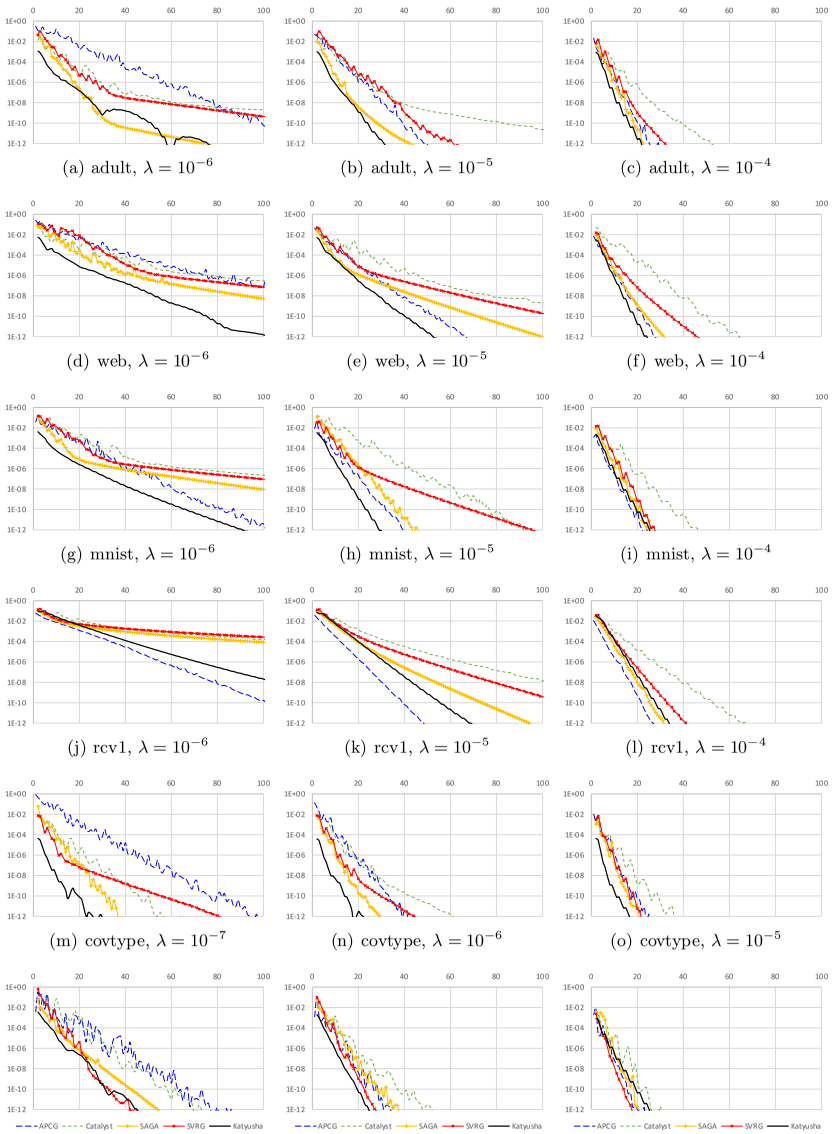
- the adult (a9a) dataset (32,561 samples and 123 features).
- the web (w8a) dataset (49,749 samples and 300 features).
- the covtype (binary.scale) dataset (581,012 samples and 54 features).
- the mnist (class 1) dataset (60,000 samples and 780 features).
- the rcv1 (train.binary) dataset (20,242 samples and 47,236 features).
- the sensit (combined) dataset (78,823 samples and 100 features).

To make easier comparison across datasets, we scale every vector by the average Euclidean norm of all the vectors in the dataset. In other words, we ensure that the data vectors have an average Euclidean norm 1. This step is for comparison only and not necessary in practice.

Parameter-tuning details. We select learning rates from the set $\{10^{-k}, 2\times 10^{-k}, 5\times 10^{-k} : k\in\mathbb{Z}\}$, and select regularizer weights (for APCG) from the set $\{10^{-k} : k\in\mathbb{Z}\}$. We have fully automated the parameter tuning procedure to ensure a fair and strong comparison.

While the learning rates were explicitly defined for SVRG and SAGA, there were implicit for all accelerated methods. For Catalyst, the learning rate is in fact their α_0 in the paper (Lin, 2016). Instead of choosing it to be the theory-predicted value, we multiply it with an extra factor to be tuned and call this factor the "learning rate". Similarly, for `Katyusha` and `Katyusha`$^{\mathsf{ns}}$, we multiply the theory-predicted τ_1 with an extra factor and this serves as a learning rate. For APCG, we use their Algorithm 1 in the paper and multiply their theory-predicted μ with an extra factor.

For Catalyst, in principle one also has to tune the stopping criterion. After communicating with an author of Catalyst, we learned that one can terminate the inner loop whenever the duality gap becomes no more than, say one fourth, of the last duality gap from the previous epoch (Lin, 2016). This stopping criterion was also found by the authors of (Allen-Zhu and Hazan, 2016b) to be a good choice for reduction-based methods.



(a) adult, $\lambda = 10^{-6}$ (b) adult, $\lambda = 10^{-5}$ (c) adult, $\lambda = 10^{-4}$

(d) web, $\lambda = 10^{-6}$ (e) web, $\lambda = 10^{-5}$ (f) web, $\lambda = 10^{-4}$

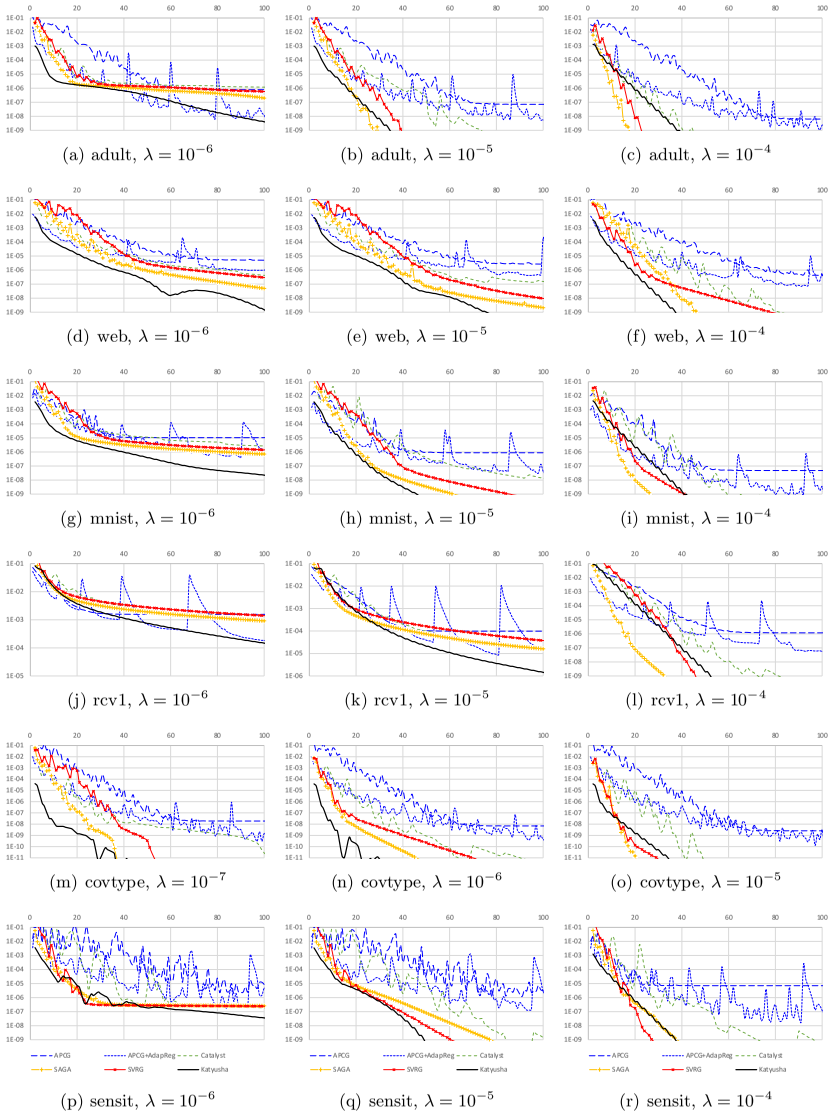
(g) mnist, $\lambda = 10^{-6}$ (h) mnist, $\lambda = 10^{-5}$ (i) mnist, $\lambda = 10^{-4}$

(j) rcv1, $\lambda = 10^{-6}$ (k) rcv1, $\lambda = 10^{-5}$ (l) rcv1, $\lambda = 10^{-4}$

(m) covtype, $\lambda = 10^{-7}$ (n) covtype, $\lambda = 10^{-6}$ (o) covtype, $\lambda = 10^{-5}$

(p) sensit, $\lambda = 10^{-6}$ (q) sensit, $\lambda = 10^{-5}$ (r) sensit, $\lambda = 10^{-4}$

Figure 3: Experiments on ridge regression with ℓ_2 regularizer weight λ.



(a) adult, $\lambda = 10^{-6}$ (b) adult, $\lambda = 10^{-5}$ (c) adult, $\lambda = 10^{-4}$

(d) web, $\lambda = 10^{-6}$ (e) web, $\lambda = 10^{-5}$ (f) web, $\lambda = 10^{-4}$

(g) mnist, $\lambda = 10^{-6}$ (h) mnist, $\lambda = 10^{-5}$ (i) mnist, $\lambda = 10^{-4}$

(j) rcv1, $\lambda = 10^{-6}$ (k) rcv1, $\lambda = 10^{-5}$ (l) rcv1, $\lambda = 10^{-4}$

(m) covtype, $\lambda = 10^{-7}$ (n) covtype, $\lambda = 10^{-6}$ (o) covtype, $\lambda = 10^{-5}$

(p) sensit, $\lambda = 10^{-6}$ (q) sensit, $\lambda = 10^{-5}$ (r) sensit, $\lambda = 10^{-4}$

Figure 4: Experiments on Lasso with ℓ_1 regularizer weight λ.

Appendix C. Appendix for Section 4

C.1 Proof of Theorem 4.1

Proof [Proof of Theorem 4.1] First of all, the parameter choices satisfy the presumptions in Lemma 2.6, so again by defining $D_k \stackrel{\text{def}}{=} F(y_k) - F(x^*)$ and $\widetilde{D}^s \stackrel{\text{def}}{=} F(\widetilde{x}^s) - F(x^*)$, we can rewrite Lemma 2.7 as follows:

$$0 \le \frac{\alpha_s(1-\tau_{1,s}-\tau_2)}{\tau_{1,s}} D_k - \frac{\alpha_s}{\tau_{1,s}} \mathbb{E}\big[D_{k+1}\big] + \frac{\alpha_s \tau_2}{\tau_{1,s}} \widetilde{D}^s + \frac{1}{2}\|z_k - x^*\|^2 - \frac{1}{2}\mathbb{E}\big[\|z_{k+1} - x^*\|^2\big] \ .$$

Summing up the above inequality for all the iterations $k = sm, sm+1, \ldots, sm+m-1$, we have

$$\begin{aligned} &\mathbb{E}\Big[\alpha_s \frac{1-\tau_{1,s}-\tau_2}{\tau_{1,s}} D_{(s+1)m} + \alpha_s \frac{\tau_{1,s}+\tau_2}{\tau_{1,s}} \sum_{j=1}^{m} D_{sm+j}\Big] \\ &\le \alpha_s \frac{1-\tau_{1,s}-\tau_2}{\tau_{1,s}} D_{sm} + \alpha_s \frac{\tau_2}{\tau_{1,s}} m\widetilde{D}^s + \frac{1}{2}\|z_{sm} - x^*\|^2 - \frac{1}{2}\mathbb{E}\big[\|z_{(s+1)m} - x^*\|^2\big] \ . \end{aligned} \tag{C.1}$$

Note that in the above inequality we have assumed all the randomness in the first $s-1$ epochs are fixed and the only source of randomness comes from epoch s.

If we define $\widetilde{x}^s = \frac{1}{m}\sum_{j=1}^{m} y_{(s-1)m+j}$, then by the convexity of function $F(\cdot)$ we have $m\widetilde{D}^s \le \sum_{j=1}^{n} D_{(s-1)m+j}$. Therefore, using the parameter choice $\alpha_s = \frac{1}{3\tau_{1,s}L}$, for every $s \ge 1$ we can derive from (C.1) that

$$\begin{aligned} &\mathbb{E}\Big[\frac{1}{\tau_{1,s}^2} D_{(s+1)m} + \frac{\tau_{1,s}+\tau_2}{\tau_{1,s}^2} \sum_{j=1}^{m-1} D_{sm+j}\Big] \\ &\le \frac{1-\tau_{1,s}}{\tau_{1,s}^2} D_{sm} + \frac{\tau_2}{\tau_{1,s}^2} \sum_{j=1}^{m-1} D_{(s-1)m+j} + \frac{3L}{2}\|z_{sm} - x^*\|^2 - \frac{3L}{2}\mathbb{E}\big[\|z_{(s+1)m} - x^*\|^2\big] \ . \end{aligned} \tag{C.2}$$

For the base case $s = 0$, we can also rewrite (C.1) as

$$\begin{aligned} &\mathbb{E}\Big[\frac{1}{\tau_{1,0}^2} D_m + \frac{\tau_{1,0}+\tau_2}{\tau_{1,0}^2} \sum_{j=1}^{m-1} D_j\Big] \\ &\le \frac{1-\tau_{1,0}-\tau_2}{\tau_{1,0}^2} D_0 + \frac{\tau_2 m}{\tau_{1,0}^2} \widetilde{D}^0 + \frac{3L}{2}\|z_0 - x^*\|^2 - \frac{3L}{2}\mathbb{E}\big[\|z_m - x^*\|^2\big] \ . \end{aligned} \tag{C.3}$$

At this point, if we choose $\tau_{1,s} = \frac{2}{s+4} \le \frac{1}{2}$, it satisfies

$$\frac{1}{\tau_{1,s}^2} \ge \frac{1-\tau_{1,s+1}}{\tau_{1,s+1}^2} \quad \text{and} \quad \frac{\tau_{1,s}+\tau_2}{\tau_{1,s}^2} \ge \frac{\tau_2}{\tau_{1,s+1}^2} \ .$$

Using these two inequalities, we can telescope (C.3) and (C.2) for all $s = 0, 1, \ldots, S-1$. We obtain in the end that

$$\begin{aligned} &\mathbb{E}\Big[\frac{1}{\tau_{1,S-1}^2} D_{Sm} + \frac{\tau_{1,S-1}+\tau_2}{\tau_{1,S-1}^2} \sum_{j=1}^{m-1} D_{(S-1)m+j} + \frac{3L}{2}\|z_{Sm} - z^*\|^2\Big] \\ &\qquad \le \frac{1-\tau_{1,0}-\tau_2}{\tau_{1,0}^2} D_0 + \frac{\tau_2 m}{\tau_{1,0}^2} \widetilde{D}^0 + \frac{3L}{2}\|z_0 - x^*\|^2 \end{aligned} \tag{C.4}$$

Since we have $\widetilde{D}^S \le \frac{1}{m}\sum_{j=1}^m D_{(S-1)m+j}$ which is no greater than $\frac{2\tau_{1,S-1}^2}{m}$ times the left hand side of (C.4), we conclude that

$$\mathbb{E}\big[F(\widetilde{x}^S)-F(x^*)\big] = \mathbb{E}\big[\widetilde{D}^S\big] \le O\big(\tfrac{\tau_{1,S}^2}{m}\big)\cdot\Big(\frac{1-\tau_{1,0}-\tau_2}{\tau_{1,0}^2}D_0 + \frac{\tau_2 m}{\tau_{1,0}^2}\widetilde{D}^0 + \frac{3L}{2}\|z_0-x^*\|^2\Big)$$
$$= O\big(\tfrac{1}{mS^2}\big)\cdot\Big(m\big(F(x_0)-F(x^*)\big)+L\|x_0-x^*\|^2\Big)\ .$$

■

Appendix D. Appendix for Section 5

D.1 One-Iteration Analysis

Similar as Section 2.1, we first analyze the behavior of `Katyusha1` in a single iteration (i.e., for a fixed k). We view y_k, z_k and x_{k+1} as fixed in this section so the only randomness comes from the choice of i in iteration k. We abbreviate in this subsection by $\widetilde{x}=\widetilde{x}^s$ where s is the epoch that iteration k belongs to, and denote by $\sigma_{k+1}^2 \stackrel{\text{def}}{=} \|\nabla f(x_{k+1})-\widetilde{\nabla}_{k+1}\|^2$.

Our first lemma is analogous to Lemma 2.3, where note that we have replaced the use of L in Lemma 2.3 with $L_\diamond \ge L$:

Lemma D.1 (proximal gradient descent) *If $L_\diamond \ge L$ and*

$$y_{k+1} = \arg\min_y \Big\{\frac{3L_\diamond}{2}\|y-x_{k+1}\|^2 + \langle\widetilde{\nabla}_{k+1}, y-x_{k+1}\rangle + \psi(y)-\psi(x_{k+1})\Big\}\ , \qquad \textit{and}$$
$$\mathsf{Prog}(x_{k+1}) \stackrel{\text{def}}{=} -\min_y \Big\{\frac{3L_\diamond}{2}\|y-x_{k+1}\|^2 + \langle\widetilde{\nabla}_{k+1}, y-x_{k+1}\rangle + \psi(y)-\psi(x_{k+1})\Big\} \ge 0\ ,$$

we have (where the expectation is only over the randomness of $\widetilde{\nabla}_{k+1}$)

$$F(x_{k+1}) - \mathbb{E}\big[F(y_{k+1})\big] \ge \mathbb{E}\big[\mathsf{Prog}(x_{k+1})\big] - \frac{1}{4L_\diamond}\mathbb{E}\big[\sigma_{k+1}^2\big]\ .$$

Proof

$$\begin{aligned}\mathsf{Prog}(x_{k+1}) &= -\min_y\Big\{\frac{3L_\diamond}{2}\|y-x_{k+1}\|^2 + \langle\widetilde{\nabla}_{k+1}, y-x_{k+1}\rangle + \psi(y)-\psi(x_{k+1})\Big\}\\ &\stackrel{①}{=} -\Big(\frac{3L_\diamond}{2}\|y_{k+1}-x_{k+1}\|^2 + \langle\widetilde{\nabla}_{k+1}, y_{k+1}-x_{k+1}\rangle + \psi(y_{k+1})-\psi(x_{k+1})\Big)\\ &= -\Big(\frac{L_\diamond}{2}\|y_{k+1}-x_{k+1}\|^2 + \langle\nabla f(x_{k+1}), y_{k+1}-x_{k+1}\rangle + \psi(y_{k+1})-\psi(x_{k+1})\Big)\\ &\quad + \Big(\langle\nabla f(x_{k+1})-\widetilde{\nabla}_{k+1}, y_{k+1}-x_{k+1}\rangle - L_\diamond\|y_{k+1}-x_{k+1}\|^2\Big)\\ &\stackrel{②}{\le} -\Big(f(y_{k+1})-f(x_{k+1})+\psi(y_{k+1})-\psi(x_{k+1})\Big) + \frac{1}{4L_\diamond}\|\nabla f(x_{k+1})-\widetilde{\nabla}_{k+1}\|^2\ .\end{aligned}$$

Above, ① is by the definition of y_{k+1}, and ② uses the smoothness of function $f(\cdot)$, as well as Young's inequality $\langle a,b\rangle - \frac{1}{2}\|b\|^2 \le \frac{1}{2}\|a\|^2$. Taking expectation on both sides we arrive at the desired result. ■

The following lemma is analogous to Lemma 2.4. The main difference is that since we have not chosen a mini-batch of size b, one should expect the variance to decrease by a factor of b. Also, since we are in the non-uniform case one should expect the use of L in Lemma 2.4 to be replaced with $\overline{L}$:

Lemma D.2 (variance upper bound)

$$\mathbb{E}\big[\|\widetilde{\nabla}_{k+1}-\nabla f(x_{k+1})\|^2\big] \le \frac{2\overline{L}}{b}\cdot\big(f(\widetilde{x})-f(x_{k+1})-\langle\nabla f(x_{k+1}), \widetilde{x}-x_{k+1}\rangle\big)\ .$$

Proof Each $f_i(x)$, being convex and L_i-smooth, implies the following inequality which is classical in convex optimization and can be found for instance in Theorem 2.1.5 of the textbook of Nesterov (2004).

$$\|\nabla f_i(x_{k+1})-\nabla f_i(\widetilde{x})\|^2 \le 2L_i\cdot\big(f_i(\widetilde{x})-f_i(x_{k+1})-\langle\nabla f_i(x_{k+1}),\widetilde{x}-x_{k+1}\rangle\big)$$

Therefore, taking expectation over the random choice of i, we have

$$\begin{aligned}&\mathbb{E}\big[\|\widetilde{\nabla}_{k+1}-\nabla f(x_{k+1})\|^2\big]\\ &= \mathbb{E}_{S_k}\Big[\Big\|\Big(\frac{1}{b}\sum_{i\in S_k}\Big(\nabla f(\widetilde{x})+\frac{1}{np_i}\big(\nabla f_i(x_{k+1})-\nabla f_i(\widetilde{x})\big)\Big)\Big)-\nabla f(x_{k+1})\Big\|^2\Big]\\ &= \frac{1}{b}\mathbb{E}_{i\sim\mathcal{D}}\Big[\Big\|\Big(\nabla f(\widetilde{x})+\frac{1}{np_i}\big(\nabla f_i(x_{k+1})-\nabla f_i(\widetilde{x})\big)\Big)-\nabla f(x_{k+1})\Big\|^2\Big]\\ &= \frac{1}{b}\mathbb{E}_{i\sim\mathcal{D}}\Big[\Big\|\frac{1}{np_i}\big(\nabla f_i(x_{k+1})-\nabla f_i(\widetilde{x})\big)-\big(\nabla f(x_{k+1})-f(\widetilde{x})\big)\Big\|^2\Big]\\ &\stackrel{①}{\le} \frac{1}{b}\mathbb{E}_{i\sim\mathcal{D}}\Big[\Big\|\frac{1}{np_i}\big(\nabla f_i(x_{k+1})-\nabla f_i(\widetilde{x})\big)\Big\|^2\Big]\\ &\stackrel{②}{\le} \frac{1}{b}\cdot\sum_{i\in[n]}\frac{2L_i}{n^2p_i}\Big(f_i(\widetilde{x})-f_i(x_{k+1})-\langle\nabla f_i(x_{k+1}),\widetilde{x}-x_{k+1}\rangle\Big)\\ &= \frac{2\overline{L}}{b}\cdot\big(f(\widetilde{x})-f(x_{k+1})-\langle\nabla f(x_{k+1}),\widetilde{x}-x_{k+1}\rangle\big)\ .\end{aligned}$$

Above, ① is because for any random vector $\zeta\in\mathbb{R}^d$, it holds that $\mathbb{E}\|\zeta-\mathbb{E}\zeta\|^2 = \mathbb{E}\|\zeta\|^2-\|\mathbb{E}\zeta\|^2$; ② follows from the first inequality in this proof. ■

The next lemma is completely identical to Lemma 2.5 so we skip the proof.

Lemma D.3 (proximal mirror descent) *Suppose $\psi(\cdot)$ is σ-SC. Then, fixing $\widetilde{\nabla}_{k+1}$ and letting*

$$z_{k+1} = \arg\min_z\Big\{\frac{1}{2}\|z-z_k\|^2 + \alpha\langle\widetilde{\nabla}_{k+1}, z-z_k\rangle + \alpha\psi(z)-\alpha\psi(z_k)\Big\}\ ,$$

it satisfies for all $u\in\mathbb{R}^d$,

$$\alpha\langle\widetilde{\nabla}_{k+1}, z_{k+1}-u\rangle + \alpha\psi(z_{k+1})-\alpha\psi(u) \le -\frac{1}{2}\|z_k-z_{k+1}\|^2 + \frac{1}{2}\|z_k-u\|^2 - \frac{1+\alpha\sigma}{2}\|z_{k+1}-u\|^2\ .$$

The following lemma combines Lemma D.1, Lemma D.2 and Lemma D.3 all together, using the special choice of x_{k+1} which is a convex combination of y_k, z_k and $\widetilde{x}$:

Lemma D.4 (coupling step 1) *If $x_{k+1} = \tau_1 z_k + \tau_2 \widetilde{x} + (1-\tau_1-\tau_2) y_k$, where $\tau_1 \le \frac{1}{3\alpha L_\diamond}$ and $\tau_2 = \frac{\overline{L}}{2L_\diamond b}$,*

$$\begin{aligned}
&\alpha\langle \nabla f(x_{k+1}), z_k - u\rangle - \alpha\psi(u) \\
\le\ & \frac{\alpha}{\tau_1}\Big(F(x_{k+1}) - \mathbb{E}\big[F(y_{k+1})\big] + \tau_2 F(\widetilde{x}) - \tau_2 f(x_{k+1}) - \tau_2\langle \nabla f(x_{k+1}), \widetilde{x} - x_{k+1}\rangle\Big) \\
&+ \frac{1}{2}\|z_k - u\|^2 - \frac{1+\alpha\sigma}{2}\mathbb{E}\big[\|z_{k+1} - u\|^2\big] + \frac{\alpha(1-\tau_1-\tau_2)}{\tau_1}\psi(y_k) - \frac{\alpha}{\tau_1}\psi(x_{k+1}) \ .
\end{aligned}$$

Proof We first apply Lemma D.3 and get

$$\begin{aligned}
&\alpha\langle \widetilde{\nabla}_{k+1}, z_k - u\rangle + \alpha\psi(z_{k+1}) - \alpha\psi(u) \\
=\ & \alpha\langle \widetilde{\nabla}_{k+1}, z_k - z_{k+1}\rangle + \alpha\langle \widetilde{\nabla}_{k+1}, z_{k+1} - u\rangle + \alpha\psi(z_{k+1}) - \alpha\psi(u) \\
\le\ & \alpha\langle \widetilde{\nabla}_{k+1}, z_k - z_{k+1}\rangle - \frac{1}{2}\|z_k - z_{k+1}\|^2 + \frac{1}{2}\|z_k - u\|^2 - \frac{1+\alpha\sigma}{2}\|z_{k+1} - u\|^2 \ . \qquad \text{(D.1)}
\end{aligned}$$

By defining $v \stackrel{\text{def}}{=} \tau_1 z_{k+1} + \tau_2 \widetilde{x} + (1-\tau_1-\tau_2) y_k$, we have $x_{k+1} - v = \tau_1(z_k - z_{k+1})$ and therefore

$$\begin{aligned}
&\mathbb{E}\Big[\alpha\langle \widetilde{\nabla}_{k+1}, z_k - z_{k+1}\rangle - \frac{1}{2}\|z_k - z_{k+1}\|^2\Big] = \mathbb{E}\Big[\frac{\alpha}{\tau_1}\langle \widetilde{\nabla}_{k+1}, x_{k+1} - v\rangle - \frac{1}{2\tau_1^2}\|x_{k+1} - v\|^2\Big] \\
=\ & \mathbb{E}\Big[\frac{\alpha}{\tau_1}\Big(\langle \widetilde{\nabla}_{k+1}, x_{k+1} - v\rangle - \frac{1}{2\alpha\tau_1}\|x_{k+1} - v\|^2 - \psi(v) + \psi(x_{k+1})\Big) + \frac{\alpha}{\tau_1}\Big(\psi(v) - \psi(x_{k+1})\Big)\Big] \\
\overset{①}{\le}\ & \mathbb{E}\Big[\frac{\alpha}{\tau_1}\Big(\langle \widetilde{\nabla}_{k+1}, x_{k+1} - v\rangle - \frac{3L_\diamond}{2}\|x_{k+1} - v\|^2 - \psi(v) + \psi(x_{k+1})\Big) + \frac{\alpha}{\tau_1}\Big(\psi(v) - \psi(x_{k+1})\Big)\Big] \\
\overset{②}{\le}\ & \mathbb{E}\Big[\frac{\alpha}{\tau_1}\Big(F(x_{k+1}) - F(y_{k+1}) + \frac{1}{4L_\diamond}\sigma_{k+1}^2\Big) + \frac{\alpha}{\tau_1}\Big(\psi(v) - \psi(x_{k+1})\Big)\Big] \\
\overset{③}{\le}\ & \mathbb{E}\Big[\frac{\alpha}{\tau_1}\Big(F(x_{k+1}) - F(y_{k+1}) + \frac{\overline{L}}{2L_\diamond b}\big(f(\widetilde{x}) - f(x_{k+1}) - \langle \nabla f(x_{k+1}), \widetilde{x} - x_{k+1}\rangle\big)\Big) \\
&+ \frac{\alpha}{\tau_1}\Big(\tau_1\psi(z_{k+1}) + \tau_2\psi(\widetilde{x}) + (1-\tau_1-\tau_2)\psi(y_k) - \psi(x_{k+1})\Big)\Big] \ . \qquad \text{(D.2)}
\end{aligned}$$

Above, ① uses our choice $\tau_1 \le \frac{1}{3\alpha L}$, ② uses Lemma D.1, ③ uses Lemma D.2 together with the convexity of $\psi(\cdot)$ and the definition of v. Finally, noticing that $\mathbb{E}[\langle \widetilde{\nabla}_{k+1}, z_k - u\rangle] = \langle \nabla f(x_{k+1}), z_k - u\rangle$ and $\tau_2 = \frac{1}{2}$, we obtain the desired inequality by combining (D.1) and (D.2). ■

The next lemma simplifies the left hand side of Lemma D.4 using the convexity of $f(\cdot)$, and gives an inequality that relates the objective-distance-to-minimizer quantities $F(y_k) - F(x^*)$, $F(y_{k+1}) - F(x^*)$, and $F(\widetilde{x}) - F(x^*)$ to the point-distance-to-minimizer quantities $\|z_k - x^*\|^2$ and $\|z_{k+1} - x^*\|^2$.

Lemma D.5 (coupling step 2) *Under the same choices of τ_1, τ_2 as in Lemma D.4, we have*

$$\begin{aligned}
0 \le\ & \frac{\alpha(1-\tau_1-\tau_2)}{\tau_1}(F(y_k) - F(x^*)) - \frac{\alpha}{\tau_1}\big(\mathbb{E}\big[F(y_{k+1})\big] - F(x^*)\big) + \frac{\alpha\tau_2}{\tau_1}\big(F(\widetilde{x}) - F(x^*)\big) \\
&+ \frac{1}{2}\|z_k - x^*\|^2 - \frac{1+\alpha\sigma}{2}\mathbb{E}\big[\|z_{k+1} - x^*\|^2\big] \ .
\end{aligned}$$

Proof We first compute that

$$\begin{aligned}
&\alpha\big(f(x_{k+1}) - f(u)\big) \overset{①}{\le} \alpha\langle \nabla f(x_{k+1}), x_{k+1} - u\rangle \\
=\ & \alpha\langle \nabla f(x_{k+1}), x_{k+1} - z_k\rangle + \alpha\langle \nabla f(x_{k+1}), z_k - u\rangle \\
\overset{②}{=}\ & \frac{\alpha\tau_2}{\tau_1}\langle \nabla f(x_{k+1}), \widetilde{x} - x_{k+1}\rangle + \frac{\alpha(1-\tau_1-\tau_2)}{\tau_1}\langle \nabla f(x_{k+1}), y_k - x_{k+1}\rangle + \alpha\langle \nabla f(x_{k+1}), z_k - u\rangle \\
\overset{③}{\le}\ & \frac{\alpha\tau_2}{\tau_1}\langle \nabla f(x_{k+1}), \widetilde{x} - x_{k+1}\rangle + \frac{\alpha(1-\tau_1-\tau_2)}{\tau_1}(f(y_k) - f(x_{k+1})) + \alpha\langle \nabla f(x_{k+1}), z_k - u\rangle \ .
\end{aligned}$$

Above, ① uses the convexity of $f(\cdot)$, ② uses the choice that $x_{k+1} = \tau_1 z_k + \tau_2\widetilde{x} + (1-\tau_1-\tau_2)y_k$, and ③ uses the convexity of $f(\cdot)$ again. By applying Lemma D.4 to the above inequality, we have

$$\begin{aligned}
&\alpha\big(f(x_{k+1}) - F(u)\big) \le \frac{\alpha(1-\tau_1-\tau_2)}{\tau_1}(F(y_k) - f(x_{k+1})) \\
&+\frac{\alpha}{\tau_1}\Big(F(x_{k+1}) - \mathbb{E}\big[F(y_{k+1})\big] + \tau_2 F(\widetilde{x}) - \tau_2 f(x_{k+1})\Big) + \frac{1}{2}\|z_k - u\|^2 - \frac{1+\alpha\sigma}{2}\mathbb{E}\big[\|z_{k+1} - u\|^2\big] - \frac{\alpha}{\tau_1}\psi(x_{k+1})
\end{aligned}$$

which implies

$$\begin{aligned}
&\alpha\big(F(x_{k+1}) - F(u)\big) \le \frac{\alpha(1-\tau_1-\tau_2)}{\tau_1}(F(y_k) - F(x_{k+1})) \\
&+\frac{\alpha}{\tau_1}\Big(F(x_{k+1}) - \mathbb{E}\big[F(y_{k+1})\big] + \tau_2 F(\widetilde{x}) - \tau_2 F(x_{k+1})\Big) + \frac{1}{2}\|z_k - u\|^2 - \frac{1+\alpha\sigma}{2}\mathbb{E}\big[\|z_{k+1} - u\|^2\big] \ .
\end{aligned}$$

After rearranging and setting $u = x^*$, the above inequality yields

$$\begin{aligned}
0 \le\ & \frac{\alpha(1-\tau_1-\tau_2)}{\tau_1}(F(y_k) - F(x^*)) - \frac{\alpha}{\tau_1}\big(\mathbb{E}\big[F(y_{k+1}) - F(x^*)\big]\big) + \frac{\alpha\tau_2}{\tau_1}\big(F(\widetilde{x}) - F(x^*)\big) \\
&+ \frac{1}{2}\|z_k - x^*\|^2 - \frac{1+\alpha\sigma}{2}\mathbb{E}\big[\|z_{k+1} - x^*\|^2\big] \ .
\end{aligned}$$

■

D.2 Proof of Theorem 5.2

We are now ready to combine the analyses across iterations, and derive our final Theorem 5.2. Our proof next requires a careful telescoping of Lemma D.5 together with our specific parameter choices.

Proof [Proof of Theorem 5.2] Define $D_k \stackrel{\text{def}}{=} F(y_k) - F(x^*)$, $\widetilde{D}^s \stackrel{\text{def}}{=} F(\widetilde{x}^s) - F(x^*)$, and rewrite Lemma D.5:

$$0 \le \frac{(1-\tau_1-\tau_2)}{\tau_1}D_k - \frac{1}{\tau_1}D_{k+1} + \frac{\tau_2}{\tau_1}\mathbb{E}\big[\widetilde{D}^s\big] + \frac{1}{2\alpha}\|z_k - x^*\|^2 - \frac{1+\alpha\sigma}{2\alpha}\mathbb{E}\big[\|z_{k+1} - x^*\|^2\big] \ .$$

At this point, let us θ be an arbitrary value in $\big[1, 1+\alpha\sigma\big]$ and multiply the above inequality by θ^j for each $k = sm + j$. Then, we sum up the resulting m inequalities for all $j =$

$0, 1, \dots, m-1$:

$$0 \le \mathbb{E}\Big[\frac{(1-\tau_1-\tau_2)}{\tau_1}\sum_{j=0}^{m-1} D_{sm+j}\cdot\theta^j - \frac{1}{\tau_1}\sum_{j=0}^{m-1} D_{sm+j+1}\cdot\theta^j\Big] + \frac{\tau_2}{\tau_1}\widetilde{D}^s\cdot\sum_{j=0}^{m-1}\theta^j + \frac{1}{2\alpha}\|z_{sm}-x^*\|^2 - \frac{\theta^m}{2\alpha}\big[\|z_{(s+1)m}-x^*\|^2\big] \ .$$

Note that in the above inequality we have assumed all the randomness in the first $s-1$ epochs are fixed and the only source of randomness comes from epoch s. We can rearrange the terms in the above inequality and get

$$\mathbb{E}\Big[\frac{\tau_1+\tau_2-(1-1/\theta)}{\tau_1}\sum_{j=1}^{m} D_{sm+j}\cdot\theta^j\Big] \le \frac{(1-\tau_1-\tau_2)}{\tau_1}\Big(D_{sm}-\theta^m\mathbb{E}\big[D_{(s+1)m}\big]\Big) + \frac{\tau_2}{\tau_1}\widetilde{D}^s\cdot\sum_{j=0}^{m-1}\theta^j + \frac{1}{2\alpha}\|z_{sm}-x^*\|^2 - \frac{\theta^m}{2\alpha}\mathbb{E}\big[\|z_{(s+1)m}-x^*\|^2\big] \ .$$

Using the special choice that $\widetilde{x}^{s+1} = \big(\sum_{j=0}^{m-1}\theta^j\big)^{-1}\cdot\sum_{j=0}^{m-1} y_{sm+j+1}\cdot\theta^j$ and the convexity of $F(\cdot)$, we derive that $\widetilde{D}^{s+1} \le \big(\sum_{j=0}^{m-1}\theta^j\big)^{-1}\cdot\sum_{j=0}^{m-1} D_{sm+j+1}\cdot\theta^j$. Substituting this into the above inequality, we get

$$\frac{\tau_1+\tau_2-(1-1/\theta)}{\tau_1}\theta\mathbb{E}\big[\widetilde{D}^{s+1}\big]\cdot\sum_{j=0}^{m-1}\theta^j \le \frac{(1-\tau_1-\tau_2)}{\tau_1}\Big(D_{sm}-\theta^m\mathbb{E}\big[D_{(s+1)m}\big]\Big) + \frac{\tau_2}{\tau_1}\widetilde{D}^s\cdot\sum_{j=0}^{m-1}\theta^j + \frac{1}{2\alpha}\|z_{sm}-x^*\|^2 - \frac{\theta^m}{2\alpha}\mathbb{E}\big[\|z_{(s+1)m}-x^*\|^2\big] \ . \quad \text{(D.3)}$$

We consider two cases (and four subcases) next.

Case 1. Suppose $L \le \frac{\overline{L}m}{b}$. In this case, we choose

$$\tau_2 = \min\Big\{\frac{\overline{L}}{2Lb}, \frac{1}{2}\Big\} \in \Big[\frac{1}{2m}, \frac{1}{2}\Big] \quad \text{and} \quad L_\diamond = \frac{\overline{L}}{2b\tau_2} \ge L$$

Case 1.1. Suppose $\frac{m\sigma b}{\overline{L}} \le \frac{3}{8}$. In this subcase, we choose

$$\alpha = \frac{\sqrt{b}}{\sqrt{6m\sigma\overline{L}}} \ , \qquad \tau_1 = \frac{1}{3\alpha L_\diamond} = 4m\alpha\sigma\tau_2 = \frac{\sqrt{8\tau_2^2 bm\sigma}}{\sqrt{3\overline{L}}} \in \big[0,\tau_2\big] \subseteq \big[0,\frac{1}{2}\big] \ , \quad \text{and} \quad \theta = 1+\alpha\sigma$$

We have

$$\alpha\sigma = \frac{1}{\sqrt{6m^2}}\frac{\sqrt{b\sigma m}}{\sqrt{\overline{L}}} \le \frac{1}{4m}$$

and therefore the following inequality holds:

$$\tau_2(\theta^{m-1}-1)+(1-1/\theta) = \tau_2((1+\alpha\sigma)^{m-1}-1)+(1-\frac{1}{1+\alpha\sigma}) \le 2\tau_2 m\alpha\sigma+\alpha\sigma \le 4\tau_2 m\alpha\sigma = \tau_1 \ .$$

In other words, we have $\tau_1+\tau_2-(1-1/\theta) \ge \tau_2\theta^{m-1}$ and thus (D.3) implies that

$$\mathbb{E}\Big[\frac{\tau_2}{\tau_1}\widetilde{D}^{s+1}\cdot\sum_{j=0}^{m-1}\theta^j + \frac{1-\tau_1-\tau_2}{\tau_1}D_{(s+1)m} + \frac{1}{2\alpha}\|z_{(s+1)m}-x^*\|^2\Big] \le \theta^{-m}\cdot\Big(\frac{\tau_2}{\tau_1}\widetilde{D}^s\cdot\sum_{j=0}^{m-1}\theta^j + \frac{1-\tau_1-\tau_2}{\tau_1}D_{sm} + \frac{1}{2\alpha}\|z_{sm}-x^*\|^2\Big) \ .$$

If we telescope the above inequality over all epochs $s = 0, 1, \dots, S-1$, we obtain

$$\begin{aligned}\mathbb{E}\big[F(x^{\mathsf{out}})-F(x^*)\big] &\overset{①}{\le} \frac{1}{\tau_2 m+(1-\tau_1-\tau_2)}\mathbb{E}\big[\tau_2\widetilde{D}^S\cdot\sum_{j=0}^{m-1}\theta^j + \big(1-\tau_1-\tau_2\big)D_{Sm}\big] \\ &\overset{②}{\le} \theta^{-Sm}\cdot O\Big(\widetilde{D}^0 + D_0 + \frac{\tau_1}{\alpha\tau_2 m}\|x_0-x^*\|^2\Big) \\ &\overset{③}{\le} \theta^{-Sm}\cdot O\Big(1+\frac{\tau_1}{\alpha\tau_2 m\sigma}\Big)\cdot\big(F(x_0)-F(x^*)\big) \\ &\overset{④}{=} O((1+\alpha\sigma)^{-Sm})\cdot\big(F(x_0)-F(x^*)\big) \ . \end{aligned} \quad \text{(D.4)}$$

Above, inequality ① uses the choice $x^{\mathsf{out}} = \frac{\tau_2 m\widetilde{x}^S+(1-\tau_1-\tau_2)y_{Sm}}{\tau_2 m+(1-\tau_1-\tau_2)}$, the convexity of $F(\cdot)$, and the fact $\sum_{j=0}^{m-1}\theta^j \ge m$; inequality ② uses the fact that $\sum_{j=0}^{m-1}\theta^j \le O(m)$ (because $\alpha\sigma \le \frac{1}{4m}$), and the fact that $\tau_2 \ge \frac{1}{2m}$; inequality ③ uses the strong convexity of $F(\cdot)$ which implies $F(x_0)-F(x^*) \ge \frac{\sigma}{2}\|x_0-x^*\|^2$; and inequality ④ uses our choice of τ_1.

Case 1.2. Suppose $\frac{m\sigma b}{\overline{L}} > \frac{3}{8}$. In this case, we choose

$$\tau_1 = \tau_2 \quad \text{and} \quad \alpha = \frac{1}{3\tau_1 L_\diamond} = \frac{2b}{3\overline{L}} \ge \frac{1}{4\sigma m} \ , \quad \theta = 1+\frac{1}{4m}$$

(Note that we can choose $\theta = 1+\frac{1}{4m}$ because $\frac{1}{4m} \le \alpha\sigma$.)

Under these parameter choices, we can calculate that

$$\frac{(\tau_1+\tau_2-(1-1/\theta))\theta}{\tau_2} = 2-\frac{1-2\tau_2}{4m\tau_2} \ge \frac{3}{2} > \frac{5}{4} \quad \text{and} \quad \theta^m \ge \frac{5}{4}$$

thus (D.3) implies that

$$\mathbb{E}\Big[\frac{\tau_2}{\tau_1}\widetilde{D}^{s+1}\cdot\sum_{j=0}^{m-1}\theta^j + \frac{1-\tau_1-\tau_2}{\tau_1}D_{(s+1)m} + \frac{1}{2\alpha}\|z_{(s+1)m}-x^*\|^2\Big] \le \frac{4}{5}\cdot\Big(\frac{\tau_2}{\tau_1}\widetilde{D}^s\cdot\sum_{j=0}^{m-1}\theta^j + \frac{1-\tau_1-\tau_2}{\tau_1}D_{sm} + \frac{1}{2\alpha}\|z_{sm}-x^*\|^2\Big) \ .$$

If we telescope the above inequality over all epochs $s = 0, 1, \ldots, S-1$, we obtain

$$\begin{aligned}\mathbb{E}\big[F(x^{\mathsf{out}}) - F(x^*)\big] &\overset{①}{\le} \frac{1}{\tau_2 m + (1-\tau_1-\tau_2)}\mathbb{E}\big[\tau_2 \widetilde{D}^S \cdot \sum_{j=0}^{m-1}\theta^j + \big(1-\tau_1-\tau_2\big)D_{Sm}\big] \\ &\overset{②}{\le} \big(\frac{5}{4}\big)^{-S}\cdot O\Big(\widetilde{D}^0 + D_0 + \frac{\tau_1}{\alpha\tau_2 m}\|x_0 - x^*\|^2\Big) \\ &\overset{③}{\le} \big(\frac{5}{4}\big)^{-S}\cdot O\Big(1 + \frac{\tau_1}{\alpha\tau_2 m\sigma}\Big)\cdot\big(F(x_0) - F(x^*)\big) \\ &\overset{④}{=} O((5/4)^{-S})\cdot\big(F(x_0) - F(x^*)\big)\ . \end{aligned} \tag{D.5}$$

Above, inequality ① uses the choice $x^{\mathsf{out}} = \frac{\tau_2 m\widetilde{x}^S + (1-\tau_1-\tau_2)y_{Sm}}{\tau_2 m + (1-\tau_1-\tau_2)}$, the convexity of $F(\cdot)$, and the fact $\sum_{j=0}^{m-1}\theta^j \ge m$; inequality ② uses the fact that $\sum_{j=0}^{m-1}\theta^j \le O(m)$, and the fact that $\tau_2 \ge \frac{1}{2m}$; inequality ③ uses the strong convexity of $F(\cdot)$ which implies $F(x_0) - F(x^*) \ge \frac{\sigma}{2}\|x_0 - x^*\|^2$; and inequality ④ uses our choice of τ_1 and α.

Case 2. Suppose $L > \frac{\overline{L}m}{b}$. In this case, we choose

$$L_\diamond = L \quad \text{and} \quad \tau_2 = \frac{\overline{L}}{2L_\diamond b} = \frac{\overline{L}}{2Lb} \in \big[0, \frac{1}{2m}\big]$$

Case 2.1. Suppose $\frac{m^2\sigma}{L} \le \frac{3}{8}$. In this subcase, we choose

$$\alpha = \frac{1}{\sqrt{6\sigma L}}\ , \qquad \tau_1 = \frac{1}{3\alpha L} = 2\alpha\sigma = \frac{\sqrt{2\sigma}}{\sqrt{3L}} \in \big[0, \frac{1}{2m}\big]\ , \quad \theta = 1 + \alpha\sigma$$

We have $\alpha\sigma \le \frac{1}{4m}$ and therefore the following inequality holds:

$$\tau_2(\theta^{m-1} - 1) + (1 - 1/\theta) = \tau_2((1+\alpha\sigma)^{m-1} - 1) + (1 - \frac{1}{1+\alpha\sigma}) \le 2\tau_2 m\alpha\sigma + \alpha\sigma \le 2\alpha\sigma = \tau_1\ .$$

In other words, we have $\tau_1 + \tau_2 - (1 - 1/\theta) \ge \tau_2\theta^{m-1}$ and thus (D.3) implies that

$$\begin{aligned}&\mathbb{E}\Big[\frac{\tau_2}{\tau_1}\widetilde{D}^{s+1}\cdot\sum_{j=0}^{m-1}\theta^j + \frac{1-\tau_1-\tau_2}{\tau_1}D_{(s+1)m} + \frac{1}{2\alpha}\|z_{(s+1)m} - x^*\|^2\Big] \\ &\le \theta^{-m}\cdot\Big(\frac{\tau_2}{\tau_1}\widetilde{D}^s\cdot\sum_{j=0}^{m-1}\theta^j + \frac{1-\tau_1-\tau_2}{\tau_1}D_{sm} + \frac{1}{2\alpha}\|z_{sm} - x^*\|^2\Big)\ .\end{aligned}$$

If we telescope the above inequality over all epochs $s = 0, 1, \ldots, S-1$, we obtain

$$\begin{aligned}\mathbb{E}\big[F(x^{\mathsf{out}}) - F(x^*)\big] &\overset{①}{\le} \frac{1}{\tau_2 m + (1-\tau_1-\tau_2)}\mathbb{E}\big[\tau_2 \widetilde{D}^S \cdot \sum_{j=0}^{m-1}\theta^j + \big(1-\tau_1-\tau_2\big)D_{Sm}\big] \\ &\overset{②}{\le} \theta^{-Sm}\cdot O\Big(\widetilde{D}^0 + D_0 + \frac{\tau_1}{\alpha}\|x_0 - x^*\|^2\Big) \\ &\overset{③}{\le} \theta^{-Sm}\cdot O\Big(1 + \frac{\tau_1}{\alpha\sigma}\Big)\cdot\big(F(x_0) - F(x^*)\big) \\ &\overset{④}{=} O((1+\alpha\sigma)^{-Sm})\cdot\big(F(x_0) - F(x^*)\big)\ . \end{aligned} \tag{D.6}$$

Above, inequality ① uses the choice $x^{\mathsf{out}} = \frac{\tau_2 m\widetilde{x}^S + (1-\tau_1-\tau_2)y_{Sm}}{\tau_2 m + (1-\tau_1-\tau_2)}$, the convexity of $F(\cdot)$, and the fact $\sum_{j=0}^{m-1}\theta^j \ge m$; inequality ② uses the fact that $\sum_{j=0}^{m-1}\theta^j \le O(m)$ (because $\alpha\sigma \le \frac{1}{4m}$), and the fact that $\tau_2 m + (1-\tau_1-\tau_2) \ge 1 - \tau_1 + (m-1)\tau_2 \ge 1/2$; inequality ③ uses the strong convexity of $F(\cdot)$ which implies $F(x_0) - F(x^*) \ge \frac{\sigma}{2}\|x_0 - x^*\|^2$; and inequality ④ uses our choice of τ_1.

Case 2.2. Suppose $\frac{m^2\sigma}{L} > \frac{3}{8}$. In this case, we choose

$$\tau_1 = \frac{1}{2m} \quad \text{and} \quad \alpha = \frac{1}{3\tau_1 L} = \frac{2m}{3L} > \frac{1}{4\sigma m}\ , \quad \theta = 1 + \frac{1}{4m}$$

(Note that we can choose $\theta = 1 + \frac{1}{4m}$ because $\frac{1}{4m} \le \alpha\sigma$.)

Under these parameter choices, we can calculate that

$$\frac{(\tau_1+\tau_2-(1-1/\theta))\theta}{\tau_2} = \frac{\tau_1+\tau_2}{\tau_2} - \frac{1-2\tau_2}{4m\tau_2} \ge 1 + \frac{\tau_1 - 1/4m}{\tau_2} \ge \frac{3}{2} > \frac{5}{4} \quad \text{and} \quad \theta^m \ge \frac{5}{4}$$

thus (D.3) implies that

$$\begin{aligned}&\mathbb{E}\Big[\frac{\tau_2}{\tau_1}\widetilde{D}^{s+1}\cdot\sum_{j=0}^{m-1}\theta^j + \frac{1-\tau_1-\tau_2}{\tau_1}D_{(s+1)m} + \frac{1}{2\alpha}\|z_{(s+1)m} - x^*\|^2\Big] \\ &\le \frac{4}{5}\cdot\Big(\frac{\tau_2}{\tau_1}\widetilde{D}^s\cdot\sum_{j=0}^{m-1}\theta^j + \frac{1-\tau_1-\tau_2}{\tau_1}D_{sm} + \frac{1}{2\alpha}\|z_{sm} - x^*\|^2\Big)\ .\end{aligned}$$

If we telescope the above inequality over all epochs $s = 0, 1, \ldots, S-1$, we obtain

$$\begin{aligned}\mathbb{E}\big[F(x^{\mathsf{out}}) - F(x^*)\big] &\overset{①}{\le} \frac{1}{\tau_2 m + (1-\tau_1-\tau_2)}\mathbb{E}\big[\tau_2 \widetilde{D}^S \cdot \sum_{j=0}^{m-1}\theta^j + \big(1-\tau_1-\tau_2\big)D_{Sm}\big] \\ &\overset{②}{\le} \big(\frac{5}{4}\big)^{-S}\cdot O\Big(\widetilde{D}^0 + D_0 + \frac{\tau_1}{\alpha}\|x_0 - x^*\|^2\Big) \\ &\overset{③}{\le} \big(\frac{5}{4}\big)^{-S}\cdot O\Big(1 + \frac{\tau_1}{\alpha\sigma}\Big)\cdot\big(F(x_0) - F(x^*)\big) \\ &\overset{④}{=} O((5/4)^{-S})\cdot\big(F(x_0) - F(x^*)\big)\ . \end{aligned} \tag{D.7}$$

Above, inequality ① uses the choice $x^{\mathsf{out}} = \frac{\tau_2 m\widetilde{x}^S + (1-\tau_1-\tau_2)y_{Sm}}{\tau_2 m + (1-\tau_1-\tau_2)}$, the convexity of $F(\cdot)$, and the fact $\sum_{j=0}^{m-1}\theta^j \ge m$; inequality ② uses the fact that $\sum_{j=0}^{m-1}\theta^j \le O(m)$, and that $\tau_2 m + (1-\tau_1-\tau_2) \ge 1 - \tau_1 + (m-1)\tau_2 \ge 1/2$; inequality ③ uses the strong convexity of $F(\cdot)$ which implies $F(x_0) - F(x^*) \ge \frac{\sigma}{2}\|x_0 - x^*\|^2$; and inequality ④ uses our choice of τ_1 and α. ■

Appendix E. Appendix for Section 6

In this section, we first include the complete pseudo-codes for `Katyusha2` and `Katyusha2`$^{\mathsf{ns}}$. Then, we provide a one-iteration analysis for both algorithms, in the same spirit as Section 2.1.

The final proofs of Theorem 6.1 and Theorem 6.2 are direct corollaries of such one-iteration analysis, where the details we have already given in Section 2.2 and in Section C.1 respectively.

E.1 Pseudo-Codes

Algorithm 4 `Katyusha2`$(x_0, S, \sigma, (L_1, \dots, L_n))$

$m \leftarrow n$; $\overline{L} = (L_1 + \cdots + L_n)/n$;
$\tau_2 \leftarrow \frac{1}{2}$, $\tau_1 \leftarrow \min\{\sqrt{m\sigma/9\overline{L}}, \frac{1}{2}\}$, $\alpha \leftarrow \frac{1}{9\tau_1\overline{L}}$;
$y_0 = z_0 = \widetilde{x}^0 \leftarrow x_0$;
for $s \leftarrow 0$ **to** $S-1$ **do**
  $\mu^s \leftarrow \nabla f(\widetilde{x}^s)$;
  for $j \leftarrow 0$ **to** $m-1$ **do**
    $k \leftarrow (sm) + j$;
    $x_{k+1} \leftarrow \tau_1 z_k + \tau_2 \widetilde{x}^s + (1-\tau_1-\tau_2) y_k$;
    Pick i randomly from $\{1, 2, \dots, n\}$, each with probability $L_i/n\overline{L}$;
    $\widetilde{\nabla}_{k+1} \leftarrow \mu^s + \nabla f_i(x_{k+1}) - \nabla f_i(\widetilde{x}^s)$;
    $z_{k+1} = \arg\min_z \{\frac{1}{\alpha} V_{z_k}(z) + \langle \widetilde{\nabla}_{k+1}, z\rangle + \psi(z)\}$;
      ◇ *$V_x(y)$ is the Bregman divergence function, see Section 6*
    $y_{k+1} \leftarrow \arg\min_y \{\frac{9\overline{L}}{2}\|y - x_{k+1}\|^2 + \langle \widetilde{\nabla}_{k+1}, y\rangle + \psi(y)\}$;
  end for
  $\widetilde{x}^{s+1} \leftarrow \big(\sum_{j=0}^{m-1}(1+\alpha\sigma)^j\big)^{-1} \cdot \big(\sum_{j=0}^{m-1}(1+\alpha\sigma)^j \cdot y_{sm+j+1}\big)$;
end for
return $\widetilde{x}^S$.

Algorithm 5 `Katyusha2`$^{\mathsf{ns}}(x_0, S, \sigma, (L_1, \dots, L_n))$

$m \leftarrow n$; $\overline{L} = (L_1 + \cdots + L_n)/n$;
$\tau_2 \leftarrow \frac{1}{2}$;
$y_0 = z_0 = \widetilde{x}^0 \leftarrow x_0$;
for $s \leftarrow 0$ **to** $S-1$ **do**
  $\tau_{1,s} \leftarrow \frac{2}{s+4}$, $\alpha_s \leftarrow \frac{1}{9\tau_{1,s}\overline{L}}$
  $\mu^s \leftarrow \nabla f(\widetilde{x}^s)$;
  for $j \leftarrow 0$ **to** $m-1$ **do**
    $k \leftarrow (sm) + j$;
    $x_{k+1} \leftarrow \tau_{1,s} z_k + \tau_2 \widetilde{x}^s + (1-\tau_{1,s}-\tau_2) y_k$;
    Pick i randomly from $\{1, 2, \dots, n\}$, each with probability $L_i/n\overline{L}$;
    $\widetilde{\nabla}_{k+1} \leftarrow \mu^s + \nabla f_i(x_{k+1}) - \nabla f_i(\widetilde{x}^s)$;
    $z_{k+1} = \arg\min_z \{\frac{1}{\alpha_s} V_{z_k}(z) + \langle \widetilde{\nabla}_{k+1}, z\rangle + \psi(z)\}$;
      ◇ *$V_x(y)$ is the Bregman divergence function, see Section 6*
    $y_{k+1} \leftarrow \arg\min_y \{\frac{9\overline{L}}{2}\|y - x_{k+1}\|^2 + \langle \widetilde{\nabla}_{k+1}, y\rangle + \psi(y)\}$;
  end for
  $\widetilde{x}^{s+1} \leftarrow \frac{1}{m}\sum_{j=1}^{m} y_{sm+j}$;
end for
return $\widetilde{x}^S$.

E.2 One-Iteration Analysis

Similar as Section 2.1, we first analyze the behavior of `Katyusha2` in a single iteration (i.e., for a fixed k). We view y_k, z_k and x_{k+1} as fixed in this section so the only randomness comes from the choice of i in iteration k. We abbreviate in this subsection by $\widetilde{x} = \widetilde{x}^s$ where s is the epoch that iteration k belongs to, and denote by $\sigma_{k+1}^2 \stackrel{\text{def}}{=} \|\nabla f(x_{k+1}) - \widetilde{\nabla}_{k+1}\|_*^2$.

Our first lemma is analogous to Lemma E.1 except the change of the parameter and the norm.

Lemma E.1 (proximal gradient descent) *If*

$$y_{k+1} = \arg\min_y \Big\{\frac{9\overline{L}}{2}\|y - x_{k+1}\|^2 + \langle \widetilde{\nabla}_{k+1}, y - x_{k+1}\rangle + \psi(y) - \psi(x_{k+1})\Big\}\ , \qquad \textit{and}$$

$$\mathsf{Prog}(x_{k+1}) \stackrel{\text{def}}{=} -\min_y \Big\{\frac{9\overline{L}}{2}\|y - x_{k+1}\|^2 + \langle \widetilde{\nabla}_{k+1}, y - x_{k+1}\rangle + \psi(y) - \psi(x_{k+1})\Big\} \geq 0\ ,$$

we have (where the expectation is only over the randomness of $\widetilde{\nabla}_{k+1}$)

$$F(x_{k+1}) - \mathbb{E}\big[F(y_{k+1})\big] \geq \mathbb{E}\big[\mathsf{Prog}(x_{k+1})\big] - \frac{1}{16\overline{L}}\mathbb{E}\big[\sigma_{k+1}^2\big]\ .$$

Proof

$$\begin{aligned}
\mathsf{Prog}(x_{k+1}) &= -\min_y \Big\{\frac{9\overline{L}}{2}\|y - x_{k+1}\|^2 + \langle \widetilde{\nabla}_{k+1}, y - x_{k+1}\rangle + \psi(y) - \psi(x_{k+1})\Big\} \\
&\stackrel{①}{=} -\Big(\frac{9\overline{L}}{2}\|y_{k+1} - x_{k+1}\|^2 + \langle \widetilde{\nabla}_{k+1}, y_{k+1} - x_{k+1}\rangle + \psi(y_{k+1}) - \psi(x_{k+1})\Big) \\
&= -\Big(\frac{\overline{L}}{2}\|y_{k+1} - x_{k+1}\|^2 + \langle \nabla f(x_{k+1}), y_{k+1} - x_{k+1}\rangle + \psi(y_{k+1}) - \psi(x_{k+1})\Big) \\
&\quad + \Big(\langle \nabla f(x_{k+1}) - \widetilde{\nabla}_{k+1}, y_{k+1} - x_{k+1}\rangle - 4\overline{L}\|y_{k+1} - x_{k+1}\|^2\Big) \\
&\stackrel{②}{\leq} -\Big(f(y_{k+1}) - f(x_{k+1}) + \psi(y_{k+1}) - \psi(x_{k+1})\Big) + \frac{1}{16\overline{L}}\|\nabla f(x_{k+1}) - \widetilde{\nabla}_{k+1}\|_*^2\ .
\end{aligned}$$

Above, ① is by the definition of y_{k+1}, and ② uses the smoothness of function $f(\cdot)$, as well as Young's inequality $\langle a, b\rangle - \frac{1}{2}\|b\|^2 \leq \frac{1}{2}\|a\|_*^2$. Taking expectation on both sides we arrive at the desired result. ■

The next lemma is analogous to Lemma 2.4 but with slightly different proof.

Lemma E.2 (variance upper bound)

$$\mathbb{E}\big[\|\widetilde{\nabla}_{k+1} - \nabla f(x_{k+1})\|_*^2\big] \leq 8\overline{L} \cdot \big(f(\widetilde{x}) - f(x_{k+1}) - \langle \nabla f(x_{k+1}), \widetilde{x} - x_{k+1}\rangle\big)\ .$$

Proof Each $f_i(x)$, being convex and L_i-smooth, implies the following inequality which is classical in convex optimization and can be found for instance in Theorem 2.1.5 of the textbook of Nesterov (2004).

$$\|\nabla f_i(x_{k+1}) - \nabla f_i(\widetilde{x})\|_*^2 \leq 2L_i \cdot \big(f_i(\widetilde{x}) - f_i(x_{k+1}) - \langle \nabla f_i(x_{k+1}), \widetilde{x} - x_{k+1}\rangle\big) \qquad \text{(E.1)}$$

Therefore, taking expectation over the random choice of i, we have

$$\begin{aligned}
&\mathbb{E}\big[\|\widetilde{\nabla}_{k+1}-\nabla f(x_{k+1})\|_*^2\big]\\
&=\mathbb{E}\big[\big\|\frac{1}{np_i}\big(\nabla f_i(x_{k+1})-\nabla f_i(\widetilde{x})\big)-\big(\nabla f(x_{k+1})-\nabla f(\widetilde{x})\big)\big\|_*^2\big]\\
&\overset{①}{\le} 2\mathbb{E}\big[\frac{1}{n^2p_i^2}\big\|\nabla f_i(x_{k+1})-\nabla f_i(\widetilde{x})\big\|_*^2\big]+2\big\|\nabla f(x_{k+1})-\nabla f(\widetilde{x})\big\|_*^2\\
&\overset{②}{\le} 4\cdot\mathbb{E}\big[\frac{L_i}{n^2p_i^2}\big(f_i(\widetilde{x})-f_i(x_{k+1})-\langle\nabla f_i(x_{k+1}),\widetilde{x}-x_{k+1}\rangle\big)\big]+2\big\|\nabla f(x_{k+1})-\nabla f(\widetilde{x})\big\|_*^2\\
&\overset{③}{=} 4\overline{L}\cdot\big(f(\widetilde{x})-f(x_{k+1})-\langle\nabla f(x_{k+1}),\widetilde{x}-x_{k+1}\rangle\big)+2\big\|\nabla f(x_{k+1})-\nabla f(\widetilde{x})\big\|_*^2\\
&\le 8\overline{L}\cdot\big(f(\widetilde{x})-f(x_{k+1})-\langle\nabla f(x_{k+1}),\widetilde{x}-x_{k+1}\rangle\big)\ .
\end{aligned}$$

Above, inequality ① is because $\|a+b\|_*^2\le(\|a\|_*+\|b\|_*)^2\le 2\|a\|_*^2+2\|b\|_*^2$; inequality ② follows from (E.1); equality ③ follows from the probability distribution that we select i with probability $p_i=L_i/(n\overline{L})$; inequality ④ uses (E.1) again but replacing $f_i(\cdot)$ with $f(\cdot)$. ■

The next lemma is classical for mirror descent with respect to a general Bregman divergence.

Lemma E.3 (proximal mirror descent) *Suppose $\psi(\cdot)$ is σ-SC with respect to $V_x(y)$. Then, fixing $\widetilde{\nabla}_{k+1}$ and letting*

$$z_{k+1}=\arg\min_z\big\{V_{z_k}(z)+\alpha\langle\widetilde{\nabla}_{k+1},z-z_k\rangle+\alpha\psi(z)-\alpha\psi(z_k)\big\}\ ,$$

it satisfies for all $u\in\mathbb{R}^d$,

$$\alpha\langle\widetilde{\nabla}_{k+1},z_{k+1}-u\rangle+\alpha\psi(z_{k+1})-\alpha\psi(u)\le-\frac{1}{2}\|z_k-z_{k+1}\|^2+V_{z_k}(u)-(1+\alpha\sigma)V_{z_{k+1}}(u)\ .$$

Proof By the minimality definition of z_{k+1}, we have that

$$\nabla V_{z_k}(z_{k+1})+\alpha\widetilde{\nabla}_{k+1}+\alpha g=0$$

where g is *some* subgradient of $\psi(z)$ at point $z=z_{k+1}$. This implies that for every u it satisfies

$$0=\big\langle\nabla V_{z_k}(z_{k+1})+\alpha\widetilde{\nabla}_{k+1}+\alpha g,z_{k+1}-u\big\rangle\ .$$

At this point, using the equality $\langle\nabla V_{z_k}(z_{k+1}),z_{k+1}-u\rangle=V_{z_k}(z_{k+1})-V_{z_k}(u)+V_{z_{k+1}}(u)$ (known as the "three-point equality of Bregman divergence", see Rakhlin (2009)), as well as the inequality $\langle g,z_{k+1}-u\rangle\ge\psi(z_{k+1})-\psi(u)+\sigma V_{z_{k+1}}(u)$ which comes from the strong convexity of $\psi(\cdot)$, we can write

$$\begin{aligned}
&\alpha\langle\widetilde{\nabla}_{k+1},z_{k+1}-u\rangle+\alpha\psi(z_{k+1})-\alpha\psi(u)\\
&=-\langle z_{k+1}-z_k,z_{k+1}-u\rangle-\langle\alpha g,z_{k+1}-u\rangle+\alpha\psi(z_{k+1})-\alpha\psi(u)\\
&\le-V_{z_k}(z_{k+1})+V_{z_k}(u)-(1+\alpha\sigma)V_{z_{k+1}}(u)\ .
\end{aligned}$$

Finally, using $V_{z_k}(z_{k+1})\ge\frac{1}{2}\|z_k-z_{k+1}\|^2$ which comes from the strong convexity of $w(x)$ with respect to $\|\cdot\|$, we complete the proof. ■

The following lemma combines Lemma E.1, Lemma E.2 and Lemma E.3 all together, using the special choice of x_{k+1} which is a convex combination of y_k, z_k and $\widetilde{x}$:

Lemma E.4 (coupling step 1) *If $x_{k+1}=\tau_1z_k+\tau_2\widetilde{x}+(1-\tau_1-\tau_2)y_k$, where $\tau_1\le\frac{1}{9\alpha\overline{L}}$ and $\tau_2=\frac{1}{2}$,*

$$\begin{aligned}
&\alpha\langle\nabla f(x_{k+1}),z_k-u\rangle-\alpha\psi(u)\\
&\le\frac{\alpha}{\tau_1}\Big(F(x_{k+1})-\mathbb{E}\big[F(y_{k+1})\big]+\tau_2F(\widetilde{x})-\tau_2\mathbb{E}\big[F(x_{k+1})\big]-\tau_2\langle\nabla f(x_{k+1}),\widetilde{x}-x_{k+1}\rangle\Big)\\
&\quad+V_{z_k}(u)-(1+\alpha\sigma)\mathbb{E}\big[V_{z_{k+1}}(u)\big]+\frac{\alpha(1-\tau_1-\tau_2)}{\tau_1}\psi(y_k)-\frac{\alpha}{\tau_1}\psi(x_{k+1})\ .
\end{aligned}$$

Proof We first apply Lemma E.3 and get

$$\begin{aligned}
&\alpha\langle\widetilde{\nabla}_{k+1},z_k-u\rangle+\alpha\psi(z_{k+1})-\alpha\psi(u)\\
&=\alpha\langle\widetilde{\nabla}_{k+1},z_k-z_{k+1}\rangle+\alpha\langle\widetilde{\nabla}_{k+1},z_{k+1}-u\rangle+\alpha\psi(z_{k+1})-\alpha\psi(u)\\
&\le\alpha\langle\widetilde{\nabla}_{k+1},z_k-z_{k+1}\rangle-\frac{1}{2}\|z_k-z_{k+1}\|^2+V_{z_k}(u)-(1+\alpha\sigma)V_{z_{k+1}}(u)\ . \qquad \text{(E.2)}
\end{aligned}$$

By defining $v\overset{\text{def}}{=}\tau_1z_{k+1}+\tau_2\widetilde{x}+(1-\tau_1-\tau_2)y_k$, we have $x_{k+1}-v=\tau_1(z_k-z_{k+1})$ and therefore

$$\begin{aligned}
&\mathbb{E}\Big[\alpha\langle\widetilde{\nabla}_{k+1},z_k-z_{k+1}\rangle-\frac{1}{2}\|z_k-z_{k+1}\|^2\Big]=\mathbb{E}\Big[\frac{\alpha}{\tau_1}\langle\widetilde{\nabla}_{k+1},x_{k+1}-v\rangle-\frac{1}{2\tau_1^2}\|x_{k+1}-v\|^2\Big]\\
&=\mathbb{E}\Big[\frac{\alpha}{\tau_1}\Big(\langle\widetilde{\nabla}_{k+1},x_{k+1}-v\rangle-\frac{1}{2\alpha\tau_1}\|x_{k+1}-v\|^2-\psi(v)+\psi(x_{k+1})\Big)+\frac{\alpha}{\tau_1}\Big(\psi(v)-\psi(x_{k+1})\Big)\Big]\\
&\overset{①}{\le}\mathbb{E}\Big[\frac{\alpha}{\tau_1}\Big(\langle\widetilde{\nabla}_{k+1},x_{k+1}-v\rangle-\frac{9\overline{L}}{2}\|x_{k+1}-v\|^2-\psi(v)+\psi(x_{k+1})\Big)+\frac{\alpha}{\tau_1}\Big(\psi(v)-\psi(x_{k+1})\Big)\Big]\\
&\overset{②}{\le}\mathbb{E}\Big[\frac{\alpha}{\tau_1}\Big(F(x_{k+1})-F(y_{k+1})+\frac{1}{16\overline{L}}\sigma_{k+1}^2\Big)+\frac{\alpha}{\tau_1}\Big(\psi(v)-\psi(x_{k+1})\Big)\Big]\\
&\overset{③}{\le}\mathbb{E}\Big[\frac{\alpha}{\tau_1}\Big(F(x_{k+1})-F(y_{k+1})+\frac{1}{2}\big(f(\widetilde{x})-f(x_{k+1})-\langle\nabla f(x_{k+1}),\widetilde{x}-x_{k+1}\rangle\big)\Big)\\
&\quad+\frac{\alpha}{\tau_1}\Big(\tau_1\psi(z_{k+1})+\tau_2\psi(\widetilde{x})+(1-\tau_1-\tau_2)\psi(y_k)-\psi(x_{k+1})\Big)\Big]\ . \qquad \text{(E.3)}
\end{aligned}$$

Above, ① uses our choice $\tau_1\le\frac{1}{9\alpha\overline{L}}$, ② uses Lemma E.1, ③ uses Lemma E.2 together with the convexity of $\psi(\cdot)$ and the definition of v. Finally, noticing that $\mathbb{E}[\langle\widetilde{\nabla}_{k+1},z_k-u\rangle]=\langle\nabla f(x_{k+1}),z_k-u\rangle$ and $\tau_2=\frac{1}{2}$, we obtain the desired inequality by combining (E.2) and (E.3). ■

The next lemma is completely analogous to Lemma 2.7 except that we use Lemma E.4 rather than Lemma 2.6. We ignore the proof since it is a simple copy-and-paste.

Lemma E.5 (coupling step 2) *Under the same choices of τ_1,τ_2 as in Lemma E.4, we have*

$$\begin{aligned}
0\le\frac{\alpha(1-\tau_1-\tau_2)}{\tau_1}(F(y_k)-F(x^*))-\frac{\alpha}{\tau_1}\big(\mathbb{E}\big[F(y_{k+1})\big]-F(x^*)\big)+\frac{\alpha\tau_2}{\tau_1}\big(F(\widetilde{x})-\tau_2F(x^*)\big)\\
+V_{z_k}(x^*)-(1+\alpha\sigma)\mathbb{E}\big[V_{z_{k+1}}(x^*)\big]\ .
\end{aligned}$$

References

Naman Agarwal, Zeyuan Allen-Zhu, Brian Bullins, Elad Hazan, and Tengyu Ma. Finding Approximate Local Minima for Nonconvex Optimization in Linear Time. In *STOC*, 2017. Full version available at `http://arxiv.org/abs/1611.01146`.

Zeyuan Allen-Zhu. Natasha 2: Faster Non-Convex Optimization Than SGD. *ArXiv e-prints*, abs/1708.08694, August 2017. Full version available at `http://arxiv.org/abs/1708.08694`.

Zeyuan Allen-Zhu. Katyusha X: Practical Momentum Method for Stochastic Sum-of-Nonconvex Optimization. In *ICML*, 2018a. Full version available at `http://arxiv.org/abs/1802.03866`.

Zeyuan Allen-Zhu. How To Make the Gradients Small Stochastically: Even Faster Convex and Nonconvex SGD. *ArXiv e-prints*, abs/1801.02982, January 2018b. Full version available at `http://arxiv.org/abs/1801.02982`.

Zeyuan Allen-Zhu and Elad Hazan. Variance Reduction for Faster Non-Convex Optimization. In *ICML*, 2016a. Full version available at `http://arxiv.org/abs/1603.05643`.

Zeyuan Allen-Zhu and Elad Hazan. Optimal Black-Box Reductions Between Optimization Objectives. In *NIPS*, 2016b.

Zeyuan Allen-Zhu and Yuanzhi Li. LazySVD: Even Faster SVD Decomposition Yet Without Agonizing Pain. In *NIPS*, 2016. Full version available at `http://arxiv.org/abs/1607.03463`.

Zeyuan Allen-Zhu and Yuanzhi Li. Faster Principal Component Regression and Stable Matrix Chebyshev Approximation. In *Proceedings of the 34th International Conference on Machine Learning*, ICML '17, 2017a.

Zeyuan Allen-Zhu and Yuanzhi Li. Doubly Accelerated Methods for Faster CCA and Generalized Eigendecomposition. In *Proceedings of the 34th International Conference on Machine Learning*, ICML '17, 2017b.

Zeyuan Allen-Zhu and Lorenzo Orecchia. Nearly-Linear Time Positive LP Solver with Faster Convergence Rate. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing*, STOC '15, 2015a.

Zeyuan Allen-Zhu and Lorenzo Orecchia. Using Optimization to Break the Epsilon Barrier: A Faster and Simpler Width-Independent Algorithm for Solving Positive Linear Programs in Parallel. In *SODA*, 2015b.

Zeyuan Allen-Zhu and Lorenzo Orecchia. Linear Coupling: An Ultimate Unification of Gradient and Mirror Descent. In *Proceedings of the 8th Innovations in Theoretical Computer Science*, ITCS '17, 2017. Full version available at `http://arxiv.org/abs/1407.1537`.

Zeyuan Allen-Zhu and Yang Yuan. Improved SVRG for Non-Strongly-Convex or Sum-of-Non-Convex Objectives. In *ICML*, 2016.

Zeyuan Allen-Zhu, Yin Tat Lee, and Lorenzo Orecchia. Using Optimization to Obtain a Width-Independent, Parallel, Simpler, and Faster Positive SDP Solver. In *SODA*, 2016a.

Zeyuan Allen-Zhu, Zhenyu Liao, and Yang Yuan. Optimization Algorithms for Faster Computational Geometry. In *ICALP*, 2016b.

Zeyuan Allen-Zhu, Peter Richtárik, Zheng Qu, and Yang Yuan. Even faster accelerated coordinate descent using non-uniform sampling. In *ICML*, 2016c.

Zeyuan Allen-Zhu, Yuanzhi Li, Rafael Oliveira, and Avi Wigderson. Much Faster Algorithms for Matrix Scaling. In *FOCS*, 2017. Full version available at `http://arxiv.org/abs/1704.02315`.

Léon Bottou. Stochastic gradient descent. `http://leon.bottou.org/projects/sgd`.

Sébastien Bubeck, Yin Tat Lee, and Mohit Singh. A geometric alternative to Nesterov's accelerated gradient descent. *ArXiv e-prints*, abs/1506.08187, June 2015.

Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. Accelerated Methods for Non-Convex Optimization. *ArXiv e-prints*, abs/1611.00756, November 2016.

Cong Dang and Guanghui Lan. Randomized First-Order Methods for Saddle Point Optimization. *ArXiv e-prints*, abs/1409.8625, sep 2014.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives. In *NIPS*, 2014.

Rong-En Fan and Chih-Jen Lin. LIBSVM Data: Classification, Regression and Multi-label. URL `http://www.csie.ntu.edu.tw/cjlin/libsvmtools/datasets`. Accessed: 2015-06.

Roy Frostig, Rong Ge, Sham M. Kakade, and Aaron Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *ICML*, 2015.

Roy Frostig, Cameron Musco, Christopher Musco, and Aaron Sidford. Principal Component Projection Without Principal Component Analysis. In *ICML*, 2016.

Dan Garber, Elad Hazan, Chi Jin, Sham M. Kakade, Cameron Musco, Praneeth Netrapalli, and Aaron Sidford. Robust shift-and-invert preconditioning: Faster and more sample efficient algorithms for eigenvector computation. In *ICML*, 2016.

Elad Hazan and Satyen Kale. Beyond the regret minimization barrier: Optimal algorithms for stochastic strongly-convex optimization. *The Journal of Machine Learning Research*, 15(1):2489–2512, 2014.

Chonghai Hu, Weike Pan, and James T Kwok. Accelerated gradient methods for stochastic optimization and online learning. In *Advances in Neural Information Processing Systems*, pages 781–789, 2009.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, NIPS 2013, pages 315–323, 2013.

Anatoli Juditsky. Convex optimization II: Algorithms. Lecture notes, November 2013.

Jakub Konečnỳ, Jie Liu, Peter Richtárik, and Martin Takáč. Mini-batch semi-stochastic gradient descent in the proximal setting. *IEEE Journal of Selected Topics in Signal Processing*, 10(2):242–255, 2016.

Guanghui Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1-2):365–397, January 2011. ISSN 0025-5610.

Guanghui Lan and Yi Zhou. An optimal randomized incremental gradient method. *ArXiv e-prints*, abs/1507.02000, October 2015.

Lihua Lei, Cheng Ju, Jianbo Chen, and Michael I Jordan. Nonconvex Finite-Sum Optimization Via SCSG Methods. In *NIPS*, 2017.

Hongzhou Lin. private communication, 2016.

Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A Universal Catalyst for First-Order Optimization. In *NIPS*, 2015.

Qihang Lin, Zhaosong Lu, and Lin Xiao. An Accelerated Proximal Coordinate Gradient Method and its Application to Regularized Empirical Risk Minimization. In *NIPS*, pages 3059–3067, 2014. URL `http://arxiv.org/abs/1407.1296`.

Zhaosong Lu and Lin Xiao. On the complexity analysis of randomized block-coordinate descent methods. *Mathematical Programming*, pages 1–28, 2013.

Mehrdad Mahdavi, Lijun Zhang, and Rong Jin. Mixed optimization for smooth functions. In *Advances in Neural Information Processing Systems*, pages 674–682, 2013.

Julien Mairal. Incremental Majorization-Minimization Optimization with Application to Large-Scale Machine Learning. *SIAM Journal on Optimization*, 25(2):829–855, April 2015. ISSN 1052-6234. Preliminary version appeared in ICML 2013.

Tomoya Murata and Taiji Suzuki. Doubly accelerated stochastic variance reduced dual averaging method for regularized empirical risk minimization. In *NIPS*, 2017.

Yurii Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. In *Doklady AN SSSR (translated as Soviet Mathematics Doklady)*, volume 269, pages 543–547, 1983.

Yurii Nesterov. *Introductory Lectures on Convex Programming Volume: A Basic course*, volume I. Kluwer Academic Publishers, 2004. ISBN 1402075537.

Yurii Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, December 2005. ISSN 0025-5610. doi: 10.1007/s10107-004-0552-5.

Yurii Nesterov. Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems. *SIAM Journal on Optimization*, 22(2):341–362, jan 2012. ISSN 1052-6234. doi: 10.1137/100802001.

Atsushi Nitanda. Stochastic proximal gradient descent with acceleration techniques. In *Advances in Neural Information Processing Systems*, pages 1574–1582, 2014.

Alexander Rakhlin. Lecture notes on online learning. *Draft*, 2009. Available at `http://www-stat.wharton.upenn.edu/~rakhlin/courses/stat991/papers/lecture_notes.pdf`.

Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In *ICML*, 2012.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *ArXiv e-prints*, abs/1309.2388, September 2013. Preliminary version appeared in NIPS 2012.

Shai Shalev-Shwartz. *Online learning: Theory, algorithms, and applications*. PhD thesis, Hebrew University, 2007.

Shai Shalev-Shwartz. SDCA without Duality, Regularization, and Individual Convexity. In *ICML*, 2016.

Shai Shalev-Shwartz and Tong Zhang. Proximal Stochastic Dual Coordinate Ascent. *ArXiv e-prints*, abs/1211.2717, 2012.

Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14:567–599, 2013. URL `http://arxiv.org/abs/1209.1873`.

Shai Shalev-Shwartz and Tong Zhang. Accelerated Proximal Stochastic Dual Coordinate Ascent for Regularized Loss Minimization. In *Proceedings of the 31st International Conference on Machine Learning*, ICML 2014, pages 64–72, 2014.

Atsushi Shibagaki and Ichiro Takeuchi. Stochastic primal dual coordinate method with non-uniform sampling based on optimality violations. *ArXiv e-prints*, abs/1703.07056, 2017.

Blake Woodworth and Nati Srebro. Tight Complexity Bounds for Optimizing Composite Objectives. In *NIPS*, 2016.

Lin Xiao and Tong Zhang. A Proximal Stochastic Gradient Method with Progressive Variance Reduction. *SIAM Journal on Optimization*, 24(4):2057—-2075, 2014.

Lijun Zhang, Mehrdad Mahdavi, and Rong Jin. Linear convergence with condition number independent access of full gradients. In *Advances in Neural Information Processing Systems*, pages 980–988, 2013.

Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the 21st International Conference on Machine Learning*, ICML 2004, 2004.

Yuchen Zhang and Lin Xiao. Stochastic Primal-Dual Coordinate Method for Regularized Empirical Risk Minimization. In *ICML*, 2015. URL `http://arxiv.org/abs/1409.3257`.

Journal of Machine Learning Research 18 (2018) 1-13 Submitted 10/16; Revised 10/17; Published 4/18

Average Stability is Invariant to Data Preconditioning. Implications to Exp-concave Empirical Risk Minimization

Alon Gonen alongnn@cs.huji.ac.il
School of Computer Science and Engineering
The Hebrew University
Jerusalem, Israel

Shai Shalev-Shwartz shais@cs.huji.ac.il
School of Computer Science and Engineering
The Hebrew University
Jerusalem, Israel

Editor: Ingo Steinwart

Abstract

We show that the average stability notion introduced by Kearns and Ron (1999); Bousquet and Elisseeff (2002) is invariant to data preconditioning, for a wide class of generalized linear models that includes most of the known exp-concave losses. In other words, when analyzing the stability rate of a given algorithm, we may assume the optimal preconditioning of the data. This implies that, at least from a statistical perspective, explicit regularization is not required in order to compensate for ill-conditioned data, which stands in contrast to a widely common approach that includes a regularization for analyzing the sample complexity of generalized linear models. Several important implications of our findings include: a) We demonstrate that the excess risk of empirical risk minimization (ERM) is controlled by the preconditioned stability rate. This immediately yields a relatively short and elegant proof for the fast rates attained by ERM in our context. b) We complement the recent bounds of Hardt et al. (2015) on the stability rate of the Stochastic Gradient Descent algorithm.

1. Introduction

Central to statistical learning theory is the notion of (algorithmic) *stability*. Since being introduced by Bousquet and Elisseeff (2002), deep connections between the *generalization* ability and the algorithmic stability of a learning algorithm have been established. It was shown by Shalev-Shwartz et al. (2010); Mukherjee et al. (2006) that stability characterizes learnability. Furthermore, in expectation, some notion of stability exactly equals the generalization error of an algorithm (namely, to the gap between true loss and train loss).

For generalized linear learning problems, a prominent geometric property which upper bounds the stability rate is the condition number of the loss function. While uniform convergence bounds Shalev-Shwartz and Ben-David (2014)[Chapter 4] mostly yield bounds that scale with $1/\sqrt{n}$, where n is the size of the sample, *well-conditioned* problems admit faster (stability) rates that scale linearly with $1/n$. The caveat is that typical (large-scale) machine learning problems are ill-conditioned. While we defer the precise definition of the *condition number* to the next part, let us mention that the condition number is controlled by two related quantities corresponding to both the choice of the loss function and the choice of the coordinate system. In a nutshell, our paper establishes the following result:

The average stability of ERM is invariant to the choice of the coordinate system.

While this observation admits a one-line proof, it has far-reaching implications. In particular, in this paper we use this observation to establish fast rates for empirical risk minimization.

The rest of the paper is organized as follows. In Section 2 we define the setting and proceed to provide basic definitions and results in stability analysis. In Section 3 we state and prove our main result. Section 4 discusses the implications to linear regression as well as improved bounds on the stability of SGD. Related work is discussed in Section 5.

2. Preliminaries

2.1 Setup

We consider the problem of minimizing the *risk* associated with *generalized linear model*:

$$\min_{w\in\mathcal{W}} L(w) := \mathbb{E}_{(x,y)\sim\mathcal{D}}[\phi_y(w^\top x)] \ . \tag{1}$$

Here, both the *domain* $\mathcal{W}$ and the *instance space* $\mathcal{X}$ are assumed to be compact and convex subsets of $\mathbb{R}^d$. We denote by $\mathcal{D}$ an arbitrary probability distribution defined over $\mathcal{X}\times\mathcal{Y}$. Each element y in the *label set* $\mathcal{Y}$ induces a twice differentiable[1] *loss function* of the form $\phi_y : \{w^\top x : \ w\in\mathcal{W},\ x\in\mathcal{X}\} \to \mathbb{R}_+$. We make the following assumptions on the loss function:

(A1) For each $y\in\mathcal{Y}$, ϕ_y is ρ-Lipschitz, i.e., $|\phi_y'(z)|\leq\rho$ for all z.
(A2) For each $y\in\mathcal{Y}$, ϕ_y is α-strongly convex, i.e., $\phi_y''(z)\geq\alpha$ for all z.

Our main example is the following formulation of *linear regression* Orabona et al. (2012).

Example 1 ***(Linear Regression:)*** *Let $\mathcal{X}$ be any compact and convex subset of $\mathbb{R}^d$ and $\mathcal{Y}$ be an interval of the form $[-Y,Y]$. The domain $\mathcal{W}$ is given by*

$$\mathcal{W} = \{w\in\mathbb{R}^d : (\forall x\in\mathcal{X})\ |w^\top x|\leq Y\} \ .$$

For all $y\in\mathcal{Y}$, let ϕ_y be the square loss, $\phi_y(z)=\frac{1}{2}(z-y)^2$. Note that for any $y\in\mathcal{Y}$ and $z\in\{w^\top x : w\in\mathcal{W},\ x\in\mathcal{X}\}$,

$$|\phi_y'(z)| = \frac{1}{2}|2(z-y)| \leq |z|+|y| \leq 2Y, \quad \|\phi_y''(z)\| = 1 \ .$$

*Hence, the assumptions **(A1-2)** are satisfied with $\rho=2Y$ and $\alpha=1$.*

More generally, our setting captures most of the known *exp-concave* functions Kivinen and Warmuth (1999). A twice-continuously differentiable function $f:\mathcal{W}\to\mathbb{R}$ is said to be $\bar{\alpha}$-exp-concave if $\nabla^2 f(w)\succeq\bar{\alpha}\nabla f(w)\nabla f(w)^\top$ for all $w\in\mathcal{W}$.

1. As we do not require smoothness of the loss function, our results can easily be extended to continuous but non-differentiable functions.

Lemma 1 *Consider a risk of the form (1) that satisfies the assumptions **(A1-2)**. Then, for any $(x, y) \in \mathcal{X} \times \mathcal{Y}$, the function $w \in \mathcal{W} \mapsto \phi_y(w^\top x)$ is α/ρ^2-exp concave.*

Proof Fix a pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$. The gradient and the Hessian of the map $\ell(w) = \phi_y(w^\top x)$ are given by

$$\nabla \ell(w) = \phi'(w^\top x)x, \quad \nabla^2 \ell(w) = \phi''(w^\top x)xx^\top \ . \tag{2}$$

By assumption $|\phi'(w^\top x)| \le \rho$ and $\phi''(w^\top x) \ge \alpha$, hence ℓ is α/ρ^2-exp concave. ■

A learning algorithm $\mathcal{A}$ receives as an input a training sequence (a.k.a. sample) of n i.i.d. pairs, $S = ((x_i, y_i))_{i=1}^n \sim \mathcal{D}^n$, and outputs a predictor, $\mathcal{A}(S) \in \mathcal{W}$. The *empirical risk function*, $\hat{L} : \mathcal{W} \to \mathbb{R}$, is defined as

$$\hat{L}_S(w) = \hat{L}(w) = \frac{1}{n} \sum_{i=1}^n \underbrace{\phi_{y_i}(w^\top x_i)}_{:=\hat{\ell}_i(w)} \ . \tag{3}$$

In this paper we focus on the ERM algorithm, whose output is a minimizer of the empirical risk.[2] We denote the output of the ERM by $\hat{w}(S)$, or simply $\hat{w}$ when S is understood from the context. The generalization error and the excess risk of $\hat{w}$ are defined by $L(\hat{w}) - \hat{L}(w)$ and $L(\hat{w}) - L(w^\star)$, respectively. For ERM, it is immediate that any upper bound on the generalization error translates into the same bound on the excess risk.

Remark 2 *While we mostly focus on exact ERM, it should be emphasized that our results are easily extended to any algorithm that approximately minimizes the empirical risk. The formulation of Lemma 3 below highlights this idea.*

2.2 Stability

In this section we review basic definitions and results on stability. For completeness, we also provide proofs of the stated results.

Let $S = ((x_i, y_i))_{i=1}^n$ be a training sequence. For every $i \in [n]$, let $\hat{w}_i$ be a minimizer of the risk w.r.t. $S \setminus \{(x_i, y_i)\}$, namely,

$$\hat{w}_i \in \operatorname*{argmin}_{w \in \mathcal{W}} \frac{1}{n-1} \sum_{j \ne i} \hat{\ell}_j(w) \ .$$

The *average stability* of ERM is defined as

$$\Delta(S, \mathcal{W}) = \frac{1}{n} \sum_{i=1}^n (\hat{\ell}_i(\hat{w}_i) - \hat{\ell}_i(\hat{w})) \ . \tag{4}$$

We omit the dependency on $\mathcal{W}$ when it is clear from the context. The next lemma shows that the expected generalization error of the ERM is equal to the expected average stability.

2. The compactness of $\mathcal{W}$ implies that both the true and the empirical risks admit minimizers.

Lemma 3 *Let $\mathcal{A}$ be a possibly randomized algorithm and denote by $\hat{w}$ its output. The generalization error of $\mathcal{A}$ satisfies*

$$\mathbb{E}_{S \sim \mathcal{D}^{n-1}}[L(\hat{w}) - \hat{L}(\hat{w})] = \mathbb{E}_{S \sim \mathcal{D}^n}[\Delta(S)] \ . \tag{5}$$

Furthermore, if $\mathcal{A}$ satisfies, for every sample S, $\mathbb{E}[\hat{L}(\hat{w})] \le \min_{w \in \mathcal{W}} \hat{L}(w) + \epsilon$, where the expectation is with respect to $\mathcal{A}$'s own randomization, then the excess risk of $\mathcal{A}$ is bounded by

$$\mathbb{E}_{S \sim \mathcal{D}^{n-1}}[L(\hat{w}) - L(w^\star)] \le \mathbb{E}_{S \sim \mathcal{D}^n}[\Delta(S)] + \epsilon \ . \tag{6}$$

Proof Since $\hat{w}_i$ does not depend on the i.i.d. pair (x_i, y_i),

$$\mathbb{E}_{S \sim \mathcal{D}^n}[\hat{\ell}_i(\hat{w}_i)] = \mathbb{E}_{S \sim \mathcal{D}^{n-1}}[L(\hat{w}(S))] \ , i = 1, \ldots, n \ .$$

By linearity of expectation, we obtain

$$\mathbb{E}_{S \sim \mathcal{D}^n}\left[\frac{1}{n} \sum_{i=1}^n \hat{\ell}_i(\hat{w}_i)\right] = \mathbb{E}_{S \sim \mathcal{D}^{n-1}}[L(\hat{w}(S))] \ .$$

Therefore,

$$\mathbb{E}[\Delta(S)] = \mathbb{E}_{S \sim \mathcal{D}^n}\left[\frac{1}{n} \sum_{i=1}^n \hat{\ell}_i(\hat{w}_i)\right] - \mathbb{E}_{S \sim \mathcal{D}^n}\left[\frac{1}{n} \sum_{i=1}^n \hat{\ell}_i(\hat{w})\right] = \mathbb{E}_{S \sim \mathcal{D}^{n-1}}[L(\hat{w})] - \mathbb{E}[\hat{L}(\hat{w})] \ .$$

This establishes the first claim.

Next, by assumption, for every S, $\mathbb{E}[\hat{L}(\hat{w})] \le \hat{L}(w^\star) + \epsilon$. Hence,

$$\mathbb{E}_{S \sim \mathcal{D}^n}[\hat{L}(\hat{w})] \le \mathbb{E}_{S \sim \mathcal{D}^n}[\hat{L}(w^\star)] + \epsilon = L(w^\star) + \epsilon \ .$$

Combining this inequality with the first claim, concludes the proof. ■

2.2.1 Stability of Well-conditioned Objectives

Lemma 3 motivates us to derive an upper bound on the average stability. A key quantity that governs $\Delta(S)$ is the condition number of the objective. We next provide exact definitions and discuss this relation.

Fix a training sequence S. We denote the empirical covariance matrix by

$$\hat{C} := \hat{C}(S) = \frac{1}{n} \sum_{i=1}^n x_i x_i^\top \ .$$

The (average) empirical condition number of $\hat{C}$ is defined as

$$\kappa(\hat{C}) = \frac{\operatorname{tr}(\hat{C})}{\lambda_{\min}(\hat{C})} \ ,$$

where $\mathrm{tr}(\hat{C})$ is the trace of $\hat{C}$ and $\lambda_{\min}(\hat{C})$ is the smallest nonzero eigenvalue of $\hat{C}$. We define the functional condition number as the ratio between the squared Lipschitz parameter and the strong convexity parameter:

$$\kappa(\phi) = \frac{\rho^2}{\alpha} \ .$$

Finally, we define the condition number of the objective as the product between the empirical and the functional condition number:

$$\kappa = \kappa(\hat{C})\kappa(\phi) \ .$$

Lemma 4 *For every training sequence S,*

$$\Delta(S) \leq \frac{2\kappa}{n} = \frac{2\kappa(\hat{C})\kappa(\phi)}{n} = \frac{2\rho^2}{\alpha\, n}\kappa(\hat{C}) \ . \tag{7}$$

To the best of our knowledge, this result has only been proved in the context of regularized loss minimization (e.g., the bound on the uniform stability in Shalev-Shwartz and Ben-David (2014)[Corollary 13.6]). Inspecting the proofs, one can notice that the role of regularization is merely to ensure the strong convexity of the objective. This simple observation is crucial for our development.

Proof (of Lemma 4) We first assume that $\hat{C}$ is of full rank. Note that for all w, the Hessian of $\hat{L}$ at w is given by

$$\nabla^2 \hat{L}(w) = \frac{1}{n}\sum_{i=1}^{n} \phi''(w^\top x_i) x_i x_i^\top \succeq \frac{1}{n}\sum_{i=1}^{n} \alpha\, x_i x_i^\top = \alpha\, \hat{C} \ . \tag{8}$$

In particular, $\hat{L}$ is strongly convex and $\hat{w}$ is uniquely defined. Denote the strong convexity parameter of $\hat{L}$ by $\hat{\mu}$. We also denote the Lipschitz parameter of each $\hat{\ell}_i$ by $\hat{\rho}_i$ and define $\hat{\rho}^2 = \frac{1}{n}\sum_{i=1}^{n} \hat{\rho}_i^2$. We will shortly derive upper and lower bounds on these parameters, but first let us relate them to the average stability.

Fix some $i \in [n]$ and let $\hat{\Delta}_i = \hat{\ell}_i(\hat{w}_i) - \hat{\ell}_i(\hat{w})$ (we do not assume that $\hat{w}_i$ is uniquely defined). The $\hat{\rho}_i$-Lipschitzness of $\hat{\ell}_i$ yields the bound

$$\hat{\Delta}_i \leq \hat{\rho}_i \|\hat{w}_i - \hat{w}\| \ .$$

The $\hat{\mu}$-strong convexity of $\hat{L}$ implies (e.g. using Shalev-Shwartz (2011)[Lemma 2.8]) that

$$\frac{\hat{\mu}}{2}\|\hat{w}_i - \hat{w}\|^2 \leq \hat{L}(\hat{w}_i) - \hat{L}(\hat{w}) \ .$$

On the other hand, since $\hat{w}_i$ minimizes the risk over $S \setminus \{(x_i, y_i)\}$, we have that

$$\hat{L}(\hat{w}_i) - \hat{L}(\hat{w}) = \frac{\sum_{j\neq i}(\hat{\ell}_j(\hat{w}_i) - \hat{\ell}_j(\hat{w}))}{n} + \frac{\hat{\ell}_i(\hat{w}_i) - \hat{\ell}_i(\hat{w})}{n} \leq 0 + \frac{\hat{\Delta}_i}{n} \ .$$

Combining the bounds, we conclude the following bound for every $i \in [n]$:

$$\hat{\Delta}_i^2 \leq \hat{\rho}_i^2 \|\hat{w}_i - \hat{w}\|^2 \leq \frac{2\hat{\rho}_i^2}{\hat{\mu}}(\hat{L}(\hat{w}_i) - \hat{L}(\hat{w})) \leq \frac{2\hat{\rho}_i^2}{n\hat{\mu}}\hat{\Delta}_i \ .$$

Dividing by $\hat{\Delta}_i$ (we may assume w.l.o.g. that $\hat{\Delta}_i > 0$), we obtain

$$\hat{\Delta}_i \leq \frac{2\hat{\rho}_i^2}{n\hat{\mu}} \ . \tag{9}$$

Let us remark that at this point, we can deduce a bound of $\max_{i\in[n]} \frac{2\hat{\rho}_i^2}{n\hat{\mu}}$ on the uniform stability. This matches the bound in Shalev-Shwartz and Ben-David (2014)[Corollary 13.6]. We next proceed to establish the claimed bound on the average stability.

By averaging (9) over $i = 1, \ldots, n$, we obtain

$$\hat{\Delta} = \frac{1}{n}\sum_{i=1}^{n} \hat{\Delta}_i \leq \left(\frac{1}{n}\sum_{i=1}^{n} \hat{\rho}_i^2\right)\frac{2}{n\hat{\mu}} = \frac{2\hat{\rho}^2}{n\hat{\mu}} \ . \tag{10}$$

It remains to derive bounds on $\hat{\rho}$ and $\hat{\mu}$. Note that

$$\|\nabla \hat{\ell}_i(w)\|^2 = \|\phi'(w^\top x_i)x_i\|^2 \leq \rho^2\|x_i\|^2 = \rho^2 \mathrm{tr}(x_i x_i^\top) \ .$$

Hence, $\hat{\rho}_i^2 \leq \rho^2 \mathrm{tr}(x_i x_i^\top)$. By averaging, we obtain that $\hat{\rho}^2 \leq \rho^2 \mathrm{tr}(\hat{C})$. Next, using (8) we obtain that $\hat{\mu} \geq \alpha\lambda_d(\hat{C})$. By substituting the bounds on $\hat{\rho}^2$ and $\hat{\mu}$ in (10), we conclude the desired bound.

Note that if $\hat{C}$ is not of full rank, we can replace each vector $x \in \mathbb{R}^d$ with $U^\top x$, where the columns of U form an orthonormal basis for $\mathrm{span}(\{x_1, \ldots, x_n\})$, without affecting $\hat{\Delta}, \hat{\Delta}_1, \ldots, \hat{\Delta}_n$ (this modification is only for the sake of the analysis). As a result, the new covariance matrix is of full rank and its eigenvalues are $\lambda_1(\hat{C}), \ldots, \lambda_{\min}(\hat{C})$. Repeating the above arguments, we conclude the proof. ■

Let us specify the bound to linear regression as formulated in Example (1). As $\alpha = 1$ and $\rho = 2Y$, the functional condition number is $4Y^2$. Hence, the average stability is bounded by

$$\Delta(S) \leq \frac{4Y^2}{n}\hat{\kappa}(\hat{C}) \ . \tag{11}$$

Using Lemma 3 we deduce the same bound on the excess risk. The weakness of this bound stems from the fact that empirically, the empirical condition number tends to be huge (e.g., see the empirical study in Gonen et al. (2016)).

In the next section we show that the (dependence on the) empirical condition number associated with our arbitrary choice of coordinate system can be replaced by the empirical condition number obtained by an optimal preconditioning.

3. Preconditioned Stability

We are now in position to describe our main result. Let P be a (symmetric) positive definite matrix, S_P be the training set obtained by replacing every x_j with $\tilde{x}_j = P^{-1/2}x_j$, and $\mathcal{W}_P = P^{1/2}\mathcal{W}$. We call $P^{-1/2}$ a *preconditioner*. Recall the definition of average stability from Equation (4). Our main theorem is:

Theorem 5 *For any training sequence S and positive definite matrix P,*

$$\Delta(S_P, \mathcal{W}_P) = \Delta(S, \mathcal{W}) \ .$$

In words, the average stability is invariant to the choice of the coordinate system.

Proof The crucial observation is that the empirical risk minimization with respect to S_P over the domain $\mathcal{W}_P$ is equivalent to the ERM w.r.t. S over the domain $\mathcal{W}$ in the following sense. For any pair $(w, \tilde{w} = P^{1/2}w) \in \mathcal{W} \times \mathcal{W}_P$ and any $j \in [n]$, the prediction $(\tilde{w})^\top \tilde{x}_j$ is equal to the prediction $w^\top x_j$. Therefore, the empirical risks $\hat{L}_{S_p}(\tilde{w})$ and $\hat{L}_S(w)$ are equal. By associating the corresponding minimizers of the empirical risk (i.e., $\hat{w}$ is associated with $P^{1/2}\hat{w}$ and for any $i \in [n]$, $\hat{w}_i$ is associated with $P^{1/2}\hat{w}_i$), we conclude our proof. ■

Theorem 5 tells us that we can analyze the stability of S_P instead of the stability of S. Crucially, this is true for every P, even one that is chosen based on S. Therefore, the expected suboptimality is upper bounded by the expected value of the quantity, $\inf_{P \succ 0} \Delta(S_P, \mathcal{W}_P)$, which we refer to as the *preconditioned average stability*. Equipped with this observation, we next choose P that leads to a minimal condition number, and consequently obtain a tighter bound on the excess risk.

Note that for every $P \succ 0$, the empirical covariance matrix that corresponds to the preconditioned training sequence, S_P, is

$$\frac{1}{n}\sum_{i=1}^{n}(P^{-1/2}x_i)(P^{-1/2}x_i)^\top = P^{-1/2}\left(\frac{1}{n}\sum_{i=1}^{n}x_i x_i^\top\right)P^{-1/2} = P^{-1/2}\hat{C}P^{-1/2} \ .$$

When $\hat{C}$ is of full rank, by choosing $P = \hat{C}$, we obtain that

$$\kappa(\underbrace{P^{-1/2}\hat{C}P^{-1/2}}_{I}) = \frac{\mathrm{tr}(I)}{\lambda_{\min}(I)} = d \ .$$

If $\hat{C}$ is not of full rank, we can add arbitrary "noise" in directions that do not lie in the column space of $\hat{C}$. For example, by choosing $P = \hat{C} + \delta(I - \hat{C}\hat{C}^\dagger)$, (where δ can be any positive scalar), we obtain that $\kappa(P^{-1/2}\hat{C}P^{-1/2}) = \mathrm{rank}(\hat{C}) \leq d$. It is easy to see that in both cases, we obtain the minimal value of $\kappa(P^{-1/2}\hat{C}P^{-1/2})$ over all matrices $P \succ 0$. Combining this bound with Lemma 3 and Lemma 4, we arrive at the following conclusion.

Corollary 6 *Consider the optimization problem (1), where for all $y \in \mathcal{Y}$, ϕ_y is ρ-Lipschitz and α-strongly convex. The expected excess risk of empirical risk minimization is bounded by*

$$\mathbb{E}_{S\sim\mathcal{D}^{n-1}}[L(\hat{w}) - L(w^\star)] \leq \mathbb{E}_{S\sim\mathcal{D}^n}[\Delta(S)] = \mathbb{E}_{S\sim\mathcal{D}^n}[\inf_{P\succ 0}\Delta(S_P)] \leq \frac{2\rho^2 d}{\alpha n} \ .$$

4. Some Implications

4.1 Linear Regression

We start by specifying our bounds to linear regression (Example (1)).

Corollary 7 ***(Linear Regression)*** *Consider linear regression as formulated in Example (1). The expected excess risk of ERM is bounded by*

$$\mathbb{E}_{S\sim\mathcal{D}^{n-1}}[L(\hat{w}) - L(w^\star)] \leq \Delta(S) \leq \frac{4Y^2 d}{n} \ .$$

Comparing the bounds in (11) and Corollary 7, we see that the dependence on $\hat{\kappa}(\hat{C})$ is replaced by the optimal empirical condition number, $\hat{\kappa}(I) = d$. As we mentioned above, this gap tends to be huge in practice.

As we discuss in Section 5, standard bounds for this setting depend on the geometry of $\mathcal{X}$ and $\mathcal{W}$. On the contrary, it follows from the generalized Cauchy-Schwarz inequality that for any choice of a norm $\|\cdot\|$ on $\mathbb{R}^d$, our bound applies to the sets[3]

$$\mathcal{X} = \mathcal{B}_{\|\cdot\|} = \{x \in \mathbb{R}^d : \|x\| \leq 1\}, \quad \mathcal{W} = Y\mathcal{B}_{\|\cdot\|^\star} := \{w \in \mathbb{R}^d : \|w\|^\star \leq Y\} \tag{12}$$

4.2 The Average Stability of Stochastic Gradient Descent

One of the most widely used algorithms in machine learning is Stochastic Gradient Descent (SGD). Besides its computational simplicity, its popularity stems also from its generalization abilities Shalev-Shwartz and Ben-David (2014)[Section 14.5]. Recently, Hardt et al. (2015) studied the (uniform) stability of SGD in various settings. Following our notation, theorem 3.9 of their paper implies a bound of $\max_i \frac{2\hat{\rho}_i^2}{\gamma n}$ on the uniform stability, where γ is the strong convexity of the entire objective, and for any $i \in [n]$, $\hat{\rho}_i$ is the Lipschitz parameter of $\hat{\ell}_i$. As the proof of Lemma 4 reveals, γ can be bounded by $\alpha\hat{\kappa}(\hat{C})$ and $\hat{\rho}_i$ is at most $\hat{\rho}^2\|x_i\|^2$. In particular, the bound depends on the choice of the coordinate system.

As implied by Bubeck (2015)[Theorem 3.2], SGD can be viewed in our context as an (approximate) ERM. Hence, the average stability of SGD is invariant to the choice of the coordinate system and the stability rate of SGD is bounded as in Corollary 6. It should be noted that our bound holds in the regime where SGD converges to a minimizer of the empirical risk, whereas the result of Hardt et al. holds for the entire trajectory of SGD.

5. Related Work

5.1 Slower rates

One of the most direct techniques for establishing bounds on the excess risk is via analyzing the Rademacher complexity Bartlett and Mendelson (2003) of the associated class of predictors. In our setting, these techniques have been employed by Kakade et al. (2009) to

3. In fact, under mild additional assumptions on $\mathcal{X}$, any two sets $\mathcal{X}$ and $\mathcal{W}$ that satisfy our assumptions can be presented in this way. Assume that $\mathcal{X}$ is symmetric (i.e., $x \in \mathcal{X}$ iff $-x \in \mathcal{X}$) and $0 \in \mathrm{int}(\mathcal{X})$. Then it is known Conway (2013) that $\mathcal{X}$ induces a norm on $\mathbb{R}^d$ through the Minkowsky functional

$$\|x\| := p(x) = \inf\{t \in \mathbb{R} : x \in tB\} \ .$$

It is immediate that the closed unit ball $\{x : \|x\| \leq 1\}$ is $\mathcal{X}$ itself. Therefore, the dual norm is simply the support function of $\mathcal{X}$

$$\|w\|^\star = \max_{x\in\mathcal{X}} w^\top x \ .$$

It follows that $\mathcal{X}$ and $\mathcal{W}$ can be described as in (12).

establish bounds of order $1/\sqrt{n}$ on the generalization error of ERM. We refer to these rates as slower due to the inferior dependence on the sample size n.

5.2 Dependence on Norm

Note that since both the uniform and the average stability of ERM are bounded above by its generalization error Shalev-Shwartz et al. (2010), the bounds of Kakade et al. (2009) translate into bounds on the average stability.

Unlike our fast rates, the exact bounds depend on the geometry of the set $\mathcal{X}$ and $\mathcal{W}$. For example: a) If both $\mathcal{X}$ and $\mathcal{W}$ are the Euclidean unit ball in $\mathbb{R}^d$, then the obtained bound scales with $1/\sqrt{n}$. b) If $\mathcal{X}$ is the unit ℓ_∞-ball and $\mathcal{W}$ is the ℓ_1-ball, then the obtained bound scales with $\sqrt{\log(d)/n}$.

5.3 Lower bounds on the excess risk

Lower bounds for stochastic minimization of exp-concave functions have been studied in Mahdavi et al. (2015). For our setting, theorem 2 in this paper implies a bound of $\Omega(d/n)$ on the excess risk of any algorithm.

For the special case of linear regression with

$$\mathcal{X} = \{x \in \mathbb{R}^d : \|x\|_2 \leq 1\},\ \mathcal{W} = \{w \in \mathbb{R}^d : \|w\|_2 \leq B\},\ \mathcal{Y} = [-Y, Y] \tag{13}$$

Shamir (2014) proved the lower bound $\Omega\left(\min\{Y^2, \frac{B^2+dY^2}{n}, \frac{BY}{\sqrt{n}}\}\right)$ on the generalization error of ERM. The left-most term is trivially attained by the predictor $w = 0$. The middle term is attained by combining the Vovk-Azoury-Warmuth forecaster Azoury and Warmuth (2001); Vovk (2001) with standard online-to-batch conversions (Cesa-Bianchi et al. (2004)). Last, the right term is attained by ERM, as implied by the aforementioned upper bound of Kakade et al. (2009).

It is left open whether the middle term in the lower bound is attained by ERM. Note that if $B = \omega(\sqrt{d}Y)$, then the middle term in the above lower bound is asymptotically larger than our upper bound. However, since in the setting of Shamir (2014), Equation (13)) the magnitude of the predictions is not uniformly upper bounded by Y, no contradiction arises.

5.4 Stability and Regularization

Previous work Bousquet and Elisseeff (2002); Shalev-Shwartz et al. (2010) studied the rate of uniform stability in various settings. For our setting, their bounds on the expected risk are identical to the bound in Lemma 4. As we explained above, these fast rates are often worse than the so-called slower rates due to the dependence on the empirical condition number. The standard approach for tackling this problem is add a regularization term. By adding the regularization term $\lambda\|w\|^2$ to the objective, one effectively increases the eigenvalues of $\hat{C}$ by λ, and consequently, the overall condition number is decreased. However, as explained in Shalev-Shwartz and Ben-David (2014)[Section 13.4], this modification usually does not preserve the fast rates.[4]

4. Namely, when tuning λ, we need to ensure that any $\epsilon/2$-approximate minimizer with respect to the regularized objective is also an ϵ-approximate minimizer with respect to the unregularized objective. As explained in Shalev-Shwartz and Ben-David (2014)[Section 13.3], by optimally controlling this tradeoff, we no longer obtain fast rates (i.e., the stability rate scales with $1/\sqrt{n}$ rather than $1/n$).

5.5 Stability and Exp-concavity

Informally, exp-concavity can be seen as a local and weaker form of strong convexity. Indeed, the Online Newton Step (ONS) of Hazan et al. (2007), which has been designed for online minimization of exp-concave functions, achieves improved (logarithmic) regret bounds that resemble the regret bounds for strongly convex functions Hazan et al. (2007). The online-to-batch analysis of Mahdavi et al. (2015) yields a bound on the excess risk that coincides with our bounds up to logarithmic factors. The main shortcoming of the ONS algorithm is that it employs expensive iterations (the runtime per iteration scales at least quadratically with d). Hence, it is natural to ask whether there exist simpler algorithms that achieve fast rates.

This question was answered affirmatively by Koren and Levy (2015). This work, which is most closely related to our work, considers the minimization of a risk of the form $F(w) = \mathbb{E}[f(w, Z)]$, where for any z, $f(\cdot, z)$ is $\bar{\beta}$-smooth[5] and $\bar{\alpha}$-exp-concave function. They established fast rates for any algorithm that minimizes the *regularized* risk $\hat{L}(w) + \frac{1}{n}R(w)$, where $R(w)$ is assumed to be a 1-strongly convex function (e.g., one can set $R(w) = \frac{1}{2}\|w\|^2$). While exp-concavity is weaker than strong convexity, Koren and Levy (2015)[section 4.2] interprets exp-concavity as strong convexity in the (local) norm induced by the outer products of the gradients and the regularization term. In other words, the problem is well-conditioned with respect to this local norm. Note that their formulation is more general in the sense that they do not assume a GLM structure. However, it should be emphasized that all the known exp-concave functions in machine learning are of the form (1)).

The above interpretation of Koren and Levy (2015) inspired us to make one step forward and directly show that regularization is not required as long as a related (preconditioned) problem is well conditioned. Besides the obvious importance of showing the insignificance of regularization in this context, we believe that the notion of preconditioned stability and its relation to the excess risk make these ideas more transparent and simplify the proofs.

The upper bound of Koren and Levy (2015) on the excess risk scales with $\frac{24\beta d}{\bar{\alpha}n} = \frac{24\beta d\rho^2}{\alpha n}$ (recall that the exp-concavity parameter $\bar{\alpha}$ is equal to α/ρ^2). Note that our analysis does not assume smoothness of the loss. This resolves the question raised by Koren and Levy (2015) regarding the necessity of the smoothness assumption. Note that for linear regression, the smoothness is 1, making our bounds identical to the bounds of Koren and Levy (2015) for this special case.

As discussed in Koren and Levy (2015), it is difficult to translate bounds on the average stability into high-probability bounds (while preserving the fast rate and introducing only logarithmic dependence on $1/\delta$).

5.6 Other Techniques and High-Probability Bounds

The bound on the expected excess risk in Corollary 6 can be obtained by using two additional techniques. Both of these techniques also yield high probability bounds. We next survey the corresponding results.

5. That is, the maximal eigenvalue of the Hessian of f at any point w is at most β.

A recent follow-up work by Mehta (2016) established a bound of $\tilde{O}(d\log(n)+\log(1/\delta)/n)$ on the excess risk of ERM, where δ is the confidence parameter.[6]. He also showed how to get rid of the $\log(n)$ factor by boosting the confidence of ERM. The proof is centered around a Bernstein condition which holds due to the exp-concavity assumption.

Another alternative, is to bound the excess risk by the local Rademacher complexity (LRC) of the associated class of predictors (e.g., using Corollary 5.3 in Bartlett et al. (2005)). In our setting, one can derive bounds on the LRC (e.g., using Koltchinskii (2008)) which coincide with our bounds.

All of these techniques employ arguably heavy machinery and lack the geometric interpretation, which is nicely captured by our notion of preconditioned stability.

Acknowledgments

We thank Iliya Tolstikhin for pointing out the alternative proof of Corollary 6 using local Rademacher complexities.

6. The dependence on the exp-concavity parameter as well as the diameter of the loss function is hidden.

References

Katy S Azoury and Manfred K Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211–246, 2001.

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *The Journal of Machine Learning Research*, 3:463–482, 2003.

Peter L Bartlett, Olivier Bousquet, and Shahar Mendelson. Local rademacher complexities. *Annals of Statistics*, pages 1497–1537, 2005.

Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Nicolo Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *Information Theory, IEEE Transactions on*, 50(9):2050–2057, 2004.

John B Conway. *A course in functional analysis*, volume 96. Springer Science & Business Media, 2013.

Alon Gonen, Francesco Orabona, and Shai Shalev-Shwartz. Solving ridge regression using sketched preconditioned svrg. *arXiv preprint arXiv:1602.02350*, 2016.

Moritz Hardt, Benjamin Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. *arXiv preprint arXiv:1509.01240*, 2015.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

Sham M Kakade, Karthik Sridharan, and Ambuj Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *Advances in neural information processing systems*, pages 793–800, 2009.

Michael Kearns and Dana Ron. Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. *Neural Computation*, 11(6):1427–1453, 1999.

Jyrki Kivinen and Manfred K Warmuth. Averaging expert predictions. In *Computational Learning Theory*, pages 153–167. Springer, 1999.

V Koltchinskii. Oracle inequalities in empirical risk minimization and sparse recovery problems: Lecture notes. Technical report, Technical report, Ecole dete de Probabilites de Saint-Flour, 2008. 12.6, 2008.

Tomer Koren and Kfir Levy. Fast rates for exp-concave empirical risk minimization. In *Advances in Neural Information Processing Systems*, pages 1477–1485, 2015.

Mehrdad Mahdavi, Lijun Zhang, and Rong Jin. Lower and upper bounds on the generalization of stochastic exponentially concave optimization. In *Proceedings of The 28th Conference on Learning Theory*, pages 1305–1320, 2015.

Nishant A Mehta. From exp-concavity to variance control: O (1/n) rates and online-to-batch conversion with high probability. *CoRR*, 2016.

Sayan Mukherjee, Partha Niyogi, Tomaso Poggio, and Ryan Rifkin. Learning theory: stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization. *Advances in Computational Mathematics*, 25(1-3):161–193, 2006.

Francesco Orabona, Nicolo Cesa-Bianchi, and Claudio Gentile. Beyond logarithmic bounds in online learning. In *AISTATS*, pages 823–831, 2012.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *The Journal of Machine Learning Research*, 11:2635–2670, 2010.

Ohad Shamir. The sample complexity of learning linear predictors with the squared loss. *arXiv preprint arXiv:1406.5143*, 2014.

Volodya Vovk. Competitive on-line statistics. *International Statistical Review*, 69(2):213–248, 2001.

Journal of Machine Learning Research 18 (2018) 1-42 Submitted 11/16; Revised 3/18; Published 6/18

Parallelizing Stochastic Gradient Descent for Least Squares Regression: Mini-batching, Averaging, and Model Misspecification

Prateek Jain, Praneeth Netrapalli {PRAJAIN,PRANEETH}@MICROSOFT.COM
Microsoft Research, Bangalore 560001, INDIA

Sham M. Kakade SHAM@CS.WASHINGTON.EDU
Paul G. Allen School of Computer Science and Department of Statistics, University of Washington, Seattle WA 98195, USA

Rahul Kidambi RKIDAMBI@UW.EDU
Department of Electrical Engineering, University of Washington, Seattle WA 98195, USA

Aaron Sidford SIDFORD@STANFORD.EDU
Department of Management Science and Engineering, Stanford University, Palo Alto CA 94305, USA

Editor: Leon Bottou

Abstract

This work characterizes the benefits of averaging techniques widely used in conjunction with stochastic gradient descent (SGD). In particular, this work presents a sharp analysis of: (1) mini-batching, a method of averaging many samples of a stochastic gradient to both reduce the variance of a stochastic gradient estimate and for parallelizing SGD and (2) tail-averaging, a method involving averaging the final few iterates of SGD in order to decrease the variance in SGD's final iterate. This work presents sharp finite sample generalization error bounds for these schemes for the stochastic approximation problem of least squares regression.

Furthermore, this work establishes a precise problem-dependent extent to which mini-batching can be used to yield provable near-linear parallelization speedups over SGD with batch size one. This characterization is used to understand the relationship between learning rate versus batch size when considering the excess risk of the final iterate of an SGD procedure. Next, this mini-batching characterization is utilized in providing a highly parallelizable SGD method that achieves the minimax risk with nearly the same number of serial updates as batch gradient descent, improving significantly over existing SGD-style methods. Following this, a non-asymptotic excess risk bound for model averaging (which is a communication efficient parallelization scheme) is provided.

Finally, this work sheds light on fundamental differences in SGD's behavior when dealing with mis-specified models in the non-realizable least squares problem. This paper shows that maximal stepsizes ensuring minimax risk for the mis-specified case *must* depend on the noise properties.

The analysis tools used by this paper generalize the operator view of averaged SGD (Défossez and Bach, 2015) followed by developing a novel analysis in bounding these operators to characterize the generalization error. These techniques are of broader interest in analyzing various computational aspects of stochastic approximation.

Keywords: Stochastic Gradient Descent, Stochastic Approximation, Least Squares Regression, Parallelization, Mini Batch SGD, Iterate Averaging, Suffix Averaging, Batchsize Doubling, Model Averaging, Parameter Mixing, Mis-specified models, Heteroscedastic Noise, Agnostic Learning

1. Introduction and Problem Setup

With the ever increasing size of modern day datasets, practical algorithms for machine learning are increasingly constrained to spend less time and use less memory. This makes it particularly desirable to employ simple streaming algorithms that generalize well in a few passes over the dataset.

Stochastic gradient descent (SGD) is perhaps the simplest and most well studied algorithm that meets these constraints. The algorithm repeatedly samples an instance from the stream of data and updates the current parameter estimate using the gradient of the sampled instance. Despite its simplicity, SGD has been immensely successful and is the de-facto method for large scale learning problems. The merits of SGD for large scale learning and the associated computation versus statistics tradeoffs is discussed in detail by the seminal work of Bottou and Bousquet (2007).

While a powerful machine learning tool, unfortunately SGD in its simplest forms is inherently serial. Over the past years, as dataset sizes have grown there have been remarkable developments in processing capabilities with multi-core/distributed/GPU computing infrastructure available in abundance. The presence of this computing power has triggered the development of parallel/distributed machine learning algorithms (Mann et al. (2009); Zinkevich et al. (2011); Bradley et al. (2011); Niu et al. (2011); Li et al. (2014); Zhang and Xiao (2015)) that possess the capability to utilize multiple cores/machines. However, despite this exciting line of work, it is yet unclear how to best parallelize SGD and fully utilize these computing infrastructures.

This paper takes a step towards answering this question, by characterizing the behavior of constant stepsize SGD for the problem of strongly convex stochastic least square regression (LSR) under two averaging schemes widely believed to improve the performance of SGD. In particular, this work considers the natural parallelization technique of *mini-batching*, where multiple data-points are processed simultaneously and the current iterate is updated by the average gradient over these samples, and combine it with variance reducing technique of *tail-averaging*, where the average of many of the final iterates are returned as SGD's estimate of the solution.

In this work, parallelization arguments are structured through the lens of a *work-depth* tradeoff: *work* refers to the total computation required to reach a certain generalization error, and *depth* refers to the number of serial updates. Depth, defined in this manner, is a reasonable estimate of the runtime of the algorithm on a large multi-core architecture with shared memory, where there is no communication overhead, and has strong implications for parallelizability on other architectures.

1.1 Problem Setup and Notations

We use boldface small letters ($\mathbf{x}, \mathbf{w}$ etc.) for vectors, boldface capital letters ($\mathbf{A}, \mathbf{H}$ etc.) for matrices and normal script font letters ($\mathcal{M}, \mathcal{T}$ etc) for tensors. We use $\otimes$ to denote the outer product of two vectors or matrices. Loewner ordering between two PSD matrices is represented using $\succeq, \preceq$.
This paper considers the stochastic approximation problem of Least Squares Regression (LSR). Let $L : \mathbb{R}^d \to \mathbb{R}$ be the expected square loss over tuples $(\mathbf{x}, y)$ sampled from a distribution $\mathcal{D}$:

$$L(\mathbf{w}) = \frac{1}{2} \cdot \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[(y - \langle \mathbf{w}, \mathbf{x}\rangle)^2] \; \forall \, \mathbf{w} \in \mathbb{R}^d. \tag{1}$$

Let $\mathbf{w}^*$ be a minimizer of the problem (1). Now, let the Hessian of the problem (1) be denoted as:

$$\mathbf{H} \stackrel{\text{def}}{=} \nabla^2 L(\mathbf{w}) = \mathbb{E}\left[\mathbf{x}\mathbf{x}^\top\right].$$

Next, we define the fourth moment tensor $\mathcal{M}$ of the inputs $\mathbf{x}$ as:

$$\mathcal{M} \stackrel{\text{def}}{=} \mathbb{E}\left[\mathbf{x} \otimes \mathbf{x} \otimes \mathbf{x} \otimes \mathbf{x}\right].$$

Let the noise $\epsilon_{\mathbf{x},y}$ in a sample $(\mathbf{x}, y) \sim \mathcal{D}$ with respect to the minimizer $\mathbf{w}^*$ of (1) be denoted as:

$$\epsilon_{\mathbf{x},y} \stackrel{\text{def}}{=} y - \langle \mathbf{w}^*, \mathbf{x} \rangle.$$

Finally, let the noise covariance matrix $\mathbf{\Sigma}$ be denoted as:

$$\mathbf{\Sigma} \stackrel{\text{def}}{=} \mathbb{E}\left[\epsilon_{\mathbf{x},y}^2 \mathbf{x}\mathbf{x}^\top\right].$$

The *homoscedastic* (or, additive noise/well specified) case of LSR refers to the case when $\epsilon_{\mathbf{x},y}$ is mutually independent from $\mathbf{x}$. This is the case, say, when $\epsilon_{\mathbf{x},y}$ sampled from a Gaussian, $N(0, \sigma^2)$ independent of $\mathbf{x}$. In this case, $\mathbf{\Sigma} = \sigma^2 \mathbf{H}$, where, $\sigma^2 = \mathbb{E}\left[\epsilon^2\right]$, where the subscript on $\epsilon_{\mathbf{x},y}$ is suppressed owing to the independence of ϵ on any sample $(\mathbf{x}, y) \sim \mathcal{D}$. On the other hand, the *heteroscedastic* (or, mis-specified) case refers to the setting when $\epsilon_{\mathbf{x},y}$ is correlated with the input $\mathbf{x}$. In this paper, all our results apply to the general mis-specified case of the LSR problem.

1.1.1 ASSUMPTIONS

We make the following assumptions about the problem.

($\mathcal{A}1$) **Finite fourth moment:** The fourth moment tensor $\mathcal{M} = \mathbb{E}\left[\mathbf{x}^{\otimes 4}\right]$ exists and is finite.

($\mathcal{A}2$) **Strong convexity:** The Hessian of $L(\cdot)$, $\mathbf{H} = \mathbb{E}\left[\mathbf{x}\mathbf{x}^\top\right]$ is positive definite i.e., $\mathbf{H} \succ 0$.

($\mathcal{A}1$) is a standard regularity assumption for the analysis of SGD and related algorithms. ($\mathcal{A}2$) is also a standard assumption and guarantees that the minimizer of (1), i.e., $\mathbf{w}^*$ is unique.

1.1.2 IMPORTANT QUANTITIES

In this section, we will introduce some important quantities required to present our results. Let $\mathbf{I}$ denote the $d \times d$ identity matrix. For any matrix $\mathbf{A}$, $\mathcal{M}\mathbf{A} \stackrel{\text{def}}{=} \mathbb{E}\left[\left(\mathbf{x}^\top \mathbf{A} \mathbf{x}\right) \mathbf{x}\mathbf{x}^\top\right]$. Let $\mathcal{H}_{\mathcal{L}} = \mathbf{H} \otimes \mathbf{I}$ and $\mathcal{H}_{\mathcal{R}} = \mathbf{I} \otimes \mathbf{H}$ represent the left and right multiplication operators of the matrix $\mathbf{H}$ so that for any matrix $\mathbf{A}$, we have $\mathcal{H}_{\mathcal{L}}\mathbf{A} = \mathbf{H}\mathbf{A}$ and $\mathcal{H}_{\mathcal{R}}\mathbf{A} = \mathbf{A}\mathbf{H}$.

- **Fourth moment bound:** Let R^2 be the smallest number such that $\mathcal{M}\mathbf{I} \preceq R^2\mathbf{H}$.
- **Smallest eigenvalue:** Let μ be the smallest eigenvalue of $\mathbf{H}$ i.e., $\mathbf{H} \succeq \mu\mathbf{I}$.

The fourth moment bound implies that $\mathbb{E}\left[\|\mathbf{x}\|^2\right] \leq R^2$. Further more, ($\mathcal{A}2$) implies that the smallest eigenvalue μ of $\mathbf{H}$ is strictly greater than zero ($\mu > 0$).

1.1.3 STOCHASTIC GRADIENT DESCENT: MINI-BATCHING AND ITERATE AVERAGING

In this paper, we work with a stochastic first order oracle. This oracle, when queried at $\mathbf{w}$ samples an instance $(\mathbf{x}, y) \sim \mathcal{D}$ and uses this to return an unbiased estimate of the gradient of $L(\mathbf{w})$:

$$\widehat{\nabla L}(\mathbf{w}) = -(y - \langle \mathbf{w}, \mathbf{x} \rangle) \cdot \mathbf{x}; \;\; \mathbb{E}\left[\widehat{\nabla L}(\mathbf{w})\right] = \nabla L(\mathbf{w}).$$

We consider the stochastic gradient descent (SGD) method (Robbins and Monro, 1951), which minimizes $L(\mathbf{w})$ by following the direction opposite to this noisy stochastic gradient estimate, i.e.:

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \gamma \cdot \widehat{\nabla L}_t(\mathbf{w}_{t-1}), \text{ with, } \widehat{\nabla L}_t(\mathbf{w}_{t-1}) = -(y_t - \langle \mathbf{w}_{t-1}, \mathbf{x}_t \rangle) \cdot \mathbf{x}_t$$

with $\gamma > 0$ being a constant step size/learning rate; $\widehat{\nabla L}_t(\mathbf{w}_{t-1})$ is the stochastic gradient evaluated using the sample $(\mathbf{x}_t, y_t) \sim \mathcal{D}$ at $\mathbf{w}_{t-1}$. We consider two algorithmic primitives used in conjunction with SGD namely, mini-batching and tail-averaging (also referred to as iterate/suffix averaging).

Mini-batching involves querying the gradient oracle several times and using the average of the returned stochastic gradients to take a single step. That is,

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \gamma \cdot \left(\frac{1}{b}\sum_{i=1}^{b} \widehat{\nabla L}_{t,i}(\mathbf{w}_{t-1})\right),$$

where, b is the batch size. Note that at iteration t, mini-batching involves repeatedly querying the stochastic gradient oracle at $\mathbf{w}_{t-1}$ for a total of b times. For every query $i = 1, ..., b$ at iteration t, the oracle samples an instance $\{\mathbf{x}_{ti}, y_{ti}\}$ and returns a stochastic gradient estimate $\widehat{\nabla L}_{t,i}(\mathbf{w}_{t-1})$. These estimates $\{\widehat{\nabla L}_{t,i}(\mathbf{w}_{t-1})\}_{i=1}^b$ are averaged and then used to perform a single step from $\mathbf{w}_{t-1}$ to $\mathbf{w}_t$. Mini-batching enables the possibility of parallelization owing to the use of cheap matrix-vector multiplication for computing stochastic gradient estimates. Furthermore, mini-batching allows for the possible reduction of variance owing to the effect of averaging several stochastic gradient estimates.

Tail-averaging (or suffix averaging) refers to returning the average of the final few iterates of a stochastic gradient method as a means to improve its variance properties (Ruppert, 1988; Polyak and Juditsky, 1992). In particular, assuming the stochastic gradient method is run for $n-$steps, tail-averaging involves returning

$$\bar{\mathbf{w}} = \frac{1}{n-s}\sum_{t=s+1}^{n} \mathbf{w}_t$$

as an estimate of $\mathbf{w}^*$. Note that s can be interpreted as being cn, with $c < 1$ being some constant.

Typical excess risk bounds (or, generalization error bounds) for the stochastic approximation problem involve the contribution of two error terms namely, (i) the bias, which refers to the dependence on the starting conditions $\mathbf{w}_0$/initial excess risk $L(\mathbf{w}_0) - L(\mathbf{w}^*)$ and, (ii) the variance, which refers to the dependence on the noise introduced by the use of a stochastic first order oracle.

1.1.4 OPTIMAL ERROR RATES FOR THE STOCHASTIC APPROXIMATION PROBLEM

Under standard regularity conditions often employed in the statistics literature, the minimax optimal rate on the excess risk is achieved by the standard Empirical Risk Minimizer (or, Maximum Likelihood Estimator) (Lehmann and Casella, 1998; van der Vaart, 2000). Given n i.i.d. samples $\mathcal{S}_n = \{\mathbf{x}_i, y_i\}_{i=1}^n$ drawn from $\mathcal{D}$, define the empirical risk minimization problem as obtaining

$$\mathbf{w}_n^* = \arg\min_{\mathbf{w}} \frac{1}{2n}\sum_{i=1}^{n}(y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle)^2.$$

Let us define the noise variance $\widehat{\sigma^2_{\text{MLE}}}$ to represent

$$\widehat{\sigma^2_{\text{MLE}}} = \mathbb{E}\left[\|\widehat{\nabla L}(\mathbf{w}^*)\|^2_{\mathbf{H}^{-1}}\right] = \text{Tr}[\mathbf{H}^{-1}\mathbf{\Sigma}].$$

The asymptotic minimax rate of the Empirical Risk Minimizer $\mathbf{w}_n^*$ on *every problem instance* is $\widehat{\sigma^2_{\text{MLE}}}/n$ (Lehmann and Casella, 1998; van der Vaart, 2000), i.e.,

$$\lim_{n\to\infty} \frac{\mathbb{E}_{\mathcal{S}_n}[L(\mathbf{w}_n^*)] - L(\mathbf{w}^*)}{\widehat{\sigma^2_{\text{MLE}}}/n} = 1.$$

For the well-specified case (i.e., the additive noise case, where, $\mathbf{\Sigma} = \sigma^2\mathbf{H}$), we have $\widehat{\sigma^2_{\text{MLE}}} = d\sigma^2$. Seminal works of Ruppert (1988); Polyak and Juditsky (1992) prove that tail-averaged SGD, with averaging from start, achieves the minimax rate for the *well-specified* case in the limit of $n \to \infty$.

Goal: In this paper, we seek to provide a non-asymptotic understanding of (a) mini-batching and issues of learning rate versus batch-size, (b) tail-averaging, (c) the effect of the model mis-specification, (d) a batch size doubling scheme for parallelizing statistical estimation, (e) a communication efficient parallelization scheme namely, parameter-mixing/model averaging and (f) the behavior of learning rate versus batch size on the final iterate of the mini-batch SGD procedure, on the behavior of excess risk of SGD (in terms of both the bias and the variance terms) for the streaming LSR problem, with the goal of achieving the minimax rate on every problem instance.

1.2 This Paper's Contributions

The main contributions of this paper are as follows:

- This work shows that mini-batching yields near-linear parallelization speedups over the standard serial SGD (i.e. with batch size 1), as long as the mini-batch size is smaller than a problem dependent quantity (which we denote by b_{thresh}). When batch-sizes increase beyond b_{thresh}, mini-batching is inefficient (owing to the lack of serial updates), thus obtaining only sub-linear speedups over mini-batching with a batch size b_{thresh}. A by-product of this analysis sheds light on how the step sizes naturally interpolate from ones used by standard serial SGD (with batch size 1) to ones used by batch gradient descent.

- While the final iterate of SGD decays the bias at a geometric rate but does not obtain minimax rates on the variance, the averaged iterate (Polyak and Juditsky, 1992; Défossez and Bach, 2015) decays the bias at a sublinear rate while achieving minimax rates on the variance. This work rigorously shows that tail-averaging obtains the best of both worlds: decaying the bias at a geometric rate and obtaining near-minimax rates (up to constants) on the variance. This result corroborates with empirical findings (Merity et al., 2017) that indicate the benefits of tail-averaging in general contexts such as training Long-Short term memory models (LSTMs).

- Next, this paper precisely characterizes the tradeoffs of learning rate versus batch size and its effect on the excess risk of the final iterate of an SGD procedure, which provides theoretical evidence to empirical observations (Goyal et al., 2017; Smith et al., 2017) described in the context of deep learning and non-convex optimization.

- Combining the above results, this paper provides a mini-batching and tail-averaging version of SGD that is highly parallelizable: the number of serial steps (which is a proxy for the un-parallelizable time) of this algorithm nearly matches that of *offline gradient descent* and is lower than the serial time of all existing streaming LSR algorithms. See Table 1 for comparison. We note that these results are obtained by providing a tight finite-sample analysis of the effects of mini-batching and tail-averaging with large constant learning rate schemes.

- We provide a non-asymptotic analysis of parameter mixing/model averaging schemes for the streaming LSR problem. Model averaging schemes are an attractive proposition for distributed learning owing to their communication efficient nature, and they are particularly effective in the regime when the estimation error (i.e. variance) is the dominating term in the excess risk. Here, we characterize the excess risk (in terms of both the bias and variance) of the model averaging procedure which sheds light on situations when it is an effective parallelization scheme (in that when this scheme yields linear parallelization speedups).

- All the results in this paper are established for the *general mis-specified* case of the streaming LSR problem. This establishes a fundamental difference in the behavior of SGD when dealing with mis-specified models in contrast to existing analyses that deal with the well-specified case. In particular, this analysis reveals a surprising insight that the maximal stepsizes (that ensure minimax optimal rates) are a function of the noise properties of the mis-specified problem instance. The main takeaway of this analysis is that the maximal step sizes (that permit achieving minimax rates) for the mis-specified case can be *much lower* than ones employed in the well-specified case: indeed, a problem instance that yields such a separation between the maximal learning rates for the well specified and the mis-specified case is presented.

The tool employed in obtaining these results generalizes the operator view of averaged SGD with batch size 1 (Défossez and Bach, 2015) and a clear exposition of the bias-variance decomposition from Jain et al. (2017a) to obtain a sharp bound on the excess risk for mini-batch, tail-averaged constant step-size SGD. Note that the work of Défossez and Bach (2015) does not establish minimax rates while working with large constant step sizes; this shortcoming is remedied by this paper through a novel sharp analysis that rigorously establishes minimax optimal rates while working with large constant step sizes. Furthermore, note that while straightforward operator norm bounds of the matrix operators suffice to show convergence of the SGD method, they turn out to be pretty loose bounds (particularly for bounding the variance). To tighten these bounds, this paper presents a fine grained analysis that bounds the trace of the SGD operators when applied to the relevant matrices. The bounds of this paper and its advantages compared to existing algorithms is indicated in table 1.

While this paper's results focus on strongly convex streaming least square regression, we believe that our techniques and results extend more broadly. This paper aims to serve as the basis for future work on analyzing SGD and parallelization of large scale algorithms for machine learning.

Paper organization: Section 2 presents the related work. Section 3 presents the main results of this work. Section 4 outlines the proof techniques. Section 5 presents experimental simulations to demonstrate the practical utility of the established mini-batching limits and tail-averaging. The proofs of all the claims and theorems are provided in the appendix.

2. Related Work

Stochastic approximation has been the focus of much efforts starting with the work of Robbins and Monro (1951), and has been analyzed in subsequent works including Nemirovsky and Yudin (1983); Kushner and Yin (1987, 2003). These questions and the related issues of computation versus statistics tradeoffs have received renewed attention owing to their relevance in the context of modern large scale machine learning, as highlighted by the work of Bottou and Bousquet (2007).

Geometric Rates on initial error: For *offline optimization* with strongly convex objectives, gradient descent (Cauchy, 1847) and fast gradient methods (Polyak, 1964; Nesterov, 1983) indicate linear

Algorithm	Final error	Runtime/Work	Depth	Streaming	Mis-specified
Gradient Descent (Cauchy, 1847)	$\mathcal{O}\left(\frac{\sigma^2 d}{n}\right)$	$\kappa n d \log \frac{n\cdot\Delta_0}{\sigma^2 d}$	$\kappa \log \frac{n\cdot\Delta_0}{\sigma^2 d}$	×	✓
SDCA (Shalev-Shwartz and Zhang, 2012)	$\mathcal{O}\left(\frac{\sigma^2 d}{n}\right)$	$(n+\frac{R^2}{\lambda_{\min}}d)d\cdot\log\frac{n\cdot\Delta_0}{\sigma^2 d}$	$(n+\frac{R^2}{\lambda_{\min}}d)\cdot\log\frac{n\cdot\Delta_0}{\sigma^2 d}$	×	✓
Averaged SGD (Défossez and Bach, 2015)[1]	$\mathcal{O}\left(\frac{1}{\lambda_{\min}^2 n^{2\gamma}}\cdot\Delta_0+\frac{\sigma^2 d}{n}\right)$	nd	n	✓	×
Streaming SVRG with initial error oracle [2] (Frostig et al., 2015b)	$\mathcal{O}\left(\exp\left(-\frac{n\lambda_{\min}(\mathbf{H})}{R^2}\right)\cdot\Delta_0\right)+\frac{\sigma^2 d}{n}$	nd	$(\frac{R^2}{\lambda_{\min}(\mathbf{H})})\cdot\log\frac{n\cdot\Delta_0}{\sigma^2 d}$	✓	✓
Algorithm 2 (this paper)	$\mathcal{O}\left(\left(\frac{R^2 t}{\|\mathbf{H}\|_2 n}\right)^{\frac{1}{\kappa\log(\kappa)}}\cdot\Delta_0+\frac{\sigma^2 d}{n}\right)$	nd	$\frac{t}{t-\kappa\log(\kappa)}\cdot\kappa\log(\kappa)\cdot\log\left(\frac{n\cdot\Delta_0}{\sigma^2 d}\cdot\frac{R^2 t}{\|\mathbf{H}\|_2}\right)$	✓	✓
Algorithm 2 with initial error oracle (this paper)	$\mathcal{O}\left(\exp\left(-\frac{n\lambda_{\min}(\mathbf{H})}{R^2\cdot\log(\kappa)}\right)\cdot\Delta_0+\frac{\sigma^2 d}{n}\right)$	nd	$\kappa\log(\kappa)\log\frac{n\cdot\Delta_0}{\sigma^2 d}$	✓	✓

Table 1: Comparison of Algorithm 2 with existing algorithms including offline methods such as Gradient Descent, SDCA and streaming methods such as averaged SGD, streaming SVRG given n samples for LSR, with $\Delta_0 = L(\mathbf{w}_0) - L(\mathbf{w}^*)$. The error of offline methods are obtained by running these algorithms so that their final error is $\mathcal{O}(\sigma^2 d/n)$ (which is the minimax rate for the well-specified case). The table is written assuming the additive noise/well specified case; for algorithms which support the mis-specified case, these bounds can be appropriately modified. Refer to Section 1.1 for the definitions of all quantities. We do not consider accelerated variants in this table. Note that the accelerated variants have served to improve running times of the offline algorithms, with the sole exception of Jain et al. (2017b). For Algorithm 2, we require $t \geq 24\kappa\log(\kappa)$. Finally, note that streaming SVRG does not conform to the first order oracle model (Agarwal et al. (2012)).

convergence. However, a multiplicative coupling of number of samples n and condition number in the computational effort is a major drawback in the large scale context. These limitations are addressed through developments in offline stochastic methods (Roux et al., 2012; Shalev-Shwartz and Zhang, 2012; Johnson and Zhang, 2013; Defazio et al., 2014) and their accelerated variants (Shalev-Shwartz and Zhang, 2013a; Frostig et al., 2015a; Lin et al., 2015; Defazio, 2016; Allen-Zhu, 2016) which offer near linear running time in the number of samples and condition number with $\log(n)$ passes over the dataset stored in memory.

For *stochastic approximation* with strongly convex objectives, SGD offers linear rates on the bias without achieving minimax rates on the variance (Bach and Moulines, 2011; Needell et al., 2016; Bottou et al., 2016). In contrast, iterate averaged SGD (Ruppert, 1988; Polyak and Juditsky, 1992) offers a sub-linear $\mathcal{O}(1/n^2)$ rate on the bias (Défossez and Bach, 2015; Dieuleveut and Bach, 2015) while achieving minimax rates on the variance. Note that all these results consider the well-specified (additive noise) case when stating the generalization error bounds. We are unaware of any results that provide sharp non-asymptotic analysis of SGD and the related step size issues in the general mis-specified case. Streaming SVRG (Frostig et al., 2015b) offers a geometric rate on the bias and optimal statistical error rates; we will return to a discussion of Streaming SVRG below. In terms of methods faster than SGD, our own effort (Jain et al., 2017b) provides the first accelerated stochastic approximation method that improves over SGD on every problem instance.

Parallelization of Machine Learning algorithms: In *offline optimization*, Bradley et al. (2011) study parallel co-ordinate descent for sparse optimization. Parallelization via mini-batching has been studied in Cotter et al. (2011); Takác et al. (2013); Shalev-Shwartz and Zhang (2013b); Takác et al. (2015). These results compare worst case upper bounds on the training error to argue parallelization speedups, thus providing weak upper bounds on mini-batching limits. Parameter mixing/Model averaging (Mann et al., 2009) guarantees linear parallelization speedups on the variance but do not improve the bias. Approaches that attempt to re-conciliate communication-computation tradeoffs (Li et al., 2014) indicate increased mini-batching hurts convergence, and this is likely an artifact of comparing weak upper bounds. Hogwild (Niu et al., 2011) indicates near-linear parallelization speedups in the harder asynchronous optimization setting, relying on specific input structures like hard sparsity; these bounds are obtained by comparing worst case upper bounds on training error. Refer to oracle models paragraph below for details on these worst case upper bounds.

In the *stochastic approximation* context, Dekel et al. (2012) study mini-batching in an oracle model that assumes bounded variance of stochastic gradients. These results compare worst case bounds on the generalization error to prescribe mini-batching limits, which renders these limits to be too loose (as mentioned in their paper). Our paper's mini-batching result offers guidelines on batch sizes for linear parallelization speedups by comparing generalization bounds that hold on a per problem basis as opposed to worst case bounds. Refer to the paragraph on oracle models for more details. Finally, parameter mixing in the stochastic approximation context (Rosenblatt and Nadler, 2014; Zhang et al., 2015) offers linear parallelization speedups on the variance error while not improving the bias (Rosenblatt and Nadler, 2014). Finally, Duchi et al. (2015) guarantees asymptotic optimality of asynchronous optimization with linear parallelization speedups on the variance.

Oracle models and optimality: In stochastic approximation, there are at least two lines of thought with regards to oracle models and notions of optimality. One line involves considering the case of bounded noise (Kushner and Yin, 2003; Kushner and Clark, 1978), or, bounded variance of the stochastic gradient, which in the least squares setting amounts to assuming bounds on

$$\widehat{\nabla L}(\mathbf{w}) - \nabla L(\mathbf{w}) = (\mathbf{x}\mathbf{x}^\top - \mathbf{H})(\mathbf{w} - \mathbf{w}^*) - \epsilon\mathbf{x}.$$

This implies additional assumptions are required on compactness of the parameter set (which are enforced via projection steps); such assumptions do not hold in practical implementation of stochastic gradient methods and in the setting considered by this paper. Thus, the mini-batching thresholds in Cotter et al. (2011); Niu et al. (2011); Dekel et al. (2012); Li et al. (2014) present bounds in the above worst-case oracle model by comparing weak upper bounds on the training/test error.

Another view of optimality (Anbar, 1971; Fabian, 1973) considers an objective where the goal is to match the rate of the statistically optimal estimator (referred to as the $M-$estimator) on every problem instance. Polyak and Juditsky (1992) consider this oracle model for the LSR problem and prove that the distribution of the averaged SGD estimator on every problem matches that of the $M-$estimator under certain regularity conditions (Lehmann and Casella, 1998). A recent line of work (Bach and Moulines, 2013; Frostig et al., 2015b) aims to provide non-asymptotic guarantees for SGD and its variants in this oracle model. This paper aims to understand mini-batching and other computational aspects of parallelizing stochastic approximation on every problem instance by working in this practically relevant oracle model. Refer to Jain et al. (2017b) for more details.

Comparing offline and streaming algorithms: Firstly, offline algorithms require performing multiple passes over a dataset stored in memory. Note that results and convergence rates established in the finite sum/offline optimization context do not translate to rates on the generalization error. Indeed, these results require going though concentration and a generalization error analysis for this translation to occur. Refer to Frostig et al. (2015b) for more details.

1. Défossez and Bach (2015)'s bound holds with learning rate $\gamma \to 0$. This work supports these bounds with $\gamma = 1/R^2$.
2. Initial error oracle provides initial excess risk $\Delta_0 = L(\mathbf{w}_0) - L(\mathbf{w}^*)$ and noise level σ^2.

Comparison to streaming SVRG: Streaming SVRG does not function in the stochastic first order oracle model (Agarwal et al., 2012) satisfied by SGD as run in practice since it requires gradients at two points from a single sample (Frostig et al., 2015b). Furthermore, in contrast to this work, its depth bounds depend on a stronger fourth moment property due to lack of mini-batching.

3. Main Results

We begin by writing out the behavior of the learning rate as a function of batch size.

Maximal Learning Rates: We write out a characterization of the largest learning rate $\gamma_{b,\max}^{div}$ that permits the convergence of the mini-batch Stochastic Gradient Descent update. The following generalized eigenvector problem allows for the computation of $\gamma_{b,\max}^{div}$:

$$\frac{2}{\gamma_{b,\max}^{div}} = \sup_{\mathbf{W}\in\mathcal{S}(d)} \frac{\langle \mathbf{W}, \mathcal{M}\mathbf{W}\rangle + (b-1)\cdot \operatorname{Tr}\mathbf{WHWH}}{b\cdot \operatorname{Tr}\mathbf{WHW}}. \tag{2}$$

This characterization generalizes the divergent stepsize characterization of Défossez and Bach (2015) for batch sizes > 1. The derivation of the above characterization can be found in appendix A.5.1. We note that this characterization sheds light on how the divergent learning rates interpolate from batch size 1 (which is $\leq 2/\operatorname{Tr}\mathbf{H}$) to the batch gradient descent learning rate (setting b to ∞), which turns out to be $2/\lambda_{\max}(\mathbf{H})$. A property of $\gamma_{b,\max}^{div}$ worth noting is that it does not depend on properties of the noise ($\mathbf{\Sigma}$), and depends only on the second and fourth moment properties of the covariate $\mathbf{x}$.

We note that in this paper, our interest does not lie in the non-divergent stepsizes $0 \leq \gamma \leq \gamma_{b,\max}^{div}$, but in the set of (maximal) stepsizes $0 \leq \gamma \leq \gamma_{b,\max}$ ($< \gamma_{b,\max}^{div}$) that are sufficient to guarantee minimax error rates of $\mathcal{O}(\widehat{\sigma^2_{\text{MLE}}}/n)$. For the LSR problem, these maximal learning rates $\gamma_{b,\max}$ are:

$$\gamma_{b,\max} \stackrel{\text{def}}{=} \frac{2b}{R^2\cdot\rho_{\text{m}} + (b-1)\|\mathbf{H}\|_2}, \text{ where, } \rho_{\text{m}} \stackrel{\text{def}}{=} \frac{d\|(\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\|_2}{\operatorname{Tr}\left((\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right)}. \tag{3}$$

Note that $\rho_{\text{m}} \geq 1$ captures a notion of "degree" of model mismatch, and how it impacts the learning rate $\gamma_{b,\max}$; for the additive noise/well specified/homoscedastic case, $\rho_{\text{m}} = 1$. Thus, for problems where R^2 and $\|\mathbf{H}\|_2$ is held the same, the well-specified variant of the LSR problem admits a strictly larger learning rate (that achieves minimax rates on the variance) compared to the mis-specified case. Furthermore, in stark contrast to the well-specified case, $\gamma_{b,\max}$ in the mis-specified case depends not just on the second and fourth moment properties of the input, but also on the noise covariance $\mathbf{\Sigma}$. We show that our characterization of $\gamma_{b,\max}$ in the mis-specified case is tight in that there exist problem instances where $\gamma_{b,\max}$ (equation 3) is off the maximal learning rate in the well-specified case (obtained by setting $\rho_{\text{m}} = 1$ in equation 3) by a factor of the dimension d and $\gamma_{b,\max}$ is still the largest step size yielding minimax rates. We also note that there could exist mis-specified problem instances where a step size γ exceeding $\gamma_{b,\max}$ achieves minimax rates. Characterizing the maximal learning rate that achieves minimax rates on *every mis-specified* problem instance is an interesting open question. We return to the characterization of $\gamma_{b,\max}$ in section 3.1.

Note that this paper characterizes the performance of Algorithms 1 and 2 when run with a step size $\gamma \leq \frac{\gamma_{b,\max}}{2}$. The proofs turn out to be significantly complicated for $\gamma \in \left(\frac{\gamma_{b,\max}}{2}, \gamma_{b,\max}\right)$ and can be found in the initial version of this paper Jain et al. (2016b) and these were obtained through generalizing the operator view of analyzing SGD methods introduced by Défossez and Bach (2015). Note that for the well-specified case, this paper's results hold for the same learning rate regimes as

Algorithm 1 Minibatch-TailAveraging-SGD

Input: Initial point $\mathbf{w}_0$, stepsize γ, minibatch size b, initial iterations s, total samples n.

for $t = 1, 2, .., \lfloor \frac{n}{b} \rfloor$ **do**
  Sample "b" tuples $\{(x_{ti}, y_{ti})\}_{i=1}^{b} \sim \mathcal{D}^b$
  $\mathbf{w}_t \leftarrow \mathbf{w}_{t-1} - \frac{\gamma}{b}\sum_{i=1}^{b} \widehat{\nabla L_{ti}}(\mathbf{w}_{t-1})$

Output: $\bar{\mathbf{w}} = \frac{1}{\lfloor \frac{n}{b} \rfloor - s}\sum_{i>s} \mathbf{w}_i$

Bach and Moulines (2013); Frostig et al. (2015b), that are known to admit statistical optimality. We also note that in the additive noise case, we are unaware of a separation between $\gamma_{b,\max}$ and $\gamma_{b,\max}^{div}$; but as we will see, this is not of much consequence given that there exists a strict separation in the learning rate $\gamma_{b,\max}$ between the well-specified and mis-specified problem instances.

Finally, we note that the stochastic process viewpoint allows us to work with learning rates that are significantly larger compared to standard analyses that use function value contraction e.g., Bottou et al. (2016, Theorem 4.6). To the best of our knowledge, all existing works establishing mini-batching thresholds in the stochastic optimization setting e.g., Dekel et al. (2012) work in the worst case (bounded noise) oracle model, with small step sizes, and draw conclusions on mini-batch thresholds and effects by comparing weak upper bounds on the excess risk.

Mini-Batched Tail-Averaged SGD for the mis-specified case: We present our main result, which is the error bound for mini-batch tail-averaged SGD for the general mis-specified LSR problem.

Theorem 1 *Consider the general mis-specified case of the LSR problem 1. Running Algorithm 1 with a batch size $b \geq 1$, step size $\gamma \leq \gamma_{b,max}/2$, number of unaveraged iterations s, total number of samples n, we obtain an iterate $\overline{\mathbf{w}}$ satisfying the following excess risk bound:*

$$\mathbb{E}\left[L(\overline{\mathbf{w}})\right] - L(\mathbf{w}^*) \leq \frac{2}{\gamma^2\mu^2}\cdot\frac{(1-\gamma\mu)^s}{\left(\frac{n}{b}-s\right)^2}\cdot\left(L(\mathbf{w}_0) - L(\mathbf{w}^*)\right) + 4\cdot\frac{\widehat{\sigma^2_{MLE}}}{b\cdot(\frac{n}{b}-s)}. \tag{4}$$

In particular, with $\gamma = \gamma_{b,max}/2$, we have the following excess risk bound:

$$L(\overline{\mathbf{w}}) - L(\mathbf{w}^*) \leq \underbrace{\frac{2\kappa_b^2}{\left(\frac{n}{b}-s\right)^2}\exp\left(-\frac{s}{\kappa_b}\right)\left(L(\mathbf{w}_0) - L(\mathbf{w}^*)\right)}_{\mathfrak{T}_1} + \underbrace{4\cdot\frac{\widehat{\sigma^2_{MLE}}}{b(\frac{n}{b}-s)}}_{\mathfrak{T}_2},$$

with $\kappa_b = \frac{R^2\cdot\rho_m + (b-1)\|\mathbf{H}\|_2}{b\lambda_{min}(\mathbf{H})}$.

Note that the above theorem indicates that the excess risk is composed of two terms, namely the bias ($\mathfrak{T}_1$), which represents the dependence on the initial conditions $\mathbf{w}_0$ and the variance ($\mathfrak{T}_2$), which depends on the statistical noise ($\widehat{\sigma^2_{\text{MLE}}}$); the bias decays geometrically during the "s" unaveraged iterations while the variance is minimax optimal (up to constants) provided $s = \mathcal{O}(n)$. We will understand this geometric decay on the bias more precisely.

Effect of tail-averaging SGD's iterates: To understand tail-averaging, we specialize theorem 1 with a batch size 1 to the well-specified case, i.e., where, $\mathbf{\Sigma} = \sigma^2\mathbf{H}$, $\widehat{\sigma^2_{\text{MLE}}} = d\sigma^2$ and $\rho_{\text{m}} = 1$.

Corollary 2 *Consider the well-specified (additive noise) case of the streaming LSR problem ($\mathbf{\Sigma} = \sigma^2\mathbf{H}$), with a batch size $b = 1$. With a learning rate $\gamma = \frac{\gamma_{1,max}}{2} = \frac{1}{R^2}$, unaveraged iterations s and total samples n, we have the following excess risk bound:*

$$L(\overline{\mathbf{w}}) - L(\mathbf{w}^*) \leq \underbrace{\frac{2\kappa_1^2}{(n-s)^2}\exp\left(-\frac{s}{\kappa_1}\right)\{L(\mathbf{w}_0) - L(\mathbf{w}^*)\}}_{\mathfrak{T}_1} + \underbrace{4\cdot\frac{d\sigma^2}{n-s}}_{\mathfrak{T}_2}, \text{where, } \kappa_1 = R^2/\mu.$$

Tail-averaging allows for a geometric decay of the initial error $\mathfrak{T}_1$, while tail-averaging over $s = c\cdot n$ (with $c < 1$), allows for the variance $\mathfrak{T}_2$ to be minimax optimal (up to constants). We note that the work of Merity et al. (2017), which studies empirical optimization for training non-convex sequence models (e.g. Long-Short term memory models (LSTMs)) also indicate the benefits of tail-averaging.

Note that this particular case (i.e. additive noise/well-specified case with batch size 1) with tail-averaging from start ($s = 0$) is precisely the setting considered in Défossez and Bach (2015), and their result (a) achieves a sub-linear $\mathcal{O}(1/n^2)$ rate on the bias and (b) their variance term is shown to be minimax optimal only with learning rates that approach zero (i.e. $\gamma \to 0$).

3.1 Effects Of Learning Rate, Batch Size and The Role of Mis-specified Models

We now consider the interplay of learning rate, batch size and how model mis-specification plays into the mix. Towards this, we split this section into three parts: (a) understanding learning rate versus mini-batch size in the well-specified case, (b) how model mis-specification leads to a significant difference in the behavior of SGD and (c) how model mis-specification manifests itself when considered in tradeoff between the learning rate versus batch-size.

Effects of mini-batching in the well-specified case: As mentioned previously, in the well-specified case, $\mathbf{\Sigma} = \sigma^2\mathbf{H}$ and $\rho_{\text{m}} = 1$. For this case, equation (3) can be specialized as:

$$\gamma_{b,\max} = \frac{2b}{R^2 + (b-1)\|\mathbf{H}\|_2}. \tag{5}$$

Observe that the learning rate $\gamma_{b,\max}$ grows linearly as a function of the batch size b until a batch size $b = b_{\text{thresh}} = 1 + \frac{R^2}{\|\mathbf{H}\|_2}$. In the regime of batch sizes $1 < b \leq b_{\text{thresh}}$, the resulting mini-batch SGD updates offer near-linear parallelization speedups over SGD with a batch size of 1. Furthermore, increasing batch sizes beyond b_{thresh} leads to sub-linear increase in the learning rate, and this implies that we lose the linear parallelization speedup offered by mini-batching with a batch-size $b \leq b_{\text{thresh}}$. Losing the linear parallelization is indicative of the following: consider the case when we double batch-size from $b > b_{\text{thresh}}$ to $2b$. Suppose the bias error $\mathfrak{T}_1$ is larger than the variance $\mathfrak{T}_2$, we require performing the same number of updates with a batch size $2b$ as we did with a batch size b to achieve a similar excess risk bound; this implies we are inefficient in terms of number of samples (or, number of gradient computations) used to achieve a given excess risk. When the estimation error ($\mathfrak{T}_2$) dominates the approximation error ($\mathfrak{T}_1$), we note that larger batch sizes b (with $b > b_{\text{thresh}}$) serves to improve the variance term, thus allowing linear parallelization speedups via mini-batching.

Note that with a batch size of $b = b_{\text{thresh}}$, the learning rate of $\mathcal{O}(1/\lambda_{\max}(\mathbf{H}))$ employed by mini-batch SGD resembles ones used by batch gradient descent. This mini-batching characterization thus allows for understanding tradeoffs of learning rate versus batch size. This behavior is noted in practice (empirically, but with no underlying rigorous theory) for a variety of problems (going beyond linear regression/convex optimization), in the deep learning context (Goyal et al., 2017).

SGD's behaviour with mis-specified models: Next, this paper attempts to shed light on some fundamental differences in the behavior of SGD when dealing with the mis-specified case (as against the well-specified case, which is the focus of existing results (Polyak and Juditsky, 1992; Bach and Moulines, 2013; Dieuleveut and Bach, 2015; Défossez and Bach, 2015)) of the LSR problem. This paper's results in general mis-specified case with batch sizes $b > 1$ specialize to existing results additive noise/well-specified case with batch size 1 (Bach and Moulines, 2013; Dieuleveut and Bach, 2015). To understand these issues better, we consider $\gamma_{b,\max}$ in equation 3 with a batch size 1:

$$\gamma_{1,\max} = \frac{2}{R^2\cdot\rho_{\text{m}}}. \tag{6}$$

Recounting that $\rho_{\text{m}} \geq 1$, observe that the mis-specified case admits a maximal learning rate (with a view of achieving minimax rates) that is at most as large as the additive noise/well-specified case, where $\rho_{\text{m}} = 1$. Note that when $\text{Tr}\left(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}}\right)^{-1}\mathbf{\Sigma}\big)$ is nearly the same (say, upto constants) as the spectral norm $\left\|\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}}\right)^{-1}\mathbf{\Sigma}\right\|_2$, then $\rho_{\text{m}} = \mathcal{O}(d)$ and $\gamma_{1,\max} = \mathcal{O}(\frac{1}{R^2 d})$. This implies that there exist mis-specified models whose noise properties (captured through the noise covariance matrix $\mathbf{\Sigma}$) prevents SGD from working with large learning rates of $\mathcal{O}(1/R^2)$ used in the well-specified case.

This notion is formalized in the following lemma, which presents an instance working with the mis-specified case, wherein, SGD *cannot* employ large learning rates used by the well-specified variant of the problem, while *retaining minimax optimality*. This behavior is in stark contrast to algorithms such as streaming SVRG (Frostig et al. (2015b)), which work with the same large learning rates in the mis-specified case as in the well-specified case, while guaranteeing minimax optimal rates. The proof of lemma 3 can be found in the appendix A.5.6.

Lemma 3 *Consider a Streaming LSR example with Gaussian covariates (i.e. $\mathbf{x} \sim \mathcal{N}(0, \mathbf{H})$) with a diagonal second moment matrix $\mathbf{H}$ that is defined by:*

$$\mathbf{H}_{ii} = \begin{cases} 1 & \text{if } i = 1 \\ 1/d & \text{if } i > 1 \end{cases}.$$

Further, let the noise covariance matrix $\mathbf{\Sigma}$ be diagonal as well, with the following entries:

$$\mathbf{\Sigma}_{ii} = \begin{cases} 1 & \text{if } i = 1 \\ 1/[(d-1)d] & \text{if } i > 1 \end{cases}.$$

For this problem instance, $\gamma_{1,max} \leq \frac{4}{(d+2)(1+\frac{1}{d})}$ is necessary for retaining minimax rates, while the well-specified variant of this problem permits a maximal learning rate $\leq \frac{d}{(d+2)(1+\frac{1}{d})}$, thus implying an $\mathcal{O}(d)$ separation in learning rates between the well-specified and mis-specified case.

Learning rate versus mini-batch size issues in the mis-specified case: Noting that for the batch size 1, as mentioned in equation 6, the learning rate for the mis-specified case in the most optimistic situation (when $\rho_{\text{m}} = \text{constant}$) can be atmost as large as the learning rate for the well-specified case. Furthermore, we also know from the observations in the mis-specified case that the learning rate tends to grow linearly as a function of the batch size until it hits the limit of $\mathcal{O}(1/\lambda_{\max}(\mathbf{H}))$. Combining these observations, we will revisit equation 3, which says:

$$\gamma_{b,\max} \overset{\text{def}}{=} \frac{2b}{R^2\cdot\rho_{\text{m}} + (b-1)\|\mathbf{H}\|_2}.$$

This implies that the mini-batching size threshold b_{thresh} can be expressed as:

$$b_{\text{thresh}} \stackrel{\text{def}}{=} 1 + \frac{R^2}{\|\mathbf{H}\|_2} \cdot \rho_{\text{m}}. \tag{7}$$

When $1 < b \leq b_{\text{thresh}}$, we achieve near linear parallelization speedups over running SGD with a batch size 1. Note that this characterization specializes to the batch size threshold b_{thresh} presented in the well-specified case (i.e. where $\rho_{\text{m}} = 1$). Furthermore, this batch size threshold (in the mis-specified case) could be much larger than the threshold in the well-specified case, which is expected since the learning rate for a batch size 1 in the mis-specified case can potentially be much smaller than ones used in the well specified case. Furthermore, with a batch size b_{thresh}, note that the learning rate is $\mathcal{O}(1/\lambda_{\max}(\mathbf{H}))$, resembling ones used with batch gradient descent.

Behavior of the final-iterate: We now present the excess risk bound offered by the final iterate of a stochastic gradient scheme. This result is of much practical relevance in the context of modern machine learning and deep learning, where final iterate is often used, and where the tradeoffs between learning rate and batch sizes are discussed in great detail (Smith et al., 2017). For this discussion, we consider the well-specified case to present our results owing to its ease in presentation. Our framework and results are generic for translating these observations to the mis-specified case.

Lemma 4 *Consider the well-specified case of the LSR problem. Running Algorithm 1 with a step size $\gamma \leq \frac{\gamma_{b,max}}{2} = \frac{b}{R^2+(b-1)\|\mathbf{H}\|_2}$, batch size b, total samples n and with no iterate averaging (i.e. with $s = n-1$) yields a result $\mathbf{w}_{\lfloor n/b \rfloor}$ that satisfies the following excess risk bound:*

$$\mathbb{E}\left[L(\mathbf{w}_{\lfloor n/b \rfloor})\right] - L(\mathbf{w}^*) \leq \kappa_b (1-\gamma\mu)^{\lfloor n/b \rfloor} \left(L(\mathbf{w}_0) - L(\mathbf{w}^*)\right) + \frac{\gamma}{b}\sigma^2 \operatorname{Tr}(\mathbf{H}), \tag{8}$$

where $\kappa_b \stackrel{\text{def}}{=} \frac{R^2+(b-1)\|\mathbf{H}\|_2}{b\mu}$. In particular, with a step size $\gamma = \frac{\gamma_{b,max}}{2} = \frac{b}{R^2+(b-1)\|\mathbf{H}\|_2}$, we have:

$$\mathbb{E}\left[L(\mathbf{w}_{\lfloor n/b \rfloor})\right] - L(\mathbf{w}^*) \leq \kappa_b \cdot e^{-\frac{\lfloor n/b \rfloor}{\kappa_b}} \cdot \left(L(\mathbf{w}_0) - L(\mathbf{w}^*)\right) + \frac{\sigma^2 \operatorname{Tr}(\mathbf{H})}{R^2 + (b-1)\|\mathbf{H}\|_2}. \tag{9}$$

Remarks: Noting that $\operatorname{Tr}(\mathbf{H}) \leq R^2$, the variance of the final iterate with batch size 1 is $\leq \sigma^2$. Next, with a batch size $b = b_{\text{thresh}}$, the final iterate has a variance $\leq \sigma^2/2$; at cursory glance this may appear interesting, in that by mini-batching, we do not appear to gain much in terms of the variance. This is unsurprising given that in the regime of $b \leq b_{\text{thresh}}$, the $\gamma_{b,\max}$ grows linearly, thus nullifying the effect of averaging multiple stochastic gradients. Furthermore, this follows in accordance with the linear parallelization speedups offered by a batch size $1 < b \leq b_{\text{thresh}}$. Note however, once $b > b_{\text{thresh}}$, any subsequent increase in batch sizes allows the variance of the final iterate to behave as $\mathcal{O}(\sigma^2/b)$. Finally, we note that once $b > b_{\text{thresh}}$, doubling batch sizes b (in equation 9) possesses the same effect as halving the learning rate from γ to $\gamma/2$ (as seen from equation 8), providing theoretical rigor to issues explored in training practical deep models (Smith et al., 2017).

3.2 Parallelization via Doubling Batch Sizes and Model Averaging

We now elaborate on a highly parallelizable stochastic gradient method, which is epoch based and relies on doubling batch sizes across epochs to yield an algorithm that offers the same generalization error as that of offline (batch) gradient descent in nearly the same number of serial updates as

Algorithm 2 MinibatchDoublingPartialAveragingSGD

Input: Initial point $\mathbf{w}_0$, stepsize γ, initial minibatch size b, number of iterations in each epoch s, number of samples n.

/*Run logarithmic number of epochs where each epoch runs t iterations of minibatch SGD (with out averaging). Double minibatch size after each epoch.*/

for $\ell = 1, 2, \cdots, \log \frac{n}{bt} - 1$ **do**

  $b_\ell \leftarrow 2^{\ell-1} b$

  $\mathbf{w}_\ell \leftarrow$ Minibatch-TailAveraging-SGD$(\mathbf{w}_{\ell-1}, \gamma, b_\ell, t-1, t \cdot b_\ell)$

/*For the last epoch, run tail averaged minibatch SGD with initial point $\mathbf{w}_t$, stepsize γ, mini-batch size $2^{\log \frac{n}{bt}-1} \cdot b = n/2t$, number of initial iterations $t/2$ and number of samples $n/2$.*/

$\overline{\mathbf{w}} \leftarrow$ Minibatch-TailAveraging-SGD$(\mathbf{w}_s, \gamma, n/2t, t/2, n/2)$

Output: $\overline{\mathbf{w}}$

batch gradient descent, while being a streaming algorithm that does not require storing the entire dataset in memory. Following this, we present a non-asymptotic bound for parameter mixing/model averaging, which is a communication efficient parallelization scheme that has favorable properties when the estimation error (i.e. variance) is the dominating term of the excess risk.

(Nearly) Matching the depth of Batch Gradient Descent: The result of theorem 1 establishes a scalar generalization error bound of Algorithm 1 for the general mis-specified case of LSR and showed that the depth (number of sequential updates in our algorithm) is decreased to n/b. This section builds upon this result to present a simple and intuitive doubling based streaming algorithm that works in epochs and processes a total of $n/2$ points. In each epoch, the minibatch size is increased by a factor of 2 while applying Algorithm 1 (with no tail-averaging) with twice as many samples as the previous epoch. After running over $n/2$ samples using this epoch based approach, we run Algorithm 1 (with tail-averaging) with the remaining $n/2$ points. Intuitively, each of the epoch decays the bias of the previous epoch linearly and halves the statistical error (owing to doubling of mini-batch sizes). The final tail-averaging phase ensures that the variance is small.

The next theorem formalizes this intuition and shows Algorithm 2 improves the depth exponentially from n/b_{thresh} to $\mathcal{O}\left(\kappa \log(d\kappa) \log(n\{L(\mathbf{w}_0) - L(\mathbf{w}^*)\}/\widehat{\sigma^2_{\text{MLE}}})\right)$ in the presence of an error oracle that provides us with the initial excess risk $L(\mathbf{w}_0) - L(\mathbf{w}^*)$ and the noise level $\widehat{\sigma^2_{\text{MLE}}}$.

Theorem 5 *Consider the general mis-specified case of LSR. Suppose in Algorithm 2, we use initial batchsize of $b = b_{thresh}$, stepsize $\gamma = \frac{\gamma_{b,max}}{2}$ and number of iterations in each epoch being $t \geq 24\kappa \log(\kappa)$, we obtain the following excess risk bound on $\overline{\mathbf{w}}$:*

$$\mathbb{E}\left[L(\overline{\mathbf{w}})\right] - L(\mathbf{w}^*) \leq \left(\frac{2bt}{n}\right)^{\frac{t}{12\kappa \log(\kappa)}} \cdot \left(L(\mathbf{w}_0) - L(\mathbf{w}^*)\right) + 80\frac{\widehat{\sigma^2_{MLE}}}{n}.$$

Remarks: The final error again has two parts: the bias term that depends on the initial error $L(\mathbf{w}_0) - L(\mathbf{w}^*)$ and the variance term that depends on the statistical noise $\widehat{\sigma^2_{\text{MLE}}}$. Note that the variance error decays at a rate of $\mathcal{O}\left(\widehat{\sigma^2_{\text{MLE}}}/n\right)$ which is minimax optimal up to constant factors.

Algorithm 2 decays the bias at a superpolynomial rate by choosing t large enough. If Algorithm 2 has access to an initial error oracle that provides $L(\mathbf{w}_0) - L(\mathbf{w}^*)$ and $\widehat{\sigma^2_{\text{MLE}}}$, we can run

Algorithm 2 with a batch size b_{thresh} until the excess risk drops to the noise level $\widehat{\sigma^2_{\text{MLE}}}$ and subsequently begin doubling the batch size. Such an algorithm indeed gives geometric convergence with a generalization error bound as:

$$\mathbb{E}\left[L(\overline{\mathbf{w}})\right] - L(\mathbf{w}^*) \leq \exp\left(-(\frac{n\lambda_{\min}}{R^2 \cdot \log(\kappa)}) \cdot \frac{1}{\rho_{\text{m}}}\right)\{L(\mathbf{w}_0) - L(\mathbf{w}^*)\} + 80\frac{\widehat{\sigma^2_{\text{MLE}}}}{n},$$

with a depth of $\mathcal{O}\left(\kappa \log(d\kappa) \log \frac{n\{L(\mathbf{w}_0)-L(\mathbf{w}^*)\}}{\widehat{\sigma^2_{\text{MLE}}}}\right)$. The proof of this claim follows relatively straightforwardly from the proof of Theorem 5. We note that this depth nearly matches (up to $\log$ factors), the depth of standard offline gradient descent despite being a streaming algorithm. This algorithm (aside from tail-averaging in the final epoch) resembles empirically effective schemes proposed in the context of training deep models (Smith et al., 2017).

Parameter Mixing/Model-Averaging: We consider a communication efficient method for distributed optimization which involves running mini-batch tail-averaged SGD independently on P separate machines (each containing their own independent samples) and averaging the resulting solution estimates. This is a well studied scheme for distributed optimization (Mann et al., 2009; Zinkevich et al., 2011; Rosenblatt and Nadler, 2014; Zhang et al., 2015). As mentioned in Rosenblatt and Nadler (2014), these schemes do not appear to offer improvements in the bias error while offering near linear parallelization speedups on the variance. We provide here a non-asymptotic characterization of the behavior of model averaging for the general mis-specified LSR problem.

Theorem 6 *Consider running Algorithm (1), i.e., mini-batch tail-averaged SGD (for the mis-specified LSR problem (1)) independently in P machines, each of which contains N/P samples. Let algorithm (1) be run with a batch size b, learning rate $\gamma \leq \gamma_{b,max}/2$, tail-averaging begun after $s-$iterations, and let each of these machines output $\{\overline{\mathbf{w}}_i\}_{i=1}^P$. The excess risk of the model-averaged estimator $\overline{\mathbf{w}} = \frac{1}{P}\sum_{i=1}^P \overline{\mathbf{w}}_i$ is upper bounded as:*

$$\mathbb{E}\left[L(\overline{\mathbf{w}})\right] - L(\mathbf{w}^*) \leq \frac{(1-\gamma\mu)^s}{\gamma^2\mu^2\left(\frac{n}{P\cdot b} - s\right)^2} \cdot \frac{2 + (P-1)(1-\gamma\mu)^s}{P} \cdot \left(L(\mathbf{w}_0) - L(\mathbf{w}^*)\right) + 4 \cdot \frac{\widehat{\sigma^2_{MLE}}}{P \cdot b \cdot \left(\frac{n}{P\cdot b} - s\right)}.$$

In particular, with $\gamma = \gamma_{b,max}/2$, we have the following excess risk bound:

$$\mathbb{E}\left[L(\overline{\mathbf{w}})\right] - L(\mathbf{w}^*) \leq \exp\left(-\frac{s}{\kappa_b}\right) \cdot \frac{\kappa_b^2}{\left(\frac{n}{P\cdot b} - s\right)^2} \cdot \frac{2 + (P-1)\cdot\exp(-s/\kappa_b)}{P} \cdot \left(L(\mathbf{w}_0) - L(\mathbf{w}^*)\right) + 4 \cdot \frac{\widehat{\sigma^2_{MLE}}}{P \cdot b \cdot \left(\frac{n}{P\cdot b} - s\right)}.$$

Remarks: We note that during the iterate-averaged phase (i.e. $t > s$), there is no reduction of the bias, whereas, during the (initial) unaveraged iterations, once $s > \kappa_b \log(P)$, we achieve linear speedups on the bias. We note that model averaging offers linear parallelization speedups on the variance error. Furthermore, when the bias reduces to the noise level, model averaging offers linear parallelization speedups on the overall excess risk. Note that if $s = c \cdot n/(P \cdot b)$, with $c < 1$, then the excess risk is minimax optimal. Finally, we note that the theorem can be generalized in a straightforward manner to the situation when each machine has different number of examples.

4. Proof Outline

We present here the framework for obtaining the results described in this paper; the framework has been introduced in the work of Défossez and Bach (2015). Towards this purpose, we begin by introducing some notations. We begin by defining the centered estimate $\boldsymbol{\eta}_t$ as:

$$\boldsymbol{\eta}_t \stackrel{\text{def}}{=} \mathbf{w}_t - \mathbf{w}^*.$$

Mini-batch SGD (with a batch size b) moves $\boldsymbol{\eta}_{t-1}$ to $\boldsymbol{\eta}_t$ using the following update:

$$\boldsymbol{\eta}_t = \left(\mathbf{I} - \frac{\gamma}{b}\cdot\sum_{i=1}^b \mathbf{x}_{ti}\otimes\mathbf{x}_{ti}\right)\boldsymbol{\eta}_{t-1} + \frac{\gamma}{b}\sum_{i=1}^b \epsilon_{ti}\mathbf{x}_{ti} = (\mathbf{I} - \gamma\widehat{\mathbf{H}}_{tb})\boldsymbol{\eta}_{t-1} + \gamma\cdot\boldsymbol{\xi}_{tb},$$

where, $\widehat{\mathbf{H}}_{tb} = \frac{1}{b}\sum_{i=1}^b \mathbf{x}_{ti}\otimes\mathbf{x}_{ti}$ and $\boldsymbol{\xi}_{tb} = \frac{1}{b}\sum_{i=1}^b \epsilon_{ti}\mathbf{x}_{ti}$. Next, the tail-averaged iterate $\bar{\mathbf{x}}_{s,n}$ is associated with its own centered estimate $\bar{\boldsymbol{\eta}}_{s,n} = \frac{1}{n-s}\sum_{i=s+1}^n \boldsymbol{\eta}_i$. The analysis proceeds by tracking the covariance of the centered estimates $\boldsymbol{\eta}_t$, i.e. by tracking $\mathbb{E}\left[\boldsymbol{\eta}_t\otimes\boldsymbol{\eta}_t\right]$.

Bias-Variance decomposition: The main results of this paper are derived by going through the bias-variance decomposition, which is well known in the context of Stochastic Approximation (Bach and Moulines, 2011, 2013; Frostig et al., 2015b). The bias-variance decomposition allows for us to bound the generalization error by analyzing two sub-problems, namely, (i) The *bias* sub-problem, which analyzes the noiseless/realizable (or the consistent linear system) problem, by setting the noise $\epsilon_{ti} = 0\ \forall\ t, i$, $\boldsymbol{\eta}_0^{\text{bias}} = \boldsymbol{\eta}_0$ and (ii) the *variance* sub-problem, which involves starting at the solution, i.e., $\boldsymbol{\eta}_0^{\text{variance}} = 0$ and allowing the noise ϵ_{ti} to drive the resulting process. The corresponding tail-averaged iterates are associated with their centered estimates $\bar{\boldsymbol{\eta}}_{s,n}^{\text{bias}}$ and $\bar{\boldsymbol{\eta}}_{s,n}^{\text{variance}}$ respectively. The bias-variance decomposition for the square loss establishes the following relation:

$$\mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s,n}\otimes\bar{\boldsymbol{\eta}}_{s,n}\right] \preceq 2\cdot\left(\mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s,n}^{\text{bias}}\otimes\bar{\boldsymbol{\eta}}_{s,n}^{\text{bias}}\right] + \mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s,n}^{\text{variance}}\otimes\bar{\boldsymbol{\eta}}_{s,n}^{\text{variance}}\right]\right). \tag{10}$$

Using the bias-variance decomposition, we obtain an estimate of the generalization error as

$$\begin{aligned}\mathbb{E}\left[L(\bar{\mathbf{x}}_{s,n})\right] - L(\mathbf{x}^*) &= \tfrac{1}{2}\cdot\langle\mathbf{H}, \mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s,n}\otimes\bar{\boldsymbol{\eta}}_{s,n}\right]\rangle \\ &\leq \mathrm{Tr}\left(\mathbf{H}\cdot\mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s,n}^{\text{bias}}\otimes\bar{\boldsymbol{\eta}}_{s,n}^{\text{bias}}\right]\right) + \mathrm{Tr}\left(\mathbf{H}\cdot\mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s,n}^{\text{variance}}\otimes\bar{\boldsymbol{\eta}}_{s,n}^{\text{variance}}\right]\right).\end{aligned}$$

We now provide a few lemmas that help us bound the behavior of the bias and variance error.

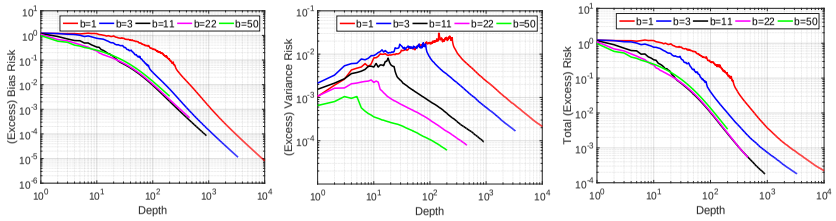
Lemma 7 *With a batch size b, step size $\gamma = \gamma_{b,max}/2$, the centered bias estimate $\boldsymbol{\eta}_t^{bias}$ exhibits the following per step contraction:*

$$\langle\mathbf{I}, \mathbb{E}\left[\boldsymbol{\eta}_t^{bias}\otimes\boldsymbol{\eta}_t^{bias}\right]\rangle \leq c_{\kappa_b}\langle\mathbf{I}, \mathbb{E}\left[\boldsymbol{\eta}_{t-1}^{bias}\otimes\boldsymbol{\eta}_{t-1}^{bias}\right]\rangle,$$

where, $c_{\kappa_b} = 1 - 1/\kappa_b$, where $\kappa_b = \frac{R^2\cdot\rho_m + (b-1)\|\mathbf{H}\|_2}{b\mu}$.

Lemma (7) ensures that the bias decays at a geometric rate during the burn-in iterations when the iterates are not averaged; this rate holds only when the excess risk is larger than the noise level σ^2.

We now turn to bounding the variance error. It turns out that it suffices to understand the behavior of limiting centered variance $\mathbb{E}\left[\boldsymbol{\eta}_\infty^{\text{variance}}\otimes\boldsymbol{\eta}_\infty^{\text{variance}}\right]$.



(a) Bias Risk (b) Variance Risk (c) Total Risk

Figure 1: Effect of increased batch sizes on the Algorithm's generalization error. The variance decreases monotonically with increasing batch size. The bias indicates that the rate of decay increases till the optimal b_{thresh}. With $b = b_{\text{thresh}}$, mini-batch SGD obtains the same generalization error as batchsize 1 using smaller number of iterations (i.e. smaller depth) compared to larger batch sizes.

Lemma 8 *Consider the well-specified case of the streaming LSR problem. With a batch size b, step size $\gamma = \gamma_{b,max}/2$, the limiting centered variance $\boldsymbol{\eta}_\infty^{variance}$ has an expected covariance that is upper bounded in a psd sense as:*

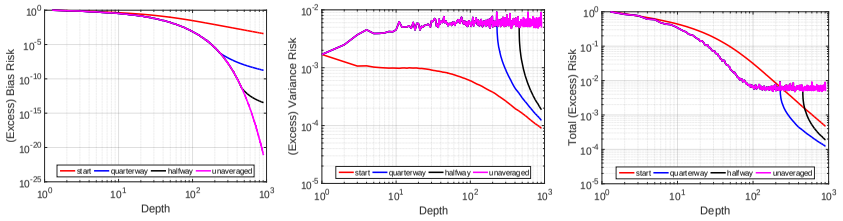
$$\mathbb{E}\left[\boldsymbol{\eta}_\infty^{variance} \otimes \boldsymbol{\eta}_\infty^{variance}\right] \preceq \frac{1}{R^2 + (b-1)\|\mathbf{H}\|_2} \cdot \sigma^2 \cdot \mathbf{I}.$$

Characterizing the behavior of the final iterate is crucial towards obtaining bounds on the behavior of the tail-averaged iterate. In particular, the final iterate having a excess variance risk $\mathcal{O}(\sigma^2)$ (as is the case with lemma (8)) appears crucial towards achieving minimax rates of the averaged iterate.

5. Experimental Simulations

We conduct experiments using a synthetic example to illustrate the implications of our theoretical results on mini-batching and tail-averaging. The data is sampled from a $50-$ dimensional Gaussian with eigenvalues decaying as $\{\frac{1}{k}\}_{k=1}^{50}$ (condition number $\kappa = 50$), and the variance σ^2 of the (additive noise) noise is 0.01. In this case, our estimated batch size according to Theorem 1 is $b_{\text{thresh}} = 11$. Our results are presented by averaging over 100 independent runs of the Algorithm, and each run employs 200κ samples. All plots are log-log with x-axis being the depth, and y-axis the excess risk. For our plots, we assume that each iteration takes constant time for all batch sizes; this is done to present evidence regarding the tightness of our mini-batching characterization limits that yield linear parallelization speedups over SGD with mini-batch size of 1.

We consider the effect of mini-batching (in figure 1) with batch sizes of 1, 3, $b_{\text{thresh}} = 11$, $2 \cdot b_{\text{thresh}} = 22$ and $d = 50$. Averaging begins after observing a fixed number of samples (set as 5κ). We see that the rate of bias decay (figure 1a) increases until reaching a mini-batch size of b_{thresh}, saturating thereafter; this implies we are inefficient in terms of sample size. As expected, the rate of decay of variance (figure 1b) is monotonic as a function of mini-batch size. Finally, the overall error (figure 1c) shows the tightness of our mini-batching characterization: with a batch size of b_{thresh}, we obtain a generalization error that is the *same* as using batch size of 1 with the number of (serial) iterations (i.e. depth) that is an order of magnitude smaller. Subsequently, we note that larger batch sizes worsen generalization error thus depicting the tightness of our characterization of b_{thresh}.



(a) Bias Risk (b) Variance Risk (c) Total Risk

Figure 2: [Zoom in to see detail] Effect of tail-averaging with mini-batch size of $b_{thresh} = 11$.

In the next experiment, we fix batch size $= b_{\text{thresh}}$ and consider the effect of when tail-averaging begins (figure 2). We consider averaging iterates from the start (as prescribed by Défossez and Bach (2015)), after a quarter/half of total number of iterations, and unaveraged SGD as well. We see that the bias (figure 2a) exhibits a geometric decay in the unaveraged phase while switching to an slower $\mathcal{O}(\frac{1}{t^2})$ rate with averaging. The variance (figure 2b) tends to increase and stabilize at $\mathcal{O}(\frac{\sigma^2}{b_{\text{thresh}}})$ in the absence of averaging, while switching to a $\mathcal{O}(\frac{1}{N})$ decay rate when averaging begins. The overall generalization error (figure 2c) shows the superiority of the scheme where averaging after a burn-in period allows the bias to decay towards the noise level at a geometric rate, following which tail-averaging allows us to decay the variance term, providing credence to our theoretical results that tail-averaged SGD allows us to obtain better generalization error as a function of sample size.

6. Concluding Remarks

This paper analyzes several algorithmic primitives often used in practice in conjunction with vanilla SGD for the stochastic approximation problem. In particular, this paper provides a sharp non-asymptotic treatment of (a) mini-batching, (b) tail-averaging, (c) effects of model mismatch, (d) behaviour of the final iterate, (e) highly parallel SGD method based on doubling batch sizes and (f) model-averaging/parameter mixing schemes for the strongly convex streaming LSR problem.

The effect of mini-batching and other algorithmic primitives mentioned above can be understood for a variety of models and/or algorithms. In particular, future directions could include understanding these issues for stochastic approximation with the Logistic Loss (Bach, 2014), streaming PCA (Jain et al., 2016a), and other algorithms such as streaming SVRG (Frostig et al., 2015b).

Acknowledgments

Sham Kakade acknowledges funding from Washington Research Foundation Fund for Innovation in Data-Intensive Discovery and National Science Foundation (NSF) through awards CCF-1703574 and CCF-1740551. Rahul Kidambi thanks James Saunderson for useful discussions on matrix operator theory.

Appendix A. Appendix

We begin with a note on the organization:

- Section A.1 introduces notations necessary for the rest of the appendix.
- Section A.2 derives the mini-batch SGD update and provides the bias-variance decomposition and reasons about its implication in bounding the generalization error.
- Section A.3 provides lemmas that are used to bound the bias error.
- Section A.4 provides lemmas that are used to bound the variance error.
- Section A.5 uses the results of the previous sections to obtain the main results of this paper.

A.1 Notations

We begin by introducing the centered iterate $\boldsymbol{\eta}_t$ i.e.:

$$\boldsymbol{\eta}_t \stackrel{\text{def}}{=} \mathbf{w}_t - \mathbf{w}^*.$$

In a manner similar to $\mathbf{w}_t$, the tail-averaged iterate $\overline{\mathbf{w}}_{t,N}$ is associated with its corresponding centered estimate $\bar{\boldsymbol{\eta}}_{t,N} \stackrel{\text{def}}{=} \overline{\mathbf{w}}_{t,N} - \mathbf{w}^* = \frac{1}{N}\sum_{s=t}^{t+N-1}(\mathbf{w}_s - \mathbf{w}^*) = \frac{1}{N}\sum_{s=t}^{t+N-1}\boldsymbol{\eta}_s$. Next, let $\boldsymbol{\Phi}_t$ denote the expected covariance of the centered estimate $\boldsymbol{\eta}_t$, i.e.

$$\boldsymbol{\Phi}_t \stackrel{\text{def}}{=} \mathbb{E}\left[\boldsymbol{\eta}_t \otimes \boldsymbol{\eta}_t\right],$$

and in a similar way as the final iterate $\mathbf{w}_t$, the tail-averaged estimate $\overline{\mathbf{w}}_{t,N}$ is associated with its expected covariance, i.e. $\bar{\boldsymbol{\Phi}}_{t,N} \stackrel{\text{def}}{=} \mathbb{E}\left[\bar{\boldsymbol{\eta}}_{t,N} \otimes \bar{\boldsymbol{\eta}}_{t,N}\right]$.

A.2 Mini-Batch Tail-Averaged SGD: Bias-Variance Decomposition

In section A.2.1, we derive the basic recursion governing the evolution of the iterates $\mathbf{w}_t$ and the tail-averaged iterate $\overline{\mathbf{w}}_{s+1,N}$. In section A.2.2 we provide the bias-variance decomposition of the final iterate. In section A.2.3, we provide the bias-variance decomposition of the tail-averaged iterate.

A.2.1 THE BASIC RECURSION

At each iteration t of Algorithm 1, we are provided with b fresh samples $\{(\mathbf{x}_{ti}, y_{ti})\}_{i=1}^b$ drawn i.i.d. from the distribution $\mathcal{D}$. We start by recounting the mini-batch gradient descent update rule that allows us to move from iterate $\mathbf{w}_{t-1}$ to $\mathbf{w}_t$:

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \frac{\gamma}{b}\sum_{i=1}^{b}(\langle \mathbf{w}_{t-1}, \mathbf{x}_{ti}\rangle - y_{ti})\mathbf{x}_{ti},$$

where, $0 < \gamma < \gamma_{b,\max}$ is the constant step size that is set to a value less than the maximum allowed learning rate $\gamma_{b,\max}$. We also recount the definition of $\overline{\mathbf{w}}_{t,N}$ which is the iterate obtained by averaging for N iterations starting from the t^{th} iteration, i.e.,

$$\overline{\mathbf{w}}_{t,N} = \frac{1}{N}\sum_{s=t}^{t+N-1}\mathbf{w}_s.$$

Let us first denote the residual error term by $\epsilon_i = y_i - \langle \mathbf{w}^*, \mathbf{x}_i\rangle$. By the first order optimality conditions of $\mathbf{w}^*$, we observe that ϵ and $\mathbf{x}$ are orthogonal, i.e, $\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\epsilon \cdot \mathbf{x}] = 0$. For any estimate $\mathbf{w}$, the excess risk/generalization error can be written as:

$$L(\mathbf{w}) - L(\mathbf{w}^*) = \frac{1}{2}\operatorname{Tr}\left(\mathbf{H}\cdot(\boldsymbol{\eta}\otimes\boldsymbol{\eta})\right), \text{ with } \boldsymbol{\eta} = \mathbf{w} - \mathbf{w}^*. \tag{11}$$

We now write out the main recursion governing the mini-batch SGD updates in terms of $\boldsymbol{\eta}_\cdot$:

$$\begin{aligned}
\boldsymbol{\eta}_t &= \Big(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^{b}\mathbf{x}_{ti}\otimes\mathbf{x}_{ti}\Big)\boldsymbol{\eta}_{t-1} + \frac{\gamma}{b}\sum_{i=1}^{b}\epsilon_{ti}\mathbf{x}_{ti} \\
&= \Big(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^{b}\mathbf{x}_{ti}\otimes\mathbf{x}_{ti}\Big)\boldsymbol{\eta}_{t-1} + \frac{\gamma}{b}\sum_{i=1}^{b}\boldsymbol{\xi}_{ti} \\
&= \mathbf{P}_{tb}\boldsymbol{\eta}_{t-1} + \gamma\boldsymbol{\zeta}_{tb},
\end{aligned} \tag{12}$$

where, $\mathbf{P}_{tb} \stackrel{\text{def}}{=} \Big(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^{b}\mathbf{x}_{ti}\otimes\mathbf{x}_{ti}\Big)$ and $\boldsymbol{\zeta}_{tb} \stackrel{\text{def}}{=} \frac{1}{b}\sum_{i=1}^{b}\boldsymbol{\xi}_{ti} = \frac{1}{b}\sum_{i=1}^{b}\epsilon_{ti}\mathbf{x}_{ti}$. Equation 12 automatically brings out the "operator" view of analyzing the (expected) covariance of the centered estimate $\boldsymbol{\Phi}_t = \mathbb{E}\left[\boldsymbol{\eta}_t\otimes\boldsymbol{\eta}_t\right]$ to provide an estimate of the generalization error. We now note the following about the covariance of $\boldsymbol{\zeta}_{tb}$:

$$\begin{aligned}
\mathbb{E}[\boldsymbol{\zeta}_{tb}\otimes\boldsymbol{\zeta}_{t'b}] &= \frac{1}{b^2}\sum_{i,j}\mathbb{E}[\boldsymbol{\xi}_{ti}\otimes\boldsymbol{\xi}_{t'j}] \\
&= \Big[\frac{1}{b^2}\sum_{i=1}^{b}\mathbb{E}[\boldsymbol{\xi}_{ti}\otimes\boldsymbol{\xi}_{ti}]\Big]\mathbb{1}[t=t'] = \frac{1}{b}\boldsymbol{\Sigma}\ \mathbb{1}[t=t'],
\end{aligned} \tag{13}$$

where, $\mathbb{1}[.]$ is the indicator function, and equals 1 if the argument inside $[.]$ is true and 0 otherwise. We note that the expectation of the cross terms in equation 13 is zero owing to independence of the samples $\{\mathbf{x}_{ti}, y_{ti}\}_{i=1}^b$ as well as between $\{\mathbf{x}_{ti}, y_{ti}\}_{i=1}^b$, $\{\mathbf{x}_{t'i}, y_{t'i}\}_{i=1}^b$ $\forall\, t \neq t'$ and owing to the first order optimality conditions. Owing to the invariance of $\boldsymbol{\zeta}_{tb}$ on the iteration t, context permitting, we sometimes drop the iteration index t from $\boldsymbol{\zeta}_{tb}$ and simply refer to it as $\boldsymbol{\zeta}_b$.

Next we expand out the recurrence (12). Let $\mathbf{Q}_{j,t} = (\prod_{k=j}^{t}\mathbf{P}_{kb})^T$ with the convention that $\mathbf{Q}_{t',t} = \mathbf{I}\ \forall\, t' > t$. With this notation we have:

$$\begin{aligned}
\boldsymbol{\eta}_t &= \mathbf{P}_{tb}\boldsymbol{\eta}_{t-1} + \gamma\boldsymbol{\zeta}_{tb} \\
&= \mathbf{P}_{tb}\mathbf{P}_{t-1,b}\ldots\mathbf{P}_{1,b}\boldsymbol{\eta}_0 + \gamma\sum_{j=0}^{t-1}\{\mathbf{P}_{tb}\ldots\mathbf{P}_{t-j+1,b}\}\boldsymbol{\zeta}_{t-j,b} \\
&= \mathbf{Q}_{1,t}\boldsymbol{\eta}_0 + \gamma\sum_{j=0}^{t-1}\mathbf{Q}_{t-j+1,t}\boldsymbol{\zeta}_{t-j,b} \\
&= \mathbf{Q}_{1,t}\boldsymbol{\eta}_0 + \gamma\sum_{j=1}^{t}\mathbf{Q}_{j+1,t}\boldsymbol{\zeta}_{j,b} \\
&= \boldsymbol{\eta}_t^{\text{bias}} + \boldsymbol{\eta}_t^{\text{variance}},
\end{aligned} \tag{14}$$

where, we note that

$$\boldsymbol{\eta}_t^{\text{bias}} \stackrel{\text{def}}{=} \mathbf{Q}_{1,t}\boldsymbol{\eta}_0 \tag{15}$$

relates to understanding the behavior of SGD on the noiseless problem (i.e. $\boldsymbol{\zeta}_{\cdot,\cdot} = 0$ a.s.) and aims to quantify the dependence on the initial conditions. Further,

$$\boldsymbol{\eta}_t^{\text{variance}} \stackrel{\text{def}}{=} \gamma \sum_{j=1}^{t} \mathbf{Q}_{j+1,t}\boldsymbol{\zeta}_{j,b} \tag{16}$$

relates to the behavior of SGD when begun at the solution (i.e. $\boldsymbol{\eta}_0 = 0$) and allowing the noise $\boldsymbol{\zeta}_{\cdot,\cdot}$ to drive the process.

Furthermore, considering the tail-averaged iterate obtained by averaging the iterates of the SGD procedure for N iterations starting from a certain number of iterations "s", i.e., we examine the quantity $\bar{\boldsymbol{\eta}}_{s+1,N} = \overline{\mathbf{w}}_{s+1,N} - \mathbf{w}^*$, where $\overline{\mathbf{w}}_{s+1,N} = \frac{1}{N}\sum_{t=s+1}^{s+N} \mathbf{w}_t$. We write out the expression for $\bar{\boldsymbol{\eta}}_{s+1,N}$ starting out from equation 14:

$$\begin{aligned}
\bar{\boldsymbol{\eta}}_{s+1,N} &= \frac{1}{N}\sum_{t=s+1}^{s+N} \boldsymbol{\eta}_t \\
&= \frac{1}{N}\sum_{t=s+1}^{s+N} \left(\boldsymbol{\eta}_t^{\text{bias}} + \boldsymbol{\eta}_t^{\text{variance}}\right) \qquad \text{(from equation 14)} \\
&= \bar{\boldsymbol{\eta}}_{s+1,N}^{\text{bias}} + \bar{\boldsymbol{\eta}}_{s+1,N}^{\text{variance}}.
\end{aligned} \tag{17}$$

A.2.2 THE FINAL ITERATE: BIAS-VARIANCE DECOMPOSITION

The behavior of the final iterate is considered to be of great practical interest and we hope to shed light on the behavior of this final iterate and the tradeoffs between the learning rate and batch size. Since the generalization error of any iterate $\mathbf{w}_N$ obtained by running mini-batch SGD with a batch size b for a total of N iterations can be estimated by tracking $\mathbb{E}\left[\boldsymbol{\eta}_N \otimes \boldsymbol{\eta}_N\right]$, where, $\boldsymbol{\eta}_N = \mathbf{w}_N - \mathbf{w}^*$, we provide a simple psd upper bound on the outer product of interest, i.e.:

$$\begin{aligned}
\mathbb{E}\left[\boldsymbol{\eta}_N \otimes \boldsymbol{\eta}_N\right] &= \mathbb{E}\left[\left(\boldsymbol{\eta}_N^{\text{bias}} + \boldsymbol{\eta}_N^{\text{variance}}\right) \otimes \left(\boldsymbol{\eta}_N^{\text{bias}} + \boldsymbol{\eta}_N^{\text{variance}}\right)\right] \qquad \text{(by substituting equation 14)} \\
&\preceq 2 \cdot \left(\mathbb{E}\left[\left(\boldsymbol{\eta}_N^{\text{bias}} \otimes \boldsymbol{\eta}_N^{\text{bias}}\right)\right] + \mathbb{E}\left[\left(\boldsymbol{\eta}_N^{\text{variance}} \otimes \boldsymbol{\eta}_N^{\text{variance}}\right)\right]\right).
\end{aligned}$$

Using this expression, we now write out the expression for the excess risk of the final iterate:

$$\begin{aligned}
\mathbb{E}\left[L(\mathbf{w}_N)\right] - L(\mathbf{w}^*) &= \frac{1}{2}\langle \mathbf{H}, \mathbb{E}\left[\boldsymbol{\eta}_N \otimes \boldsymbol{\eta}_N\right]\rangle \\
&\leq \frac{1}{2}\langle \mathbf{H}, 2 \cdot \left(\mathbb{E}\left[\boldsymbol{\eta}_N^{\text{bias}} \otimes \boldsymbol{\eta}_N^{\text{bias}}\right] + \mathbb{E}\left[\boldsymbol{\eta}_N^{\text{variance}} \otimes \boldsymbol{\eta}_N^{\text{variance}}\right]\right)\rangle \\
&\leq 2 \cdot \left(\frac{1}{2}\langle \mathbf{H}, \mathbb{E}\left[\boldsymbol{\eta}_N^{\text{bias}} \otimes \boldsymbol{\eta}_N^{\text{bias}}\right]\rangle + \frac{1}{2}\langle \mathbf{H}, \mathbb{E}\left[\boldsymbol{\eta}_N^{\text{variance}} \otimes \boldsymbol{\eta}_N^{\text{variance}}\right]\rangle\right) \\
&= 2 \cdot \left(\left(\mathbb{E}\left[L(\mathbf{w}_N^{\text{bias}})\right] - L(\mathbf{w}^*)\right) + \left(\mathbb{E}\left[L(\mathbf{w}_N^{\text{variance}})\right] - L(\mathbf{w}^*)\right)\right).
\end{aligned} \tag{18}$$

A.2.3 THE TAIL-AVERAGED ITERATE: BIAS-VARIANCE DECOMPOSITION

Now, considering the fact that the excess risk/generalization error (equation 11) involves tracking $\mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N} \otimes \bar{\boldsymbol{\eta}}_{s+1,N}\right]$, we see that the quantity of interest can be bounded by considering the behavior of SGD on bias and variance sub-problem. In particular, writing out the outerproduct of equation 17, we see the following inequality holds through a straightforward application of Cauchy-Shwartz inequality:

$$\mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N} \otimes \bar{\boldsymbol{\eta}}_{s+1,N}\right] \preceq 2 \cdot \left(\mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N}^{\text{bias}} \otimes \bar{\boldsymbol{\eta}}_{s+1,N}^{\text{bias}}\right] + \mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N}^{\text{variance}} \otimes \bar{\boldsymbol{\eta}}_{s+1,N}^{\text{variance}}\right]\right). \tag{19}$$

The above equation is referred to as the bias-variance decomposition and is well known from previous work on Stochastic Approximation (Bach and Moulines, 2013; Frostig et al., 2015b; Défossez and Bach, 2015). This implies that an upper bound on the generalization error (equation 11) is:

$$\begin{aligned}
L(\overline{\mathbf{w}}_{s+1,N}) - L(\mathbf{w}^*) &= \frac{1}{2}\langle \mathbf{H}, \mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N} \otimes \bar{\boldsymbol{\eta}}_{s+1,N}\right]\rangle \\
&\leq \langle \mathbf{H}, \mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N}^{\text{bias}} \otimes \bar{\boldsymbol{\eta}}_{s+1,N}^{\text{bias}}\right]\rangle + \langle \mathbf{H}, \mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N}^{\text{variance}} \otimes \bar{\boldsymbol{\eta}}_{s+1,N}^{\text{variance}}\right]\rangle.
\end{aligned} \tag{20}$$

Here, we adopt the proof approach of Jain et al. (2017a). In particular, Jain et al. (2017a) provide a clean way to simplify the expression corresponding to the tail-averaged iterate. Let us consider $\mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N} \otimes \bar{\boldsymbol{\eta}}_{s+1,N}\right]$ and simplify the resulting expression: in particular,

$$\begin{aligned}
\mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N}\bar{\boldsymbol{\eta}}_{s+1,N}^\top\right] &= \frac{1}{N^2}\sum_{l=s+1}^{s+N}\sum_{k=s+1}^{s+N} \mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_k\right] \\
&= \frac{1}{N^2} \cdot \left(\sum_{l \geq k} \mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_k\right] + \sum_{l<k} \mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_k\right]\right) \\
&\preceq \frac{1}{N^2} \cdot \left(\sum_{l \geq k} \mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_k\right] + \sum_{l \leq k} \mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_k\right]\right) \qquad (*) \\
&= \frac{1}{N^2} \cdot \left(\sum_{l \geq k} (\mathbf{I} - \gamma\mathbf{H})^{l-k}\mathbb{E}\left[\boldsymbol{\eta}_k \otimes \boldsymbol{\eta}_k\right] + \sum_{l \leq k} \mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right](\mathbf{I} - \gamma\mathbf{H})^{k-l}\right) \quad (**) \\
&= \frac{1}{N^2} \cdot \sum_{l \leq k}\left(\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right](\mathbf{I} - \gamma\mathbf{H})^{k-l} + (\mathbf{I} - \gamma\mathbf{H})^{k-l}\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right]\right) \\
&= \frac{1}{N^2} \cdot \sum_{l=s+1}^{s+N}\sum_{k=l}^{s+N}\left(\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right](\mathbf{I} - \gamma\mathbf{H})^{k-l} + (\mathbf{I} - \gamma\mathbf{H})^{k-l}\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right]\right) \\
&= \frac{1}{N^2} \cdot \sum_{l=s+1}^{s+N}\sum_{k=l}^{\infty}\left(\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right](\mathbf{I} - \gamma\mathbf{H})^{k-l} + (\mathbf{I} - \gamma\mathbf{H})^{k-l}\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right]\right) \\
&\quad - \frac{1}{N^2} \cdot \sum_{l=s+1}^{s+N}\sum_{k=s+N+1}^{\infty}\left(\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right](\mathbf{I} - \gamma\mathbf{H})^{k-l} + (\mathbf{I} - \gamma\mathbf{H})^{k-l}\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right]\right) \\
&= \frac{1}{N^2} \cdot \sum_{l=s+1}^{s+N}\left(\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right](\gamma\mathbf{H})^{-1} + (\gamma\mathbf{H})^{-1}\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right]\right)
\end{aligned}$$

$$- \frac{1}{N^2} \sum_{l=s+1}^{s+N} \sum_{k=s+N+1}^{\infty} \left(\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right] \left(\mathbf{I} - \gamma \mathbf{H}\right)^{k-l} + \left(\mathbf{I} - \gamma \mathbf{H}\right)^{k-l} \mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right] \right) \tag{21}$$
$$(***),$$

where, $(*)$ is a valid PSD upper bound as we add and subtract the diagonal terms $\{\mathbb{E}\left[\boldsymbol{\eta}_k \boldsymbol{\eta}_k^\top\right]\}_{k=s+1}^{s+N}$. $(**)$ follows because of the following (assume $l > k$; the other case follows similarly):

$$\begin{aligned}
\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_k\right] &= \mathbb{E}\left[\left(\mathbf{P}_{lb}\boldsymbol{\eta}_{l-1} + \gamma \boldsymbol{\zeta}_{lb}\right) \otimes \boldsymbol{\eta}_k\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\left(\mathbf{P}_{lb}\boldsymbol{\eta}_{l-1} + \gamma \boldsymbol{\zeta}_{lb}\right) \otimes \boldsymbol{\eta}_k | \mathcal{F}_{l-1}\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\left(\mathbf{P}_{lb}\boldsymbol{\eta}_{l-1} + \gamma \boldsymbol{\zeta}_{lb}\right) | \mathcal{F}_{l-1}\right] \otimes \boldsymbol{\eta}_k\right] \\
&= \left(\mathbf{I} - \gamma \mathbf{H}\right) \mathbb{E}\left[\boldsymbol{\eta}_{l-1} \otimes \boldsymbol{\eta}_k\right],
\end{aligned}$$

where, the final equation follows since $\mathbb{E}\left[\mathbf{P}_{lb} | \mathcal{F}_{l-1}\right] = \mathbb{E}\left[\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^{b} \mathbf{x}_{li} \otimes \mathbf{x}_{li} | \mathcal{F}_{l-1}\right] = \mathbf{I} - \gamma \mathbf{H}$ and $\mathbb{E}\left[\boldsymbol{\zeta}_{lb} | \mathcal{F}_{l-1}\right] = 0$ from first order optimality conditions. Recursing over l yields the result. $(***)$ follows from summing a (convergent) geometric series.

This implies that the excess risk corresponding to the bias/variance term can be obtained from equation 21 by taking an inner product with $\mathbf{H}$, i.e.:

$$\begin{aligned}
\left\langle \mathbf{H}, \mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N} \otimes \bar{\boldsymbol{\eta}}_{s+1,N}\right] \right\rangle &\leq \frac{1}{N^2} \cdot \sum_{l=s+1}^{s+N} \left(\left\langle \mathbf{H}, \mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right] \left(\gamma \mathbf{H}\right)^{-1} + \left(\gamma \mathbf{H}\right)^{-1} \mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right] \right\rangle \right) \\
&\quad - \frac{1}{N^2} \cdot \sum_{l=s+1}^{s+N} \sum_{k=s+N+1}^{\infty} \left(\left\langle \mathbf{H}, \mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right] \left(\mathbf{I} - \gamma \mathbf{H}\right)^{k-l} \right.\right. \\
&\qquad \left.\left. + \left(\mathbf{I} - \gamma \mathbf{H}\right)^{k-l} \mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right] \right\rangle \right) \\
&\leq \frac{1}{N^2} \cdot \sum_{l=s+1}^{s+N} \left(\left\langle \mathbf{H}, \mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right] \left(\gamma \mathbf{H}\right)^{-1} + \left(\gamma \mathbf{H}\right)^{-1} \mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right] \right\rangle \right) \\
&= \frac{2}{\gamma N^2} \cdot \sum_{l=s+1}^{s+N} \operatorname{Tr}\left(\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right] \right).
\end{aligned} \tag{22}$$

The upper bound on the final line follows because each term within the summation in the second line is negative owing to the following argument. Consider say,

$$\begin{aligned}
&\left\langle \mathbf{H}, \mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right] \left(\mathbf{I} - \gamma \mathbf{H}\right)^{k-l} + \left(\mathbf{I} - \gamma \mathbf{H}\right)^{k-l} \mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right] \right\rangle \\
&= 2 \operatorname{Tr}\left[\mathbf{H}\left(\mathbf{I} - \gamma \mathbf{H}\right)^{k-l} \mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right]\right] \geq 0.
\end{aligned}$$

Note that $\mathbf{H}$ and $(\mathbf{I} - \gamma \mathbf{H})$ commute and both are psd, implying that $\mathbf{H}(\mathbf{I} - \gamma \mathbf{H})^{k-l}$ is PSD. Finally, the trace of the product of two PSD matrices is positive with $\mathbf{H}(\mathbf{I} - \gamma \mathbf{H})^{k-l}$ being one of these PSD matrices and $\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right]$ being the other, thus yielding the claimed bound in equation 22.

This implies that the overall error (through equation 11) can be upperbounded as:

$$\mathbb{E}\left[L(\overline{\mathbf{w}}_{s+1,N})\right] - L(\mathbf{w}^*) = \frac{1}{2} \cdot \left\langle \mathbf{H}, \mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N} \otimes \bar{\boldsymbol{\eta}}_{s+1,N}\right] \right\rangle$$

$$\begin{aligned}
&\leq \frac{1}{\gamma N^2} \sum_{l=s+1}^{s+N} \operatorname{Tr}\left(\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right] \right) \\
&\leq \frac{2}{\gamma N^2} \cdot \sum_{l=s+1}^{s+N} \left(\operatorname{Tr}\left(\mathbb{E}\left[\boldsymbol{\eta}_l^{\text{bias}} \otimes \boldsymbol{\eta}_l^{\text{bias}}\right] \right) + \operatorname{Tr}\left(\mathbb{E}\left[\boldsymbol{\eta}_l^{\text{variance}} \otimes \boldsymbol{\eta}_l^{\text{variance}}\right] \right) \right),
\end{aligned} \tag{23}$$

where the final line follows from equation 19. We will now bound each of these terms to precisely characterize the excess risk of mini-batch tail-averaged SGD. We refer to the bias error of the tail-averaged iterate as the following:

$$\mathbb{E}\left[L(\overline{\mathbf{w}}_{s+1,N}^{\text{bias}})\right] - L(\mathbf{w}^*) \stackrel{\text{def}}{=} \frac{2}{\gamma N^2} \sum_{l=s+1}^{s+N} \operatorname{Tr}\left(\mathbb{E}\left[\boldsymbol{\eta}_l^{\text{bias}} \otimes \boldsymbol{\eta}_l^{\text{bias}}\right] \right). \tag{24}$$

Similarly, we refer to the variance error of the tail-averaged iterate as the following:

$$\mathbb{E}\left[L(\overline{\mathbf{w}}_{s+1,N}^{\text{variance}})\right] - L(\mathbf{w}^*) \stackrel{\text{def}}{=} \frac{2}{\gamma N^2} \sum_{l=s+1}^{s+N} \operatorname{Tr}\left(\mathbb{E}\left[\boldsymbol{\eta}_l^{\text{variance}} \otimes \boldsymbol{\eta}_l^{\text{variance}}\right] \right). \tag{25}$$

A.3 Lemmas For Bounding The Bias Error

Lemma 9 *With $\gamma \leq \frac{\gamma_{b,max}}{2} = \frac{b}{R^2 \cdot \rho_m + (b-1)\|\mathbf{H}\|_2}$, the following bound holds:*

$$\left\| \mathbb{E}\left[\left(\mathbf{I} - \frac{\gamma}{b}\sum_{j=1}^{b} \mathbf{x}_{li} \otimes \mathbf{x}_{li}\right)\left(\mathbf{I} - \frac{\gamma}{b}\sum_{j=1}^{b} \mathbf{x}_{li} \otimes \mathbf{x}_{li}\right) \right] \right\|_2 \leq 1 - \gamma\mu.$$

Proof This lemma generalizes one appearing in Jain et al. (2017a) to the mini-batch size b case. Denote by $\mathbf{U}$ the matrix of interest and consider the following:

$$\begin{aligned}
\mathbf{U} &= \mathbb{E}\left[\left(\mathbf{I} - \frac{\gamma}{b}\sum_{j=1}^{b} \mathbf{x}_{li} \otimes \mathbf{x}_{li}\right)\left(\mathbf{I} - \frac{\gamma}{b}\sum_{j=1}^{b} \mathbf{x}_{li} \otimes \mathbf{x}_{li}\right) \right] \\
&= \mathbf{I} - \gamma\mathbf{H} - \gamma\mathbf{H} + \left(\frac{\gamma}{b}\right)^2 \cdot \left(b\mathbb{E}\left[\|\mathbf{x}\|^2 \mathbf{x}\mathbf{x}^\top\right] + b(b-1)\mathbf{H}^2 \right) \\
&\preceq \mathbf{I} - 2\gamma\mathbf{H} + \frac{\gamma^2}{b} \cdot \left(R^2\mathbf{H} + (b-1)\|\mathbf{H}\|_2\right)\mathbf{H} \\
&= \mathbf{I} - \gamma\mathbf{H},
\end{aligned}$$

from which a spectral norm bound implied by the lemma naturally follows. ■

Lemma 10 *For any learning rate $\gamma \leq \gamma_{b,max}/2$, the bias error of the tail-averaged iterate $\overline{\mathbf{w}}_{s+1,N}^{bias}$ is upper bounded as:*

$$\mathbb{E}\left[L(\overline{\mathbf{w}}_{s+1,N}^{bias})\right] - L(\mathbf{w}^*) \leq \frac{2}{\gamma^2 N^2 \mu^2}(1 - \gamma\mu)^{s+1} \cdot \left(L(\mathbf{w}_0) - L(\mathbf{w}^*)\right).$$

Proof Before writing out the proof of the bound in the lemma, we require to bound the per step contraction properties of an SGD update in the case of the bias error (i.e. $\boldsymbol{\zeta}_\cdot = 0$):

$$\begin{aligned}
\mathbb{E}\left[\|\boldsymbol{\eta}_l\|^2\right] &= \mathbb{E}\left[\boldsymbol{\eta}_{l-1}^\top (\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^b \mathbf{x}_{li}\otimes\mathbf{x}_{li})(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^b \mathbf{x}_{li}\otimes\mathbf{x}_{li})\boldsymbol{\eta}_{l-1}\right] \\
&= \mathbb{E}\left[\boldsymbol{\eta}_{l-1}^\top \mathbb{E}\left[(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^b \mathbf{x}_{li}\otimes\mathbf{x}_{li})(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^b \mathbf{x}_{li}\otimes\mathbf{x}_{li})\middle|\mathcal{F}_{l-1}\right]\boldsymbol{\eta}_{l-1}\right] \\
&\leq (1-\gamma\mu)\mathbb{E}\left[\|\boldsymbol{\eta}_{l-1}\|^2\right] \quad \text{(using lemma 9)}.
\end{aligned}$$

This implies that a recursive application of the above bound yields $\mathbb{E}\left[\|\boldsymbol{\eta}_l\|^2\right] \leq (1-\gamma\mu)^l\mathbb{E}\left[\|\boldsymbol{\eta}_0\|^2\right]$.

Next, we consider the bias error from equation 24:

$$\begin{aligned}
\mathbb{E}\left[L(\overline{\mathbf{w}}^{\text{bias}}_{s+1,N})\right] - L(\mathbf{w}^*) &= \frac{2}{\gamma N^2}\sum_{t=s+1}^{s+N}\mathbb{E}\left[\|\boldsymbol{\eta}_t\|^2\right] \\
&\leq \frac{2}{\gamma N^2}\sum_{t=s+1}^{\infty}\mathbb{E}\left[\|\boldsymbol{\eta}_t\|^2\right] \\
&\leq \frac{2}{\gamma N^2}\sum_{t=s+1}^{\infty}(1-\gamma\mu)^t\|\boldsymbol{\eta}_0\|^2 \\
&= \frac{2}{\gamma N^2}(\gamma\mu)^{-1}(1-\gamma\mu)^{s+1}\|\boldsymbol{\eta}_0\|^2 \\
&= \frac{2}{\gamma^2\mu N^2}(1-\gamma\mu)^{s+1}\|\boldsymbol{\eta}_0\|^2 \\
&= \frac{2}{\gamma^2\mu^2 N^2}(1-\gamma\mu)^{s+1}\cdot\left(\mu\cdot\|\boldsymbol{\eta}_0\|^2\right) \\
&\leq \frac{2}{\gamma^2\mu^2 N^2}(1-\gamma\mu)^{s+1}\cdot\left(L(\mathbf{w}_0) - L(\mathbf{w}^*)\right),
\end{aligned}$$

where in the final line, we use the fact that $\mu\mathbf{I} \preceq \mathbf{H}$. This proves the claimed bound. ■

Lemma 11 *For any learning rate $\gamma \leq \gamma_{b,max}/2$, the bias error of the* **final** *iterate $\mathbf{w}_N^{bias}$ is upper bounded as:*

$$\mathbb{E}\left[L(\mathbf{w}_N^{bias})\right] - L(\mathbf{w}^*) \leq \frac{\kappa}{2}\cdot(1-\gamma\mu)^N\cdot\big(L(\mathbf{w}_0) - L(\mathbf{w}^*)\big).$$

Proof Similar to the tail-averaged case, we require to bound the per step contraction properties of an SGD update in the case of the bias error (i.e. $\boldsymbol{\zeta}_\cdot = 0$):

$$\begin{aligned}
\mathbb{E}\left[\|\boldsymbol{\eta}_N\|^2\right] &= \mathbb{E}\left[\boldsymbol{\eta}_{N-1}^\top (\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^b \mathbf{x}_{Ni}\otimes\mathbf{x}_{Ni})(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^b \mathbf{x}_{Ni}\otimes\mathbf{x}_{Ni})\boldsymbol{\eta}_{N-1}\right] \\
&= \mathbb{E}\left[\boldsymbol{\eta}_{N-1}^\top \mathbb{E}\left[(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^b \mathbf{x}_{Ni}\otimes\mathbf{x}_{Ni})(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^b \mathbf{x}_{Ni}\otimes\mathbf{x}_{Ni})\middle|\mathcal{F}_{N-1}\right]\boldsymbol{\eta}_{N-1}\right]
\end{aligned}$$

$$\leq (1-\gamma\mu)\mathbb{E}\left[\|\boldsymbol{\eta}_{N-1}\|^2\right] \quad \text{(using lemma 9)}.$$

This implies that a recursive application of the above bound yields $\mathbb{E}\left[\|\boldsymbol{\eta}_N\|^2\right] \leq (1-\gamma\mu)^N\mathbb{E}\left[\|\boldsymbol{\eta}_0\|^2\right]$. Then,

$$\begin{aligned}
\mathbb{E}\left[L(\mathbf{w}_N^{\text{bias}})\right] - L(\mathbf{w}^*) &= \frac{1}{2}\operatorname{Tr}\left((\boldsymbol{\eta}_N^{\text{bias}})^\top\mathbf{H}\boldsymbol{\eta}_N^{\text{bias}}\right) \\
&\leq \frac{\lambda_{\max}(\mathbf{H})}{2}\operatorname{Tr}\left(\|\boldsymbol{\eta}_N^{\text{bias}}\|^2\right) \\
&\leq \frac{\lambda_{\max}(\mathbf{H})(1-\gamma\mu)^N}{2\lambda_{\min}(\mathbf{H})}\operatorname{Tr}\left(\lambda_{\min}(\mathbf{H})\|\boldsymbol{\eta}_0\|^2\right) \\
&\leq \frac{\lambda_{\max}(\mathbf{H})(1-\gamma\mu)^N}{2\lambda_{\min}(\mathbf{H})}\Big(L(\mathbf{w}_0) - L(\mathbf{w}^*)\Big) \qquad (\text{since, } \mathbf{w}_0 = \mathbf{w}_0^{\text{bias}}). \\
&\leq \frac{\kappa}{2}\cdot(1-\gamma\mu)^N\Big(L(\mathbf{w}_0) - L(\mathbf{w}^*)\Big).
\end{aligned}$$

■

A.4 Lemmas For Bounding The Variance Error

Now, we seek to understand the behavior of the variance error of the tail-averaged iterate $\overline{\mathbf{w}}_{s+1,N}$. We begin by noting here that the variance error is analyzed by beginning the optimization at the solution, i.e. $\boldsymbol{\eta}_0^{\text{variance}} = 0$ and allowing the noise to drive the process. In particular, we write out the recursive updates that characterize the variance error:

$$\boldsymbol{\eta}_t^{\text{variance}} = \mathbf{P}_{tb}\boldsymbol{\eta}_{t-1}^{\text{variance}} + \gamma\boldsymbol{\zeta}_{tb}, \text{ with } \boldsymbol{\eta}_0^{\text{variance}} = 0.$$

This implies that by defining $\boldsymbol{\Phi}_t^{\text{variance}} \stackrel{\text{def}}{=} \mathbb{E}\left[\boldsymbol{\eta}_t^{\text{variance}}\otimes\boldsymbol{\eta}_t^{\text{variance}}\right]$, we have:

$$\begin{aligned}
\boldsymbol{\Phi}_t^{\text{variance}} &= \mathbb{E}\left[\boldsymbol{\eta}_t^{\text{variance}}\otimes\boldsymbol{\eta}_t^{\text{variance}}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\left(\mathbf{P}_{tb}\boldsymbol{\eta}_{t-1}^{\text{variance}} + \gamma\boldsymbol{\zeta}_{tb}\right)\otimes\left(\mathbf{P}_{tb}\boldsymbol{\eta}_{t-1}^{\text{variance}} + \gamma\boldsymbol{\zeta}_{tb}\right)|\mathcal{F}_{t-1}\right]\right] \\
&= \mathbb{E}\left[\mathbf{P}_{tb}\boldsymbol{\Phi}_{t-1}^{\text{variance}}\mathbf{P}_{tb}^\top\right] + \frac{\gamma^2}{b}\boldsymbol{\Sigma}. \qquad (26)
\end{aligned}$$

where, $\mathcal{F}_{t-1}$ is the filtration defined using the samples $\{\mathbf{x}_{ji}, y_{ji}\}_{j=1,i=1}^{j=t-1,i=b}$. Furthermore cross terms are zero since $\mathbb{E}\left[\boldsymbol{\zeta}_{tb}|\mathcal{F}_{t-1}\right] = 0$ owing to first order optimality conditions. Recounting that $\mathbf{P}_{tb} = \mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^b \mathbf{x}_{ti}\otimes\mathbf{x}_{ti}$, we express equation 26 using a linear operator as follows:

$$\begin{aligned}
\mathbb{E}\left[\mathbf{P}_{tb}\boldsymbol{\Phi}_{t-1}^{\text{variance}}\mathbf{P}_{tb}^\top\right] &= \mathbb{E}\left[\left(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^b \mathbf{x}_{ti}\otimes\mathbf{x}_{ti}\right)\boldsymbol{\Phi}_{t-1}^{\text{variance}}\left(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^b \mathbf{x}_{ti}\otimes\mathbf{x}_{ti}\right)\right] \\
&\stackrel{\text{def}}{=} (\mathcal{I} - \gamma\mathcal{T}_b)\boldsymbol{\Phi}_{t-1}^{\text{variance}},
\end{aligned}$$

with $\mathcal{T}_b$ representing the following linear operator:

$$\mathcal{T}_b = \mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}} - \frac{\gamma}{b}\mathcal{M} - \gamma\frac{b-1}{b}\mathcal{H}_{\mathcal{L}}\mathcal{H}_{\mathcal{R}},$$

with $\mathcal{M} = \mathbb{E}[\mathbf{x} \otimes \mathbf{x} \otimes \mathbf{x} \otimes \mathbf{x}]$, $\mathcal{H}_{\mathcal{L}} = \mathbf{H} \otimes \mathbf{I}$ and $\mathcal{H}_{\mathcal{R}} = \mathbf{I} \otimes \mathbf{H}$ representing the left and right multiplication linear operators corresponding to the matrix $\mathbf{H}$. Given this notation, we consider $\mathbf{\Phi}_t^{\text{variance}}$:

$$\begin{aligned}\mathbf{\Phi}_t^{\text{variance}} &= (\mathcal{I} - \gamma\mathcal{T}_b)\mathbf{\Phi}_{t-1}^{\text{variance}} + \frac{\gamma^2}{b}\mathbf{\Sigma} \\ &= \frac{\gamma^2}{b}\left(\sum_{k=0}^{t-1}(\mathcal{I} - \gamma\mathcal{T}_b)^k\right)\mathbf{\Sigma}. \end{aligned} \tag{27}$$

Before bounding the variance error, we will describe a lemma that shows that the expected covariance of the variance error $\mathbf{\Phi}_t^{\text{variance}}$ initialized at 0 grows monotonically to its steady state value (in a PSD sense).

Lemma 12 *The sequence of centered variance iterates $\boldsymbol{\eta}_t^{variance}$ have expected covariances that monotonically grow in a PSD sense, i.e.:*

$$0 = \mathbf{\Phi}_0^{variance} \preceq \mathbf{\Phi}_1^{variance} \preceq \mathbf{\Phi}_2^{variance} \ldots. \preceq \mathbf{\Phi}_\infty^{variance}.$$

Proof This lemma generalizes the lemma appearing in Jain et al. (2017a,b). We begin by recounting the t^{th} variance iterate, i.e.:

$$\boldsymbol{\eta}_t^{\text{variance}} = \gamma\sum_{j=1}^{t}\mathbf{Q}_{j+1,t}\boldsymbol{\zeta}_{j,b}.$$

This implies in particular that

$$\begin{aligned}\mathbf{\Phi}_t^{\text{variance}} &= \mathbb{E}\left[\boldsymbol{\eta}_t^{\text{variance}} \otimes \boldsymbol{\eta}_t^{\text{variance}}\right] \\ &= \gamma^2\sum_{j=1}^{t}\sum_{l=1}^{t}\mathbb{E}\left[\mathbf{Q}_{j+1,t}\boldsymbol{\zeta}_{j,b} \otimes \boldsymbol{\zeta}_{l,b}\mathbf{Q}_{l+1,b}^\top\right] \quad \text{(from equation 14)} \\ &= \gamma^2\sum_{j=1}^{t}\sum_{l=1}^{t}\mathbb{E}\left[\mathbf{Q}_{j+1,t}\mathbb{E}\left[\boldsymbol{\zeta}_{j,b} \otimes \boldsymbol{\zeta}_{l,b}|\mathcal{F}_{j-1}\right]\mathbf{Q}_{l+1,b}^\top\right] \\ &= \gamma^2\sum_{j=1}^{t}\mathbb{E}\left[\mathbf{Q}_{j+1,t}\boldsymbol{\zeta}_{j,b} \otimes \boldsymbol{\zeta}_{j,b}\mathbf{Q}_{j+1,t}^\top\right] \\ &= \frac{\gamma^2}{b}\sum_{j=1}^{t}\mathbb{E}\left[\mathbf{Q}_{j+1,t}\mathbf{\Sigma}\mathbf{Q}_{j+1,t}^\top\right].\end{aligned}$$

where, the third line follows since $\mathbb{E}\left[\boldsymbol{\zeta}_{l,b} \otimes \boldsymbol{\zeta}_{j,b}\right] = 0$ for $j \neq l$, similar to arguments in equation 13. This immediately reveals that the sequence of covariances grows as a function of time, since,

$$\mathbf{\Phi}_{t+1}^{\text{variance}} - \mathbf{\Phi}_t^{\text{variance}} = \frac{\gamma^2}{b}\mathbb{E}\left[\mathbf{Q}_{2,t+1}\mathbf{\Sigma}\mathbf{Q}_{2,t+1}^\top\right] \succeq 0.$$

■

This lemma leads to a natural upper bound on the variance error, as expressed below:

Lemma 13 *With $\gamma < \frac{\gamma_{b,max}}{2}$, the variance error of the tail-averaged iterate $\overline{\mathbf{w}}_{s+1,N}^{variance}$ is upper bounded as:*

$$\mathbb{E}\left[L(\overline{\mathbf{w}}_{s+1,N}^{variance})\right] - L(\mathbf{w}^*) \leq \frac{2}{Nb}\operatorname{Tr}\left(\mathcal{T}_b^{-1}\mathbf{\Sigma}\right).$$

Proof Considering the variance error of tail-averaged iterate from equation 25:

$$\begin{aligned}\mathbb{E}\left[L(\overline{\mathbf{w}}_{s+1,N}^{\text{variance}})\right] - L(\mathbf{w}^*) &= \frac{2}{\gamma N^2}\cdot\sum_{l=s+1}^{s+N}\left(\operatorname{Tr}\left(\mathbb{E}\left[\mathbf{\Phi}_l^{\text{variance}}\right]\right)\right) \\ &\leq \frac{2}{\gamma N}\cdot\operatorname{Tr}\left(\mathbb{E}\left[\mathbf{\Phi}_\infty^{\text{variance}}\right]\right) \quad \text{(from lemma 12)} \\ &= \frac{2}{\gamma N}\cdot\frac{\gamma^2}{b}\cdot\operatorname{Tr}\left(\sum_{k=0}^{\infty}(\mathcal{I} - \gamma\mathcal{T}_b)^k\mathbf{\Sigma}\right) \quad \text{(from equation 14)} \\ &= \frac{2}{Nb}\operatorname{Tr}\left(\mathcal{T}_b^{-1}\mathbf{\Sigma}\right).\end{aligned}$$

■

Lemma 14 *With $\gamma < \frac{\gamma_{b,max}}{2}$, the variance error of the* **final** *iterate $\mathbf{w}_N^{variance}$, obtained by running mini-batch SGD for N steps is upper bounded as:*

$$\mathbb{E}\left[L(\mathbf{w}_N^{variance})\right] - L(\mathbf{w}^*) \leq \frac{\gamma}{2b}\operatorname{Tr}\left(\mathbf{H}\mathcal{T}_b^{-1}\mathbf{\Sigma}\right).$$

Proof We note that since we deal with the square loss case,

$$\begin{aligned}\mathbb{E}\left[L(\mathbf{w}_N^{\text{variance}})\right] - L(\mathbf{w}^*) &= \frac{1}{2}\operatorname{Tr}\left(\mathbf{H}\mathbf{\Phi}_N^{\text{variance}}\right) \\ &\leq \frac{1}{2}\operatorname{Tr}\left(\mathbf{H}\mathbf{\Phi}_\infty^{\text{variance}}\right) \quad \text{(using lemma 12)} \\ &= \frac{\gamma^2}{2b}\operatorname{Tr}\left(\mathbf{H}\sum_{j=0}^{\infty}(\mathcal{I} - \gamma\mathcal{T}_b)^j\mathbf{\Sigma}\right) \\ &= \frac{\gamma}{2b}\operatorname{Tr}\left(\mathbf{H}\mathcal{T}_b^{-1}\mathbf{\Sigma}\right).\end{aligned}$$

■

Lemma 15 *Denoting the assumption (A) $\gamma \leq \gamma_{b,max}/2$,*

1. *With (A) in place, $\mathcal{T}_b \succeq 0$.*
2. *With (A) in place, $\mathcal{T}_b^{-1}\mathbf{W} \succeq 0$ for every $\mathbf{W} \in \mathcal{S}(d)$, $\mathbf{W} \succeq 0$.*
3. $\operatorname{Tr}\left((\mathcal{H}_{\mathcal{R}} + \mathcal{H}_{\mathcal{L}})^{-1}\mathbf{A}\right) = \frac{1}{2}\operatorname{Tr}\left(\mathbf{H}^{-1}\mathbf{A}\right)\ \forall\ \mathbf{A} \in \mathcal{S}^+(\mathbb{R}^d)$.

4. With (A) in place,

$$\operatorname{Tr}\left(\mathcal{T}_b^{-1}\boldsymbol{\Sigma}\right) \leq 2\operatorname{Tr}\left(\mathbf{H}^{-1}\boldsymbol{\Sigma}\right).$$

Proof

Proof of claim 1 in Lemma 15: $\mathcal{T}_b \succeq 0$ implies that for all symmetric matrices $\mathbf{A} \in \mathcal{S}(d)$, we have $\operatorname{Tr}(\mathbf{A}\mathcal{T}_b\mathbf{A}) \geq 0$, and this is true owing to the following inequalities:

$$\begin{aligned}
\langle \mathbf{A}, \mathcal{T}_b\mathbf{A}\rangle &= 2\operatorname{Tr}(\mathbf{AHA}) - \frac{\gamma}{b}\mathbb{E}\left[\langle \mathbf{x}, \mathbf{Ax}\rangle^2\right] - \frac{\gamma(b-1)}{b}\langle \mathbf{H}, \mathbf{AHA}\rangle \\
&\geq 2\operatorname{Tr}(\mathbf{AHA}) - \frac{\gamma}{b}\mathbb{E}\left[\|\mathbf{x}\|^2\|\mathbf{Ax}\|^2\right] - \frac{\gamma(b-1)}{b}\|\mathbf{H}\|\operatorname{Tr}(\mathbf{AHA}) \\
&\geq 2\operatorname{Tr}(\mathbf{AHA}) - \frac{\gamma}{b}R^2\mathbb{E}\left[\|\mathbf{Ax}\|^2\right] - \frac{\gamma(b-1)}{b}\|\mathbf{H}\|\operatorname{Tr}(\mathbf{AHA}) \\
&\geq \left(2 - \frac{\gamma}{b}\left(R^2 + (b-1)\|\mathbf{H}\|\right)\right)\operatorname{Tr}(\mathbf{AHA}).
\end{aligned}$$

Using the definition of $\gamma_{b,\max}$ completes the proof of the claim.

Proof of claim 2 in Lemma 15: We require to prove $\mathcal{T}_b^{-1}$ operating on a PSD matrix produces a PSD matrix, or in other words, $\mathcal{T}_b^{-1}$ is a PSD map.

$$\begin{aligned}
\mathcal{T}_b^{-1} &= [\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R} - \frac{\gamma}{b}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R})]^{-1} \\
&= (\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{\frac{1}{2}}[\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R} - \frac{\gamma}{b}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R})]^{-1} \cdot \\
&\qquad (\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{\frac{1}{2}}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}} \\
&= (\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}[\mathcal{I} - \frac{\gamma}{b}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R})(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}]^{-1}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}.
\end{aligned} \tag{28}$$

Now, we prove that $\|\frac{\gamma}{b}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R})(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}\| < 1$. Given $\gamma < \gamma_{b,\max}/2$, we employ claim 1 to note that $\mathcal{T}_b \succ 0$.

$$\begin{aligned}
&\mathcal{T}_b \succ 0 \\
&\Rightarrow \mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R} - \frac{\gamma}{b}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R}) \succ 0 \\
&\Rightarrow \frac{\gamma}{b}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R}) \prec \mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R} \\
&\Rightarrow \frac{\gamma}{b}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R})(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}} \prec \mathcal{I} \\
&\Rightarrow \|\frac{\gamma}{b}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R})(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}\| < 1.
\end{aligned}$$

With this fact in place, we employ Taylor series to expand $\mathcal{T}_b^{-1}$ in equation 28, i.e.:

$$\begin{aligned}
\mathcal{T}_b^{-1} &= (\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}\sum_{i=0}^{\infty}(\frac{\gamma}{b}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R})(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}})^i(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}} \\
&= \sum_{i=0}^{\infty}(\frac{\gamma}{b}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-1}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R}))^i(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-1}.
\end{aligned}$$

The proof completes by employing the following facts: Using Lyapunov's theorem (Bhatia (2007) proposition A 1.2.6), we know $(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-1}$ is a PSD map, i.e. if $(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-1}(A) = B$, then, if A is PSD $\implies$ B is PSD. Furthermore, $\mathcal{M}$ is also a PSD map, i.e. if A_1 is PSD, $\mathcal{M}(A_1) = E[(x^TA_1x)x \otimes x]$ is PSD as well. Finally, $\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R}$ is also a PSD map, since, if A_2 is PSD, then, $\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R}(A_2) = HA_2H$ which is PSD as well. With all these facts in place, we note that each term in the Taylor's expansion above is a PSD map implying the overall map is PSD as well, thus rounding up the proof to claim 2 in Lemma 15.

Proof of claim 3 in Lemma 15:

We know that the operator $(\mathcal{H}_\mathcal{R} + \mathcal{H}_\mathcal{L})^{-1}$ is a PSD map, i.e, it maps PSD matrices to PSD matrices. Since $\mathbf{A} \succeq 0$, we replace this condition with $\mathbf{U} = (\mathcal{H}_\mathcal{R} + \mathcal{H}_\mathcal{L})^{-1}\mathbf{A} \succeq 0$ implying, we need to show the following:

$$\operatorname{Tr}(\mathbf{U}) = \frac{1}{2}\operatorname{Tr}\left(\mathbf{H}^{-1}\mathbf{A}\right) \ \forall\, \mathbf{U} \succeq 0.$$

Examining the right hand side, we see the following:

$$\begin{aligned}
\frac{1}{2}\operatorname{Tr}\left(\mathbf{H}^{-1}\mathbf{A}\right) &= \frac{1}{2}\operatorname{Tr}\left(\mathbf{H}^{-1}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})\mathbf{U}\right) \\
&= \frac{1}{2}\operatorname{Tr}\left(\mathbf{H}^{-1}\mathbf{HU} + \mathbf{H}^{-1}\mathbf{UH}\right) \\
&= \operatorname{Tr}(\mathbf{U}).
\end{aligned}$$

thus wrapping up the proof of claim 4.

Proof of claim 4 in Lemma 15: Let $\mathcal{U} = \mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R} - \frac{\gamma}{b}\cdot(b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R}$. Then,

$$\begin{aligned}
\mathcal{T}_b^{-1}\boldsymbol{\Sigma} &= \left(\mathcal{U} - \frac{\gamma}{b}\mathcal{M}\right)^{-1}\boldsymbol{\Sigma} \\
&= \sum_{i=0}^{\infty}\left(\frac{\gamma}{b}\mathcal{U}^{-1}\mathcal{M}\right)^i\mathcal{U}^{-1}\boldsymbol{\Sigma}.
\end{aligned}$$

Let $\mathbf{A} = \mathcal{U}^{-1}\boldsymbol{\Sigma}$, $\mathbf{A}' = \mathcal{U}^{-1}\mathbf{H}$. Then,

$$\mathcal{T}_b^{-1}\boldsymbol{\Sigma} = \sum_{i=1}^{\infty}\left(\frac{\gamma}{b}\mathcal{U}^{-1}\mathcal{M}\right)^i\mathbf{A}.$$

The $i = 0$ term is just $= \mathbf{A}$. Now, considering $i = 1$, we have:

$$\begin{aligned}
\frac{\gamma}{b}\mathcal{U}^{-1}\mathcal{M}\mathbf{A} &\preceq \frac{\gamma}{b}\|\mathbf{A}\|_2\mathcal{U}^{-1}\mathcal{M}\mathbf{I} \\
&\preceq \frac{\gamma}{b}\|\mathbf{A}\|_2R^2\mathcal{U}^{-1}\mathbf{H} = \frac{\gamma}{b}\|\mathbf{A}\|_2R^2\mathbf{A}'.
\end{aligned}$$

Next, considering $i = 2$, we have:

$$\left(\frac{\gamma}{b}\mathcal{U}^{-1}\mathcal{M}\right)^2\mathbf{A} = \left(\frac{\gamma}{b}\mathcal{U}^{-1}\mathcal{M}\right)\cdot\left(\frac{\gamma}{b}\mathcal{U}^{-1}\mathcal{M}\right)\mathbf{A}$$

$$\preceq \left(\frac{\gamma}{b}\|\mathbf{A}\|_2 R^2\right) \cdot \left(\frac{\gamma}{b}\mathcal{U}^{-1}\mathcal{M}\right)\mathbf{A}'$$
$$\preceq \left(\frac{\gamma}{b}\|\mathbf{A}\|_2 R^2\right) \cdot \left(\frac{\gamma}{b}\mathcal{U}^{-1}\right) \cdot \|\mathbf{A}'\|_2 \cdot R^2\mathbf{H}$$
$$\preceq \left(\frac{\gamma}{b}\|\mathbf{A}\|_2 R^2\right) \cdot \left(\frac{\gamma}{b}\|\mathbf{A}'\|_2 R^2\right) \cdot \mathbf{A}'.$$

Noting this recursive structure, we see that:

$$\begin{aligned}\mathcal{T}_b^{-1}\mathbf{\Sigma} &= \sum_{i=0}^{\infty}\left(\frac{\gamma}{b}\mathcal{U}^{-1}\mathcal{M}\right)^i \mathbf{A} \\ &\preceq \mathbf{A} + \sum_{i=1}^{\infty}\left(\frac{\gamma}{b}\|\mathbf{A}\|_2 R^2\right)\cdot\left(\frac{\gamma}{b}\|\mathbf{A}'\|_2 R^2\right)^{i-1}\cdot\mathbf{A}' \\ &= \mathbf{A} + \frac{\left(\frac{\gamma}{b}\|\mathbf{A}\|_2 R^2\right)}{1-\left(\frac{\gamma}{b}\|\mathbf{A}'\|_2 R^2\right)}\cdot\mathbf{A}'.\end{aligned}$$

Note that this summation is finite iff $\gamma \leq \frac{b}{R^2\|\mathbf{A}'\|_2}$. Further, applying the trace operator on both sides, we have:

$$\operatorname{Tr}\left(\mathcal{T}_b^{-1}\mathbf{\Sigma}\right) \leq \operatorname{Tr}\left(\mathbf{A}\right) + \frac{\left(\frac{\gamma}{b}\|\mathbf{A}\|_2 R^2\right)}{1-\left(\frac{\gamma}{b}\|\mathbf{A}'\|_2 R^2\right)}\operatorname{Tr}\left(\mathbf{A}'\right). \tag{29}$$

Now, for any psd matrix $\mathbf{B} \succeq 0$, let us upperbound $\mathcal{U}^{-1}\mathbf{B}$:

$$\mathcal{U}^{-1}\mathbf{B} = \sum_{j=0}^{\infty}\left(\gamma\cdot\frac{b-1}{b}\cdot(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\cdot\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R}\right)^i(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathbf{\Sigma}.$$

The recursion can be bounded by analyzing $i = 1$:

$$\begin{aligned}&\gamma\cdot\frac{b-1}{b}\cdot(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\cdot\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R}\cdot(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathbf{B} \\ &\preceq \|(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathbf{B}\|_2\cdot\gamma\cdot\frac{b-1}{b}\cdot(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\cdot\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R}\cdot\mathbf{I} \\ &\preceq \|(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathbf{B}\|_2\cdot\gamma\cdot\frac{b-1}{b}\cdot(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\cdot\|\mathbf{H}\|_2\mathbf{H} \\ &= \|(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathbf{B}\|_2\cdot\gamma\cdot\frac{b-1}{2b}\cdot\|\mathbf{H}\|_2\cdot\mathbf{I}.\end{aligned}$$

This indicates the means to recurse for bounding terms $i \geq 2$:

$$\mathcal{U}^{-1}\mathbf{B} \preceq \sum_{j=0}^{\infty}\|(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathbf{B}\|_2\left(\gamma\cdot\frac{b-1}{2b}\cdot\|\mathbf{H}\|_2\right)^j\cdot\mathbf{I}$$

$$= \frac{\|(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathbf{B}\|_2}{1-\gamma\cdot\frac{(b-1)\|\mathbf{H}\|_2}{2b}}\cdot\mathbf{I}.$$

The upperbound above is true as long as $\gamma < \frac{2b}{(b-1)\|\mathbf{H}\|_2}$. This now allows us to obtain bounds on $\|\mathbf{A}\|_2, \|\mathbf{A}'\|_2, \operatorname{Tr}(\mathbf{A}')$:

$$\|\mathbf{A}\|_2 \leq \frac{\|(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathbf{\Sigma}\|_2}{1-\gamma\cdot\frac{b-1}{2b}\cdot\|\mathbf{H}\|_2}$$
$$\|\mathbf{A}'\|_2 \leq \frac{1/2}{1-\gamma\cdot\frac{b-1}{2b}\cdot\|\mathbf{H}\|_2}$$
$$\operatorname{Tr}\left(\mathbf{A}'\right) \leq \frac{d/2}{1-\gamma\cdot\frac{b-1}{2b}\cdot\|\mathbf{H}\|_2}.$$

Substituting these in equation 29:

$$\operatorname{Tr}\left(\mathcal{T}_b^{-1}\mathbf{\Sigma}\right) \leq \operatorname{Tr}\left(\mathbf{A}\right) + \frac{\frac{\gamma R^2}{2b}\cdot d\|(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathbf{\Sigma}\|_2}{\left(1-\frac{\gamma}{2b}\cdot(R^2+(b-1)\|\mathbf{H}\|_2)\right)\cdot\left(1-\gamma\cdot\frac{b-1}{2b}\|\mathbf{H}\|_2\right)}. \tag{30}$$

with the conditions on γ being: $\gamma \leq \frac{2b}{(b-1)\|\mathbf{H}\|_2}$, $\gamma \leq \frac{2b}{R^2+(b-1)\|\mathbf{H}\|_2}$, $\gamma \leq \frac{2b}{R^2}$. These are combined using $\gamma \leq \frac{2b}{R^2+(b-1)\|\mathbf{H}\|_2}$. Once this condition is satisfied, the denominator of the second term can be upperbounded by atmost a constant. Next, looking at the numerator of the second term, we see that $\gamma \leq \frac{2b}{R^2\cdot\frac{d\|(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathbf{\Sigma}\|_2}{\operatorname{Tr}((\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathbf{\Sigma})}} = \frac{2b}{R^2\rho_\mathrm{m}}$ allows for the second term to be upperbounded by $\mathcal{O}(\operatorname{Tr}((\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathbf{\Sigma}))$. This is clearly satisfied if $\gamma \leq \frac{2b}{R^2\cdot\rho_\mathrm{m}+(b-1)\|\mathbf{H}\|_2}$. In particular, setting γ to be half of this maximum, we have:

$$\operatorname{Tr}\left(\mathcal{T}_b^{-1}\mathbf{\Sigma}\right) \leq \operatorname{Tr}\left(\mathbf{A}\right) + 2\operatorname{Tr}\left((\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathbf{\Sigma}\right). \tag{31}$$

Denoting $\hat{\mathbf{\Sigma}} = (\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R}-\gamma\cdot\frac{b-1}{b}\cdot\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R})^{-1}\mathbf{\Sigma}$, in order to bound $\operatorname{Tr}(\mathbf{A})$, we require comparing $\operatorname{Tr}\left(\hat{\mathbf{\Sigma}}\right)$ with $\operatorname{Tr}\left(\tilde{\mathbf{\Sigma}}\right) = \operatorname{Tr}\left((\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathbf{\Sigma}\right)$. For this, without loss of generality, we can consider $\mathbf{H}$ to be diagonal, and this implies that comparing the diagonal elements of $\hat{\mathbf{\Sigma}}_{ii} = \mathbf{\Sigma}_{ii}/(2\lambda_i - \gamma\frac{b-1}{b}\lambda_i^2)$ while $\tilde{\mathbf{\Sigma}}_{ii} = \mathbf{\Sigma}_{ii}/2\lambda_i$. Comparing these, we see that

$$\begin{aligned}\operatorname{Tr}\left(\hat{\mathbf{\Sigma}}\right) = \operatorname{Tr}\left((\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R}-\gamma\cdot\frac{b-1}{b}\cdot\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R})^{-1}\mathbf{\Sigma}\right) &\leq \frac{1}{1-\gamma\frac{b-1}{2b}\|\mathbf{H}\|_2}\operatorname{Tr}\left(\tilde{\mathbf{\Sigma}}\right) \\ &= \frac{1}{1-\gamma\frac{b-1}{2b}\|\mathbf{H}\|_2}\operatorname{Tr}\left((\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathbf{\Sigma}\right).\end{aligned}$$

Noting that $\operatorname{Tr}(\mathbf{A}) = \operatorname{Tr}\left(\hat{\mathbf{\Sigma}}\right)$, we see that substituting the above in equation 31, we have:

$$\begin{aligned}\operatorname{Tr}\left(\mathcal{T}_b^{-1}\mathbf{\Sigma}\right) &\leq \frac{1}{1-\gamma\frac{b-1}{2b}\|\mathbf{H}\|_2}\operatorname{Tr}\left((\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathbf{\Sigma}\right) + 2\operatorname{Tr}\left((\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathbf{\Sigma}\right) \\ &\leq 4\operatorname{Tr}\left((\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathbf{\Sigma}\right) = 2\operatorname{Tr}\left(\mathbf{H}^{-1}\mathbf{\Sigma}\right).\end{aligned}$$

■

Corollary 16 *Consider the mis-specified case of the streaming LSR problem. With* $\gamma \leq \frac{\gamma_{b,max}}{2}$, *the variance error of the tail-averaged iterate* $\overline{\mathbf{w}}_{s+1,N}^{variance}$ *is upper bounded as:*

$$\mathbb{E}\left[L(\overline{\mathbf{w}}_{s+1,N}^{variance})\right] - L(\mathbf{w}^*) \leq \frac{4}{Nb} \cdot \widehat{\sigma^2_{MLE}}.$$

Proof The result follows in a straightforward manner by noting that $\gamma \leq \frac{\gamma_{b,\max}}{2}$ implying that $\operatorname{Tr}\left(\mathcal{T}_b^{-1}\mathbf{\Sigma}\right) \leq 2\operatorname{Tr}\left(\mathbf{H}^{-1}\mathbf{\Sigma}\right)$ and by substituting into the result of lemma 13. ■

Corollary 17 *With* $\gamma \leq \frac{\gamma_{b,max}}{2}$, $\mathbf{\Sigma} = \sigma^2\mathbf{H}$ *the variance error of the* **final** *iterate* $\mathbf{w}_N^{variance}$, *obtained by running mini-batch SGD for* N *steps is upper bounded as:*

$$\mathbb{E}\left[L(\mathbf{w}_N^{variance})\right] - L(\mathbf{w}^*) \leq \frac{\gamma\sigma^2}{2b}\operatorname{Tr}\mathbf{H}.$$

Proof This follows from the fact that $\mathcal{T}_b^{-1}\mathbf{\Sigma} \preceq \sigma^2\mathbf{I}$, implying that $\mathbf{H}\mathcal{T}_b^{-1}\mathbf{\Sigma} \preceq \sigma^2\mathbf{H}$ and then applying the trace operator on the result of lemma 14. ■

A.5 Main Results

A.5.1 Derivation of Divergent Learning Rate

A necessary condition for the convergence of Stochastic Gradient Updates is $\mathcal{T}_b \succeq 0$, and this by definition implies,

$$\begin{aligned}
&\langle \mathbf{W}, \mathcal{T}_b\mathbf{W}\rangle \geq 0, \quad \mathbf{W} \in \mathcal{S}(d) \\
\implies &2\operatorname{Tr}(\mathbf{WHW}) - \frac{\gamma}{b}\operatorname{Tr}(\mathbf{W}\mathcal{M}\mathbf{W}) - \gamma\left(\frac{b-1}{b}\right)\operatorname{Tr}(\mathbf{WHWH}) \geq 0 \\
\implies &\frac{2}{\gamma} \geq \frac{\operatorname{Tr}(\mathbf{W}\mathcal{M}\mathbf{W}) + (b-1)\operatorname{Tr}(\mathbf{WHWH})}{b\operatorname{Tr}(\mathbf{WHW})} \\
\implies &\frac{2}{\gamma_{b,\max}^{div}} = \sup_{\mathbf{W}\in\mathcal{S}(d)} \frac{\operatorname{Tr}(\mathbf{W}\mathcal{M}\mathbf{W}) + (b-1)\operatorname{Tr}(\mathbf{WHWH})}{b\operatorname{Tr}(\mathbf{WHW})}.
\end{aligned}$$

A.5.2 Proof of Theorem 1

Proof [proof of Theorem 1] The proof of theorem 1 follows from characterizing bias-variance decomposition for the tail-averaged iterate in section A.2.3 with equation 23.

The bias error of the tail-averaged iterate (equation 24) is bounded with lemma 9 and lemma 10 in section A.3.

The variance error of the tail-averaged iterate (equation 25) is bounded with lemma 12, lemma 13, lemma 15 and corollary 16 in section A.4.

The final expression follows through substituting the result of lemma 10 and corollary 16 into equation 23, with appropriate parameters of the problem, i.e., with a batch size b, number of burn-in iterations s, number of tail-averaged iterations $n/b - s$ to provide the claimed excess risk bound of

Algorithm 1:

$$\mathbb{E}\left[L(\overline{\mathbf{w}})\right] - L(\mathbf{w}^*) \leq \frac{2}{\gamma^2\mu^2(\frac{n}{b}-s)^2} \cdot (1-\gamma\mu)^s \cdot \left(L(\mathbf{w}_0) - L(\mathbf{w}^*)\right) + 4 \cdot \frac{\widehat{\sigma^2_{\text{MLE}}}}{b\cdot(\frac{n}{b}-s)}.$$

■

A.5.3 Proof of Lemma 4

Proof [proof of Lemma 4] The proof of lemma 4 follows from characterizing bias-variance decomposition for the final iterate in section A.2.2 with equation 18.

The bias error of the final iterate is bounded with lemma 9 and lemma 11 in section A.3.

The variance error of the final iterate is bounded with lemma 12, lemma 14, lemma 15 and corollary 17 in section A.4.

The final expression follows through substituting the result of lemma 11 and corollary 17 into equation 18, with appropriate parameters of the problem, i.e., with a batch size b, number of samples n and number of iterations $\lfloor n/b \rfloor$, to provide the claimed excess risk bound:

$$\mathbb{E}\left[L(\mathbf{w}_{\lfloor n/b\rfloor})\right] - L(\mathbf{w}^*) \leq \kappa_b(1-\gamma\mu)^{\lfloor n/b\rfloor}\left(L(\mathbf{w}_0) - L(\mathbf{w}^*)\right) + \frac{\gamma}{b}\sigma^2\operatorname{Tr}(\mathbf{H}).$$

■

A.5.4 Proof of Theorem 5

Proof Let $\widetilde{L_e} = \mathbb{E}\left[L(\mathbf{w}_e)\right] - L(\mathbf{w}^*)$. We will first provide a recursive bound for $\widetilde{L_e}$ for $e \leq \log\left(\frac{n}{bt}\right) - 1$ using theorem 1, with a mini-batch size of $b_e = 1 + 2^{e-1}b$, where, $b = b_{\text{thresh}} - 1$, $n_e = b_e \cdot t$, $s = t - 1$:

$$\begin{aligned}
\widetilde{L_e} &\leq 2\kappa_{b_e}^2 \exp\left(-\frac{n_e}{b_e\cdot\kappa_{b_e}}\right)\widetilde{L_{e-1}} + 4\frac{\widehat{\sigma^2_{\text{MLE}}}}{b_e} \\
&\leq \exp\left(-\frac{n_e}{3b_e\kappa_e\log(\kappa_e)}\right)\cdot\widetilde{L_{e-1}} + 4\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}}{b_e}.
\end{aligned}$$

Next, denote $\kappa = \|\mathbf{H}\|_2/\mu$; now, let us bound κ_{b_e}:

$$\begin{aligned}
\kappa_{b_e} &= \frac{R^2\cdot\frac{d\left\|(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathbf{\Sigma}\right\|_2}{\operatorname{Tr}((\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathbf{\Sigma})} + (b_e-1)\|\mathbf{H}\|_2}{b_e\mu} \\
&= \kappa\cdot\frac{b_{\text{thresh}}-1+b_e-1}{b_e} = \kappa\cdot\frac{b_{\text{thresh}}-1+2^{e-1}(b_{\text{thresh}}-1)}{2^{e-1}(b_{\text{thresh}}-1)} \\
&= \kappa\cdot\frac{1+2^{e-1}}{2^{e-1}} \leq 2\kappa.
\end{aligned}$$

This implies $\kappa_{b_e}\log(\kappa_{b_e}) \le 4\kappa\log(\kappa)$. This implies, revisiting the recursion on $\widetilde{L_e}$, we have:

$$\begin{aligned}
\widetilde{L_e} &\le \exp\left(-\frac{n_e}{12b_e\kappa\log(\kappa)}\right)\cdot\widetilde{L_{e-1}} + 4\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}}{b_e}\\
&\le \exp\left(-\frac{t}{12\kappa\log(\kappa)}\right)\cdot\widetilde{L_{e-1}} + 4\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}}{2^{e-1}b}\\
&\le \exp\left(-\frac{te}{12\kappa\log(\kappa)}\right)\cdot\widetilde{L_0} + \frac{4\widehat{\sigma^2_{\text{MLE}}}}{b}\cdot\sum_{j=1}^{e}\frac{\exp\left(-\frac{t(j-1)}{12\kappa\log(\kappa)}\right)}{2^{e-j}}\\
&\le \exp\left(-\frac{te}{12\kappa\log(\kappa)}\right)\cdot\widetilde{L_0} + \frac{4\widehat{\sigma^2_{\text{MLE}}}}{b}\cdot\frac{1/2^{e-1}}{1-2\cdot\exp\left(-\frac{t}{12\kappa\log\kappa}\right)}\\
&\le \exp\left(-\frac{te}{12\kappa\log(\kappa)}\right)\cdot\widetilde{L_0} + \frac{12\widehat{\sigma^2_{\text{MLE}}}}{2^e b} \quad (\text{since } t > 24\kappa\log(\kappa))\\
&= \exp\left(-\frac{te}{12\kappa\log(\kappa)}\right)\cdot\widetilde{L_0} + \frac{12\widehat{\sigma^2_{\text{MLE}}}}{b\cdot n}\cdot(4bt) \quad (\text{since } 2^e = n/(4bt))\\
&= \exp\left(-\frac{te}{12\kappa\log(\kappa)}\right)\cdot\widetilde{L_0} + 48\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}t}{n}. \qquad (32)
\end{aligned}$$

Next, for the final epoch, we have $b = n/2t$, $s = t/2$, and a total of $n/2$ samples, implying:

$$\begin{aligned}
\widetilde{L_{e+1}} &\le \frac{2\kappa_b^2}{\left(t/2\right)^2}\cdot\exp\left(-\frac{t}{2\kappa_b}\right)\widetilde{L_e} + 4\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}}{b\cdot(n/4b)} = \frac{8\kappa_b^2}{t^2}\cdot\exp\left(-\frac{t}{2\kappa_b}\right)\cdot\widetilde{L_e} + 16\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}}{n}\\
&\le \frac{32\kappa^2}{t^2}\cdot\exp\left(-\frac{t}{4\kappa}\right)\cdot\widetilde{L_e} + 16\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}}{n} \quad (\text{since } \kappa_b \le 2\kappa)\\
&\le \frac{32\kappa^2}{t^2}\cdot\exp\left(-\frac{t}{4\kappa}\right)\cdot\left(\exp\left(-\frac{te}{12\kappa\log(\kappa)}\right)\cdot\widetilde{L_0} + 48\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}t}{n}\right) + 16\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}}{n}\\
&\le \frac{32\kappa^2}{t^2}\cdot\exp\left(-\frac{t}{4\kappa}\right)\cdot\exp\left(-\frac{te}{12\kappa\log(\kappa)}\right)\cdot\widetilde{L_0} + 64\kappa\exp\left(-t/4\kappa\right)\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}}{n} + 16\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}}{n}\\
&\le \frac{32\kappa^2}{t^2}\cdot\exp\left(-\frac{t}{4\kappa}\right)\cdot\exp\left(-\frac{te}{12\kappa\log(\kappa)}\right)\cdot\widetilde{L_0} + 80\frac{\widehat{\sigma^2_{\text{MLE}}}}{n}\\
&\le \exp\left(-\frac{t(e+1)}{12\kappa\log(\kappa)}\right)\cdot\widetilde{L_0} + 80\frac{\widehat{\sigma^2_{\text{MLE}}}}{n}\\
&= \left(\frac{2bt}{n}\right)^{\frac{t}{12\kappa\log(\kappa)}}\widetilde{L_0} + 80\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}}{n}, \qquad (33)
\end{aligned}$$

which rounds up the proof of the theorem. ■

A.5.5 PROOF OF THEOREM 6

Proof For analyzing the parameter mixing scheme, we require tracking the progress of the i^{th} machine's SGD updates using its centered estimate $\boldsymbol{\eta}_k^{(i)}$. Furthermore, the tail-averaged iterate for the i^{th} machine is represenetead as $\bar{\boldsymbol{\eta}}^{(i)} \stackrel{\text{def}}{=} \frac{1}{N}\sum_{k=s+1}^{s+N}\boldsymbol{\eta}_k^{(i)}$. Finally, the model averaged estimate is represented with its own centered estimate defined as $\bar{\boldsymbol{\eta}} = \frac{1}{P}\sum_{i=1}^{P}\bar{\boldsymbol{\eta}}^{(i)}$. Now, in a manner similar to standard mini-batch tail-averaged SGD on a single machine, the model averaged iterate admits its own bias variance decomposition, through which $\bar{\boldsymbol{\eta}} = \bar{\boldsymbol{\eta}}^{\text{bias}} + \bar{\boldsymbol{\eta}}^{\text{variance}}$ and an upperbound on the excess risk is written as:

$$\begin{aligned}
\mathbb{E}\left[L(\overline{\mathbf{w}})\right] - L(\mathbf{w}^*) &= \mathbb{E}\left[\frac{1}{2}\langle(\overline{\mathbf{w}}-\mathbf{w}^*),\mathbf{H}(\overline{\mathbf{w}}-\mathbf{w}^*)\rangle\right] = \mathbb{E}\left[\frac{1}{2}\langle\bar{\boldsymbol{\eta}},\mathbf{H}\bar{\boldsymbol{\eta}}\rangle\right]\\
&\le \mathbb{E}\left[\langle\bar{\boldsymbol{\eta}}^{\text{bias}},\mathbf{H}\bar{\boldsymbol{\eta}}^{\text{bias}}\rangle\right] + \mathbb{E}\left[\langle\bar{\boldsymbol{\eta}}^{\text{variance}},\mathbf{H}\bar{\boldsymbol{\eta}}^{\text{variance}}\rangle\right].
\end{aligned}$$

We will first handle the variance since it is straightforward given that the noise $\boldsymbol{\zeta}$ is independent for different machines SGD runs. What this implies is the following:

$$\begin{aligned}
\bar{\boldsymbol{\eta}}^{\text{variance}} &= \frac{1}{P}\sum_{i=1}^{P}\bar{\boldsymbol{\eta}}^{(i),\text{variance}}\\
\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{\text{variance}}\otimes\bar{\boldsymbol{\eta}}^{\text{variance}}\right] &= \frac{1}{P^2}\sum_{i,j}\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{variance}}\otimes\bar{\boldsymbol{\eta}}^{(j),\text{variance}}\right]\\
&= \frac{1}{P^2}\left(\sum_{i}\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{variance}}\otimes\bar{\boldsymbol{\eta}}^{(i),\text{variance}}\right] + \sum_{i\neq j}\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{variance}}\otimes\bar{\boldsymbol{\eta}}^{(j),\text{variance}}\right]\right)\\
&= \frac{1}{P}\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(1),\text{variance}}\otimes\bar{\boldsymbol{\eta}}^{(1),\text{variance}}\right]. \qquad (34)
\end{aligned}$$

Where, the final line follows because $\forall\ i \neq j$, the terms are in expectation equal to zero since in expectation each of the noise terms is zero (from first order optimality conditions). The other observation is that the only terms left are P independent runs of tail-averaged SGD in each of the machine, whose risk is straightforward to bound from corollary 16. This implies

$$\langle\mathbf{H},\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{\text{variance}}\otimes\bar{\boldsymbol{\eta}}^{\text{variance}}\right]\rangle \le \frac{4}{PNb}\cdot\widehat{\sigma^2_{\text{MLE}}}. \quad (\text{using corollary 16}) \qquad (35)$$

Next, let us consider the bias error:

$$\begin{aligned}
\bar{\boldsymbol{\eta}}^{\text{bias}} &= \frac{1}{P}\sum_{i=1}^{P}\bar{\boldsymbol{\eta}}^{(i),\text{bias}}\\
\implies \mathbb{E}\left[\bar{\boldsymbol{\eta}}^{\text{bias}}\otimes\bar{\boldsymbol{\eta}}^{\text{bias}}\right] &= \frac{1}{P^2}\sum_{i,j}\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{bias}}\otimes\bar{\boldsymbol{\eta}}^{(j),\text{bias}}\right]\\
&= \frac{1}{P^2}\left(\sum_{i=1}^{P}\underbrace{\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{bias}}\otimes\bar{\boldsymbol{\eta}}^{(i),\text{bias}}\right]}_{\text{independent runs of tail-averaged SGD}} + \sum_{i\neq j}\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{bias}}\otimes\bar{\boldsymbol{\eta}}^{(j),\text{bias}}\right]\right),
\end{aligned}$$

(36)

which implies that we require bounding $\forall\, i \neq j, \mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{bias}} \otimes \bar{\boldsymbol{\eta}}^{(j),\text{bias}}\right]$.

$$\begin{aligned}
\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{bias}} \otimes \bar{\boldsymbol{\eta}}^{(j),\text{bias}}\right] &= \frac{1}{N^2}\sum_{k,l=s+1}^{s+N} \mathbb{E}\left[\boldsymbol{\eta}_k^{(i),\text{bias}} \otimes \boldsymbol{\eta}_l^{(j),\text{bias}}\right] \\
&= \frac{1}{N^2}\sum_{k,l=s+1}^{s+N} \mathbb{E}\left[\boldsymbol{\eta}_k^{(i),\text{bias}}\right] \otimes \mathbb{E}\left[\boldsymbol{\eta}_l^{(j),\text{bias}}\right] \\
&= \frac{1}{N^2}\sum_{k,l=s+1}^{s+N} \mathbb{E}\left[\mathbf{Q}_{1:k}^{(i)}\boldsymbol{\eta}_0\right] \otimes \mathbb{E}\left[\mathbf{Q}_{1:l}^{(j)}\boldsymbol{\eta}_0\right] \quad \text{(from equation 15)} \\
&= \frac{1}{N^2}\left(\sum_{k=s+1}^{s+N}(\mathbf{I}-\gamma\mathbf{H})^k\right)\boldsymbol{\eta}_0 \otimes \boldsymbol{\eta}_0\left(\sum_{l=s+1}^{s+N}(\mathbf{I}-\gamma\mathbf{H})^l\right) \\
&\preceq \frac{1}{N^2}\left(\sum_{k=s+1}^{\infty}(\mathbf{I}-\gamma\mathbf{H})^k\right)\boldsymbol{\eta}_0 \otimes \boldsymbol{\eta}_0\left(\sum_{l=s+1}^{\infty}(\mathbf{I}-\gamma\mathbf{H})^l\right) \\
&= \frac{1}{\gamma^2N^2}\mathbf{H}^{-1}(\mathbf{I}-\gamma\mathbf{H})^{s+1}\boldsymbol{\eta}_0 \otimes \boldsymbol{\eta}_0(\mathbf{I}-\gamma\mathbf{H})^{s+1}\mathbf{H}^{-1}.
\end{aligned}$$

This implies that,

$$\begin{aligned}
\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{bias}} \otimes \bar{\boldsymbol{\eta}}^{(j),\text{bias}}\right] &\leq \frac{1}{\gamma^2N^2}\cdot\langle \mathbf{H}, \mathbf{H}^{-1}(\mathbf{I}-\gamma\mathbf{H})^{s+1}\boldsymbol{\eta}_0 \otimes \boldsymbol{\eta}_0(\mathbf{I}-\gamma\mathbf{H})^{s+1}\mathbf{H}^{-1}\rangle \\
&= \frac{1}{\gamma^2N^2}\cdot\boldsymbol{\eta}_0^\top(\mathbf{I}-\gamma\mathbf{H})^{s+1}\mathbf{H}^{-1}\mathbf{H}\mathbf{H}^{-1}(\mathbf{I}-\gamma\mathbf{H})^{s+1}\boldsymbol{\eta}_0 \\
&\leq \frac{(1-\gamma\mu)^{2s+2}}{\mu\gamma^2N^2}\|\boldsymbol{\eta}_0\|^2 \leq \frac{(1-\gamma\mu)^{2s+2}}{\mu^2\gamma^2N^2}\cdot\Big(L(\mathbf{w}_0)-L(\mathbf{w}^*)\Big). \qquad (37)
\end{aligned}$$

Combining the bound for the cross terms in equation 37 and lemma 10 for the self-terms, we get:

$$\langle \mathbf{H}, \mathbb{E}\left[\bar{\boldsymbol{\eta}}^{\text{bias}} \otimes \bar{\boldsymbol{\eta}}^{\text{bias}}\right]\rangle \leq \frac{(1-\gamma\mu)^{s+1}}{\mu^2\gamma^2N^2}\cdot\frac{2+(1-\gamma\mu)^{s+1}\cdot(P-1)}{P}\cdot\Big(L(\mathbf{w}_0)-L(\mathbf{w}^*)\Big). \qquad (38)$$

The proof wraps up by substituting the relation $N = n/(P\cdot b) - s$ in equations 35 and 38. ■

A.5.6 PROOF OF LEMMA 3

For this problem instance, we begin by noting that $(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\boldsymbol{\Sigma}$ is diagonal as well, with entries:

$$\{(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\boldsymbol{\Sigma}\}_{ii} = \frac{1}{2}\{\mathbf{H}^{-1}\boldsymbol{\Sigma}\}_{ii} = \begin{cases} 1/2 & \text{if } i=1 \\ 1/2(d-1) & \text{if } i>1 \end{cases}.$$

Let us consider the case with batch size $b=1$. With the appropriate choice of step size γ that ensure contracting operators, we require considering $\operatorname{Tr}\left(\mathcal{T}_b^{-1}\boldsymbol{\Sigma}\right)$ as in equation 29, which corresponds to bounding the leading order term in the variance. We employ the taylor's expansion (just as in claim 2 of lemma 15) to expand the term of interest $\mathcal{T}_b^{-1}\boldsymbol{\Sigma}$:

$$\mathcal{T}_b^{-1}\boldsymbol{\Sigma} = \sum_{i=0}^{\infty}\left(\gamma(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathcal{M}\right)^i(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\boldsymbol{\Sigma}$$

$$\begin{aligned}
&= (\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\boldsymbol{\Sigma} + \sum_{i=1}^{\infty}\left(\gamma(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathcal{M}\right)^i(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\boldsymbol{\Sigma} \\
\Rightarrow \operatorname{Tr}\mathcal{T}_b^{-1}\boldsymbol{\Sigma} &= \operatorname{Tr}(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\boldsymbol{\Sigma} + \sum_{i=1}^{\infty}\operatorname{Tr}\left[\left(\gamma(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathcal{M}\right)^i(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\boldsymbol{\Sigma}\right] \\
\operatorname{Tr}\mathcal{T}_b^{-1}\boldsymbol{\Sigma} &= \frac{1}{2}\operatorname{Tr}\mathbf{H}^{-1}\boldsymbol{\Sigma} + \sum_{i=1}^{\infty}\operatorname{Tr}\left[\left(\gamma(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathcal{M}\right)^i(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\boldsymbol{\Sigma}\right].
\end{aligned}$$

We observe that the term corresponding to $i=0$ works out regardless of the choice of stepsize γ; we then switch our attention to the second term, i.e., the term corresponding to $i=1$:

$$\operatorname{Tr}\left(\gamma(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathcal{M}\right)(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\boldsymbol{\Sigma} = \frac{d+2}{4}\cdot\operatorname{Tr}(\boldsymbol{\Sigma}).$$

We require that this term should be $\leq \operatorname{Tr}(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\boldsymbol{\Sigma}$, implying,

$$\gamma < \frac{4\operatorname{Tr}(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\boldsymbol{\Sigma}}{(d+2)\operatorname{Tr}(\boldsymbol{\Sigma})}.$$

For this example, we observe that this yields $\gamma < \frac{4}{(d+2)(1+\frac{1}{d})}$, which clearly is off by a factor d compared to the well-specified case which requires $\gamma < \frac{d}{(d+2)(1+\frac{1}{d})}$, establishing a clear separation between the step sizes required by SGD for the well-specified and mis-specified cases.

A.5.7 PROOFS OF SUPPORTING LEMMAS

Proof of lemma 7

Proof [Proof of lemma 7] We begin by considering $\langle \mathbf{I}, \mathbb{E}\left[\boldsymbol{\eta}_t^{\text{bias}} \otimes \boldsymbol{\eta}_t^{\text{bias}}\right]\rangle$:

$$\begin{aligned}
\langle \mathbf{I}, \mathbb{E}\left[\boldsymbol{\eta}_t^{\text{bias}} \otimes \boldsymbol{\eta}_t^{\text{bias}}\right]\rangle &= \mathbb{E}\left[\|\boldsymbol{\eta}_t^{\text{bias}}\|^2\right] \\
&= \mathbb{E}\left[(\boldsymbol{\eta}_{t-1}^{\text{bias}})^\top\left(\mathbf{I}-\frac{\gamma}{b}\sum_{i=1}^{b}\mathbf{x}_{ti}\mathbf{x}_{ti}^\top\right)\left(\mathbf{I}-\frac{\gamma}{b}\sum_{i=1}^{b}\mathbf{x}_{ti}\mathbf{x}_{ti}^\top\right)\boldsymbol{\eta}_{t-1}^{\text{bias}}\right] \\
&\leq (1-\gamma\mu)\cdot\mathbb{E}\left[\|\boldsymbol{\eta}_{t-1}^{\text{bias}}\|^2\right] \qquad \text{(from lemma 9)},
\end{aligned}$$

from where the lemma follows through substitution of $\gamma = \gamma_{b,\max}/2$. ■

Proof of lemma 8

Proof [Proof of lemma 8] From equation 27, we have that:

$$\begin{aligned}
\boldsymbol{\Phi}_t^{\text{variance}} &= \mathbb{E}\left[\boldsymbol{\eta}_t^{\text{variance}} \otimes \boldsymbol{\eta}_t^{\text{variance}}\right] \\
&= \frac{\gamma^2}{b}\left(\sum_{k=0}^{t-1}(\mathcal{I}-\gamma\mathcal{T}_b)^k\right)\boldsymbol{\Sigma}.
\end{aligned}$$

Allowing $t\to\infty$, we have:

$$\boldsymbol{\Phi}_\infty^{\text{variance}} = \frac{\gamma}{b}\mathcal{T}_b^{-1}\boldsymbol{\Sigma} \preceq \frac{\gamma}{b}\cdot\sigma^2\mathbf{I} \quad \text{(from claim 4 in lemma 15 since } \gamma \leq \gamma_{b,\max}/2,\ \boldsymbol{\Sigma} = \sigma^2\mathbf{H}).$$

Substituting $\gamma = \gamma_{b,\max}/2$, the result follows. ■

References

Alekh Agarwal, Peter L. Bartlett, Pradeep Ravikumar, and Martin J. Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory*, 2012.

Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *CoRR*, abs/1603.05953, 2016.

Dan Anbar. *On Optimal Estimation Methods Using Stochastic Approximation Procedures*. University of California, 1971.

Francis Bach and Eric Moulines. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Neural Information Processing Systems (NIPS) 24*, 2011.

Francis R. Bach. Adaptivity of averaged stochastic gradient descent to local strong convexity for logistic regression. *In Journal of Machine Learning Research (JMLR)*, volume 15, 2014.

Francis R. Bach and Eric Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate O(1/n). In *Neural Information Processing Systems (NIPS) 26*, 2013.

Rajendra Bhatia. *Positive Definite Matrices*. Princeton Series in Applied Mathematics. Princeton University Press, 2007.

Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In *Neural Information Processing Systems (NIPS) 20*, 2007.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *arXiv preprint arXiv:1606.04838*, 2016.

Joseph K. Bradley, Aapo Kyrola, Danny Bickson, and Carlos Guestrin. Parallel coordinate descent for l1-regularized loss minimization. In *International Conference on Machine Learning (ICML)*, 2011.

Louis Augustin Cauchy. Méthode générale pour la résolution des systémes d'équations simultanees. *C. R. Acad. Sci. Paris*, 1847.

Andrew Cotter, Ohad Shamir, Nati Srebro, and Karthik Sridharan. Better mini-batch algorithms via accelerated gradient methods. In *Neural Information Processing Systems (NIPS) 24*, 2011.

Aaron Defazio. A simple practical accelerated method for finite sums. In *Neural Information Processing Systems (NIPS) 29*, 2016.

Aaron Defazio, Francis R. Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Neural Information Processing Systems (NIPS) 27*, 2014.

Alexandre Défossez and Francis R. Bach. Averaged least-mean-squares: Bias-variance trade-offs and optimal sampling distributions. In *Artifical Intelligence and Statistics (AISTATS)*, 2015.

Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research (JMLR)*, volume 13, 2012.

Aymeric Dieuleveut and Francis Bach. Non-parametric stochastic approximation with large step sizes. *The Annals of Statistics*, 2015.

John C. Duchi, Sorathan Chaturapruek, and Christopher Ré. Asynchronous stochastic convex optimization. *CoRR*, abs/1508.00882, 2015.

Vaclav Fabian. Asymptotically efficient stochastic approximation; the RM case. *Annals of Statistics*, 1(3), 1973.

Roy Frostig, Rong Ge, Sham Kakade, and Aaron Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *International Conference on Machine Learning (ICML)*, 2015a.

Roy Frostig, Rong Ge, Sham M. Kakade, and Aaron Sidford. Competing with the empirical risk minimizer in a single pass. In *Conference on Learning Theory (COLT)*, 2015b.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Prateek Jain, Chi Jin, Sham M. Kakade, Praneeth Netrapalli, and Aaron Sidford. Streaming pca: Matching matrix bernstein and near-optimal finite sample guarantees for oja's algorithm. In *Conference on Learning Theory (COLT)*, 2016a.

Prateek Jain, Sham M Kakade, Rahul Kidambi, Praneeth Netrapalli, and Aaron Sidford. Parallelizing stochastic approximation through mini-batching and tail-averaging. *arXiv preprint arXiv:1610.03774*, 2016b.

Prateek Jain, Sham M Kakade, Rahul Kidambi, Praneeth Netrapalli, Venkata Krishna Pillutla, and Aaron Sidford. A markov chain theory approach to characterizing the minimax optimality of stochastic gradient descent (for least squares). *arXiv preprint arXiv:1710.09430*, 2017a.

Prateek Jain, Sham M Kakade, Rahul Kidambi, Praneeth Netrapalli, and Aaron Sidford. Accelerating stochastic gradient descent. *arXiv preprint arXiv:1704.08227*, 2017b.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Neural Information Processing Systems (NIPS) 26*, 2013.

Harold J. Kushner and Dean S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Springer-Verlag, 1978.

Harold J. Kushner and G. Yin. Asymptotic properties of distributed and communicating stochastic approximation algorithms. *SIAM Journal on Control and Optimization*, 25(5):1266–1290, 1987.

Harold J. Kushner and G. Yin. Stochastic approximation and recursive algorithms and applications. *Springer-Verlag*, 2003.

Erich L. Lehmann and George Casella. *Theory of Point Estimation*. Springer Texts in Statistics. Springer, 1998.

Mu Li, Tong Zhang, Yuqiang Chen, and Alexander J. Smola. Efficient mini-batch training for stochastic optimization. In *Knowledge Discovery and Data Mining (KDD)*, 2014.

Hongzhou Lin, Julien Mairal, and Zaïd Harchaoui. A universal catalyst for first-order optimization. In *Neural Information Processing Systems (NIPS)*, 2015.

Gideon Mann, Ryan T. McDonald, Mehryar Mohri, Nathan Silberman, and Dan Walker. Efficient large-scale distributed training of conditional maximum entropy models. In *Neural Information Processing Systems (NIPS) 22*, 2009.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. Regularizing and optimizing lstm language models. *arXiv preprint arXiv:1708.02182*, 2017.

Deanna Needell, Nathan Srebro, and Rachel Ward. Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. *Mathematical Programming*, volume 155, 2016.

Arkadi S. Nemirovsky and David B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. John Wiley, 1983.

Yurii E. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. *Doklady AN SSSR*, 269, 1983.

Feng Niu, Benjamin Recht, Christopher Re, and Stephen J. Wright. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Neural Information Processing Systems (NIPS) 24*, 2011.

Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4, 1964.

Boris T. Polyak and Anatoli B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM J Control Optim*, volume 30, 1992.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *Ann. Math. Stat.*, vol. 22, 1951.

Jonathan Rosenblatt and Boaz Nadler. On the optimality of averaging in distributed statistical learning. *CoRR*, abs/1407.2724, 2014.

Nicolas Le Roux, Mark Schmidt, and Francis R. Bach. A stochastic gradient method with an exponential convergence rate for strongly-convex optimization with finite training sets. In *Neural Information Processing Systems (NIPS) 25*, 2012.

David Ruppert. Efficient estimations from a slowly convergent robbins-monro process. *Tech. Report, ORIE, Cornell University*, 1988.

Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *CoRR*, abs/1209.1873, 2012.

Shai Shalev-Shwartz and Tong Zhang. Accelerated mini-batch stochastic dual coordinate ascent. In *Neural Information Processing Systems (NIPS) 26*, 2013a.

Shai Shalev-Shwartz and Tong Zhang. Accelerated mini-batch stochastic dual coordinate ascent. In *Neural Information Processing Systems (NIPS) 26*, 2013b.

Samuel L Smith, Pieter-Jan Kindermans, and Quoc V Le. Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*, 2017.

Martin Takác, Avleen Singh Bijral, Peter Richtárik, and Nati Srebro. Mini-batch primal and dual methods for SVMs. In *International Conference on Machine Learning (ICML)*, volume 28, 2013.

Martin Takác, Peter Richtárik, and Nati Srebro. Distributed mini-batch sdca. *CoRR*, abs/1507.08322, 2015.

Aad W. van der Vaart. *Asymptotic Statistics*. Cambridge University Publishers, 2000.

Yuchen Zhang and Lin Xiao. Disco: Distributed optimization for self-concordant empirical loss. In *International Conference on Machine Learning (ICML)*, 2015.

Yuchen Zhang, John C. Duchi, and Martin Wainwright. Divide and conquer ridge regression: A distributed algorithm with minimax optimal rates. *Journal of Machine Learning Research (JMLR)*, volume 16, 2015.

Martin A. Zinkevich, Alex Smola, Markus Weimer, and Lihong Li. Parallelized stochastic gradient descent. In *Neural Information Processing Systems (NIPS) 24*, 2011.

Journal of Machine Learning Research 18 (2018) 1-44 Submitted 4/17; Revised 12/17; Published 4/18

Learning Quadratic Variance Function (QVF) DAG Models via OverDispersion Scoring (ODS)

Gunwoong Park gwpark23@uos.ac.kr
Department of Statistics
University of Seoul
Seoul, 02592, South Korea

Garvesh Raskutti raskutti@stat.wisc.edu
Department of Statistics
Department of Computer Science
Wisconsin Institute for Discovery, Optimization Group
University of Wisconsin
Madison, WI 53706, USA

Editor: Hui Zou

Abstract

Learning DAG or Bayesian network models is an important problem in multi-variate causal inference. However, a number of challenges arises in learning large-scale DAG models including model identifiability and computational complexity since the space of directed graphs is huge. In this paper, we address these issues in a number of steps for a broad class of DAG models where the noise or variance is signal-dependent. Firstly we introduce a new class of identifiable DAG models, where each node has a distribution where the variance is a quadratic function of the mean (QVF DAG models). Our QVF DAG models include many interesting classes of distributions such as Poisson, Binomial, Geometric, Exponential, Gamma and many other distributions in which the noise variance depends on the mean. We prove that this class of QVF DAG models is identifiable, and introduce a new algorithm, the OverDispersion Scoring (ODS) algorithm, for learning large-scale QVF DAG models. Our algorithm is based on firstly learning the moralized or undirected graphical model representation of the DAG to reduce the DAG search-space, and then exploiting the quadratic variance property to learn the ordering. We show through theoretical results and simulations that our algorithm is statistically consistent in the high-dimensional $p > n$ setting provided that the degree of the moralized graph is bounded and performs well compared to state-of-the-art DAG-learning algorithms. We also demonstrate through a real data example involving multi-variate count data, that our ODS algorithm is well-suited to estimating DAG models for count data in comparison to other methods used for discrete data.

Keywords: Bayesian Networks, Directed Acyclic Graph, Identifiability, Multi-variate Count Distribution, Overdispersion

1. Introduction

Probabilistic directed acyclic graphical (DAG) models or Bayesian networks provide a widely used framework for representing causal or directional dependence relationships amongst multiple variables. DAG models have applications in various areas including genomics, neuroimaging, statistical physics, spatial statistics and many others (see e.g., Doya 2007; Friedman et al. 2000; Kephart and White 1991). One of the fundamental problems associated with DAG models or Bayesian networks is structure learning from observational data.

If the number of variables is large, a number of challenges arise that make learning large-scale DAG models extremely difficult even when variables have a natural causal or directional structure. These challenges include: (1) identifiability since inferring causal directions from only observational data is in general not possible in the absence of additional assumptions; (2) computational complexity since it is NP-hard to search over the space of DAGs (Chickering, 1996); (3) providing sample size guarantee in the setting where the number of nodes p is large. In this paper we develop a general framework and algorithm for learning large-scale DAG models that addresses these challenges in a number of steps: Firstly, we introduce a new class of provably identifiable DAG models where each node has a conditional distribution where the variance is a quadratic function of the mean, which we refer to as QVF (quadratic variance function) distributions; secondly, we introduce a general OverDispersion Scoring (ODS) algorithm for learning large-scale QVF DAG models; thirdly, we provide theoretical guarantees for our ODS algorithm which proves that our algorithm is consistent in the high-dimensional setting $p > n$ provided that the moralized graph of the DAG is sparse; and finally, we show through a simulation study that our ODS algorithm supports our theoretical result has favorable performance to a number of state-of-the-art algorithms for learning both low-dimensional and high-dimensional DAG models.

Our algorithm is based on combining two ideas: *overdispersion* and *moralization*. Overdispersion is a property of Poisson and other random variables where the variance depends on the mean and we use overdispersion to address the identifiability issue. While overdispersion is a known phenomena used and exploited in many applications (see e.g., Dean 1992; Zheng et al. 2006), overdispersion has never been exploited for learning DAG models aside from our prior work (Park and Raskutti, 2015) which focuses on Poisson DAG models. In this paper, we show that overdispersion applies much more broadly and is used to prove identifiability for a broad class of DAG models. To provide a scalable algorithm with statistical guarantees, even in the high-dimensional setting, we exploit the moralized graph, that is the undirected representation of the DAG. Learning the moralized graph allows us to exploit sparsity and considerably reduces the DAG search-space which has both computational and statistical benefits. Furthermore, moralization allows us to use existing scalable algorithms and theoretical guarantees for learning large-scale undirected graphical models (e.g., Friedman et al. 2009; Yang et al. 2012).

A number of approaches have been used to address the identifiability challenge by imposing additional assumptions. For example ICA-based methods for learning ordering requires independent noise and non-Gaussianity (see e.g., Shimizu et al. 2006), structural equation models with Gaussian noise with equal or known variances (Peters et al., 2012), and non-parametric structural equation models with independent noise (see e.g., Peters and Bühlmann 2013). These approaches are summarized elegantly in an information-theoretic framework in Janzing and Scholkopf (2010). Our approach is along similar lines in that we impose overdispersion as an additional assumption which induces asymmetry and guarantees identifiability. However by exploiting overdispersion, our approach applies when the noise distribution of each node depends on its mean whereas prior approaches apply when

the additive noise variance is independent of the mean. Additionally, we exploit graph sparsity which has also been exploited in prior work by Loh and Bühlmann (2014); Raskutti and Uhler (2013); van de Geer and Bühlmann (2013) for various DAG models with independent additive noise components. Furthermore, sparsity allows us to develop a tractable algorithm where we reduce the DAG space by learning the moralized graph, an idea which has been used in prior work in Tsamardinos and Aliferis (2003).

1.1 Our Contributions

We summarize the major contributions of the paper as follows:

- We introduce the class of QVF DAG models, that include many interesting classes of multi-variate distributions and provide conditions under which QVF DAG models are identifiable.
- Using QVF DAG models, we develop the reliable and scalable generalized ODS algorithm which learns any large-scale QVF DAG model. Our algorithm combines two key ideas, moralization and overdispersion. Moralization significantly reduces computational complexity by exploiting sparsity of the moralized graph, while overdispersion exploits properties of QVF DAG models to estimate the causal ordering. The generalized ODS algorithm adapts the algorithm developed in Park and Raskutti (2015) to general QVF DAG models whilst the algorithm in Park and Raskutti (2015) focuses exclusively on Poisson DAG models.
- We provide statistical guarantees to show that our ODS algorithm is consistent for learning QVF DAG models, even in the high-dimensional $p > n$ setting, provided that the degree of the moralized graph is bounded. To the best of our knowledge, this is the only theoretical result that applies to the high-dimensional setting when the variables at each node model counts.
- We demonstrate through simulation studies and a real data application involving multi-variate count data that our ODS algorithm performs favorably compared to the state-of-the-art GES and MMHC algorithms. In our simulation study, we consider both the low-dimensional and high-dimensional setting. Our real data example involving NBA player statistics for 2009/10 season shows that our ODS algorithm is applicable to multi-variate count data while the GES and MMHC algorithms tend to select very few edges when variables represent counts.

The remainder of the paper is organized as follows: In Section 2, we define QVF DAG models and prove identifiability for this class of models. In Section 3, we introduce our polynomial-time DAG learning algorithm which we refer to as the generalized OverDispersion Scoring (ODS). Statistical guarantees for learning QVF DAG models using our ODS algorithm are provided in Section 3.2, and we provide numerical experiments on both small DAGs and large-scale DAGs with node-size up to 5000 nodes in Section 4. Our theoretical guarantees in Section 3.2 prove that even in the setting where the number of nodes p is larger than the sample size n, it is possible to learn the DAG structure under the assumption that the degree d of the so-called moralized graph of the DAG is small. Our numerical experiments in Section 4 support the theoretical results and show that our algorithm performs

well compared to other state-of-the-art DAG learning methods. Our numerical experiments confirm that our algorithm is one of the few DAG-learning algorithms that performs well in terms of statistical and computational complexity in high-dimensional $p > n$ settings, provided that the degree of the moralized graph d is bounded. Finally in Section 5 we show that our ODS algorithm performs well in terms of the eye test compared to state-of-the-art algorithms for a multi-variate count data set that involves basketball statistics.

2. Quadratic Variance Function (QVF) DAG Models and Identifiability

A DAG $G = (V, E)$ consists of a set of nodes V and a set of directed edges $E \in V \times V$ where each $e \in E$ is an ordered pair of distinct nodes. Hence our graphs are *simple*, i.e., there are no directed cycles or multiple edges between any pair of nodes. A directed edge from node j to k is denoted by (j, k) or $j \to k$. The set of *parents* of node k denoted by $\text{pa}(k)$ consists of all nodes j such that $(j, k) \in E$. If there is a directed path $j \to \cdots \to k$, then k is called a *descendant* of j and j is an *ancestor* of k. The set $\text{de}(k)$ denotes the set of all descendants of node k. The *non-descendants* of node k are $\text{nd}(k) := V \setminus (\{k\} \cup \text{de}(k))$. An important property of DAGs is that there exists an (possibly non-unique) *ordering* π^* of a directed graph that represents directions of edges such that for every directed edge $(j, k) \in E$, j comes before k in the ordering. Without loss of generality, we set $V = \{1, 2, \cdots, p\}$ and assume the true ordering is $\pi^* = (1, 2, \cdots, p)$.

We consider a set of random variables $X := (X_j), j \in V$ with probability distribution $\mathbb{P}$ taking values in probability space $\mathcal{X}_v$ over the nodes in G. Suppose that a random vector X has joint probability density function $f_G(X)$. For any subset S of V, let $X_S := \{X_s : s \in S \subset V\}$ and $\mathcal{X}(S) := \times_{j \in S} \mathcal{X}_j$. For $j \in V$, $f_j(X_j \mid X_S)$ denotes the conditional distribution of a random variable X_v given a random vector X_S. Then, a probabilistic DAG model has the following factorization (Lauritzen, 1996):

$$f_G(X) = \prod_{j \in V} f_j(X_j \mid X_{\text{pa}(j)}), \tag{1}$$

where $f_j(X_j \mid X_{\text{pa}(j)})$ refers to the conditional distribution of a random variable X_j in terms of its parents $X_{\text{pa}(j)} := \{X_s : s \in \text{pa}(j)\}$.

A core concept in this paper is *identifiability* for a family of probability distributions defined by the DAG factorization provided above. Intuitively identifiability addresses the question of what assumption we make on the conditional distributions $f_j(X_j \mid X_{\text{pa}(j)})$ allows us to uniquely determine the structure of that DAG G given the joint PDF $f_G(X)$.

To define identifiability precisely, let $\mathcal{P}$ denote the set of *conditional distributions* $f_j(X_j \mid X_{\text{pa}(j)})$ for all $j \in V$. Further for a graph $G = (V, E)$, define the class of *joint distributions* with respect to graph G and class of distributions $\mathcal{P}$ by

$$\mathcal{F}(G; \mathcal{P}) := \{f_G(X) = \prod_{j \in V} f_j(X_j \mid X_{\text{pa}(j)}) \text{ ; where } f_j(X_j \mid X_{\text{pa}(j)}) \in \mathcal{P} \; \forall \; j \in V\}.$$

Next let $\mathcal{G}_p$ denote the set of p-node directed acyclic graphs. Now we define identifiability for the class $\mathcal{P}$ over the space of DAGs $\mathcal{G}_p$.

Definition 1 (Identifiability) *A class of conditional distributions $\mathcal{P}$ is* identifiable *over $\mathcal{G}_p$ if $G \neq G'$ where $G, G' \in \mathcal{G}_p$, there exists no $f_G \in \mathcal{F}(G;\mathcal{P})$ and $f_{G'} \in \mathcal{F}(G';\mathcal{P})$ such that $f_G = f_{G'}$.*

Prior work has addressed the question of identifiability for different classes of $\mathcal{P}$. For example ICA-based methods make the assumption that $\mathcal{P}$ are independent error with non-Gaussian components (Shimizu et al., 2006) and prove that this class is identifiable as well as $\mathcal{P}$ corresponding to a non-parametric model with additive independent noise (Peters and Bühlmann, 2013), $\mathcal{P}$ represents structural linear equation models with Gaussian errors with equal or known variances (Peters et al., 2012). On the other hand, if $\mathcal{P}$ represents structural linear equation models with Gaussian errors with general variance is not identifiable and only the Markov equivalence class of DAG models is identifiable (Heckerman et al., 1995).

The main results of our paper give another class of identifiable graphical models. In our setting, $\mathcal{P}$ is a setting where the variance is a linear function of the mean so we deal with signal-dependent noise or variance, and more importantly is applicable for discrete distributions. We define $\mathcal{P}$ more precisely in the next section.

2.1 Quadratic Variance Function (QVF) DAG Models

Now we define quadratic variance function (QVF) DAG models. For QVF DAG models each node has a conditional distribution $\mathcal{P}$ given its parents with the property that the variance is a quadratic function of the mean. More precisely,

Definition 2 (QVF DAG models) *Quadratic variance function (QVF) DAG models are DAG models where conditional distribution of each node given its parents satisfies the* quadratic variance function property*: for all $j \in V$, there exist $\beta_{j0}, \beta_{j1} \in \mathbb{R}$ such that*

$$Var(X_j \mid X_{pa(j)}) = \beta_{j0}\mathbb{E}(X_j \mid X_{pa(j)}) + \beta_{j1}\mathbb{E}(X_j \mid X_{pa(j)})^2. \tag{2}$$

To the best of our knowledge, quadratic variance function (QVF) probability distributions were first introduced in the context of natural parameter exponential families (NEF) (Morris, 1982) which include Poisson, Binomial, Negative Binomial and Gamma distributions. In the directed graphical model framework, each node distribution is influenced by its parents. Hence for natural exponential families with quadratic variance functions (NEF-QVF), we provide an explicit form of joint distributions for DAG models.

For NEF-QVF, the conditional distribution of each node given its parents takes the simple form:

$$P(X_j \mid X_{\mathrm{pa}(j)}) = \exp\left(\theta_{jj}X_j + \sum_{(k,j)\in E}\theta_{jk}X_kX_j - B_j(X_j) - A_j\left(\theta_{jj} + \sum_{(k,j)\in E}\theta_{jk}X_k\right)\right)$$

where $A_j(\cdot)$ is the log-partition function, $B_j(\cdot)$ is determined by a chosen exponential family, and $\theta_{jk} \in \mathbb{R}$ is a parameter corresponding to a node j. By the factorization property (1), the joint distribution of a NEF-QVF DAG model takes the following form:

$$P(X) = \exp\left(\sum_{j\in V}\theta_{jj}X_j + \sum_{(k,j)\in E}\theta_{jk}X_kX_j - \sum_{j\in V}B_j(X_j) - \sum_{j\in V}A_j\left(\theta_{jj} + \sum_{(k,j)\in E}\theta_{jk}X_k\right)\right). \tag{3}$$

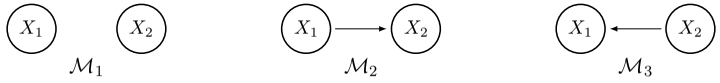


Figure 1: Directed graphical models of $\mathcal{M}_1$, $\mathcal{M}_2$ and $\mathcal{M}_3$

From Equation (3), we provide examples of classes of NEF-QVF DAG models. For Poisson DAG models studied in Park and Raskutti (2015) the log-partition function $A_j(\cdot) = \exp(\cdot)$, and $B_j(\cdot) = \log(\cdot!)$. Similarly, Binomial DAG models can be derived as an example of QVF DAG models where the conditional distribution for each node is binomial with known parameter N_j and the log-partition function $A_j(\cdot) = N_j\log(1+\exp(\cdot))$, and $B_j(\cdot) = -\log\binom{N_j}{\cdot}$. Another interesting instance is Exponential DAG models where each node conditional distribution given its parents is Exponential. Then, $A_j(\cdot) = -\log(-\cdot)$ and $B_j(\cdot) = 0$.

Our framework also naturally extends to mixed DAG models, where the conditional distributions have different distributions which incorporates different data types. In addition, our models extend to nonlinear and nonparametric DAG models depending on the data as long as the node distribution $\mathcal{P}$ satisfies the QVF property in Equation (2). This means our model is identifiable without information on how a node and its parents are related. In Section 4, we will provide numerical experiments on Poisson and Binomial DAG models.

2.2 Identifiability of QVF DAG Models

In this section we prove that QVF DAG models are identifiable. To provide intuition, we prove identifiability for the two-node Poisson DAG model in Park and Raskutti (2015). Consider all three models illustrated in Figure 1: $\mathcal{M}_1 : X_1 \sim \mathrm{Poisson}(\lambda_1)$, $X_2 \sim \mathrm{Poisson}(\lambda_2)$, where X_1 and X_2 are independent; $\mathcal{M}_2 : X_1 \sim \mathrm{Poisson}(\lambda_1)$ and $X_2 \mid X_1 \sim \mathrm{Poisson}(g_2(X_1))$; and $\mathcal{M}_3 : X_2 \sim \mathrm{Poisson}(\lambda_2)$ and $X_1 \mid X_2 \sim \mathrm{Poisson}(g_1(X_2))$ for arbitrary positive functions $g_1, g_2 : \mathbb{N}\cup\{0\} \to \mathbb{R}^+$. Our goal is to determine whether the underlying DAG model is $\mathcal{M}_1, \mathcal{M}_2$ or $\mathcal{M}_3$.

We exploit the equidispersion property that for a Poisson random variable X, $\mathrm{Var}(X) = \mathbb{E}(X)$, while for a distribution which is conditionally Poisson, the marginal variance is overdispersed relative to the marginal expectation, $\mathrm{Var}(X) > \mathbb{E}(X)$. Hence for $\mathcal{M}_1$, $\mathrm{Var}(X_1) = \mathbb{E}(X_1)$ and $\mathrm{Var}(X_2) = \mathbb{E}(X_2)$. For $\mathcal{M}_2$, $\mathrm{Var}(X_1) = \mathbb{E}(X_1)$, while

$$\mathrm{Var}(X_2) = \mathbb{E}(\mathrm{Var}(X_2 \mid X_1)) + \mathrm{Var}(\mathbb{E}(X_2 \mid X_1)) = \mathbb{E}(\mathbb{E}(X_2 \mid X_1)) + \mathrm{Var}(g_2(X_1)) > \mathbb{E}(X_2),$$

as long as $\mathrm{Var}(g_2(X_1)) > 0$. The first equality follows from the total variance decomposition and the second equality follows from the equidispersion property of Poisson distribution.

Similarly under $\mathcal{M}_3$, $\mathrm{Var}(X_2) = \mathbb{E}(X_2)$ and $\mathrm{Var}(X_1) > \mathbb{E}(X_1)$ as long as $\mathrm{Var}(g_1(X_2)) > 0$. Hence we can distinguish models $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$ by testing whether the variance is greater than or equal to the expectation. With finite samples, the quantities $\mathbb{E}(\cdot)$ and $\mathrm{Var}(\cdot)$ can be estimated from data and we describe this more precisely in Sections 3 and 3.2.

For general QVF DAG models, the variance for each node distribution is not necessarily equal to the mean. Hence we introduce a linear transformation $T_j(X_j)$ such that

Distribution	$\mathcal{P}$	β_0	β_1	ω
Binomial	Bin(N,p)	1	$-\frac{1}{N}$	$\frac{N}{N-\mu}$
Poisson	Poi(λ)	1	0	1
Geometric	Geo(p)	1	1	$\frac{1}{1+\mu}$
Negative Binomial	NB(R,p)	1	$\frac{1}{R}$	$\frac{R}{R+\mu}$
Exponential	Exp(λ)	0	1	$\frac{1}{\mu}$
Gamma	Gamma(α,β)	0	$\frac{1}{\alpha}$	$\frac{\alpha}{\mu}$

Table 1: Examples of distributions for QVF DAG models with β_0, β_1 and ω where μ is its expectation

$\mathrm{Var}(T_j(X_j) \mid X_{\mathrm{pa}(j)}) = \mathbb{E}(T_j(X_j) \mid X_{\mathrm{pa}(j)})$ in Proposition 3. This transformation enables us to use the notion of *overdispersion* for recovering QVF DAG models. We present examples of distributions $\mathcal{P}$ for QVF DAG models with the triple $(\beta_0, \beta_1, \omega)$ in Table 1.

Proposition 3 *Let $X = (X_1, X_2, \cdots, X_p)$ be a random vector associated with a QVF DAG model with quadratic variance coefficients $(\beta_{j0}, \beta_{j1})_{j=1}^p$ specified in Equation (2). Then, there exists a transformation $T_j(X_j) = \omega_j X_j$ for any node $j \in V$ where $\omega_j = (\beta_{j0} + \beta_{j1}\mathbb{E}(X_j \mid X_{pa(j)}))^{-1}$ such that*

$$Var(T_j(X_j) \mid X_{pa(j)}) = \mathbb{E}(T_j(X_j) \mid X_{pa(j)}).$$

Proof For any node $j \in V$,

$$\begin{aligned} \mathrm{Var}(\omega_j X_j \mid X_{\mathrm{pa}(j)}) &= \omega_j^2 \mathrm{Var}(X_j \mid X_{\mathrm{pa}(j)}) \\ &\overset{(a)}{=} \omega_j^2(\beta_{j0}\mathbb{E}(X_j \mid X_{\mathrm{pa}(j)}) + \beta_{j1}\mathbb{E}(X_j \mid X_{\mathrm{pa}(j)})^2) \\ &\overset{(b)}{=} \omega_j \mathbb{E}(X_j \mid X_{\mathrm{pa}(j)}) \\ &= \mathbb{E}(\omega_j X_j \mid X_{\mathrm{pa}(j)}). \end{aligned}$$

Equality (a) follows from the quadratic variance property (2), and (b) follows from the definition of ω_j. ■

Now we extend to general p-variate QVF DAG models. The key idea to extending identifiability from the bivariate to multivariate scenario involves conditioning on parents of each node, and then testing overdispersion.

Assumption 4 *For all $j \in V$ and any $pa_0(j) \subset pa(j)$ where $pa_0(j) \neq \emptyset$ and $S \subset nd(j) \setminus pa_0(j)$:*

(a) $Var(\mathbb{E}(X_j \mid X_{pa(j)}) \mid X_S) > 0$, and

(b) $\beta_{j0} + \beta_{j1}\mathbb{E}(X_j \mid X_S) \neq 0$.

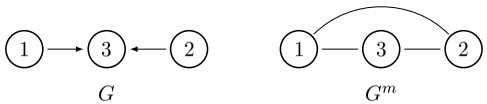


Figure 2: Moralized graph G^m for DAG G

Assumption 4(a) ensures that all parents of node j contribute to its variability, hence a conditional variance is bigger than the conditional expectation. Assumption 4(b) rules out the extremely skewed distributions. For example, Binomial distribution with a parameter N has $\beta_0 = 1$ and $\beta_1 = -\frac{1}{N}$. Then, the assumption is satisfied as long as the conditional expectation is less than N. Similarly Exponential distribution has $\beta_0 = 0$ and $\beta_1 = 1$, hence the assumption is satisfied as long as the conditional expectation is positive. Poisson distribution has $\beta_0 = 1$ and $\beta_1 = 0$, so the assumption is always satisfied.

Theorem 5 (Identifiability for p-variate QVF DAG models) *Consider the class of QVF DAG models (1) with quadratic variance coefficients $(\beta_{j0}, \beta_{j1})_{j=1}^p$ (2). If for all $j \in V$, $\beta_{j1} > -1$ and Assumption 4 is satisfied, then the class of QVF DAG models is identifiable according to Def. 1.*

The proof is provided in Appendix A. Theorem 5 shows that any QVF DAG model is identifiable under the assumption that all parents of node j contribute to its variability. The condition $\beta_{j1} > -1$ rules out DAG models with Bernoulli and multinomial distributions which are known to be non-identifiable (Heckerman et al., 1995) with $\beta_{j1} = -1$.

3. OverDispersion Scoring (ODS) Algorithm

In this section, we present our generalized OverDispersion Scoring (ODS) algorithm. The ODS algorithm is introduced by Park and Raskutti (2015) for Poisson DAG models, however it does not cover other QVF distributions. In this paper, we generalize the ODS algorithm for recovering QVF DAG models. An important concept we need to introduce for the generalized ODS algorithm is the *moral* graph or undirected graphical model representation of a DAG (see e.g., Cowell et al. 1999). The moralized graph G^m for a DAG $G = (V, E)$ is an undirected graph where $G^m = (V, E^m)$ where E^m includes the edge set E for the DAG G with directions removed plus edges between any nodes that are parents of a common child. Figure 2 represents the moralized graph for a simple 3-node example where $E = \{(1,3),(2,3)\}$ for the DAG G. Since nodes 1 and 2 are parents with a common child 3, the additional edge $(1,2)$ arises, and therefore $E^m = \{(1,2),(1,3),(2,3)\}$. Finally, the *neighborhood* for a node j refers to the adjacent nodes to j in the moralized graph, and is denoted by $\mathcal{N}(j) := \{k \in V \mid (j,k) \text{ or } (k,j) \in E^m\}$.

Our generalized ODS algorithm (Algorithm 1) has three main steps: 1) estimate the moralized graph G^m for the DAG G; 2) estimate the ordering of the DAG G using overdispersion scoring based on the moralized graph from step 1); and 3) estimate the DAG structure, given the ordering from step 2). Although Steps 2) and 3) are sufficient to recover

Algorithm 1: Generalized OverDispersion Scoring (ODS)

Input : n i.i.d. samples from a QVF-DAG model

Output: Estimated ordering $\widehat{\pi} \in \mathbb{N}^p$ and an edge structure, $\widehat{E} \in V \times V$

Step 1: Estimate the undirected edges $\widehat{E}^m = \cup_{j \in V} \cup_{k \in \widehat{\mathcal{N}}(j)} (j,k)$ where $\widehat{\mathcal{N}}(j)$ is estimated neighborhood set of a node j in the moralized graph;

Step 2: Estimate the ordering using overdispersion scores;

for $j \in \{1, 2, \cdots, p\}$ **do**

  Calculate overdispersion scores $\widehat{\mathcal{S}}(1, j)$ using Equation (4);

end

The first element of the ordering $\widehat{\pi}_1 = \arg\min_j \widehat{\mathcal{S}}(1, j)$;

for $m = \{2, 3, \cdots, p-1\}$ **do**

  for $j \in \{1, 2, \cdots, p\} \setminus \{\widehat{\pi}_1, \widehat{\pi}_2, \cdots, \widehat{\pi}_{m-1}\}$ **do**

    Find candidate parents set $\widehat{C}_{mj} = \widehat{\mathcal{N}}(j) \cap \{\widehat{\pi}_1, \widehat{\pi}_2, \cdots, \widehat{\pi}_{m-1}\}$;

    Calculate overdispersion scores $\widehat{\mathcal{S}}(m, j)$ using Equation (5);

  end

  The m^{th} element of an ordering $\widehat{\pi}_m = \arg\min_j \widehat{\mathcal{S}}(m, j)$;

  Step 3: Estimate the directed edges toward $\widehat{\pi}_m$, denoted by $\widehat{D}_m$;

end

The last element of the ordering $\widehat{\pi}_p = \{1, 2, \cdots, p\} \setminus \{\widehat{\pi}_1, \widehat{\pi}_2, \cdots, \widehat{\pi}_{p-1}\}$;

The directed edges toward $\widehat{\pi}_p$, denoted by $\widehat{D}_p = \{(j, \widehat{\pi}_p) \mid j \in \widehat{\mathcal{N}}(\widehat{\pi}_p)\}$;

Return: $\widehat{\pi} = (\widehat{\pi}_1, \widehat{\pi}_2, \cdots, \widehat{\pi}_p)$, and $\widehat{E} = \cup_{m=\{2,3,\cdots,p\}} \widehat{D}_m$

DAG structures, Step 1) is performed because it reduces both computational and sample complexity by exploiting the sparsity of the moralized graph for the DAG.

The main purpose of Step 1) is to reduce the search-space by exploiting sparsity of the moralized graph. The moralized graph provides a *candidate parents* set for each node. Similar ideas of reducing the search-space by utilizing the moralized graph or different undirected graphs are applied in existing algorithms (e.g., Tsamardinos and Aliferis 2003; Friedman et al. 1999; Loh and Bühlmann 2014). The concept of candidate parents set exploits two properties; (i) the neighborhood of a node is a superset of its parents, and (ii) a node should appear later than its parents in the ordering. Hence, the candidate parents set for a given node j is the intersection of its neighborhood and elements of the ordering which appear before that node j, and is denoted by $C_{mj} := \mathcal{N}(j) \cap \{\pi_1, \pi_2, ..., \pi_{m-1}\}$ where m^{th} element of the ordering is j (i.e., $\pi_m = j$). The estimated candidate parents set is $\widehat{C}_{mj} := \widehat{\mathcal{N}}(j) \cap \{\widehat{\pi}_1, \widehat{\pi}_2, ..., \widehat{\pi}_{m-1}\}$ that is also specified in Algorithm 1.

This candidate parents set is used as a conditioning set for the overdispersion score in Step 2). In principle, the size of the conditioning set for an overdispersion score could be $p-1$ if the moralized graph is not used. Since Step 2) requires computation of a conditional mean and variance, both the computational complexity and sample complexity depend significantly on the number of variables we condition on as illustrated in Sections 3.1

and 3.2. Therefore by making the conditioning set for the overdispersion score of each node as small as possible, we gain significant computational and statistical improvements.

A number of choices are available for estimation of the moralized graph. Since the moralized graph is an undirected graph, standard undirected graph learning algorithms such as HITON (Aliferis et al., 2003) and MMPC algorithms (Tsamardinos and Aliferis, 2003) as well as ℓ_1-penalized likelihood regression for generalized linear models (GLM) (Friedman et al., 2009). In addition, standard DAG learning algorithms such as PC (Spirtes et al., 2000), GES (Chickering, 2003) and MMHC algorithms (Tsamardinos and Aliferis, 2003) can be applied to estimate the Markov equivalence class and then the moralized graph is generated from the Markov equivalence class.

Step 2) of the generalized ODS algorithm involves learning the ordering by comparing overdispersion scores of nodes using Equations (4) and (5). The basic idea is to determine which nodes are overdispersed based on the sample conditional mean and conditional variance after the transformation in Proposition 3. The ordering is determined one node at a time by selecting the node with the smallest overdispersion score which is representative of a node that is least likely to be overdispersed.

Regarding the overdispersion scores, suppose that there are n i.i.d. samples $X^{1:n} := (X^{(i)})_{i=1}^n$ where $X^{(i)} := (X_1^{(i)}, X_2^{(i)}, \cdots, X_p^{(i)})$ is a p-variate random vector drawn from an underlying QVF DAG model with quadratic variance coefficients $(\beta_{j0}, \beta_{j1})_{j=1}^p$. We use the notation $\widehat{\cdot}$ to denote an estimate based on $X^{1:n}$. In addition, we use $n(x_S) := \sum_{i=1}^n \mathbf{1}(X_S^{(i)} = x_S)$ for $x_S \in \mathcal{X}(S)$ to denote the conditional sample size, and $n_S := \sum_{x_S} n(x_S)\mathbf{1}(n(x_S) \geq c_0 \cdot n)$ for an arbitrary $c_0 \in (0,1)$ to denote a truncated conditional sample size. We discuss the choice of c_0 shortly.

More precisely the overdispersion scores in Step 2) of Algorithm 1 involves the following equations:

$$\widehat{\mathcal{S}}(1,j) \;:=\; \widehat{\omega}_j^2 \cdot \widehat{\mathrm{Var}}(X_j) - \widehat{\omega}_j \cdot \widehat{\mathbb{E}}(X_j) \quad \text{where} \quad \widehat{\omega}_j := (\beta_{10} + \beta_{11}\widehat{\mathbb{E}}(X_j))^{-1}, \tag{4}$$

$$\widehat{\mathcal{S}}(m,j) \;:=\; \sum_{x \in \mathcal{X}_{\widehat{C}_{mj}}} \frac{n(x)}{n_{\widehat{C}_{mj}}} \Big[\widehat{\omega}_{mj}(x)^2 \widehat{\mathrm{Var}}(X_j \mid X_{\widehat{C}_{mj}} = x) - \widehat{\omega}_{mj}(x)\widehat{\mathbb{E}}(X_j \mid X_{\widehat{C}_{mj}} = x)\Big] \tag{5}$$

where $\widehat{\omega}_{mj}(x) := (\beta_{j0} + \beta_{j1}\widehat{\mathbb{E}}(X_j \mid X_{\widehat{C}_{mj}} = x))^{-1}$. $\widehat{C}_{mj}$ is the estimated candidate parents set of node j for the m^{th} element of the ordering and $\mathcal{X}_{\widehat{C}_{mj}} := \{x_{\widehat{C}_{mj}} \in \mathcal{X}(\widehat{C}_{mj}) : n(x_{\widehat{C}_{mj}}) \geq c_0 \cdot n\}$ to ensure we have enough samples for each element of an overdispersion score. c_0 is a tuning parameter of our algorithm that we specify in Theorem 14 and our numerical experiments. $\widehat{\omega}_{mj}(x)$ is an empirical version of the transformation in Proposition 3 assuming $\widehat{C}_{mj}$ is the parents of a node j. Since there are many conditional distributions, our overdispersion score is the weighted average of differences between conditional sample means and variances after the estimated transformation $\widehat{\omega}_{mj}(x)$. Then, the score is a measure of the level of overdispersion. As demonstrated in Section 2.2, the correct elements of an ordering achieve zero overdispersion score, otherwise positive in population.

Finding the set of parents of a node j boils down to selecting the parents out of all elements before a node j in the ordering. Hence given the estimated ordering from Step 2),

Step 3) can be reduced to p neighborhood selection problems which can be performed using ℓ_1-penalized likelihood regression for GLMs (Friedman et al., 2009) as well as standard DAG learning algorithms such as the PC (Spirtes et al., 2000), GES (Chickering, 2003), and MMHC algorithms (Tsamardinos and Aliferis, 2003).

3.1 Computational Complexity

For Steps 1) and 3) of the generalized ODS algorithm, we use off-the-shelf algorithms and the computational complexity depends on the choice of algorithm. For example, if we use the neighborhood selection ℓ_1-penalized likelihood regression for GLMs (Friedman et al., 2009) as is used in Yang et al. (2012), the worst-case complexity is $O(\min(n,p)np)$ for a single ℓ_1-penalized likelihood regression, but since there are p nodes, the total worst-case complexity is $O(\min(n,p)np^2)$. Similarly, if we use ℓ_1-penalized likelihood regression for Step 3) the worst-case complexity is also $O(\min(n,p)np^2)$ but maybe less if the degree d of the moralized graph is small.

For Step 2) where we estimate the ordering, there are $(p-1)$ iterations and each iteration has a number of overdispersion scores $\widehat{\mathcal{S}}(m,j)$ to be computed which is bounded by $O(p)$. Hence the total number of overdispersion scores that need to be computed is $O(p^2)$. Since the time for calculating each overdispersion score is proportional to the sample size n, the complexity is $O(np^2)$.

Hence, Step 1) is the main computational bottleneck of the generalized ODS algorithm. The addition of Step 2) which estimates the ordering does not significantly add to the computational bottleneck. Consequently, the generalized ODS algorithm, which is designed for learning DAGs is almost as computationally efficient as standard methods for learning undirected graphical models. As we show in numerical experiments, the ODS algorithm using ℓ_1-penalized likelihood regression for GLMs in both Steps 1) and 3) is faster than the state-of-the-art GES algorithm.

3.2 Statistical Guarantees

In this section, we provide theoretical guarantees for our generalized ODS algorithm. We provide sample complexity guarantees for the algorithm in the high-dimensional setting in three steps, by proving consistency of Steps 1), 2) and 3) in Sections 3.2.1, 3.2.2 and 3.2.3, respectively. All three main results are expressed in terms of the triple (n,p,d).

Although any off-the-shelf algorithms can be used in Steps 1) and 3), our theoretical guarantees focus on the case when we use the R package glmnet (Friedman et al., 2009) for neighborhood selection. We focus on glmnet since there exist provable theoretical guarantees for neighborhood selection for graphical model learning in the high-dimensional setting (see e.g., Yang et al. 2012; Ravikumar et al. 2010) and performs well in our simulation study. The glmnet package involves minimizing the ℓ_1-penalized generalized linear model loss.

Without loss of generality, assume that $(1,2,\cdots,p)$ is the true ordering and for ease of notation let $[\cdot]_k$ and $[\cdot]_S$ denotes parameter(s) corresponding to the variable X_k and random vector X_S, respectively. Suppose that $\theta^*_{D_j} \in \Theta_{D_j}$ denotes the solution of the following GLM problem where $\Theta_{D_j} := \{\theta \in \mathbb{R}^p : [\theta]_k = 0 \text{ for } k \notin \text{pa}(j)\}$.

$$\theta^*_{D_j} := \arg\min_{\theta\in\Theta_{D_j}} \mathbb{E}\left(-X_j([\theta]_j + \langle [\theta]_{\text{pa}(j)}, X_{\text{pa}(j)}\rangle) + A_j([\theta]_j + \langle [\theta]_{\text{pa}(j)}, X_{\text{pa}(j)}\rangle)\right), \quad (6)$$

where $A_j(\cdot)$ is the log-partition function determined by the GLM family (3), and $\langle\cdot,\cdot\rangle$ represents the inner product. In the special case where X_j has an NEF-QVF distribution (3) with log-partition function $A_j(\cdot)$, $\theta^*_{D_j}$ corresponds exactly to the set of true parameters, that is θ^*_{jk} is the coefficient $k \in \text{pa}(j)$ which represents the influence of node k on node j. However our results apply more generally and we do not require that X_j belongs to an NEF-QVF DAG model.

Similar definitions are required for parameters associated with the moralized graph G^m. Define $\theta^*_{M_j} \in \Theta_{M_j}$ as the solution of the following GLM problem for a node j over its neighbors where $\Theta_{M_j} := \{\theta \in \mathbb{R}^p : [\theta]_k = 0 \text{ for } k \notin \mathcal{N}(j)\}$.

$$\theta^*_{M_j} := \arg\min_{\theta\in\Theta_{M_j}} \mathbb{E}\left(-X_j([\theta]_j + \langle [\theta]_{\mathcal{N}(j)}, X_{\mathcal{N}(j)}\rangle) + A_j([\theta]_j + \langle [\theta]_{\mathcal{N}(j)}, X_{\mathcal{N}(j)}\rangle)\right). \quad (7)$$

We impose the following identifiability assumptions on $\theta^*_{D_j}$ and $\theta^*_{M_j}$ for ensuring each parents and each neighbor has non-zero influence on a node j, respectively.

Assumption 6 *(a) For any node $j \in V$ and $k \in pa(j)$,*

$$Cov(X_j, X_k) \neq Cov(X_k, \triangledown A_j([\theta^*_{D_j}]_j + \langle [\theta^*_{D_j}]_{pa(j)\setminus k}, X_{pa(j)\setminus j}\rangle)).$$

(b) For any node $j \in V$ and $k \in \mathcal{N}(j)$,

$$Cov(X_j, X_k) \neq Cov(X_k, \triangledown A_j([\theta^*_{M_j}]_j + \langle [\theta^*_{M_j}]_{\mathcal{N}(j)\setminus k}, X_{\mathcal{N}(j)\setminus j}\rangle)).$$

Assumption 6 can be understood as a notion of restricted faithfulness only for neighbors and parents for each node. To provide intuition consider the special case of Gaussian DAG models. The log-partition function is $A_j(\eta) = \frac{\eta^2}{2}$, so that $\triangledown A_j(\eta) = \eta$. Then, the condition boils down to $\text{Cov}(X_j, X_k) \neq \sum_{m\in\text{pa}(j)\setminus k}[\theta^*_{D_j}]_m \text{Cov}(X_k, X_m)$, meaning the directed path from X_k to X_j does not exactly cancel the sum of paths from other parents of X_k. For general exponential families, the right-hand side involves non-linear functions of the variables of X corresponding to sets of measure 0. Under Assumption 6, the following result holds.

Lemma 7 *(a) Under Assumption 6(a), for all $1 \le j \le p$, $supp(\theta^*_{D_j}) = pa(j)$.*

(b) Under Assumption 6(b), for all $1 \le j \le p$, $supp(\theta^*_{M_j}) = \mathcal{N}(j)$.

Using the parameters $(\theta^*_{M_j})_{j=1}^p$ and $(\theta^*_{D_j})_{j=1}^p$ and their relationships to $\text{pa}(j)$ and $\mathcal{N}(j)$ respectively, we provide consistency guarantees for Steps 1) and 3) respectively.

3.2.1 Step 1): Recovery of the Moralized Graph via ℓ_1-penalized Likelihood Regression for GLMs

We first focus on the theoretical guarantee for recovering the moralized graph G^m. As we mentioned earlier, we approach this problem by solving an empirical version of the ℓ_1-penalized likelihood regression. Given n i.i.d. samples $X^{1:n} = (X^{(i)})_{i=1}^n$ where $X^{(i)} =$

$(X_1^{(i)}, X_2^{(i)} \cdots, X_p^{(i)})$ is a p-variate random vector drawn from the underlying DAG model, we define the conditional negative log-likelihood for a variable X_j:

$$\ell_j(\theta; X^{1:n}) := \frac{1}{n}\sum_{i=1}^{n}\left(-X_j^{(i)}([\theta]_j + \langle [\theta]_{V\setminus j}, X_{V\setminus j}^{(i)}\rangle) + A_j([\theta]_j + \langle [\theta]_{V\setminus j}, X_{V\setminus j}^{(i)}\rangle)\right) \quad (8)$$

where $\theta \in \mathbb{R}^p$ and $A_j(\cdot)$ is the log-partition function determined based on the chosen GLM family (3).

We analyze the ℓ_1-penalized log-likelihood for each node $j \in V$:

$$\hat{\theta}_{M_j} := \arg\min_{\theta\in\mathbb{R}^p} \ell_j(\theta; X^{1:n}) + \lambda_n \|[\theta]_{V\setminus j}\|_1 \quad (9)$$

where $\lambda_n > 0$ is the regularization parameter. Based on $\hat{\theta}_{M_j}$, the estimated neighborhood of node j is $\widehat{\mathcal{N}}(j) := \{k \in V \setminus j : [\hat{\theta}_M]_k \neq 0\}$. Based on Lemma 7, $\mathrm{supp}(\theta^*_{M_j}) = \mathcal{N}(j)$ where $\theta^*_{M_j}$ is defined by (7). Hence if for each j, $\hat{\theta}_{M_j}$ in (9) is sufficiently close to $\theta^*_{M_j}$, we conclude that $\widehat{\mathcal{N}}(j) = \mathcal{N}(j)$.

We begin by discussing the assumptions we impose on the DAG G. Since we apply the neighborhood selection strategy in Steps 1) and 3), we will present assumptions for both steps here. Most of the assumptions are similar to those imposed in Yang et al. (2012) where neighborhood selection is used for graphical model learning. Important quantities are the Hessian matrices of the negative conditional log-likelihood of a variable X_j given either the rest of the nodes $Q^{M_j} = \bigtriangledown^2 \ell_j(\theta^*_{M_j}; X^{1:n})$, and the nodes before j in the ordering $Q^{D_j} = \bigtriangledown^2 \ell_j^D(\theta^*_{D_j}; X^{1:n})$ which we discuss in Section 3.2.3. Let A_{SS} be the $|S| \times |S|$ submatrix of the matrix A_j corresponding to variables X_S.

Assumption 8 (Dependence assumption) *There exists a constant $\rho_{\min} > 0$ such that*

$$\min_{j\in V} \min\left(\lambda_{\min}(Q^{M_j}_{\mathcal{N}(j)\mathcal{N}(j)}), \lambda_{\min}(Q^{D_j}_{pa(j)pa(j)})\right) \geq \rho_{\min}.$$

Moreover, there exists a constant $\rho_{\max} < \infty$ such that

$$\max_{j\in V}\left(\lambda_{\max}\left(\frac{1}{n}\sum_{i=1}^{n} X^{(i)}_{\mathcal{N}(j)}(X^{(i)}_{\mathcal{N}(j)})^T\right)\right) \leq \rho_{\max}$$

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

Assumption 9 (Incoherence assumption) *There exists a constant $\alpha \in (0, 1]$ such that*

$$\max_{j\in V} \max\left(\max_{t\in\mathcal{N}(j)^c} \|Q^{M_j}_{t\mathcal{N}(j)}(Q^{M_j}_{\mathcal{N}(j)\mathcal{N}(j)})^{-1}\|_1, \max_{t'\in pa(j)^c} \|Q^{D_j}_{t'pa(j)}(Q^{D_j}_{pa(j)pa(j)})^{-1}\|_1\right) \leq 1-\alpha.$$

The dependence assumption 8 can be interpreted as ensuring that the variables in both $\mathcal{N}(j)$ and $\mathrm{pa}(j)$ are not too dependent. In addition, the incoherence assumption 9 ensures that variables that are not in the set of true variables are not highly correlated with

variables in the true variable set. These two assumptions are standard in all neighborhood regression approaches for variable selection involving ℓ_1-based methods and these conditions have imposed in proper work both for high-dimensional regression and graphical model learning (Yang et al., 2012; Meinshausen and Bühlmann, 2006; Wainwright et al., 2006; Ravikumar et al., 2011).

To ensure suitable concentration bounds hold, we impose two further technical assumptions. Firstly we require a boundedness assumption on the moment generating function to control the tail behavior.

Assumption 10 (Concentration bound assumption) *There exists a constant $M > 0$ such that*

$$\max_{j\in V} \mathbb{E}(exp(|X_j|)) < M.$$

We also require conditions on the first and third derivatives on the log-partition functions $A_j(.)$ for $1 \leq j \leq p$ in Equations (8) and (10). Let $A'_j(.)$ and $A'''_j(.)$ are the first and third derivatives of $A_j(.)$ respectively.

Assumption 11 (Log-partition assumption) *For the log-partition functions $A_j(\cdot)$ in Equation (8) or (10), there exist constants κ_1 and κ_2 such that $\max_{j\in V}\{|A'_j(a)|, |A'''_j(a)|\} \leq n^{\kappa_2}$ for $a \in [0, \kappa_1 \max\{\log(n), \log(p)\})$, $\kappa_1 \geq 6\max(\|\theta^*_{M_j}\|_1, \|\theta^*_{D_j}\|_1)$ and $\kappa_2 \in [0, 1/4]$.*

Prior work in Yang et al. (2012); Ravikumar et al. (2011); Jalali et al. (2011) impose similar technical conditions that control the tail behavior of $(X_j)_{j=1}^p$. It is important to note that there exist many distributions and associated parameters that satisfy these assumptions. For example the Binomial, Multinomial or Exponential distributions, the log-partition assumption 11 is satisfied with $\kappa_2 = 0$ because the log-partition function $A_j(\cdot)$ is bounded. For the Poisson distribution which has one of the steepest log-partition function, $A_j(\cdot) = \exp(\cdot)$. Hence, in order to satisfy Assumption 11, we require $\|\theta^*_{M_j}\|_1 \leq \frac{\log n}{48 \log p}$ with $\kappa_2 = \frac{1}{8}$.

Putting together Assumptions 8 9, 10, and 11, we have the following main result that the moralized graph can be recovered via ℓ_1-penalized likelihood regression for GLMs in high-dimensional settings.

Theorem 12 (Learning the moralized graph) *Consider the DAG model (1) satisfying the QVF property (2) and d is the maximum degree of the moralized graph. Suppose that Assumptions 6(b), 8, 9, 10 and 11 are satisfied. Assume $\hat{\theta}_{M_j}$ is any solution to the optimization problem (9) and $\frac{9\log^2(\max\{n,p\})}{n^a} \leq \lambda_n \leq \frac{\rho^2_{\min}}{30n^{\kappa_2}\log(\max\{n,p\})d\rho_{\max}}$ for some $a \in (2\kappa_2, 1/2)$, and $\min_{j\in V}\min_{t\in\mathcal{N}(j)}|[\theta^*_M]_t| \geq \frac{10}{\rho_{\min}}\sqrt{d}\lambda_n$. Then for any constant $\epsilon > 0$, there exists a positive constant C_ϵ such that if $n \geq C_\epsilon(d\log^3\max\{n,p\})^{\frac{1}{a-\kappa_2}}$,*

$$\mathbb{P}(supp(\hat{\theta}_{M_j}) = \mathcal{N}(j)) \geq 1-\epsilon,$$

for all $j \in V$.

We defer the proof to Appendix C. The key technique for the proof is that standard *primal-dual witness* method used in Wainwright et al. (2006); Ravikumar et al. (2011); Jalali et al. (2011); and Yang et al. (2012). Theorem 12 shows that the moralized graph G^m can be recovered via ℓ_1-penalized likelihood regression if sample size $n = \Omega((d\log^3(\max\{n,p\}))^{\frac{1}{a-\kappa_2}})$ with high probability.

3.2.2 Step 2): Recovering the Ordering using OverDispersion Scores

In this section, we provide theoretical guarantees for recovering the ordering for the DAG G via our generalized ODS algorithm. The first required condition is a stronger version of the identifiability assumption (Assumption 4) since we move from the population distribution to the finite sample setting.

Assumption 13 *For all $j \in V$ and any $pa_0(j) \subset pa(j)$ where $pa_0(j) \neq \emptyset$ and $S \subset nd(j) \setminus pa_0(j)$:*

(a) There exists an $M_{\min} > 0$ such that $Var(\mathbb{E}(X_j \mid X_{pa(j)}) \mid X_S) > M_{\min}$.

(b) There exists an $\omega_{\min} > 0$ such that $|\beta_{j0} + \beta_{j1}\mathbb{E}(X_j \mid X_S)| > \omega_{\min}$.

Assumption 10 is required since the overdispersion score is sensitive to the accuracy of the sample conditional mean and conditional variance. Since the true ordering π^* may not be unique, we use $\mathcal{E}(\pi^*)$ to denote the set of all the orderings that are consistent with the true DAG G.

Theorem 14 (Recovery of the ordering) *Consider the DAG model (1) satisfying the QVF property (2) with co-efficients $(\beta_{j0}, \beta_{j1})_{j=1}^p$ and d is the maximum degree of the moralized graph. Suppose that $\beta_{j1} > -1$ for all $j \in V$, and the structure of the moralized graph G^m is known. Suppose also that Assumptions 10 and 13 are satisfied. Then for any $\epsilon > 0$ and $c_0 \geq \log^d(\max\{n,p\})$, there exists a positive constant K_ϵ such that for $n \geq K_\epsilon \log^{5+d}(\max\{n,p\})$,*

$$P(\widehat{\pi} \in \mathcal{E}(\pi^*)) \geq 1 - \epsilon.$$

The detail of the proof is provided in Appendix D. The proof is novel and involves the combination of the transformation and overdispersion property exploited in Theorem 5. Intuitively, the estimated overdispersion scores $\widehat{\mathcal{S}}(m,j)$ converge to the true overdispersion scores $\mathcal{S}(m,j)$ as the sample size n increases which is where we exploit Assumption 10. This allows us to recover a true ordering for the DAG G. Assuming the moralized graph G^m is known is essential to exploiting the degree condition on the moralized graph and emphasizes the importance of Step 1) and Theorem 12.

Theorem 14 claims that if the triple (n,d,p) satisfies $n = \Omega(\log^{5+d} p)$, our generalized ODS algorithm correctly estimates the true ordering. Therefore if the moralized graph is sparse (i.e., $d = \Omega(\log p)$), our generalized ODS algorithm recovers the true casual ordering in the high-dimensional settings. Note that if the moralized graph is not sparse and $d = \Omega(p)$, the generalized ODS algorithm requires an extremely large sample size. Prior work on DAG learning algorithms in the high-dimensional setting has been based on learning the Markov equivalence class in settings with additive independent noise (see e.g., Loh and Bühlmann 2014; van de Geer and Bühlmann 2013).

3.2.3 Step 3): Recovery of the DAG via ℓ_1-penalized Likelihood Regression

Similar to Step 1), we provide a theoretical guarantee for Step 3) using ℓ_1-penalized likelihood regression where we estimate the parents of each node $\text{pa}(j)$. Importantly, we assume that Step 2) of the ODS algorithm has occurred and using Theorem 14, a true ordering has been learned. Recall that we impose the assumption that the true ordering is $\pi^* = (1, 2, \cdots, p)$. Then, we estimate the parents of a node j over the possible parents $\{1, 2, \cdots, j-1\}$.

For notational convenience, we use $X_{1:j} = (X_1, X_2, \cdots, X_j)$. Then for any variable X_j, the conditional negative log-likelihood for a given GLM is as follows:

$$\ell_j^D(\theta; X^{1:n}) := \frac{1}{n}\sum_{i=1}^n \left(-X_j^{(i)}([\theta]_j + \langle [\theta]_{1:j-1}, X_{1:j-1}^{(i)}\rangle) + A_j([\theta]_j + \langle [\theta]_{1:j-1}, X_{1:j-1}^{(i)}\rangle)\right) \quad (10)$$

where $\theta \in \mathbb{R}^j$, and $A_j(\cdot)$ is the log-partition function determined by a chosen GLM family.

We solve the negative conditional log-likelihood with ℓ_1 norm penalty for each variable X_j:

$$\hat{\theta}_{D_j} := \arg\min_{\theta \in \mathbb{R}^j} \ell_j^D(\theta; x) + \lambda_n^D \|[\theta]_{1:j-1}\|_1. \quad (11)$$

Recall that under Assumption 6(a), Lemma 7(a) shows that $\text{supp}(\theta^*_{D_j}) = \text{pa}(j)$. Hence if the solution of Equation (11) for each node $j \in V$ is close to $\theta^*_{D_j}$ in Equation (6), ℓ_1-penalized likelihood regression successfully recovers the parents of node j.

Theorem 15 (Learning DAG structure) *Consider the DAG model (1) satisfying the QVF property (2) and d is the maximum degree of the moralized graph. Suppose that Assumptions 6(a), 8, 9, 10 and 11 are satisfied. Assume $\hat{\theta}_{D_j}$ is any solution to the optimization problem (11) and $\frac{9\log^2(\max\{n,p\})}{n^a} \leq \lambda_n^D \leq \frac{\rho_{\min}^2}{30n^{\kappa_2}\log(\max\{n,p\})d\rho_{\max}}$ for some $a \in (2\kappa_2, 1/2)$, and $\min_{j\in V}\min_{t\in\mathcal{N}(j)}|[\theta^*_D]_t| \geq \frac{10}{\rho_{\min}}\sqrt{d}\lambda_n$. Then for any $\epsilon > 0$, there exists a positive constant C_ϵ such that if $n \geq C_\epsilon(d\log^3(\max\{n,p\}))^{\frac{1}{a-\kappa_2}}$,*

$$\mathbb{P}(supp(\hat{\theta}_{D_j}) = pa(j)) \geq 1 - \epsilon,$$

for all $j \in V$.

The details of the proof are provided in Appendix E. The proof technique is again based on the primal-dual technique as is used for the proof of Theorem 12. Theorem 15 shows that ℓ_1-penalized likelihood regression successfully recovers the structure of G if the sample size is $n = \Omega((d\log^3(\max\{n,p\}))^{\frac{1}{a-\kappa_2}})$ given the true ordering. Note once again that we exploit the sparsity d of the moralized graph.

So far, we have provided sample complexity guarantees for all three steps of the generalized ODS algorithm. Combining Theorems 12, 14, and 15, we reach our final main result that the generalized ODS algorithm successfully recovers the true structure of a QVF DAG with high probability. Furthermore if G is sparse (i.e., $d = \Omega(\log p)$), the generalized ODS algorithm recovers the structure of QVF DAG models in the high-dimensional setting.

Corollary 16 (Learning QVF DAG models) *Consider the DAG model* (1) *satisfying the QVF property* (2) *and d is the maximum degree of the moralized graph. Suppose that Assumptions 6, 8, 9, 10 and 11 are satisfied and all other conditions of Theorems 12, 14, and 15 are satisfied and $\widehat{G}$ is the output of the ODS algorithm. Then for any $\epsilon > 0$, there exists a positive constant C_ϵ such that if $n \geq C_\epsilon \max(d\log^3(\max\{n,p\}))^{\frac{1}{a-\kappa_2}}, \log^{5+d} p)$,*

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

Concretely, we apply Corollary 16 to popular examples for our class of QVF DAG models. As we discussed earlier, Poisson DAG models have $(\beta_{j0}, \beta_{j1}) = (1, 0)$, the steepest log-partition function $A_j(\cdot) = \exp(\cdot)$, and $\kappa_2 = \frac{1}{8}$ if $\|\theta^*_{M_j}\|_1 \leq \frac{\log n}{48\log(\max\{n,p\})}$. Then, our generalized ODS algorithm recovers Poisson DAG models with high probability if $n = \Omega(\max\{(d\log^3 p)^4, \log^{5+d} p\})$ and $a = \frac{3}{8}$. Binomial DAG models have $(\beta_{0j}, \beta_{1j}) = (0, -\frac{1}{N})$ where N is a binomial distribution parameter, the log-partition function $A_j(\cdot) = N\log(1 + \exp(\cdot))$, $\kappa_2 = 0$. Then, the generalized ODS algorithm recover Binomial DAG models with high probability if $n = \Omega(\max\{(d\log^3 p)^3, \log^{5+d} p\})$ and $a = \frac{1}{3}$.

4. Simulation Experiments

In this section, we support our theoretical guarantees with numerical experiments and show that our generalized ODS algorithm 1 performs favorably compared to state-of-the-art DAG learning algorithms when applied to QVF DAG models. In order to validate Theorems 12, 14, and 15, we conduct a simulation study using 50 realizations of p-node Poisson and Binomial DAG models (3). That is, the conditional distribution for each node given its parents is either Poisson and Binomial. For all our simulation results, we generate DAG models (see Figure 3) that ensure a unique ordering $\pi^* = (1, 2, \cdots, p)$ with edges randomly generated while respecting the desired maximum number of parents constraints for the DAG. In our experiments, we always set the number of parents to two (the number of neighbors of each node is at least three, and therefore $d \in [3, p-1]$).

The set of parameters (θ_{jk}) for our GLM DAG models (3) encodes the DAG structure as follows: if there is no directed edge from node k to j, $\theta_{jk} = 0$, otherwise $\theta_{jk} \neq 0$. Non-zero parameters $\theta_{jk} \in E$ were generated uniformly at random in the range $\theta_{jk} \in [-1, -0.5]$ for Poisson DAG models and $\theta_{jk} \in [0.5, 1]$ for Binomial DAG models. In addition, we fixed parameters $N_1, N_2, \cdots, N_p = 4$ for Binomial DAG models. These parameter values were chosen to ensure Assumptions 10 and 11 are satisfied and most importantly, the count values do not blow up. Lastly, we set the thresholding constant for computing the ODS score to $c_0 = 0.005$ although any value below 0.01 seems to work well in practice. We consider more general parameter choices but for brevity, focus on these parameter settings.

To validate Theorems 12 and 14, we plot the proportion (out of 50) of simulations in which our generalized ODS algorithm recovers the correct ordering to validate π^* in Figure 4. We plot the accuracy rates in recovering the true ordering $\mathbf{1}(\hat{\pi} = \pi^*)$ as a function of the sample size ($n \in \{100, 500, 1000, 2500, 5000, 10000\}$) for different node sizes ($p = 10$ for (a) and (c), and $p = 100$ for (b) and (d)) and different distributions (Poisson for (a) and (b) and Binomial for (c) and (d)). In each sub-figure, two different choices for off-the-shelf algorithms for Step 1) are used; (i) ℓ_1 penalized likelihood regression (Friedman et al., 2009) where we chose the regularization parameter $\lambda = \frac{0.75}{\log(\max\{n,p\})}$ for Poisson DAG models and

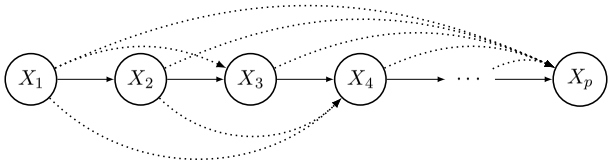
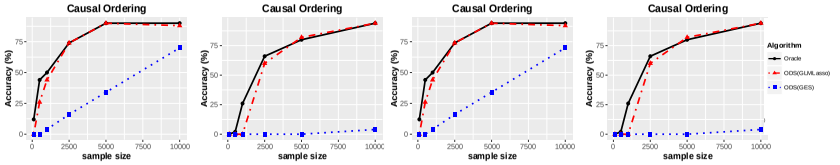


Figure 3: Structure of the DAG we used in numerical experiments. Solid directed edges are always present and dotted directed edges are randomly chosen based on the given number of parents of each node constraints



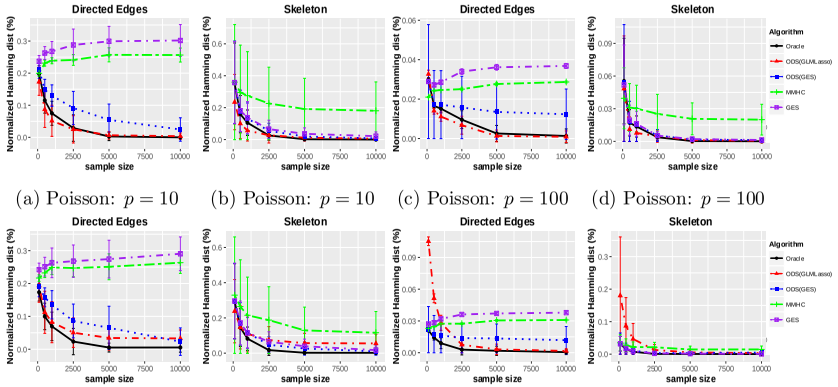
(a) Poisson: $p = 10$ (b) Poisson: $p = 100$ (c) Binomial: $p = 10$ (d) Binomial: $p = 100$

Figure 4: Probability of recovering the ordering of a DAG via our generalized ODS algorithm using two different algorithms (ℓ_1-penalized likelihood regression and GES algorithm) in Step 1)

$\lambda = \frac{.10}{\log(\max\{n,p\})}$ for Binomial DAG models; and (ii) the GES algorithm (Chickering, 2003) is applied for Step 1) where we used the mBDe (modified Bayesian Dirichlet equivalent, Heckerman et al. 1995) score and then the moralized graph is generated by moralizing the estimated DAG.

Figure 4 shows that our generalized ODS algorithm recovers the true ordering π^* well if the sample size is large, which supports our theoretical results. In addition, we can see that the ℓ_1-penalized based generalized ODS algorithm seems to perform substantially better than the GES-based ODS algorithm. Furthermore, since ℓ_1-penalized likelihood regression is the only algorithm that scales to the high-dimensional setting ($p \geq 1000$), we used ℓ_1-penalized likelihood regression in Steps 1) and 3) of the generalized ODS algorithm for large-scale DAG models.

Figure 5 provides a comparison of how accurately our generalized ODS algorithm performs in terms of recovering the full DAG model. We use two comparison metrics related to how many edges and directions are incorrect. First, we measured the Hamming distance between the skeleton (edges without directions) of the true DAG and the estimated DAG in (a), (c), (e) and (g). In addition, we measured the Hamming distance between the estimated and true DAG models (with directions) in (b), (d), (f), and (h). We normalized the Hamming distances by dividing by the maximum number of errors $\binom{p}{2}$ for the skeleton and



(a) Poisson: $p = 10$ (b) Poisson: $p = 10$ (c) Poisson: $p = 100$ (d) Poisson: $p = 100$

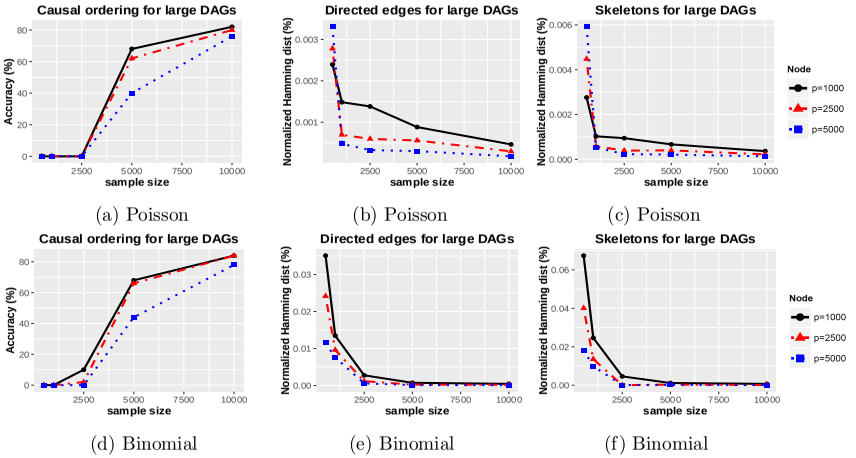
(e) Binomial: $p = 10$ (f) Binomial: $p = 10$ (g) Binomial: $p = 100$ (h) Binomial: $p = 100$

Figure 5: Comparison of the generalized ODS algorithms using ℓ_1-penalized likelihood regression (in Steps 1) and 3)) and the GES algorithm (in Steps 1) and 3)) to two state-of-the-art DAG learning algorithms (the MMHC and the GES algorithms) in terms of Hamming distance to skeletons and directed edges of Poisson and Binomial DAG models.

$p(p-1)$ for the full DAG respectively meaning the maximum normalized distance is 1. We compare to two state-of-the-art directed graphical model learning algorithms, the MMHC and GES algorithms for both Poisson and Binomial DAG models. Similar to learning the ordering, we used two generalized ODS algorithms exploiting ℓ_1-penalization in both Steps 1) and 3) and the GES algorithm in both Steps 1) and 3). We considered small-scale DAG models with $p = 10$ in (a), (b), (e) and (f), and $p = 100$ in (c), (d), (g) and (h).

As we see in Figure 5, the ODS algorithms significantly out-perform state-of-the-art MMHC and GES algorithms in terms of directed edges and skeleton. For small sample sizes, the generalized ODS algorithms have poor performance because they fail to recover the ordering, however we can see that the GES-based generalized ODS algorithm always performs better than the GES algorithm. This is because the generalized ODS algorithm adds directional information to the estimated skeleton via the GES algorithm, and hence the GES-based generalized ODS algorithm cannot be worse than the GES algorithm in terms of recovering both directed edges and skeleton. Furthermore Figure 5 shows that as sample size increases, our generalized ODS algorithms recovers the true directed edges and the skeleton for the DAG more accurately than state-of-the-art methods, which is consistent with our theoretical results.

Next we consider the performance for large-scale DAG models to show that the ODS algorithm works in the high-dimensional setting. In all experiments, we used the ℓ_1-penalized likelihood regression for GLMs in Steps 1) and 3) for the generalized ODS algorithm since it is the only graph-learning algorithm that scales. Figure 6 plots the statistical performance



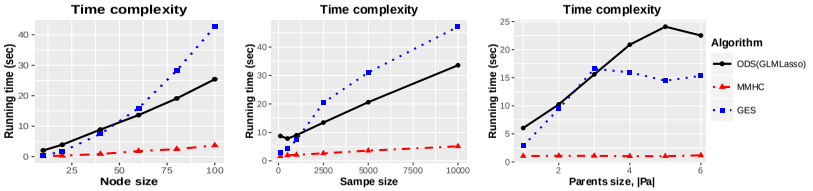
(a) Poisson (b) Poisson (c) Poisson

(d) Binomial (e) Binomial (f) Binomial

Figure 6: Performance of the generalized ODS algorithm using ℓ_1-penalized likelihood regression in both Steps 1) and 3) for large-scale DAG models with the node size $p = \{1000, 2500, 5000\}$

of the generalized ODS algorithm for large-scale Poisson DAGs in (a), (b), and (c) and Binomial DAGs in (d), (e), and (f). Furthermore, (a) and (d) represent the accuracy rates of the recovering the ordering, (b) and (e) show the normalized Hamming distance to the true skeleton, and (c) and (f) show the normalized Hamming distance for the true edge set of the DAG. Accuracies vary as a function of sample size ($n \in \{500, 1000, 2500, 5000, 10000\}$) for each node size ($p = \{1000, 2500, 5000\}$). Similar to small-scale DAG models, Figure 6 shows that the generalized ODS algorithm recovers the ordering and the skeleton of the DAG in the high-dimensional settings.

In Figure 7, we compared the run-time of the generalized ODS algorithms using ℓ_1-penalized likelihood regression for GLMs in Steps 1) and 3) to the run-time of the MMHC and the GES algorithms. We measured the run-time for Poisson DAG models by varying (a) node size $p \in \{10, 20, 40, 60, 80, 100\}$ with fixed sample size $n = 10000$ and exactly two parents of each node, (b) sample size $n \in \{100, 500, 1000, 2500, 5000, 10000\}$ with the fixed node size $p = 100$ and two parents of each node, and (c) the number of parents of each node $\in \{1, 2, 3, 4, 5, 6\}$ with the fixed sample size $n = 10000$ and node size $p = 20$. The results of (a) and (b) show that the generalized ODS algorithm is not always slower than the GES algorithm. In addition, (c) also shows that the run-time of the generalized ODS algorithm depends significantly on the number of parents for each node. Figure 7 shows that the generalized ODS algorithm is significantly slower than the MMHC algorithm, however this is because the MMHC algorithm often stops earlier before they reach the true DAG (see Figure 5).



(a) Poisson: $n = 10^4, d \geq 3$ (b) Poisson: $p = 100, d \geq 3$ (c) Poisson: $n = 10^4, p = 20$

Figure 7: Comparison of the generalized ODS algorithms using ℓ_1-penalized likelihood regression in Steps 1) and 3) to two standard DAG learning algorithms (the MMHC and the GES algorithms) in terms of running time with respect to (a) node size p, (b) sample size n, and (c) number of parents of each node

5. Real Multi-variate Count Data: 2009/2010 NBA Player Statistics

In terms of real data applications, one of the advantages of our ODS algorithm is that it provides a scalable approach for learning DAG models when variables are counts. In particular other approaches such as GES, MMHC and approaches based on conditional independence testing suffer severely from the fact that we are dealing with discrete variables where the number of discrete states is potentially large or infinite and represents counts. In this section, we demonstrate this advantage using a simple data set that involves multi-variate count data which models basketball statistics for NBA players during the 2009/10 season. To the best of our knowledge, our ODS algorithm is the only algorithm that provides a reliable and scalable approach for DAG learning with multi-variate count data, albeit under strong assumptions.

Our data set consists of 441 NBA player statistics from season 2009/2010 (see R package SportsAnalytics for detailed information). The original data set contains 24 covariates: player name, team name, players position (PG, SG, SF, PF or C), total minutes played, total number of field goals made, field goals attempted, threes made, threes attempted, free throws made, free throws attempted, offensive rebounds, rebounds, assists, steals, turnovers, blocks, personal fouls, disqualifications, technicals fouls, ejections, flagrant fouls, games started and total points. We eliminated player name, team name, number of games played, and players position, because our focus is to find the directional or causal relationships between statistics. We also eliminated ejections and flagrant fouls because both did not occur in our data set. Therefore the data set we consider contains 18 variables.

As we see in Figure 8 (left), all 18 variables are positively correlated. This makes sense because the total minutes played is likely to be positively correlated with other statistics, and some statistics have causal relationships (e.g., the more shooting attempt implies the more shooting made). The box plots in Figure 8 (right) show that the NBA statistics are significantly different depending on the player position. This is also makes sense because each position takes a different role. C and PF are expected to play near the baseline, hence they have more rebounds, blocks, and fouls. PG is expected to pass a ball and play far

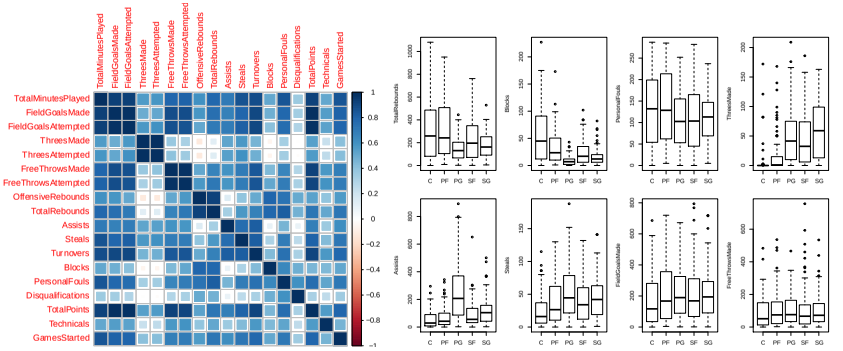


Figure 8: Correlation Plots for NBA statistics (left). Blue represents a high correlation and white represents a small correlation. Box plots for some NBA statistics depending on positions (right). Box plots consider the total number of rebounds, blocks, personal fouls, assists, steals, three points made, field goals made, and free throws made.

from the basket, hence PG has more steals, assists, turnover and the number of three points made. Hence the directed graph for each position may not be the same. For example, the directed graph for the position C and PF may not have an edge between total points and three points made because those positions players usually make a very small number of three points made while the directed graph for other positions may have an edge between total points and three points made. We also plotted DAG models for different positions, but for ease of presentation we combined all positions.

We assumed each node conditional distribution given its parents is Poisson because most of NBA statistics we consider are the number of successes or attempts counted in the season. Hence we applied the ODS algorithm 1 for Poisson DAG models where ℓ_1-penalized likelihood regression is used in Steps 1) and 3). We used leave-one-out cross validation to choose the tuning parameters, and chose the largest value where mean squared error is within 1 standard error of the minimum mean squared of error because we prefer a sparse graph containing only legitimate edges.

Figure 9 (left) shows the directed graph estimated by our method. The estimated graph reveals clear causal/directional relationships between statistics. A large number of shootings attempted implies a large number of shootings made that implies large total points. Moreover, a large number of rebounds implies a large number of offensive rebounds, and a large number of fouls implies more frequent disqualifications. Lastly, the more total minutes played, the more number of games started, total points and other statistics.

We also find the two clusters related to positions; (i) C and PF related nodes (blocks, offensive rebounds, rebounds, personal fouls, technical fouls, and disqualification) (ii) PG related nodes (steals, assists, turnover, and the number of three points attempts and made). Within the clusters, the nodes are highly connected although there may be no causal or

Eliminated Edges 1	Assist → ThreesMade, Turnovers → FreeThrowsMade, Disqualification → GamesStarted, Technicals → Blocks, Steal → ThreesMade, Steal → TotalPoints,
Eliminated Edge 2	TotalPoints → ThreesMade
Added Edge	Steals → TotalMinutesPlayed

Table 2: The differences between the estimated DAGs in Figure 9.

directional relationships. It can be understood that position variable is a latent variables, and if the position variable is considered in the graph, some false directed edges may be eliminated. However, we do not add the position variable in the graph because Multinomial distribution does not belong to the class of QVF distribution.

There are many unexplainable edges in Figure 9 (left) due to the assumptions made which are not completely satisfied by the real data. In order to obtain a sparser graph with legitimate edges, we applied the ODS algorithm with the same procedures except that we chose a larger tuning parameter where mean squared of error is within 2.5 standard error of the minimum mean squared of error.

Figure 9 (right) shows the estimated directed graph using large tuning parameters. Compared to Figure 9 (left), the estimated DAG has fewer edges as expected. Specifically, the estimated DAG in Figure 9 (right) excludes unrealistic edges (Eliminated Edges 1 in Table. 2). However the estimated DAG also loses a legitimate edge (Eliminated Edge 2 in Table. 2) because C and PF have fewer number of three points made. Lastly, the estimated DAG includes explainable additional edge (Added Edge in Table. 2) because Step 1) of the ODS algorithm reduces the search-space of DAGs well, and improves the accuracy of the graph structure learning.

We acknowledge that our estimated DAG model makes many errors due to the restrictive assumption. However the benefit is best seen by comparing to other DAG learning approaches and an undirected graphical model. In particular, we applied Poisson undirected graphical models (Yang et al., 2013) which is the same procedure of Step 1) of our algorithm. The estimated undirected graph in Figure 10 (left) shows that a lot of nodes are connected by edges, and many edges are not explainable because the Poisson undirected graphical model only permits negative conditional relationships while all 18 variables are positively correlated. Hence it is not useful to understand the relationships between NBA statistics. We provide the estimated undirected graph with larger tuning parameter where mean squared of error is within 2.5 standard error of the minimum mean squared of error.

We also compare to the GES and MMHC algorithms. In particular, the estimated graphs in Figure 10 (right) are the same and both algorithms use the Bayesian Dirichlet score for count data which prefers a sparse graph when the positivity assumption is violated (i.e., $\widehat{P}(X_j = x_j \mid \mathrm{pa}(X_j)) \approx 0$). Since all statistics have high cardinality, which means each variable has almost no repeats in its data range, the positivity assumption is not satisfied. Hence the estimated directed graphs are extremely sparse which have a single directed edge between technical fouls and disqualification.

Since our method is the first identifiability result for the count data to the best of our knowledge, our method more reliably recovers the directional/causal relationships between

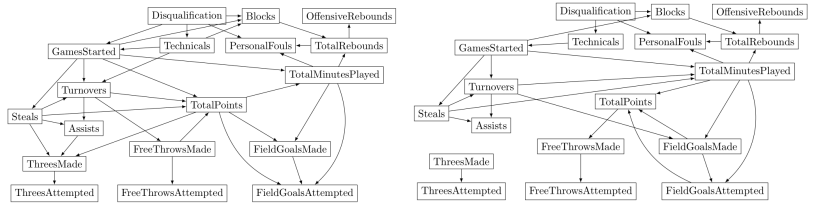


Figure 9: NBA players statistics directed graph estimated by the ODS algorithm for Poisson DAG models using ℓ_1-penalized likelihood regression in Steps 1) and 3) with small tuning parameters (left) and large tuning parameters (right).

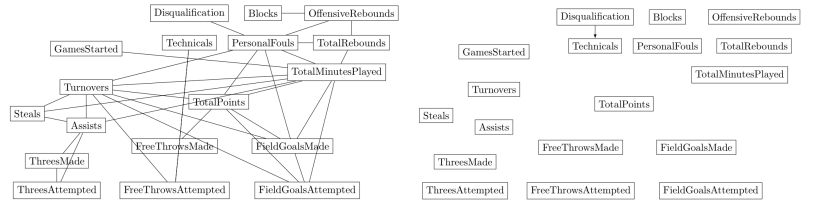


Figure 10: NBA players statistics undirected graph estimated by ℓ_1-penalized likelihood regression (left) and directed acyclic graph estimated by GES and MMHC algorithms (right).

NBA statistics. However we acknowledge that like most other DAG-learning approaches, very strong assumptions are required fore reliable recovery.

Acknowledgments

We would like to acknowledge support for this project from the National Science Foundation (NSF DMS-1407028).

Appendix A. Proof for Theorem 5

Proof Without loss of generality, we assume the ordering is $\pi^* = (1, 2, \cdots, p)$. For notational convenience, we define $X_{1:j} = \{X_1, X_2, \cdots, X_j\}$ and $X_{1:0} = \emptyset$. For $m \in \{1, 2, ..., p-1\}$ and $j \in \{m, m+1, \cdots, p\}$, let $\omega_{jm} = (\beta_0 + \beta_1 \mathbb{E}(X_j \mid X_{1:m-1}))^{-1}$ and $\omega_{j1} = (\beta_0 + \beta_1 \mathbb{E}(X_j))^{-1}$. Recall that the overdispersion score of node j for m^{th} element of the ordering is Equation (5):

$$\mathcal{S}(m, j) = \omega_{jm}^2 \mathrm{Var}(X_j \mid X_{1:m-1}) - \omega_{jm} \mathbb{E}(X_j \mid X_{1:m-1}).$$

We now prove identifiability of our class of DAG models by induction. For the first element of the ordering ($m = 1$),

$$\begin{aligned}
\mathcal{S}(1, j) &= \omega_{j1}^2 \mathrm{Var}(X_j) - \omega_{j1} \mathbb{E}(X_j) \\
&\overset{(a)}{=} \omega_{j1}^2 \big\{ \mathrm{Var}(\mathbb{E}(X_j \mid X_{\mathrm{pa}(j)})) + \mathbb{E}(\mathrm{Var}(X_j \mid X_{\mathrm{pa}(j)})) - \omega_{j1}^{-1} \mathbb{E}(X_j) \big\} \\
&\overset{(b)}{=} \omega_{j1}^2 \big\{ \mathrm{Var}(\mathbb{E}(X_j \mid X_{pa(j)})) + \mathbb{E}(\beta_0 \mathbb{E}(X_j \mid X_{\mathrm{pa}(j)}) + \beta_1 \mathbb{E}(X_j \mid X_{\mathrm{pa}(j)})^2) \\
&\qquad - (\beta_0 + \beta_1 \mathbb{E}(X_j)) \mathbb{E}(X_j) \big\} \\
&= \omega_{j1}^2 \big\{ \mathrm{Var}(\mathbb{E}(X_j \mid X_{pa(j)})) + \beta_1 \mathbb{E}(\mathbb{E}(X_j \mid X_{\mathrm{pa}(j)})^2) - \beta_1 \mathbb{E}(X_j)^2 \big\} \\
&= \omega_{j1}^2 (1 + \beta_1) \mathrm{Var}(\mathbb{E}(X_j \mid X_{pa(j)})).
\end{aligned}$$

(a) follows from the variance decomposition formula $\mathrm{Var}(Y) = \mathbb{E}(\mathrm{Var}(Y \mid X)) + \mathrm{Var}(\mathbb{E}(Y \mid X))$ for some random variables X and Y. In addition (b) follows from the quadratic variance property (2) of our class of distributions and the definition of ω_{j1}. Note that the score of the first element of the ordering is $\mathcal{S}(1, 1) = 0$ because $\mathrm{Var}(E(X_1)) = 0$, and other scores are strictly positive $\mathcal{S}(j, 1) > 0$ by the assumption $\beta_1 > -1$. Therefore 1 is the first element of the ordering.

For the $(m-1)^{th}$ element of the ordering, assume that the first $m-1$ elements of the ordering are correctly estimated. Now, we consider the m^{th} element of the ordering. Then, for $j \in \{m, m+1, \cdots, p\}$,

$$\begin{aligned}
\mathcal{S}(m, j) &= \omega_{jm}^2 \mathrm{Var}(X_j \mid X_{1:m-1}) - \omega_{jm} \mathbb{E}(X_j \mid X_{1:m-1}) \\
&\overset{(a)}{=} \omega_{jm}^2 \big\{ \mathrm{Var}(\mathbb{E}(X_j \mid X_{\mathrm{pa}(j)}) \mid X_{1:m-1}) + \mathbb{E}(\mathrm{Var}(X_j \mid X_{\mathrm{pa}(j)}) \mid X_{1:m-1}) - \omega_{jm}^{-1} \mathbb{E}(X_j \mid X_{1:m-1}) \big\} \\
&\overset{(b)}{=} \omega_{jm}^2 \big\{ \mathrm{Var}(\mathbb{E}(X_j \mid X_{pa(j)}) \mid X_{1:m-1}) + \mathbb{E}(\beta_0 \mathbb{E}(X_j \mid X_{\mathrm{pa}(j)} \mid X_{1:m-1}) \\
&\qquad + \beta_1 \mathbb{E}(X_j \mid X_{\mathrm{pa}(j)} \mid X_{1:m-1})^2) - (\beta_0 + \beta_1 \mathbb{E}(X_j \mid X_{1:m-1})) \mathbb{E}(X_j \mid X_{1:m-1}) \big\} \\
&= \omega_{jm}^2 \big\{ \mathrm{Var}(\mathbb{E}(X_j \mid X_{pa(j)}) \mid X_{1:m-1}) + \beta_1 \mathbb{E}(\mathbb{E}(X_j \mid X_{\mathrm{pa}(j)})^2 \mid X_{1:m-1}) - \beta_1 \mathbb{E}(X_j \mid X_{1:m-1})^2 \big\} \\
&= \omega_{jm}^2 (1 + \beta_1) \mathrm{Var}(\mathbb{E}(X_j \mid X_{pa(j)}) \mid X_{1:m-1}).
\end{aligned}$$

Again (a) follows from the variance decomposition formula, and (b) follows from the quadratic variance property (2) of our class of distributions and the definition of ω_{jm}. If $\mathrm{pa}(j) \backslash \{1, 2, \cdots, m-1\}$ is empty, $\mathrm{Var}(\mathbb{E}(X_j \mid X_{pa(j)}) \mid X_{1:m-1}) = 0$, and hence $\mathcal{S}(m, m) = 0$. On the other hand, for any node j in which $\mathrm{pa}(j) \backslash \{1, 2, \cdots, m-1\}$ is non-empty, $\mathcal{S}(m, j) > 0$ by the assumption $\beta_{j1} > -1$, which excludes it from being next in the ordering. Therefore, we can estimate a valid m^{th} component of the ordering, $\widehat{\pi}_m = m$. By induction this completes the proof. ■

Appendix B. Proof for Lemma 7

Proof We begin with part (a). By the construction $\theta^*_{D_j}$ in Equation (6), $[\theta^*_{D_j}]_k = 0$ for any node $k \notin \mathrm{pa}(j)$. Hence, it is sufficient to show that for any $k \in \mathrm{pa}(j)$, $[\theta^*_{D_j}]_k \neq 0$. Assume for the sake of contradiction that $[\theta^*_{D_j}]_k = 0$. Applying the first order optimality condition to Equation (6), we have

$$\begin{aligned}
\mathbb{E}(X_j) &= \mathbb{E}(A_j'([\theta^*_{D_j}]_j + \langle [\theta^*_{D_j}]_{\mathrm{pa}(j)}, X_{\mathrm{pa}(j)} \rangle)) \\
\mathbb{E}(X_j X_k) &= \mathbb{E}(A_j'([\theta^*_{D_j}]_j + \langle [\theta^*_{D_j}]_{\mathrm{pa}(j)}, X_{\mathrm{pa}(j)} \rangle) X_k).
\end{aligned} \tag{12}$$

By the definition of the covariance, we obtain

$$\begin{aligned}
\mathbb{E}(X_j X_k) &= \mathrm{Cov}(A'([\theta^*_{D_j}]_j + \langle [\theta^*_{D_j}]_{\mathrm{pa}(j)}, X_{\mathrm{pa}(j)} \rangle), X_k), \\
&\quad + \mathbb{E}(A'([\theta^*_{D_j}]_j + \langle [\theta^*_{D_j}]_{\mathrm{pa}(j)}, X_{\mathrm{pa}(j)} \rangle)) \mathbb{E}(X_k).
\end{aligned}$$

By Equation (12),

$$\mathbb{E}(X_j X_k) = \mathrm{Cov}(A'([\theta^*_{D_j}]_j + \langle [\theta^*_{D_j}]_{\mathrm{pa}(j)}, X_{\mathrm{pa}(j)} \rangle), X_k) + \mathbb{E}(X_j) \mathbb{E}(X_k).$$

Therefore,

$$\mathrm{Cov}(X_j, X_k) = \mathrm{Cov}(A'([\theta^*_{D_j}]_j + \langle [\theta^*_{D_j}]_{\mathrm{pa}(j)}, X_{\mathrm{pa}(j)} \rangle), X_k).$$

By Assumption 6 (a), we have $[\theta^*_{D_j}]_k = 0$, and

$$\mathrm{Cov}(X_j, X_k) = \mathrm{Cov}(D'([\theta^*_{D_j}]_j + \langle [\theta^*_{D_j}]_{\mathrm{pa}(j) \backslash k}, X_{\mathrm{pa}(j) \backslash j} \rangle), X_k),$$

which is a contradiction by our earlier assumption. Therefore $[\theta^*_{D_j}]_k \neq 0$. Furthermore since $k \in \mathrm{pa}(j)$ is arbitrary, the proof is complete. The proof for part (b) follows exactly the same line of reasoning. ■

Appendix C. Proof for Theorem 12

In this section, we provide the proof for Theorem 12 using the *primal-dual witness method* that also used many works (see e.g., Yang et al. 2012; Meinshausen and Bühlmann 2006; Wainwright et al. 2006; Ravikumar et al. 2011). We begin by introducing propositions to control the tail behavior for the distribution of each node:

Proposition 17 *Define*

$$\xi_1 := \{ \max_{j \in V} \max_{i \in \{1, \cdots, n\}} |X_j^{(i)}| < 4 \log(\eta) \}.$$

Under Assumption 10, $P(\xi_1^c) \leq M \cdot \eta^{-2}$.

Proposition 18 *Suppose that X is a random vector according to the DAG model* (1)*, and Assumption 10 is satisfied. Then, for any vector $u \in \mathbb{R}^p$ such that $\|u\|_1 \leq c'$, for any positive constant δ,*

$$P(|\langle u, X \rangle| \geq \delta \log \eta) \leq M \cdot p \cdot \eta^{-\delta / c'}. \tag{13}$$

Using these concentration results, we show that ℓ_1-penalized regression recovers the neighborhood for a fixed node $j \in V$ with high probability. For ease of notation, we define a new parameter $\theta \in \mathbb{R}^{p-1}$ without the node j since the node j is not penalized in regression problem (9). Then, the conditional negative log-likelihood of the GLM (8) is:

$$\ell_j(\theta; X^{1:n}) := \frac{1}{n}\sum_{i=1}^{n}\left(-X_j^{(i)}\langle\theta, X_{V\setminus j}^{(i)}\rangle + A_j(\langle\theta, X_{V\setminus j}^{(i)}\rangle)\right).$$

The main goal of the proof is to find the unique minimizer of the following convex problem:

$$\widehat{\theta}_{M_j} := \arg\min_{\theta\in\mathbb{R}^{p-1}} \mathcal{L}_j(\theta, \lambda_n) = \arg\min_{\theta\in\mathbb{R}^{p-1}} \{\ell_j(\theta; X^{1:n}) + \lambda_n\|\theta\|_1\}. \tag{14}$$

By setting the *sub-differential* to 0, $\widehat{\theta}_{M_j}$ must satisfy the following condition:

$$\bigtriangledown_\theta \mathcal{L}_j(\widehat{\theta}_{M_j}, \lambda_n) = \bigtriangledown_\theta \ell_j(\widehat{\theta}_{M_j}; X^{1:n}) + \lambda_n \widehat{Z} = 0 \tag{15}$$

where $\widehat{Z} \in \mathbb{R}^{p-1}$ and $\widehat{Z}_t = \text{sign}([\widehat{\theta}_{M_j}]_t)$ if $t \in \mathcal{N}(j)$, otherwise $|\widehat{Z}_t| < 1$.

The following Lemma 19 directly follows from prior works in Ravikumar et al. (2010) and Yang et al. (2012) where each node conditional distribution is in the form of a generalized linear model. For notational convenience, let $S = \mathcal{N}(j)$.

Lemma 19 *Suppose that $|\widehat{Z}_t| < 1$ for $t \notin S$. Then, the solution $\widehat{\theta}_{M_j}$ of (14) satisfies $[\widehat{\theta}_{M_j}]_t = 0$ for $t \notin S$. Furthermore, if the sub-matrix of the Hessian matrix $Q_{SS}^{M_j}$ is invertible, then $\widehat{\theta}_{M_j}$ is unique.*

The remainder of the proof is to show $|\widetilde{Z}_t| < 1$ for all $t \notin S$. Note that the restricted solution in Equation (19) is $(\widetilde{\theta}_{M_j}, \widetilde{Z})$. Equation (15) with the dual solution can be represented by

$$\bigtriangledown^2 \ell_j(\theta^*_{M_j}; X^{1:n})(\widetilde{\theta}_{M_j} - \theta^*_{M_j}) = -\lambda_n \widetilde{Z} - W_j^n + R_j^n$$

where:

(a) W_j^n is the sample score function.

$$W_j^n := -\bigtriangledown \ell_j(\theta^*_{M_j}; X^{1:n}). \tag{16}$$

(b) $R_j^n = (R_{j1}^n, R_{j2}^n, \cdots, R_{jp-1}^n)$ and R_{jk}^n is the remainder term by applying the coordinate-wise mean value theorem.

$$R_{jk}^n := [\bigtriangledown^2 \ell_j(\theta^*_{M_j}; X^{1:n}) - \bigtriangledown^2 \ell_j(\bar{\theta}^{(k)}_{M_j}; X^{1:n})]_k^T(\widetilde{\theta}^{(k)}_{M_j} - \theta^*_{M_j}). \tag{17}$$

Here $\bar{\theta}^{(k)}_{M_j}$ is a vector on the line between $\widetilde{\theta}$ and $\theta^*_{M_j}$ and $[\cdot]_k^T$ is the k^{th} row of a matrix.

Then, the following proposition provides a sufficient condition to control $\widetilde{Z}$.

Proposition 20 *Suppose that* $\max(\|W_j^n\|_\infty, \|R_j^n\|_\infty) \le \frac{\lambda_n\alpha}{4(2-\alpha)}$. *Then $|\widetilde{Z}_t| < 1$ for all $t \notin S$.*

Next we introduce the following three lemmas to show that conditions in Proposition 20 hold. For ease of notation, let $\eta = \max\{n, p\}$ and $\widetilde{\theta}_S = [\widetilde{\theta}_{M_j}]_S$ and $\widetilde{\theta}_{S^c} = [\widetilde{\theta}_{M_j}]_{S^c}$. Suppose that Assumptions 8, 9, 10, and 11 are satisfied.

Lemma 21 *Suppose that $\lambda_n \ge \frac{16\max\{n^{\kappa_2}\log\eta, \log^2\eta)\}}{n^a}$ for some $a \in \mathbb{R}$. Then,*

$$P\left(\frac{\|W_j^n\|_\infty}{\lambda_n} \le \frac{\alpha}{4(2-\alpha)}\right) \ge 1 - 2d\cdot exp(-\frac{\alpha^2}{8(2-\alpha)^2}\cdot n^{1-2a}) - M\cdot\eta^{-2}.$$

Lemma 22 *Suppose that $\|W_j^n\|_\infty \le \frac{\lambda_n}{4}$. For $\lambda_n \le \frac{1}{40}\frac{\rho^2_{\min}}{\rho_{\max}}\frac{1}{n^{\kappa_2} d\log\eta}$,*

$$P\left(\|\widetilde{\theta}_S - \theta^*_S\|_2 \le \frac{5}{\lambda_{\min}}\sqrt{d}\lambda_n\right) \ge 1 - 2M\cdot\eta^{-2}.$$

Lemma 23 *Suppose that $\|W_j^n\|_\infty \le \frac{\lambda_n}{4}$. For $\lambda_n \le \frac{\alpha}{400(2-\alpha)}\frac{\rho^2_{\min}}{\rho_{\max}}\frac{1}{n^{\kappa_2} d\log\eta}$,*

$$P\left(\frac{\|R_j^n\|_\infty}{\lambda_n} \le \frac{\alpha}{4(2-\alpha)}\right) \ge 1 - 2M\cdot\eta^{-2}.$$

The rest of the proof is straightforward using Lemmas 21, 22, and 23. Consider the choice of regularization parameter $\lambda_n = \frac{16\max\{n^{\kappa_2}\log\eta, \log^2\eta\}}{n^a}$ for a constant $a \in (2\kappa_2, 1/2)$ where κ_2 is determined by Assumption 11. Then, the condition for Lemma 21 is satisfied, and therefore $\|W_n\|_\infty \le \frac{\lambda_n}{4}$. Moreover, the conditions for Lemmas 22 and 23 are satisfied for $n \ge C'\max\{(d\log^2\eta)^{\frac{1}{a-2\kappa_2}}, (d\log^3\eta)^{\frac{1}{a-\kappa_2}}\}$ for some positive constant C'. Then,

$$\|\widetilde{Z}_{S^c}\|_\infty \le (1-\alpha) + (2-\alpha)\left[\frac{\|W_j^n\|_\infty}{\lambda_n} + \frac{\|R_j^n\|_\infty}{\lambda_n}\right] \le (1-\alpha) + \frac{\alpha}{4} + \frac{\alpha}{4} < 1, \tag{18}$$

with probability of at least $1 - C_1 d\exp(-C_2 n^{1-2a}) - C_3\eta^{-2}$ for positive constants C_1, C_2 and C_3.

To prove sign consistency, it is sufficient to show that $\|\widehat{\theta}_{M_j} - \theta^*_{M_j}\|_\infty \le \frac{\|\theta^*_{M_j}\|_{\min}}{2}$. By Lemma 22, we have $\|\widehat{\theta}_{M_j} - \theta^*_{M_j}\|_\infty \le \|\widehat{\theta}_{M_j} - \theta^*_{M_j}\|_2 \le \frac{5}{\lambda_{\min}}\sqrt{d}\ \lambda_n \le \frac{\|\theta^*_{M_j}\|_{\min}}{2}$ as long as $\|\theta^*_{M_j}\|_{\min} \ge \frac{10}{\lambda_{\min}}\sqrt{d}\ \lambda_n$.

Lemma 7(b) guarantees that ℓ_1-penalized likelihood regression recovers the true neighborhood for each node with high probability. Because we have p likelihood regression problems, if $n \ge C'(d\log^2\eta)^{\frac{1}{a-2\kappa_2}})$, it follows that:

$$P(\widehat{G^m} = G^m) \ge 1 - C_1 d\cdot p\cdot\exp(-C_2 n^{1-2a}) - C_3\eta^{-1}.$$

C.1 Proof for Proposition 17

Proof Applying the union bound and the Chernoff bound,

$$P(\xi_1^c) \le n.p.\max_{j\in V}\max_{i\in\{1,\cdots,n\}} P\left(|X_j^{(i)}| > 4\log\eta\right) \le \eta^{-2}\max_{i,j}\mathbb{E}[\exp(|X_j^{(i)}|)].$$

By Assumption 10, we obtain $\max_{i,j}\mathbb{E}(\exp(|X_j|^{(i)})) < M$, which completes the proof. ■

C.2 Proof for Proposition 18

Proof We exploit Hölder's inequality $\langle u, X\rangle \leq \|u\|_1 \max_{j\in V}|X_j|$. Therefore, we have

$$P(|\langle u, X\rangle)| \geq \delta \log \eta) \leq P(\max_{j\in V}|X_j| \geq \frac{\delta}{\|u\|_1}\log\eta).$$

Using the union bound, we have

$$P(\max_{j\in V}|X_j| \geq \frac{\delta}{\|u\|_1}\log\eta) \leq p\cdot\max_{j\in V}P(|X_j| \geq \frac{\delta}{\|u\|_1}\log\eta).$$

Applying the Chernoff bounding technique and Assumption 10 $\max_j \mathbb{E}(\exp(|X_j|) < M$, we obtain

$$p\cdot\max_{j\in V}P(|X_j| \geq \frac{\delta}{\|u\|_1}\log\eta) \leq M\cdot p\cdot\eta^{-\frac{\delta}{\|u\|_1}}.$$

By the assumption $\|u\|_1 \leq c'$, we compete the proof. ■

C.3 Proof for Proposition 20

Proof Since $\widetilde{\theta}_{S^c} = (0, 0, ..., 0) \in \mathbb{R}^{|S^c|}$ in our primal-dual construction, we can re-state condition (15) in block form as follows. For notational simplicity, $Q := Q^{M_j}$.

$$\begin{aligned} Q_{S^cS}[\widetilde{\theta}_S - \theta_S] &= W^n_{S^c} - \lambda_n\widetilde{Z}_{S^c} + R^n_{S^c}, . \\ Q_{SS}[\widetilde{\theta}_S - \theta^*_S] &= W^n_S - \lambda_n\widetilde{Z}_S + R^n_S, \end{aligned}$$

where W^n_S and R^n_S are sub-vectors of W^n_j and R^n_j indexed by S, respectively.

Since the matrix Q_{SS} is invertible, the above equations can be rewritten as

$$Q_{S^cS}Q^{-1}_{SS}[W^n_S - \lambda_n\widetilde{Z}_S - R^n_S] = W^n_{S^c} - \lambda_n\widetilde{Z}_{S^c} - R^n_{S^c}.$$

Therefore

$$[W^n_{S^c} - R^n_{S^c}] - Q_{S^cS}Q^{-1}_{SS}[W^n_S - R^n_S] + \lambda_n Q_{S^cS}Q^{-1}_{SS}\widetilde{Z}_S = \lambda_n\widetilde{Z}_{S^c}.$$

Taking the ℓ_∞ norm of both sides yields

$$\|\widetilde{Z}_{S^c}\|_\infty \leq |||Q_{S^cS}Q^{-1}_{SS}|||_\infty\left[\frac{\|W^n_S\|_\infty}{\lambda_n} + \frac{\|R^n_S\|_\infty}{\lambda_n} + 1\right] + \frac{\|W^n_{S^c}\|_\infty}{\lambda_n} + \frac{\|R^n_{S^c}\|_\infty}{\lambda_n}.$$

Recalling Assumption (9), we obtain $|||Q_{S^cS}Q^{-1}_{SS}|||_\infty \leq (1-\alpha)$, hence we have

$$\begin{aligned} \|\widetilde{Z}_{S^c}\|_\infty &\leq (1-\alpha)\left[\frac{\|W^n_S\|_\infty}{\lambda_n} + \frac{\|R^n_S\|_\infty}{\lambda_n} + 1\right] + \frac{\|W^n_{S^c}\|_\infty}{\lambda_n} + \frac{\|R^n_{S^c}\|_\infty}{\lambda_n} \\ &\leq (1-\alpha) + (2-\alpha)\left[\frac{\|W^n_j\|_\infty}{\lambda_n} + \frac{\|R^n\|_\infty}{\lambda_n}\right]. \end{aligned}$$

If $\|W^n_j\|_\infty$ and $\|R^n_j\|_\infty \leq \frac{\lambda_n\alpha}{4(2-\alpha)}$ as assumed,

$$\|\widetilde{Z}_{S^c}\|_\infty \leq (1-\alpha) + \frac{\alpha}{2} \leq 1.$$

■

C.4 Proof for Lemma 19

Proof The main idea of the proof is the *primal-dual-witness* method which asserts that there is a solution to the dual problem $\widetilde{\theta}_{M_j} = \widehat{\theta}_{M_j}$ if the following KKT conditions are satisfied:

(a) We define $\widetilde{\theta}_{M_j} \in \Theta_{M_j}$ where $\Theta_{M_j} = \{\theta \in \mathbb{R}^{p-1} : \theta_{S^c} = 0\}$ as the solution to the following optimization problem.

$$\widetilde{\theta}_{M_j} := \arg\min_{\theta\in\Theta_{M_j}} \mathcal{L}(\theta, \lambda_n) = \arg\min_{\theta\in\Theta_{M_j}} \{\ell_j(\theta; X^{1:n}) + \lambda_n\|\theta\|_1\}. \tag{19}$$

(b) Define $\widetilde{Z}$ to be a sub-differential for the regularizer $\|\cdot\|_1$ evaluated at $\widetilde{\theta}_{M_j}$. For any $t \in S$, $\widetilde{Z}_t = \text{sign}([\widetilde{\theta}_{M_j}]_t)$.

(c) For any $t \notin S$, $|\widetilde{Z}_t| < 1$.

If conditions (a), (b), and (c) are satisfied, $\widehat{\theta}_{M_j} = \widetilde{\theta}_{M_j}$, meaning that the solution of the unrestricted problem (14) is the same as the solution of the restricted problem (19). Conditions (a), (b) and (c) suffice to obtain a pair $(\widetilde{\theta}_{M_j}, \widetilde{Z})$ that satisfies the optimality condition (15), but do not guarantee that $\widetilde{Z}$ is an element of the sub-differential $\|\widetilde{\theta}_{M_j}\|_1$ (see details in Ravikumar et al. 2010, 2011). Since the sub-matrix of the Hessian $Q^{M_j}_{SS}$ is invertible, the restricted problem (19) is strictly convex, $\widetilde{\theta}_{M_j}$ is unique. ■

C.5 Proof for Lemma 21

Proof Each entry of the sample score function W^n_j in Equation (16) has the form $W^n_{jt} = \frac{1}{n}\sum_{i=1}^n W^{(i)}_{jt}$ for any $t \in S$. In addition, $W^n_{jt} = 0$ for all $t \notin S$ since $[\theta^*_{M_j}]_t = 0$ by the construction of $\theta^*_{M_j}$ in Equation (7). For any $t \in S$ and $i \in \{1, 2, \cdots, n\}$, $W^{(i)}_{jt} = X^{(i)}_t X^{(i)}_j - A'_j(\langle\theta^*_S, X^{(i)}_S\rangle)X^{(i)}_t$ are independent and have mean 0.

Now, we show that $(|W^{(i)}_{jt}|)^n_{i=1}$ are bounded with high probability given the following event ξ_1 using Hoeffding's inequality. Event ξ_1 is defined as follows:

$$\xi_1 := \left\{\max_{j\in V}\max_{i\in\{1,\cdots,n\}}|X^{(i)}_j| < 4\log\eta\right\}.$$

Conditioning on ξ_1, it follows that $\langle\theta^*_S, X^{(i)}_S\rangle < 4\log(\eta)\cdot\|\theta^*_S\|_1$, Assumption 11 is satisfied. Hence $\max_i|A'_j(\langle\theta^*_S, X^{(i)}_S\rangle)| \leq n^{\kappa_2}$. Furthermore given ξ_1, $\max_i X^{(i)}_t X^{(i)}_j < 16\log^2\eta$. Therefore there exists a constant $C_{\max}(\eta, \kappa_2) := 16\max\{n^{\kappa_2}\log\eta, \log^2\eta\}$ such that $\max_{i,j,t}|W^{(i)}_{jt}| \leq C_{\max}(\eta, \kappa_2)$.

Recall that d is the maximum degree of the moralized graph, therefore $|S| \leq d$. Applying the union bound,

$$P(\|W^n_j\|_\infty > \delta, \xi_1) \leq d\cdot\max_{t\in S}P(|W^n_{jt}| > \delta, \xi_1).$$

Using Hoeffding's inequality,

$$d \cdot \max_{t \in S} P(|W_{jt}^n| > \delta, \xi_1) \leq 2d \cdot \exp(-\frac{2n\delta^2}{C_{\max}(\eta,\kappa_2)^2}).$$

Suppose that $\delta = \frac{\lambda_n \alpha}{4(2-\alpha)}$ and $\lambda_n \geq \frac{C_{\max}(\eta,\kappa_2)}{n^a}$ for some $a \in [0, 1/2)$. Then

$$\begin{aligned} P(\frac{\|W_j^n\|_\infty}{\lambda_n} > \frac{\alpha}{4(2-\alpha)}, \xi_1) &\leq 2d \cdot \exp\Big(- \frac{\alpha^2}{8(2-\alpha)^2} \frac{n\lambda_n^2}{C_{\max}(\eta,\kappa_2)^2}\Big) \\ &\leq 2d \cdot \exp\Big(- \frac{\alpha^2}{8(2-\alpha)^2} n^{1-2a}\Big). \end{aligned} \tag{20}$$

Since $P(A) = P(A \cap B) + P(A \cap B^c) \leq P(A \cap B) + P(B^c)$,

$$P(\frac{\|W_j^n\|_\infty}{\lambda_n} > \frac{\alpha}{4(2-\alpha)}) \quad \leq \quad P(\frac{\|W_j^n\|_\infty}{\lambda_n} > \frac{\alpha}{4(2-\alpha)}, \xi_1) + P(\xi_1^c).$$

Then, the probability bound in Equation (20) and Proposition 17 $P(\xi_1^c) \leq M \cdot \eta^{-2}$ directly implies that

$$P(\frac{\|W_j^n\|_\infty}{\lambda_n} > \frac{\alpha}{4(2-\alpha)}) \leq 2d \cdot \exp\Big(- \frac{\alpha^2}{8(2-\alpha)^2} n^{1-2a}\Big) + M \cdot \eta^{-2}.$$

■

C.6 Proof for Lemma 22

Proof In order to establish the error bound $\|\widetilde{\theta}_S - \theta_S^*\| \leq B$ for some radius B, several works (Yang et al., 2012; Ravikumar et al., 2010, 2011) already proved that it suffices to show $F(u_S) > 0$ for all $u_S := \widetilde{\theta}_S - \theta_S^*$ such that $\|u_S\|_2 = B$ where

$$F(a) := \ell_j(\theta_S^* + a; X^{1:n}) - \ell_j(\theta_S^*; X^{1:n}) + \lambda_n(\|\theta_S^* + a\|_1 - \|\theta_S^*\|_1). \tag{21}$$

More specifically, since $u_S = \widetilde{\theta}_S - \theta_S^*$ is the minimizer of F and $F(0) = 0$ by the construction of Equation (21), $F(u_S) \leq 0$. Note that F is convex, and therefore we have $F(u_S) < 0$. Next we claim that $\|u_S\|_2 \leq B$. In fact, if u_S lies outside the ball of radius B, then the convex combination $v \cdot u_S + (1-v) \cdot 0$ would lie on the boundary of the ball, for an appropriately chosen $v \in (0, 1)$. By convexity,

$$F(v \cdot u_S + (1-v) \cdot 0) \leq v \cdot F(u_S) + (1-v) \cdot 0 \leq 0 \tag{22}$$

contradicting the assumed strict positivity of F on the boundary.

Thus it suffices to establish strict positivity of F on the boundary of the ball with radius $B := M_1 \lambda_n \sqrt{d}$ where $M_1 > 0$ is a parameter to be chosen later in the proof. Let $u_S \in \mathbb{R}^{|S|}$ be an arbitrary vector with $\|u_S\|_2 = B$. By the Taylor series expansion of F (21),

$$F(u_S) = (W_S^n)^T u_S + u_S^T [\nabla^2 \ell_j(\theta_M^* + v u_S; x)] u_S + \lambda_n(\|\theta_S^* + u_S\|_1 - \|\theta_S^*\|_1), \tag{23}$$

for some $v \in [0, 1]$. Since $\|W_S^n\|_\infty \leq \frac{\lambda_n}{4}$ by assumption and $\|u_S\|_1 \leq \sqrt{d}\|u_S\|_2 \leq \sqrt{d} \cdot B$, the first term in Equation (23) has the following bound:

$$|(W_S^n)^T u_S| \leq \|W_S^n\|_\infty \|u_S\|_1 \leq \|W_S^n\|_\infty \sqrt{d}\|u_S\|_2 \leq (\lambda_n \sqrt{d})^2 \frac{M_1}{4}.$$

Applying the triangle inequality to the last part of Equation (23), we have the following bound.

$$\lambda_n(\|\theta_S^* + u_S\|_1 - \|\theta_S^*\|_1) \geq -\lambda_n \|u_S\|_1 \geq -\lambda_n \sqrt{d}\|u_S\|_2 = -M_1(\lambda_n \sqrt{d})^2.$$

Next we bound $\lambda_{\min}\left(\nabla^2 \ell_j(\theta_S^* + v u_S)\right)$ where $\lambda_{\min}(\cdot)$ is the minimum eigenvalue of a matrix:

$$\begin{aligned} q^* &:= \lambda_{\min}\left(\nabla^2 \ell_j(\theta_S^* + v u_S)\right) \\ &\geq \min_{v \in [0,1]} \lambda_{\min}\left(\nabla^2 \ell_j(\theta_S^* + v u_S)\right) \\ &\geq \lambda_{\min}\left(\nabla^2 \ell_j(\theta_S^*)\right) - \max_{v \in [0,1]} \|\frac{1}{n} \sum_{i=1}^n A_j'''(\langle \theta_S^* + v u_S, X_S \rangle) u_S^T X_S^{(i)} X_S^{(i)} (X_S^{(i)})^T\|_2 \\ &\geq \rho_{\min} - \max_{v \in [0,1]} \max_{y : \|y\|_2 = 1} \frac{1}{n} \sum_{i=1}^n |A_j'''(\langle \theta_S^* + v u_S, X_S \rangle)| \cdot |u_S^T X_S^{(i)}| \cdot (y^T X_S^{(i)})^2. \end{aligned} \tag{24}$$

Next we define the event ξ_2 in order to bound $A_j'''(\langle \theta_S^* + v u_S, X_S \rangle)$.

$$\xi_2 := \{ \max_{i \in \{1, \cdots, n\}} \langle \theta_S^* + v u_S, X_S^{(i)} \rangle < \kappa_1 \log \eta \}.$$

On ξ_2, Assumption 11 is satisfied and

$$A_j'''(\langle \theta_S^* + v u_S, X_S \rangle) \leq n^{\kappa_2}. \tag{25}$$

In addition, we bound the second term in Equation (24). Recall that $\|X_S^{(i)}\|_\infty \leq 4 \log \eta$ for all $i \in \{1, 2, \cdots, n\}$ on ξ_1. Since $\|u_S\|_1 \leq \sqrt{d}\|u_S\|_2 \leq \sqrt{d} \cdot B$,

$$|u_S^T X_S^{(i)}| \leq 4 \log(\eta) \sqrt{d} \|u_S\|_2 \leq 4 \log(\eta) \cdot M_1 \lambda_n d. \tag{26}$$

Lastly, it is clear that $\max_{y : \|y\|_2 = 1} (y^T X_S^{(i)})^2 \leq \rho_{\max}$ by the definition of the maximum eigenvalue and Assumption 8. Together with the bounds of Equations (25) and (26) on the events ξ_1 and ξ_2,

$$q^* \leq \rho_{\min} - 4n^{\kappa_2} \log(\eta) \cdot M_1 \lambda_n d \ \rho_{\max}.$$

For $\lambda_n \leq \frac{\rho_{\min}}{8n^{\kappa_2} \log(\eta) M_1 d \rho_{\max}}$, we have $q^* \leq \frac{\rho_{\min}}{2}$. Therefore,

$$F(u) \geq (\lambda_n \sqrt{n})^2 \Big\{ - \frac{1}{4} M_1 + \frac{\rho_{\min}}{2} M_1^2 - M_1 \Big\},$$

which is strictly positive for $M_1 = \frac{5}{\rho_{\min}}$. Therefore for $\lambda_n \leq \frac{\rho_{\min}^2}{40 n^{\kappa_2} \log(\eta) d \rho_{\max}}$ given ξ_1 and ξ_2,

$$\|\widetilde{\theta}_S - \theta_S^*\|_2 \leq \frac{5}{\rho_{\min}} \sqrt{d} \lambda_n.$$

Since $P(A) = P(A \cap B \cap C) + P(A \cap (B \cap C)^c) \leq P(A \cap B \cap C) + P(B^c) + P(C^c)$,

$$P\left(\|\widetilde{\theta}_S - \theta^*_S\|_2 > \frac{5}{\rho_{\min}}\sqrt{d}\lambda_n\right) \leq P\left(\|\widetilde{\theta}_S - \theta^*_S\|_2 > \frac{5}{\rho_{\min}}\sqrt{d}\lambda_n, \xi_1, \xi_2\right) + P(\xi_1^c) + P(\xi_2^c).$$

Here the probability of ξ_2^c is upped bounded as follows.

$$\begin{aligned} P(\xi_2^c) &\overset{(a)}{\leq} n \max_i P(\langle \theta^*_{M_j} + v u_S, X_S^{(i)} \rangle > \kappa_1 \log \eta) \\ &\overset{(b)}{\leq} n \cdot M \cdot \eta^{-\frac{\kappa_1}{2\|\theta^*_{M_j}\|_1}} \\ &\overset{(c)}{\leq} M \cdot \eta^{-2}. \end{aligned}$$

(a) follows from the union bound, and (b) follows from Proposition 18, and $\|u_S\|_1 \leq \sqrt{d}\|u_S\|_2 \leq dM_1\lambda_n \leq \|\theta^*_{M_j}\|_1$ and $\min_{j\in V} \min_{t\in S} |[\theta^*_M]_t| \geq \frac{10}{\rho_{\min}}\sqrt{d}\lambda_n$. Lastly (c) follows from Assumption 11 that $\kappa_1 \geq 6\|\theta^*_{M_j}\|_1$.

In addition the probability bound of ξ_1^c is provided in Proposition 17. Therefore

$$P\left(\|\widetilde{\theta}_S - \theta^*_S\|_2 \leq \frac{5}{\lambda_{\min}}\sqrt{d}\ \lambda_n\right) \geq 1 - 2M \cdot \eta^{-2}.$$

■

C.7 Proof for Lemma 23

Proof According to Equation (17), R^n_{jt} for any $t \in S$ can be expressed as

$$\begin{aligned} R^n_{jt} &= \frac{1}{n}\sum_{i=1}^n [\bigtriangledown^2 \ell_j(\theta^*_{M_j}; X^{1:n}) - \bigtriangledown^2 \ell_j(\bar{\theta}^{(t)}_{M_j}; X^{1:n})]^T_t (\widetilde{\theta} - \theta^*_{M_j}) \\ &= \frac{1}{n}\sum_{i=1}^n [A''_j(\langle \theta^*_S, X^{(i)}_{V\setminus j}\rangle) - A''_j(\langle \bar{\theta}^{(t)}_{M_j}, X^{(i)}_{V\setminus j}\rangle)][X^{(i)}_{V\setminus j}(X^{(i)}_{V\setminus j})^T]^T_t (\widetilde{\theta} - \theta^*_{M_j}) \end{aligned}$$

for $\bar{\theta}^{(t)}_{M_j}$ which is some point in the line between $\widetilde{\theta}_{M_j}$ and $\theta^*_{M_j}$ (i.e., $\bar{\theta}^{(t)}_{M_j} = v\cdot\widetilde{\theta}_{M_j} + (1-v)\cdot\theta^*_{M_j}$ for some $v \in [0,1]$).

By the mean value theorem,

$$R^t_{jt} = \frac{1}{n}\sum_{i=1}^n \left\{A'''_j(\langle \bar{\bar{\theta}}^{(t)}_{M_j}, X^{(i)}_{V\setminus j}\rangle) X^{(i)}_t\right\}\left\{v(\widetilde{\theta}_{M_j} - \theta^*_{M_j})^T X^{(i)}_{V\setminus j}(X^{(i)}_{V\setminus j})^T(\widetilde{\theta}_{M_j} - \theta^*_{M_j})\right\}$$

for $\bar{\bar{\theta}}^{(t)}_{M_j}$ which is a point on the line between $\bar{\theta}^{(t)}_{M_j}$ and $\theta^*_{M_j}$.

By Proposition 17, $\max_{i,j} |X^{(i)}_j| \leq 4\log\eta$ given ξ_1. Furthermore in Section C.6, we showed that $A'''_j(\langle \bar{\bar{\theta}}^{(t)}_{M_j}, X_{M\setminus j}\rangle) \leq n^{\kappa_2}$ given ξ_2 . Therefore, on ξ_1 and ξ_2, it follows that:

$$|R^n_{jt}| \leq 4n^{\kappa_2}\log(\eta)\rho_{\max}\|\widetilde{\theta} - \theta^*_M\|_2^2.$$

We showed that $\|\widetilde{\theta} - \theta^*_M\|_2 \leq \frac{5}{\rho_{\min}}\sqrt{d}\lambda_n$ for $\lambda_n \leq \frac{\alpha}{400(2-\alpha)}\frac{\rho^2_{\min}}{\rho_{\max}}\frac{1}{dn^{\kappa_2}\log(\eta)}$ given ξ_1 and ξ_2 in the proof of Lemma 22. Therefore we obtain

$$\|R^n\|_\infty \leq \frac{100\rho_{\max}}{\rho^2_{\min}} d\ n^{\kappa_2}\log(\eta)\ \lambda_n^2 \leq \frac{\alpha\lambda_n}{4(2-\alpha)}.$$

Since $P(A) = P(A \cap B \cap C) + P(A \cap (B \cap C)^c) \leq P(A \cap B \cap C) + P(B^c) + P(C^c)$,

$$P\left(\|R^n\|_\infty > \frac{\alpha\lambda_n}{4(2-\alpha)}\right) \leq P\left(\|R^n\|_\infty > \frac{\alpha\lambda_n}{4(2-\alpha)}, \xi_1, \xi_2\right) + P(\xi_1^c) + P(\xi_2^c).$$

Putting the probability bounds for ξ_1^c and ξ_2^c specified in Proposition 17 and Section C.6 together, we have

$$P\left(\|R^n_j\|_\infty \leq \frac{\alpha\lambda_n}{4(2-\alpha)}\right) \geq 1 - 2M \cdot \eta^{-2}.$$

■

Appendix D. Proof for Theorem 14

Proof Without loss of generality, assume that the true ordering is $\pi^* = (1, 2, \cdots, p)$. Let $T_j(X_j) := \omega_j X_j$ where $\omega_j = (\beta_0 + \beta_1 \mathbb{E}(X_j \mid X_{\text{pa}(j)}))^{-1}$ (specified in Proposition 3). For any node $j \in V$ and $S \subset V \setminus \{j\}$, let $\mu_{j|S}$ and $\sigma^2_{j|S}$ represent $\mathbb{E}(T_j(X_j) \mid X_S)$ and $\text{Var}(T_j(X_j) \mid X_S)$ respectively. For realizations x_S, let $\mu_{j|S}(x_S)$ and $\sigma^2_{j|S}(x_S)$ denote $\mathbb{E}(T_j(X_j) \mid X_S = x_S)$ and $\text{Var}(T_j(X_j) \mid X_S = x_S)$, respectively. Let $n(x_S) = \sum_{i=1}^n \mathbf{1}(X_S^{(i)} = x_S)$ denote the total conditional sample size, and $n_S = \sum_{x_S} n(x_S)\mathbf{1}(n(x_S) \geq c_0 \cdot n)$ for an arbitrary $c_0 \in (0,1)$ to denote the truncated conditional sample size.

Let E^m denote the set of undirected edges corresponding to the *moralized* graph. Recall the definitions $\mathcal{N}(j) = \{k \in V : (j,k) \text{ or } (k,j) \in E^m\}$ denote the neighborhood set of node j in the moralized graph, $C_{jk} = \mathcal{N}(k) \cap \{\pi_1, \pi_2, \cdots, \pi_{j-1}\}$. Since we assume the structure of the moralized graph is provided, $\widehat{C}_{jk} = C_{jk}$. Hence C_{jk} is used instead of an estimated set $\widehat{C}_{jk}$.

The overdispersion score of node $k \in V \setminus \{\pi_1, ..., \pi_{j-1}\}$ for the j^{th} component of the ordering π_j only depends on $\mathcal{X}(C_{jk}) = \{x \in \{X^{(1)}_{C_{jk}}, X^{(2)}_{C_{jk}}, \cdots, X^{(n)}_{C_{jk}}\} : n(x) \geq c_0 \cdot n\}$, so we only count up elements that occur sufficiently frequently.

According to the generalized ODS algorithm, the truncated sample conditional mean and variance of $T_j(X_j)$ given $X_S = x_S$ are:

$$\begin{aligned} \widehat{\mu}_{j|S}(x_S) &:= \frac{1}{n_S(x_S)}\sum_{i=1}^n T_j(X_j^{(i)})\mathbf{1}(X_S^{(i)} = x_S), \\ \widehat{\sigma}^2_{j|S}(x_S) &:= \frac{1}{n_S(x_S) - 1}\sum_{i=1}^n (T_j(X_j^{(i)}) - \widehat{\mu}_{j|S}(x_S))^2\mathbf{1}(X_S^{(i)} = x_S). \end{aligned}$$

We rewrite the overdispersion score (5) of node $k \in V \setminus \{\pi_1, ..., \pi_{j-1}\}$ for π_j as follows:

$$\begin{aligned}
\widehat{\mathcal{S}}(1,k) &:= \left[\left(\frac{\widehat{\sigma}_k}{\beta_0+\beta_1\widehat{\mu}_k}\right)^2 - \frac{\widehat{\mu}_k}{\beta_0+\beta_1\widehat{\mu}_k}\right], \\
\widehat{\mathcal{S}}(m,k) &:= \sum_{x\in\mathcal{X}(C_{mk})} \frac{n(x)}{n_{C_{mk}}}\left[\left(\frac{\widehat{\sigma}_{k|C_{mk}}(x)}{\beta_0+\beta_1\widehat{\mu}_{k|C_{mk}}(x)}\right)^2 - \frac{\widehat{\mu}_{k|C_{mk}}(x)}{\beta_0+\beta_1\widehat{\mu}_{k|C_{mk}}(x)}\right].
\end{aligned}$$

For notational convenience, let each entry of the overdispersion score $\widehat{\mathcal{S}}(m,k)$ for $x \in \mathcal{X}(C_{mk})$ be defined as:

$$\widehat{\mathcal{S}}(m,k)(x) := \left(\frac{\widehat{\sigma}_{k|C_{mk}}(x)}{\beta_0+\beta_1\widehat{\mu}_{k|C_{mk}}(x)}\right)^2 - \frac{\widehat{\mu}_{k|C_{mk}}(x)}{\beta_0+\beta_1\widehat{\mu}_{k|C_{mk}}(x)}. \tag{27}$$

The true overdispersion scores are:

$$\begin{aligned}
\mathcal{S}^*(1,k) &:= \left[\left(\frac{\sigma_k}{\beta_0+\beta_1\mu_k}\right)^2 - \frac{\mu_k}{\beta_0+\beta_1\mu_k}\right], \\
\mathcal{S}^*(m,k) &:= \sum_{x\in\mathcal{X}(C_{mk})} \frac{n(x)}{n_{C_{mk}}}\left[\left(\frac{\sigma_{k|C_{mk}}(x)}{\beta_0+\beta_1\mu_{k|C_{mk}}(x)}\right)^2 - \frac{\mu_{k|C_{mk}}(x)}{\beta_0+\beta_1\mu_{k|C_{mk}}(x)}\right], \\
\mathcal{S}^*(m,k)(x) &:= \left(\frac{\sigma_{k|C_{mk}}(x)}{\beta_0+\beta_1\mu_{k|C_{mk}}(x)}\right)^2 - \frac{\mu_{k|C_{mk}}(x)}{\beta_0+\beta_1\mu_{k|C_{mk}}(x)} \quad \text{for } x \in \mathcal{X}(C_{mk}).
\end{aligned}$$

Next we introduce Proposition 24 which ensures the each component of the true overdispersion score $\mathcal{S}^*(m,k)(x)$ for $k \neq \pi_m$ is bounded away from $m_{\min} > 0$.

Proposition 24 *For all $j \in V$, $pa_0(j) \subset pa(j)$, $pa_0(j) \neq \emptyset$ and $S \subset nd(j) \setminus pa_0(j)$, there exists $m_{\min} > 0$ such that*

$$Var(T_j(X_j) \mid X_S) - \mathbb{E}(T_j(X_j) \mid X_S) > m_{\min}.$$

Now we define the following two events: For any $j \in V$, $k \in V \setminus \{\pi_1, ..., \pi_{j-1}\}$ and $m \in \{1, 2, ..., p-1\}$

$$\begin{aligned}
\xi_1 &:= \{\max_j \max_{i\in\{1,2,\cdots,n\}} |X_j^{(i)}| < 4\log\eta\} \\
\xi_3 &:= \{\max_{m,k} |\widehat{\mathcal{S}}(m,k) - \mathcal{S}^*(m,k)| < \frac{m_{\min}}{2}\}.
\end{aligned}$$

Then,

$$\begin{aligned}
P(\widehat{\pi} \neq \pi^*) &\overset{(a)}{\leq} P(\widehat{\pi} \neq \pi^*, \xi_3) + P(\xi_3^c, \xi_1) + P(\xi_1^c) \\
&\overset{(b)}{\leq} P(\widehat{\pi}_1 \neq \pi_1^*, \xi_3) + P(\widehat{\pi}_2 \neq \pi_2^*, \xi_3 \mid \widehat{\pi}_1 = \pi_1^*) + \cdots \\
&\quad + P(\widehat{\pi}_p \neq \pi_p^*, \xi_3 \mid \widehat{\pi}_1 = \pi_1^*, \cdots, \widehat{\pi}_{p-1} = \pi_{p-1}^*) + P(\xi_3^c, \xi_1) + P(\xi_1^c). \quad (28)
\end{aligned}$$

(a) follows from $P(A) \leq P(A \cap B) + P(B^c)$, and (b) follows from the induction and the fact $P(A \cup B) = P(A) + P(B \cap A^c) = P(A) + P(B \mid A^c)P(A^c) \leq P(A) + P(B \mid A^c)$.

We prove the probability bound (28) by induction. For the first step ($m = 1$), overdispersion scores of π_1 in Equation (4) are used where a set of candidate element of π_1 is $\{1, 2, \cdots, p\}$. Then,

$$\begin{aligned}
P(\widehat{\pi}_1 \neq \pi_1^*, \xi_3) &= P\left(\exists k' \in V \setminus \{\pi_1^*\} \text{ such that } \widehat{\mathcal{S}}(1,\pi_1^*) > \widehat{\mathcal{S}}(1,k'), \xi_3\right) \\
&\overset{(a)}{\leq} (p-1) \max_{k'\in V\setminus\{\pi_1^*\}} P\left(\mathcal{S}^*(1,\pi_1^*) + \frac{m_{\min}}{2} > \mathcal{S}^*(1,k') - \frac{m_{\min}}{2}, \xi_3\right) \\
&\overset{(b)}{=} (p-1) \max_{k'\in V\setminus\{\pi_1^*\}} P\left(m_{\min} > \mathcal{S}^*(1,k), \xi_3\right) \\
&\overset{(c)}{=} 0.
\end{aligned}$$

(a) follows from the union bound and the definition of ξ_3. (b) follows from that $\mathcal{S}^*(1,\pi_1^*) = 0$ by the property of the transformation $T_j(\cdot)$ specified in Proposition 3, and (c) follows from Proposition 24.

For the $m = (j-1)^{th}$ step, assume that the first $j-1$ elements of the estimated ordering are correct $(\widehat{\pi}_1, \widehat{\pi}_2, \cdots, \widehat{\pi}_{j-1}) = (\pi_1^*, \cdots, \pi_{j-1}^*)$. Then for the $m = j^{th}$ step, we consider the probability of a false recovery of π_j^* given $(\pi_1^*, \cdots, \pi_{j-1}^*)$. Using the same argument as the first step, the following result is straightforward.

$$\begin{aligned}
P(\widehat{\pi}_j \neq \pi_j^*, \xi_3 \mid \pi_1^*, \cdots, \pi_{j-1}^*) &= P\left(\exists k \in V \setminus \{\pi_j^*\} \text{ such that } \widehat{\mathcal{S}}(j,\pi_j^*) > \widehat{\mathcal{S}}(j,k), \xi_3\right) \\
&\overset{(a)}{\leq} p \max_{k\in V\setminus\{\pi_j^*\}} P\left(\mathcal{S}^*(j,\pi_j^*) + \frac{m_{\min}}{2} > \mathcal{S}^*(j,k) - \frac{m_{\min}}{2}, \xi_3\right) \\
&\overset{(b)}{=} p \max_{k\in V\setminus\{\pi_j^*\}} P\left(m_{\min} > \mathcal{S}^*(j,k), \xi_3\right) \\
&\overset{(c)}{=} 0.
\end{aligned}$$

Therefore, for any $j \in V$,

$$P(\widehat{\pi}_j \neq \pi_j^*, \xi_3 \mid \widehat{\pi}_1 = \pi_1^*, \cdots, \widehat{\pi}_{j-1} = \pi_{j-1}^*) = 0.$$

Then, the probability bound (28) is reduced to $P(\widehat{\pi} \neq \pi^*) \leq P(\xi_3^c, \xi_1) + P(\xi_1^c)$. Note that $P(\xi_1^c) \leq M \cdot \eta^{-2}$ by Proposition 17. The following lemma provides the upper bound of $P(\xi_3^c, \xi_1)$.

Lemma 25 *There exist positive constants C_1 and C_2 such that*

$$P(\xi_3^c, \xi_1) \leq C_1 p^2 c_0^{-1} exp\left(-C_2 \frac{c_0 \cdot n}{\log^4 \eta}\right).$$

where c_0 is the sample cut-off parameter.

Lastly, we define a condition on the sample cut-off parameter c_0. Intuitively if c_0 is too small, the estimated overdispersion scores may be biased due to the lack of samples.

In contrast, if c_0 is too large, all components of the conditioning set C_{mk} may not have enough samples size ($> c_0 \cdot n$), and therefore overdispersion scores cannot be calculated. The following proposition provides a maximum value of c_0 ensuring that overdispersion scores exist.

Proposition 26 *On the event ξ_1, if $c_0 \leq (3\log(\eta))^{-d}$ then the conditioning set C_{mk} has at least $c_0 \cdot n$ samples.*

The combination of Lemma 25 and Proposition 26 imply that for some C_1 and C_2

$$P(\xi_3^c, \xi_1) \leq C_1 p^2 \log^d(\eta) \exp\left(-C_2 \frac{n}{(\log(\eta))^{4+d}}\right).$$

Therefore,

$$P(\widehat{\pi} \neq \pi^*) \leq C_1 p^2 \log^d(\eta) \exp\left(-C_2 \frac{n}{\log^{4+d}\eta}\right) + \frac{M}{\eta^2}.$$

■

D.1 Proof for Proposition 24

Proof In the proof of the identifiability theorem in Appendix A, we obtain

$$\mathrm{Var}(T_j(X_j) \mid X_S) - \mathbb{E}(T_j(X_j) \mid X_S) = \frac{(1+\beta_1)\mathrm{Var}(\mathbb{E}(X_j \mid X_{pa(j)}) \mid X_S)}{(\beta_0 + \beta_1 \mathbb{E}(X_j \mid X_S))^2}.$$

By Assumption 13, $\mathrm{Var}(\mathbb{E}(X_j \mid X_{pa(j)}) \mid X_S) > M_{\min}$ and $|\beta_{j0} + \beta_{j1}\mathbb{E}(X_j \mid X_S)| > \omega_{\min}$. Then,

$$\mathrm{Var}(T_j(X_j) \mid X_S) - \mathbb{E}(T_j(X_j) \mid X_S) \geq \frac{(1+\beta_1)M_{\min}}{\omega_{\min}^2}.$$

Since $\beta_1 > -1$, the proof is complete. ■

D.2 Proof for Proposition 26

Proof Let $|X_S|$ denote the cardinality of a set $\{X_S^{(1)}, X_S^{(2)}, \cdots, X_S^{(n)}\}$ and $|\mathcal{X}(S)|$ denote the cardinality of the truncated set $\mathcal{X}(S) := \{x \in \{X_S^{(1)}, X_S^{(2)}, \cdots, X_S^{(n)}\} : n(x) \geq c_0 \cdot n\}$.

If $|\mathcal{X}(S)| = 1$, for all $x \in \{X_S^{(1)}, X_S^{(2)}, \cdots, X_S^{(n)}\}$, $n_S(x) = c_0 \cdot n - 1$ except for a single $z \in \mathcal{X}(S)$ where $n_S(z) \geq c_0.n$. In this case, the total sample size $n = n_S(z) + (|X_S| - 1)(c_0 \cdot n - 1)$. Hence

$$n_S(z) = n - (|X_S| - 1)(c_0.n - 1) = n - c_0 \cdot n \cdot |X_S| + c_0 \cdot n + |X_S| - 1.$$

Since $c_0 \cdot n \leq n_S(z)$,

$$c_0 \leq \frac{n + |X_S| - 1}{n \cdot |X_S|}.$$

Note that $\frac{1}{|X_S|} \leq \frac{n+|X_S|-1}{n \cdot |X_S|}$ and $|X_j^{(i)}| \leq 4\log(\eta)$ for all $j \in V$ and $i \in \{1, 2, \cdots, n\}$ given ξ_1. Then the maximum cardinality of X_S is $(4\log(\eta))^{|S|}$. Hence if $c_0 \leq (4\log(\eta))^{-|S|}$ there exists a $z \in \mathcal{X}(S)$.

Recall that the size of a candidate parents set C_{mk} is bounded by the maximum degree of the moralized graph d. Therefore if $c_0 \leq 4\log(\eta)^{-d}$, there exists at least one $z \in \mathcal{X}(C_{mk})$. ■

D.3 Proof for Lemma 25

Proof For ease of notation, let $n_{mk} = n_{C_{mk}}$ and $n_{mk}(x) = n_{C_{mk}}(x)$ for $x \in \mathcal{X}(C_{mk})$. Using the union bound, for $m \in \{1, 2, ..., p-1\}$ and $k \in V \setminus \{\pi_1, ..., \pi_{j-1}\}$

$$\begin{aligned} P(\xi_3^c, \xi_1) &= P(\max_{m,k} |\widehat{\mathcal{S}}(m,k) - \mathcal{S}^*(m,k)| > \frac{m_{\min}}{2}, \xi_1) \\ &\leq p^2 \max_{m,k} P(|\widehat{\mathcal{S}}(m,k) - \mathcal{S}^*(m,k)| > \frac{m_{\min}}{2}, \xi_1). \end{aligned}$$

Since overdispersion scores have an additive form,

$$P(|\widehat{\mathcal{S}}(m,k) - \mathcal{S}^*(m,k)| > \frac{m_{\min}}{2}, \xi_1) \leq P(\sum_{x \in \mathcal{X}(C_{mk})} \frac{n_{mk}(x)}{n_{mk}} |\widehat{\mathcal{S}}(m,k)(x) - \mathcal{S}^*(m,k)(x)| > \frac{m_{\min}}{2}, \xi_1).$$

Applying $P(\sum_i Y_i > \delta) \leq \sum_i P(Y_i > \omega_i \delta)$ for any $\delta \in \mathbb{R}$ and $\omega_i \in \mathbb{R}^+$ such that $\sum_i \omega_i = 1$, we have

$$\begin{aligned} P(\sum_{x \in \mathcal{X}(C_{mk})} \frac{n_{mk}(x)}{n_{mk}} |\widehat{\mathcal{S}}(m,k)(x) - \mathcal{S}^*(m,k)(x)| > \frac{m_{\min}}{2}, \xi_1) \\ \leq \sum_{x \in \mathcal{X}(C_{mk})} P(|\widehat{\mathcal{S}}(m,k)(x) - \mathcal{S}^*(m,k)(x)| > \frac{m_{\min}}{2}, \xi_1). \end{aligned}$$

Applying the union bound,

$$\begin{aligned} \sum_{x \in \mathcal{X}(C_{mk})} P(|\widehat{\mathcal{S}}(m,k)(x) - \mathcal{S}^*(m,k)(x)| > \frac{m_{\min}}{2}, \xi_1) \\ \leq |\mathcal{X}(C_{mk})| \max_{x \in \mathcal{X}(C_{mk})} P(|\widehat{\mathcal{S}}(m,k)(x) - \mathcal{S}(m,k)^*(x)| > \frac{m_{\min}}{2}, \xi_1). \end{aligned}$$

Since we only consider $x \in \mathcal{X}(C_{mk})$, it follows that $n_{mk}(x) \geq c_0 \cdot n$. Further since the total truncated sample size is less than total sample size, $c_0 \cdot n \cdot |\mathcal{X}(C_{mk})| \leq n$, and therefore the cardinality of C_{mk} is at most c_0^{-1}. Hence

$$\begin{aligned} |\mathcal{X}(C_{mk})| \max_{x \in \mathcal{X}(C_{mk})} P(|\widehat{\mathcal{S}}(m,k)(x) - \mathcal{S}^*(m,k)(x)| > \frac{m_{\min}}{2}, \xi_1) \\ \leq c_0^{-1} \max_{x \in \mathcal{X}(C_{mk})} P(|\widehat{\mathcal{S}}(m,k)(x) - \mathcal{S}^*(m,k)(x)| > \frac{m_{\min}}{2}, \xi_1). \end{aligned}$$

Since the overdispersion score is the difference between the conditional mean and conditional variance, the remainder of the proof is reduced to finding the sample complexity for

the sample conditional mean and variance. Suppose that $\epsilon := \widehat{\mu}_{k|C_{mk}}(x) - \mu_{k|C_{mk}}(x)$ and $\kappa \cdot \epsilon := \widehat{\sigma}^2_{k|C_{mk}}(x) - \sigma^2_{k|C_{mk}}(x)$ for some $\kappa \in \mathbb{R}$. By the definition of the overdispersion scores in Equation (27), we have

$$
\begin{aligned}
&\{\epsilon : |\widehat{\mathcal{S}}(m,k)(x) - \mathcal{S}^*(m,k)(x)| > \frac{m_{\min}}{2}\} \\
&\subset \left\{\epsilon : \left| \left(\frac{\sigma_{k|C_{mk}}(x) + \kappa\epsilon}{\beta_0 + \beta_1 \mu_{k|C_{mk}}(x) + \epsilon}\right)^2 - \frac{\mu_{k|C_{mk}}(x) + \epsilon}{\beta_0 + \beta_1 \mu_{k|C_{mk}}(x) + \epsilon} \right.\right. \\
&\qquad \left.\left. - \left(\frac{\sigma_{k|C_{mk}}(x)}{\beta_0 + \beta_1 \mu_{k|C_{mk}}(x)}\right)^2 - \frac{\mu_{k|C_{mk}}(x)}{\beta_0 + \beta_1 \mu_{k|C_{mk}}(x)} \right| > \frac{m_{\min}}{2} \right\} \\
&= \{\epsilon : \epsilon \in (\epsilon_1, \epsilon_2) \cup (\epsilon_3, \epsilon_4)\}\,.
\end{aligned}
$$

where $\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4$ are constants that depend on $\mu, \sigma^2, \beta_0, \beta_1, m_{\min}$, and κ and are constructed as follows:

$$
\begin{aligned}
\zeta_1(\mu,\sigma^2,\beta_0,\beta_1,m_{\min},\kappa) &:= \beta_0^3(1+\beta_1 m_{\min}) - \beta_1^4 m_{\min}\mu^3 + 2\beta_1^2\mu^2\kappa\sigma^2 - 2\beta_1^2\mu\sigma^4 \\
&\quad + \beta_0^2(-2\beta_1\mu - 3\beta_1^2 m_{\min}\mu + 2\kappa\sigma^2) - \beta_0\beta_1\{\beta_1\mu^2 + 3\beta_1^2 m_{\min}\mu^2 + 2\sigma^2(-2\kappa\mu + \sigma^2)\}, \\
\zeta_2(\mu,\sigma^2,\beta_0,\beta_1,m_{\min},\kappa) &:= (\beta_0+\beta_1\mu)^2\Big[\beta_0^4(1+2\kappa\mu) + 2\beta_1^2(\kappa\mu - \sigma^2)^2(\beta_1^2\mu^2 m_{\min} + 2\sigma^4) \\
&\quad + 4\beta_0\beta_1(\kappa\mu - \sigma^2)\{\beta_1^2\mu m_{\min}(2\kappa\mu - \sigma^2) + \beta_1\mu\sigma^2 - 2\kappa\sigma^2\} \\
&\quad + 2\beta_0^3\{-2\kappa\sigma^2 + \beta_1(\mu + 4m_{\min}\kappa^2\mu - 2m_{\min}\kappa\sigma^2)\} \\
&\quad + \beta_0^2\{4\kappa^2\sigma^4 + 4\beta_1\sigma^2(-2\kappa\mu + \sigma^2) + \beta_1^2(\mu^2 + 12m_{\min}\kappa^2\mu^2 - 12m_{\min}\mu\kappa\sigma^2 + 2m_{\min}\sigma^4)\}\Big], \\
\zeta_3(\mu,\sigma^2,\beta_0,\beta_1,m_{\min},\kappa) &:= \beta_0^2(-2\kappa^2 + 2\beta_1 + \beta_1^2 m_{\min}) + 2\beta_0\beta\mu(\beta_1 + \beta_1^2 m_{\min} - \kappa^2) \\
&\quad + \beta_1^2(\beta_1^2 m_{\min}\mu^2 + 2\sigma^4 - 2\kappa^2\mu^2).
\end{aligned}
$$

Given $\zeta_1, \zeta_2, \zeta_3$,

$$
\begin{aligned}
\epsilon_1' &= \frac{\zeta_1(\mu_{k|C_{mk}}(x), \sigma^2_{k|C_{mk}}(x), \beta_0, \beta_1, m_{\min}, \kappa) + \sqrt{\zeta_2(\mu_{k|C_{mk}}(x), \sigma^2_{k|C_{mk}}(x), \beta_0, \beta_1, m_{\min}, \kappa)}}{\zeta_3(\mu_{k|C_{mk}}(x), \sigma^2_{k|C_{mk}}(x), \beta_0, \beta_1, m_{\min}, \kappa)}, \\
\epsilon_2' &= \frac{-\zeta_1(\mu_{k|C_{mk}}(x), \sigma^2_{k|C_{mk}}(x), \beta_0, \beta_1, m_{\min}, \kappa) + \sqrt{\zeta_2(\mu_{k|C_{mk}}(x), \sigma^2_{k|C_{mk}}(x), \beta_0, \beta_1, m_{\min}, \kappa)}}{\zeta_3(\mu_{k|C_{mk}}(x), \sigma^2_{k|C_{mk}}(x), \beta_0, \beta_1, m_{\min}, \kappa)}, \\
\epsilon_3' &= \frac{\zeta_1(\mu_{k|C_{mk}}(x), \sigma^2_{k|C_{mk}}(x), \beta_0, \beta_1, -m_{\min}, \kappa) + \sqrt{\zeta_2(\mu_{k|C_{mk}}(x), \sigma^2_{k|C_{mk}}(x), \beta_0, \beta_1, -m_{\min}, \kappa)}}{\zeta_3(\mu_{k|C_{mk}}(x), \sigma^2_{k|C_{mk}}(x), \beta_0, \beta_1, -m_{\min}, \kappa)}, \\
\epsilon_4' &= \frac{-\zeta_1(\mu_{k|C_{mk}}(x), \sigma^2_{k|C_{mk}}(x), \beta_0, \beta_1, -m_{\min}, \kappa) + \sqrt{\zeta_2(\mu_{k|C_{mk}}(x), \sigma^2_{k|C_{mk}}(x), \beta_0, \beta_1, -m_{\min}, \kappa)}}{\zeta_3(\mu_{k|C_{mk}}(x), \sigma^2_{k|C_{mk}}(x), \beta_0, \beta_1, -m_{\min}, \kappa)}.
\end{aligned}
$$

Let $(\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4)$ be the ordered values of $(\epsilon_1', \epsilon_2', \epsilon_3', \epsilon_4')$ from smallest to largest. Since $m_{\min} > 0$ it follows that $\epsilon_1, \epsilon_2 < 0$ and $\epsilon_3, \epsilon_4 > 0$.

For ease of notation, $\epsilon_{\min} = \min\{|\epsilon_2|, |\epsilon_3|\}$. Then,

$$
\{\epsilon : |\widehat{\mathcal{S}}(j,k)(x) - \mathcal{S}^*(j,k)(x)| > \frac{m_{\min}}{2}\} \subset (-\infty, -\epsilon_{\min}) \cup (\epsilon_{\min}, \infty).
$$

Hence

$$
\begin{aligned}
&P\{|\widehat{\mathcal{S}}(m,k)(x) - \mathcal{S}^*(m,k)(x)| > \frac{m_{\min}}{2}\} \\
&\qquad \le P\left(|\widehat{\mu}_{k|C_{mk}}(x) - \mu_{k|C_{mk}}(x)| > \epsilon_{\min}\right) + P\left(|\widehat{\sigma}^2_{k|C_{mk}}(x) - \sigma^2_{k|C_{mk}}(x)| > \kappa\epsilon_{\min}\right).
\end{aligned}
$$

On ξ_1, $\max_{i,j} |X_j^{(i)}| \le 4\log(\eta)$. Furthermore recall that $n_{mk}(x) \ge c_0 \cdot n$. By applying Hoeffding's inequality,

$$
P(|\widehat{\mu}_{k|C_{mk}}(x) - \mu_{k|C_{mk}}(x)| > \epsilon_{\min}, \xi_1) \le 2\exp\left(-\frac{\epsilon^2_{\min} c_0.n}{8\log^2\eta}\right).
$$

Note that sample variance can be decomposed as follows:

$$
\frac{1}{n-1}\left(\sum_i^n X_i^2 - \frac{1}{n}(\sum_i^n X_i)^2\right) = \frac{1}{n}\sum_i^n X_i^2 - \frac{1}{n(n-1)}\sum_{i\neq j} X_i X_j.
$$

Using Hoeffding's inequality for the decomposed sample variance,

$$
P(|\widehat{\sigma}^2_{k|C_{mk}}(x) - \sigma^2_{k|C_{mk}}(x)| > |\kappa| \cdot \epsilon_{\min}, \xi_1) \le 2\exp\left(-\frac{\kappa^2\epsilon^2_{\min} c_0 \cdot n}{128\log^4\eta}\right) + 2\exp\left(-\frac{\kappa^2\epsilon^2_{\min} c_0 \cdot n}{256\log^4\eta}\right).
$$

Therefore,

$$
\begin{aligned}
&P\{|\widehat{\mathcal{S}}(m,k)(x) - \mathcal{S}^*(m,k)(x)| > \frac{m_{\min}}{2}, \xi_1\} \\
&\qquad \le 2\left(\exp\left(-\frac{\epsilon^2_{\min} c_0.n}{8\log^2\eta}\right) + \exp\left(-\frac{\kappa^2\epsilon^2_{\min} c_0 \cdot n}{128\log^4\eta}\right) + \exp\left(-\frac{\kappa^2\epsilon^2_{\min} c_0 \cdot n}{256\log^4\eta}\right)\right).
\end{aligned}
$$

This completes the proof since there exist constants C_1 and C_2 such that

$$
P(\xi_3^c, \xi_1) \le C_1 p^2 c_0^{-1} \exp\left(-C_2 \frac{c_0 \cdot n}{\log^4\eta}\right).
$$

■

Appendix E. Proof for Theorem 15

Proof Once again we use the *primal-dual witness* method used in the the proof for Theorem 12. The only difference is the conditioning set. In this proof, the conditioning set is all elements of the ordering before node j rather than j is $V \setminus \{j\}$. Without loss of generality, we assume the true ordering is $\pi^* = (1, 2, \cdots, p)$. Then the conditioning set is $\{1, 2, \cdots, j-1\}$.

For ease of notation, we define the parameter $\theta \in \mathbb{R}^{j-1}$ since the node j is not penalized in (11). Then, the conditional negative log-likelihood of a GLM (10) for X_j given $X_{1:j-1}$ is:

$$\ell_j^D(\theta; X^{1:n}) = \frac{1}{n}\sum_{i=1}^{n}\left(-X_j^{(i)}\langle\theta, X_{1:j-1}^{(i)}\rangle + A_j(\langle\theta, X_{1:j-1}^{(i)}\rangle)\right).$$

Recall that for any node $j \in V$:

$$\widehat{\theta}_{D_j} := \arg\min_{\theta\in\mathbb{R}^{j-1}} \mathcal{L}_j^D(\theta, \lambda_n^D) = \arg\min_{\theta\in\mathbb{R}^{j-1}} \{\ell_j^D(\theta; X^{1:n}) + \lambda_n^D\|\theta\|_1\}.$$

Using the *sub-differential*, $\widehat{\theta}_{D_j}$ should satisfy the following condition. For notational simplicity, let $S = \mathrm{pa}(j)$ for node $j \in V$.

$$\bigtriangledown_\theta \mathcal{L}_j^D(\widehat{\theta}_{D_j}, \lambda_n^D) = \bigtriangledown_\theta \ell_j^D(\widehat{\theta}_{D_j}; X^{1:n}) + \lambda_n^D \widehat{Z} = 0 \tag{29}$$

where $\widehat{Z} \in \mathbb{R}^{j-1}$ and $\widehat{Z}_t = \mathrm{sign}([\widehat{\theta}_{D_j}]_t)$ if a node $t \in S$, otherwise $|\widehat{Z}_t| < 1$.

By Lemma 19, it is sufficient the show that $|\widehat{Z}_t| < 1$ for all $t \in S$. We note that the restricted solution is $(\widetilde{\theta}_{D_j}, \widetilde{Z})$. Equation (29) with the dual solution $(\widetilde{\theta}_{D_j}, \widetilde{Z})$ can be represented as $\bigtriangledown^2\ell_j^D(\theta^*_{D_j}; X^{1:n})(\widetilde{\theta}_{D_j} - \theta^*_{D_j}) = -\lambda_n^D\widetilde{Z} - W_{Dj}^n + R_{Dj}^n$ by using the mean value theorem where:

(a) W_{Dj}^n is the sample score function,

$$W_{Dj}^n := -\bigtriangledown \ell_j^D(\theta^*_{D_j}; X^{1:n}). \tag{30}$$

(b) $R_{Dj}^n = (R_{Dj1}^n, R_{Dj2}^n, \cdots, R_{Djj-1}^n)$ and R_{Djk}^n is the remainder term by applying coordinate-wise mean value theorem,

$$R_{Djk}^n := [\bigtriangledown^2\ell_j^D(\theta^*_{D_j}; X^{1:n}) - \bigtriangledown^2\ell_j^D(\bar{\theta}_{D_j}^{(k)}; X^{1:n})]_k^T(\widetilde{\theta}_{D_j}^{(k)} - \theta^*_{D_j}) \tag{31}$$

where $\bar{\theta}_{D_j}^{(j)}$ is a vector on the line between $\widetilde{\theta}_{D_j}$ and $\theta^*_{D_j}$ and $[\cdot]_k^T$ is the k^{th} row of a matrix.

Similar to Proposition 20, the following corollary provides a sufficient condition to control $\widetilde{Z}$.

Corollary 27 *Suppose that* $\max(\|W_{Dj}^n\|_\infty, \|R_{Dj}^n\|_\infty) \le \frac{\lambda_n\alpha}{4(2-\alpha)}$. *Then,* $|\widetilde{Z}_t| < 1$ *for all* $t \notin pa(j)$.

Now we introduce the following three corollaries, to verify that the conditions in Proposition 27 hold, and the deviation $\widetilde{\theta}_{M_j} - \theta^*_{D_j}$ is sufficiently small to conclude $\widehat{\mathrm{pa}}(j) = \mathrm{pa}(j)$ with high probability. For ease of notation, let $\eta = \max\{n, p\}$ and For notational convenience, we use $\widetilde{\theta}_S = [\widetilde{\theta}_{D_j}]_S$ and $\widetilde{\theta}_{S^c} = [\widetilde{\theta}_{D_j}]_{S^c}$. Suppose that Assumptions 8, 9, 10, and 11 are satisfied.

Corollary 28 *Suppose that* $\lambda_n^D \ge \frac{16\max\{n^{\kappa_2}\log\eta, \log^2\eta\}}{n^a}$ *for some* $a \in \mathbb{R}$. *Then,*

$$P(\frac{\|W_{Dj}^n\|_\infty}{\lambda_n^D} \le \frac{\alpha}{4(2-\alpha)}) \ge 1 - 2d\cdot exp(-\frac{\alpha^2}{8(2-\alpha)^2}\cdot n^{1-2a}) - M\cdot\eta^{-2}.$$

Corollary 29 *Suppose that* $\|W_{Dj}^n\|_\infty \le \frac{\lambda_n^D}{4}$. *For* $\lambda_n^D \le \frac{\rho_{\min}^2}{40\rho_{\max}}\frac{1}{n^{\kappa_2}\log\eta d}$,

$$P\left(\|\widetilde{\theta}_S - \theta_S^*\|_2 \le \frac{5}{\lambda_{\min}}\sqrt{d}\lambda_n^D\right) \ge 1 - 2M\cdot\eta^{-2}.$$

Corollary 30 *Suppose that* $\|W_{Dj}^n\|_\infty \le \frac{\lambda_n^D}{4}$. *For* $\lambda_n^D \le \frac{\alpha}{400(2-\alpha)}\frac{\rho_{\min}^2}{\rho_{\max}}\frac{1}{n^{\kappa_2}d\log\eta}$,

$$P\left(\|R_{Dj}^n\|_\infty \le \frac{\alpha\lambda_n^D}{4(2-\alpha)}\right) \ge 1 - 2M\cdot\eta^{-2}.$$

Consider the choice of regularization parameter $\lambda_n^D = \frac{16\max\{n^{\kappa_2}\log\eta, \log^2\eta\}}{n^a}$ where $a \in (2\kappa_2, 1/2)$. Then, the condition for Corollary 28 is satisfied, and therefore $\|W_{Dj}^n\|_\infty \le \frac{\lambda_n^D}{4}$. Moreover, the conditions for Corollaries 29 and 30 are satisfied for a sufficiently large sample size $n \ge D'\max\{(d\log^2\eta)^{\frac{1}{a-2\kappa_2}}, (d\log^3\eta)^{\frac{1}{a-\kappa_2}}\}$ for a positive constant D'. Therefore, there exist some positive constants D_1, D_2 and D_3 such that

$$\|\widetilde{Z}_{S^c}\|_\infty \le (1-\alpha) + (2-\alpha)\left[\frac{\|W_{Dj}^n\|_\infty}{\lambda_n^D} + \frac{\|R_{Dj}^n\|_\infty}{\lambda_n^D}\right] \le (1-\alpha) + \frac{\alpha}{4} + \frac{\alpha}{4} < 1, \tag{32}$$

with probability of at least $1 - D_1 d\exp(-D_2 n^{1-2a}) - D_3\eta^{-2}$.

For sign consistency, it is sufficient to show that $\|\widehat{\theta}_{D_j} - \theta^*_{D_j}\|_\infty \le \frac{\|\theta^*_{D_j}\|_{\min}}{2}$. By Corollary 29, we have $\|\widehat{\theta}_{D_j} - \theta^*_{D_j}\|_\infty \le \|\widehat{\theta}_{D_j} - \theta^*_{D_j}\|_2 \le \frac{5}{\lambda_{\min}}\sqrt{d}\ \lambda_n^D \le \frac{\|\theta^*_{D_j}\|_{\min}}{2}$ as long as $\|\theta^*_{D_j}\|_{\min} \ge \frac{10}{\lambda_{\min}}\sqrt{d}\ \lambda_n^D$.

Lastly, Lemma 7(a) guarantees that ℓ_1-penalized likelihood regression recovers the parent set for each node with high probability. Because we have p regression problems if $n \ge D'\max\{(d\log^2\eta)^{\frac{1}{a-2\kappa_2}}, (d\log^3\eta)^{\frac{1}{a-\kappa_2}}\}$, the full DAG model is recovered with high probability:

$$P(\widehat{G} = G) \ge 1 - D_1 d\cdot p\cdot\exp(-D_2 n^{1-2a}) - D_3\eta^{-1}.$$

■

References

Constantin F Aliferis, Ioannis Tsamardinos, and Alexander Statnikov. HITON: a novel Markov Blanket algorithm for optimal variable selection. In *AMIA Annual Symposium Proceedings*, volume 2003, page 21. American Medical Informatics Association, 2003.

David Maxwell Chickering. Learning Bayesian networks is NP-complete. In *Learning from data*, pages 121–130. Springer, 1996.

David Maxwell Chickering. Optimal structure identification with greedy search. *The Journal of Machine Learning Research*, 3:507–554, 2003.

Robert G. Cowell, Phillip A. Dawid, Steffen L. Lauritzen, and David J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer-Verlag, 1999.

Charmaine B Dean. Testing for overdispersion in Poisson and binomial regression models. *Journal of the American Statistical Association*, 87(418):451–457, 1992.

Kenji Doya. *Bayesian brain: Probabilistic approaches to neural coding*. MIT press, 2007.

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. glmnet: Lasso and elastic-net regularized generalized linear models. *R package version*, 1, 2009.

Nir Friedman, Iftach Nachman, and Dana Peér. Learning bayesian network structure from massive datasets: the sparse candidate algorithm. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 206–215. Morgan Kaufmann Publishers Inc., 1999.

Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. Using bayesian networks to analyze expression data. *Journal of computational biology*, 7(3-4):601–620, 2000.

David Heckerman, Dan Geiger, and David M Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

Ali Jalali, Pradeep D Ravikumar, Vishvas Vasuki, and Sujay Sanghavi. On learning discrete graphical models using group-sparse regularization. In *International Conference on Artificial Intelligence and Statistics*, pages 378–387, 2011.

Dominik Janzing and Bernhard Scholkopf. Causal inference using the algorithmic markov condition. *IEEE Trans. on Infor. Theory*, 56(10):5168–5194, 2010.

Jeffrey O Kephart and Steve R White. Directed-graph epidemiological models of computer viruses. In *Research in Security and Privacy, 1991. Proceedings., 1991 IEEE Computer Society Symposium on*, pages 343–359. IEEE, 1991.

Steffen L Lauritzen. *Graphical models*. Oxford University Press, 1996.

Po-Ling Loh and Peter Bühlmann. High-dimensional learning of linear causal networks via inverse covariance estimation. *The Journal of Machine Learning Research*, 15(1): 3065–3105, 2014.

Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, pages 1436–1462, 2006.

Carl N Morris. Natural exponential families with quadratic variance functions. *The Annals of Statistics*, pages 65–80, 1982.

Gunwoong Park and Garvesh Raskutti. Learning large-scale poisson dag models based on overdispersion scoring. In *Advances in Neural Information Processing Systems*, pages 631–639, 2015.

Jonas Peters and Peter Bühlmann. Identifiability of Gaussian structural equation models with equal error variances. *Biometrika*, page ast043, 2013.

Jonas Peters, Joris Mooij, Dominik Janzing, et al. Identifiability of causal graphs using functional models. *arXiv preprint arXiv:1202.3757*, 2012.

Garvesh Raskutti and Caroline Uhler. Learning directed acyclic graphs based on sparsest permutations. *arXiv preprint arXiv:1307.0366*, 2013.

Pradeep Ravikumar, Martin J Wainwright, John D Lafferty, et al. High-dimensional ising model selection using 1-regularized logistic regression. *The Annals of Statistics*, 38(3): 1287–1319, 2010.

Pradeep Ravikumar, Martin J Wainwright, Garvesh Raskutti, Bin Yu, et al. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, and Antti Kerminen. A linear non-Gaussian acyclic model for causal discovery. *The Journal of Machine Learning Research*, 7:2003–2030, 2006.

Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2000.

Ioannis Tsamardinos and Constantin F Aliferis. Towards principled feature selection: Relevancy, filters and wrappers. In *Proceedings of the ninth international workshop on Artificial Intelligence and Statistics*. Morgan Kaufmann Publishers: Key West, FL, USA, 2003.

S. van de Geer and P. Bühlmann. Penalized maximum likelihood estimation for sparse directed acyclic graphs. *Annals of Statistics*, 41:536–567, 2013.

Martin J Wainwright, John D Lafferty, and Pradeep K Ravikumar. High-dimensional graphical model selection using ℓ_1-regularized logistic regression. In *Advances in neural information processing systems*, pages 1465–1472, 2006.

Eunho Yang, Genevera Allen, Zhandong Liu, and Pradeep K Ravikumar. Graphical models via generalized linear models. In *Advances in Neural Information Processing Systems*, pages 1358–1366, 2012.

Eunho Yang, Pradeep Ravikumar, Genevera I Allen, and Zhandong Liu. On graphical models via univariate exponential family distributions. *arXiv preprint arXiv:1301.4183*, 2013.

Tian Zheng, Matthew J Salganik, and Andrew Gelman. How many people do you know in prison? Using overdispersion in count data to estimate social structure in networks. *Journal of the American Statistical Association*, 101(474):409–423, 2006.

Journal of Machine Learning Research 18 (2018) 1-49 Submitted 5/17; Revised 2/18; Published 4/18

Improved spectral community detection in large heterogeneous networks

Hafiz TIOMOKO ALI HAFIZ.TIOMOKOALI@CENTRALESUPELEC.FR
CentraleSupélec
Université Paris Saclay
Laboratoire des Signaux et Systèmes
3 rue Joliot Curie, 91192 Gif-Sur-Yvette

Romain COUILLET ROMAIN.COUILLET@CENTRALESUPELEC.FR
CentraleSupélec
Université Paris Saclay
Laboratoire des Signaux et Systèmes
3 rue Joliot Curie, 91192 Gif-Sur-Yvette

Editor: Ulrike von Luxburg

Abstract

In this article, we propose and study the performance of spectral community detection for a family of "α-normalized" adjacency matrices $\mathbf{A}$, of the type $\mathbf{D}^{-\alpha}\mathbf{A}\mathbf{D}^{-\alpha}$ with $\mathbf{D}$ the degree matrix, in heterogeneous dense graph models. We show that the previously used normalization methods based on $\mathbf{A}$ or $\mathbf{D}^{-1}\mathbf{A}\mathbf{D}^{-1}$ are in general suboptimal in terms of correct recovery rates and, relying on advanced random matrix methods, we prove instead the existence of an optimal value α_{opt} of the parameter α in our generic model; we further provide an online estimation of α_{opt} only based on the node degrees in the graph. Numerical simulations show that the proposed method outperforms state-of-the-art spectral approaches on moderately dense to dense heterogeneous graphs.

Keywords: community detection, random networks, heterogeneous graphs, random matrix theory, spectral clustering.

1. Introduction and Motivations

The advent of the big data era is creating an unprecedented need for automating large network analysis. Community detection is among the most important tasks in automated network mining (Fortunato, 2010). Given a network graph, detecting communities consists in retrieving hidden clusters of nodes based on some similarity metric (the edges are dense inside communities and sparse across communities). While quite simple to define, community detection is usually not an easy task and many methods arising from different fields have been proposed to carry it out. The most important of them are statistical inference, modularity maximization and graph partitioning methods. Statistical inference methods consist in fitting the observed network to a structured network model and infer its parameters (among which the assignment of the nodes to the communities) (Hastings, 2006; Newman and Leicht, 2007). Modularity maximization algorithms rely instead on the modularity metric which quantifies the subdivision of networks into communities (Fortunato, 2010).[1] However, retrieving the modularity maximizing graph partition is generally an NP-hard problem and many polynomial-time approximation methods have been proposed: greedy methods (Newman, 2004), simulated annealing (Guimera et al., 2004), extremal optimization (Duch and Arenas, 2005) and spectral methods (Newman, 2006b). Spectral algorithms consist in retrieving the communities from the eigenvectors associated with the dominant eigenvalues of some matrix representation of the graph structure (adjacency matrix, modularity matrix, Laplacian matrix). By relaxing the modularity optimization problem from binary values of the community memberships to continuous scores, it is shown that approximate modularity maximization and even statistical inference methods can be performed via a low dimensional clustering of the entries of the dominant eigenvectors of the representation matrix (Ng et al., 2002; Newman, 2016) in polynomial time. Precisely, the steps of spectral methods for community detection are described in the following algorithm.

Algorithm 1: Spectral algorithm

1: Compute the, say, ℓ eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_\ell \in \mathbb{R}^n$ corresponding to the dominant (largest or smallest) eigenvalues of one of the matrix representations of the network (adjacency, modularity, Laplacian) of size $n \times n$.
2: Stack the vectors $\mathbf{u}_i$'s columnwise in a matrix $\mathbf{W} = [\mathbf{u}_i, \ldots, \mathbf{u}_\ell] \in \mathbb{R}^{n \times \ell}$.
3: Let $\mathbf{r}_1, \ldots, \mathbf{r}_n \in \mathbb{R}^\ell$ be the rows of $\mathbf{W}$. Cluster $\mathbf{r}_i \in \mathbb{R}^\ell$, $1 \leq i \leq n$ in one of K groups using any low-dimensional classification algorithm (e.g., k-means (Hartigan and Wong, 1979) or Expectation Maximization (EM) (Ng et al., 2012)). The label assigned to $\mathbf{r}_i$ then corresponds to the label of node i.

Most of the works proposing statistical analysis of the performance of community detection (for dense as well as sparse networks) consider the basic Stochastic Block Model (SBM) as a model for networks decomposable into communities. Denoting $\mathcal{G}$ a K-class graph of n vertices with communities $\mathcal{C}_1, \ldots, \mathcal{C}_K$ with g_i the group assignment of node i, the SBM assumes an adjacency matrix $\mathbf{A} \in \{0,1\}^{n \times n}$ with A_{ij} independent Bernoulli random variables with parameter $P_{g_i g_j}$ where P_{ab} represents the probability that any node of class $\mathcal{C}_a$ is connected to any node of class $\mathcal{C}_b$. The main limitation of this model is that it is only suited to homogeneous graphs where all nodes have the same average degree in each community (besides, class sizes are often taken equal). As suggested in the practical case of the popular Political Blogs graph (Adamic and Glance, 2005) (see Figure 4), a more realistic model, the Degree-Corrected SBM (DCSBM), was proposed in (Coja-Oghlan and Lanka, 2009; Karrer and Newman, 2011) to account for degree heterogeneity inside communities. For the same graph $\mathcal{G}$ defined above, by letting q_i, $1 \leq i \leq n$, be some intrinsic weights which affect the probability for node i to connect to any other network node, the adjacency matrix $\mathbf{A} \in \{0,1\}^{n \times n}$ of the graph generated by the DCSBM is such that A_{ij} are independent Bernoulli random variables with parameter $q_i q_j C_{g_i g_j}$, where $C_{g_i g_j}$ is a class-wise correction factor.

Community detection in DCSBMs has recently been studied, providing "consistent"[2] algorithms ranging from modularity/likelihood based approaches to spectral clustering methods.

1. Precisely, the modularity is defined as the difference between the total number of edges inside the communities for a given partition and the total number of edges if the partition were created randomly in the graph.
2. Consistency is mainly defined in two forms. Informally, a community detection algorithm is **weakly** consistent whenever the fraction of misclassified nodes vanishes asymptotically with high probability and

JMLR 18(225):1-49, 2018

Sufficient conditions under which likelihood based approaches (Karrer and Newman, 2011) and modularity optimization methods (Newman, 2006b) are weakly and strongly consistent, have been provided in (Zhao et al., 2012). The so-called CMM (Convexified Modularity Maximization) algorithm was proposed in (Chen et al., 2015) to cope with the computational expensiveness of modularity/likelihood methods (Karrer and Newman, 2011; Newman, 2006b) by solving a convex programming relaxation of the modularity optimization. Asymptotic minimax risks for misclassification loss under the DCSBM have been established in (Gao et al., 2016). There a consistent algorithm achieving the minimax optimal rates was derived, which is similar to spectral methods but proceeds without the explicit computation of eigenvectors and is hence computationally less expensive. As far as spectral clustering methods are concerned, (Lyzinski et al., 2014) and (Lei et al., 2015) show consistency of the classical spectral clustering procedure for community detection applied to the adjacency matrix of moderately sparse DCSBM (for not too irregular degree distributions) where the expected degree is as small as $\log n$. Later, its has been shown (Coja-Oghlan and Lanka, 2009; Qin and Rohe, 2013; Jin et al., 2015; Gulikers et al., 2015) that when the degrees are highly heterogeneous, the classical spectral methods fail to detect the genuine communities. To illustrate those limitations of spectral methods under the DCSBM, the two graphs of Figure 1 provide 2D representations of dominant eigenvector 1 versus eigenvector 2 for the standard modularity matrix and the Bethe Hessian matrix[3], when three quarters of the nodes connect with low weight $q_{(1)}$ and one quarter of the nodes with high weight $q_{(2)}$. For both methods, it is clear that k-means or EM alike would erroneously induce the detection of extra communities and even a confusion of genuine communities in the Bethe Hessian approach. Those extra communities are produced by some biases created by the intrinsic weights q_i's; intuitively, nodes sharing the same intrinsic connection weights tend to create their own sub-cluster inside each community, thereby forming additionnal sub-communities inside the genuine communities. To overcome this issue, a number of regularized spectral clustering techniques have been proposed to normalize either the adjacency matrix or the leading eigenvectors by the degrees. In (Coja-Oghlan and Lanka, 2009; Gulikers et al., 2015), the authors have proposed to cluster the nodes based on the eigenvectors of a normalized adjacency matrix $\mathbf{D}^{-1}\mathbf{A}\mathbf{D}^{-1}$ with $\mathbf{D}$ the diagonal matrix containing the observed degrees on the main diagonal. The SCORE algorithm devised in (Jin et al., 2015) consists instead in using the leading eigenvectors of the adjacency matrix (pre-normalized by the dominant eigenvector which, as shown subsequently, is equivalent to normalizing by the inverse degree matrix $\mathbf{D}^{-1}$) and (Qin and Rohe, 2013) proposed to use the eigenvectors of the Laplacian matrix $\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$.

As previously stated, the aforementioned works have shown that under some regularity (or regularization) conditions, an almost perfect or perfect reconstruction of the nodes labels can be achieved asymptotically. Our motivation in this article is to go beyond mere consistency results by understanding the performances of the different regularized spectral clustering algorithms for large but finite network sizes n. To this end, we place ourselves in a regime where communities are too close to induce perfect reconstructions. In order to encompass most aforementioned methods, we study here a generalized regularization of the adjacency

a community detection algorithm is **strongly** consistent whenever the labels estimated by the algorithm match exactly the true labelling asymptotically with high probability.

3. The Bethe Hessian (BH) spectral method (Saade et al., 2014) is based on the union of the eigenvectors associated to the negative eigenvalues of $H(r_c)$ and $H(-r_c)$ respectively where $H(r) = (r^2-1)\mathbf{I}_n - r\mathbf{A} + \mathbf{D}$ for $r_c = \frac{\sum_i d_i^2}{\sum_i d_i} - 1$ with d_i the degree of node i ($\mathbf{D}$ and d_i are defined subsequently).

matrix[4] given, for any $\alpha \in \mathbb{R}$, by

$$\mathbf{L}_\alpha = (2m)^\alpha \frac{1}{\sqrt{n}} \mathbf{D}^{-\alpha} \left[\mathbf{A} - \frac{\mathbf{d}\mathbf{d}^\mathsf{T}}{2m} \right] \mathbf{D}^{-\alpha}$$

where $\mathbf{d}$ is the vector of degrees ($d_i = \sum_{j=1}^n A_{ij}$), $\mathbf{D}$ is the diagonal matrix of degrees (containing $\mathbf{d}$ on the main diagonal) and $m = \frac{1}{2}\mathbf{d}^\mathsf{T}\mathbf{1}_n$ is the number of edges in the network. In particular, $\mathbf{L}_0$ is the modularity matrix (Newman, 2006b; Jin et al., 2015), $\mathbf{L}_{\frac{1}{2}}$ is a modularity equivalent to the normalized Laplacian matrix (Qin and Rohe, 2013; Chung, 1997) and $\mathbf{L}_1$ is the form used in (Coja-Oghlan and Lanka, 2009; Gulikers et al., 2015; Tiomoko Ali and Couillet, 2016).

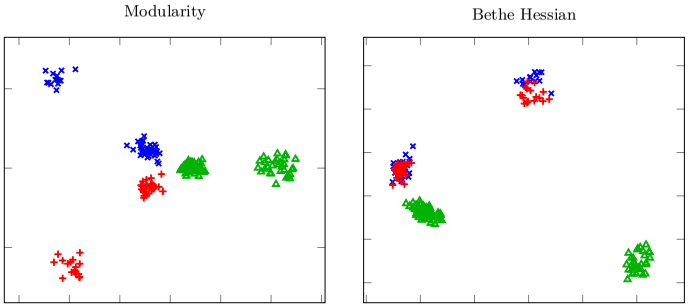


Figure 1: Two dominant eigenvectors (x-y axes) for $n = 2000$, $K = 3$ classes $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$ of sizes $|\mathcal{C}_1| = |\mathcal{C}_2| = \frac{n}{4}$, $|\mathcal{C}_3| = \frac{n}{2}$, $\frac{3}{4}$ of the nodes having $q_i = 0.1$ and $\frac{1}{4}$ of the nodes having $q_i = 0.5$, matrix of weights $\mathbf{C} = \mathbf{1}_3\mathbf{1}_3^\mathsf{T} + \frac{100}{\sqrt{n}}\mathbf{I}_3$. Colors and shapes correspond to ground truth classes.

Besides, while we believe (up to more involved mathematical treatment) that our results essentially hold true in moderately sparse graphs (of average degree of order $\Omega(\log n)$), we focus here on a *dense* DCSBM model where $q_i = \Omega(1)$. In this regime, when the correction factors $C_{g_i g_j}$ differ by a rate greater than $\mathcal{O}(n^{-\frac{1}{2}})$, weak consistency is shown for all regularized spectral algorithms. Instead, we induce a regime where clusters remain at an asymptotically "constant" distance. This is ensured by letting $C_{g_i g_j} = \Omega(1)$ individually but with the $C_{g_i g_j}$'s differing by $\mathcal{O}(n^{-\frac{1}{2}})$. Under this regime, we are able to fully study the dominant eigenvalues and associated eigenvectors (used for classification) of $\mathbf{L}_\alpha$ for large dimensional dense graphs following the DCSBM, thus allowing one to assess the performances for very large but finite-size graphs.

In a nutshell, our main findings are as follows.

4. The leading term $\frac{\mathbf{d}\mathbf{d}^\mathsf{T}}{2m}$ (not providing any information about the communities) is shown in simulations to have asymptotically no impact on the clustering performance. It is discarded here mostly for mathematical simplicity. Note in passing that $\mathbf{A} - \frac{\mathbf{d}\mathbf{d}^\mathsf{T}}{2m}$ corresponds to the so-called modularity matrix (Newman, 2006a), therefore $\mathbf{L}_\alpha$ may be seen as a "α-normalized" modularity matrix.

- We prove the existence of and obtain an expression for an optimal value α_{opt} of α for which the community detectability threshold[5] is maximally achievable.This value needs not be either 0 or 1 and its proper choice is of utmost importance in highly heterogeneous graphs.
- We provide a consistent estimator $\hat{\alpha}_{\text{opt}}$ of α_{opt} based on $\mathbf{d}$ alone.
- We show that to achieve consistent clustering in the DCSBM model, the dominant eigenvectors used for clustering should be pre-multiplied by $\mathbf{D}^{\alpha-1}$ prior to the low dimensional classification (step 3 of Algorithm 1), thereby recovering the SCORE algorithm (Jin et al., 2015) for $\alpha = 0$ and the algorithm in (Gulikers et al., 2015) for $\alpha = 1$, as special cases.
- Our proposed method is summarized under the form of Algorithm 2. As the estimation of α_{opt} is essentially linear with n, Algorithm 2 does not impair the computational cost of the underlying spectral method. A Python implementation of the algorithm is available in (Tiomoko Ali and Couillet, 2017).
- A deeper study of the regularized eigenvectors allows us to improve the initial setting of the EM algorithm (in the step 3 of the spectral algorithm described above) in comparison with a random setting.
- Numerical simulations (throughout the article) show that our methods outperform state-of-the-art spectral methods both on synthetic graphs and on real world networks.

All proofs are deferred to a separate section while sketches are provided for the main results of the article.

Notations: Vectors (matrices) are denoted by lowercase (uppercase) boldface letters. $\{\mathbf{v}_a\}_{a=1}^n$ is the column vector $\mathbf{v}$ with (scalar or vector) entries v_a and $\{\mathbf{V}_{ab}\}_{a,b=1}^n$ is the matrix $\mathbf{V}$ with (scalar or matrix) entries V_{ab}. For a vector $\mathbf{v}$, the operator $\mathcal{D}(\mathbf{v}) = \mathcal{D}\left(\{v_a\}_{a=1}^n\right)$ is the diagonal matrix having the scalars v_a down its diagonal and for a matrix $\mathbf{V}$, $\mathcal{D}(\mathbf{V})$ is the vector containing the diagonal entries of $\mathbf{V}$. The vector $\mathbf{1}_n \in \mathbb{R}^n$ stands for the column vector filled with ones. The Dirac measure at x is δ_x. The vector $\mathbf{j}_a$ is the canonical vector of class $\mathcal{C}_a$ defined by $(\mathbf{j}_a)_i = \delta_{i\in\mathcal{C}_a}$ and $\mathbf{J} = [\mathbf{j}_1, \ldots, \mathbf{j}_K] \in \{0,1\}^{n\times K}$. The set $\mathbb{C}^+$ is $\{z \in \mathbb{C},\ \Im[z] > 0\}$. We denote $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ to indicate that $\mathbf{x}$ is a Gaussian distributed random vector with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.

2. Preliminaries

This section describes the network model under study, which is based on the DCSBM defined in the previous section, and provides preliminary technical results.

Consider an n-node random graph with K classes $\mathcal{C}_1, \ldots, \mathcal{C}_K$ of sizes $|\mathcal{C}_k| = n_k$. Each node is characterized by an intrinsic connexion weight q_i which affects the probability that this node gets attached to another node in the graph. A null model would consider that the existence of an edge between i and j has probability $q_i q_j$. In order to take into account the

5. The community detectability threshold is the point beyond which there exists a clustering algorithm which can do better than a random guess.

membership of the nodes to some group, we define $\mathbf{C} \in \mathbb{R}^{K\times K}$ as a matrix of class weights C_{ab}, independent of the q_i's, affecting the connection probability between nodes in $\mathcal{C}_a$ and nodes in $\mathcal{C}_b$. Following (Karrer and Newman, 2011), the adjacency matrix $\mathbf{A}$ of the graph generated from a DCSBM model has independent entries (up to symmetry) which are Bernoulli random variables with parameter $P_{ij} = q_i q_j C_{g_i g_j} \in (0,1)$ where g_i is the group assignment of node i. We set $A_{ii} = 0$ for all i. For convenience of exposition and without loss of generality, we assume that node indices are sorted by clusters, i.e nodes 1 to n_1 constitutes $\mathcal{C}_1$, nodes n_1+1 to n_1+n_2 form $\mathcal{C}_2$, and so on.

The matrix under study is given by

$$\mathbf{L}_\alpha = (2m)^\alpha \frac{1}{\sqrt{n}} \mathbf{D}^{-\alpha} \left[\mathbf{A} - \frac{\mathbf{d}\mathbf{d}^\mathsf{T}}{2m}\right] \mathbf{D}^{-\alpha} \tag{1}$$

where $\mathbf{d} = \mathbf{A}\mathbf{1}_n$, $\mathbf{D} = \mathcal{D}(\mathbf{d})$ and $m = \frac{1}{2}\mathbf{d}^\mathsf{T}\mathbf{1}_n$.

We are mainly interested in a dense network regime where clustering is not asymptotically trivial. This regime is ensured by the following growth rate conditions.

Assumption 1 *As $n \to \infty$, K remains fixed and, for all $i, j \in \{1, \ldots, n\}$*

1. *$C_{g_i g_j} = 1 + \frac{M_{g_i g_j}}{\sqrt{n}}$, where $M_{g_i g_j} = \Omega(1)$; we shall denote $\mathbf{M} = \{M_{ab}\}_{a,b=1}^K$.*
2. *q_i are i.i.d. random variables with measure μ having compact support in $(0,1)$.*
3. *$\frac{n_i}{n} \to c_i > 0$ and we will denote $\mathbf{c} = \{c_k\}_{k=1}^K$.*

The goal of the article is to study deeply the eigenstructure of $\mathbf{L}_\alpha$ in order to understand the different mechanisms into play when performing spectral clustering on $\mathbf{L}_\alpha$. As can be observed, $\mathbf{L}_\alpha$ has non independent entries as $\mathbf{D}$ (and $\mathbf{d}$) depend on $\mathbf{A}$, and it thus does not follow a standard random matrix model. Our strategy is to approximate $\mathbf{L}_\alpha$ by a more tractable random matrix $\tilde{\mathbf{L}}_\alpha$ which asymptotically preserves the eigenvalue distribution and isolated eigenvectors of $\mathbf{L}_\alpha$. We obtain the corresponding approximate of $\mathbf{L}_\alpha$ as follows.

Theorem 2 *Let Assumption 1 hold and let $\mathbf{L}_\alpha$ be given by (1). Then, for $\mathbf{D}_q \triangleq \mathcal{D}(\mathbf{q})$, as $n \to \infty$, $\|\mathbf{L}_\alpha - \tilde{\mathbf{L}}_\alpha\| \to 0$ in operator norm, almost surely, where*

$$\begin{aligned}
\tilde{\mathbf{L}}_\alpha &= \frac{1}{\sqrt{n}} \mathbf{D}_q^{-\alpha} \mathbf{X} \mathbf{D}_q^{-\alpha} + \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\mathsf{T}, \\
\mathbf{U} &= \begin{bmatrix} \frac{\mathbf{D}_q^{1-\alpha}\mathbf{J}}{\sqrt{n}} & \frac{\mathbf{D}_q^{-\alpha}\mathbf{X}\mathbf{1}_n}{\mathbf{q}^\mathsf{T}\mathbf{1}_n} \end{bmatrix}, \\
\boldsymbol{\Lambda} &= \begin{bmatrix} \left(\mathbf{I}_K - \mathbf{1}_K \mathbf{c}^T\right) \mathbf{M} \left(\mathbf{I}_K - \mathbf{c}\mathbf{1}_K^T\right) & -\mathbf{1}_K \\ -\mathbf{1}_K^T & 0 \end{bmatrix},
\end{aligned}$$

with $\mathbf{X} = \{X_{ij}\}_{i,j=1}^n$ symmetric with independent entries (up to the symmetry), X_{ij} having zero mean and variance $q_i q_j (1 - q_i q_j)$, and $\mathbf{J} = [\mathbf{j}_1, \ldots, \mathbf{j}_K] \in \{0,1\}^{n\times K}$ with $(\mathbf{j}_a)_i = \delta_{\{g_i = a\}}$.

Proof [Sketch] The proof relies on the fact that we may write $A_{ij} = q_i q_j + q_i q_j \frac{M_{g_i g_j}}{\sqrt{n}} + X_{ij}$ where X_{ij} is a zero mean random variable with variance $q_i q_j (1 - q_i q_j) + \Theta(n^{-\frac{1}{2}})$, since A_{ij} is a Bernoulli random variable with parameter $q_i q_j (1 + \frac{M_{g_i g_j}}{\sqrt{n}})$. From there, the terms: $\mathbf{d} = \mathbf{A}\mathbf{1}_n$,

$\mathbf{d}^\mathsf{T}\mathbf{1}_n$, $\mathbf{d}\mathbf{d}^\mathsf{T}$ and $\mathbf{D} = \mathcal{D}(\mathbf{d})$ composing $\mathbf{L}_\alpha$ can be evaluated. Notably, $\mathbf{D}$ and $\mathbf{d}^\mathsf{T}\mathbf{1}_n$ can be decomposed as the sum of dominant terms (with higher spectral norms with respect to n) and trailing terms (vanishing spectral norms with respect to n), so that we can write a Taylor expansion of $\mathbf{D}^{-\alpha}$ and $(\mathbf{d}^\mathsf{T}\mathbf{1}_n)^\alpha$ for $\alpha \in \mathbb{R}$. By computing $\mathbf{L}_\alpha$ using the asymptotic approximations of $\mathbf{D}^{-\alpha}$, $(\mathbf{d}^\mathsf{T}\mathbf{1}_n)^\alpha$, $\mathbf{A}$, $\mathbf{d}\mathbf{d}^\mathsf{T}$, we obtain $\tilde{\mathbf{L}}_\alpha$. The complete proof is provided in Section 6.1. ■

This result immediately implies the following Corollary.

Corollary 3 *Under Assumption 1, let $\lambda_i(\mathbf{L}_\alpha)$ (resp., $\lambda_i(\tilde{\mathbf{L}}_\alpha)$) be the eigenvalues of $\mathbf{L}_\alpha$ (resp., $\tilde{\mathbf{L}}_\alpha$) with associated eigenvectors $\mathbf{u}_i(\mathbf{L}_\alpha)$ (resp., $\mathbf{u}_i(\tilde{\mathbf{L}}_\alpha)$). We have*

$$\max_{1\le i\le n} \left|\lambda_i(\mathbf{L}_\alpha) - \lambda_i(\tilde{\mathbf{L}}_\alpha)\right| \xrightarrow{\text{a.s.}} 0$$

and, if $\liminf_n \min_{j\neq i} |\lambda_i(\mathbf{L}_\alpha) - \lambda_j(\tilde{\mathbf{L}}_\alpha)| > 0$,

$$\left\|\mathbf{u}_i(\mathbf{L}_\alpha) - \mathbf{u}_i(\tilde{\mathbf{L}}_\alpha)\right\| \xrightarrow{\text{a.s.}} 0.$$

Thus, for large enough n, the spectral analysis of $\mathbf{L}_\alpha$ can be performed through that of $\tilde{\mathbf{L}}_\alpha$.

The matrix $\tilde{\mathbf{L}}_\alpha$ is essentially a classical random matrix model and the study of its eigenvalues and dominant eigenvectors can be performed using standard random matrix theory (RMT) approaches (Benaych-Georges and Nadakuditi, 2012; Hachem et al., 2013).

3. Main Results

3.1 Spike model and dominant eigenvector regularization

The matrix $\tilde{\mathbf{L}}_\alpha$ is an additive spiked random matrix (Baik et al., 2005) as it is the sum of the standard full rank symmetric random matrix $n^{-\frac{1}{2}}\mathbf{D}_q^{-\alpha}\mathbf{X}\mathbf{D}_q^{-\alpha}$ having independent zero mean entries and a low rank matrix $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\mathsf{T}$. As shown in Figure 2, the spectrum (eigenvalue distribution) of spiked random matrices is generally composed of (one or several) bulks of concentrated eigenvalues and, when a phase transition is met, of additionnal eigenvalues which isolate from the aforementioned bulks. The eigenvectors corresponding to the isolated eigenvalues of the spiked random matrix become more correlated to the eigenvectors of the low rank matrix when the corresponding eigenvalues are far away from the rest of the eigenvalues.

From Theorem 2, the low rank matrix $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\mathsf{T}$ contains the matrix $\mathbf{D}_q^{1-\alpha}\mathbf{J}$; so, when the phase transition is met, the eigenvectors of $\tilde{\mathbf{L}}_\alpha$ will be correlated to some extent to $\mathbf{D}_q^{1-\alpha}\mathbf{J}$ as long as the corresponding informative eigenvalues are isolated from the bulk of eigenvalues. This is well illustrated in Figure 2 where the eigenvectors associated to non-isolated eigenvalues are noisy, i.e., classes can be barely distinguished from those eigenvectors. On the other hand, the eigenvectors associated to isolated eigenvalues consist of noisy plateaus characterizing the classes and thus a consistent classification can be expected using those eigenvectors. However, for a better clustering, one expects instead the vectors used for classification to be correlated to the canonical vectors $\mathbf{j}_a$, $1 \le a \le K$, instead of $\mathbf{D}_q^{1-\alpha}\mathbf{j}_a$.

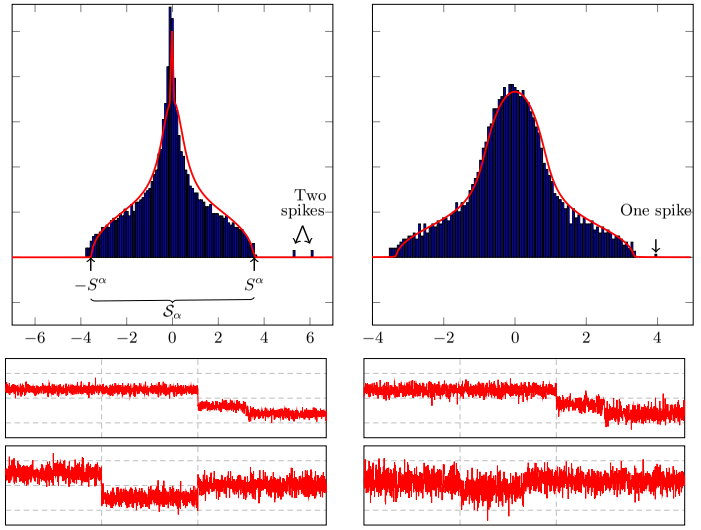


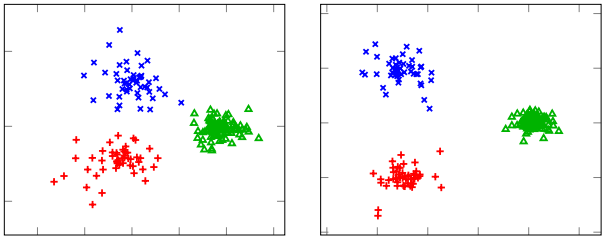
Figure 2: Two graphs generated upon the DCSBM with $K = 3$, $n = 2000$, $c_1 = 0.3, c_2 = 0.3, c_3 = 0.4$, $\mu = \frac{1}{2}\delta_{q_{(1)}} + \frac{1}{2}\delta_{q_{(2)}}$, $q_{(1)} = 0.4$, $q_{(2)} = 0.9$ and two different affinity matrices $\mathbf{M}$. **(Left)** $M_{ii} = 12$, $M_{ij} = -4, i \neq j$, **(Right)**: $M_{ii} = -3$, $M_{ij} = -10, i \neq j$, **(Top)**: Eigenvalue distribution of $\mathbf{L}_\alpha$, $\alpha = 0$. **(Bottom)**: First and second leading eigenvectors of $\mathbf{L}_\alpha$, $\alpha = 0$.

As a consequence, we claim that, letting $\mathbf{u}_1, \ldots, \mathbf{u}_l$ be the eigenvectors associated to the ℓ isolated eigenvalues of $\mathbf{L}_\alpha$, the vectors $\mathbf{v}_i = \mathbf{D}^{\alpha-1}\mathbf{u}_1$ for $1 \le i \le \ell$ should be the ones used for the classification instead of the $\mathbf{u}_i$'s.[6]

This important observation helps correcting the biases (creation of artificial classes) introduced by the degree heterogeneity observed earlier in Figure 1. As shown in Figure 3, which assumes the same setting as Figure 1, when the aforementioned eigenvector regularization is performed prior to EM or k-means classification, the genuine communities are correctly recovered.

As mentionned earlier, the eigenvectors corresponding to eigenvalues in the bulk are asymptotically of no use for clustering. It is thus important to characterize the phase transition point beyond which eigenvalues isolate from the bulk and determine which α best ensures this transition. To this end, we will first determine the support $\mathcal{S}^\alpha$ of the limiting spectral

6. As far as the eigenvectors are concerned, we may freely replace $\mathbf{D}_q$ (unknown in practice) by $\mathbf{D}$ (which can be computed from the observed graph) since, from Lemma 11 in the subsequent section 3.4, the vector of degrees $\mathbf{d}$ is, up to a scale factor β, a consistent estimator of the vector of intrinsic weights $\mathbf{q}$ and thus $\|\beta\mathbf{D} - \mathbf{D}_q\| \to 0$ almost surely.



$(\alpha = 1)$ $(\alpha = \hat{\alpha}_{\text{opt}})$

Figure 3: Two dominant eigenvectors of $\mathbf{L}_\alpha$ pre-multiplied by $\mathbf{D}^{\alpha-1}$(x-y axes) for $n = 2000$, $K = 3$, $\mu = \frac{3}{4}\delta_{q_{(1)}} + \frac{1}{4}\delta_{q_{(2)}}$, $q_{(1)} = 0.1$, $q_{(2)} = 0.5$, $c_1 = c_2 = \frac{1}{4}$, $c_3 = \frac{1}{2}$, $\mathbf{M} = 100\mathbf{I}_3$ with $\hat{\alpha}_{\text{opt}}$ defined in Section 3.4. Same setting as Figure 1.

distribution of $\mathbf{L}_\alpha$. Then, following popular spiked model tools, we will find conditions for the existence of isolated eigenvalues. This is the objective of the next sections.

3.2 Limiting support

In this section, we characterize the limiting eigenvalue distribution of $\mathbf{L}_\alpha$ where most eigenvalues concentrate. This in turn shall allow to determine the transition point beyond which informative eigenvalues isolate from the main bulk of eigenvalues and consistent clustering can thus be achieved by using the corresponding eigenvectors associated to those eigenvalues. The limiting eigenvalue distribution of $\mathbf{L}_\alpha$ is given in the following result.

Theorem 4 (Limiting spectrum) *Let $\pi_n^\alpha = \frac{1}{n}\sum_{i=1}^n \delta_{\lambda_i(\mathbf{L}_\alpha)}$ be the empirical spectral distribution (e.s.d.) of $\mathbf{L}_\alpha$. Then, as $n \to \infty$, $\pi_n^\alpha \to \bar{\pi}^\alpha$ almost surely where $\bar{\pi}^\alpha$ is a probability measure with compact symmetric support $\mathcal{S}^\alpha = [-S_\alpha, S_\alpha]$ defined, for $z \in \mathbb{C}^+ \setminus \mathcal{S}^\alpha$, by its Stieltjes transform*

$$m^\alpha(z) \equiv \int (t-z)^{-1}\, \mathrm{d}\bar{\pi}^\alpha(t) = \int \frac{1}{-z - f^\alpha(z) q^{1-2\alpha} + g^\alpha(z) q^{2-2\alpha}} \mu(dq)$$

where $(f^\alpha(z), g^\alpha(z)) \in (\mathbb{C}^+)^2$ (resp., $(\mathbb{R}^-)^2$) is the unique solution for $z \in \mathbb{C}^+$ (resp., $\mathbb{R}^+$), of

$$f^\alpha(z) = \int \frac{q^{1-2\alpha}\mu(dq)}{-z - f^\alpha(z) q^{1-2\alpha} + g^\alpha(z) q^{2-2\alpha}}$$
$$g^\alpha(z) = \int \frac{q^{2-2\alpha}\mu(dq)}{-z - f^\alpha(z) q^{1-2\alpha} + g^\alpha(z) q^{2-2\alpha}}. \tag{2}$$

Proof [Sketch] Since $\tilde{\mathbf{L}}_\alpha = \frac{1}{\sqrt{n}}\mathbf{D}_q^{-\alpha}\mathbf{X}\mathbf{D}_q^{-\alpha} + \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\mathsf{T}$ is a spiked random matrix, the e.s.d. π_n^α of $\mathbf{L}_\alpha$ converges weakly to the the e.s.d. $\tilde{\pi}_n^\alpha$ of $\frac{1}{\sqrt{n}}\mathbf{D}_q^{-\alpha}\mathbf{X}\mathbf{D}_q^{-\alpha}$ (by Weyl interlacing lemma) since $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\mathsf{T}$ is a low rank matrix. We thus find an asymptotic limit $\bar{\pi}^\alpha$ for $\tilde{\pi}_n^\alpha$ so that $\pi_n^\alpha \to \bar{\pi}^\alpha$ almost surely. To do so, we show that the Stieltjes transform of $\tilde{\pi}_n^\alpha$ converges to $m^\alpha(z)$ for

$z \in \mathbb{C}^+$, which is the Stieltjes transform of the probability measure $\bar{\pi}^\alpha$ so that the convergence also holds for the probability measures (the e.s.d.). The Stieltjes transform of the e.s.d. $\tilde{\pi}_n^\alpha$ is $n^{-1}\operatorname{tr}(\frac{1}{\sqrt{n}}\mathbf{D}_q^{-\alpha}\mathbf{X}\mathbf{D}_q^{-\alpha} - z\mathbf{I}_n)^{-1}$ (where $(\frac{1}{\sqrt{n}}\mathbf{D}_q^{-\alpha}\mathbf{X}\mathbf{D}_q^{-\alpha} - z\mathbf{I}_n)^{-1}$ is the so-called resolvent of the random matrix $\frac{1}{\sqrt{n}}\mathbf{D}_q^{-\alpha}\mathbf{X}\mathbf{D}_q^{-\alpha}$), the deterministic limit of which gives $m^\alpha(z)$, computed using classical random matrix theory (RMT) tools (Pastur et al., 2011). The calculus details are provided in Section 6.2. ■

Remark 5 (Stochastic Block Model) *Particularizing Theorem 4 to the Stochastic Block Model (SBM) (where $q_i = q_0$ for all i), the limiting probability measure $\bar{\pi}^\alpha$ is the popular semi-circle distribution with density $\bar{\pi}^\alpha(dt) = \frac{2}{\pi(S^\alpha)^2}\sqrt{\max\{(S^\alpha)^2 - t^2, 0\}}dt$ with $S^\alpha = 2q_0^{1-2\alpha}\sqrt{1-q_0^2}$. The associated Stieljes transform $m^\alpha(z)$ is explicit with in particular*

$$q_0^{1-2\alpha} m^\alpha(z q_0^{1-2\alpha}) = q_0^{\frac{1}{2}-\alpha} f^\alpha(z q_0^{\frac{1}{2}-\alpha}) = q_0^{-1} g^\alpha(z q_0^{1-2\alpha}) = -\frac{z}{2(1-q_0^2)} - \sqrt{\left(\frac{z}{2(1-q_0^2)}\right)^2 - \frac{1}{1-q_0^2}}.$$

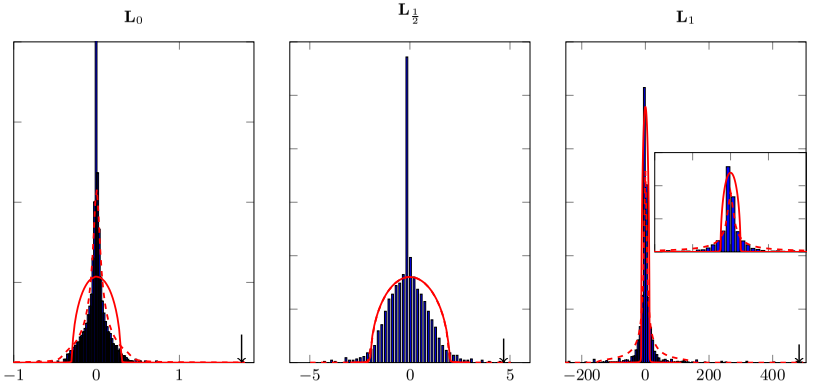


Figure 4: Political blogs (Adamic and Glance, 2005) network. Empirical versus Theoretical law of the eigenvalues of $\mathbf{L}_{\hat{\alpha}_{\text{opt}}}$ when fitting this network with the DCSBM (dashed) and the SBM (solid). Here $\hat{\alpha}_{\text{opt}} = 0$. The arrow shows the position of the largest eigenvalue.

The top of Figure 2, already discussed above, shows the density of the limiting $\bar{\pi}^\alpha$, for $\alpha = 0$, superimposed over the histogram of π_n^α. Figure 4 similarly displays the histogram π^α of the empirical eigenvalues of $\mathbf{L}_\alpha$ corresponding to the real network of Political blogs (Adamic and Glance, 2005) versus the theoretical limiting distribution $\bar{\pi}^\alpha$ obtained by fitting the network to the DCSBM (from Theorem 4, with μ the actual degree distribution of the graph) and the theoretical limiting distribution obtained by fitting the network to the SBM instead

(in solid lines).[7] We note importantly that the DCSBM is a good fit for the political blogs network except possibly for $L_{\frac{1}{2}}$ while the SBM does not fit the network in any case. This suggests that the DCSBM is a more appropriate model when studying real world networks.

3.3 Phase transition

We also observe in Figure 4 (and more obviously in the synthetic case of Figure 2) that different choices of α lead to different behaviors in the position of the dominant eigenvalues. We shall determine here when separation of one or several eigenvalues from the bulk occurs. To this end, we follow popular spiked model techniques (Benaych-Georges and Nadakuditi, 2012; Hachem et al., 2013) for phase transition characterization. This entails the following result.

Theorem 6 (Phase transition) *Let Assumption 1 hold and let $\lambda(\bar{\mathbf{M}})$ be a non zero eigenvalue with multiplicity η of $\bar{\mathbf{M}} \equiv \left(\mathcal{D}(\mathbf{c}) - \mathbf{c}\mathbf{c}^\mathsf{T}\right)\mathbf{M}$. Then, for $\alpha \in \mathbb{R}$, there exists corresponding isolated eigenvalues $\lambda_i(\mathbf{L}_\alpha), \ldots, \lambda_{i+\eta-1}(\mathbf{L}_\alpha) \in \mathbb{R}\backslash\mathcal{S}^\alpha$ of $\mathbf{L}_\alpha$ all converging to $\rho \in \mathbb{R} \setminus \mathcal{S}^\alpha$, as $n \to \infty$, almost surely, if and only if* [8]

$$\left|\lambda(\bar{\mathbf{M}})\right| > \tau^\alpha \triangleq -\lim_{x \downarrow \mathcal{S}^\alpha} \frac{1}{g^\alpha(x)},$$

with $g^\alpha(x)$ defined in Theorem 4. In this case, ρ is defined by

$$\rho = (g^\alpha)^{-1}\left(-\frac{1}{\lambda(\bar{\mathbf{M}})}\right).$$

Proof [Sketch] From Theorem 4, the e.s.d. of $\mathbf{L}_\alpha$ converges weakly to the e.s.d. of $\frac{1}{\sqrt{n}}\mathbf{D}_q^{-\alpha}\mathbf{X}\mathbf{D}_q^{-\alpha}$ with support $\mathcal{S}^\alpha$ (defined in Theorem 4) but since $\frac{1}{\sqrt{n}}\mathbf{D}_q^{-\alpha}\mathbf{X}\mathbf{D}_q^{-\alpha}$ and $\mathbf{L}_\alpha$ only differ by a finite rank matrix $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\mathsf{T}$, some eigenvalues of $\mathbf{L}_\alpha$ may isolate from the support $\mathcal{S}^\alpha$. To find those isolated eigenvalues, we solve for $\rho \notin \mathcal{S}^\alpha$, $\det(\mathbf{L}_\alpha - \rho\mathbf{I}_n) = 0$. This leads to find the ρ's for which $0 = \det(\mathbf{I}_{K+1} + \mathbf{U}^\mathsf{T}\mathbf{Q}_\rho^\alpha\mathbf{U}\mathbf{\Lambda})$ where $\mathbf{Q}_\rho^\alpha = (\frac{1}{\sqrt{n}}\mathbf{D}_q^{-\alpha}\mathbf{X}\mathbf{D}_q^{-\alpha} - \rho\mathbf{I}_n)^{-1}$ is the resolvent of $\frac{1}{\sqrt{n}}\mathbf{D}_q^{-\alpha}\mathbf{X}\mathbf{D}_q^{-\alpha}$. By using standard RMT calculus (Benaych-Georges and Nadakuditi, 2012), we obtain a deterministic approximation of $\mathbf{I}_{K+1} + \mathbf{U}^\mathsf{T}\mathbf{Q}_z^\alpha\mathbf{U}\mathbf{\Lambda}$ which leads to the phase transition condition in Theorem 6. ∎

Remark 7 (τ^α in SBM setting) *From Remark 5, in the SBM setting, τ^α no longer depends on α and is given by $\tau^\alpha = \frac{\sqrt{1-q_0^2}}{q_0}$.*

Remark 8 (Number of isolated eigenvalues) *From Theorem 6, there is a one-to-one mapping between the limiting isolated eigenvalues ρ of $\mathbf{L}_\alpha$ and non zero eigenvalues of $\bar{\mathbf{M}} = \left(\mathcal{D}(\mathbf{c}) - \mathbf{c}\mathbf{c}^\mathsf{T}\right)\mathbf{M}$. As $\mathbf{1}_K^\mathsf{T}\bar{\mathbf{M}} = 0$, $\bar{\mathbf{M}}$ has a maximum of $K-1$ non zero eigenvalues which means that at most $K-1$ eigenvalues of $\mathbf{L}_\alpha$ can be found at macroscopic distance from $\mathcal{S}^\alpha$. Thus, at most $K-1$ eigenvectors of $\mathbf{L}_\alpha$ can be used in the first step of the spectral algorithm described in the introduction.*

7. The SBM assumes here $q_i = q_0$ for all i.
8. The limit $\lim_{x \downarrow \mathcal{S}^\alpha} g^\alpha(x)$ is well defined in $(-\infty, 0]$ as $x \mapsto g^\alpha(x)$ can be shown to be a continuous growing negative function on the right side of $\mathcal{S}^\alpha$.

Remark 9 (The complete spectrum of $\mathbf{L}_\alpha$) *Strictly speaking, the aforementioned statements are somewhat inaccurate. An exhaustive analysis of $\mathbf{L}_\alpha$ indeed reveals that, under some conditions on μ, and irrespective of the clustering matrix $\mathbf{M}$, extra isolated eigenvalues can be found in the spectrum of $\mathbf{L}_\alpha$, the eigenvectors of which do not contain any structural information about the classes. This rather unfamiliar scenario has also been evidenced in the context of spectral kernel clustering in (Couillet et al., 2016). Since this hypothetical eigenvalue and eigenvector pair is of no value for the interest of clustering, it shall no longer be discussed in the following. Besides, most settings of practical interest do not present this singular behavior. A thorough discussion of this peculiarity is provided in Section 6.5.*

The value τ^α defined in Theorem 6 is a community detectability threshold which in the dense regime for the SBM case was shown to split the community detectability into two regions: a region where no algorithm can succed better than a random guess in classifying the nodes and a region where a non trivial detection is possible (Decelle et al., 2011; Nadakuditi and Newman, 2012). When the separability condition of Theorem 6 is ensured, the alignment between the properly normalized eigenvectors of $\mathbf{L}_\alpha$ and linear combinations of the class vectors $\mathbf{j}_a$'s (defined in Theorem 2) is away from zero, thus ensuring a non trivial classification performance. The larger $\lambda(\bar{\mathbf{M}})$, the closer are the vectors used for classification to the class vectors $\mathbf{j}_a$'s.

Theorem 10 *Under Assumption 1, let $\lambda(\bar{\mathbf{M}})$ and $\lambda(\mathbf{L}_\alpha)$ be an eigenvalue pair as defined in Theorem 6. We further assume $\lambda(\bar{\mathbf{M}})$ of unit multiplicity and denote $\mathbf{u}$ the eigenvector associated to the eigenvalue $\lambda(\mathbf{L}_\alpha)$. Then, letting $\bar{\mathbf{v}} = \frac{\mathbf{D}^{\alpha-1}\mathbf{u}}{\|\mathbf{D}^{\alpha-1}\mathbf{u}\|}$ and $\mathbf{\Pi} = \sum_{a=1}^K \frac{\mathbf{j}_a\mathbf{j}_a^\mathsf{T}}{n_a}$, for all $\epsilon > 0$, there exists $\gamma_-, \gamma_+ > 0$ such that, for all n large, almost surely,*

$$0 < \left|\lambda(\bar{\mathbf{M}})\right| - \tau^\alpha < \gamma_- \Rightarrow \bar{\mathbf{v}}^\mathsf{T}\mathbf{\Pi}\bar{\mathbf{v}} < \epsilon$$
$$\left|\lambda(\bar{\mathbf{M}})\right| - \tau^\alpha > \gamma_+ \Rightarrow \bar{\mathbf{v}}^\mathsf{T}\mathbf{\Pi}\bar{\mathbf{v}} > 1 - \epsilon.$$

This result is a direct corrolary of Theorem 15 which is introduced later in Section 3.5.

Figure 5 illustrates Theorem 10, which confirms that, below the phase transition threshold τ^α, there is asymptotically no correlation between the vectors $\bar{\mathbf{v}}$ and the class canonical vectors $\mathbf{j}_a$'s and thus no consistent clustering can be achieved in this regime. The theoretical curve is obtained by using the deterministic asymptotic approximation of $\bar{\mathbf{v}}^\mathsf{T}\mathbf{\Pi}\bar{\mathbf{v}}$ which is explicitely given in Section 6.4.

3.4 Optimal α

In this section, we determine the values of α for which the community detectability threshold is maximally achieved. This, in turn, is expected to allow for the optimal extraction of information about the classes from the extreme eigenvectors although this is not easily proved (and in our opinion, most likely not always true).

From Theorem 6, since $\bar{\mathbf{M}}$ does not depend on α, the smaller τ^α the more likely the detectability condition $\left|\lambda(\bar{\mathbf{M}})\right| > \tau^\alpha$ is met. We then seek α for which τ^α is minimal. For any compact set $\mathcal{A} \subset \mathbb{R}$, we may thus define

$$\alpha_{\text{opt}} \triangleq \text{argmin}_{\alpha \in \mathcal{A}}\{\tau^\alpha\}$$

which we shall assume is unique (if $q_i = q_0$ is constant, τ^α is constant across α; this case is thus excluded). The estimation of α_{opt} however requires the knowledge of $g^\alpha(x)$ for each

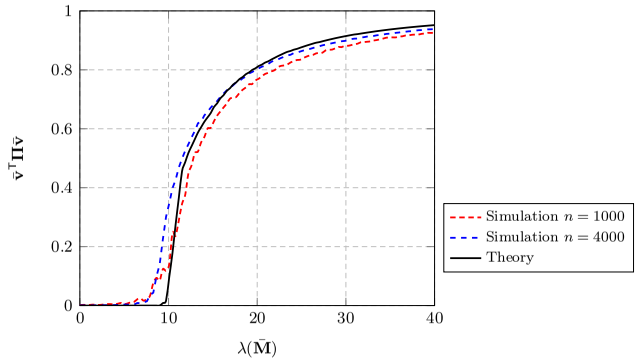


Figure 5: Simulated versus empirical $\bar{\mathbf{v}}^\mathsf{T}\mathbf{\Pi}\bar{\mathbf{v}}$ for $K = 3$, $\mu = \frac{3}{4}\delta_{q_{(1)}} + \frac{1}{4}\delta_{q_{(2)}}$, $q_{(1)} = 0.1$, $q_{(2)} = 0.2$, $c_1 = c_2 = \frac{1}{4}$, $c_3 = \frac{1}{2}$, $\mathbf{M} = \Delta\mathbf{I}_3$ with Δ ranging from 0 to 100.

$\alpha \in \mathcal{A}$. The estimation of $g^\alpha(x)$ can be done numerically by solving the fixed point equation defined in Theorem 4 provided μ is known. As a direct consequence of Assumption 1-(1), μ can in fact be estimated from the empirical graph degrees irrespective of the class matrix $\mathbf{C}$, according to the following result.

Lemma 11 *Let $\hat{q}_i = \frac{d_i}{\sqrt{\mathbf{d}^\mathsf{T}\mathbf{1}_n}}$. Then, under Assumption 1,*

$$\max_{1\le i\le n} |q_i - \hat{q}_i| \to 0 \tag{3}$$

almost surely.

We thus have all the ingredients to estimate α_{opt}.[9]

Proposition 12 *Define $\hat{\mu} \triangleq \frac{1}{n}\sum_{i=1}^n \delta_{\hat{q}_i}$ with $\hat{q}_i = \frac{d_i}{\sqrt{\mathbf{d}^\mathsf{T}\mathbf{1}_n}}$ and $\hat{\mathcal{S}}^\alpha$, $\hat{f}^\alpha(z)$, $\hat{g}^\alpha(z)$, as in Theorem 4 but for μ replaced by $\hat{\mu}$. Then, as $n \to \infty$,*

$$\hat{\alpha}_{\text{opt}} \to \alpha_{\text{opt}}$$

almost surely, where $\hat{\alpha}_{\text{opt}} \triangleq \text{argmin}_{\alpha\in\mathcal{A}}\{\hat{\tau}^\alpha\}$ with

$$\hat{\tau}_\alpha \equiv -\frac{1}{\lim_{x\downarrow\hat{S}^\alpha} \hat{g}^\alpha(x)}.$$

Remark 13 (Numerical evaluation of S^α) *Estimating $\hat{\tau}_\alpha$ requires to determine $\hat{\mathcal{S}}^\alpha$. To this end, we use the fact that $\hat{g}^\alpha(x)$ is only defined for $x \notin \hat{\mathcal{S}}^\alpha$. We thus evaluate $\hat{\mathcal{S}}^\alpha$ by an iterative dichotomic search in intervals of the type $[l, r]$ for which $\hat{g}^\alpha(l)$ is undefined (and thus the algorithm in Equation 2 does not converge) and $\hat{g}^\alpha(r)$ is defined (the algorithm converges), starting from e.g., $l = 0$ and r quite large.*

9. Note here that imposing $\mathcal{A}$ to be a compact set ensures the uniform validity of Theorem 4.

Remark 14 (Relevance of the choice of α) *Following Remarks 5 and 7, note that the choice of α is only relevant to heterogeneous graphs, as in the SBM case, the phase transition threshold τ^α is constant irrespective of α. This suggests that the more heterogeneous the graph the more important an appropriate setting of α.*

The aforementionned importance of choosing $\alpha = \hat{\alpha}_{\text{opt}}$ along with the need to pre-multiply the dominant eigenvectors of $\mathbf{L}_\alpha$ by $\mathbf{D}^{\alpha-1}$ before classification, as discussed after exposing Theorem 2, naturally bring us to an improved version of Algorithm 1 provided below. The

Algorithm 2: Improved spectral algorithm

1: Evaluate $\alpha = \hat{\alpha}_{\text{opt}} = \text{argmin}_{\alpha\in\mathcal{A}} \lim_{x\downarrow\hat{S}^\alpha} \hat{g}^\alpha(x)$ as per Proposition 12.
2: Retrieve the ℓ eigenvectors corresponding to the ℓ largest eigenvalues of $\mathbf{L}_\alpha = (2m)^\alpha \frac{1}{\sqrt{n}}\mathbf{D}^{-\alpha}\left[\mathbf{A} - \frac{\mathbf{d}\mathbf{d}^\mathsf{T}}{2m}\right]\mathbf{D}^{-\alpha}$. Denote $\mathbf{u}_1^\alpha, \ldots, \mathbf{u}_\ell^\alpha$ those eigenvectors.
3: Letting $\mathbf{v}_i^\alpha = \mathbf{D}^{\alpha-1}\mathbf{u}_i^\alpha$ and $\bar{\mathbf{v}}_i^\alpha = \frac{\mathbf{v}_i^\alpha}{\|\mathbf{v}_i^\alpha\|}$, stack the vectors $\bar{\mathbf{v}}_i^\alpha$'s columnwise in a matrix $\mathbf{W} = [\bar{\mathbf{v}}_1^\alpha, \ldots, \bar{\mathbf{v}}_\ell^\alpha] \in \mathbb{R}^{n\times\ell}$.
4: Let $\mathbf{r}_1, \ldots, \mathbf{r}_n \in \mathbb{R}^\ell$ be the rows of $\mathbf{W}$. Cluster $\mathbf{r}_i \in \mathbb{R}^\ell$, $1 \le i \le n$ in one of the K groups using any low-dimensional classification algorithm (e.g., k-means or EM). The label assigned to $\mathbf{r}_i$ then corresponds to the label of node i.

performances of Algorithm 2 mainly depend on the content of the eigenvectors $\bar{\mathbf{v}}_i^\alpha$'s. These regularized eigenvectors happen to be shapped like noisy "plateaus" (step functions), each plateau characterizing a class. The objective of the next section is to provide deterministic limits of the parameters of those noisy plateaus from which the asymptotic performances of Algorithm 2 unfold.

3.5 Eigenvectors and improvement of Expectation Maximization (EM) algorithm

In this section, we provide a precise characterization of the asymptotic class means and class covariances of the dominant eigenvectors entries (used for clustering) which in turn allows to improve the classical EM algorithm used in the last step of spectral clustering procedures. The eigenvectors of $\mathbf{L}_\alpha$ have the property of remaining "stable" in the large dimensional limit, thereby allowing for a precise characterization of their content. This behavior (classical in the spike model analysis of random matrices) however only holds for eigenvectors associated to strictly isolated eigenvalues (in the sense that the latter remain at macroscopic distance of all other eigenvalues). In the remainder, we thus assume that the normalized eigenvector $\bar{\mathbf{v}}_i^\alpha$ under study is associated with such a strictly isolated eigenvalue.

As one can see in Figure 3, the different clusters of points (rows of $\mathbf{W}$ in Algorithm 2) have different dispersions (variances) in the DCSBM model under consideration. The most appropriate algorithm to use in step 4 of Algorithm 2 is the expectation maximization (EM) method. EM considers each point $\mathbf{r}_i \in \mathbb{R}^\ell$ arising from $[\bar{\mathbf{v}}_1^\alpha, \ldots, \bar{\mathbf{v}}_\ell^\alpha]$ as a mixture of K Gaussian random vectors with means $\boldsymbol{\nu}_{EM}^a$ and covariances $\mathbf{\Sigma}_{EM}^a \in \mathbb{R}^{\ell\times\ell}$, $a \in \{1, \ldots, K\}$. Starting from initial means and covariances, they are sequentially updated until convergence. To identify $\boldsymbol{\nu}_{EM}^a$, $\mathbf{\Sigma}_{EM}^a$ and thus understand the performance of Algorithm 2, we may write $\bar{\mathbf{v}}_i^\alpha$ [10] as the

10. Recall that the graph nodes were assumed labeled by class, and thus the entries of $\bar{\mathbf{v}}_i^\alpha$ are similarly sorted by class.

"noisy plateaus" vector

$$\bar{\mathbf{v}}_i^\alpha = \sum_{a=1}^{K} \nu_i^a \frac{\mathbf{j}_a}{\sqrt{n_a}} + \sqrt{\sigma_{ii}^a}\mathbf{w}_i^a \tag{4}$$

where $\mathbf{w}_i^a \in \mathbb{R}^n$ is a random vector orthogonal to $\mathbf{j}_a$, of norm $\sqrt{n_a}$ and supported on the indices of $\mathcal{C}_a$ and

$$\nu_i^a = \frac{1}{\sqrt{n_a}} (\bar{\mathbf{v}}_i^\alpha)^\mathsf{T} \mathbf{j}_a = \frac{1}{\sqrt{n_a}} \frac{(\mathbf{u}_i^\alpha)^\mathsf{T} \mathbf{D}^{\alpha-1} \mathbf{j}_a}{\sqrt{(\mathbf{u}_i^\alpha)^\mathsf{T} \mathbf{D}^{2(\alpha-1)} \mathbf{u}_i^\alpha}} \tag{5}$$

$$\sigma_{ij}^a = \frac{(\mathbf{u}_i^\alpha)^\mathsf{T} \mathbf{D}^{\alpha-1} \mathcal{D}_a \mathbf{D}^{\alpha-1} \mathbf{u}_j^\alpha}{\sqrt{(\mathbf{u}_i^\alpha)^\mathsf{T} \mathbf{D}^{2(\alpha-1)} \mathbf{u}_i} \sqrt{(\mathbf{u}_j^\alpha)^\mathsf{T} \mathbf{D}^{2(\alpha-1)} \mathbf{u}_j^\alpha}} - \nu_i^a \nu_j^a \tag{6}$$

with $\mathcal{D}_a = \mathcal{D}(\mathbf{j}_a)$. The vector $\boldsymbol{\nu}^a = (\nu_i^a)_{i=1}^\ell \in \mathbb{R}^\ell$ and the matrix $\boldsymbol{\Sigma}^a = (\sigma_{ij}^a)_{i,j=1}^\ell \in \mathbb{R}^{\ell \times \ell}$ represent respectively the empirical means and empirical covariances of the points $\mathbf{r}_i$ (defined in Algorithm 2) belonging to class $\mathcal{C}_a$. Thus, provided that EM converges to the correct solution, $(\boldsymbol{\nu}_{EM}^a)_i$ and $(\boldsymbol{\Sigma}_{EM}^a)_{ij}$ shall converge asymptotically to the limiting values of $\nu_i^a \in \mathbb{R}$ and σ_{ij}^a respectively. Clearly, for small values of $\boldsymbol{\Sigma}^a$ compared to $\boldsymbol{\nu}^a$, clustering the vectors $\bar{\mathbf{v}}_i^\alpha$ shall lead to good performances.

We find the asymptotic limits of the class means ν_i^a and the class covariances σ_{ij}^a. The explicit expressions of those limits are provided in the proof section (Theorems 22 and 23) for readability reasons.

Theorem 15 *For ν_i^a, σ_{ij}^a defined in* (26), (27)*respectively, there exist deterministic limits $\nu_i^{a,\infty}$ and $\sigma_{ij}^{a,\infty}$ (explicitely defined in Theorems 22 and 23 in Section 6.4) such that, as $n \to \infty$, almost surely*

$$\left|(\nu_i^a)^2 - (\nu_i^{a,\infty})^2\right| \to 0$$
$$\left|\sigma_{ij}^a - \sigma_{ij}^{a,\infty}\right| \to 0.$$

Proof [Sketch] Technically, the standard tools used in spiked random matrix analysis do not allow for an immediate assessment of the quantities ν_i^a and σ_{ij}^a. As a workaround, we follow the approach used in (Couillet et al., 2016) which relies on the possibility to estimate bilinear forms of the type $\mathbf{a}^\mathsf{T}\mathbf{u}_i^\alpha(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{b}$ for given vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ and unit multiplicity eigenvectors $\mathbf{u}_i^\alpha$ of $\mathbf{L}_\alpha$ since we have from Cauchy formula, as $n \to \infty$ almost surely, (since $\lambda_i(\mathbf{L}_\alpha) \to \rho$)

$$\mathbf{a}^\mathsf{T}\mathbf{u}_i^\alpha(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{b} = -\frac{1}{2\pi i} \oint_{\Gamma_\rho} \mathbf{a}^\mathsf{T} (\mathbf{L}_\alpha - z\mathbf{I}_n)^{-1} \mathbf{b} \mathrm{d}z$$

and for a given matrix $\mathbf{D}$

$$(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}\mathbf{u}_i^\alpha = \mathrm{tr}\, \mathbf{u}_i^\alpha(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D} = -\frac{1}{2\pi i} \oint_{\Gamma_\rho} \mathrm{tr}\, (\mathbf{L}_\alpha - z\mathbf{I}_n)^{-1} \mathbf{D} \mathrm{d}z$$

where Γ_ρ is a positively oriented contour circling around the limiting eigenvalue ρ of $\lambda_i(\mathbf{L}_\alpha)$ associated to the eigenvector $\mathbf{u}_i^\alpha$ of $\mathbf{L}_\alpha$. The calculus details are provided in Section 6.4. ■

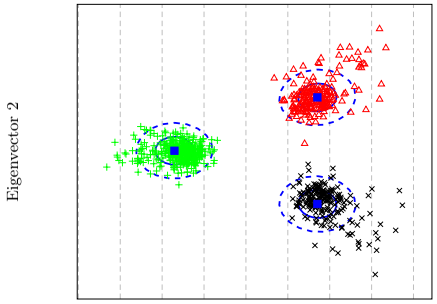


Figure 6: $n = 800$, $K = 3$ classes $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$ of sizes $|\mathcal{C}_1| = |\mathcal{C}_2| = \frac{n}{4}$, $|\mathcal{C}_3| = \frac{n}{2}$, $\frac{3}{4}$ of the nodes having $q_i = 0.3$ and the others having $q_i = 0.8$, matrix of weights $\mathbf{C} = \mathbf{1}_3\mathbf{1}_3^\mathsf{T} + \frac{30}{\sqrt{n}}\mathbf{I}_3$. Two dimensional representation of the dominant eigenvectors 1 and 2 of $\mathbf{L}_\alpha$. In blue, theoretical means and one- and two- standard deviations.

Using the asymptotic results in Theorem 15, we display in Figures 6 and 7 the theoretical means and standard deviations versus ground truths for each class-wise block of the eigenvectors entries. The good fit between the ground truths and the theoretical findings of the class means and class covariances, calls for the improvement of the random initialization of the EM procedure in the last step of spectral clustering.

The performances of EM highly depend on the chosen starting parameters; a first natural choice is to set them randomly, which as we shall see leads to poor performances especially in cases where the clusters are not easily separable. Since the theoretical limiting means $\boldsymbol{\nu}^{a,\infty}$ and covariances $\boldsymbol{\Sigma}^{a,\infty}$ are respectively the limiting values of $\boldsymbol{\nu}_{EM}^a$ and covariances $\boldsymbol{\Sigma}_{EM}^a$ provided EM converges to the correct solution, we may set as initial parameters of EM our findings $\boldsymbol{\nu}^{a,\infty}$ (Theorem 22) and $\boldsymbol{\Sigma}^{a,\infty}$ (Theorem 23) for $a \in \{1, \ldots, K\}$ provided those can be estimated. In most scenarios, the many unknowns prevent such an estimation. Nonetheless, from Corollary 25 (Section 6.4), provided the class proportions (or the sizes of each class) are (more or less) known, we can consistently estimate $\boldsymbol{\nu}^\infty$ and $\boldsymbol{\Sigma}^\infty$ in a 2-class scenario. As we shall see, this new setting of initial parameters is much better than other initializations approaches.

To show the effect of our setting of initial parameters of EM based on the findings $\boldsymbol{\nu}^\infty$ and $\boldsymbol{\Sigma}^\infty$, Figure 8 compares the empirical performances of our new spectral algorithm based on the regularized eigenvector of $\mathbf{L}_{0.5}$ for different initial settings of the EM parameters *i)* random setting (Random EM) *ii)* our theoretical setting (by assuming that the class proportions are known) and *iii)* the ground truth setting (oracle EM where we set the initial points to the empirically evaluated means and covariances of each cluster based on ground truth). Below the transition point, no consistent clustering can be achieved for large n using the eigenvectors associated to highest eigenvalues since the clusters are not separable and our theoretical limiting means and covariances are not defined since there is no isolated eigenvalues

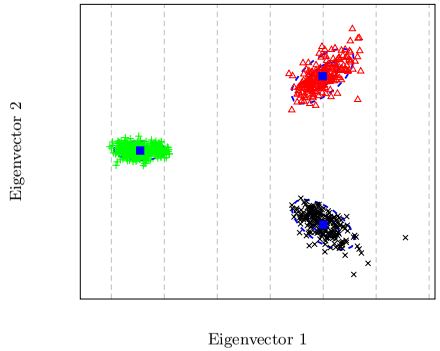


Figure 7: $n = 800$, $K = 3$ classes $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$ of sizes $|\mathcal{C}_1| = |\mathcal{C}_2| = \frac{n}{4}$, $|\mathcal{C}_3| = \frac{n}{2}$, q_i's uniformly distributed over $[0.1, 0.9]$, matrix of weights $\mathbf{C} = \mathbf{1}_3\mathbf{1}_3^\mathsf{T} + \frac{100}{\sqrt{n}}\mathbf{I}_3$. Two dimensional representation of the dominant eigenvectors 1 and 2 of $\mathbf{L}_\alpha$. In blue, theoretical means and one- and two- standard deviations

in that case. We have thus initialized EM at random in this non interesting regime (as for Random EM). The EM algorithm may in that regime set all the nodes to the same cluster, which will then result to a classification rate close to the proportion of the nodes in the cluster of largest size. In the interesting regime (after the transition point), we see that the performances (in terms of correct classification rate) of the algorithm using our theoretical setting of EM closely match the performances of an ideal setting with ground truth (oracle EM). The performances of the algorithm using a random initialization (Random EM) are completely degraded especially around critical cases (small values of Δ). Random EM becomes reliable only for very large values of Δ where clustering is somewhat trivial.

4. Numerical simulations

We restrict ourselves to $\alpha \in \mathcal{A} = [0, 1]$ for the numerical simulations. To illustrate the importance of the choice of α_{opt}, Figure 9 presents the theoretical (asymptotic) ratio between the limiting largest eigenvalue ρ of $\mathbf{L}_\alpha$ and the right edge S^α of the limiting support $\mathcal{S}^\alpha$ with respect to the amplitude of the eigenvalues of $\bar{\mathbf{M}}$. Although α_{opt} only ensures in theory to have the best isolation of the eigenvalues only in "worst cases scenarios"(i.e., when $\lambda(\bar{\mathbf{M}})$ is slighty larger than $\tau^{\alpha_{\text{opt}}}$), Figure 9 shows that taking $\alpha = \alpha_{\text{opt}}$ provides the largest gap $\frac{\rho}{S^\alpha}$ for all values of $\lambda(\bar{\mathbf{M}})$. This suggests (again, without any theoretical support) best performances with $\alpha = \alpha_{\text{opt}}$ in all cases (for any value of $\mathbf{M}$).

In the sequel, to compare the different algorithms, we will use the performance evaluation measure known as *overlap to ground truth communities*, defined in (Krzakala et al., 2013) as

$$\text{Overlap} \equiv \frac{\frac{1}{n}\sum_{i=1}^n \delta_{g_i \hat{g}_i} - \frac{1}{K}}{1 - \frac{1}{K}},$$

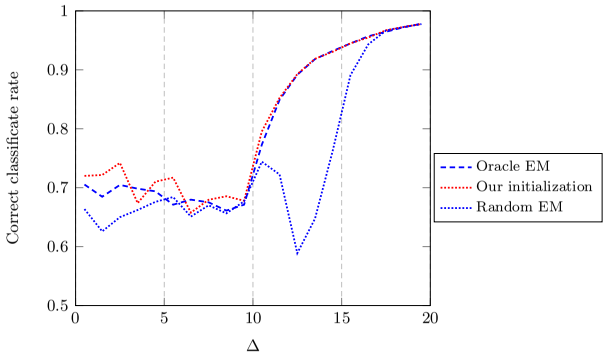


Figure 8: Probability of correct recovery for $\alpha = 0.5$, $n = 4000$, $K = 2$, $c_1 = 0.8$, $c_2 = 0.2$, $\mu = \frac{3}{4}\delta_{q_{(1)}} + \frac{1}{4}\delta_{q_{(2)}}$ with $q_{(1)} = 0.2$ and $q_{(2)} = 0.8$, $\mathbf{M} = \Delta\mathbf{I}_2$, for $\Delta \in [0, 20]$.

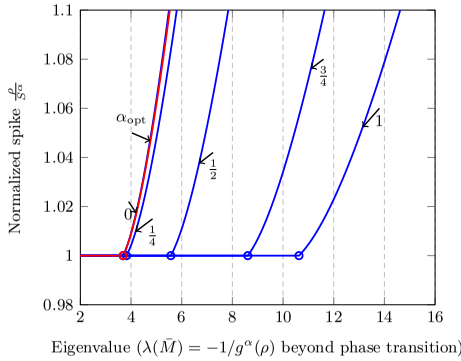


Figure 9: Ratio between the limiting largest eigenvalue ρ of $\mathbf{L}_\alpha$ and the right edge of the support $\mathcal{S}^\alpha$, as a function of the largest eigenvalue $\lambda(\bar{\mathbf{M}})$ of $\bar{\mathbf{M}}$, $\mathbf{M} = \Delta\mathbf{I}_3$, $c_i = \frac{1}{3}$, for $\Delta \in [10, 150]$, $\mu = \frac{3}{4}\delta_{q_{(1)}} + \frac{1}{4}\delta_{q_{(2)}}$ with $q_{(1)} = 0.1$ and $q_{(2)} = 0.5$, for $\alpha \in \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1, \alpha_{\text{opt}}\}$ (indicated on the curves of the graph). Here, $\alpha_{\text{opt}} = 0.07$. Circles indicate phase transition.

where g_i and $\hat{g}_i$ are the true and estimated labels of node i, respectively. Figure 10 subsequently shows the overlap performance under the setting of Figure 9 for a simulated graph of $n = 3000$ nodes. Note that the empirically observed phase transitions closely match the theoretical ones (drawn in circles and the same as in Figure 9). We then consider in Fig-

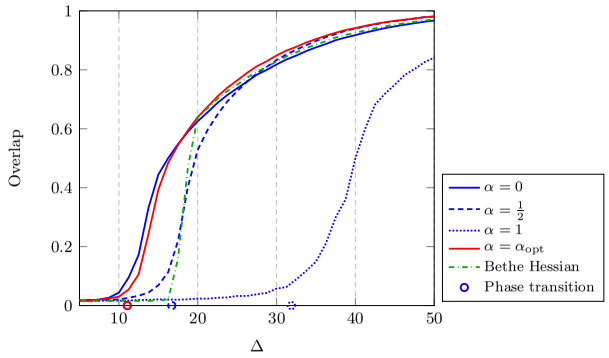


Figure 10: Overlap performance for $n = 3000$, $K = 3$, $c_i = \frac{1}{3}$, $\mu = \frac{3}{4}\delta_{q_{(1)}} + \frac{1}{4}\delta_{q_{(2)}}$ with $q_{(1)} = 0.1$ and $q_{(2)} = 0.5$, $\mathbf{M} = \Delta\mathbf{I}_3$, for $\Delta \in [5, 50]$. Here $\alpha_{\text{opt}} = 0.07$.

ure 11 a DCSBM graph where $\mathbf{M}$ is fixed and three quarters of the nodes connect with a fixed intrinsic low weight $q_{(1)} = 0.1$ and we vary the intrinsic weights $q_{(2)}$ of the remaining quarter of the nodes from low to high weights. We observe a sudden drop of the BH overlap for large $q_{(2)} - q_{(1)}$. This phenomenon is consistent with the fact, observed earlier in Figure 1, that BH creates artificial communities out of nodes with the same q_i parameter. This is a practical demonstration of the need for a proper eigenvector normalization to avoid degree biases. This observation has recently led (Newman, 2013) to consider a regularization for the non-backtracking operator on which the BH method is based, which still awaits for proper analysis.

In Figure 12, we consider a more realistic synthetic graph where the q_i's assume a power law of support $[0.05, 0.3]$ which simulates a sparse graph characteristic of real world networks. Although this is not the regime we study in this article, our method for $\alpha = \hat{\alpha}_{\text{opt}}$ still competes with the BH method which was developped for sparse homogeneous graphs. However, it is seen that the theoretical phase transitions do not closely match the empirical ones especially for the case $\alpha = 1$. This mismatch is likely due to the fact that our theoretical results in this article require $P_{ij} = \Omega(1)$ which is not always the case in this scenario.

We finally confront the performances (in terms of overlap and modularity [11]) of the different spectral algorithms on the Political blogs graph (Adamic and Glance, 2005) in Table 1. We should note that while $\alpha_{\text{opt}} = 0$ in this case, it achieves the best performance both in terms of the overlap to the ground truth and of the modularity. [12] Likely, the reason why

11. The modularity Q for a given graph partition with class labels g_i's is defined as : $Q = \frac{1}{2m}\sum_{i,j=1}^{n}\left(A_{ij} - \frac{d_i d_j}{2m}\right)\delta_{g_i=g_j}$ where $\mathbf{d} = \mathbf{A}\mathbf{1}_n$ is the degree vector and $m = \frac{1}{2}\mathbf{d}^{\mathsf{T}}\mathbf{1}_n$ is the total number of edges.

12. We should note here that the scores for the BH are different from the ones found in the article (Saade et al., 2014) since here we are running k-means algorithm in the last step of the spectral algorithm while the authors of (Saade et al., 2014) have instead used a sign classification of the eigenvector components for networks with two communities.

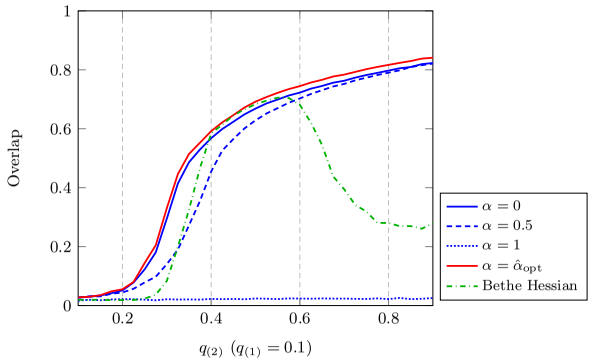


Figure 11: Overlap for $n = 3000$, $K = 3$, $\mu = \frac{3}{4}\delta_{q_{(1)}} + \frac{1}{4}\delta_{q_{(2)}}$ with $q_{(1)} = 0.1$ and $q_{(2)} \in [0.1, 0.9]$, $\mathbf{M}$ defined by $M_{ii} = 10$, $M_{ij} = -10, i \neq j$, $c_i = \frac{1}{3}$.

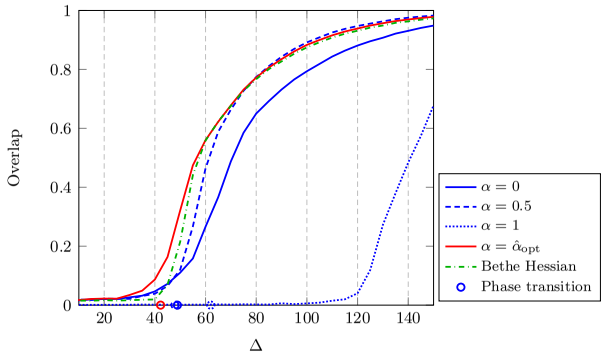


Figure 12: Overlap for $n = 3000$, $K = 3$, $c_i = \frac{1}{3}$, μ a power law with exponent 3 and support $[0.05, 0.3]$, $\mathbf{M} = \Delta\mathbf{I}_3$, for $\Delta \in [10, 150]$. Here $\hat{\alpha}_{\text{opt}} = 0.28$.

$\alpha = 0$ is optimal on the Political blogs dataset can be seen in Figure 4, where $\mathbf{L}_0$ is the similarity matrix for which the isolated eigenvalue is the farthest from the bulk of the other eigenvalues and thus the associated eigenvector is more aligned to the classes compared to the eigenvectors of $\mathbf{L}_{\frac{1}{2}}$ and $\mathbf{L}_1$.

Algo	Overlap	Modularity
$\hat{\alpha}_{\text{opt}}$ ($\simeq 0$)	**0.897**	**0.4246**
$\alpha = 0.5$	0.035	$\simeq 0$
$\alpha = 1$	0.040	$\simeq 0$
BH	0.304	0.2723

Table 1: Overlap performance and Modularity after applying the different spectral algorithms on the Political blogs graph (Adamic and Glance, 2005).

5. Concluding Remarks

The thorough study of $\mathbf{L}_\alpha$ performed in this article allows us to go further than the observation of (Gulikers et al., 2015) and (Jin et al., 2015) which state that it is important to use the eigenvectors of $\mathbf{L}_1$ or a normalization of the eigenvectors of $\mathbf{L}_0$ rather than the eigenvectors of $\mathbf{L}_0$ themselves for community detection when the network has an heterogeneous degree distribution (to avoid misclassifications induced by degree biases). Our main finding in particular is to show that there exists an optimal α, denoted here α_{opt}, for which taking the eigenvectors of $\mathbf{L}_{\alpha_{\text{opt}}}$ pre-multiplied by $\mathbf{D}^{\alpha_{\text{opt}}-1}$ ensures best performances (or to be more precise best asymptotic cluster detectability).

The results and methods in this article are all based on the strong assumption that the average node degree is of order $\mathcal{O}(n)$ and that the class-wise correction factors $C_{g_ig_j}$ differ by $\mathcal{O}(n^{-\frac{1}{2}})$ since $\forall i,j \in \{1,\dots,n\}$, $C_{g_ig_j} = 1 + \frac{M_{g_ig_j}}{\sqrt{n}}$. Previous works (Lyzinski et al., 2014; Lei et al., 2015; Gulikers et al., 2015) suggest that the present analysis, which only considers "first order spectral statistics", should naturally extend to moderately sparse graphs (of as little as $\Omega(\log n)$ average degree). Under the sparse DCSBM graph assumption, strikingly different tools are required, opening up a challenging area of improved algorithm research. Similarly, if the $C_{g_ig_j}$'s differ at a rate $n^{-\frac{1}{2}} \ll r_n \ll 1$, mere refinements of our analysis ensure asymptotic weak consistency for all values of α based on the present tools. In passing, this shows that identifiability considerations are equivalent to those delineated for any α, as in (Gulikers et al., 2015) for $\alpha = 1$. Formally, the case where $r_n = \Omega(1)$ breaks Lemma 11 and therefore the validity of our present analysis but this scenario is also by and far covered by previous works.

6. Proofs

Preliminaries

The random matrix under study $\mathbf{L}_\alpha$ is not a classically studied matrix in random matrix theory. We will thus first find in Section 6.1 an approximate tractable random matrix $\tilde{\mathbf{L}}_\alpha$ which asymptotically preserves the eigenvalue distribution and the extreme eigenvectors of $\mathbf{L}_\alpha$. In Section 6.2, we study the empirical distribution of the eigenvalues of $\mathbf{L}_\alpha$ and in Section 6.3, we characterize the exact localizations of those eigenvalues. Finally, a thorough study of the eigenvectors associated to the aforementioned eigenvalues is investigated in Sections 6.4 and 6.5.

We follow here the proof technique of (Couillet et al., 2016). In the sequel, we will make some approximations of random variables in the asymptotic regime where $n \to \infty$. For

the sake of random variables comparisons, we give the following stochastic definitions. For $x \equiv x_n$ a random variable and $u_n \geq 0$, we write $x = \mathcal{O}(u_n)$ if for any $\eta > 0$ and $D > 0$, $n^D \mathbb{P}(x \geq n^\eta u_n) \to 0$ as $n \to \infty$. For $\mathbf{v}$ a vector or a diagonal matrix with random entries, $\mathbf{v} = \mathcal{O}(u_n)$ means that the maximal entry of $\mathbf{v}$ in absolute value is $\mathcal{O}(u_n)$ in the sense defined previously. When $\mathbf{M}$ is a square matrix, $\mathbf{M} = \mathcal{O}(u_n)$ means that the operator norm of $\mathbf{M}$ is $\mathcal{O}(u_n)$. For $\mathbf{x}$ a vector or a matrix with random entries, $\mathbf{x} = o(u_n)$ means that there is $\kappa > 0$ such that $\mathbf{x} = \mathcal{O}(n^{-\kappa}u_n)$.

Most of the proofs here are classical in random matrix theory (see e.g., (Baik and Silverstein, 2006)) but require certain controls inherent to our model. The goal of the article not being an exhaustive development of the proofs techniques, we will admit a number of technical results already studied in the literature. However, we will exhaustively develop the calculus to obtain our final results which are not trivial.

6.1 Random equivalent for $\mathbf{L}_\alpha$

The matrix $\mathbf{L}_\alpha = (\mathbf{d}^\mathsf{T}\mathbf{1}_n)^\alpha \frac{1}{\sqrt{n}} \mathbf{D}^{-\alpha} \left[\mathbf{A} - \frac{\mathbf{d}\mathbf{d}^\mathsf{T}}{\mathbf{d}^\mathsf{T}\mathbf{1}_n}\right] \mathbf{D}^{-\alpha}$ has non independent entries and is not a classical random matrix model. The idea is thus to approximate $\mathbf{L}_\alpha$ by a more tractable random matrix model $\tilde{\mathbf{L}}_\alpha$ in such a way that they share asymptotically the same set of outlying eigenvalues/eigenvectors which are of interest in our clustering scenario. We recall that the entries A_{ij} of the adjacency matrix is defined from the DCSBM model as independent Bernoulli random variables with parameter $q_iq_j\left(1 + \frac{M_{g_ig_j}}{\sqrt{n}}\right)$; one may thus write

$$A_{ij} = q_iq_j + q_iq_j\frac{M_{g_ig_j}}{\sqrt{n}} + X_{ij}$$

where X_{ij}, $1 \leq i,j \leq n$, are independent (up to the symmetry) zero mean random variables of variance $q_iq_j(1-q_iq_j) + \mathcal{O}(n^{-\frac{1}{2}})$, since A_{ij} has mean $q_iq_j + q_iq_j\frac{M_{g_ig_j}}{\sqrt{n}}$ and variance $q_iq_j(1-q_iq_j) + \mathcal{O}(n^{-\frac{1}{2}})$. We can then write the normalized adjacency matrix as follows

$$\frac{1}{\sqrt{n}}\mathbf{A} = \frac{1}{\sqrt{n}}\mathbf{q}\mathbf{q}^\mathsf{T} + \frac{1}{n}\left\{\mathbf{q}_{(a)}\mathbf{q}_{(b)}^\mathsf{T}M_{ab}\right\}_{a,b=1}^K + \frac{1}{\sqrt{n}}\mathbf{X} \tag{7}$$

$$= \underbrace{\frac{\mathbf{q}\mathbf{q}^\mathsf{T}}{\sqrt{n}}}_{\mathbf{A}_{d,\sqrt{n}}} + \underbrace{\frac{1}{n}\mathbf{D}_q\mathbf{J}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q}_{\mathbf{A}_{d,1}} + \underbrace{\frac{\mathbf{X}}{\sqrt{n}}}_{\mathbf{A}_{r,1}}, \tag{8}$$

where[13] $\mathbf{q}_{(i)} = [q_{n_1+\ldots+n_{i-1}+1}, \ldots, q_{n_1+\ldots+n_i}]^\mathsf{T} \in \mathbb{R}^{n_i}$ ($n_0 = 0$) , $\mathbf{X} = \{X_{ij}\}_{i,j=1}^n$ and $\mathbf{D}_q = \mathcal{D}(\mathbf{q})$. The idea of the proof is to write all the terms of $\mathbf{L}_\alpha$ based on Equation (8), since all those terms depend on $\mathbf{A}$. To this end, we will evaluate successively $\mathbf{d} = \mathbf{A}\mathbf{1}_n$, $\mathbf{D} = \mathcal{D}(\mathbf{d})$, $\mathbf{d}\mathbf{d}^\mathsf{T}$ and $2m = \mathbf{d}^\mathsf{T}\mathbf{1}_n$. It will appear that $\mathbf{D}$ and $\mathbf{d}^\mathsf{T}\mathbf{1}_n$ are composed of dominant terms (with higher operator norm) and vanishing terms (with smaller operator norm); we may then proceed to writing a Taylor expansion of $\mathbf{D}^{-\alpha}$ and $(2m)^\alpha = (\mathbf{d}^\mathsf{T}\mathbf{1}_n)^\alpha$ for any α around their dominant terms to finally retrieve a Taylor expansion of $\mathbf{L}_\alpha$.

13. We recall that subscript 'd, n^k' stands for deterministic term whose operator norm is of order n^k and 'r, n^k' for random term with operator norm of order n^k.

Let us start by developing the degree vector $\mathbf{d} = \mathbf{A}\mathbf{1}_n$. We have

$$\mathbf{d} = \mathbf{q}\mathbf{q}^\mathsf{T}\mathbf{1}_n + \frac{1}{\sqrt{n}}\mathbf{D}_q\mathbf{J}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q\mathbf{1}_n + \mathbf{X}\mathbf{1}_n = \mathbf{q}^\mathsf{T}\mathbf{1}_n\Big(\underbrace{\mathbf{q}}_{\mathcal{O}(n^{\frac{1}{2}})} + \underbrace{\frac{1}{\sqrt{n}}\frac{\mathbf{D}_q\mathbf{J}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q\mathbf{1}_n}{\mathbf{q}^\mathsf{T}\mathbf{1}_n}}_{\mathcal{O}(n^{-\frac{1}{2}})} + \underbrace{\frac{\mathbf{X}\mathbf{1}_n}{\mathbf{q}^\mathsf{T}\mathbf{1}_n}}_{\mathcal{O}(n^{-\frac{1}{2}})} \Big). \quad (9)$$

Let us then write the expansions of $\mathbf{d}^\mathsf{T}\mathbf{1}_n$, $(\mathbf{d}^\mathsf{T}\mathbf{1}_n)^\alpha$, $\mathbf{d}\mathbf{d}^\mathsf{T}$ and $\frac{\mathbf{d}\mathbf{d}^\mathsf{T}}{(\mathbf{d}^\mathsf{T}\mathbf{1}_n)}$ respectively. From (9), we obtain

$$\mathbf{d}^\mathsf{T}\mathbf{1}_n = (\mathbf{q}^\mathsf{T}\mathbf{1}_n)^2\Big[1 + \underbrace{\frac{1}{\sqrt{n}}\frac{\mathbf{1}_n^\mathsf{T}\mathbf{D}_q\mathbf{J}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q\mathbf{1}_n}{(\mathbf{q}^\mathsf{T}\mathbf{1}_n)^2}}_{\mathcal{O}(n^{-\frac{1}{2}})} + \underbrace{\frac{\mathbf{1}_n^\mathsf{T}\mathbf{X}\mathbf{1}_n}{(\mathbf{q}^\mathsf{T}\mathbf{1}_n)^2}}_{\mathcal{O}(n^{-\frac{1}{2}})}\Big]. \quad (10)$$

Thus for any α, proceeding to a 1^{st} order Taylor expansion, we may write

$$(\mathbf{d}^\mathsf{T}\mathbf{1}_n)^\alpha = (\mathbf{q}^\mathsf{T}\mathbf{1}_n)^{2\alpha}\Big[1 + \frac{\alpha}{\sqrt{n}}\underbrace{\frac{\mathbf{1}_n^\mathsf{T}\mathbf{D}_q\mathbf{J}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q\mathbf{1}_n}{(\mathbf{q}^\mathsf{T}\mathbf{1}_n)^2}}_{\mathcal{O}(n^{-\frac{1}{2}})} + \alpha\underbrace{\frac{\mathbf{1}_n^\mathsf{T}\mathbf{X}\mathbf{1}_n}{(\mathbf{q}^\mathsf{T}\mathbf{1}_n)^2}}_{\mathcal{O}(n^{-\frac{1}{2}})} + o(n^{-\frac{1}{2}})\Big]. \quad (11)$$

Besides, from (9) we have

$$\begin{aligned}\mathbf{d}\mathbf{d}^\mathsf{T} = (\mathbf{q}^\mathsf{T}\mathbf{1}_n)^2\Big[&\underbrace{\mathbf{q}\mathbf{q}^\mathsf{T}}_{\mathcal{O}(n)} + \underbrace{\frac{1}{\sqrt{n}}\frac{\mathbf{q}\mathbf{1}_n^\mathsf{T}\mathbf{D}_q\mathbf{J}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q}{\mathbf{q}^\mathsf{T}\mathbf{1}_n}}_{\mathcal{O}(\sqrt{n})} + \underbrace{\frac{1}{\sqrt{n}}\frac{\mathbf{D}_q\mathbf{J}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q\mathbf{1}_n\mathbf{q}^\mathsf{T}}{\mathbf{q}^\mathsf{T}\mathbf{1}_n}}_{\mathcal{O}(\sqrt{n})} + \underbrace{\frac{\mathbf{q}\mathbf{1}_n^\mathsf{T}\mathbf{X}}{\mathbf{q}^\mathsf{T}\mathbf{1}_n}}_{\mathcal{O}(\sqrt{n})} + \underbrace{\frac{\mathbf{X}\mathbf{1}_n\mathbf{q}^\mathsf{T}}{\mathbf{q}^\mathsf{T}\mathbf{1}_n}}_{\mathcal{O}(\sqrt{n})} \\ &+ \underbrace{\frac{1}{n}\frac{\mathbf{D}_q\mathbf{J}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q\mathbf{1}_n\mathbf{1}_n^\mathsf{T}\mathbf{D}_q\mathbf{J}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q}{(\mathbf{q}^\mathsf{T}\mathbf{1}_n)^2}}_{\mathcal{O}(1)} + \underbrace{\frac{1}{\sqrt{n}}\frac{\mathbf{D}_q\mathbf{J}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q\mathbf{1}_n\mathbf{1}_n^\mathsf{T}\mathbf{X}}{(\mathbf{q}^\mathsf{T}\mathbf{1}_n)^2}}_{\mathcal{O}(1)} + \underbrace{\frac{1}{\sqrt{n}}\frac{\mathbf{X}\mathbf{1}_n\mathbf{1}_n^\mathsf{T}\mathbf{D}_q\mathbf{J}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q}{(\mathbf{q}^\mathsf{T}\mathbf{1}_n)^2}}_{\mathcal{O}(1)} \\ &+ \underbrace{\frac{\mathbf{X}\mathbf{1}_n\mathbf{1}_n^\mathsf{T}\mathbf{X}}{(\mathbf{q}^\mathsf{T}\mathbf{1}_n)^2}}_{\mathcal{O}(1)} + o(1)\Big]. \end{aligned} \quad (12)$$

Keeping in mind that we shall only need terms with non vanishing operator norms asymptotically, we will require $\frac{1}{\sqrt{n}}\left[\mathbf{A} - \frac{\mathbf{d}\mathbf{d}^\mathsf{T}}{\mathbf{d}^\mathsf{T}\mathbf{1}_n}\right]$ to have terms with spectral norms of order at least $\mathcal{O}(1)$. We get from multiplying (12) and (11) (with $\alpha = -1$)

$$\begin{aligned}\frac{1}{\sqrt{n}}\frac{\mathbf{d}\mathbf{d}^\mathsf{T}}{\mathbf{d}^\mathsf{T}\mathbf{1}_n} = &\frac{\mathbf{q}\mathbf{q}^\mathsf{T}}{\sqrt{n}} + \frac{1}{n}\frac{\mathbf{q}\mathbf{1}_n^\mathsf{T}\mathbf{D}_q\mathbf{J}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q}{\mathbf{q}^\mathsf{T}\mathbf{1}_n} + \frac{1}{n}\frac{\mathbf{D}_q\mathbf{J}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q\mathbf{1}_n\mathbf{q}^\mathsf{T}}{\mathbf{q}^\mathsf{T}\mathbf{1}_n} + \frac{1}{\sqrt{n}}\frac{\mathbf{q}\mathbf{1}_n^\mathsf{T}\mathbf{X}}{\mathbf{q}^\mathsf{T}\mathbf{1}_n} + \frac{1}{\sqrt{n}}\frac{\mathbf{X}\mathbf{1}_n\mathbf{q}^\mathsf{T}}{\mathbf{q}^\mathsf{T}\mathbf{1}_n} \\ &- \frac{1}{n}\frac{\mathbf{1}_n^\mathsf{T}\mathbf{D}_q\mathbf{J}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q\mathbf{1}_n}{(\mathbf{q}^\mathsf{T}\mathbf{1}_n)^2}\mathbf{q}\mathbf{q}^\mathsf{T} - \frac{1}{\sqrt{n}}\frac{\mathbf{1}_n^\mathsf{T}\mathbf{X}\mathbf{1}_n}{(\mathbf{q}^\mathsf{T}\mathbf{1}_n)^2}\mathbf{q}\mathbf{q}^\mathsf{T} + \mathcal{O}(n^{-\frac{1}{2}}). \end{aligned} \quad (13)$$

By subtracting (13) from (8), we obtain

$$\begin{aligned}\frac{1}{\sqrt{n}}\left(\mathbf{A} - \frac{\mathbf{d}\mathbf{d}^\mathsf{T}}{\mathbf{d}^\mathsf{T}\mathbf{1}_n}\right) = &\frac{1}{n}\mathbf{D}_q\mathbf{J}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q - \frac{1}{n}\frac{\mathbf{q}\mathbf{1}_n^\mathsf{T}\mathbf{D}_q\mathbf{J}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q}{\mathbf{q}^\mathsf{T}\mathbf{1}_n} - \frac{1}{n}\frac{\mathbf{D}_q\mathbf{J}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q\mathbf{1}_n\mathbf{q}^\mathsf{T}}{\mathbf{q}^\mathsf{T}\mathbf{1}_n} \\ &+ \frac{1}{n}\frac{\mathbf{1}_n^\mathsf{T}\mathbf{D}_q\mathbf{J}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q\mathbf{1}_n}{(\mathbf{q}^\mathsf{T}\mathbf{1}_n)^2}\mathbf{q}\mathbf{q}^\mathsf{T} + \frac{\mathbf{X}}{\sqrt{n}} - \frac{1}{\sqrt{n}}\frac{\mathbf{q}\mathbf{1}_n^\mathsf{T}\mathbf{X}}{\mathbf{q}^\mathsf{T}\mathbf{1}_n} - \frac{1}{\sqrt{n}}\frac{\mathbf{X}\mathbf{1}_n\mathbf{q}^\mathsf{T}}{\mathbf{q}^\mathsf{T}\mathbf{1}_n} \\ &+ \frac{1}{\sqrt{n}}\frac{\mathbf{1}_n^\mathsf{T}\mathbf{X}\mathbf{1}_n}{(\mathbf{q}^\mathsf{T}\mathbf{1}_n)^2}\mathbf{q}\mathbf{q}^\mathsf{T} + \mathcal{O}(n^{-\frac{1}{2}}). \end{aligned} \quad (14)$$

It then remains to evaluate $\mathbf{D}^{-\alpha}$. From (9), we may write $\mathbf{D} = \mathcal{D}(\mathbf{d})$ as

$$\mathbf{D} = \mathbf{q}^\mathsf{T}\mathbf{1}_n\Big(\underbrace{\mathbf{D}_q}_{\mathcal{O}(1)} + \underbrace{\mathcal{D}\left(\frac{1}{\sqrt{n}}\frac{\mathbf{D}_q\mathbf{J}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q\mathbf{1}_n}{\mathbf{q}^\mathsf{T}\mathbf{1}_n}\right)}_{\mathcal{O}(n^{-\frac{1}{2}})} + \underbrace{\mathcal{D}\left(\frac{\mathbf{X}\mathbf{1}_n}{\mathbf{q}^\mathsf{T}\mathbf{1}_n}\right)}_{\mathcal{O}(n^{-\frac{1}{2}})}\Big).$$

The right hand side of $\mathbf{D}$ (in brackets) having a leading term in $\mathcal{O}(1)$ and residual terms in $\mathcal{O}(n^{-\frac{1}{2}})$, the Taylor expansion of the $(-\alpha)$-power of $\mathbf{D}$ is then retrieved

$$\mathbf{D}^{-\alpha} = \left(\mathbf{q}^\mathsf{T}\mathbf{1}_n\right)^{-\alpha}\Big(\underbrace{\mathbf{D}_q^{-\alpha}}_{\mathcal{O}(1)} - \alpha\mathbf{D}_q^{-(\alpha+1)}\underbrace{\mathcal{D}\left(\frac{1}{\sqrt{n}}\frac{\mathbf{D}_q\mathbf{J}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q\mathbf{1}_n}{\mathbf{q}^\mathsf{T}\mathbf{1}_n}\right)}_{\mathcal{O}(n^{-\frac{1}{2}})} - \alpha\mathbf{D}_q^{-(\alpha+1)}\underbrace{\mathcal{D}\left(\frac{\mathbf{X}\mathbf{1}_n}{\mathbf{q}^\mathsf{T}\mathbf{1}_n}\right)}_{\mathcal{O}(n^{-\frac{1}{2}})} + \mathcal{O}(n^{-1})\Big). \quad (15)$$

By combining the expressions (11), (14) and (15), we obtain a Taylor approximation of $\mathbf{L}_\alpha$ as follows

$$\begin{aligned}\mathbf{L}_\alpha = &\mathbf{D}_q^{-\alpha}\frac{\mathbf{X}}{\sqrt{n}}\mathbf{D}_q^{-\alpha} + \frac{1}{n}\mathbf{D}_q^{1-\alpha}\mathbf{J}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q^{1-\alpha} - \frac{1}{n}\frac{\mathbf{D}_q^{1-\alpha}\mathbf{1}_n\mathbf{1}_n^\mathsf{T}\mathbf{D}_q\mathbf{J}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q^{1-\alpha}}{\mathbf{q}^\mathsf{T}\mathbf{1}_n} \\ &- \frac{1}{n}\frac{\mathbf{D}_q^{1-\alpha}\mathbf{J}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q\mathbf{1}_n\mathbf{1}_n^\mathsf{T}\mathbf{D}_q^{1-\alpha}}{\mathbf{q}^\mathsf{T}\mathbf{1}_n} + \frac{1}{n}\frac{\mathbf{1}_n^\mathsf{T}\mathbf{D}_q\mathbf{J}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q\mathbf{1}_n}{(\mathbf{q}^\mathsf{T}\mathbf{1}_n)^2}\mathbf{D}_q^{1-\alpha}\mathbf{1}_n\mathbf{1}_n^\mathsf{T}\mathbf{D}_q^{1-\alpha} - \frac{1}{\sqrt{n}}\frac{\mathbf{D}_q^{1-\alpha}\mathbf{1}_n\mathbf{1}_n^\mathsf{T}\mathbf{X}\mathbf{D}_q^{-\alpha}}{\mathbf{q}^\mathsf{T}\mathbf{1}_n} \\ &- \frac{1}{\sqrt{n}}\frac{\mathbf{D}_q^{-\alpha}\mathbf{X}\mathbf{1}_n\mathbf{1}_n^\mathsf{T}\mathbf{D}_q^{1-\alpha}}{\mathbf{q}^\mathsf{T}\mathbf{1}_n} + \frac{1}{\sqrt{n}}\frac{\mathbf{1}_n^\mathsf{T}\mathbf{X}\mathbf{1}_n}{(\mathbf{q}^\mathsf{T}\mathbf{1}_n)^2}\mathbf{D}_q^{1-\alpha}\mathbf{1}_n\mathbf{1}_n^\mathsf{T}\mathbf{D}_q^{1-\alpha} + \mathcal{O}(n^{-\frac{1}{2}}). \end{aligned}$$

The three following arguments allow to complete the proof

- $\mathbf{1}_n = \mathbf{J}\mathbf{1}_K$ and $\mathbf{D}_q\mathbf{1}_n = \mathbf{q}$.
- We may write $(\frac{1}{n}\mathbf{J}^\mathsf{T}\mathbf{q})_i = \frac{n_i}{n}\left(\frac{1}{n_i}\sum_{a\in\mathcal{C}_i} q_a\right)$. For classes of large sizes n_i, from the law of large numbers, $\left(\frac{1}{n_i}\sum_{a\in\mathcal{C}_i} q_a\right) \xrightarrow{\text{a.s.}} m_\mu$ and so, $\frac{1}{n}\mathbf{J}^\mathsf{T}\mathbf{q} \xrightarrow{\text{a.s.}} m_\mu\mathbf{c}$ where we recall that $m_\mu = \int t\mu(\mathrm{d}t)$.
- As $\mathbf{X}$ is a symmetric random matrix having independent entries of zero mean and finite variance, from the law of large numbers, we have $\frac{1}{n}\frac{\mathbf{1}_n^\mathsf{T}\mathbf{X}\mathbf{1}_n}{\sqrt{n}} \xrightarrow{\text{a.s.}} 0$.

Using those three arguments, $\mathbf{L}_\alpha$ may be further rewritten

$$\begin{aligned}\mathbf{L}_\alpha = &\mathbf{D}_q^{-\alpha}\frac{\mathbf{X}}{\sqrt{n}}\mathbf{D}_q^{-\alpha} + \frac{1}{n}\mathbf{D}_q^{1-\alpha}\mathbf{J}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q^{1-\alpha} - \frac{1}{n}\mathbf{D}_q^{1-\alpha}\mathbf{J}\mathbf{1}_K\mathbf{c}^\mathsf{T}\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{D}_q^{1-\alpha} \\ &- \frac{1}{n}\mathbf{D}_q^{1-\alpha}\mathbf{J}\mathbf{M}\mathbf{c}\mathbf{1}_K^\mathsf{T}\mathbf{J}^\mathsf{T}\mathbf{D}_q^{1-\alpha} + \frac{1}{n}\mathbf{D}_q^{1-\alpha}\mathbf{J}\mathbf{1}_K\mathbf{c}^\mathsf{T}\mathbf{M}\mathbf{c}\mathbf{1}_K^\mathsf{T}\mathbf{J}^\mathsf{T}\mathbf{D}_q^{1-\alpha} \\ &- \frac{1}{\sqrt{n}\mathbf{q}^\mathsf{T}\mathbf{1}_n}\mathbf{D}_q^{1-\alpha}\mathbf{J}\mathbf{1}_K\mathbf{1}_n^\mathsf{T}\mathbf{X}\mathbf{D}_q^{-\alpha} - \frac{1}{\sqrt{n}\mathbf{q}^\mathsf{T}\mathbf{1}_n}\mathbf{D}_q^{-\alpha}\mathbf{X}\mathbf{1}_n\mathbf{1}_K^\mathsf{T}\mathbf{J}^\mathsf{T}\mathbf{D}_q^{1-\alpha} + \mathcal{O}(n^{-\frac{1}{2}}). \end{aligned} \quad (16)$$

By rearranging the terms of (16), we obtain the expected result

$$\begin{aligned}\mathbf{L}_\alpha = &\mathbf{D}_q^{-\alpha}\frac{\mathbf{X}}{\sqrt{n}}\mathbf{D}_q^{-\alpha} \\ &+ \begin{bmatrix}\frac{\mathbf{D}_q^{1-\alpha}\mathbf{J}}{\sqrt{n}} & \frac{\mathbf{D}_q^{-\alpha}\mathbf{X}\mathbf{1}_n}{\mathbf{q}^\mathsf{T}\mathbf{1}_n}\end{bmatrix}\begin{bmatrix}\left(\mathbf{I}_K - \mathbf{1}_K\mathbf{c}^T\right)\mathbf{M}\left(\mathbf{I}_K - \mathbf{c}\mathbf{1}_K^T\right) & -\mathbf{1}_K \\ -\mathbf{1}_K^T & 0\end{bmatrix}\begin{bmatrix}\frac{\mathbf{J}^\mathsf{T}\mathbf{D}_q^{1-\alpha}}{\sqrt{n}} \\ \frac{\mathbf{1}_n^\mathsf{T}\mathbf{X}\mathbf{D}_q^{-\alpha}}{\mathbf{q}^\mathsf{T}\mathbf{1}_n}\end{bmatrix} + \mathcal{O}(n^{-\frac{1}{2}}). \end{aligned}$$

This proves Theorem 2.

6.2 Limiting spectral distribution of $\mathbf{L}_\alpha$

It follows from Theorem 2 that $\tilde{\mathbf{L}}_\alpha = \mathbf{D}_q^{-\alpha}\frac{\mathbf{X}}{\sqrt{n}}\mathbf{D}_q^{-\alpha} + \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\mathsf{T}$ is equivalent to an additive spiked random matrix (Chapon et al., 2012) where

$$\mathbf{U} = \begin{bmatrix} \frac{\mathbf{D}_q^{1-\alpha}\mathbf{J}}{\sqrt{n}} & \frac{\mathbf{D}_q^{-\alpha}\mathbf{X}\mathbf{1}_n}{\mathbf{q}^\mathsf{T}\mathbf{1}_n} \end{bmatrix},$$
$$\mathbf{\Lambda} = \begin{bmatrix} \left(\mathbf{I}_K - \mathbf{1}_K\mathbf{c}^T\right)\mathbf{M}\left(\mathbf{I}_K - \mathbf{c}\mathbf{1}_K^T\right) & -\mathbf{1}_K \\ -\mathbf{1}_K^T & 0 \end{bmatrix},$$

with the difference that the deterministic part $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\mathsf{T}$ is not independent of the random part $\mathbf{D}_q^{-\alpha}\frac{\mathbf{X}}{\sqrt{n}}\mathbf{D}_q^{-\alpha}$ (an issue that we solve here) and $\mathbf{U}$ is not composed of orthonormal vectors. Let us then study $\bar{\mathbf{X}} = \mathbf{D}_q^{-\alpha}\frac{\mathbf{X}}{\sqrt{n}}\mathbf{D}_q^{-\alpha}$ (having entries $\bar{X}_{ij}$ with zero mean and variance σ_{ij}^2/n with $\sigma_{ij}^2 = q_iq_j(1-q_iq_j) + \mathcal{O}(n^{-\frac{1}{2}})$) and show that its empirical spectral distribution (e.s.d.) $\tilde{\pi}^\alpha$ converges weakly to $\bar{\pi}^\alpha$ with Stieljes transform $e_{00}^\alpha(z) = \int (t-z)^{-1}\,\mathrm{d}\bar{\pi}^\alpha(t)$ for $z \in \mathbb{C}^+$. This will imply (By Weyl interlacing formula) that the empirical spectral measure $\pi^\alpha \equiv \frac{1}{n}\sum_{i=1}^n \delta_{\lambda_i(\tilde{\mathbf{L}}_\alpha)}$ (with $\lambda_i(\tilde{\mathbf{L}}_\alpha)$ eigenvalues of $\tilde{\mathbf{L}}_\alpha$) will also converge to $\bar{\pi}^\alpha$.

The matrix $\bar{\mathbf{X}}$ is a classical random matrix model in RMT already studied in similar cases (Pastur et al., 2011). It is well known for those random matrix models (having entries with given means, variances and bounded first order moments) that the law of the $\bar{X}_{ij}$'s does not change the results on the limiting law of the e.s.d. $\tilde{\pi}^\alpha$: this property is kwown as *universality* (e.g., (Silverstein and Bai, 1995)). For technical reasons, we can thus assume that the $\bar{X}_{ij}$'s are Gaussian random variables with the same means and variances in order to use standard Gaussian calculus, introduced in (Pastur et al., 2011). The objective of the proof is to find the deterministic limit $e_{00}^\alpha(z)$ for the random quantity $\frac{1}{n}\operatorname{tr}\left(\bar{\mathbf{X}} - z\mathbf{I}_n\right)^{-1}$ which is the Stieljes transform of the e.s.d. $\tilde{\pi}^\alpha$. Deterministic equivalents for the Stieljes transform of empirical spectral measures associated with centered and symmetric random matrix models with a variance profile have already been studied in for example (Ajanki et al., 2015; Hachem et al., 2007). We give in Appendix C an exhaustive development of the Gaussian calculus to obtain $e_{00}^\alpha(z)$. The final result is as follows.

Lemma 16 (A first deterministic equivalent) *Let $\mathbf{Q} = (\bar{\mathbf{X}} - z\mathbf{I}_n)^{-1}$. Then, for all $z \in \mathbb{C}^+$,*

$$\mathbf{Q} \leftrightarrow \bar{\mathbf{Q}} = \left(-z\mathbf{I}_n - \mathcal{D}\left(e_i(z)\right)_{i=1}^n\right)^{-1} \tag{17}$$

where $e_i(z)$ the unique solution of $e_i(z) = \frac{1}{n}\operatorname{tr}\mathcal{D}\left(\sigma_{ij}^2\right)_{j=1}^n\left(-z\mathbf{I}_n - \mathcal{D}\left(e_j(z)\right)_{j=1}^n\right)^{-1}$ and the notation $\mathbf{A} \leftrightarrow \mathbf{B}$ stands for $\frac{1}{n}\operatorname{tr}\mathbf{C}\mathbf{A} - \frac{1}{n}\operatorname{tr}\mathbf{C}\mathbf{B} \to 0$ and $\mathbf{d}_1^\mathsf{T}(\mathbf{A}-\mathbf{B})\mathbf{d}_2 \to 0$ almost surely, for all deterministic Hermitian matrix $\mathbf{C}$ and deterministic vectors $\mathbf{d}_i$ of bounded norms (spectral norm for matrices and Euclidian norm for vectors).

From Lemma 16, we get directly $\frac{1}{n}\operatorname{tr}\mathbf{Q} - e_{00}^\alpha(z) \xrightarrow{\text{a.s.}} 0$ with $e_{00}^\alpha(z) = \frac{1}{n}\sum_{i=1}^n \frac{1}{-z-e_i(z)}$. Observe now that

$$\begin{aligned} e_i(z) &= \frac{1}{n}\sum_{j=1}^n \frac{q_i^{1-2\alpha}q_j^{1-2\alpha} - q_i^{2-2\alpha}q_j^{2-2\alpha}}{-z - e_j(z)} \\ &= q_i^{1-2\alpha}e_{11}^\alpha(z) - q_i^{2-2\alpha}e_{21}^\alpha(z) \end{aligned} \tag{18}$$

where

$$\begin{aligned} e_{11}^\alpha(z) &= \frac{1}{n}\sum_{j=1}^n \frac{q_j^{1-2\alpha}}{-z - q_j^{1-2\alpha}e_{11}^\alpha(z) + q_j^{2-2\alpha}e_{21}^\alpha(z)} \\ e_{21}^\alpha(z) &= \frac{1}{n}\sum_{j=1}^n \frac{q_j^{2-2\alpha}}{-z - q_j^{1-2\alpha}e_{11}^\alpha(z) + q_j^{2-2\alpha}e_{21}^\alpha(z)} \end{aligned} \tag{19}$$

from which we get

$$e_{00}^\alpha(z) = \int \frac{1}{-z - e_{11}^\alpha(z)q^{1-2\alpha} + e_{21}^\alpha(z)q^{2-2\alpha}}\mu(dq).$$

where for $z \in \mathbb{C}^+$ and $a, b \in \mathbb{Z}$ we define

$$e_{ab}^\alpha(z) = \int \frac{q^{a-2b\alpha}\mu(dq)}{-z - e_{11}^\alpha(z)q^{1-2\alpha} + e_{21}^\alpha(z)q^{2-2\alpha}}. \tag{20}$$

with $\mu(\mathrm{d}q) = \lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^n \delta_{q_i}$. From this, we have that $e_{00}^\alpha(z)$ does not depend on n, so that $\frac{1}{n}\operatorname{tr}\mathbf{Q} \xrightarrow{\text{a.s.}} E_0^\alpha(z)$, $\tilde{\pi}^\alpha \to \bar{\pi}^\alpha$, and thus $\pi^\alpha \to \bar{\pi}^\alpha$ since $\tilde{\mathbf{L}}_\alpha$ and $\bar{\mathbf{X}}$ only differ by a finite rank matrix. This proves Theorem 4.

In the main core of the article, we have defined $e_{00}^\alpha(z) \triangleq m^\alpha(z)$, $e_{11}^\alpha(z) \triangleq f^\alpha(z)$ and $e_{21}^\alpha(z) \triangleq g^\alpha(z)$ for readability reasons. For future use, we define for $z, \tilde{z} \in \mathbb{C}\setminus\mathcal{S}^\alpha$

$$e_{ab;2}^\alpha(z,\tilde{z}) = \int \frac{q^{a-2b\alpha}\mu(dq)}{(-z - E_1^\alpha(z)q^{1-2\alpha} + E_2^\alpha(z)q^{2-2\alpha})(-\tilde{z} - E_1^\alpha(\tilde{z})q^{1-2\alpha} + E_2^\alpha(\tilde{z})q^{2-2\alpha})} \tag{21}$$

and

$$e_{ab;3}^\alpha(z,\tilde{z}) = \int \frac{q^{a-2b\alpha}\mu(dq)}{(-z - E_1^\alpha(z)q^{1-2\alpha} + E_2^\alpha(z)q^{2-2\alpha})^2(-\tilde{z} - E_1^\alpha(\tilde{z})q^{1-2\alpha} + E_2^\alpha(\tilde{z})q^{2-2\alpha})}. \tag{22}$$

Convergence of the e_i's

Similar results to Lemma 16 have been derived for example in (Hachem et al., 2007) and the fixed point algorithm (17) which consists of iterating the e_i's is shown to converge. Since the calcultation of the e_{ab}'s is an intermediary step of (17) from (18), the fixed point algorithm (19) also converges. From the analyticity of the Stieljes transform outside its support, Lemma 16 extends naturally to $\mathbb{C}\setminus\mathcal{S}^\alpha$. This proves Theorem 4.

Remark 17 *Similarly to (Hachem et al., 2007), when none of the $(\mathbf{D}_q^{-\alpha})_{ii}$'s is isolated, the random matrix $\bar{\mathbf{X}}$ does not produce isolated eigenvalues outside the support $\mathcal{S}^\alpha$ of $\bar{\pi}^\alpha$. Here, for large n, this property is verified since from Assumption 1, the q_i's are i.i.d. arising from a law with compact support (the probability that a $(\mathbf{D}_q^{-\alpha})_{ii}$ gets isolated tends to 0 asymptotically). This gives Proposition 18 which we will not prove here; similar proofs are provided for example in (Bai and Silverstein, 1998).*

Proposition 18 (No eigenvalues outside the support) *Following the statement of Theorem 4, let S_-^α and S_+^α be respectively the left and right edges of $\mathcal{S}^\alpha$. Then, for any $\epsilon > 0$, by letting $\mathcal{S}_\epsilon^\alpha = [S_-^\alpha - \epsilon; S_+^\alpha + \epsilon]$, for all large n almost surely,*

$$\left\{\lambda_i\Big(\mathbf{D}_q^{-\alpha}\frac{\mathbf{X}}{\sqrt{n}}\mathbf{D}_q^{-\alpha}\Big), 1 \le i \le n\right\} \cap (\mathbb{R} \setminus \mathcal{S}_\epsilon^\alpha) = \emptyset.$$

Remark 19 *The support $\mathcal{S}^\alpha$ is symmetric i.e., $\bar\pi^\alpha([a,b]) = \bar\pi^\alpha([-b,-a])$. We have in particular $S_-^\alpha = -S_+^\alpha = -S^\alpha$ where we denote $S_+^\alpha \triangleq \sup \mathcal{S}^\alpha$ and $S_-^\alpha \triangleq \inf \mathcal{S}^\alpha$.*

6.3 Isolated eigenvalues of $\mathbf{L}_\alpha$ and phase transition.

In the previous section, we have shown that the e.s.d. of $\mathbf{L}_\alpha$ converges weakly to the limiting law of the eigenvalues of $\bar{\mathbf{X}}$ since they only differ by a finite rank matrix. We shall have in addition isolated eigenvalues of $\mathbf{L}_\alpha$ induced by the aforementionned low rank matrix. We are interested here in the localization of eigenvalues of $\mathbf{L}_\alpha$ isolated from the support $\mathcal{S}^\alpha$ of the limiting law of its e.s.d. According to Proposition 18, there is almost surely no eigenvalue of $\bar{\mathbf{X}}$ at non-vanishing distance from $\mathcal{S}^\alpha$ asymptotically as $n \to \infty$ and hence the plausible isolated eigenvalues of $\mathbf{L}_\alpha$ are only due to the matrix $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\mathsf{T}$. We follow classical random matrix approaches used for the study of the spectrum of spiked random matrices (Benaych-Georges and Nadakuditi, 2012; Chapon et al., 2012). From Theorem 2, the eigenvalues ρ of $\mathbf{L}_\alpha$ falling at non-vanishing distance from the limiting support $\mathcal{S}^\alpha$ solve for large n, $0 = \det(\mathbf{L}_\alpha - \rho\mathbf{I}_n)$ almost surely for $\rho \notin \mathcal{S}^\alpha$. Since $\|\mathbf{L}_\alpha - \tilde{\mathbf{L}}_\alpha\| \xrightarrow{\text{a.s.}} 0$, $\rho_i(\mathbf{L}_\alpha) - \rho_i(\tilde{\mathbf{L}}_\alpha) \xrightarrow{\text{a.s.}} 0$ for all eigenvalues $\rho_i(\mathbf{L}_\alpha)$. We may then just solve $0 = \det(\mathbf{D}_q^{-\alpha}\frac{\mathbf{X}}{\sqrt{n}}\mathbf{D}_q^{-\alpha} + \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\mathsf{T} - \rho\mathbf{I}_n)$. Now, as from Proposition 18, the random matrix $\bar{\mathbf{X}}$ does not have eigenvalues at non-vanishing distance from $\mathcal{S}^\alpha$ asymptotically, for $\rho \notin \mathcal{S}^\alpha$, we can thus factor and cancel out $\det(\bar{\mathbf{X}} - \rho\mathbf{I}_n)$ from the previous determinant equation, so that we are left to solve

$$0 = \det(\mathbf{I}_n + \mathbf{Q}_\rho^\alpha \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\mathsf{T}) = \det(\mathbf{I}_{K+1} + \mathbf{U}^\mathsf{T}\mathbf{Q}_\rho^\alpha \mathbf{U}\mathbf{\Lambda})$$

where $\mathbf{Q}_\rho^\alpha = (\bar{\mathbf{X}} - \rho\mathbf{I}_n)^{-1}$. As we will show next, the matrix $\mathbf{I}_{K+1} + \mathbf{U}^\mathsf{T}\mathbf{Q}_\rho^\alpha\mathbf{U}\mathbf{\Lambda}$ converges to a deterministic matrix, almost surely for large n. By the argument principle (similar to e.g., (Chapon et al., 2012)), the roots of $\mathbf{I}_{K+1} + \mathbf{U}^\mathsf{T}\mathbf{Q}_\rho^\alpha\mathbf{U}\mathbf{\Lambda}$ are asymptotically those of the limiting matrix, with same multiplicity and it suffices to study the latter.

We then proceed to retrieving a limit for $\mathbf{I}_{K+1} + \mathbf{U}^\mathsf{T}\mathbf{Q}_\rho^\alpha\mathbf{U}\mathbf{\Lambda}$. From Theorem 2, we have

$$\mathbf{U}^\mathsf{T}\mathbf{Q}_\rho^\alpha\mathbf{U} = \begin{pmatrix} \frac{1}{n}\mathbf{J}^\mathsf{T}\mathbf{D}_q^{1-\alpha}\mathbf{Q}_\rho^\alpha\mathbf{D}_q^{1-\alpha}\mathbf{J} & \frac{1}{\sqrt{n}(\mathbf{q}^\mathsf{T}\mathbf{1}_n)}\mathbf{J}^\mathsf{T}\mathbf{D}_q^{1-\alpha}\mathbf{Q}_\rho^\alpha\mathbf{D}_q^{-\alpha}\mathbf{X}\mathbf{1}_n \\ \frac{1}{\sqrt{n}(\mathbf{q}^\mathsf{T}\mathbf{1}_n)}\mathbf{1}_n^\mathsf{T}\mathbf{X}\mathbf{D}_q^{-\alpha}\mathbf{Q}_\rho^\alpha\mathbf{D}_q^{1-\alpha}\mathbf{J} & \frac{1}{(\mathbf{q}^\mathsf{T}\mathbf{1}_n)^2}\mathbf{1}_n^\mathsf{T}\mathbf{X}\mathbf{D}_q^{-\alpha}\mathbf{Q}_\rho^\alpha\mathbf{D}_q^{-\alpha}\mathbf{X}\mathbf{1}_n \end{pmatrix}.$$

The entries $(1,2)$, $(2,1)$ and $(2,2)$ of $\mathbf{U}^\mathsf{T}\mathbf{Q}_\rho^\alpha\mathbf{U}$ are random as they contain the random matrix $\mathbf{X}$ but tend to be deterministic in the limit. In fact, using the resolvent identity, we have that $\mathbf{Q}_\rho^\alpha\mathbf{D}_q^{-\alpha}\frac{\mathbf{X}}{\sqrt{n}}\mathbf{D}_q^{-\alpha} = \mathbf{I}_n + \rho\mathbf{Q}_\rho^\alpha$, the entry $(1,2)$ becomes $\frac{1}{(\mathbf{q}^\mathsf{T}\mathbf{1}_n)}\mathbf{J}^\mathsf{T}\mathbf{D}_q\mathbf{1}_n + \rho\frac{1}{\sqrt{n}(\mathbf{q}^\mathsf{T}\mathbf{1}_n)}\mathbf{J}^\mathsf{T}\mathbf{D}_q^{1-\alpha}\mathbf{Q}_\rho^\alpha\mathbf{D}_q^\alpha\mathbf{1}_n$ and the entry $(2,2)$ is equal to $\frac{n}{(\mathbf{q}^\mathsf{T}\mathbf{1}_n)^2}\Big(\mathbf{1}_n^\mathsf{T}\mathbf{X}\mathbf{1}_n + \rho\mathbf{1}_n^\mathsf{T}\mathbf{D}_q^{2\alpha}\mathbf{1}_n + \rho^2\mathbf{1}_n^\mathsf{T}\mathbf{D}_q^\alpha\mathbf{Q}_\rho^\alpha\mathbf{D}_q^\alpha\mathbf{1}_n\Big)$. Now, we can freely use Lemma 16 to evaluate the limits of the entries of $\mathbf{U}^\mathsf{T}\mathbf{Q}_\rho^\alpha\mathbf{U}$ since all the terms are of the form $\mathbf{a}^\mathsf{T}\mathbf{Q}_\rho^\alpha\mathbf{b}$ with $\mathbf{a}$ and $\mathbf{b}$ deterministic vectors. From Lemma 16, the entries $(1,1)$, $(1,2)$ and $(2,2)$ converge almost surely respectively to $\frac{1}{n}\mathbf{J}^\mathsf{T}\mathbf{D}_q^{1-\alpha}\bar{\mathbf{Q}}_\rho^\alpha\mathbf{D}_q^{1-\alpha}\mathbf{J}$, $\frac{1}{(\mathbf{q}^\mathsf{T}\mathbf{1}_n)}\mathbf{J}^\mathsf{T}\mathbf{D}_q\mathbf{1}_n + \rho\frac{1}{(\mathbf{q}^\mathsf{T}\mathbf{1}_n)}\mathbf{J}^\mathsf{T}\mathbf{D}_q^{1-\alpha}\bar{\mathbf{Q}}_\rho^\alpha\mathbf{D}_q^\alpha\mathbf{1}_n$ and $\frac{n}{(\mathbf{q}^\mathsf{T}\mathbf{1}_n)^2}\Big(\mathbf{1}_n^\mathsf{T}\mathbf{X}\mathbf{1}_n + \rho\mathbf{1}_n^\mathsf{T}\mathbf{D}_q^{2\alpha}\mathbf{1}_n + \rho^2\mathbf{1}_n^\mathsf{T}\mathbf{D}_q^\alpha\bar{\mathbf{Q}}_\rho^\alpha\mathbf{D}_q^\alpha\mathbf{1}_n\Big)$ for large n.

Now, using the fact that for any bounded continuous function f, from the law of large numbers,

$$\frac{1}{n}\sum_{j\in\mathcal{C}_i} f(q_j) = \frac{n_i}{n}\frac{1}{n_i}\sum_{j\in\mathcal{C}_i} f(q_j) \xrightarrow{\text{a.s.}} c_i \int f(q)\mu(dq). \tag{23}$$

After some algebra, we obtain $\frac{1}{n}\mathbf{J}^\mathsf{T}\mathbf{D}_q^{1-\alpha}\bar{\mathbf{Q}}_\rho^\alpha\mathbf{D}_q^{1-\alpha}\mathbf{J} \xrightarrow{\text{a.s.}} e_{21}^\alpha(\rho)\mathcal{D}(\mathbf{c})$ where the e_{ij}'s are given in Theorem 4. Similarly for the terms $(1,2)$ and $(2,2)$, we obtain respectively

$$\frac{1}{(\mathbf{q}^\mathsf{T}\mathbf{1}_n)}\mathbf{J}^\mathsf{T}\mathbf{D}_q\mathbf{1}_n + \rho\frac{1}{(\mathbf{q}^\mathsf{T}\mathbf{1}_n)}\mathbf{J}^\mathsf{T}\mathbf{D}_q^{1-\alpha}\bar{\mathbf{Q}}_\rho^\alpha\mathbf{D}_q^\alpha\mathbf{1}_n \xrightarrow{\text{a.s.}} \left(1 + \frac{\rho}{m_\mu}e_{10}^\alpha(\rho)\right)\mathbf{c}$$

and

$$\frac{n}{(\mathbf{q}^\mathsf{T}\mathbf{1}_n)^2}\Big(\mathbf{1}_n^\mathsf{T}\mathbf{X}\mathbf{1}_n + \rho\mathbf{1}_n^\mathsf{T}\mathbf{D}_q^{2\alpha}\mathbf{1}_n + \rho^2\mathbf{1}_n^\mathsf{T}\mathbf{D}_q^\alpha\bar{\mathbf{Q}}_\rho^\alpha\mathbf{D}_q^\alpha\mathbf{1}_n\Big) \xrightarrow{\text{a.s.}} \frac{1}{m_\mu^2}\left(\rho v_\mu + \rho^2 e_{0;-1}^\alpha(\rho)\right)$$

with $v_\mu = \int q^{2\alpha}\mu(\mathrm{d}q)$ and where we have also used the fact that $\frac{1}{n}\mathbf{1}_n^\mathsf{T}\frac{\mathbf{X}}{\sqrt{n}}\mathbf{1}_n \xrightarrow{\text{a.s.}} 0$ again from the law of large numbers.

The limit of $\mathbf{I}_{K+1} + \mathbf{U}^\mathsf{T}\mathbf{Q}_\rho^\alpha\mathbf{U}\mathbf{\Lambda}$ is then obtained as

$$\mathbf{I}_{K+1} + \mathbf{U}^\mathsf{T}\mathbf{Q}_\rho^\alpha\mathbf{U}\mathbf{\Lambda} \xrightarrow{\text{a.s.}} \begin{pmatrix} \mathbf{I}_K + e_{21}^\alpha(\rho)(\mathcal{D}(\mathbf{c}) - \mathbf{c}\mathbf{c}^\mathsf{T})\mathbf{M}(\mathbf{I}_K - \mathbf{c}\mathbf{1}_K^\mathsf{T}) - \left(1 + \frac{\rho}{m_\mu}e_{10}^\alpha(\rho)\right)\mathbf{c}\mathbf{1}_K^\mathsf{T} & -e_{21}^\alpha(\rho)\mathbf{c} \\ \frac{\rho}{m_\mu^2}\left(v_\mu + \rho e_{0;-1}^\alpha(\rho)\right)\mathbf{1}_K^\mathsf{T} & -\rho\frac{e_{10}^\alpha(\rho)}{m_\mu} \end{pmatrix}.$$

Using the Schur complement formula for the determinant of block matrices, we have that the determinant of the RHS matrix is zero whenever

$$-\rho\frac{e_{10}^\alpha(\rho)}{m_\mu}\det\Big[\mathbf{I}_K + e_{21}^\alpha(\rho)(\mathcal{D}(\mathbf{c}) - \mathbf{c}\mathbf{c}^\mathsf{T})\mathbf{M}(\mathbf{I}_K - \mathbf{c}\mathbf{1}_K^\mathsf{T}) - \left(1 + \frac{\rho}{m_\mu}e_{10}^\alpha(\rho)\right)\mathbf{c}\mathbf{1}_K^\mathsf{T} + \frac{\left(v_\mu + \rho e_{0;-1}^\alpha(\rho)\right)e_{21}^\alpha(\rho)}{m_\mu e_{10}^\alpha(\rho)}\mathbf{c}\mathbf{1}_K^\mathsf{T}\Big] = 0$$

or equivalently $\det(\underline{\mathbf{G}}_\rho^\alpha) = 0$ where

$$\underline{\mathbf{G}}_\rho^\alpha = \mathbf{I}_K + e_{21}^\alpha(\rho)(\mathcal{D}(\mathbf{c}) - \mathbf{c}\mathbf{c}^\mathsf{T})\mathbf{M}(\mathbf{I}_K - \mathbf{c}\mathbf{1}_K^\mathsf{T}) + \theta^\alpha(\rho)\mathbf{c}\mathbf{1}_K^\mathsf{T}$$
$$\theta^\alpha(\rho) = -1 + \frac{\rho}{m_\mu}e_{10}^\alpha(\rho) + \frac{\left(v_\mu + \rho e_{0;-1}^\alpha(\rho)\right)e_{21}^\alpha(\rho)}{m_\mu e_{10}^\alpha(\rho)}.$$

The isolated eigenvalues ρ of $\mathbf{L}_\alpha$, which are the ρ for which $\det(\mathbf{I}_{K+1} + \mathbf{U}^\mathsf{T}\mathbf{Q}_\rho^\alpha\mathbf{U}\mathbf{\Lambda}) = 0$, are then asymptotically the ρ such that $\det(\underline{\mathbf{G}}_\rho^\alpha) = 0$.

Remark 20 (Two types of isolated eigenvalues) *From the previous paragraph, $1+\theta^\alpha(\rho)$ is an eigenvalue of $\underline{\mathbf{G}}_\rho^\alpha$ with associated left eigenvector $\mathbf{1}_K$ and right eigenvector $\mathbf{c}$ since $\mathbf{1}_K^T\underline{\mathbf{G}}_\rho^\alpha = (1+\theta^\alpha(\rho))\,\mathbf{1}_K^T$ and $\underline{\mathbf{G}}_\rho^\alpha\mathbf{c} = (1+\theta^\alpha(\rho))\,\mathbf{c}$.*

Letting ρ be such that $\det(\underline{\mathbf{G}}_\rho^\alpha) = 0$, we can discriminate two cases

- *$1+\theta^\alpha(\rho) = 0$: isolated eigenvalues are found for those $\rho \in \mathbb{R}\setminus\mathcal{S}^\alpha$ such that $1+\theta^\alpha(\rho) = 0$. We shall denote by $\tilde\rho$ such eigenvalues when they exist.*

- $1+\theta^\alpha(\rho)\neq 0$*: the left and right eigenvectors associated to the zero eigenvalues of* $\underline{\mathbf{G}}_\rho^\alpha$ *are respectively orthogonal to the right and left eigenvectors associated to the non-zero eigenvalues. So, by letting* $\mathbf{V}_l$, $\mathbf{V}_r$ *be matrices containing in columns the respectively left and right eigenvectors of* $\underline{\mathbf{G}}_\rho^\alpha$ *associated with the zero eigenvalues, we have* $\mathbf{V}_l^\mathsf{T}\mathbf{c}=\mathbf{0}$ *and* $\mathbf{1}_K^\mathsf{T}\mathbf{V}_r=\mathbf{0}$ *since* $1+\theta^\alpha(\rho)\neq 0$*. It is thus immediate that* $(\mathbf{V}_l,\mathbf{V}_r)$ *is also a pair of eigenvectors (with multiplicity) of* $\mathbf{I}_K+e_{21}^\alpha(\rho)\left(\mathcal{D}\left(\mathbf{c}\right)-\mathbf{c}\mathbf{c}^\mathsf{T}\right)\mathbf{M}\left(\mathbf{I}_K-\mathbf{c}\mathbf{1}_K^T\right)$ *associated to the zero eigenvalues.*

As we show in Section 6.5, for $1+\theta^\alpha(\tilde\rho)=0$, the eigenvectors associated to the aforementioned isolated eigenvalues $\tilde\rho$ will not contain information about the classes. This case is thus of no interest for clustering. It is nevertheless important from a practical viewpoint to note that, even in the absence of communities, spurious isolated eigenvalues may be found that may deceive the experimenter in suggesting the presence of node clusters. From now on, we will only consider the isolated eigenvalues ρ for which $1+\theta^\alpha(\rho)\neq 0$.

We now have all the ingredients to determine the conditions under which we may have eigenvalues of $\mathbf{L}_\alpha$ which isolate from $\mathcal{S}^\alpha$. Let l be a non zero eigenvalue of $\mathbf{G}_\rho^\alpha=(\mathcal{D}\left(\mathbf{c}\right)-\mathbf{c}\mathbf{c}^\mathsf{T})\mathbf{M}(\mathbf{I}_K-\mathbf{c}\mathbf{1}_K^T)$. Since $\det((\mathcal{D}(\mathbf{c})-\mathbf{c}\mathbf{c}^\mathsf{T})\mathbf{M}(\mathbf{I}_K-\mathbf{c}\mathbf{1}_K^\mathsf{T}))=\det((\mathbf{I}_K-\mathbf{c}\mathbf{1}_K^\mathsf{T})(\mathcal{D}(\mathbf{c})-\mathbf{c}\mathbf{c}^\mathsf{T})\mathbf{M})=\det((\mathcal{D}(\mathbf{c})-\mathbf{c}\mathbf{c}^\mathsf{T})\mathbf{M})$, l is also a non zero eigenvalue of $\bar{\mathbf{M}}=(\mathcal{D}(\mathbf{c})-\mathbf{c}\mathbf{c}^\mathsf{T})\mathbf{M}$. For each isolated eigenvalue ρ of $\mathbf{L}_\alpha$ we have a one-to-one mapping with a non zero eigenvalue l of $\bar{\mathbf{M}}$ such that $l=-\frac{1}{E_2^\alpha(\rho)}$. Hence, to show the existence of isolated eigenvalues of $\mathbf{L}_\alpha$, we need to solve for $\rho\in\mathbb{R}\setminus\mathcal{S}^\alpha$, $l=-\frac{1}{E_2^\alpha(\rho)}$ for each non zero eigenvalue l of $\bar{\mathbf{M}}$. Precisely, let us write $\mathcal{S}^\alpha=\bigcup_{m=1}^M[S_{m,-}^\alpha,S_{m,+}^\alpha]$ with $S_{1,-}^\alpha\leq S_{1,+}^\alpha<S_{2,-}^\alpha\leq\ldots<S_{M,+}^\alpha$ and define $S_{0,+}=-\infty$ and $S_{M+1,-}=+\infty$. Then, recalling that the Stieltjes transform of a real supported measure is necessarily increasing on $\mathbb{R}$, there exist isolated eigenvalues of $\mathbf{L}^\alpha$ in $(S_{m,+}^\alpha,S_{m+1,-}^\alpha)$, $m\in\{0,\ldots,M\}$, for all large n almost surely, if and only if there exists eigenvalues ℓ of $\bar{\mathbf{M}}$ such that

$$\lim_{x\downarrow S_{m,+}^\alpha}E_2^\alpha(x)<-\ell^{-1}<\lim_{x\uparrow S_{m+1,-}^\alpha}E_2^\alpha(x). \tag{24}$$

In particular, when $\mathcal{S}^\alpha=[S_-^\alpha,S_+^\alpha]$ is composed of a single connected component (as when $\mathcal{S}^\alpha$ is the support of the semi-circle law as well as most cases met in practice), then isolated eigenvalues of $\mathbf{L}^\alpha$ may only be found beyond S_+^α if $\ell>\lim_{x\downarrow S_+^\alpha}-\frac{1}{E_2^\alpha(x)}$ ($l>0$) or below S_-^α if $\ell<\lim_{x\uparrow S_-^\alpha}-\frac{1}{E_2^\alpha(x)}$ ($l<0$), for some non-zero eigenvalue ℓ of $\bar{\mathbf{M}}$. From the asymptotic spectrum of $\mathbf{L}_\alpha$, $S_-^\alpha=-S_+^\alpha$ as one can show that for any $z\in\mathbb{R}\setminus\mathcal{S}^\alpha$, $E_2^\alpha(-z)=-E_2^\alpha(z)$ so that both previous conditions reduce to $\mid\ell\mid>\lim_{x\downarrow S_+^\alpha}-\frac{1}{E_2^\alpha(x)}$. This proves Theorem 6.

The next section is advocated to the study of the eigenvectors associated to isolated eigenvalues of $\mathbf{L}_\alpha$.

6.4 Informative eigenvectors

In this section, in order to fully characterize the performances of Algorithm 2, we study in depth the normalized eigenvectors $\bar{\mathbf{v}}_i^\alpha$ used for the classification in the algorithm (step 3 of Algorithm2). We consider here the eigenvectors corresponding to the eigenvalues for which $1+\theta^\alpha(\rho)\neq 0$ (when $1+\theta^\alpha(\rho)=0$, the corresponding eigenvectors do not contain any structural information about the classes; this case is treated in Section 6.5). For technical

reasons, we restrict ourselves here to those eigenpairs $(\lambda_i,\bar{\mathbf{v}}_i^\alpha)$'s for which there exists no $\lambda_j\neq\lambda_i$ such that, if $\lambda_i\to\rho$, $\lambda_j\to\rho$.

We recall that we may write $\bar{\mathbf{v}}_i^\alpha$ [14] as the "noisy plateaus" vector

$$\bar{\mathbf{v}}_i^\alpha=\sum_{a=1}^K\nu_i^a\frac{\mathbf{j}_a}{\sqrt{n_a}}+\sqrt{\sigma_{ii}^a}\mathbf{w}_i^a \tag{25}$$

where $\mathbf{w}_i^a\in\mathbb{R}^n$ is a random vector orthogonal to $\mathbf{j}_a$, of norm $\sqrt{n_a}$ and supported on the indices of $\mathcal{C}_a$ and

$$\nu_i^a=\frac{1}{\sqrt{n_a}}\left(\bar{\mathbf{v}}_i^\alpha\right)^\mathsf{T}\mathbf{j}_a=\frac{1}{\sqrt{n_a}}\frac{(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{j}_a}{\sqrt{(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{2(\alpha-1)}\mathbf{u}_i^\alpha}} \tag{26}$$

$$\sigma_{ij}^a=\frac{(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{\alpha-1}\mathcal{D}_a\mathbf{D}^{\alpha-1}\mathbf{u}_j^\alpha}{\sqrt{(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{2(\alpha-1)}\mathbf{u}_i}\sqrt{(\mathbf{u}_j^\alpha)^\mathsf{T}\mathbf{D}^{2(\alpha-1)}\mathbf{u}_j^\alpha}}-\nu_i^a\nu_j^a \tag{27}$$

with $\mathcal{D}_a=\mathcal{D}(\mathbf{j}_a)$.

- We estimate the ν_i^a's by obtaining an estimator of the $K\times K$ matrix

$$\frac{1}{n}\frac{\mathbf{J}^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{u}_i^\alpha(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{J}}{(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{2(\alpha-1)}\mathbf{u}_i^\alpha},$$

  the diagonal entries of which allow to estimate $|\nu_i^a|$ while the off-diagonal entries are used to decide on the signs of the ν_i^a's (up to a convention in the sign of $\mathbf{u}_i^\alpha$).

- Similarly, we first estimate the more involved object

$$\frac{1}{n}\frac{\mathbf{J}^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{u}_i^\alpha(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{\alpha-1}\mathcal{D}_a\mathbf{D}^{\alpha-1}\mathbf{u}_j^\alpha(\mathbf{u}_j^\alpha)^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{J}}{\left((\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{2(\alpha-1)}\mathbf{u}_i^\alpha\right)\left((\mathbf{u}_j^\alpha)^\mathsf{T}\mathbf{D}^{2(\alpha-1)}\mathbf{u}_j^\alpha\right)}$$

  from which $\frac{(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{\alpha-1}\mathcal{D}_a\mathbf{D}^{\alpha-1}\mathbf{u}_j^\alpha}{\sqrt{(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{2(\alpha-1)}\mathbf{u}_i^\alpha}\sqrt{(\mathbf{u}_j^\alpha)^\mathsf{T}\mathbf{D}^{2(\alpha-1)}\mathbf{u}_j^\alpha}}$ is retrieved by dividing any entry e,f of the former quantity by non-vanishing quantities $\nu_i^e\nu_i^f$. For the eigenvectors $\mathbf{u}_i^\alpha$ used for clustering, there is always at least one index f such that ν_i^f is non zero (otherwise, this eigenvector is of no use for clustering).

6.4.1 Evaluation of the class means ν_i^a's

The estimation of the ν_i^a's requires the evaluation of $\frac{1}{n}\frac{\mathbf{J}^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{u}_i^\alpha(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{J}}{(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{2(\alpha-1)}\mathbf{u}_i^\alpha}$ for $\mathbf{u}_i^\alpha$ eigenvector associated to a limiting isolated eigenvalue ρ with unit multiplicity of $\mathbf{L}_\alpha$. By residue calculus, we have that

$$\frac{1}{n}\mathbf{J}^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{u}_i^\alpha(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{J}=-\frac{1}{2\pi i}\oint_{\Gamma_\rho}\frac{1}{n}\mathbf{J}^\mathsf{T}\mathbf{D}^{\alpha-1}\left(\mathbf{L}_\alpha-z\mathbf{I}_n\right)^{-1}\mathbf{D}^{\alpha-1}\mathbf{J}\mathrm{d}z \tag{28}$$

14. Recall that the graph nodes were assumed labeled by class, and thus the entries of $\bar{\mathbf{v}}_i^\alpha$ are similarly sorted by class.

for large n almost surely, where Γ_ρ is a complex (positively oriented) contour circling around the limiting eigenvalue ρ only. As from Theorem 2, $\mathbf{L}_\alpha = \mathbf{D}_q^{-\alpha}\frac{\mathbf{X}}{\sqrt{n}}\mathbf{D}_q^{-\alpha} + \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\mathsf{T} + o(1)$, we apply the Woodburry identity to the inverse in the previous integrand and we get

$$\frac{1}{n}\mathbf{J}^\mathsf{T}\mathbf{D}^{\alpha-1}\left(\mathbf{L}_\alpha - z\mathbf{I}_n\right)^{-1}\mathbf{D}^{\alpha-1}\mathbf{J} = \frac{1}{n}\mathbf{J}^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{Q}_z^\alpha\mathbf{D}^{\alpha-1}\mathbf{J} + \frac{1}{n}\mathbf{J}^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{Q}_z^\alpha\mathbf{U}\mathbf{\Lambda}\left(\mathbf{I}_{K+1} + \mathbf{U}^\mathsf{T}\mathbf{Q}_z^\alpha\mathbf{U}\mathbf{\Lambda}\right)^{-1}\mathbf{U}^\mathsf{T}\mathbf{Q}_z^\alpha\mathbf{D}^{\alpha-1}\mathbf{J} + o(1).$$

The first right-hand side has asymptotically no residue when we integrate over the contour Γ_ρ (as per Proposition 18 there is no eigenvalues of $\bar{\mathbf{X}}$ in Γ_ρ for all large n almost surely). We are then left with the second right-most term. Using the block structure used in Section 6.3, we may write

$$\left(\mathbf{I}_{K+1} + \mathbf{U}^\mathsf{T}\mathbf{Q}_z^\alpha\mathbf{U}\mathbf{\Lambda}\right)^{-1} \xrightarrow{\text{a.s.}} \begin{pmatrix} \mathbf{I}_K + e_{21}^\alpha(z)(\mathcal{D}(\mathbf{c}) - \mathbf{c}\mathbf{c}^\mathsf{T})\mathbf{M}(\mathbf{I}_K - \mathbf{c}\mathbf{1}_K^\mathsf{T}) - \left(1 + \frac{z}{m_\mu}e_{10}^\alpha(z)\right)\mathbf{c}\mathbf{1}_K^\mathsf{T} & -e_{21}^\alpha(z)\mathbf{c} \\ \frac{z}{m_\mu^2}\left(v_\mu + ze_{0;-1}^\alpha(z)\right)\mathbf{1}_K^\mathsf{T} & -z\frac{e_{10}^\alpha(z)}{m_\mu} \end{pmatrix}^{-1}.$$

Let us write $\gamma(z) = \frac{z}{m_\mu^2}\left(v_\mu + ze_{0;-1}^\alpha(z)\right)$. We can now use a block inversion formula to write

$$\left(\mathbf{I}_{K+1} + \mathbf{U}^T\mathbf{Q}_z^\alpha\mathbf{U}\mathbf{\Lambda}\right)^{-1} \xrightarrow{\text{a.s.}} \begin{pmatrix} (\underline{\mathbf{G}}_z^\alpha)^{-1} & -\frac{e_{10}^\alpha(z)\left[\underline{\mathbf{G}}_z^\alpha - \frac{\gamma(z)m_\mu e_{21}^\alpha(z)}{ze_{10}^\alpha(z)}\mathbf{c}\mathbf{1}_K^\mathsf{T}\right]^{-1}\mathbf{c}}{-\frac{ze_{21}^\alpha(z)}{m_\mu} + \gamma(z)e_{10}^\alpha(z)\mathbf{1}_K^\mathsf{T}\left[\underline{\mathbf{G}}_z^\alpha - \frac{\gamma(z)m_\mu e_{21}^\alpha(z)}{ze_{10}^\alpha(z)}\mathbf{c}\mathbf{1}_K^\mathsf{T}\right]^{-1}\mathbf{c}} \\ \frac{\gamma(z)m_\mu}{ze_{21}^\alpha(z)}\mathbf{1}_K^\mathsf{T}(\underline{\mathbf{G}}_z^\alpha)^{-1} & \frac{1}{-\frac{ze_{21}^\alpha(z)}{m_\mu} + \gamma(z)e_{10}^\alpha(z)\mathbf{1}_K^\mathsf{T}\left[\underline{\mathbf{G}}_z^\alpha - \frac{\gamma(z)m_\mu e_{21}^\alpha(z)}{ze_{10}^\alpha(z)}\mathbf{c}\mathbf{1}_K^\mathsf{T}\right]^{-1}\mathbf{c}} \end{pmatrix} \tag{29}$$

with $\underline{\mathbf{G}}_z^\alpha = \mathbf{I}_K + e_{21}^\alpha(z)\left(\mathcal{D}(\mathbf{c}) - \mathbf{c}\mathbf{c}^\mathsf{T}\right)\mathbf{M}\left(\mathbf{I}_K - \mathbf{c}\mathbf{1}_K^T\right) + \theta^\alpha(z)\mathbf{c}\mathbf{1}_K^T$. The entries of the previous matrix seem to be cumbersome but as we will see, the residue calculus will greatly simplify. In fact, we have that $\mathbf{1}_K^T\underline{\mathbf{G}}_z^\alpha = (1+\theta^\alpha(z))\mathbf{1}_K^T$ so that $\mathbf{1}_K^T(\underline{\mathbf{G}}_z^\alpha)^{-1} = \frac{1}{1+\theta^\alpha(z)}\mathbf{1}_K^T$ which is well defined since we are considering the case $1 + \theta^\alpha(z) \neq 0$. Similarly, we have that

$$\left[\underline{\mathbf{G}}_z^\alpha - \frac{\gamma(z)m_\mu e_{10}^\alpha(z)}{ze_{21}^\alpha(z)}\mathbf{c}\mathbf{1}_K^\mathsf{T}\right]\mathbf{c} = \left(-z\frac{e_{10}^\alpha(z)}{m_\mu}\right)\mathbf{c}$$

meaning that $\left[\underline{\mathbf{G}}_z^\alpha - \frac{\gamma(z)m_\mu e_{21}^\alpha(z)}{ze_{10}^\alpha(z)}\mathbf{c}\mathbf{1}_K^\mathsf{T}\right]^{-1}\mathbf{c} = -\frac{m_\mu}{ze_{10}^\alpha(z)}\mathbf{c}$. So finally, the terms $(1,2)$, $(2,1)$ and $(2,2)$ of $\left(\mathbf{I}_{K+1} + \mathbf{U}^T\mathbf{Q}_z^\alpha\mathbf{U}\mathbf{\Lambda}\right)^{-1}$ do no longer depend on $(\underline{\mathbf{G}}_z^\alpha)^{-1}$ and thus do not have poles in the contour Γ_ρ. We can then write

$$\left(\mathbf{I}_{K+1} + \mathbf{U}^T\mathbf{Q}_z^\alpha\mathbf{U}\mathbf{\Lambda}\right)^{-1} = \begin{pmatrix} (\underline{\mathbf{G}}_z^\alpha)^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \mathbf{R}_1(z)$$

with $\mathbf{R}_1(z)$ having no residue in the contour Γ_ρ. Thus, to perform the contour integration of the integrand in (28) around Γ_ρ, we just need to evaluate the top-left entries of $\mathbf{J}^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{Q}_z^\alpha\mathbf{U}\mathbf{\Lambda}$ and $\mathbf{U}^\mathsf{T}\mathbf{Q}_z^\alpha\mathbf{D}^{\alpha-1}\mathbf{J}$. Those are easily retrieved from the calculus in Section 6.3.

We have in particular $(\frac{1}{\sqrt{n}}\mathbf{J}^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{Q}_z^\alpha\mathbf{U}\mathbf{\Lambda})_{11} \xrightarrow{\text{a.s.}} e_{00}^\alpha(z)(\mathcal{D}(\mathbf{c}) - \mathbf{c}\mathbf{c}^\mathsf{T})\mathbf{M}(\mathbf{I}_K - \mathbf{c}\mathbf{1}_K^\mathsf{T}) - \beta^\alpha(z)\mathbf{c}\mathbf{1}_K^\mathsf{T}$ where $\beta^\alpha(z) = \frac{1}{m_\mu}\left[\int t^{2\alpha-1}\mu(\mathrm{d}t) + e_{-1;-1}^\alpha(z)\right]$ and similarly $(\mathbf{U}^\mathsf{T}\mathbf{Q}_z^\alpha\mathbf{D}^{\alpha-1}\mathbf{J})_{11} \xrightarrow{\text{a.s.}} e_{00}^\alpha(z)\mathcal{D}(\mathbf{c})$, so that finally

$$\frac{1}{n}\mathbf{J}^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{u}_i^\alpha(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{J} \xrightarrow{\text{a.s.}} -\frac{1}{2\pi i}\oint_{\Gamma_\rho}\left[\left(e_{00}^\alpha(z)(\mathcal{D}(\mathbf{c}) - \mathbf{c}\mathbf{c}^\mathsf{T})\mathbf{M}(\mathbf{I}_K - \mathbf{c}\mathbf{1}_K^\mathsf{T}) - \beta^\alpha(z)\mathbf{c}\mathbf{1}_K^\mathsf{T}\right)(\underline{\mathbf{G}}_z^\alpha)^{-1} \times e_{00}^\alpha(z)\mathcal{D}(\mathbf{c}) + \mathbf{R}_2(z)\mathrm{d}z\right]$$

where $\mathbf{R}_2(z)$ is a matrix having no residue in the considered contour. Now, we are ready to compute the integral. From the Cauchy integral formula,

$$\frac{1}{n}\mathbf{J}^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{u}_i^\alpha(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{J} \xrightarrow{\text{a.s.}} \lim_{z\to\rho}(z-\rho)\left[e_{00}^\alpha(z)(\mathcal{D}(\mathbf{c}) - \mathbf{c}\mathbf{c}^\mathsf{T})\mathbf{M}(\mathbf{I}_K - \mathbf{c}\mathbf{1}_K^\mathsf{T}) - \beta^\alpha(z)\mathbf{c}\mathbf{1}_K^\mathsf{T}\right](\underline{\mathbf{G}}_z^\alpha)^{-1} \times e_{00}^\alpha(z)\mathcal{D}(\mathbf{c}).$$

By writing $\underline{\mathbf{G}}_z^\alpha = \rho_z\mathbf{v}_{r,z}\mathbf{v}_{l,z}^T + \tilde{\mathbf{V}}_{r,z}\tilde{\mathbf{\Sigma}}_z\tilde{\mathbf{v}}_{l,z}^T$ where $\mathbf{v}_{r,z}$ and $\mathbf{v}_{l,z}$ are respectively right and left eigenvectors associated with the vanishing eigenvalue ρ_z of $\underline{\mathbf{G}}_z^\alpha$ when $z \to \rho$; $\tilde{\mathbf{V}}_{r,z} \in \mathbb{R}^{n\times\eta_\rho}$ and $\tilde{\mathbf{V}}_{l,z}\mathbb{R}^{n\times\eta_\rho}$ are respectively sets of right and left eigenspaces associated with non vanishing eigenvalues, we then have

$$\lim_{z\to\rho}(z-\rho)(\underline{\mathbf{G}}_z^\alpha)^{-1} \overset{(1)}{=} \lim_{z\to\rho}(z-\rho)\frac{\mathbf{v}_{r,z}\mathbf{v}_{l,z}^T}{\rho_z'}$$

where we have used the l'Hopital rule and the fact that the non vanishing eigenvalue part of $\underline{\mathbf{G}}_z^\alpha$ will produce zero in the limit $z \to \rho$. Using $\rho_z = \mathbf{v}_{l,z}^\mathsf{T}\underline{\mathbf{G}}_z^\alpha\mathbf{v}_{r,z}$, we obtain

$$\frac{1}{n}\mathbf{J}^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{u}_i^\alpha(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{J} \xrightarrow{\text{a.s.}} \left[e_{00}^\alpha(\rho)(\mathcal{D}(\mathbf{c}) - \mathbf{c}\mathbf{c}^\mathsf{T})\mathbf{M}(\mathbf{I}_K - \mathbf{c}\mathbf{1}_K^\mathsf{T}) - \beta^\alpha(\rho)\mathbf{c}\mathbf{1}_K^\mathsf{T}\right]\frac{\mathbf{v}_{r,\rho}\mathbf{v}_{l,\rho}^\mathsf{T}}{\left(\mathbf{v}_{l,z}^\mathsf{T}\underline{\mathbf{G}}_z^\alpha\mathbf{v}_{r,z}\right)'_{z=\rho}} \times e_{00}^\alpha(\rho)\mathcal{D}(\mathbf{c}).$$

Since $(\mathbf{v}_{l,\rho})^\mathsf{T}\underline{\mathbf{G}}_\rho^\alpha = \underline{\mathbf{G}}_\rho^\alpha\mathbf{v}_{r,\rho} = 0$,

$$\begin{aligned}\left((\mathbf{v}_{l,z})^T\underline{\mathbf{G}}_z^\alpha\mathbf{v}_{r,z}\right)'_{z=\rho} &= \left((\mathbf{v}_{l,z})^T\right)'_{z=\rho}\underline{\mathbf{G}}_\rho^\alpha\mathbf{v}_{r,\rho} + (\mathbf{v}_{l,\rho})^T\left(\underline{\mathbf{G}}_z^\alpha\right)'_{z=\rho}\mathbf{v}_{r,\rho} + (\mathbf{v}_{l,\rho})^T\underline{\mathbf{G}}_\rho^\alpha(\mathbf{v}_{r,z})'_{z=\rho} \\ &= (\mathbf{v}_{l,\rho})^T\left(\underline{\mathbf{G}}_z^\alpha\right)'_{z=\rho}\mathbf{v}_{r,\rho} \\ &= (e_{21}^\alpha(\rho))'(\mathbf{v}_{l,\rho})^T\left(\mathcal{D}(\mathbf{c}) - \mathbf{c}\mathbf{c}^\mathsf{T}\right)\mathbf{M}\left(\mathbf{I}_K - \mathbf{c}\mathbf{1}_K^T\right)\mathbf{v}_{r,\rho}\end{aligned}$$

where the subscript $'$ denotes the first derivative with respect to z. Using the fact that $\mathbf{v}_{r,\rho}$ is orthogonal to $\mathbf{1}_K^T$, and $(\mathbf{v}_{r,\rho}, \mathbf{v}_{l,\rho})$ is also a pair of eigenvectors of $\left(\mathcal{D}(\mathbf{c}) - \mathbf{c}\mathbf{c}^\mathsf{T}\right)\mathbf{M}\left(\mathbf{I}_K - \mathbf{c}\mathbf{1}_K^T\right)$ associated with eigenvalue $-\frac{1}{e_{21}^\alpha(\rho)}$, we get

$$\frac{1}{n}\mathbf{J}^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{u}_i^\alpha(\mathbf{u}_i\alpha)^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{J} \xrightarrow{\text{a.s.}} \frac{(e_{00}^\alpha(\rho))^2}{e_{21}^\alpha(\rho)'}\frac{\mathbf{v}_{r,\rho}(\mathbf{v}_{l,\rho})^T}{\mathbf{v}_{l,\rho}^T\mathbf{v}_{r,\rho}}\mathcal{D}(\mathbf{c}). \tag{30}$$

By introducing $\mathbf{v}_\rho = \mathcal{D}(\mathbf{c})^{\frac{1}{2}}\mathbf{v}_{l,\rho} = \mathcal{D}(\mathbf{c})^{-\frac{1}{2}}\mathbf{v}_{r,\rho}$ eigenvector of the symmetric matrix $\mathcal{D}(\mathbf{c})^{\frac{1}{2}}\left(\mathbf{I}_K - \mathbf{1}_K\mathbf{c}^\mathsf{T}\right)\mathbf{M}\left(\mathbf{I}_K - \mathbf{c}\mathbf{1}_K^T\right)\mathcal{D}(\mathbf{c})^{\frac{1}{2}}$, we obtain the final result

$$\frac{1}{n}\mathbf{J}^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{u}_i^\alpha(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{J} \xrightarrow{\text{a.s.}} \frac{\left(e_{00}^\alpha(\rho)\right)^2}{e_{21}^\alpha(\rho)'}\mathcal{D}\left(\mathbf{c}\right)^{1/2}\mathbf{v}_\rho(\mathbf{v}_\rho)^T\mathcal{D}\left(\mathbf{c}\right)^{1/2}. \tag{31}$$

Next, we need to estimate the denominator term $(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{2(\alpha-1)}\mathbf{u}_i^\alpha$ of $\frac{1}{n}\frac{\mathbf{J}^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{u}_i^\alpha(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{J}}{(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{2(\alpha-1)}\mathbf{u}_i^\alpha}$ of ν_i^α.For $\mathbf{u}_i^\alpha$ an eigenvector of $\mathbf{L}_\alpha$ associated to an isolated eigenvalue converging to ρ asymptotically, we have

$$\begin{aligned}(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{2(\alpha-1)}\mathbf{u}_i^\alpha &= \operatorname{tr}(\mathbf{u}_i^\alpha(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{2(\alpha-1)}) \\ &= \operatorname{tr}\left(-\frac{1}{2\pi i}\oint_{\Gamma_\rho}\left(\mathbf{L}_\alpha - z\mathbf{I}_n\right)\mathbf{D}^{2(\alpha-1)}\mathrm{d}z\right).\end{aligned}$$

As in the previous section, by applying Woodburry idendity, this is equivalent to evaluating

$$\operatorname{tr}\left(-\frac{1}{2\pi i}\oint_{\Gamma_\rho}\left[\mathbf{U}^\mathsf{T}\mathbf{Q}_z^\alpha\mathbf{D}^{2(\alpha-1)}\mathbf{Q}_z^\alpha\mathbf{U}\mathbf{\Lambda}\begin{pmatrix}(\underline{\mathbf{G}}_z^\alpha)^{-1} & 0\\ 0 & 0\end{pmatrix} + \mathbf{R}_3(z)\right]\mathrm{d}z\right),$$

where $\mathbf{R}_3(z)$ is a matrix having no residue in the considered contour.

Again here, we just need the top left entry of $\mathbf{U}^\mathsf{T}\mathbf{Q}_z^\alpha\mathbf{D}^{2(\alpha-1)}\mathbf{Q}_z^\alpha\mathbf{U}\mathbf{\Lambda}$ which is given from Theorem 2 by

$$(\mathbf{U}^\mathsf{T}\mathbf{Q}_z^\alpha\mathbf{D}^{2(\alpha-1)}\mathbf{Q}_z^\alpha\mathbf{U}\mathbf{\Lambda})_{11} = \underbrace{\frac{1}{n}\mathbf{J}^\mathsf{T}\mathbf{D}^{1-\alpha}\mathbf{Q}_z^\alpha\mathbf{D}^{2(\alpha-1)}\mathbf{Q}_z^\alpha\mathbf{D}^{1-\alpha}\mathbf{J}(\mathbf{I}_K - \mathbf{1}_K\mathbf{c}^\mathsf{T})\mathbf{M}(\mathbf{I}_K - \mathbf{c}\mathbf{1}_K^\mathsf{T})}_{(\mathbf{I})} \tag{32}$$

$$\underbrace{-\frac{1}{\sqrt{n}(\mathbf{q}^\mathsf{T}\mathbf{1}_n)}\mathbf{D}^{1-\alpha}\mathbf{Q}_z^\alpha\mathbf{D}^{2(\alpha-1)}\mathbf{Q}_z^\alpha\mathbf{D}^{-\alpha}\mathbf{X}\mathbf{1}_n\mathbf{1}_K^\mathsf{T}}_{(\mathbf{II})}. \tag{33}$$

We can get rid of the term (**II**) since after residue calculus, we will get (similar to Equation (30)) $\mathbf{1}_K^\mathsf{T}\mathbf{v}_{r,\rho} = 0$ which cancels out the whole term. Let us now concentrate on the term (**I**). At this point, we need to introduce the following result which, for any deterministic vectors of bounded Euclidean norm $\mathbf{a}$, $\mathbf{b}$ and any deterministic diagonal matrix $\mathbf{\Xi}$, approximates the random quantity $\mathbf{a}^\mathsf{T}\mathbf{Q}_{z_1}^\alpha\mathbf{\Xi}\mathbf{Q}_{z_2}^\alpha\mathbf{b}$ by a deterministic equivalent.

Lemma 21 (Second deterministic equivalents) *For all $z \in \mathbb{C}\backslash\mathcal{S}^\alpha$, we have the following deterministic equivalent*

$$\mathbf{Q}_{z_1}^\alpha\mathbf{\Xi}\mathbf{Q}_{z_2}^\alpha \leftrightarrow \bar{\mathbf{Q}}_{z_1}^\alpha\mathbf{\Xi}\bar{\mathbf{Q}}_{z_2}^\alpha + \bar{\mathbf{Q}}_{z_1}^\alpha\mathcal{D}\left[\left(\mathbf{I}_n - \mathbf{\Upsilon}_{z_1,z_2}\right)^{-1}\mathbf{\Upsilon}_{z_1,z_2}\operatorname{diag}\left(\mathbf{\Xi}\right)\right]\bar{\mathbf{Q}}_{z_2}^\alpha$$

where $\mathbf{\Xi}$ is any diagonal matrix, $\bar{\mathbf{Q}}_z^\alpha$ is given in Lemma 16 and

$$\Upsilon_{z_1,z_2}(i,j) = \frac{1}{n}\frac{q_i^{1-2\alpha}q_j^{1-2\alpha}\left(1-q_iq_j\right)}{\left(-z_1 - e_{11}^\alpha(z_1)q_i^{1-2\alpha} + e_{21}^\alpha(z_1)q_i^{2-2\alpha}\right)\left(-z_2 - e_{11}^\alpha(z_2)q_j^{1-2\alpha} + e_{21}^\alpha(z_2)q_j^{2-2\alpha}\right)}.$$

The equivalence relation $\leftrightarrow$ is as defined in Lemma 16.

Thanks to Lemma 21 (proof provided in Appendix D), a deterministic approximation of the term (**I**) in Equation (32) can be obtained. We get in particular

$$\frac{1}{n}\mathbf{J}^\mathsf{T}\mathbf{D}^{1-\alpha}\mathbf{Q}_z^\alpha\mathbf{D}^{2(\alpha-1)}\mathbf{Q}_z^\alpha\mathbf{D}^{1-\alpha}\mathbf{J} = \frac{1}{n}\mathbf{J}^\mathsf{T}\mathbf{D}^{1-\alpha}\bar{\mathbf{Q}}_z^\alpha\mathbf{D}^{2(\alpha-1)}\bar{\mathbf{Q}}_z^\alpha\mathbf{D}^{1-\alpha}\mathbf{J} \tag{34}$$

$$+\frac{1}{n}\mathbf{J}^\mathsf{T}\mathbf{D}^{1-\alpha}\bar{\mathbf{Q}}_z^\alpha\mathcal{D}\left[\left(\mathbf{I}_n - \mathbf{\Upsilon}_{z,z}\right)^{-1}\mathbf{\Upsilon}_{z,z}\mathbf{d}^\alpha\right]\bar{\mathbf{Q}}_z^\alpha\mathbf{D}^{1-\alpha}\mathbf{J} \tag{35}$$

where $\mathbf{d}^\alpha = \{q_i^{2(\alpha-1)}\}_{i=1}^n$ and $\mathbf{\Upsilon}_{z_1,z_2}$ was defined in Lemma 21. Using similar argument as in Equation (23), we can easily show that the first right hand side term of (34) converges almost surely to $e_{00;2}^\alpha\mathcal{D}(\mathbf{c})$. It then remains to estimate the second right-most term of (34). $\mathbf{\Upsilon}_{z_1,z_2}$ (as defined in Lemma 21) may be written as the sum of two rank-one matrices

$$\mathbf{\Upsilon}_{z_1,z_2} = \frac{1}{n}\left(\mathbf{a}_{z_1}\mathbf{a}_{z_2}^\mathsf{T} - \mathbf{b}_{z_1}\mathbf{b}_{z_2}^\mathsf{T}\right)$$

where $\mathbf{a}_z = \left\{\frac{q_j^{1-2\alpha}}{-z-q_j^{1-2\alpha}e_{11}^\alpha(z)+q_j^{2-2\alpha}e_{21}^\alpha(z)}\right\}_{j=1}^n$ and $\mathbf{b}_z = \left\{\frac{q_j^{2-2\alpha}}{-z-q_j^{1-2\alpha}e_{11}^\alpha(z)+q_j^{2-2\alpha}e_{21}^\alpha(z)}\right\}_{j=1}^n$.

The matrix $\mathbf{\Upsilon}_{z,z}$ can thus be further written $\mathbf{\Upsilon}_{z,z} = \frac{1}{n}\begin{pmatrix}\mathbf{a}_z & \mathbf{b}_z\end{pmatrix}\mathbf{I}_2\begin{pmatrix}\mathbf{a}_z^\mathsf{T}/n\\ -\mathbf{b}_z^\mathsf{T}/n\end{pmatrix}$. Using matrix inversion lemmas, we have

$$\left(\mathbf{I}_n - \mathbf{\Upsilon}_{z,z}\right)^{-1}\mathbf{\Upsilon}_{z,z}\mathbf{d}^\alpha = \begin{pmatrix}\mathbf{a}_z & \mathbf{b}_z\end{pmatrix}\begin{pmatrix}1-\frac{\mathbf{a}_z^\mathsf{T}\mathbf{a}_z}{n} & -\frac{\mathbf{a}_z^\mathsf{T}\mathbf{b}_z}{n}\\ \frac{\mathbf{b}_z^\mathsf{T}\mathbf{a}_z}{n} & 1+\frac{\mathbf{b}_z^\mathsf{T}\mathbf{b}_z}{n}\end{pmatrix}^{-1}\begin{pmatrix}\frac{\mathbf{a}_z^\mathsf{T}\mathbf{d}^\alpha}{n}\\ -\frac{\mathbf{b}_z^\mathsf{T}\mathbf{d}^\alpha}{n}\end{pmatrix}.$$

Using again the argument in Equation (23) , we can easily show that $\frac{\mathbf{a}_z^\mathsf{T}\mathbf{a}_z}{n}$, $\frac{\mathbf{a}_z^\mathsf{T}\mathbf{b}_z}{n}$, $\frac{\mathbf{b}_z^\mathsf{T}\mathbf{b}_z}{n}$, $\frac{\mathbf{a}_z^\mathsf{T}\mathbf{d}^\alpha}{n}$ and $\frac{\mathbf{b}_z^\mathsf{T}\mathbf{d}^\alpha}{n}$ converge for large n almost surely respectively to $e_{22;2}^\alpha(z)$, $e_{32;2}^\alpha(z)$, $e_{42;2}^\alpha(z)$, $e_{-1;0}^\alpha(z)$ and $e_{00}^\alpha(z)$ with $e_{ij;2}^\alpha$ defined in Equation (21). This given, we can show that

$$\frac{1}{n}\mathbf{J}^\mathsf{T}\mathbf{D}^{1-\alpha}\bar{\mathbf{Q}}_z^\alpha\mathcal{D}\left[\left(\mathbf{I}_n - \mathbf{\Upsilon}_{z,z}\right)^{-1}\mathbf{\Upsilon}_{z,z}\mathbf{d}^\alpha\right]\bar{\mathbf{Q}}_z^\alpha\mathbf{D}^{1-\alpha}\mathbf{J} \xrightarrow{\text{a.s.}} \chi^\alpha(z)\mathcal{D}(\mathbf{c})$$

with

$$\chi^\alpha(z) = \frac{\left[\left(1+e_{42;2}^\alpha(z)\right)e_{-1,0}^\alpha(z) - e_{32;2}^\alpha(z)e_{00}^\alpha(z)\right]e_{32;3}^\alpha(z) - \left[e_{22;2}^\alpha(z)e_{-1,0}^\alpha(z) + \left(1-e_{22;2}^\alpha(z)\right)e_{00}^\alpha(z)\right]e_{42;3}^\alpha(z)}{\left(1+e_{42;2}^\alpha(z)\right)\left(1-e_{22;2}^\alpha(z)\right) + \left[e_{32;2}^\alpha(z)\right]^2}.$$

We thus have

$$(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{2(\alpha-1)}\mathbf{u}_i^\alpha \xrightarrow{\text{a.s.}} \operatorname{tr}\left(\lim_{z\to\rho}\left(e_{00;2}^\alpha(z) + \chi^\alpha(z)\right)\left(\mathcal{D}(\mathbf{c}) - \mathbf{c}\mathbf{c}^\mathsf{T}\right)\mathbf{M}(\mathbf{I}_K - \mathbf{c}\mathbf{1}_K^\mathsf{T})(\underline{\mathbf{G}}_z^\alpha)^{-1}\right).$$

By applying l'Hopital rule to evaluate this limit as in the previous section, we obtain

$$(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{2(\alpha-1)}\mathbf{u}_i^\alpha \xrightarrow{\text{a.s.}} \frac{e_{00;2}(\rho) + \chi^\alpha(\rho)}{\left(e_{21}^\alpha(\rho)\right)'}.$$

Finally,

$$\frac{1}{n}\frac{\mathbf{J}^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{u}_i^\alpha(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{J}}{(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{2(\alpha-1)}\mathbf{u}_i^\alpha} \xrightarrow{\text{a.s.}} \frac{\left(e_{00}^\alpha(\rho)\right)^2}{e_{00;2}(\rho) + \chi^\alpha(\rho)}\mathcal{D}\left(\mathbf{c}\right)^{1/2}\mathbf{v}_\rho(\mathbf{v}_\rho)^T\mathcal{D}\left(\mathbf{c}\right)^{1/2}. \tag{36}$$

We recall that one goal of this section is to estimate $\nu_i^a = \frac{1}{\sqrt{n_a}} \frac{\mathbf{u}_i^\mathsf{T} \mathbf{D}^{\alpha-1} \mathbf{j}_a}{\sqrt{\mathbf{u}_i^\mathsf{T} \mathbf{D}^{2(\alpha-1)} \mathbf{u}_i}}$, the square of which is $\frac{n}{n_a} \left[\frac{1}{n} \frac{\mathcal{D}(\mathbf{c})^{-\frac{1}{2}} \mathbf{J}^\mathsf{T} \mathbf{D}^{\alpha-1} \mathbf{u}_i^\alpha (\mathbf{u}_i^\alpha)^\mathsf{T} \mathbf{D}^{\alpha-1} \mathbf{J} \mathcal{D}(\mathbf{c})^{-\frac{1}{2}}}{(\mathbf{u}_i^\alpha)^\mathsf{T} \mathbf{D}^{2(\alpha-1)} \mathbf{u}_i^\alpha} \right]_{aa}$. From Equation (36), the former quantity is easily retrieved and we have

$$|\nu_i^a|^2 = \frac{(e_{00}^\alpha(\rho_i))^2}{e_{00;2}^\alpha(\rho_i) + \chi^\alpha(\rho_i)} \left| v_a^i \right|^2. \tag{37}$$

This proves the following theorem giving the limit of the empirical class means ν_i^a's.

Theorem 22 (Means) *For each eigenpair $(\lambda(\bar{\mathbf{M}}), \mathbf{v})$ of $\mathcal{D}(\mathbf{c})^{\frac{1}{2}} \left(\mathbf{I}_K - \mathbf{1}_K \mathbf{c}^\mathsf{T} \right) \mathbf{M} \left(\mathbf{I}_K - \mathbf{c} \mathbf{1}_K^T \right) \mathcal{D}(\mathbf{c})^{\frac{1}{2}}$ of unit multiplicity, mapped to eigenpair $(\rho, \mathbf{u}_i^\alpha)$ of $\mathbf{L}_\alpha$ as defined in Corollary 6, under the conditions of Assumption 1 and for ν_i^a defined in (26), we have almost surely as $n \to \infty$, $\left| (\nu_i^a)^2 - (\nu_i^{a,\infty})^2 \right| \to 0$ where*

$$(\nu_i^{a,\infty})^2 \equiv \frac{\left[e_{00}^\alpha(\rho) \right]^2}{e_{00;2}^\alpha(\rho, \rho) + \chi^\alpha(\rho)} (v_a)^2$$

with
$\chi^\alpha(\rho) = \frac{\left[\left(1 + e_{42;2}^\alpha(\rho)\right) e_{-1,0}^\alpha(\rho) - e_{32;2}^\alpha(\rho) e_{00}^\alpha(\rho) \right] e_{32;3}^\alpha(\rho) - \left[e_{22;2}^\alpha(\rho) e_{-1,0}^\alpha(\rho) + \left(1 - e_{22;2}^\alpha(\rho)\right) e_{00}^\alpha(\rho) \right] e_{42;3}^\alpha(\rho)}{\left(1 + e_{42;2}^\alpha(\rho)\right)\left(1 - e_{22;2}^\alpha(\rho)\right) + \left[e_{32;2}^\alpha(\rho) \right]^2}$ *and v_a is the component a of $\mathbf{v}$.*

Using the definition of ν_a^i in (26) and of $\bar{\mathbf{v}}$, $\mathbf{\Pi}$ in Theorem 10, Theorem 10 unfolds easily since $\bar{\mathbf{v}}^\mathsf{T} \mathbf{\Pi} \bar{\mathbf{v}} = \sum_{a=1}^K (\nu_a^i)^2 = \frac{\left[e_{00}^\alpha(\rho) \right]^2}{e_{00;2}^\alpha(\rho,\rho) + \chi^\alpha(\rho)} (v_a)^2$.

6.4.2 Evaluation of the class covariances σ_{ij}^a's

We have shown at the beginning of this section that to estimate the σ_{ij}^a's, we need to evaluate the more involved object

$$\frac{1}{n} \frac{\mathbf{J}^\mathsf{T} \mathbf{D}^{\alpha-1} \mathbf{u}_i^\alpha (\mathbf{u}_i^\alpha)^\mathsf{T} \mathbf{D}^{\alpha-1} \mathcal{D}_a \mathbf{D}^{\alpha-1} \mathbf{u}_j^\alpha (\mathbf{u}_j^\alpha)^\mathsf{T} \mathbf{D}^{\alpha-1} \mathbf{J}}{\left((\mathbf{u}_i^\alpha)^\mathsf{T} \mathbf{D}^{2(\alpha-1)} \mathbf{u}_i^\alpha \right) \left((\mathbf{u}_j^\alpha)^\mathsf{T} \mathbf{D}^{2(\alpha-1)} \mathbf{u}_j^\alpha \right)}.$$

Similarly to what was done previously for the estimation of $\frac{1}{n} \frac{\mathbf{J}^\mathsf{T} \mathbf{D}^{\alpha-1} \mathbf{u}_i^\alpha (\mathbf{u}_i^\alpha)^\mathsf{T} \mathbf{D}^{\alpha-1} \mathbf{J}}{(\mathbf{u}_i^\alpha)^\mathsf{T} \mathbf{D}^{2(\alpha-1)} \mathbf{u}_i^\alpha}$, we need here to evaluate

$$\left(\frac{1}{2\pi i} \right)^2 \oint_{\Gamma_{\rho 1}} \oint_{\Gamma_{\rho 2}} \frac{1}{n} \mathbf{J}^\mathsf{T} \mathbf{D}^{\alpha-1} \left(\mathbf{L}_\alpha - z_1 \mathbf{I}_n \right)^{-1} \mathbf{D}^{\alpha-1} \mathbf{D}_a \mathbf{D}^{\alpha-1} \left(\mathbf{L}_\alpha - z_2 \mathbf{I}_n \right)^{-1} \mathbf{D}^{\alpha-1} \mathbf{J} \mathrm{d}z_1 \mathrm{d}z_2$$

where Γ_{ρ_1} and Γ_{ρ_2} are two positively oriented contours circling around some limiting isolated eigenvalues ρ_1 and ρ_2 respectively. We will use the same technique as in the proof of Theorem 22 to evaluate this integrand. Namely, by applying the Woodburry identity to each of the inverse in the integrand, we get

$$\left(\frac{1}{2\pi i} \right)^2 \oint_{\Gamma_{\rho 1}} \oint_{\Gamma_{\rho 2}} \frac{1}{n} \mathbf{J}^T \mathbf{D}^{\alpha-1} \mathbf{Q}_{z_1}^\alpha \mathbf{U} \mathbf{\Lambda} \left(\mathbf{I}_{K+1} + \mathbf{U}^T \mathbf{Q}_{z_1}^\alpha \mathbf{U} \mathbf{\Lambda} \right)^{-1} \mathbf{U}^T \mathbf{Q}_{z_1}^\alpha \mathbf{D}^{\alpha-1} \mathbf{D}_a \mathbf{D}^{\alpha-1} \mathbf{Q}_{z_2}^\alpha \mathbf{U}$$
$$\times \mathbf{\Lambda} \left(\mathbf{I}_{K+1} + \mathbf{U}^T \mathbf{Q}_{z_2}^\alpha \mathbf{U} \mathbf{\Lambda} \right)^{-1} \mathbf{U}^T \mathbf{Q}_{z_2}^\alpha \mathbf{D}^{\alpha-1} \mathbf{J} \mathrm{d}z_1 \mathrm{d}z_2$$

where we have used the fact that the cross-terms $\frac{1}{n} \mathbf{J}^T \mathbf{D}^{\alpha-1} \mathbf{Q}_{z_i}^\alpha \mathbf{D}^{\alpha-1} \mathbf{J}$, $i = 1, 2$ will vanish asymptotically as the latter do not have poles in the considered contours. By using the identity $\mathbf{\Lambda} \left(\mathbf{I}_{K+1} + \mathbf{U}^T \mathbf{Q}_{z_1}^\alpha \mathbf{U} \mathbf{\Lambda} \right)^{-1} \mathbf{U}^T = \left(\mathbf{I}_{K+1} + \mathbf{\Lambda} \mathbf{U}^T \mathbf{Q}_{z_1}^\alpha \mathbf{U} \right)^{-1} \mathbf{\Lambda} \mathbf{U}^T$, the previous integral writes

$$\left(\frac{1}{2\pi i} \right)^2 \oint_{\Gamma_{\rho 1}} \oint_{\Gamma_{\rho 2}} \frac{1}{n} \mathbf{J}^T \mathbf{D}^{\alpha-1} \mathbf{Q}_{z_1}^\alpha \mathbf{U} \mathbf{\Lambda} \left(\mathbf{I}_{K+1} + \mathbf{U}^T \mathbf{Q}_{z_1}^\alpha \mathbf{U} \mathbf{\Lambda} \right)^{-1} \mathbf{U}^T \mathbf{Q}_{z_1}^\alpha \mathbf{D}^{\alpha-1} \mathbf{D}_a \mathbf{D}^{\alpha-1} \mathbf{Q}_{z_2}^\alpha \mathbf{U}$$
$$\times \left(\mathbf{I}_{K+1} + \mathbf{\Lambda} \mathbf{U}^T \mathbf{Q}_{z_2}^\alpha \mathbf{U} \right)^{-1} \mathbf{\Lambda} \mathbf{U}^T \mathbf{Q}_{z_2}^\alpha \mathbf{D}^{\alpha-1} \mathbf{J} \mathrm{d}z_1 \mathrm{d}z_2$$

Most of those quantities have been evaluated in the evaluation of the ν_i^a's. We thus obtain

$$\left(\frac{1}{2\pi i} \right)^2 \oint_{\Gamma_{\rho 1}} \oint_{\Gamma_{\rho 2}} \left[\frac{1}{n} \mathbf{J}^T \mathbf{D}^{\alpha-1} \mathbf{Q}_{z_1}^\alpha \mathbf{U} \mathbf{\Lambda} \begin{pmatrix} (\underline{\mathbf{G}}_{z_1}^\alpha)^{-1} & 0 \\ 0 & 0 \end{pmatrix} \mathbf{U}^T \mathbf{Q}_{z_1}^\alpha \mathbf{D}^{\alpha-1} \mathbf{D}_a \mathbf{D}^{\alpha-1} \mathbf{Q}_{z_2}^\alpha \mathbf{U} \right.$$
$$\left. \times \begin{pmatrix} \left((\underline{\mathbf{G}}_{z_2}^\alpha)^{-1} \right)^T & 0 \\ 0 & 0 \end{pmatrix} \mathbf{\Lambda} \mathbf{U}^T \mathbf{Q}_{z_2}^\alpha \mathbf{D}^{\alpha-1} \mathbf{J} + \mathbf{R}_4(z_1, z_2) \right] \mathrm{d}z_1 \mathrm{d}z_2$$

where $\mathbf{R}_4(z_1, z_2)$ has no poles in the considered contours. It is then sufficient to evaluate the top left entry of each of the matrices $\mathbf{J}^T \mathbf{D}^{\alpha-1} \mathbf{Q}_{z_1}^\alpha \mathbf{U} \mathbf{\Lambda}$, $\mathbf{U}^T \mathbf{Q}_{z_1}^\alpha \mathbf{D}^{\alpha-1} \mathbf{D}_a \mathbf{D}^{\alpha-1} \mathbf{Q}_{z_2}^\alpha \mathbf{U}$ and $\mathbf{\Lambda} \mathbf{U}^T \mathbf{Q}_{z_2}^\alpha \mathbf{D}^{\alpha-1} \mathbf{J}$ to compute the whole integrand. The first and the third of the latter matrices have been evaluated in the proof of Theorem 22. We are then left with the top left entry of $\mathbf{U}^T \mathbf{Q}_{z_1}^\alpha \mathbf{D}^{\alpha-1} \mathbf{D}_a \mathbf{D}^{\alpha-1} \mathbf{Q}_{z_2}^\alpha \mathbf{U}$ which is $\frac{1}{n} \mathbf{J}^T \mathbf{D}^{\alpha-1} \mathbf{Q}_{z_1}^\alpha \mathbf{D}^{\alpha-1} \mathbf{D}_a \mathbf{D}^{\alpha-1} \mathbf{Q}_{z_2}^\alpha \mathbf{D}^{\alpha-1} \mathbf{J}$ from Theorem 2. The former quantity has already been evaluated in the previous section but for (z, z) replaced by (z_1, z_2) and the diagonal matrix between $\mathbf{Q}_{z_1}^\alpha$ and $\mathbf{Q}_{z_2}^\alpha$ being here $\mathbf{D}^{\alpha-1} \mathbf{D}_a \mathbf{D}^{\alpha-1}$ instead of $\mathbf{D}^{2(\alpha-1)}$. We thus have

$$\left(\mathbf{U}^T \mathbf{Q}_{z_1}^\alpha \mathbf{D}^{\alpha-1} \mathbf{D}_a \mathbf{D}^{\alpha-1} \mathbf{Q}_{z_2}^\alpha \mathbf{U} \right)_{11} \xrightarrow{\text{a.s.}} c_a \left(e_{00;2}^\alpha(z_1, z_2) \mathcal{D} \left(\boldsymbol{\delta}_{i=a} \right)_{i=1}^K + \chi^\alpha(z_1, z_2) \mathcal{D}(\mathbf{c}) \right).$$

Finally, we are left to evaluate

$$\left(\frac{1}{2\pi i} \right)^2 \oint_{\Gamma_{\rho 1}} \oint_{\Gamma_{\rho 2}} \frac{1}{n} \left[e_{00}^\alpha(z_1) (\mathcal{D}(\mathbf{c}) - \mathbf{c}\mathbf{c}^\mathsf{T}) \mathbf{M} (\mathbf{I}_K - \mathbf{c}\mathbf{1}_K^\mathsf{T}) - \beta^\alpha(z_1) \mathbf{c}\mathbf{1}_K^\mathsf{T} \right] (\underline{\mathbf{G}}_{z_1}^\alpha)^{-1}$$
$$\times c_a \left(e_{00;2}^\alpha(z_1, z_2) \mathcal{D} \left(\boldsymbol{\delta}_{i=a} \right)_{i=1}^K + \chi^\alpha(z_1, z_2) \mathcal{D}(\mathbf{c}) \right)$$
$$\times \left((\underline{\mathbf{G}}_{z_2}^\alpha)^{-1} \right)^\mathsf{T} \left[e_{00}^\alpha(z_2) (\mathbf{I}_K - \mathbf{1}_K \mathbf{c}^\mathsf{T}) \mathbf{M} (\mathcal{D}(\mathbf{c}) - \mathbf{c}\mathbf{c}^\mathsf{T}) - \beta^\alpha(z_2) \mathbf{1}_K \mathbf{c}^\mathsf{T} \right] \mathrm{d}z_1 \mathrm{d}z_2.$$

We can then perform a residue calculus similar to what was done in the proof of Theorem 22. Additionnaly, we use the fact that the eigenvectors $\mathbf{v}_{\rho_1}$ and $\mathbf{v}_{\rho_2}$ corresponding to distinct eigenvalues ρ_1 and ρ_2 of the symmetric matrix $\mathcal{D}(\mathbf{c})^{\frac{1}{2}} \left(\mathbf{I}_K - \mathbf{1}_K \mathbf{c}^\mathsf{T} \right) \mathbf{M} \left(\mathbf{I}_K - \mathbf{c}\mathbf{1}_K^T \right) \mathcal{D}(\mathbf{c})^{\frac{1}{2}}$ are orthogonals. All calculus done, we get

$$\left(\frac{1}{n} \frac{\mathbf{J}^\mathsf{T} \mathbf{D}^{\alpha-1} \mathbf{u}_i^\alpha (\mathbf{u}_i^\alpha)^\mathsf{T} \mathbf{D}^{\alpha-1} \mathbf{D}_a \mathbf{D}^{\alpha-1} \mathbf{u}_j^\alpha (\mathbf{u}_j^\alpha)^\mathsf{T} \mathbf{D}^{\alpha-1} \mathbf{J}}{\left((\mathbf{u}_i^\alpha)^\mathsf{T} \mathbf{D}^{2(\alpha-1)} \mathbf{u}_i^\alpha \right) \left((\mathbf{u}_j^\alpha)^\mathsf{T} \mathbf{D}^{2(\alpha-1)} \mathbf{u}_j^\alpha \right)} \right)_{ef} \xrightarrow{\text{a.s.}}$$
$$\frac{e_{00}^\alpha(\rho_i) e_{00}^\alpha(\rho_j)}{\left(e_{00;2}^\alpha(\rho_i, \rho_i) + \chi^\alpha(\rho_i) \right) \left(e_{00;2}^\alpha(\rho_j, \rho_j) + \chi^\alpha(\rho_j) \right)}$$
$$\times \left[e_{00;2}^\alpha(\rho_i, \rho_j) \sqrt{c_e c_f} v_e^i v_f^j v_a^i v_a^j + \delta_{\rho_i = \rho_j} c_a \chi^\alpha(\rho_i) \sqrt{c_e c_f} v_e^i v_f^j \right]. \tag{38}$$

We are thus now ready to evaluate the σ_{ij}^a's. By definition,

$$\sigma_{ij}^a = \left[\frac{(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{D}_a\mathbf{D}^{\alpha-1}\mathbf{u}_j^\alpha}{\sqrt{(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{2(\alpha-1)}\mathbf{u}_i^\alpha}\sqrt{(\mathbf{u}_j^\alpha)^\mathsf{T}\mathbf{D}^{2(\alpha-1)}\mathbf{u}_j^\alpha}} - \frac{1}{n_a}\frac{(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{j}_a}{\sqrt{(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{2(\alpha-1)}\mathbf{u}_i^\alpha}}\frac{(\mathbf{u}_j^\alpha)^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{j}_a}{\sqrt{(\mathbf{u}_j^\alpha)^\mathsf{T}\mathbf{D}^{2(\alpha-1)}\mathbf{u}_j^\alpha}} \right]. \tag{39}$$

The first right hand side term is estimated by dividing $\left(\frac{1}{n}\frac{\mathbf{J}^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{u}_i^\alpha(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{D}_a\mathbf{D}^{\alpha-1}\mathbf{u}_j^\alpha(\mathbf{u}_j^\alpha)^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{J}}{((\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{2(\alpha-1)}\mathbf{u}_i^\alpha)((\mathbf{u}_j^\alpha)^\mathsf{T}\mathbf{D}^{2(\alpha-1)}\mathbf{u}_j^\alpha)}\right)_{ef}$ (Equation (38)) by $\frac{1}{\sqrt{n}}\frac{(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{j}_e}{\sqrt{(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{2(\alpha-1)}\mathbf{u}_i^\alpha}} \neq 0$ and $\frac{1}{\sqrt{n}}\frac{(\mathbf{u}_j^\alpha)^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{j}_f}{\sqrt{(\mathbf{u}_j^\alpha)^\mathsf{T}\mathbf{D}^{2(\alpha-1)}\mathbf{u}_j^\alpha}} \neq 0$ for any couple of indexes (e, f) such that the aforementioned quantities are non zeros. Indeed from the definition of ν_i^a and Equation 37, we get

$$\frac{1}{\sqrt{n}}\frac{(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{\alpha-1}\mathbf{j}_e}{\sqrt{(\mathbf{u}_i^\alpha)^\mathsf{T}\mathbf{D}^{2(\alpha-1)}\mathbf{u}_i^\alpha}} \xrightarrow{\text{a.s.}} \sqrt{c_e}\frac{e_{00}^\alpha(\rho)}{\sqrt{e_{00;2}^\alpha(\rho,\rho)+\chi^\alpha(\rho)}}\left|v_e^i\right|. \tag{40}$$

The covariances σ_{ij}^a's are then found by combining the previous estimates (38) and (40) as per the Definition (39) of the σ_{ij}^a's. This proves the following theorem giving the limit of the empirical class covariances σ_{ij}^a's.

Theorem 23 (Covariances) *For two unit multiplicity eigenpairs $(\lambda_1(\bar{\mathbf{M}}), \mathbf{v}^1)$ and $(\lambda_2(\bar{\mathbf{M}}), \mathbf{v}^2)$ of $\mathcal{D}(\mathbf{c})^{\frac{1}{2}}\left(\mathbf{I}_K - \mathbf{1}_K\mathbf{c}^\mathsf{T}\right)\mathbf{M}\left(\mathbf{I}_K - \mathbf{c}\mathbf{1}_K^T\right)\mathcal{D}(\mathbf{c})^{\frac{1}{2}}$ mapped respectively to $(\rho_1, \mathbf{u}_i^\alpha)$ and $(\rho_2, \mathbf{u}_j^\alpha)$ eigenpairs of $\mathbf{L}_\alpha$ and for σ_{ij}^a defined in (27), we have almost surely as $n \to \infty$, $\left|\sigma_{ij}^a - \sigma_{ij}^{a,\infty}\right| \to 0$ where*

$$\sigma_{ij}^{a,\infty} \equiv \frac{\left[\left(e_{00;2}^\alpha(\rho_1,\rho_2) - e_{00}^\alpha(\rho_1)e_{00}^\alpha(\rho_2)\right)v_a^{\rho_1}v_a^{\rho_2} + \delta_{\rho_1}^{\rho_2}c_a\chi^\alpha(\rho_1)\right]}{\sqrt{e_{00;2}^\alpha(\rho_1)+\chi^\alpha(\rho_1)}\sqrt{e_{00;2}^\alpha(\rho_2)+\chi^\alpha(\rho_2)}}$$

where $\chi^\alpha(\rho)$ is defined in Theorem 22.

From Theorems 22 and 23, $\nu_i^{a,\infty}$ and $\sigma_{ij}^{a,\infty}$ depend on the e_{ij}'s (defined in Theorem 4), the normalized eigenvectors $\mathbf{v}$ of $\mathcal{D}(\mathbf{c})^{\frac{1}{2}}\left(\mathbf{I}_K - \mathbf{1}_K\mathbf{c}^\mathsf{T}\right)\mathbf{M}\left(\mathbf{I}_K - \mathbf{c}\mathbf{1}_K^T\right)\mathcal{D}(\mathbf{c})^{\frac{1}{2}}$ and the proportions c_a's of classes. Thanks to Lemma 11, the e_{ij}'s can consistently be estimated similarly to what was described in Proposition 12. Namely, the q_i's can be estimated using $\hat{q}_i = \frac{d_i}{\sqrt{\mathbf{d}^\mathsf{T}\mathbf{1}_n}}$ and replaced in Equations (20), (21), (22) to obtain consistent estimates for the e_{ij}'s. However, the eigenvectors $\mathbf{v}$ and the class proportions are not directly accessible in practice. Nevertheless, in the particular case of $K = 2$ classes, we know exactly $\mathbf{v}$.

Remark 24 ($K = 2$ classes) *Here, only one isolated eigenvector is used for the classification. Since $\mathbf{v}_r$ (right eigenvector of $\bar{\mathbf{M}}$) is orthogonal to $\mathbf{1}_2$, $\mathbf{v}_r$ is necessarily the vector $[1, -1]^\mathsf{T}$. Hence, the normalized eigenvector $\mathbf{v} = \frac{\mathcal{D}(\mathbf{c})^{-\frac{1}{2}}\mathbf{v}_r}{\|\mathcal{D}(\mathbf{c})^{-\frac{1}{2}}\mathbf{v}_r\|}$ is $\frac{1}{\sqrt{1/c_1+1/c_2}}\left[\frac{1}{\sqrt{c_1}}, -\frac{1}{\sqrt{c_2}}\right]^\mathsf{T}$.*

We thus obtain from Theorems 22 and 23 along with Remark 24,

Corollary 25 (Means and covariances for $K = 2$ classes) *For $a = 1, 2$*

$$(\nu^{a,\infty})^2 = \frac{\left[e_{00}^\alpha(\rho)\right]^2}{\left(e_{00;2}^\alpha(\rho,\rho)+\chi^\alpha(\rho)\right)\left(1+\frac{c_a}{1-c_a}\right)}$$

$$(\sigma^{a,\infty})^2 = \frac{\left[\frac{\left(e_{00;2}^\alpha(\rho,\rho)-e_{00}^\alpha(\rho)^2\right)}{\left(1+\frac{c_a}{1-c_a}\right)} + c_a\chi^\alpha(\rho)\right]}{e_{00;2}^\alpha(\rho,\rho)+\chi^\alpha(\rho)}$$

for ρ the unique isolated eigenvalue of $\mathbf{L}_\alpha$ (if it exists).

6.5 Non informative eigenvectors

The objective of this section is to show that the eigenvectors $\tilde{\mathbf{u}}^\alpha$ of $\mathbf{L}_\alpha$ associated to the limiting eigenvalue $\tilde{\rho}$ for which $1+\theta^\alpha(\tilde{\rho}) = 0$ (Remark 20) are not useful for the classification.

Let us write as in Section 6.4

$$\tilde{\mathbf{u}}^\alpha = \sum_{a=1}^K \tilde{\nu}^a\mathbf{j}_a + \sqrt{\tilde{\sigma}_{ii}^a}\mathbf{w}^a \tag{41}$$

where $\mathbf{w}^a \in \mathbb{R}^n$ is a random vector orthogonal to $\mathbf{j}_a$ of norm $\sqrt{n_a}$, supported on the indices of $\mathcal{C}_a$ with identically distributed entries. We shall show that $\tilde{\nu}^a$ is independent of class $\mathcal{C}_a$ and thus, any correct classification cannot be done using $\tilde{\mathbf{u}}^\alpha$. From (41), $\tilde{\nu}^a = \frac{(\tilde{\mathbf{u}}^\alpha)^\mathsf{T}\mathbf{j}_a}{n_a}$ which can be retrieved from the diagonal elements of $\frac{1}{n}\mathbf{J}^\mathsf{T}\tilde{\mathbf{u}}^\alpha(\tilde{\mathbf{u}}^\alpha)^\mathsf{T}\mathbf{J}$. We will evaluate this object by using the same technique as in Section 6.4. By the residue formula, we have

$$\frac{1}{n}\mathbf{J}^\mathsf{T}\tilde{\mathbf{u}}^\alpha(\tilde{\mathbf{u}}^\alpha)^\mathsf{T}\mathbf{J} = -\frac{1}{2\pi i}\oint_{\Gamma_{\tilde{\rho}}}\frac{1}{n}\mathbf{J}^\mathsf{T}\left(\mathbf{L}_\alpha - z\mathbf{I}_n\right)^{-1}\mathbf{J}\mathrm{d}z \tag{42}$$

$$= -\frac{1}{2\pi i}\oint_{\Gamma_{\tilde{\rho}}}\frac{1}{n}\mathbf{J}^\mathsf{T}\mathbf{Q}_z^\alpha\mathbf{J}\mathrm{d}z + \frac{1}{2\pi i}\oint_{\Gamma_{\tilde{\rho}}}\frac{1}{n}\mathbf{J}^\mathsf{T}\mathbf{Q}_z^\alpha\mathbf{U}\mathbf{\Lambda}\left(\mathbf{I}_{K+1} + \mathbf{U}^\mathsf{T}\mathbf{Q}_z^\alpha\mathbf{U}\mathbf{\Lambda}\right)^{-1}\mathbf{U}^\mathsf{T}\mathbf{Q}_z^\alpha\mathbf{J} \tag{43}$$

for large n almost surely, where $\Gamma_{\tilde{\rho}}$ is a complex (positively oriented) contour circling around the limiting eigenvalue $\tilde{\rho}$ only. The first integral $-\frac{1}{2\pi i}\oint_{\Gamma_{\tilde{\rho}}}\frac{1}{n}\mathbf{J}^\mathsf{T}\mathbf{Q}_z^\alpha\mathbf{J}\mathrm{d}z$ is asymptotically zero since, from Proposition 18, the integrand has no poles in the contour $\Gamma_{\tilde{\rho}}$. We thus obtain similarly as in Section 6.4

$$\frac{1}{n}\mathbf{J}^\mathsf{T}\tilde{\mathbf{u}}^\alpha(\tilde{\mathbf{u}}^\alpha)^\mathsf{T}\mathbf{J} = \frac{1}{n}\mathbf{J}^\mathsf{T}\mathbf{Q}_{\tilde{\rho}}^\alpha\mathbf{U}\mathbf{\Lambda}\left[\lim_{z\to\tilde{\rho}}(z-\tilde{\rho})(\mathbf{I}_{K+1} + \mathbf{U}^\mathsf{T}\mathbf{Q}_z^\alpha\mathbf{U}\mathbf{\Lambda})^{-1}\right]\mathbf{U}^\mathsf{T}\mathbf{Q}_{\tilde{\rho}}^\alpha\mathbf{J}. \tag{44}$$

From (29), the entries $(1, 2)$ and $(2, 2)$ of $(\mathbf{I}_{K+1} + \mathbf{U}^\mathsf{T}\mathbf{Q}_z^\alpha\mathbf{U}\mathbf{\Lambda})^{-1}$ do not contain $(\underline{\mathbf{G}}_z^\alpha)^{-1}$ since $\left[\underline{\mathbf{G}}_z^\alpha - \frac{\gamma(z)m_\mu e_{21}^\alpha(z)}{ze_{10}^\alpha(z)}\mathbf{c}\mathbf{1}_K^\mathsf{T}\right]^{-1}\mathbf{c} = -\frac{m_\mu}{ze_{10}^\alpha(z)}\mathbf{c}$ and thus, the above limit will give zero for those entries. We thus get

$$\frac{1}{n}\mathbf{J}^\mathsf{T}\tilde{\mathbf{u}}^\alpha(\tilde{\mathbf{u}}^\alpha)^\mathsf{T}\mathbf{J} = \frac{1}{n}\mathbf{J}^\mathsf{T}\mathbf{Q}_{\tilde{\rho}}^\alpha\mathbf{U}\mathbf{\Lambda}\left[\lim_{z\to\tilde{\rho}}(z-\tilde{\rho})\begin{pmatrix}(\underline{\mathbf{G}}_z^\alpha)^{-1} & 0\\ \frac{\gamma(z)m_\mu}{ze_{21}^\alpha(z)}\mathbf{1}_K^\mathsf{T}(\underline{\mathbf{G}}_z^\alpha)^{-1} & 0\end{pmatrix}\right]\mathbf{U}^\mathsf{T}\mathbf{Q}_{\tilde{\rho}}^\alpha\mathbf{J}. \tag{45}$$

We recall that in the case under study ($1 + \theta^\alpha(\tilde{\rho}) = 0$), $\mathbf{1}_K$ and $\mathbf{c}$ are respectively left and right eigenvectors of $\underline{\mathbf{G}}_z^\alpha$ associated to the vanishing eigenvalue. We can thus write

$\underline{\mathbf{G}}_z^\alpha = \rho_z \mathbf{c}\mathbf{1}_K^\mathsf{T} + \tilde{\mathbf{V}}_{r,z}\tilde{\mathbf{\Sigma}}_z\tilde{\mathbf{V}}_{l,z}^T$ where ρ_z is the vanishing eigenvalue when $z \to \tilde{\rho}$ and $\tilde{\mathbf{V}}_{r,z}$ and $\tilde{\mathbf{V}}_{l,z}$ are respectively sets of right and left eigenspaces associated with non vanishing eigenvalues. Hence, we have

$$\lim_{z\to\tilde{\rho}}(z-\tilde{\rho})(\underline{\mathbf{G}}_z^\alpha)^{-1} \overset{(1)}{=} \lim_{z\to\tilde{\rho}} \frac{\mathbf{c}\mathbf{1}_K^\mathsf{T}}{\rho_z'} \overset{(2)}{=} \lim_{z\to\tilde{\rho}} \frac{\mathbf{c}\mathbf{1}_K^\mathsf{T}}{\mathbf{1}_K^\mathsf{T}(\underline{\mathbf{G}}_z^\alpha)'\mathbf{c}} \overset{(3)}{=} \lim_{z\to\tilde{\rho}} \frac{\mathbf{c}\mathbf{1}_K^\mathsf{T}}{(\theta^\alpha(z))'} \overset{(4)}{=} \frac{\mathbf{c}\mathbf{1}_K^\mathsf{T}}{(\theta^\alpha(\tilde{\rho}))'} \tag{46}$$

where in (1) we have used the l'Hopital rule, in (2) we used the fact that ρ_z can be written $\rho_z = \mathbf{1}_K^\mathsf{T}\underline{\mathbf{G}}_z^\alpha\mathbf{c}$ and in (3) we have used $(\underline{\mathbf{G}}_{\tilde{\rho}}^\alpha)' = (e_{21}^\alpha(\tilde{\rho}))'\left(\mathcal{D}(\mathbf{c}) - \mathbf{c}\mathbf{c}^\mathsf{T}\right)\mathbf{M}\left(\mathbf{I}_K - \mathbf{c}\mathbf{1}_K^T\right) + (\theta^\alpha(\tilde{\rho}))'\mathbf{c}\mathbf{1}_K^T$ and $\mathbf{1}_K^\mathsf{T}\mathbf{c} = 1$. We then have

$$\frac{1}{n}\mathbf{J}^\mathsf{T}\tilde{\mathbf{u}}^\alpha(\tilde{\mathbf{u}}^\alpha)^\mathsf{T}\mathbf{J} = \frac{1}{n}(\mathbf{J}^\mathsf{T}\mathbf{Q}_{\tilde{\rho}}^\alpha\mathbf{U}\mathbf{\Lambda})_{11}\frac{\mathbf{c}\mathbf{1}_K^\mathsf{T}}{(\theta^\alpha(\tilde{\rho}))'}(\mathbf{U}^\mathsf{T}\mathbf{Q}_{\tilde{\rho}}^\alpha\mathbf{J})_{11} \tag{47}$$

$$+ \frac{1}{n}(\mathbf{J}^\mathsf{T}\mathbf{Q}_{\tilde{\rho}}^\alpha\mathbf{U}\mathbf{\Lambda})_{12}\frac{\gamma(\tilde{\rho})m_\mu\mathbf{1}_K^\mathsf{T}}{\tilde{\rho}e_{21}^\alpha(\tilde{\rho})(\theta^\alpha(\tilde{\rho}))'}(\mathbf{U}^\mathsf{T}\mathbf{Q}_{\tilde{\rho}}^\alpha\mathbf{J})_{11}. \tag{48}$$

All calculus done similarly as in Section 6.4, we get

$$\begin{aligned}\frac{1}{n}\mathbf{J}^\mathsf{T}\tilde{\mathbf{u}}^\alpha(\tilde{\mathbf{u}}^\alpha)^\mathsf{T}\mathbf{J} \xrightarrow{\text{a.s.}} \frac{e_{1,\frac{1}{2}}^\alpha(\tilde{\rho})}{(\theta^\alpha(\tilde{\rho}))'}&\left[e_{1,\frac{1}{2}}^\alpha(\tilde{\rho})\left(\mathcal{D}(\mathbf{c}) - \mathbf{c}\mathbf{c}^\mathsf{T}\right)\mathbf{M}\left(\mathbf{I}_K - \mathbf{c}\mathbf{1}_K^T\right)\right.\\ &\left. - \frac{1}{m_\mu}\left(\int t^\alpha\mu(\mathrm{d}t) + e_{0,\frac{1}{2}}^\alpha(\tilde{\rho})\right)\mathbf{c}\mathbf{1}_K^\mathsf{T}\right]\mathbf{c}\mathbf{c}^\mathsf{T}\\ &- (e_{1,\frac{1}{2}}^\alpha(\tilde{\rho}))^2\frac{\gamma(\tilde{\rho})m_\mu}{\tilde{\rho}e_{21}^\alpha(\tilde{\rho})(\theta^\alpha(\tilde{\rho}))'}\mathbf{c}\mathbf{c}^\mathsf{T}.\end{aligned}$$

Finally,

$$\frac{1}{n}\mathbf{J}^\mathsf{T}\tilde{\mathbf{u}}^\alpha(\tilde{\mathbf{u}}^\alpha)^\mathsf{T}\mathbf{J} \xrightarrow{\text{a.s.}} -\frac{e_{1,\frac{1}{2}}^\alpha(\tilde{\rho})}{m_\mu(\theta^\alpha(\tilde{\rho}))'}\left[\int t^\alpha\mu(\mathrm{d}t) + e_{0,\frac{1}{2}}^\alpha(\tilde{\rho}) + \frac{e_{1,\frac{1}{2}}^\alpha(\tilde{\rho})\gamma(\tilde{\rho})m_\mu^2}{\tilde{\rho}e_{21}^\alpha(\tilde{\rho})}\right]\mathbf{c}\mathbf{c}^\mathsf{T}. \tag{49}$$

By recalling that $\tilde{\nu}^a = \frac{(\tilde{\mathbf{u}}^\alpha)^\mathsf{T}\mathbf{j}_a}{n_a} = \sqrt{\frac{1}{n_a}\left[\frac{1}{n}\mathcal{D}(\mathbf{c})^{-\frac{1}{2}}\mathbf{J}^\mathsf{T}\tilde{\mathbf{u}}^\alpha(\tilde{\mathbf{u}}^\alpha)^\mathsf{T}\mathbf{J}\mathcal{D}(\mathbf{c})^{-\frac{1}{2}}\right]_{aa}}$, from (49) we deduce that

$$\tilde{\nu}^a \xrightarrow{\text{a.s.}} -\frac{1}{\sqrt{n}}\frac{e_{1,\frac{1}{2}}^\alpha(\tilde{\rho})}{m_\mu(\theta^\alpha(\tilde{\rho}))'}\left[\int t^\alpha\mu(\mathrm{d}t) + e_{0,\frac{1}{2}}^\alpha(\tilde{\rho}) + \frac{e_{1,\frac{1}{2}}^\alpha(\tilde{\rho})\gamma(\tilde{\rho})m_\mu^2}{\tilde{\rho}e_{21}^\alpha(\tilde{\rho})}\right]$$

which is independent of the class information (class proportions or inter-class affinities). This concludes the proof.

Appendix A. Stein Lemma and Nash Poincare inequality

Lemma 26 *Let x be a standard real Gaussian random variable and $f : \mathbb{R} \to \mathbb{R}$ be a $\mathbb{C}^1$ function with first derivative $f'(x)$ having at most polynomial growth. Then,*

$$\mathbb{E}[xf(x)] = \mathbb{E}[f'(x)].$$

Lemma 27 *Let x be a standard real Gaussian random variable and $f : \mathbb{R} \to \mathbb{R}$ be a $\mathbb{C}^1$ function with first derivative $f'(x)$. Then, we have*

$$\mathrm{Var}[f(x)] \leq \mathbb{E}[|f'(x)|^2].$$

The proofs of those lemma can be found in (Pastur et al., 2011).

Appendix B. Consistent estimates of the averages connectivity weights q_i's

Lemma 28 *Under Assumption 1,*

$$\max_{1\leq i\leq n}|q_i - \hat{q}_i| \to 0 \tag{50}$$

almost surely, where $\hat{q}_i = \frac{d_i}{\sqrt{\mathbf{d}^\mathsf{T}\mathbf{1}_n}}$.

We need to prove that $\sum_{n=1}^\infty \mathbb{P}\left(\max_{1\leq i\leq n}|q_i - \hat{q}_i| > \eta\right) < \infty$ for any $\eta > 0$ so that we can conclude from the first Borel Cantelli lemma (Theorem 4.3 in (Billingsley, 1995)) that $\mathbb{P}\left(\limsup_n \max_{1\leq i\leq n}|q_i - \hat{q}_i| > \eta\right) = 0$ from which Lemma 28 unfolds. We have that

$$\begin{aligned}\mathbb{P}\left(\max_{1\leq i\leq n}|q_i - \hat{q}_i| > \eta\right) &\leq \sum_{i=1}^n \mathbb{P}(|q_i - \hat{q}_i| > \eta)\\ &\leq \sum_{i=1}^n \mathbb{P}(\hat{q}_i - q_i > \eta) + \mathbb{P}(q_i - \hat{q}_i > \eta).\end{aligned} \tag{51}$$

Let us treat for instance the term $\mathbb{P}(\hat{q}_i - q_i > \eta)$ in the following. Since $A_{ij} = q_iq_j + q_iq_j\frac{M_{g_ig_j}}{\sqrt{n}} + X_{ij}$ with X_{ij} a zero mean random variable, we have $\frac{1}{n}\sum_{j=1}^n \mathbb{E}A_{ij} \to q_im_\mu$ and $\frac{1}{n^2}\sum_{i,j}\mathbb{E}A_{ij} \to m_\mu^2$ in the limit $n \to \infty$. For $\hat{q}_i = \frac{\sum_{j=1}^n A_{ij}}{\sqrt{\sum_{i,j}A_{ij}}}$, we can write

$$\hat{q}_i - q_i = \underbrace{\frac{\frac{1}{n}\sum_{j=1}^n(A_{ij} - \mathbb{E}A_{ij})}{\sqrt{\frac{1}{n^2}\sum_{i,j}\mathbb{E}A_{ij}}}}_{A} + \underbrace{\frac{\frac{1}{n}\sum_{j=1}^n A_{ij}}{\sqrt{\frac{1}{n^2}\sum_{i,j}A_{ij}}} - \frac{\frac{1}{n}\sum_{j=1}^n A_{ij}}{\sqrt{\frac{1}{n^2}\sum_{i,j}\mathbb{E}A_{ij}}}}_{B} + \underbrace{\frac{\frac{1}{n}\sum_{j=1}^n \mathbb{E}A_{ij}}{\sqrt{\frac{1}{n^2}\sum_{i,j}\mathbb{E}A_{ij}}} - q_i}_{C}$$

Since A, B and C tend to zero in the limit $n \to \infty$, we will next use the fact that $\mathbb{P}(\hat{q}_i - q_i > \eta) \leq \mathbb{P}(A > \eta/3) + \mathbb{P}(B > \eta/3) + \mathbb{P}(C > \eta/3)$ and show that all those individual probabilities vanish asymptotically. Since the term C is deterministic and tends to zero in the limit $n \to \infty$, we have $\mathbb{P}(C > \eta/3) = 0$ for all large n. Let us then control $\mathbb{P}(A > \eta/3)$ and $\mathbb{P}(B > \eta/3)$. We have

$$\begin{aligned}\mathbb{P}(A > \eta/3) &= \mathbb{P}\left(\frac{1}{n}\sum_{j=1}^n(A_{ij} - \mathbb{E}A_{ij}) > \frac{\eta m_\mu}{3} + o(1)\right)\\ &\leq \exp\left[-\frac{n\eta^2m_\mu^2}{18\left(\sigma^2 + \eta m_\mu/9\right))} + o(1)\right]\end{aligned} \tag{52}$$

with $\sigma^2 = \limsup_n \max_{1\leq i\leq n} q_i(\sum_j q_j) - q_i^2(\sum_j q_j^2)$ and where in the last inequality of (52), we have used Bernstein's inequality (Theorem 3 in (Boucheron et al., 2013)) since the A_{ij}'s

are independent Bernoulli random variables with variance $\sigma_{ij}^2 = q_i q_j(1-q_i q_j) + \mathcal{O}(n^{-\frac{1}{2}})$. For the term B we have

$$\begin{aligned}
\mathbb{P}(B > \eta/3) &= \mathbb{P}\left(\frac{1}{n}\sum_{j=1}^n A_{ij} \frac{\sqrt{\frac{1}{n^2}\sum_{i,j}\mathbb{E}A_{ij}} - \sqrt{\frac{1}{n^2}\sum_{i,j}A_{ij}}}{\sqrt{\frac{1}{n^2}\sum_{i,j}A_{ij}}} > \frac{\eta m_\mu}{3} + o(1) \right) \\
&\overset{(1)}{\leq} \mathbb{P}\left(\left| \frac{\sqrt{\frac{1}{n^2}\sum_{i,j}\mathbb{E}A_{ij}} - \sqrt{\frac{1}{n^2}\sum_{i,j}A_{ij}}}{\sqrt{\frac{1}{n^2}\sum_{i,j}A_{ij}}} \right| > \frac{\eta m_\mu}{3} + o(1) \right) \\
&\overset{(2)}{\leq} \mathbb{P}\left(\left| \sqrt{\frac{1}{n^2}\sum_{i,j}\mathbb{E}A_{ij}} - \sqrt{\frac{1}{n^2}\sum_{i,j}A_{ij}} \right| > \frac{\eta(m_\mu + o(1))\sqrt{\frac{1}{n^2}\sum_{i,j}A_{ij}}}{3}, \frac{1}{n^2}\sum_{i,j}A_{ij} > \psi \right) \\
&+ \mathbb{P}\left(\left| \sqrt{\frac{1}{n^2}\sum_{i,j}\mathbb{E}A_{ij}} - \sqrt{\frac{1}{n^2}\sum_{i,j}A_{ij}} \right| > \frac{\eta m_\mu \sqrt{\frac{1}{n^2}\sum_{i,j}A_{ij}}}{3} + o(1), \frac{1}{n^2}\sum_{i,j}A_{ij} \leq \psi \right) \\
&\overset{(3)}{\leq} \underbrace{\mathbb{P}\left(\left| \sqrt{\frac{1}{n^2}\sum_{i,j}\mathbb{E}A_{ij}} - \sqrt{\frac{1}{n^2}\sum_{i,j}A_{ij}} \right| > \frac{\eta m_\mu \sqrt{\psi}}{3} + o(1) \right)}_{B_1} + \underbrace{\mathbb{P}\left(\frac{1}{n^2}\sum_{i,j}A_{ij} \leq \psi \right)}_{B_2}
\end{aligned} \tag{53}$$

where in the inequality (1) we have used the fact that $n^{-1}\sum_{j=1}^n A_{ij} \leq 1$; in the inequality (2) $\psi > 0$ is any constant smaller than m_μ^2 and in the inequality (3) we have used $\sqrt{n^{-2}\sum_{i,j}A_{ij}} > \sqrt{\psi}$ and the fact that the probability of the intersection between two events is always smaller than the probability of one of those events. It then remains to control B_1 and B_2. For B_2 we have

$$\mathbb{P}\left(\frac{1}{n^2}\sum_{i,j}A_{ij} \leq \psi \right) \leq \exp\left[-\frac{n(m_\mu^2 - \psi)^2}{2\left(\sigma^2 + (m_\mu^2 - \psi)/3\right)} + o(1) \right] \tag{54}$$

where the inequality follows from Bernstein's inequality with the similar arguments as previously. Finally for the term B_1 we have

$$\begin{aligned}
&\mathbb{P}\left(\left| \sqrt{\frac{1}{n^2}\sum_{i,j}\mathbb{E}A_{ij}} - \sqrt{\frac{1}{n^2}\sum_{i,j}A_{ij}} \right| > \frac{\eta m_\mu \sqrt{\psi}}{3} + o(1) \right) \\
&= \mathbb{P}\left(\left| \frac{\frac{1}{n^2}\sum_{i,j}\mathbb{E}A_{ij} - \frac{1}{n^2}\sum_{i,j}A_{ij}}{\sqrt{\frac{1}{n^2}\sum_{i,j}\mathbb{E}A_{ij}} + \sqrt{\frac{1}{n^2}\sum_{i,j}A_{ij}}} \right| > \frac{\eta m_\mu \sqrt{\psi}}{3} + o(1) \right) \\
&\overset{(1)}{\leq} \mathbb{P}\left(\left| \frac{1}{n^2}\sum_{i,j}\mathbb{E}A_{ij} - \frac{1}{n^2}\sum_{i,j}A_{ij} \right| > \frac{\eta m_\mu \sqrt{\psi}(m_\mu + \sqrt{\psi})}{3} + o(1) \right) + \mathbb{P}\left(\frac{1}{n^2}\sum_{i,j}A_{ij} \leq \psi \right) \\
&\overset{(2)}{\leq} \exp\left[-\frac{n\psi\left[\eta m_\mu(m_\mu + \sqrt{\psi})\right]^2}{18\left(\sigma^2 + \eta m_\mu\sqrt{\psi}(m_\mu + \sqrt{\psi})/9\right)} + o(1) \right] + \exp\left[-\frac{n(m_\mu^2 - \psi)^2}{2\left(\sigma^2 + (m_\mu^2 - \psi)/3\right)} + o(1) \right]
\end{aligned} \tag{55}$$

where in the inequality (1) of Equation (55) we have used the same arguments as in the inequalities (2)−(3) of Equation (52) and in the inequality (2) we have used Bernstein's inequality along with Equation (54). From Equations (52)(54)(55), we conclude that $\sum_{n=1}^\infty \sum_{i=1}^n \mathbb{P}(\hat{q}_i - q_i > \eta) < \infty$ since $m_\mu^2 - \psi > 0$. It follows the same lines to show that $\sum_{n=1}^\infty \sum_{i=1}^n \mathbb{P}(q_i - \hat{q}_i > \eta) < \infty$ which concludes the proof.

Appendix C. First deterministic equivalents

Let $\mathbf{Q}_z^\alpha = \left(\bar{\mathbf{X}} - z\mathbf{I}_n\right)^{-1}$ with $\bar{\mathbf{X}}$ a symmetric random matrix having independent entries $\bar{X}_{ij}$ which are Gaussian random variables with zero mean and variance $\frac{\sigma_{ij}^2}{n}$. For short, we shall denote $\mathbf{Q}_z^\alpha$ by $\mathbf{Q}$. We want to find a deterministic equivalent $\bar{\mathbf{Q}}$ of $\mathbf{Q}$ in the sense that $\frac{1}{n}\operatorname{tr}\mathbf{C}\mathbf{Q} - \frac{1}{n}\operatorname{tr}\mathbf{C}\bar{\mathbf{Q}} \to 0$ and $\mathbf{d}_1^\mathsf{T}(\mathbf{Q} - \bar{\mathbf{Q}})\mathbf{d}_2 \to 0$ almost surely, for all deterministic Hermitian matrix $\mathbf{C}$ and deterministic vectors $\mathbf{d}_i$ of bounded norms (spectral norm for matrices and Euclidian norm for vectors). To this end, we will evaluate $\mathbb{E}(\mathbf{Q})$ since using Lemma 27, one can show that $n^{-1}\operatorname{tr}(\mathbf{C}\mathbf{Q})$ and $\mathbf{a}^\mathsf{T}\mathbf{Q}\mathbf{b}$ concentrate respectively around $n^{-1}\operatorname{tr}(\mathbf{A}\mathbb{E}Q)$ and $\mathbf{d}_1^\mathsf{T}\mathbb{E}\mathbf{Q}\mathbf{d}_2$ for all bounded norm matrix $\mathbf{C}$ and vectors $\mathbf{d}_1, \mathbf{d}_2$. For the computations, we use standard Gaussian calculus introduced in (Pastur et al., 2011). Using the resolvent identity (for two invertible matrices $\mathbf{A}$ and $\mathbf{B}$, $\mathbf{A}^{-1} - \mathbf{B}^{-1} = -\mathbf{A}^{-1}(\mathbf{A} - \mathbf{B})\mathbf{B}^{-1}$), one has

$$\mathbf{Q} = \frac{1}{z}\bar{\mathbf{X}}\mathbf{Q} - \frac{1}{z}\mathbf{I}_n. \tag{56}$$

We then first compute $\mathbb{E}(\bar{\mathbf{X}}\mathbf{Q})$. By writing $\bar{X}_{il} = \frac{\sigma_{il}}{\sqrt{n}}Z_{il}$ where Z_{il} is a random variable with zero mean and unit variance, we thus have

$$\mathbb{E}(\bar{\mathbf{X}}\mathbf{Q})_{ij} = \sum_{l=1}^n \frac{\sigma_{il}}{\sqrt{n}}\mathbb{E}(Z_{il}Q_{lj}).$$

By applying Stein's Lemma (Lemma 26 in Section A), we have

$$\begin{aligned}
\mathbb{E}(Z_{il}Q_{lj}) &= \mathbb{E}\left(\frac{\partial(\bar{\mathbf{X}} - z\mathbf{I})^{-1}_{lj}}{\partial Z_{il}} \right) \\
&= \mathbb{E}\left(-(\bar{\mathbf{X}} - z\mathbf{I})^{-1}\frac{\partial\bar{\mathbf{X}}}{\partial Z_{il}}(\bar{\mathbf{X}} - z\mathbf{I})^{-1} \right)_{lj} \\
&= \mathbb{E}\left(-(\bar{\mathbf{X}} - z\mathbf{I})^{-1}\frac{\sigma_{il}}{\sqrt{n}}(\mathbf{E}_{il} + \mathbf{E}_{li})(\bar{\mathbf{X}} - z\mathbf{I})^{-1} \right)_{lj}
\end{aligned}$$

where $\mathbf{E}_{il}$ is the matrix with all entries equal to 0 but the entry (i,l) which is equal to 1. Using simple algebra, we have

$$\left((\bar{\mathbf{X}} - z\mathbf{I})^{-1}\mathbf{E}_{il}(\bar{\mathbf{X}} - z\mathbf{I})^{-1}\right)_{lj} = (\bar{\mathbf{X}} - z\mathbf{I})^{-1}_{li}(\bar{\mathbf{X}} - z\mathbf{I})^{-1}_{lj}$$

and

$$\left((\bar{\mathbf{X}} - z\mathbf{I})^{-1}\mathbf{E}_{li}(\bar{\mathbf{X}} - z\mathbf{I})^{-1}\right)_{lj} = (\bar{\mathbf{X}} - z\mathbf{I})^{-1}_{ll}(\bar{\mathbf{X}} - z\mathbf{I})^{-1}_{ij}.$$

We thus get

$$\mathbb{E}(\bar{\mathbf{X}}\mathbf{Q})_{ij} = \sum_{l=1}^n -\frac{\sigma_{il}^2}{n}\left(\mathbb{E}\left[Q_{li}Q_{lj}\right] + \mathbb{E}\left[Q_{ll}Q_{ij}\right]\right).$$

Going back to (56), we thus have

$$\begin{aligned}\mathbb{E}(Q_{ij}) &= -\frac{1}{z}\sum_{l=1}^{n}\frac{\sigma_{il}^2}{n}\mathbb{E}\left[Q_{li}Q_{lj}\right] - \frac{1}{z}\sum_{l=1}^{n}\frac{\sigma_{il}^2}{n}\mathbb{E}\left[Q_{ll}Q_{ij}\right] - \frac{1}{z}\delta_{ij} \\ &= -\frac{1}{z}\mathbb{E}\left[\mathbf{Q}\frac{\mathbf{\Sigma}_i}{n}\mathbf{Q}\right]_{ij} - \frac{1}{z}\mathbb{E}\left[Q_{ij}\operatorname{tr}\left(\frac{\mathbf{\Sigma}_i\mathbf{Q}}{n}\right)\right] - \frac{1}{z}\delta_{ij} \end{aligned} \tag{57}$$

where $\mathbf{\Sigma}_i = \mathcal{D}\left(\sigma_{ij}^2\right)_{j=1}^n$. Since the goal is to retrieve $\mathbb{E}(Q_{ij})$, the following lemma allows to split $\mathbb{E}\left[Q_{ij}\operatorname{tr}\left(\frac{\mathbf{\Sigma}_i\mathbf{Q}}{n}\right)\right]$ into $\mathbb{E}\left[Q_{ij}\right]$ and $\mathbb{E}\left[\operatorname{tr}\left(\frac{\mathbf{\Sigma}_i\mathbf{Q}}{n}\right)\right]$.

Lemma 29 *For $\mathbf{Q} = (\bar{\mathbf{X}} - z\mathbf{I}_n)^{-1}$ and $\mathbf{\Sigma}_i = \mathcal{D}(\sigma_{ij}^2)_{j=1}^n$, where $\bar{\mathbf{X}}$ is a symmetric random matrix having independent entries (up to the symmetry) of zero mean and variance $\frac{\sigma_{ij}^2}{n}$, we have*

$$\mathbb{E}\left[Q_{ij}\operatorname{tr}\left(\frac{\mathbf{\Sigma}_i\boldsymbol{Q}}{n}\right)\right] = \mathbb{E}\left[Q_{ij}\right]\mathbb{E}\left[\operatorname{tr}\left(\frac{\mathbf{\Sigma}_i\boldsymbol{Q}}{n}\right)\right] + o(1).$$

Proof For two real random variables x and y, by Cauchy-Shwarz's inequality,

$$\left|\mathbb{E}\left[(x-\mathbb{E}(x))(y-\mathbb{E}(y))\right]\right| \leq \sqrt{\operatorname{Var}(x)}\sqrt{\operatorname{Var}(y)}$$

which, for $x = \operatorname{tr}\left(\frac{\mathbf{\Sigma}_i\mathbf{Q}}{n}\right)$ and $y = Q_{ij} - \mathbb{E}(Q_{ij})$ gives

$$\left|\mathbb{E}\left[Q_{ij}\operatorname{tr}\left(\frac{\mathbf{\Sigma}_i\mathbf{Q}}{n}\right)\right] - \mathbb{E}\left[Q_{ij}\right]\mathbb{E}\left[\operatorname{tr}\left(\frac{\mathbf{\Sigma}_i\mathbf{Q}}{n}\right)\right]\right| \leq \sqrt{\operatorname{Var}(x)}\sqrt{\operatorname{Var}(y)}$$

since $\mathbb{E}(y)$ is equal to 0 in that case. Using Nash Poincaré inequality (Lemma 27 in Section A), one can show that $\operatorname{Var}(x) = \mathcal{O}\left(\frac{1}{n^2}\right)$(Hachem et al., 2007). Additionally, $\forall i,j$ and $z \in \mathbb{C}^+$, $|Q_{ij}| \leq \frac{1}{|\Im(z)|}$. This finally implies that $\mathbb{E}\left[Q_{ij}\operatorname{tr}\left(\frac{\mathbf{\Sigma}_i\mathbf{Q}}{n}\right)\right] - \mathbb{E}\left[Q_{ij}\right]\mathbb{E}\left[\operatorname{tr}\left(\frac{\mathbf{\Sigma}_i\mathbf{Q}}{n}\right)\right] = \mathcal{O}(n^{-1})$. ■

Since $\Im(-z-\mathbb{E}\operatorname{tr}(\frac{\mathbf{\Sigma}_i\mathbf{Q}}{n})) < -\Im(z)$ for $z \in \mathbb{C}^+$, $-z-\mathbb{E}\operatorname{tr}(\frac{\mathbf{\Sigma}_i\mathbf{Q}}{n})$ does not vanish asymptotically. Going back to $\mathbb{E}(Q_{ij})$ in Equation (57), we may then write

$$\mathbb{E}(Q_{ij}) = \frac{\mathbb{E}\left[\mathbf{Q}\frac{\mathbf{\Sigma}_i}{n}\mathbf{Q}\right]_{ij} + \delta_{ij}}{-z - \mathbb{E}\left[\operatorname{tr}\left(\frac{\mathbf{\Sigma}_i\mathbf{Q}}{n}\right)\right]} + \mathcal{O}(n^{-1}). \tag{58}$$

Multiplying Equation (58) by $\frac{\sigma_{ki}^2}{n}$, taking $j = i$, summing over i and scaling by n, we get

$$\operatorname{tr}\mathbb{E}\left(\frac{\mathbf{\Sigma}_k\mathbf{Q}}{n}\right) = \sum_{i=1}^{n}\left(\frac{\mathbb{E}\left[\frac{\mathbf{\Sigma}_k}{n}\mathbf{Q}\frac{\mathbf{\Sigma}_i}{n}\mathbf{Q}\right]_{ii} + \frac{\sigma_{ki}^2}{n}}{-z - \mathbb{E}\left[\operatorname{tr}\left(\frac{\mathbf{\Sigma}_i\mathbf{Q}}{n}\right)\right]}\right) + \mathcal{O}(n^{-1}).$$

Using a similar approach to the proof of Lemma 29, we can show that $\sum_{i=1}^n \mathbb{E}\left[\frac{\mathbf{\Sigma}_k}{n}\mathbf{Q}\frac{\mathbf{\Sigma}_i}{n}\mathbf{Q}\right]_{ii} = \mathcal{O}(n^{-1})$. We thus have

$$\frac{1}{n}\operatorname{tr}\mathbb{E}\left(\mathbf{\Sigma}_k\mathbf{Q}\right) = \sum_{i=1}^{n}\frac{\frac{\sigma_{ki}^2}{n}}{-z - \frac{1}{n}\mathbb{E}\left[\operatorname{tr}\left(\mathbf{\Sigma}_i\mathbf{Q}\right)\right]} + o(1).$$

By using standard techniques (Hachem et al., 2007), one can show that the unique solution $e_i(z)$ to $e_i(z) = \frac{1}{n}\sum_{j=1}^n \frac{\sigma_{ij}^2}{-z-e_j(z)}$ is such that $\frac{1}{n}\operatorname{tr}\mathbb{E}\left(\mathbf{\Sigma}_i\mathbf{Q}\right) - e_i(z) \xrightarrow{\text{a.s.}} 0$. Going back to Equation (58), we can thus write for large n

$$\mathbb{E}\left[(-z\mathbf{I} - \mathcal{D}(e_i(z)))\mathbf{Q}\right]_{ij} = \mathbb{E}\left[\mathbf{Q}\frac{\mathbf{\Sigma}_i}{n}\mathbf{Q}\right]_{ij} + \delta_{ij} + o(1). \tag{59}$$

Let us denote $\mathbf{\Xi} = -z\mathbf{I} - \mathcal{D}(e_i(z))$. Since $-z - \mathbb{E}\operatorname{tr}\left(\frac{\mathbf{\Sigma}_i\mathbf{Q}}{n}\right)$ is away from zero for $z \in \mathbb{C}^+$ so is $-z - e_i(z)$ and thus $\mathbf{\Xi}$ is invertible and bounded. For large n, we can write for a given deterministic matrix $\mathbf{C}$ of bounded norm

$$\begin{aligned}\mathbb{E}\left[\frac{1}{n}\operatorname{tr}\mathbf{C}\mathbf{Q}\right] &= \frac{1}{n}\sum_{i,j}(\mathbf{C}\mathbf{\Xi}^{-1})_{ji}\mathbb{E}\left(\mathbf{\Xi}\mathbf{Q}\right)_{ij} \\ &\overset{(1)}{=} \frac{1}{n}\sum_{i,j}(\mathbf{C}\mathbf{\Xi}^{-1})_{ji}\left(\mathbb{E}\left[\mathbf{Q}\frac{\mathbf{\Sigma}_i}{n}\mathbf{Q}\right]_{ij} + \delta_{ij}\right) + o(1) \\ &= \frac{1}{n}\operatorname{tr}\mathbb{E}\left(\mathbf{C}\mathbf{\Xi}^{-1}\mathbf{Q}\frac{\mathbf{\Sigma}_i}{n}\mathbf{Q}\right) + \frac{1}{n}\operatorname{tr}(\mathbf{C}\mathbf{\Xi}^{-1}) + o(1)\end{aligned}$$

where (1) follows from Equation (59). We can then prove that $\frac{1}{n}\operatorname{tr}\mathbb{E}\left(\mathbf{C}\mathbf{\Xi}^{-1}\mathbf{Q}\frac{\mathbf{\Sigma}_i}{n}\mathbf{Q}\right) = \mathcal{O}(n^{-1})$ using a similar approach to the proof of Lemma 29. Hence for large n

$$\mathbb{E}\left[\frac{1}{n}\operatorname{tr}\mathbf{C}\mathbf{Q}\right] = \frac{1}{n}\operatorname{tr}(\mathbf{C}\mathbf{\Xi}^{-1}) + o(1). \tag{60}$$

Similarly, for any vectors $\mathbf{a}$, $\mathbf{b}$ of bounded norms, we may write

$$\begin{aligned}\mathbb{E}\left[\mathbf{a}^*\mathbf{Q}\mathbf{b}\right] &= \sum_{i,j}(\mathbf{a}^*\mathbf{\Xi}^{-1})_i\mathbb{E}\left(\mathbf{\Xi}\mathbf{Q}\right)_{ij}\mathbf{b}_j \\ &= \sum_{i,j}(\mathbf{a}^*\mathbf{\Xi}^{-1})_i\mathbb{E}\left[\mathbf{Q}\frac{\mathbf{\Sigma}_i}{n}\mathbf{Q}\right]_{ij}\mathbf{b}_j + \mathbf{a}^*\mathbf{\Xi}^{-1}\mathbf{b} + o(1).\end{aligned}$$

We also have that $\sum_{i,j}(\mathbf{a}^*\mathbf{\Xi}^{-1})_i\mathbb{E}\left[\mathbf{Q}\frac{\mathbf{\Sigma}_i}{n}\mathbf{Q}\right]_{ij}\mathbf{b}_j = \mathcal{O}(n^{-1})$. This can be proved similarly to the proof of Lemma 29. Hence,

$$\mathbb{E}\left[\mathbf{a}^*\mathbf{Q}\mathbf{b}\right] = \mathbf{a}^*\mathbf{\Xi}^{-1}\mathbf{b} + o(1). \tag{61}$$

Appendix D. Second deterministic equivalents

Our goal is to find a deterministic equivalent to the random quantity $\mathbf{Q}_{z_1}^{\alpha}\mathbf{\Xi}\mathbf{Q}_{z_2}^{\alpha}$ for any diagonal deterministic matrix $\mathbf{\Xi}$ where we recall that $\mathbf{Q}_{z_1}^{\alpha} = \left(\frac{\bar{\mathbf{X}}}{\sqrt{n}} - z_1\mathbf{I}_n\right)^{-1}$ with $\bar{\mathbf{X}}$ defined previously in Appendix C. The proof follows the same techniques as the proof of the first deterministic equivalent $\mathbf{Q}_z^{\alpha}$ in Appendix C but here, the resolvent identity is either applied on $\mathbf{Q}_{z_1}^{\alpha}$ or $\mathbf{Q}_{z_2}^{\alpha}$. The technical details will be omitted as the key techniques have already been developped in

Appendix C. For the sake of readability, we will denote $\mathbf{Q}^{\alpha}_{z_1} \equiv \mathbf{Q}_1$ and $\mathbf{Q}^{\alpha}_{z_2} \equiv \mathbf{Q}_2$. As in Appendix C, we will evaluate $\mathbb{E}(\mathbf{Q}_1\mathbf{\Xi}\mathbf{Q}_2)$. By the resolvent identity, we have

$$\begin{aligned}\mathbb{E}(\mathbf{Q}_1\mathbf{\Xi}\mathbf{Q}_2)_{ij} &= -\frac{1}{z_1}\mathbb{E}(\mathbf{\Xi}\mathbf{Q}_2)_{ij} + \frac{1}{z_1}\mathbb{E}(\mathbf{X}\mathbf{Q}_1\mathbf{\Xi}\mathbf{Q}_2)_{ij} \\ &= -\frac{1}{z_1}\Xi_{ii}\mathbb{E}(\mathbf{Q}_2)_{ij} + \frac{1}{z_1}\mathbb{E}\sum_{k,l} X_{ik}(Q_1)_{kl}\Xi_{ll}(Q_2)_{lj}.\end{aligned}$$

We have from Lemma 26, $\mathbb{E}\sum_{k,l} X_{ik}(Q_1)_{kl}\Xi_{ll}(Q_2)_{lj} = \sum_{k,l}\frac{\sigma_{ik}}{\sqrt{n}}\Xi_{ll}\mathbb{E}\frac{\partial[(Q_1)_{kl}(Q_2)_{lj}]}{\partial Z_{ik}}$. By expanding all terms and all calculus done, we obtain

$$\begin{aligned}\mathbb{E}(\mathbf{Q}_1\mathbf{\Xi}\mathbf{Q}_2)_{ij} = -\frac{1}{z_1}\mathbb{E}(\mathbf{\Xi}\mathbf{Q}_2)_{ij} - \frac{1}{z_1}\sum_{k,l}\frac{\sigma^2_{ik}}{n}\Xi_{ll}\mathbb{E}\Bigg[&\underbrace{(Q_1)_{ki}(Q_1)_{kl}(Q_2)_{lj}}_{(1)} + \underbrace{(Q_1)_{kk}(Q_1)_{il}(Q_2)_{lj}}_{(2)} \\ &+ \underbrace{(Q_1)_{kl}(Q_2)_{li}(Q_2)_{kj}}_{(3)} + \underbrace{(Q_1)_{kl}(Q_2)_{lk}(Q_2)_{ij}}_{(4)}\Bigg].\end{aligned}$$

Asymptotically, the non vanishing terms are (2) and (4) so that

$$\begin{aligned}\mathbb{E}(\mathbf{Q}_1\mathbf{\Xi}\mathbf{Q}_2)_{ij} &= -\frac{1}{z_1}\mathbb{E}(\mathbf{\Xi}\mathbf{Q}_2)_{ij} - \frac{1}{z_1}\sum_{k,l}\frac{\sigma^2_{ik}}{n}\Xi_{ll}\mathbb{E}\left[(Q_1)_{kk}(Q_1)_{il}(Q_2)_{lj} + (Q_1)_{kl}(Q_2)_{lk}(Q_2)_{ij}\right] \\ &\quad + o(1) \\ &= -\frac{1}{z_1}\mathbb{E}(\mathbf{\Xi}\mathbf{Q}_2)_{ij} - \frac{1}{z_1}\frac{1}{n}\mathbb{E}\left[\mathrm{tr}(\mathbf{\Sigma}_i\mathbf{Q}_1)(\mathbf{Q}_1\mathbf{\Xi}\mathbf{Q}_2)_{ij}\right] - \frac{1}{z_1}\frac{1}{n}\mathbb{E}\left[\mathrm{tr}(\mathbf{\Sigma}_i\mathbf{Q}_2\mathbf{\Xi}\mathbf{Q}_1)(\mathbf{Q}_2)_{ij}\right] \\ &\quad + o(1). \qquad (62)\end{aligned}$$

Similarly to what was done in the proof of of Lemma 29, we can show that $\mathbb{E}\frac{1}{n}\mathrm{tr}(\mathbf{\Sigma}_i\mathbf{Q}_1)\mathbb{E}(\mathbf{Q}_1\mathbf{\Xi}\mathbf{Q}_2)_{ij} = \mathbb{E}\left(\frac{1}{n}\mathrm{tr}(\mathbf{\Sigma}_i\mathbf{Q}_1)\right)\mathbb{E}\left((\mathbf{Q}_1\mathbf{\Xi}\mathbf{Q}_2)_{ij}\right)+o(1)$. We can then write from (62)

$$\begin{aligned}&\mathbb{E}\left(\left(\mathbf{I}_n + \frac{1}{z_1}\mathcal{D}\left(\frac{1}{n}\mathrm{tr}(\mathbf{\Sigma}_i\mathbf{Q}_1)\right)_{i=1}^n\right)\mathbf{Q}_1\mathbf{\Xi}\mathbf{Q}_2\right) = \\ &-\frac{1}{z_1}\mathbb{E}\left(\mathbf{\Xi} + \mathcal{D}\left(\frac{1}{n}\mathrm{tr}(\mathbf{\Sigma}_i\mathbf{Q}_2\mathbf{\Xi}\mathbf{Q}_1)\right)_{i=1}^n\right)\mathbf{Q}_2 + o(1). \qquad (63)\end{aligned}$$

From (63) and the result of Lemma 16, this entails

$$\mathbb{E}\left(\mathbf{Q}_1\mathbf{\Xi}\mathbf{Q}_2\right) \longleftrightarrow \bar{\mathbf{Q}}_1\mathbf{\Xi}\bar{\mathbf{Q}}_2 + \bar{\mathbf{Q}}_1\mathcal{D}\left(\mathbb{E}\frac{1}{n}\mathrm{tr}(\mathbf{\Sigma}_i\mathbf{Q}_2\mathbf{\Xi}\mathbf{Q}_1)\right)_{i=1}^n\bar{\mathbf{Q}}_2. \qquad (64)$$

Every object in (64) is known but $\mathbb{E}\frac{1}{n}\mathrm{tr}(\mathbf{\Sigma}_i\mathbf{Q}_2\mathbf{\Xi}\mathbf{Q}_1)$ which we need to evaluate now. By left-multiplying (64) by $\mathbf{\Sigma}_i$ and taking the normalized trace, we get

$$\mathbb{E}\frac{1}{n}\mathrm{tr}(\mathbf{\Sigma}_i\mathbf{Q}_2\mathbf{\Xi}\mathbf{Q}_1) = \frac{1}{n}\mathrm{tr}\left(\mathbf{\Sigma}_i\bar{\mathbf{Q}}_1\mathbf{\Xi}\bar{\mathbf{Q}}_2\right) + \mathbb{E}\frac{1}{n}\mathrm{tr}\left(\mathbf{\Sigma}_i\bar{\mathbf{Q}}_1\mathcal{D}\left(\frac{1}{n}\mathrm{tr}(\mathbf{\Sigma}_i\mathbf{Q}_2\mathbf{\Xi}\mathbf{Q}_1)\right)_{i=1}^n\bar{\mathbf{Q}}_2\right). \qquad (65)$$

By denoting $f_i = \frac{1}{n}\mathbb{E}\left(\mathrm{tr}(\mathbf{\Sigma}_i\mathbf{Q}_2\mathbf{\Xi}\mathbf{Q}_1)\right)$, Equation (65) leads to

$$\mathbf{f} = \left\{\frac{1}{n}\mathrm{tr}\left(\mathbf{\Sigma}_i\bar{\mathbf{Q}}_1\mathbf{\Xi}\bar{\mathbf{Q}}_2\right)\right\}_{i=1}^n + \frac{1}{n}\left\{\left(\bar{\mathbf{Q}}_2\mathbf{\Sigma}_i\bar{\mathbf{Q}}_1\right)_{jj}\right\}_{i,j=1}^n\mathbf{f}$$

which finally entails

$$\mathbf{f} = \left(\mathbf{I}_n - \frac{1}{n}\left\{\left(\bar{\mathbf{Q}}_2\mathbf{\Sigma}_i\bar{\mathbf{Q}}_1\right)_{jj}\right\}_{i,j=1}^n\right)^{-1}\frac{1}{n}\left\{\bar{\mathbf{Q}}_2\mathbf{\Sigma}_i\bar{\mathbf{Q}}_1\right\}_{i,j=1}^n\mathrm{diag}(\mathbf{\Xi}).$$

To complete the proof of Lemma 21, we need to show that $\mathrm{Var}\left(\frac{1}{n}\mathrm{tr}(\mathbf{Q}_1\mathbf{\Xi}\mathbf{Q}_2)\right)$ and $\mathrm{Var}\left(\frac{1}{n}\mathrm{tr}(\mathbf{\Sigma}_i\mathbf{Q}_2\mathbf{\Xi}\mathbf{Q}_1)\right)$ are asymptotically summable so that by the Borell Cantelli Lemma, $\frac{1}{n}\mathrm{tr}(\mathbf{Q}_1\mathbf{\Xi}\mathbf{Q}_2)$ and $\frac{1}{n}\mathrm{tr}(\mathbf{\Sigma}_i\mathbf{Q}_2\mathbf{\Xi}\mathbf{Q}_1)$ converge respectively almost surely to their expectations. Those follow directly by using Nash Poincare inequality (Lemma 27) similarly to what was done in the proof of Lemma 29.

Acknowledgements

This work is supported by the ANR RMT4GRAPH Project (ANR-14-CE28-0006).

References

Lada A Adamic and Natalie Glance. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43. ACM, 2005.

Oskari Ajanki, Laszlo Erdos, and Torben Krüger. Quadratic vector equations on complex upper half-plane. *arXiv preprint arXiv:1506.05095*, 2015.

Zhi-Dong Bai and Jack W Silverstein. No eigenvalues outside the support of the limiting spectral distribution of large-dimensional sample covariance matrices. *Annals of probability*, pages 316–345, 1998.

Jinho Baik and Jack W Silverstein. Eigenvalues of large sample covariance matrices of spiked population models. *Journal of Multivariate Analysis*, 97(6):1382–1408, 2006.

Jinho Baik, Gérard Ben Arous, and Sandrine Péché. Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices. *Annals of Probability*, pages 1643–1697, 2005.

Florent Benaych-Georges and Raj Rao Nadakuditi. The singular values and vectors of low rank perturbations of large rectangular random matrices. *Journal of Multivariate Analysis*, 111:120–135, 2012.

Patrick Billingsley. Probability and measure. wiley series in probability and mathematical statistics, 1995.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

Francois Chapon, Romain Couillet, Walid Hachem, and Xavier Mestre. The outliers among the singular values of large rectangular random matrices with additive fixed rank deformation. *arXiv preprint arXiv:1207.0471*, 2012.

Yudong Chen, Xiaodong Li, and Jiaming Xu. Convexified modularity maximization for degree-corrected stochastic block models. *arXiv preprint arXiv:1512.08425*, 2015.

Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

Amin Coja-Oghlan and André Lanka. Finding planted partitions in random graphs with general degree distributions. *SIAM Journal on Discrete Mathematics*, 23(4):1682–1714, 2009.

Romain Couillet, Florent Benaych-Georges, et al. Kernel spectral clustering of large dimensional data. *Electronic Journal of Statistics*, 10(1):1393–1454, 2016.

Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Inference and phase transitions in the detection of modules in sparse networks. *Physical Review Letters*, 107(6):065701, 2011.

Jordi Duch and Alex Arenas. Community detection in complex networks using extremal optimization. *Physical review E*, 72(2):027104, 2005.

Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

Chao Gao, Zongming Ma, Anderson Y Zhang, and Harrison H Zhou. Community detection in degree-corrected block models. *arXiv preprint arXiv:1607.06993*, 2016.

Roger Guimera, Marta Sales-Pardo, and Luís A Nunes Amaral. Modularity from fluctuations in random graphs and complex networks. *Physical Review E*, 70(2):025101, 2004.

Lennart Gulikers, Marc Lelarge, and Laurent Massoulié. A spectral method for community detection in moderately-sparse degree-corrected stochastic block models. *arXiv preprint arXiv:1506.08621*, 2015.

Walid Hachem, Philippe Loubaton, Jamal Najim, et al. Deterministic equivalents for certain functionals of large random matrices. *The Annals of Applied Probability*, 17(3):875–930, 2007.

Walid Hachem, Philippe Loubaton, Xavier Mestre, Jamal Najim, and Pascal Vallet. A subspace estimator for fixed rank perturbations of large random matrices. *Journal of Multivariate Analysis*, 114:427–447, 2013.

John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

Matthew B Hastings. Community detection as an inference problem. *Physical Review E*, 74 (3):035102, 2006.

Jiashun Jin et al. Fast community detection by score. *The Annals of Statistics*, 43(1):57–89, 2015.

Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107, 2011.

Florent Krzakala, Cristopher Moore, Elchanan Mossel, Joe Neeman, Allan Sly, Lenka Zdeborová, and Pan Zhang. Spectral redemption in clustering sparse networks. *Proceedings of the National Academy of Sciences*, 110(52):20935–20940, 2013.

Jing Lei, Alessandro Rinaldo, et al. Consistency of spectral clustering in stochastic block models. *The Annals of Statistics*, 43(1):215–237, 2015.

Vince Lyzinski, Daniel L Sussman, Minh Tang, Avanti Athreya, Carey E Priebe, et al. Perfect clustering for stochastic blockmodel graphs via adjacency spectral embedding. *Electronic Journal of Statistics*, 8(2):2905–2922, 2014.

Raj Rao Nadakuditi and Mark EJ Newman. Graph spectra and the detectability of community structure in networks. *Physical review letters*, 108(18):188701, 2012.

Mark EJ Newman. Fast algorithm for detecting community structure in networks. *Physical review E*, 69(6):066133, 2004.

Mark EJ Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006a.

Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006b.

Mark EJ Newman and Elizabeth A Leicht. Mixture models and exploratory analysis in networks. *Proceedings of the National Academy of Sciences*, 104(23):9564–9569, 2007.

MEJ Newman. Spectral community detection in sparse networks. *arXiv preprint arXiv:1308.6494*, 2013.

MEJ Newman. Community detection in networks: Modularity optimization and maximum likelihood are equivalent. *arXiv preprint arXiv:1606.02319*, 2016.

Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.

Shu Kay Ng, Thriyambakam Krishnan, and Geoffrey J McLachlan. The em algorithm. In *Handbook of computational statistics*, pages 139–172. Springer, 2012.

Leonid Andreevich Pastur, Mariya Shcherbina, and Mariya Shcherbina. *Eigenvalue distribution of large random matrices*, volume 171. American Mathematical Society Providence, RI, 2011.

Tai Qin and Karl Rohe. Regularized spectral clustering under the degree-corrected stochastic blockmodel. In *Advances in Neural Information Processing Systems*, pages 3120–3128, 2013.

Alaa Saade, Florent Krzakala, and Lenka Zdeborová. Spectral clustering of graphs with the bethe hessian. In *Advances in Neural Information Processing Systems*, pages 406–414, 2014.

Jack W Silverstein and ZD Bai. On the empirical distribution of eigenvalues of a class of large dimensional random matrices. *Journal of Multivariate analysis*, 54(2):175–192, 1995.

Hafiz Tiomoko Ali and Romain Couillet. Performance analysis of spectral community detection in realistic graph models. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP16)*, 2016.

Hafiz Tiomoko Ali and Romain Couillet. Improved spectral community detection algorithm in large dense graphs. `https://github.com/hafizTiomoko/improved_spectral_community_detection`, March 2017.

Yunpeng Zhao, Elizaveta Levina, Ji Zhu, et al. Consistency of community detection in networks under degree-corrected stochastic block models. *The Annals of Statistics*, 40 (4):2266–2292, 2012.

Journal of Machine Learning Research 18 (2018) 1-92 Submitted 8/17; Revised 8/17; Published 5/18

Statistical Inference on Random Dot Product Graphs: a Survey

Avanti Athreya dathrey1@jhu.edu
Donniell E. Fishkind def@jhu.edu
Minh Tang mtang10@jhu.edu
Carey E. Priebe cep@jhu.edu
Department of Applied Mathematics and Statistics
Johns Hopkins University
Baltimore, MD, 2128, USA

Youngser Park youngser@jhu.edu
Center for Imaging Science
Johns Hopkins University
Baltimore, MD, 21218, USA

Joshua T. Vogelstein jovo@jhu.edu
Department of Biomedical Engineering
Johns Hopkins University
Baltimore, MD, 21218, USA

Keith Levin klevin@umich.edu
Department of Statistics
University of Michigan
Ann Arbor, MI, 48109, USA

Vince Lyzinski vlyzinski@umass.edu
Department of Mathematics and Statistics
University of Massachusetts
Amherst, MA, 01003-9305, USA

Yichen Qin qinyn@ucmail.uc.edu
Department of Operations, Business Analytics, and Information Systems, College of Business
University of Cincinnati
Cincinnati, OH, 45221-0211, USA

Daniel L Sussman sussman@bu.edu
Department of Mathematics and Statistics
Boston University
Boston, MA, 02215, USA

Editor: Edoardo M. Airoldi

Abstract

The random dot product graph (RDPG) is an independent-edge random graph that is analytically tractable and, simultaneously, either encompasses or can successfully approximate a wide range of random graphs, from relatively simple stochastic block models to complex latent position graphs. In this survey paper, we describe a comprehensive paradigm for statistical inference on random dot product graphs, a paradigm centered on spectral embeddings of adjacency and Laplacian matrices. We examine the graph-inferential analogues of several canonical tenets of classical Euclidean inference. In particular, we summarize a body of existing results on the consistency and asymptotic normality of the adjacency and Laplacian spectral embeddings, and the role these spectral embeddings can play in the construction of single- and multi-sample hypothesis tests for graph data. We investigate several real-world applications, including community detection and classification in large social networks and the determination of functional and biologically relevant network properties from an exploratory data analysis of the *Drosophila* connectome. We outline requisite background and current open problems in spectral graph inference.

Keywords: Random dot product graph, adjacency spectral embedding, Laplacian spectral embedding, multi-sample graph hypothesis testing, semiparametric modeling

1. Introduction

Random graph inference is an active, interdisciplinary area of current research, bridging combinatorics, probability, statistical theory, and machine learning, as well as a wide spectrum of application domains from neuroscience to sociology. Statistical inference on random graphs and networks, in particular, has witnessed extraordinary growth over the last decade: see, for example, Goldenberg et al. (2010) and Kolaczyk (2009) for a discussion of the considerable applications in recent network science of several canonical random graph models.

Of course, combinatorial graph theory itself is centuries old—indeed, in his resolution to the problem of the bridges of Königsberg, Leonard Euler first formalized graphs as mathematical objects consisting of vertices and edges. The notion of a random graph, however, and the modern theory of inference on such graphs, is comparatively new, and owes much to the pioneering work of Erdős, Rényi, and others in the late 1950s. E.N. Gilbert's short 1959 paper (Gilbert, 1959) considered a random graph for which the existence of edges between vertices are independent Bernoulli random variables with common probability p; roughly concurrently, Erdős and Rényi provided the first detailed analysis of the probabilities of the emergence of certain types of subgraphs within such graphs (Erdős and Rényi, 1960), and today, graphs in which the edges arise independently and with common probability p are known as *Erdős-Rényi* (or ER) graphs.

The Erdős-Rényi (ER) model is one of the simplest generative models for random graphs, but this simplicity belies astonishingly rich behavior (see Alon and Spencer, 2008; Bollobás et al., 2007). Nevertheless, in many applications, the requirement of a common connection probability is too stringent: graph vertices often represent heterogeneous entities, such as different people in a social network or cities in a transportation graph, and the connection probability p_{ij} between vertex i and j may well change with i and j or depend on underlying attributes of the vertices. Moreover, these heterogeneous vertex attributes may not be observable; for example, given the adjacency matrix of a Facebook community, the specific interests of the individuals may remain hidden. To more effectively model such real-world

networks, we consider *latent position* random graphs (Hoff et al., 2002). In a latent position graph, to each vertex i in the graph there is associated an element x_i of the so-called *latent space* $\mathcal{X}$, and the probability of connection p_{ij} between any two edges i and j is given by a *link* or *kernel* function $\kappa : \mathcal{X} \times \mathcal{X} \to [0, 1]$. That is, the edges are generated independently (so the graph is an *independent-edge* graph) and $p_{ij} = \kappa(x_i, x_j)$.

The *random dot product graph* (RDPG) of Young and Scheinerman (Young and Scheinerman, 2007) is an especially tractable latent position graph; here, the latent space is an appropriately constrained subspace of Euclidean space $\mathbb{R}^d$, and the link function is simply the dot or inner product of the pair of d-dimensional latent positions. Thus, in a d-dimensional random dot product graph with n vertices, the latent positions associated to the vertices can be represented by an $n \times d$ matrix $\mathbf{X}$ whose rows are the latent positions, and the matrix of connection probabilities $\mathbf{P} = (\mathbf{P}_{ij})$ is given by $\mathbf{P} = \mathbf{X}\mathbf{X}^\top$. Conditional on this matrix $\mathbf{P}$, the RDPG has an adjacency matrix $\mathbf{A} = (\mathbf{A}_{ij})$ whose entries are Bernoulli random variables with probability $\mathbf{P}_{ij}$. For simplicity, we will typically consider symmetric, *hollow* RDPG graphs; that is, undirected, unweighted graphs in which $\mathbf{A}_{ii} = 0$, so there are no self-edges. In our real data analysis of a neural connectome in Section 6.3, however, we describe how to adapt our results to weighted and directed graphs.

In any latent position graph, the latent positions associated to graph vertices can themselves be random; for instance, the latent positions may be independent, identically distributed random variables with some distribution F on $\mathbb{R}^d$. The well-known *stochastic blockmodel* (SBM), in which each vertex belongs to one of K subsets known as *blocks*, with connection probabilities determined solely by block membership (Holland et al., 1983), can be represented as a random dot product graph in which all the vertices in a given block have the same latent positions (or, in the case of random latent positions, an RDPG for which the distribution F is supported on a finite set). Despite their structural simplicity, stochastic block models are the building blocks for all independent-edge random graphs; in Wolfe and Olhede (2013), the authors demonstrate that any independent-edge random graph can be well-approximated by a stochastic block model with a sufficiently large number of blocks. Since stochastic block models can themselves be viewed as random dot product graphs, we see that suitably high-dimensional random dot product graphs can provide accurate approximations of latent position graphs (Tang et al., 2013), and, in turn, independent-edge graphs. Thus, the architectural simplicity of the random dot product graph makes it particularly amenable to analysis, and its near-universality in graph approximation renders it expansively applicable. In addition, the cornerstone of our analysis of random dot product graphs is a set of classical probabilistic and linear algebraic techniques that are useful in much broader settings, such as random matrix theory. As such, the random dot product graph is both a rich and interesting object of study in its own right and a natural point of departure for wider graph inference.

A classical inference task for Euclidean data is to estimate, from sample data, certain underlying distributional parameters. Similarly, for a latent position graph, a classical graph inference task is to infer the graph parameters from an observation of the adjacency matrix $\mathbf{A}$. Indeed, our overall paradigm for random graph inference is inspired by the fundamental tenets of classical statistical inference for Euclidean data. Namely, our goal

is to construct methods and estimators of graph parameters or graph distributions; and, for these estimators, to analyze their (1) consistency; (2) asymptotic distributions; (3) asymptotic relative efficiency; (4) robustness to model misspecification; and (5) implications for subsequent inference including one- and multi-sample hypothesis testing. In this paper, we summarize and synthesize a considerable body of work on spectral methods for inference in random dot product graphs, all of which not only advance fundamental tenets of this paradigm, but do so within a unified and parsimonious framework. The random graph estimators and test statistics we discuss all exploit the *adjacency spectral embedding* (ASE) or the *Laplacian spectral embedding* (LSE), which are eigendecompositions of the adjacency matrix $\mathbf{A}$ and *normalized* Laplacian matrix $\mathbf{L} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, where $\mathbf{D}$ is the diagonal degree matrix $\mathbf{D}_{ii} = \sum_{j \neq i} \mathbf{A}_{ij}$.

The ambition and scope of our approach to graph inference means that mere upper bounds on discrepancies between parameters and their estimates will not suffice. Such bounds are legion. In our proofs of consistency, we improve several bounds of this type, and in some cases improve them so drastically that concentration inequalities and asymptotic limit distributions emerge in their wake. We stress that aside from specific cases (see Füredi and Komlós, 1981; Tao and Vu, 2012; Lei, 2016), limiting distributions for eigenvalues and eigenvectors of random graphs are notably elusive. For the adjacency and Laplacian spectral embedding, we discuss not only consistency, but also asymptotic normality, robustness, and the use of the adjacency spectral embedding in the nascent field of multi-graph hypothesis testing. We illustrate how our techniques can be meaningfully applied to thorny and very sizable real data, improving on previously state-of-the-art methods for inference tasks such as community detection and classification in networks. What is more, as we now show, spectral graph embeddings are relevant to many complex and seemingly disparate aspects of graph inference.

A bird's-eye view of our methodology might well start with the stochastic blockmodel. For an SBM with a finite number of blocks of stochastically equivalent vertices, in Sussman et al. (2012) and Fishkind et al. (2013), we establish that k-means clustering of the rows of the adjacency spectral embedding accurately partitions the vertices into the correct blocks, even when the embedding dimension is misspecified or the number of blocks is unknown. Furthermore, in Lyzinski et al. (2014) and Lyzinski et al. (2017) we give a significant improvement in the misclassification rate, by exhibiting an almost-surely perfect clustering in which, in the limit, no vertices whatsoever are misclassified. For random dot product graphs more generally, we show in Sussman et al. (2014) that the latent positions are consistently estimated by the embedding, which then allows for accurate learning in a supervised vertex classification framework. In Tang et al. (2013), these results are extended to more general latent position models, establishing a powerful universal consistency result for vertex classification in general latent position graphs, and also exhibiting an efficient embedding of vertices which were not observed in the original graph. In Athreya et al. (2016) and Tang and Priebe (2018), we supply distributional results, akin to a central limit theorem, for both the adjacency and Laplacian spectral embedding, respectively; the former leads to a nontrivially superior algorithm for the estimation of block memberships in a stochastic block model (Suwan et al., 2016), and the latter resolves, through an elegant

comparison of Chernoff information, a long-standing open question of the relative merits of the adjacency and Laplacian graph representations.

Morever, graph embedding plays a central role in the foundational work on hypothesis testing of Tang et al. (2017a) and Tang et al. (2017b) for two-sample graph comparison: these papers provide theoretically justified, valid and consistent hypothesis tests for the semiparamatric problem of determining whether two random dot product graphs have the same latent positions and the nonparametric problem of determining whether two random dot product graphs have the same underlying distributions. This, then, yields a systematic framework for determining statistical similarity across graphs, which in turn underpins yet another provably consistent algorithm for the decomposition of random graphs with a hierarchical structure Lyzinski et al. (2017). In Levin et al. (2017), distributional results are given for an omnibus embedding of multiple random dot product graphs on the same vertex set, and this embedding performs well both for latent position estimation and for multi-sample graph testing. For the critical inference task of vertex nomination, in which the inference goal is to produce an ordering of vertices of interest (see, for instance Coppersmith, 2014), we find in Fishkind et al. (2015a) an array of principled vertex nomination algorithms —-the canonical, maximum likelihood and spectral vertex nomination schemes—and a demonstration of the algorithms' effectiveness on both synthetic and real data. In Lyzinski et al. (2016b) the consistency of the maximum likelihood vertex nomination scheme is established, a scalable restricted version of the algorithm is introduced, and the algorithms are adapted to incorporate general vertex features.

Overall, we stress that these principled techniques for random dot product graphs exploit the Euclidean nature of graph embeddings but are general enough to yield meaningful results for a wide variety of random graphs. Because our focus is, in part, on spectral methods, and because the adjacency matrix $\mathbf{A}$ of an independent-edge graph can be regarded as a noisy version of the matrix of probabilities $\mathbf{P}$ (Oliveira, 2009), we rely on several classical results on matrix perturbations, most prominently the Davis-Kahan Theorem (see Bhatia (1997) for the theorem itself, Rohe et al. (2011) for an illustration of its role in graph inference, and Yu et al. (2015) for a very useful variant). We also depend on the aforementioned spectral bounds in Oliveira (2009) and a more recent sharpening due to Lu and Peng (Lu and Peng, 2013). We leverage probabilistic concentration inequalities, such as those of Hoeffding and Bernstein (Tropp, 2015). Finally, several of our results do require suitable eigengaps for $\mathbf{P}$ and lower bounds on graph density, as measured by the maximum degree and the size of the smallest eigenvalue of $\mathbf{P}$. It is important to point out that in our analysis, we assume that the embedding dimension d of our graphs is known and fixed. In real data applications, such an embedding dimension is not known, and in Section 6.3, we discuss approaches (see Chatterjee, 2015; Zhu and Ghodsi, 2006) to estimating the embedding dimension. Robustness of our procedures to errors in embedding dimension is a problem of current investigation.

In the study of stochastic blockmodels, there has been a recent push to understand the fundamental information-theoretic limits for community detection and graph partitioning (see Mossel et al., 2018, 2014; Abbe and Sandon, 2015; Abbe et al., 2016). These bounds are typically algorithm-free and focus on stochastic blockmodels with constant or logarithmic

average degree, in which differences between vertices in different blocks are assumed to be at the boundary of detectability. Our efforts have a somewhat different flavor, in that we seek to understand the precise behavior of a widely applicable procedure in a more general model. Additionally, we treat sparsity as a secondary concern, and typically do not broach the question of the exact limits of our procedures. Our spectral methods may not be optimal for all random graph models, of course (Krzakala et al., 2013; Kawamoto and Kabashima, 2015), but they are very useful, in that they rely on well-optimized computational methods, can be implemented quickly in many standard languages, extend readily to other models, and serve as a foundation for more complex analyses.

Finally, we would be remiss not to point out that while spectral decompositions and clusterings of the adjacency matrix are appropriate for graph inference, they are also of considerable import in combinatorial graph theory: readers may recall, for instance, the combinatorial *ratio-cut* problem, whose objective is to partition the vertex set of a graph into two disjoint sets in a way that minimizes the number of edges between vertices in the two sets. The minimizer of a relaxation to the ratio-cut problem (Fiedler, 1973) is the eigenvector associated to the second smallest eigenvalue of the graph Laplacian $\mathbf{L}$. While we do not pursue more specific combinatorial applications of spectral methods here, we note that Chung (1997) provides a comprehensive overview, and von Luxburg (2007) gives an accessible tutorial on spectral methods.

We organize the paper as follows. In Section 2, we define random dot product graphs and the adjacency spectral embedding, and we recall important linear algebraic background. In Section 4, we discuss consistency, asymptotic normality, and hypothesis testing, as well as inference for hierarchical models. In Section 6, we discuss applications of these results to real data. Finally, in Section 7 we discuss current theoretical and computational difficulties and open questions, including issues of optimal embedding dimension, model limitations, robustness to errorful observations, and joint graph inference. Finally, in A, we provide further details in the proofs of our main results.

2. Definitions, notation, and background

2.1 Preliminaries and notation

We begin by establishing notation. For a positive integer n, we let $[n] = \{1, 2, \cdots, n\}$. For a vector $\mathbf{v} \in \mathbb{R}^n$, we let $\|\mathbf{v}\|$ denote the Euclidean norm of $\mathbf{v}$. We denote the identity matrix, zero matrix, and the square matrix of all ones by, $\mathbf{I}$, $\mathbf{0}$, and $\mathbf{J}$, respectively. We use $\otimes$ to denote the Kronecker product. For an $n_1 \times n_2$ matrix $\mathbf{H}$, we let $\mathbf{H}_{ij}$ denote its i, jth entry; we denote by $\mathbf{H}_{\cdot j}$ the column vector formed by the j-th column of $\mathbf{H}$; and we denote by $\mathbf{H}_{i\cdot}$ the row vector formed by the i-th row of $\mathbf{H}$. For a slight abuse of notation, we also let $\mathbf{H}_i \in \mathbb{R}^{n_2}$ denote the *column* vector formed by transposing the i-th row of $\mathbf{H}$. That is, $\mathbf{H}_i = (\mathbf{H}_{i\cdot})^\top$. Given any suitably specified ordering on eigenvalues of a square matrix $\mathbf{H}$, we let $\lambda_i(\mathbf{H})$ denote the i-th eigenvalue (under such an ordering) of $\mathbf{H}$ and $\sigma_i(\mathbf{H}) = \sqrt{\lambda_i(\mathbf{H}^\top\mathbf{H})}$ the i-th singular value of $\mathbf{H}$. We let $\|\mathbf{H}\|$ denote the spectral norm of $\mathbf{H}$ and $\|\mathbf{H}\|_F$ denote the Frobenius norm of $\mathbf{H}$. We let $\|\mathbf{H}\|_{2\to\infty}$ denote the maximum of the Euclidean norms of

the rows of $\mathbf{H}$, so that $\|\mathbf{H}\|_{2\to\infty} = \max_i \|\mathbf{H}_i\|$. We denote the trace of a matrix $\mathbf{H}$ by $\mathrm{tr}(\mathbf{H})$. For an $n \times n$ symmetric matrix $\mathbf{H}$ whose entries are all non-negative, we will frequently have to account for terms related to matrix sparsity, and we define $\delta(\mathbf{H})$ and $\gamma(\mathbf{H})$ as follows:

$$\delta(\mathbf{H}) = \max_{1\le i\le n} \sum_{j=1}^{n} \mathbf{H}_{ij} \qquad \gamma(\mathbf{H}) = \frac{\sigma_d(\mathbf{H}) - \sigma_{d+1}(\mathbf{H})}{\delta(\mathbf{H})} \tag{1}$$

(We remark that when H is a matrix of connection probabilities for a random graph, δ can be interpreted as the maximum expected degree, and γ relates this quantity to an eigengap that is especially useful in the case of probability matrices of suitably low rank.) In a number of cases, we need to consider a sequence of matrices. We will denote such a sequence by $\mathbf{H}_n$, where n is typically used to denote the index of the sequence. The distinction between a particular element $\mathbf{H}_n$ in a sequence of matrices and a particular row $\mathbf{H}_i$ of a matrix will be clear from context, and our convention is typically to use n to denote the index of a sequence and i or h to denote a particular row of a matrix. In the case where we need to consider the ith row of a matrix that is itself the nth element of a sequence, we will use the notation $(\mathbf{H}_n)_i$.

We define a *graph* G to be an ordered pair of (V, E) where V is the so-called *vertex* or *node* set, and E, the set of *edges*, is a subset of the Cartesian product of $V \times V$. In a graph whose vertex set has cardinality n, we will usually represent V as $V = \{1, 2, \cdots, n\}$, and we say there *is an edge between* i and j if $(i, j) \in E$. The *adjacency* matrix $\mathbf{A}$ provides a compact representation of such a graph:

$$\mathbf{A}_{ij} = 1 \text{ if } (i, j) \in E, \text{ and } \mathbf{A}_{ij} = 0 \text{ otherwise.}$$

Where there is no danger of confusion, we will often refer to a graph G and its adjacency matrix $\mathbf{A}$ interchangeably.

Our focus is random graphs, and thus we will let Ω denote our sample space, $\mathcal{F}$ the σ-algebra of subsets of Ω and $\mathbb{P}$ our probability measure $\mathbb{P} : \mathcal{F} \to [0, 1]$. We will denote the expectation of a (potentially multi-dimensional) random variable X with respect to this measure by $\mathbb{E}$. Given an event $F \in \mathcal{F}$, we denote its complement by F^c, and we let $\Pr(F)$ denote the probability of F. As we will see, in many cases we can choose Ω to be subset of Euclidean space. Because we are interested in large-graph inference, we will frequently need to demonstrate that probabilities of certain events decay at specified rates. This motivates the following definition.

Definition 1 (Convergence a.a.s. and w.h.p.) *Given a sequence of events $\{F_n\} \in \mathcal{F}$, where $n = 1, 2, \cdots$, we say that F_n occurs* asymptotically almost surely *(a.a.s.) if $\Pr(F_n) \to 1$ as $n \to \infty$. We say that F_n* occurs with high probability *(w.h.p.), and write F_n w.h.p. , if for any $c_0 > 1$, there exists finite positive constant C_0 depending on c_0 such that $\Pr[F_n^c] \le C_0 n^{-c_0}$ for all n. We note that F_n occurring w.h.p. is stronger than F_n occurring asymptotically almost surely. Morever, F_n occurring with high probability implies, by the Borel-Cantelli Lemma Chung (2001), that with probability 1 there exists an n_0 such that F_n holds for all $n \ge n_0$.*

Moreover, since our goal is often to understand large-graph inference, we need to consider asymptotics as a function of graph size n. As such, we recall familiar asymptotic notation:

Definition 2 (Asymptotic notation) *If $w(n)$ is a quantity depending on n, we will say that w* is of order $\alpha(n)$ *and use the notation $w(n) \sim \Theta(\alpha(n))$ to denote that there exist positive constants c, C such that for n sufficiently large,*

$$c\alpha(n) \le w(n) \le C\alpha(n).$$

When the quantity $w(n)$ is clear and $w(n) \sim \Theta(\alpha(n))$, we sometimes simply write "w is of order $\alpha(n)$". We write $w(n) \sim O(n)$ if there exists a constant C such that for n sufficiently large, $w(n) \le Cn$. We write $w(n) \sim o(n)$ if $w(n)/n \to 0$ as $n \to \infty$, and $w(n) \sim o(1)$ if $w(n) \to 0$ as $n \to \infty$. We write $w(n) \sim \Omega(n)$ if there exists a constant C such that for all n sufficiently large, $w(n) \ge Cn$.

Throughout, we will use $C > 0$ to denote a constant, not depending on n, which may vary from one line to another.

2.2 Models

Since our focus is on d-dimensional random dot product graphs, we first define an *an inner product distribution* as a probability distribution over a suitable subset of $\mathbb{R}^d$, as follows:

Definition 3 (d-dimensional inner product distribution) *Let F be a probability distribution whose support is given by $\operatorname{supp} F = \mathcal{X}_d \subset \mathbb{R}^d$. We say that F is a* d-dimensional inner product distribution *on $\mathbb{R}^d$ if for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}_d = \operatorname{supp} F$, we have $\mathbf{x}^\top \mathbf{y} \in [0, 1]$.*

Next, we define a random dot product graph as an independent-edge random graph for which the edge probabilities are given by the dot products of the latent positions associated to the vertices. We restrict our attention here to graphs that are undirected and in which no vertex has an edge to itself.

Definition 4 (RDPG with distribution F) *Let F be a d-dimensional inner product distribution with $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n \overset{i.i.d.}{\sim} F$, collected in the rows of the matrix*

$$\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n]^\top \in \mathbb{R}^{n\times d}.$$

Suppose $\mathbf{A}$ is a random adjacency matrix given by

$$\Pr[\mathbf{A}|\mathbf{X}] = \prod_{i<j} (\mathbf{X}_i^\top \mathbf{X}_j)^{\mathbf{A}_{ij}} (1 - \mathbf{X}_i^\top \mathbf{X}_j)^{1-\mathbf{A}_{ij}} \tag{2}$$

We then write $(\mathbf{A}, \mathbf{X}) \sim \mathrm{RDPG}(F, n)$ and say that $\mathbf{A}$ is the adjacency matrix of a random dot product graph (RDPG) of dimension or rank at most d *and with* latent positions *given by the rows of $\mathbf{X}$. If $\mathbf{X}\mathbf{X}^\top$ is, in fact, a rank d matrix, we say $\mathbf{A}$ is the adjacency matrix of a rank d random dot product graph.*

Next, for such random dot product graphs, we frequently need to consider the $d \times d$ *second moment matrix* Δ, defined as follows:

Definition 5 (Second moment matrix) *Suppose A is an adjacency matrix from the random dot product graph model with distribution F. Let $X_1 \sim F$ be one of the i.i.d vectors of latent positions. Define the* second moment matrix Δ *via the expectation*

$$\Delta = E[\mathbf{X}_1 \mathbf{X}_1^\top]$$

While our notation for a random dot product graph with distribution F is $(\mathbf{A}, \mathbf{X}) \sim \text{RDPG}(F)$, we emphasize that in this paper the latent positions $\mathbf{X}$ are always assumed to be unobserved. An almost identical definition holds for random dot product graphs with fixed but unobserved latent positions:

Definition 6 (RDPG with fixed latent positions) *In the definition 4 given above, the latent positions are themselves random. If, instead, the latent positions are given by a fixed matrix $\mathbf{X}$ and, given this matrix, the graph is generated according to Eq.(2), we say that $\mathbf{A}$ is a realization of a random dot product graph with latent positions $\mathbf{X}$, and we write $\mathbf{A} \sim \text{RDPG}(\mathbf{X})$.*

Remark 7 (Nonidentifiability) *Given a graph distributed as an RDPG, the natural task is to recover the latent positions $\mathbf{X}$ that gave rise to the observed graph. However, the RDPG model has an inherent nonidentifiability: let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be a matrix of latent positions and let $\mathbf{W} \in \mathbb{R}^{d \times d}$ be a unitary matrix. Since $\mathbf{X}\mathbf{X}^\top = (\mathbf{X}\mathbf{W})(\mathbf{X}\mathbf{W})^\top$, it is clear that the latent positions $\mathbf{X}$ and $\mathbf{X}\mathbf{W}$ give rise to the same distribution over graphs in Equation (2). Note that most latent position models, as defined below, also suffer from similar types of non-identifiability as edge-probabilities may be invariant to various transformations.*

As we mentioned, the random dot product graph is a specific instance of the more general *latent position random graph* with *link* or *kernel* function κ. Indeed, the latent positions themselves need not belong to Euclidean space per se, and the link function need not be an inner product.

Definition 8 (Latent position random graph with kernel κ) *Let $\mathcal{X}$ be a set and $\kappa : \mathcal{X} \times \mathcal{X} \to [0,1]$ a symmetric function. Suppose to each $i \in [n]$ there is associated a point $\mathbf{X}_i \in \mathcal{X}$. Given $\mathbf{X} = \{\mathbf{X}_1, \cdots, \mathbf{X}_n\}$ consider the graph with adjacency matrix $\mathbf{A}$ defined by*

$$\Pr[\mathbf{A}|\mathbf{X}] = \prod_{i<j} \kappa(\mathbf{X}_i, \mathbf{X}_j)^{\mathbf{A}_{ij}} (1 - \kappa(\mathbf{X}_i, \mathbf{X}_j))^{1-\mathbf{A}_{ij}} \tag{3}$$

Then $\mathbf{A}$ is the adjacency matrix of a latent position random graph with latent position $\mathbf{X}$ and link function κ.

Similarly, we can define independent edge graphs for which latent positions need not play a role.

Definition 9 (Independent-edge graphs) *For a matrix symmetric matrix $\mathbf{P}$ of probabilities, we say that $\mathbf{A}$ is distributed as an independent edge graph with probabilities $\mathbf{P}$ if*

$$\Pr[\mathbf{A}|\mathbf{X}] = \prod_{i<j} \mathbf{P}_{ij}^{\mathbf{A}_{ij}} (1 - \mathbf{P}_{ij})^{1-\mathbf{A}_{ij}} \tag{4}$$

By their very structure, latent position random graphs, for fixed latent positions, are independent-edge random graphs. In general, for any latent position graph the matrix of edge probabilities $\mathbf{P}$ is given by $\mathbf{P}_{ij} = \kappa(\mathbf{X}_i, \mathbf{X}_j)$ Of course, in the case of an random dot product graph with latent position matrix $\mathbf{X}$, the probability $\mathbf{P}_{ij}$ of observing an edge between vertex i and vertex j is simply $\mathbf{X}_i^\top \mathbf{X}_j$. Thus, for an RDPG with latent positions $\mathbf{X}$, the matrix $\mathbf{P} = [p_{ij}]$ is given by $\mathbf{P} = \mathbf{X}\mathbf{X}^\top$.

In order to more carefully relate latent position models and RDPGs, we can consider the set of positive semidefinite latent position graphs. Namely, we will say that a latent position random graph is positive semidefinite if the matrix $\mathbf{P}$ is positive semidefinite. In this case, we note that an RDPG can be used to approximate the latent position random graph distribution. (In fact, as we describe further below, new work of Rubin-Delanchy, Tang, and Priebe (Rubin-Delanchy et al., 2017) presents a generalization of the RDPG which allows us to drop the positive semidefiniteness requirement.) The best rank-d approximation of $\mathbf{P}$, in terms of the Frobenius norm (Eckart and Young, 1936), will correspond to a RDPG with d-dimensional latent positions. In this sense, by allowing d to be as large as necessary, any positive semi-definite latent position random graph distribution can be approximated by a RDPG distribution to arbitrary precision (Tang et al., 2013).

While latent position models generalize the random dot product graph, RDPGs can be easily related to the more limited *stochastic blockmodel* (SBM) graph (Holland et al., 1983). The stochastic block model is also an independent-edge random graph whose vertex set is partitioned into K groups, called *blocks*, and the stochastic blockmodel is typically parameterized by (1) a $K \times K$ matrix of probabilities $\mathbf{B}$ of adjacencies between vertices in each of the blocks, and (2) a *block-assignment vector* $\tau : [n] \to [K]$ which assigns each vertex to its block. That is, for any two vertices i, j, the probability of their connection is

$$\mathbf{P}_{ij} = \mathbf{B}_{\tau(i),\tau(j)},$$

and we typically write $\mathbf{A} \sim \text{SBM}(\mathbf{B}, \tau)$. Here we present an alternative definition in terms of the RDPG model.

Definition 10 (Positive semidefinite k-block SBM) *We say an RDPG with latent positions $\mathbf{X}$ is an SBM with K blocks if the number of distinct rows in $\mathbf{X}$ is K, denoted $\mathbf{X}_{(1)}, \cdots, \mathbf{X}_{(K)}$ In this case, we define the block membership function $\tau : [n] \mapsto [K]$ to be a function such that $\tau(i) = \tau(j)$ if and only if $\mathbf{X}_i = \mathbf{X}_j$. We then write*

$$\mathbf{A} \sim \text{SBM}(\tau, \{\mathbf{X}_{(i)}\}_{i=1}^K)$$

In addition, we also consider the case of a stochastic block model in which the block memberships of each vertex is randomly assigned. More precisely, let $\pi \in (0,1)^K$ with $\sum_{k=1}^n \pi_k = 1$ and suppose that $\tau(1), \tau(2), \ldots, \tau(n)$ are now i.i.d. random variables with distribution Categorical(π)*; that is,* $\Pr(\tau(i) = k) = \pi_k$ *for all k. Then we say $\mathbf{A}$ is an* SBM with i.i.d block memberships*, and we write*

$$\mathbf{A} \sim \text{SBM}(\pi, \{X_{(i)}\}).$$

We also consider the *degree-corrected* stochastic block model (Karrer and Newman, 2011):

Definition 11 (Degree Corrected SBM) *We say an RDPG is a degree-corrected stochastic block model (DCSBM) with K blocks if there exist K unit vectors $y_1, \ldots, y_K \in \mathbb{R}^d$ such that for each $i \in [n]$, there exists $k \in [K]$ and $c_i \in (0,1)$ such that $X_i = c_i y_k$.*

Remark 12 *The degree-corrected stochastic blockmodel model is inherently more flexible than the standard SBM because it allows for vertices within each block/community to have different expected degrees. This flexibility has made it a popular choice for modeling network data (again, see Karrer and Newman, 2011).*

For an important generalization that offers more flexibility in block memberships, we consider the *mixed-membership stochastic block model* (MMSBM) of Airoldi et al. (2008):

Definition 13 (Mixed Membership SBM) *We say an RDPG is a mixed membership stochastic block model (MMSBM) with K blocks if there exists K unit vectors $y_1, \dots, y_K \in \mathbb{R}^d$ such that for each $i \in [n]$, there exists $\alpha_1, \dots, \alpha_K > 0$ such that $\sum_{k=1}^K \alpha_k = 1$ and $X_i = \sum_{k=1}^K \alpha_k y_k$.*

Remark 14 *The mixed membership SBM allows for each vertex to be in a mixture of different blocks. Additionally, note that every RDPG is a MMSBM for some choice of K.*

Our next theorem summarizes the relationship between these models.

Theorem 15 *Considered as statistical models for graphs—to wit, sets of probability distributions on graphs—the positive-semidefinite K-block SBM is a subset of the K-block DCSBM and the K-block MMSBM. Both the positive semidefinite K-block DCSBM and K-block MMSBM are subsets of the RDPG model with K-dimensional latent positions. Finally, the union of all possible RDPG models, without restriction of latent position dimension, is dense in the set of positive semidefinite latent position models.*

We emphasize that, as mentioned previously, very recent work of Rubin-Delanchy, Priebe, and Tang (Rubin-Delanchy et al., 2017) on the *generalized random dot product graph* (gRDPG) extends the essential construction of the random dot product graph to encompass non-positive-semidefinite stochastic block models and mixed membership block models, and the gRDPG possesses an elegant uniqueness property that permits the representation of mixed membership as a convex combination of extreme points in a simplex. So, while we impose some restrictions on positive-definiteness here, these are primarily for ease of exposition. Our methodology is general enough to encompass networks that exhibit both homophily and heterophily (see, for example, Hoff, 2007). Moreoever, when considering normalized latent positions for random dot product graphs, our requirement of positive definiteness is, in effect, a degree-corrected "affinity." Also, as we show in Section 6.3, our techniques have proven empirically appropriate in the analysis of certain directed graphs, wherein we consider right- and left-singular vectors in a singular value decomposition of the adjacency matrix.

2.3 Embeddings

Since we rely on spectral decompositions, we begin with describing the notations for the spectral decomposition of the rank d positive semidefinite matrix $\mathbf{P} = \mathbf{X}\mathbf{X}^\top$.

Definition 16 (Spectral Decomp. of P) *Since $\mathbf{P}$ is symmetric and positive semidefinite, let $\mathbf{P} = \mathbf{U_P}\mathbf{S_P}\mathbf{U_P}^\top$ denote its spectral decomposition, with $\mathbf{U_P} \in \mathbb{R}^{n\times d}$ having orthonormal columns and $\mathbf{S_P} \in \mathbb{R}^{d\times d}$ a diagonal matrix with nonincreasing entries*

$$(\mathbf{S_P})_{1,1} \geq (\mathbf{S_P})_{2,2} \geq \cdots \geq (\mathbf{S_P})_{d,d} > 0.$$

As with the spectral decomposition of the matrix $\mathbf{P}$, given an adjancency matrix $\mathbf{A}$, we define its adjacency spectral embedding as follows;

Definition 17 (Adjacency spectral embedding (ASE)) *Given a positive integer $d \geq 1$, the* adjacency spectral embedding *(ASE) of $\mathbf{A}$ into $\mathbb{R}^d$ is given by $\hat{\mathbf{X}} = \mathbf{U_A}\mathbf{S_A}^{1/2}$ where*

$$|\mathbf{A}| = [\mathbf{U_A}|\mathbf{U_A^\perp}][\mathbf{S_A} \bigoplus \mathbf{S_A^\perp}][\mathbf{U_A}|\mathbf{U_A^\perp}]^\top$$

is the spectral decomposition of $|\mathbf{A}| = (\mathbf{A^T A})^{1/2}$ and $\mathbf{S_A}$ is the diagonal matrix of the d largest eigenvalues of $|\mathbf{A}|$ and $\mathbf{U_A}$ is the $n \times d$ matrix whose columns are the corresponding eigenvectors.

The intuition behind the notion of adjacency spectral embedding is as follows. Our goal is to estimate the latent position matrix $\mathbf{X}$. Now, if the matrix $\mathbf{P}$ were actually observable, then the spectral embedding of $\mathbf{P}$, given by $\mathbf{U_P}\mathbf{S_P}^{1/2}$, is simply some orthogonal transformation of $\mathbf{X}$. Of course, $\mathbf{P}$ is typically not observable; instead we observe $\mathbf{A}$, a noisy version of $\mathbf{P}$. The ASE will be a good estimate of $\mathbf{X}$ provided that the noise does not greatly impact the embedding—that is, if $\mathbf{A}$ and $\mathbf{P}$ are suitably close. As we will see shortly, one can show that $\|\mathbf{A} - \mathbf{X}\mathbf{X}^\top\| = O(\|\mathbf{X}\|) = o(\|\mathbf{X}\mathbf{X}^\top\|)$ with high probability (Oliveira, 2009; Lu and Peng, 2013; Tropp, 2015; Lei and Rinaldo, 2015). That is to say, $\mathbf{A}$ can be viewed as a comparatively small perturbation of $\mathbf{X}\mathbf{X}^\top$. Weyl's inequality or the Kato-Temple inequality (Cape et al., 2017a; Kato, 1950) guarantee that the eigenvalues of $\mathbf{A}$ are "close" to the eigenvalues of $\mathbf{X}\mathbf{X}^\top$. In addition, by the Davis-Kahan theorem (Davis and Kahan, 1970), the subspace spanned by the top d eigenvectors of $\mathbf{X}\mathbf{X}^\top$ is well-approximated by the subspace spanned by the top d eigenvectors of $\mathbf{A}$.

Having defined the adjacency spectral embedding, we next define the analogous Laplacian spectral embedding, which uses the spectral decomposition of the normalized Laplacian matrix.

Definition 18 (Laplacian Spectral Embedding (LSE)) *Let $\mathcal{L}(\mathbf{A}) = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ denote the normalized Laplacian of $\mathbf{A}$ where $\mathbf{D}$ is the diagonal matrix whose diagonal entries $\mathbf{D}_{ii} = \sum_{j\neq i} \mathbf{A}_{ij}$. Given a positive integer $d \geq 1$, the* Laplacian spectral embedding *(LSE) of $\mathbf{A}$ into $\mathbb{R}^d$ is given by $\check{\mathbf{X}} = \mathbf{U}_{\mathcal{L}(\mathbf{A})}\tilde{\mathbf{S}}_{\mathbf{A}}^{1/2}$ where*

$$|\mathcal{L}(\mathbf{A})| = \left[\mathbf{U}_{\mathcal{L}(\mathbf{A})}|\mathbf{U}^\perp_{\mathcal{L}(\mathbf{A})}\right]\left[\mathbf{S}_{\mathcal{L}(\mathbf{A})} \bigoplus \mathbf{S}^\perp_{\mathcal{L}(\mathbf{A})}\right]\left[\mathbf{U}_{\mathcal{L}(\mathbf{A})}|\mathbf{U}^\perp_{\mathcal{L}(\mathbf{A})}\right]^\top$$

is the spectral decomposition of $|\mathcal{L}(\mathbf{A})| = (\mathcal{L}(\mathbf{A})^\top\mathcal{L}(\mathbf{A}))^{1/2}$ and $\mathbf{S}_{\mathcal{L}(\mathbf{A})}$ is the diagonal matrix containg the d largest eigenvalues of $|\mathcal{L}(\mathbf{A})|$ on the diagonal and $\mathbf{U}_{\mathcal{L}(\mathbf{A})}$ is the $n \times d$ matrix whose columns are the corresponding eigenvectors.

Finally, there are a variety other matrices for which spectral decompositions may be applied to yield an embedding of the graph (Le et al., 2017). These are often dubbed as "regularized" embeddings, and many such embeddings are constructed with the aim of improving robustness of spectral decompositions to graph sparsity. While we do not analyze these embeddings directly, a number of our approaches can be adapted to these other embeddings.

3. Core proof techniques: probabilistic and linear algebraic bounds

In this section, we give an overview of the core background results used in our proofs. While we defer to the appendix the proofs of our primary results on consistency and normality for spectral graph estimates, we provide key background here to allow the reader a consolidated, accessible guide to the essential linear algebraic foundations of our approach. The key tools to several of our results on consistency and normality of the adjacency spectral embedding depend on a triumvirate of matrix concentration inequalities, the Davis-Kahan Theorem, and detailed bounds via the power method.

3.1 Concentration inequalities

Concentration inequalities for real- and matrix-valued data are a critical component to our proofs of consistency for spectral estimates. We make use of classical inequalities, such as Hoeffding's inequality, for real-valued random variables, and we also exploit more recent work on the concentration of sums of random matrices and matrix martingales around their expectation. For a careful study of several important matrix concentration inequalities, see Tropp (2015).

We begin by recalling Hoeffding's inequality, which bounds the deviations between a sample mean of independent random variables and the expected value of that sample mean.

Theorem 19 *Let X_i, $1 \leq i \leq n$, be independent, bounded random variables defined on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Suppose a_i, b_i are real numbers such that $a_i \leq X_i \leq b_i$. Let $\bar{X}$ be their sample mean:*

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

Then

$$\Pr\left(\bar{X} - \mathbb{E}(\bar{X}) \geq t\right) \leq \exp\left(\frac{[-2n^2t^2]}{\sum_{i=1}^{n}(b_i - a_i)^2}\right) \tag{5}$$

and

$$\Pr\left(|\bar{X} - \mathbb{E}(\bar{X})| \geq t\right) \leq 2\exp\left(\frac{[-2n^2t^2]}{\sum_{i=1}^{n}(b_i - a_i)^2}\right) \tag{6}$$

For an undirected, hollow RDPG with probability matrix $\mathbf{P}$, $\mathbb{E}(\mathbf{A}_{ij}) = \mathbf{P}_{ij}$ for all $i \neq j$. As such, one can regard $\mathbf{A}$ as a "noisy" version of $\mathbf{P}$. It is tempting to believe that $\mathbf{A}$ and $\mathbf{P}$ are close in terms of the Frobenius norm, but this is sadly not true; indeed, it is easy to see that

$$\|\mathbf{A} - \mathbf{P}\|_F^2 = \Theta(\|\mathbf{P}\|_F^2)$$

To overcome this using only Hoeffding's inequality, we can instead consider the difference $(\mathbf{A}^2 - \mathbf{P}^2)_{ij}$, which is a sum of independent random variables. Hence, Hoeffding's inequality implies that

$$|(\mathbf{A}^2 - \mathbf{P}^2)_{ij}|^2 = o(|\mathbf{P}^2{}_{ij}|^2)$$

Since the eigenvectors of $\mathbf{A}$ and $\mathbf{A}^2$ coincide, this is itself sufficient to show concentration of the adjacency spectral embedding (Sussman et al., 2012; Rohe et al., 2011). However,

somewhat stronger and more elegant results can be shown by considering the spectral norm instead. In particular, a nontrivial body of recent work on matrix concentration implies that, under certain assumptions on the sparsity of $\mathbf{P}$, the spectral norm of $\mathbf{A} - \mathbf{P}$ can be well-controlled. We focus on the following important results of Oliveira (2009) and Tropp (2015) and further improvements of Lu and Peng (2013) and Lei and Rinaldo (2015), all of which establish that the $\mathbf{A}$ and $\mathbf{P}$ are close in spectral norm.

Theorem 20 (Spectral norm control of $\mathbf{A} - \mathbf{P}$) *Suppose Let $\mathbf{A}$ be the adjacency matrix of an independent-edge random graph on $[n]$ with matrix of edge probabilities $\mathbf{P}$. For any constant c, there exists another constant C, independent of n and $\mathbf{P}$, such that if $\delta(\mathbf{P}) > C \ln n$, then for any $n^{-c} < \eta < 1/2$,*

$$\Pr\left(\|\mathbf{A} - \mathbf{P}\| \leq 4\sqrt{\delta(P)\ln(n/\eta)}\right) \geq 1 - \eta. \tag{7}$$

In Lu and Peng (2013), the authors give an improvement under slightly stronger density assumptions[1]:

Theorem 21 (Refined spectral norm control of $\mathbf{A} - \mathbf{P}$) *With notation as above, suppose there exist positive constants such that for n sufficiently large, $\delta(\mathbf{P}) > (\log n)^{4+a}$. Then for any $c > 0$ there exists a constant C depending on c such that*

$$\mathbb{P}\left(\|\mathbf{A} - \mathbf{P}\| \leq 2\sqrt{\delta(\mathbf{P})} + C\delta^{1/4}(\mathbf{P})\ln n\right) \geq 1 - n^{-c}. \tag{8}$$

3.2 Matrix perturbations and spectral decompositions

The above results formalize our intuition that $\mathbf{A}$ provides a "reasonable" estimate for $\mathbf{P}$. Moreover, in the RDPG case, where $\mathbf{P}$ is of low rank and is necessarily positive semidefinite, these results have implications about the nonnegativity of the eigenvalues of $\mathbf{A}$. Specifically, we use Weyl's Theorem to infer bounds on the differences between the eigenvalues of $\mathbf{A}$ and $\mathbf{P}$ from the spectral norm of their difference, and the Gerschgorin Disks Theorem to infer lower bounds on the maximum row sums of $\mathbf{P}$ from assumptions on the eigengap of $\mathbf{P}$ (since both $\mathbf{P}$ and $\mathbf{A}$ are nonnegative matrices, one could also obtain the same lower bounds by invoking the Perron-Frobenius Theorem). For completeness, we recall the Gerschgorin Disks Theorem and Weyl's Theorem (these can be found, for example, in Horn and Johnson, 1985). The former relates the eigenvalues of a matrix to the sums of the absolute values of the entries in each row, and the latter establishes bounds on the differences in eigenvalues between a matrix and a perturbation.

Theorem 22 (Gerschgorin Disks) *Let $\mathbf{H}$ be a complex $n \times n$ matrix, with entries $\mathbf{H}_{ij}$. For $i \in \{1, \cdots, n\}$ let $R_i = \sum_{j \neq i} |\mathbf{H}_{ij}|$. Let the ith Gerschgorin disk $D(\mathbf{H}_{ii}, R_i)$ be the closed disk centered at $\mathbf{H}_{ii}$ with radius R_i. Then every eigenvalue of $\mathbf{H}$ lies within at least one of the Gershgorin discs $D(\mathbf{H}_{ii}, R_i)$.*

1. A similar bound is provided in Lei and Rinaldo (2015), but with $\delta(\mathbf{P})$ defined as $\delta(\mathbf{P}) = n\max_{ij}\mathbf{P}_{ij}$ and a density assumption of the form $(n\max_{ij}\mathbf{P}_{ij}) > (\log n)^{1+a}$.

Theorem 23 (Weyl) *Let $\mathbf{M}, \mathbf{H}$, and $\mathbf{R}$ be $n \times n$ Hermitian matrices, and suppose $\mathbf{M} = \mathbf{H} + \mathbf{R}$. Suppose $\mathbf{H}$ and $\mathbf{R}$ have eigenvalues $\nu_1 \geq \cdots \geq \nu_n$ and $r_1 \geq \cdots \geq r_n$, respectively. Suppose the eigenvalues of $\mathbf{M}$ are given by $\mu_1 \geq \cdots \geq \mu_n$. Then*

$$\nu_i + r_n \leq \mu_i \leq \nu_i + r_1$$

From our random graph model assumptions and our graph density assumptions, we can conclude that with for sufficiently large n, the top d eigenvalues of $\mathbf{A}$ will be nonnegative:

Remark 24 (Nonnegativity of the top d eigenvalues of A) *Suppose $\mathbf{A} \sim \mathrm{RDPG}(\mathbf{X})$. Since $\mathbf{P} = \mathbf{X}\mathbf{X}^\top$, it is necessarily positive semidefinite, and thus has nonnegative eigenvalues. If we now assume that $\gamma(\mathbf{P}) > c_0$ for some constant c_0, then along with the Gershgorin Disks Theorem, guarantee that the top d eigenvalues of $\mathbf{P}$ are all of order $\delta(\mathbf{P})$, and our rank assumption on $\mathbf{P}$ mandates that the remaining eigenvalues be zero. If $\delta(\mathbf{P}) > \log^{4+a'} n$, the spectral norm bound in (8) applies, ensuring that for n sufficiently large, $\|A - P\|_2 \sim O(\sqrt{\delta(\mathbf{P})})$ with high probability. Thus, by Weyl's inequality, we see that the top d eigenvalues of $\mathbf{A}$ are, with high probability, of order δ, and the remaining are, with high probability, within $\sqrt{\delta}$ of zero.*

Since $\mathbf{P} = \mathbf{X}\mathbf{X}^\top = \mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{1/2}(\mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{1/2})^\top$ and $\mathbf{A}$ is close to $\mathbf{P}$, it is intuitively appealing to conjecture that, in fact, $\hat{\mathbf{X}} = \mathbf{U}_\mathbf{A}\mathbf{S}_\mathbf{A}^{1/2}$ should be close to some rotation of $\mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{1/2}$. That is, if $\mathbf{X}$ is the matrix of true latent positions—-so $\mathbf{X}\mathbf{W} = \mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{1/2}$ for some orthogonal matrix $\mathbf{W}$—then it is plausible that $\|\hat{\mathbf{X}} - \mathbf{X}\mathbf{W}\|_F$ ought to be comparatively small. To make this precise, however, we need to understand how both eigenvalues and eigenvectors of a matrix behave when the matrix is perturbed. Weyl's inequality addresses the former. The impact of matrix perturbations on associated eigenspaces is significantly more complicated, and the Davis-Kahan Theorem (Davis and Kahan, 1970; Bhatia, 1997) provides one approach to the latter. The Davis-Kahan has a significant role in several approaches to spectral estimation for graphs: for example, Rohe, Chatterjee, and Yu leverage it in Rohe et al. (2011) to prove the accuracy of spectral estimates (specifically, the graph Laplacian) in high-dimensional stochastic blockmodels. The version we give below is from Yu et al. (2015), which is a user-friendly guide to the the Davis-Kahan Theorem and its statistical implications.

The Davis-Kahan Theorem is often stated as a result on canonical angles between subspaces. To that end, we recall that if $\mathbf{U}$ and $\mathbf{V}$ are two $n \times d$ matrices with orthonormal columns, then we define the vector of d *canonical* or *principal* angles between their column spaces to be the vector Θ such that

$$\Theta = (\theta_1 = \cos^{-1}\sigma_1, \cdots, \theta_d = \cos^{-1}\sigma_d)^\top$$

where $\sigma_1, \cdots, \sigma_d$ are the singular values of $\mathbf{U}^\top\mathbf{V}$. We define the matrix $\sin(\Theta)$ to be the $d \times d$ diagonal matrix for which $\sin(\theta)_{ii} = \sin\theta_i$.

Theorem 25 (A variant of Davis-Kahan) *Suppose $\mathbf{H}$ and $\mathbf{H}'$ are two symmetric $n \times n$ matrices with real entries with spectrum given by $\lambda_1 \geq \lambda_2 \cdots \geq \lambda_n$ and $\lambda_1' \geq \lambda_2' \cdots \geq \lambda_n'$, respectively; and let $\mathbf{v}_1, \cdots, \mathbf{v}_n$ and $\mathbf{v}_1', \cdots, \mathbf{v}_n'$ denote their corresponding orthonormal eigenvectors.*

Let $0 < d \leq n$ be fixed, and let $\mathbf{V}$ be the matrix of whose columns are the eigenvectors $v_1, \cdots, v_d$, and similarly $\mathbf{V}'$ the matrix whose columns are the eigenvectors $\mathbf{v}_1', \cdots \mathbf{v}_n'$. Then

$$\|\sin(\Theta)\|_F \leq \frac{2\sqrt{d}\|\mathbf{V} - \mathbf{V}'\|}{\lambda_d(\mathbf{H}) - \lambda_{d+1}(\mathbf{H})}.$$

Observe that if we assume that $\mathbf{P}$ is of rank d and has a sufficient eigengap, the Davis-Kahan Theorem gives us an immediate bound on the spectral norm of the difference between $\mathbf{U}_\mathbf{A}\mathbf{U}_\mathbf{A}^\top$ and $\mathbf{U}_\mathbf{P}\mathbf{U}_\mathbf{P}^\top$ in terms of this eigengap and the spectral norm difference of $\mathbf{A} - \mathbf{P}$, namely:

$$\|\mathbf{U}_\mathbf{A}\mathbf{U}_\mathbf{A}^\top - \mathbf{U}_\mathbf{P}\mathbf{U}_\mathbf{P}^\top\| = \max_i \|\sin(\theta_i)\| \leq \frac{C\|\mathbf{A} - \mathbf{P}\|}{\lambda_d(\mathbf{P})}.$$

Recall that the Frobenius norm of a matrix $\mathbf{H}$ satisfies

$$(\|\mathbf{H}\|_F)^2 = \sum_{i,j} \mathbf{H}_{ij}^2 = \mathrm{tr}(\mathbf{H}^\top\mathbf{H}) \geq \|\mathbf{H}\|^2$$

and further that if $\mathbf{H}$ is of rank d, then

$$(\|\mathbf{H}\|_F)^2 \leq d\|\mathbf{H}\|^2$$

and hence for rank d matrices, spectral norm bounds are easily translated into bounds on the Frobenius norm. It is worth noting that Rohe et al. (2011) guarantees that a difference in projections can be transformed into a difference between eigenvectors themselves: namely, given the above bound for $\|\mathbf{U}_\mathbf{A}\mathbf{U}_\mathbf{A}^\top - \mathbf{U}_\mathbf{P}\mathbf{U}_\mathbf{P}^\top\|_F$, there is a constant C and an orthonormal matrix $W \in \mathbb{R}^{d \times d}$ such that

$$\|\mathbf{U}_\mathbf{P}\mathbf{W} - \mathbf{U}_\mathbf{A}\|_F \leq C\sqrt{d}\frac{\|\mathbf{A} - \mathbf{P}\|}{\lambda_d(\mathbf{P})}. \tag{9}$$

While these important results provide the backbone for much of our theory, the detailed bounds and distributional results described in the next section rely on a decomposition of $\hat{\mathbf{X}}$ in terms of $(\mathbf{A} - \mathbf{P})\mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{-1/2}$ and a remainder. This first term can be viewed as an application of the power method for finding eigenvectors. Additionally, standard univariate and multivariate concentration inequalities and distributional results can be readily applied to this term. On the other hand, the remainder term can be shown to be of smaller order than the first, and much of the technical challenges of this work rely on carefully bounding the remainder term.

4. Spectral embeddings and estimation for RDPGs

There is a wealth of literature on spectral methods for estimating model parameters in random graphs, dating back more than half a century to estimation in simple Erdős-Rényi models. More specifically, for Erdős-Rényi graphs, we would be remiss not to point to Furedi and Komlos's classic work (Füredi and Komlós, 1981) on the eigenvalues and eigenvectors

of the adjacency matrix of an ER graph. Again, despite their model simplicity, Erdős-Rényi graphs veritably teem with open questions; to cite but one example, in a very recent manuscript, Arias-Castro and Verzhelen (Arias-Castro and Verzelen, 2014) address, in the framework of hypothesis testing, the question of subgraph detection within an ER graph.

Moving up to the slightly more heterogeneous stochastic block model, we again find a rich literature on consistent block estimation in stochastic block models. Fortunato (Fortunato, 2010) provides an overview of partitioning techniques for graphs in general, and consistent partitioning of stochastic block models for two blocks was accomplished by Snijders and Nowicki (Snijders and K.Nowicki, 1997) and for equal-sized blocks by Condon and Karp in 2001. For the more general case, Bickel and Chen (Bickel and Chen, 2009) demonstrate a stronger version of consistency via maximizing Newman-Girvan modularity (Newman, 2006) and other modularities. For a growing number of blocks, Choi et al. (Bickel et al., 2013) prove consistency of likelihood based methods, and Bickel et al. (Bickel et al., 2011) provide a method to consistently estimate the stochastic block model parameters using subgraph counts and degree distributions. This work and the work of Bickel and Chen (Bickel and Chen, 2009) both consider the case of very sparse graphs. In Airoldi et al. (2008), Airoldi et al. define the important generalization of a *mixed-membership* stochastic blockmodel, in which block membership may change depending on vertex-to-vertex interactions, and the authors demonstrate methods of inference for the mixed membership and block probabilities.

Rohe, Chatterjee and Yu show in Rohe et al. (2011) that spectral embeddings of the Laplacian give consistent estimates of block memberships in a stochastic block model, and they also demonstrate how to deploy spectral approaches to estimate block memberships even in strongly heterophilic block models. One of the earliest corresponding results on the consistency of the adjacency spectral embedding the estimation of block memberships is given by Sussman and coauthors in Sussman et al. (2012), where it is proved that for a stochastic block model with K blocks and a rank d block probability matrix B, clustering the rows of the adjacency spectral embedding via k-means clustering (see Pollard, 1981) results in at most $\log n$ vertices being misclassified. An improvement to this can be found in Fishkind et al. (2013), where consistency recovery is possible even if the rank of the embedding dimension is unknown.

In Lyzinski et al. (2014), under the assumption of distinct eigenvalues for the second moment matrix Δ (recall we defined this in 5) of a random dot product graph, it is shown that clustering the rows of the adjacency spectral embedding results in asymptotically almost surely perfect recovery of the block memberships in a stochastic blockmodel—that is, for sufficiently large n, the probability of all vertices being correctly assigned is close to 1. An especially strong recovery is exhibited here: it is shown that in the $2 \to \infty$ norm, $\hat{\mathbf{X}}$ is sufficiently close to a rotation of the the true latent positions. In fact, each *row* in $\hat{\mathbf{X}}$ is within $C \log n/\sqrt{n}$ of the corresponding row in $\mathbf{X}$. Unlike a Frobenius norm bound, in which it is possible that some rows of $\hat{\mathbf{X}}$ may be close to the true positions but others may be significantly farther away, this $2 \to \infty$ bound implies that the adjacency spectral embedding has a *uniform* consistency.

Furthermore, in Tang et al. (2017a), one finds a nontrivial tightening of the Frobenius norm bound for the difference between the (rotated) true and estimated latent positions: in fact the Frobenius norm is not merely bounded from above by a term of order $\log n$, but rather concentrates around a *constant*. This constant-order Frobenius bound forms the basis of a principled two-sample hypothesis test for determining whether two random dot product graphs have the same generating latent positions (see Section 5.2).

In Lyzinski et al. (2017), the $2 \to \infty$-norm bound is extended even to the case when the second moment matrix Δ does not have distinct eigenvalues. This turns out to be critical in guaranteeing that the adjacency spectral embedding can be effectively deployed for community detection in hierarchical block models. We present this bound for the $2 \to \infty$ norm in some detail here; it illustrates the confluence of our key techniques and provides an effective roadmap for several subsequent results on asymptotic normality and two-sample testing.

4.1 Consistency of latent position estimates

We state here one of our lynchpin results on consistency, in the $2 \to \infty$ norm, of the adjacency spectral embedding for latent position estimation of a random dot product graph. We given an outline of the proof here, and refer the reader to the Appendix A for the details, which essentially follow the proof given in Lyzinski et al. (2017). We emphasize our setting is a sequence of random dot product graphs $\mathbf{A}_n \sim \text{RDPG}(\mathbf{X}_n)$ for increasing n and thus we make the following density assumption on $\mathbf{P}_n$ as n increases:

Assumption 1 *Let $\mathbf{A}_n \sim \text{RDPG}(\mathbf{X}_n)$ for $n \geq 1$ be a sequence of random dot product graphs with $\mathbf{A}_n$ being a $n \times n$ adjacency matrix. Suppose that $\mathbf{X}_n$ is of rank d for all n sufficiently large. Suppose also that there exists constants $a > 0$ and $c_0 > 0$ such that for all n sufficiently large,*

$$\delta(\mathbf{P}_n) = \max_i \sum_{j=1}^{n} (\mathbf{P}_n)_{ij} \geq \log^{4+a}(n); \quad \gamma(\mathbf{P}_n) = \frac{\lambda_d(\mathbf{P}_n)}{\delta(\mathbf{P}_n)} \geq c_0.$$

We remark that this assumption is, in essence, a restriction on graph sparsity, and not a particular onerous one; d-dimensional random dot product graphs with i.i.d latent positions that are not "too" degenerate will meet these requirements, and bounds on δ of the polylogarithmic order given above are quite common (see Oliveira, 2009; Lu and Peng, 2013).

Our consistency result for the $2 \to \infty$ norm is Theorem 26 below. In the proof of this particular result, we consider a particular random dot product graph with non-random (that is, fixed) latent positions, but our results apply also to the case of random latent positions. In Section 4.2, where we provide a central limit theorem, we focus on the case in which the latent positions are themselves random. Similarly, in Section 5.2, in our analysis of the semiparametric two-sample hypothesis test for the equality of latent positions in a pair of random dot product graphs, we return to the setting in which the latent positions are fixed, and in the nonparametric hypothesis test of equality of distributions, we analyze again the case when the latent positions are random. It is convenient to be able to move

fluidly between the two versions of a random dot product graph, adapting our results as appropriate in each case.

In the Appendix, we give a detailed proof of Theorem 26 and we point out the argument used therein also sets the stage for the central limit theorem for the rows of the adjacency spectral embedding given in Subsection 4.2.

Theorem 26 *Let $\mathbf{A}_n \sim \text{RDPG}(\mathbf{X}_n)$ for $n \geq 1$ be a sequence of random dot product graphs where the $\mathbf{X}_n$ is assumed to be of rank d for all n sufficiently large. Denote by $\hat{\mathbf{X}}_n$ the adjacency spectral embedding of $\mathbf{A}_n$ and let $(\hat{\mathbf{X}}_n)_i$ and $(\mathbf{X}_n)_i$ be the i-th row of $\hat{\mathbf{X}}_n$ and $\mathbf{X}_n$, respectively. Let E_n be the event that there exists an orthogonal transformation $\mathbf{W}_n \in \mathbb{R}^{d\times d}$ such that*

$$\max_i \|(\hat{\mathbf{X}}_n)_i - \mathbf{W}_n(\mathbf{X}_n)_i\| \leq \frac{Cd^{1/2}\log^2 n}{\delta^{1/2}(\mathbf{P}_n)}$$

where $C > 0$ is some fixed constant and $\mathbf{P}_n = \mathbf{X}_n\mathbf{X}_n^\top$. Then E_n occurs asymptotically almost surely; that is, $\Pr(E_n) \to 1$ as $n \to \infty$.

Under the stochastic blockmodel, previous bounds on $\|\mathbf{X} - \hat{\mathbf{X}}\|_F$ implied that k-means applied to the rows of $\hat{\mathbf{X}}$ would approximately correctly partition the vertices into their the true blocks with up to $O(\log n)$ errors. However, this Frobenius norm bound does not preclude the possibility of large outliers in the individual rows of $\mathbf{X} - \hat{\mathbf{X}}$. The improvements provided by Theorem 26 overcome this hurdle and, as shown in Lyzinski et al. (2014), under suitable sparsity and eigengap assumptions, k-means applied to $\hat{\mathbf{X}}$ will *exactly correctly* partition the vertices into their true blocks. This implication demonstrates the importance of improving the overall bounds and in focusing on the correct metrics for a given task—in this case, for instance, block identification.

While we defer the proof of this result to the Appendix, in A.1, we provide a brief outline and note several key ingredients here. First is a lemma guaranteeing the existence of an orthogonal matrix $\mathbf{W}^*$ such that

$$\|\mathbf{W}^*\mathbf{S}_\mathbf{A} - \mathbf{S}_\mathbf{P}\mathbf{W}^*\|_F = O((\log n)\delta^{1/2}(P)$$

That is, there is an approximate commutativity between right and left multiplication of the corresponding matrices of eigenvalues by this orthogonal transformation. The second essential component is, at heart, a bound inspired by the power method. Specifically, we show that there exists an orthogonal matrix

$$\|\hat{\mathbf{X}} - \mathbf{X}\mathbf{W}\| = \|(\mathbf{A} - \mathbf{P})\mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{-1/2}\|_F + O((\log n)\delta^{-1/2})$$

Finally, from this point, establishing the bound on the $2 \to \infty$ norm is a consequence of Hoeffding's inequality applied to sums of the form

$$\sum_k (\mathbf{A}_{ik} - \mathbf{P}_{ik}\mathbf{U}_{kj})$$

.

The $2 \to \infty$ bound in Theorem 26 has several important implications. As we mentioned, the results in Lyzinski et al. (2014) establish an earlier form of this bound, with more

restrictive assumptions on the the second moment matrix, and illustrate how this can be used to cluster vertices in an SBM perfectly; that is, with no vertices misclassified. In addition, in Lyzinski et al. (2014), we show that clustering the rows of the ASE can be useful for inference in a degree-corrected stochastic block model as well. In Section 6, we see that because of Theorem 26, the adjacency spectral embedding and a novel angle-based clustering procedure can be used for accurately identifying subcommunities in an affinity-structured, hierarchical stochastic blockmodel (Lyzinski et al., 2017). In the next section, we see how our proof technique here can be used to obtain distributional results for the rows of the adjacency spectral embedding.

4.2 Distributional results for the ASE

In the classical statistical task of parametric estimation, one observes a collection of i.i.d observations $X_1, \cdots, X_n$ from some family of distributions $F_\theta : \theta \in \Theta$, where Θ is some subset of Euclidean space, and one seeks to find a consistent estimator $T(X_1, \cdots, X_n)$ for θ. As we mentioned in Section 1, often a next task is to determine the asymptotic distribution, as $n \to \infty$, of a suitable normalization of this estimator T. Such distributional results, in turn, can be useful for generating confidence intervals and testing hypotheses.

We adopt a similar framework for random graph inference. In the previous section, we demonstrate the consistency of the adjacency spectral embedding for the true latent position of a random dot product graph. In this section, we establish the asymptotic normality of the rows of this embedding and the Laplacian spectral embedding. In the subsequent section, we examine how our methodology can be deployed for multisample graph hypothesis testing.

We emphasize that distributional results for spectral decompositions of random graphs are comparatively few. The classic results of Füredi and Komlós (Füredi and Komlós, 1981) describe the eigenvalues of the Erdős-Rényi random graph and the work of Tao and Vu (Tao and Vu, 2012) is focused on distributions of eigenvectors of more general random matrices under moment restrictions, but Athreya et al. (2016) and Tang and Priebe (2018) are among the only references for central limit theorems for spectral decompositions of adjacency and Laplacian matrices for a wider class of independent-edge random graphs than merely the Erdős-Rényi model. Apart from their inherent interest, these limit theorems point us to current open questions on efficient estimation and the relative merits of different estimators and embeddings, in part by rendering possible a comparison of asymptotic variances and allowing us to quantify relative efficiency (see Tang et al. (2016) and to precisely conjecture a decomposition of the sources of variance in different spectral embeddings for multiple graphs (see Levin et al., 2017).

Specifically, we show that for a d-dimensional random dot product graph with i.i.d latent positions, there exists a sequence of $d \times d$ orthogonal matrices $\mathbf{W}_n$ such that for any row index i, $\sqrt{n}(\mathbf{W}_n(\hat{\mathbf{X}}_n)_i - (\mathbf{X}_n)_i)$ converges as $n \to \infty$ to a mixture of multivariate normals.

Theorem 27 (Central Limit Theorem for rows of ASE) *Let $(\mathbf{A}_n, \mathbf{X}_n) \sim \text{RDPG}(F)$ be a sequence of adjacency matrices and associated latent positions of a d-dimensional random dot product graph according to an inner product distribution F. Let $\Phi(\mathbf{x}, \mathbf{\Sigma})$ denote*

the cdf of a (multivariate) Gaussian with mean zero and covariance matrix $\mathbf{\Sigma}$*, evaluated at* $\mathbf{x} \in \mathbb{R}^d$*. Then there exists a sequence of orthogonal* d*-by-*d *matrices* $(\mathbf{W}_n)_{n=1}^{\infty}$ *such that for all* $\boldsymbol{z} \in \mathbb{R}^d$ *and for any fixed index* i*,*

$$\lim_{n\to\infty} \Pr\left[n^{1/2}\left(\hat{\mathbf{X}}_n\mathbf{W}_n - \mathbf{X}_n\right)_i \le \boldsymbol{z}\right] = \int_{\operatorname{supp} F} \Phi\left(\boldsymbol{z}, \mathbf{\Sigma}(\mathbf{x})\right) dF(\mathbf{x}),$$

where

$$\mathbf{\Sigma}(\mathbf{x}) = \Delta^{-1}\mathbb{E}\left[(\mathbf{x}^\top\mathbf{X}_1 - (\mathbf{x}^\top\mathbf{X}_1)^2)\mathbf{X}_1\mathbf{X}_1^\top\right]\Delta^{-1}; \quad \textit{and } \Delta = \mathbb{E}[\mathbf{X}_1\mathbf{X}_1^\top]. \tag{10}$$

We also note the following important corollary of Theorem 27 for when F is a mixture of K point masses; that is, $(\mathbf{X}, \mathbf{A}) \sim \text{RDPG}(F)$ is a K-block stochastic blockmodel graph. Then for any fixed index i, the event that $\mathbf{X}_i$ is assigned to block $k \in \{1, 2, \ldots, K\}$ has non-zero probability and hence one can condition on the block assignment of $\mathbf{X}_i$ to show that the conditional distribution of $\sqrt{n}(\mathbf{W}_n(\hat{\mathbf{X}}_n)_i - (\mathbf{X}_n)_i)$ converges to a multivariate normal. This is in contrast to the unconditional distribution being a mixture of multivariate normals as given in Theorem 27.

Corollary 28 (SBM) *Assume the setting and notations of Theorem 27 and let*

$$F = \sum_{k=1}^{K} \pi_k \delta_{\nu_k}, \quad \pi_1, \cdots, \pi_K > 0, \sum_k \pi_k = 1$$

be a mixture of K *point masses in* $\mathbb{R}^d$ *where* δ_{ν_k} *is the Dirac delta measure at* ν_k*. Then there exists a sequence of orthogonal matrices* $\mathbf{W}_n$ *such that for all* $\boldsymbol{z} \in \mathbb{R}^d$ *and for any fixed index* i*,*

$$\mathbb{P}\left\{\sqrt{n}(\mathbf{W}_n\hat{\mathbf{X}}_n - \mathbf{X}_n)_i \le \boldsymbol{z} \mid \mathbf{X}_i = \nu_k\right\} \longrightarrow \Phi(\boldsymbol{z}, \Sigma_k) \tag{11}$$

where $\Sigma_k = \Sigma(\nu_k)$ *is as defined in Eq.* (52)*.*

We relegate the full details of the proof of this central limit theorem to the Appendix, in Section A.2, but a few points bear noting here. First, both Theorem 27 and Corollary 28 are very similar to results proved in Athreya et al. (2016), but with the crucial difference being that we no longer require that the second moment matrix $\Delta = \mathbb{E}[\mathbf{X}_1\mathbf{X}_1^\top]$ of $\boldsymbol{X}_1 \sim F$ have distinct eigenvalues (for more details, see Tang and Priebe (2018)). As in Athreya et al. (2016), our proof here depends on writing the difference between a row of the adjacency spectral embedding and its corresponding latent position as a pair of summands: the first, to which a classical Central Limit Theorem can be applied, and the second, essentially a combination of residual terms, which we show, using techniques similar to those in the proof of Theorem 26, converges to zero. The weakening of the assumption of distinct eigenvalues necessitates significant changes from Athreya et al. (2016) in how to bound the residual terms, because the arguments in Athreya et al. (2016) require an adaptation of a result of Bickel and Sarkar (2015)—the latter of which depends on the assumption of distinct eigenvalues—to control these terms. Here, we resort to somewhat different methodology: we prove instead that analogous bounds to those in Lyzinski et al. (2017) and in Tang and Priebe (2018) hold for the estimated latent positions. This enables us to establish that here, too, the rows of the adjacency spectral embedding are also approximately normally distributed.

We stress that this central limit theorem depends on a delicate bounding of a sequence of so-called residual terms, but its essence is straightforward. Specifically, there exists an orthogonal transformation $\mathbf{W}^*$ such that we can write the ith row of the matrix

$$\sqrt{n}(\mathbf{U}_\mathbf{A}\mathbf{S}_\mathbf{A}^{1/2} - \mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{1/2}\mathbf{W}^*)$$

as

$$\sqrt{n}(\mathbf{U}_\mathbf{A}\mathbf{S}_\mathbf{A}^{1/2} - \mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{1/2}\mathbf{W}^*)_i = \sqrt{n}(\mathbf{A} - \mathbf{P})\mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{-1/2}\mathbf{W}^*)_i + \text{Residual terms} \tag{12}$$

where the residual terms are all of order $O(n^{-1/2}\log n)$ in probability. Now, to handle the first term in Eq.(12), we can condition on a fixed latent position $\mathbf{X}_i = \mathbf{x}_i$, and when this is fixed, the classical Lindeberg-Feller Central Limit Theorem establishes the asymptotic normality of this term. The order of the residual terms then guarantees, by Slutsky's Theorem, the desired asymptotic normality of the gap between estimated and true latent positions, and finally we need only integrate over the possible latent positions to obtain a mixture of normals.

4.3 An example under the stochastic block model

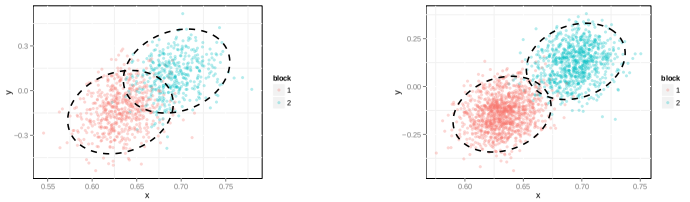
To illustrate Theorem 27, we refer the reader to Athreya et al. (2016), from which we reproduce the following simulation. Consider random graphs generated according to a stochastic block model with parameters

$$B = \begin{bmatrix} 0.42 & 0.42 \\ 0.42 & 0.5 \end{bmatrix} \quad \text{and} \quad \pi = (0.6, 0.4). \tag{13}$$

In this model, each node is either in block 1 (with probability 0.6) or block 2 (with probability 0.4). Adjacency probabilities are determined by the entries in B based on the block memberships of the incident vertices. The above stochastic blockmodel corresponds to a random dot product graph model in $\mathbb{R}^2$ where the distribution F of the latent positions is a mixture of point masses located at $x_1 \approx (0.63, -0.14)$ (with probability 0.6) and $x_2 \approx (0.69, 0.13)$ (with probability 0.4).

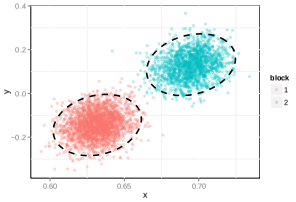
We sample an adjacency matrix $\mathbf{A}$ for graphs on n vertices from the above model for various choices of n. For each graph G, let $\hat{\mathbf{X}} \in \mathbb{R}^{n\times 2}$ denote the embedding of A and let $\hat{\mathbf{X}}_{i\cdot}$ denote the ith row of $\hat{\mathbf{X}}$. In Figure 1, we plot the n rows of $\hat{\mathbf{X}}$ for the various choices of n. The points are denoted by symbols according to the block membership of the corresponding vertex in the stochastic blockmodel. The ellipses show the 95% level curves for the distribution of $\hat{X}_i$ for each block as specified by the limiting distribution.

We estimate the covariance matrices for the residuals. The theoretical covariance matrices are given in the last column of Table 1, where Σ_1 and Σ_2 are the covariance matrices for the residual $\sqrt{n}(\hat{\mathbf{X}}_i - \mathbf{X}_i)$ when X_i is from the first block and second block, respectively. The empirical covariance matrices, denoted $\hat{\Sigma}_1$ and $\hat{\Sigma}_2$, are computed by evaluating the sample covariance of the rows of $\sqrt{n}\hat{\mathbf{X}}_i$ corresponding to vertices in block 1 and 2 respectively. The estimates of the covariance matrices are given in Table 1. We see that as n increases, the sample covariances tend toward the specified limiting covariance matrix given in the last column.

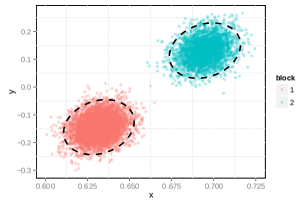


(a) $n = 1000$

(b) $n = 2000$



(c) $n = 4000$



(d) $n = 8000$

Figure 1: Plot of the estimated latent positions in a two-block stochastic blockmodel graph on n vertices. The points are denoted by symbols according to the block membership of the corresponding vertices. Dashed ellipses give the 95% level curves for the distributions as specified in Theorem 27. Figure duplicated from Athreya et al. (2016).

n	2000	4000	8000	16000	∞
$\hat{\Sigma}_1$	$\begin{bmatrix} .58 & .54 \\ .54 & 16.56 \end{bmatrix}$	$\begin{bmatrix} .58 & .63 \\ .63 & 14.87 \end{bmatrix}$	$\begin{bmatrix} .60 & .61 \\ .61 & 14.20 \end{bmatrix}$	$\begin{bmatrix} .59 & .58 \\ .58 & 13.96 \end{bmatrix}$	$\begin{bmatrix} .59 & .55 \\ .55 & 13.07 \end{bmatrix}$
$\hat{\Sigma}_2$	$\begin{bmatrix} .58 & .75 \\ .75 & 16.28 \end{bmatrix}$	$\begin{bmatrix} .59 & .71 \\ .71 & 15.79 \end{bmatrix}$	$\begin{bmatrix} .58 & .54 \\ .54 & 14.23 \end{bmatrix}$	$\begin{bmatrix} .61 & .69 \\ .69 & 13.92 \end{bmatrix}$	$\begin{bmatrix} .60 & .59 \\ .59 & 13.26 \end{bmatrix}$

Table 1: The sample covariance matrices for $\sqrt{n}(\hat{X}_i - X_i)$ for each block in a stochastic blockmodel with two blocks. Here $n \in \{2000, 4000, 8000, 16000\}$. In the last column are the theoretical covariance matrices for the limiting distribution. Table reproduced from Athreya et al. (2016).

We also investigate the effects of the multivariate normal distribution as specified in Theorem 27 on inference procedures. It is shown in Sussman et al. (2012, 2014) that the approach of embedding a graph into some Euclidean space, followed by inference (for example, clustering or classification) in that space can be consistent. However, these consistency results are, in a sense, only first-order results. In particular, they demonstrate only that the error of the inference procedure converges to 0 as the number of vertices in the graph increases. We now illustrate how Theorem 27 may lead to a more refined error analysis.

We construct a sequence of random graphs on n vertices, where n ranges from 1000 through 4000 in increments of 250, following the stochastic blockmodel with parameters as given above in Eq. (13). For each graph G_n on n vertices, we embed G_n and cluster the embedded vertices of G_n via Gaussian mixture model and K-means clustering. Gaussian mixture model-based clustering is done using the MCLUST implementation of (Fraley and Raftery, 1999). We then measure the classification error of the clustering solution. We repeat this procedure 100 times to obtain an estimate of the misclassification rate. The results are plotted in Figure 2. For comparison, we plot the Bayes optimal classification error rate under the assumption that the embedded points do indeed follow a multivariate normal mixture with covariance matrices Σ_1 and Σ_2 as given in the last column of Table 1. We also plot the misclassification rate of $(C \log n)/n$ as given in Sussman et al. (2012) where the constant C was chosen to match the misclassification rate of K-means clustering for $n = 1000$. For the number of vertices considered here, the upper bound for the constant C from Sussman et al. (2012) will give a vacuous upper bound of the order of 10^6 for the misclassification rate in this example. Finally, we recall that the $2 \to \infty$ norm bound of Theorem 26 implies that, for large enough n, even the k-means algorithm will exactly recover the true block memberships with high probability (Lyzinski et al., 2016a).

For yet another application of the central limit theorem, we refer the reader to Suwan et al. (2016), where the authors discuss the assumption of multivariate normality for estimated latent positions and how this can lead to a significantly improved empirical-Bayes framework for the estimation of block memberships in a stochastic blockmodel.

4.4 Distributional results for Laplacian spectral embedding

We now present the analogous central limit theorem results of Section 4.2 for the normalized *Laplacian spectral embedding* (see Definition 18). We first recall the definition of the Laplacian spectral embedding.

Theorem 29 (Central Limit Theorem for rows of LSE) *Let* $(\mathbf{A}_n, \mathbf{X}_n) \sim \text{RDPG}(F)$ *for* $n \geq 1$ *be a sequence of* d*-dimensional random dot product graphs distributed according to some inner product distribution* F*. Let* $\boldsymbol{\mu}$ *and* $\tilde{\Delta}$ *denote the quantities*

$$\boldsymbol{\mu} = \mathbb{E}[\mathbf{X}_1] \in \mathbb{R}^d; \quad \tilde{\Delta} = \mathbb{E}\Big[\frac{\mathbf{X}_1\mathbf{X}_1^\top}{\mathbf{X}_1^\top\boldsymbol{\mu}}\Big] \in \mathbb{R}^{d\times d}. \tag{14}$$

Also denote by $\tilde{\Sigma}(\mathbf{x})$ *the* $d \times d$ *matrix*

$$\mathbb{E}\Big[\Big(\frac{\tilde{\Delta}^{-1}\mathbf{X}_1}{\mathbf{X}_1^\top\boldsymbol{\mu}} - \frac{\mathbf{x}}{2\mathbf{x}^\top\boldsymbol{\mu}}\Big)\Big(\frac{\mathbf{X}_1^\top\tilde{\Delta}^{-1}}{\mathbf{X}_1^\top\boldsymbol{\mu}} - \frac{\mathbf{x}^\top}{2\mathbf{x}^\top\boldsymbol{\mu}}\Big)\frac{(\mathbf{x}^\top\mathbf{X}_1 - \mathbf{x}^\top\mathbf{X}_1\mathbf{X}_1^\top\mathbf{x})}{\mathbf{x}^\top\boldsymbol{\mu}}\Big]. \tag{15}$$

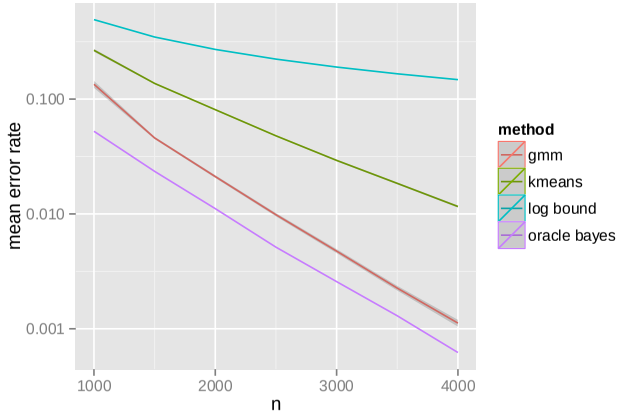


Figure 2: Comparison of classification error for Gaussian mixture model (red curve), K-Means (green curve), and Bayes classifier (cyan curve). The classification errors for each $n \in \{1000, 1250, 1500, \ldots, 4000\}$ were obtained by averaging 100 Monte Carlo iterations and are plotted on a $\log_{10}$ scale. The plot indicates that the assumption of a mixture of multivariate normals can yield non-negligible improvement in the accuracy of the inference procedure. The log-bound curve (purple) shows an upper bound on the error rate as derived in Sussman et al. (2012). Figure duplicated from Athreya et al. (2016).

Then there exists a sequence of $d \times d$ orthogonal matrices $(\mathbf{W}_n)$ such that for each fixed index i and any $\boldsymbol{x} \in \mathbb{R}^d$,

$$\Pr\Big\{n\big(\mathbf{W}_n(\breve{\mathbf{X}}_n)_i - \tfrac{(\mathbf{X}_n)_i}{\sqrt{\sum_j (\mathbf{X}_n)_i^\top (\mathbf{X}_n)_j}}\big) \le \boldsymbol{x}\Big\} \longrightarrow \int \Phi(\boldsymbol{x}, \tilde{\Sigma}(\boldsymbol{y}))dF(\boldsymbol{y}) \tag{16}$$

When F is a mixture of point masses—specifically, when $\mathbf{A} \sim \mathrm{RDPG}(F)$ is a stochastic blockmodel graph—we also have, below, the following limiting conditional distribution for

$$n\left(\mathbf{W}_n(\breve{\mathbf{X}}_n)_i - \frac{(\mathbf{X}_n)_i}{\sqrt{\sum_j (\mathbf{X}_n)_i^\top (\mathbf{X}_n)_j}}\right).$$

Theorem 30 *Assume the setting and notations of Theorem 29 and let*

$$F = \sum_{k=1}^{K} \pi_k \delta_{\nu_k}, \quad \pi_1, \cdots, \pi_K > 0, \sum_k \pi_k = 1$$

be a mixture of K point masses in $\mathbb{R}^d$. Then there exists a sequence of $d \times d$ orthogonal matrices $\mathbf{W}_n$ such that for any fixed index i,

$$\mathbb{P}\Big\{n\big(\mathbf{W}_n(\breve{\mathbf{X}}_n)_i - \tfrac{\nu_k}{\sqrt{\sum_l n_l \nu_k^\top \nu_l}}\big) \le \boldsymbol{z} \mid (\mathbf{X}_n)_i = \nu_k\Big\} \longrightarrow \boldsymbol{\Phi}(\boldsymbol{z}, \tilde{\Sigma}_k) \tag{17}$$

where $\tilde{\Sigma}_k = \tilde{\Sigma}(\nu_k)$ is as defined in Eq. (15) and n_k for $k \in \{1, 2, \ldots, K\}$ denote the number of vertices in $\mathbf{A}$ that are assigned to block k.

Remark 31 *As a special case of Theorem 30, we note that if $\mathbf{A}$ is an Erdős-Rényi graph on n vertices with edge probability p^2 – which corresponds to a random dot product graph where the latent positions are identically p – then for each fixed index i, the normalized Laplacian embedding satisfies*

$$n\big(\breve{\mathbf{X}}_i - \tfrac{1}{\sqrt{n}}\big) \xrightarrow{\mathrm{d}} \mathcal{N}\big(0, \tfrac{1-p^2}{4p^2}\big).$$

Recall that $\breve{\mathbf{X}}_i$ is proportional to $1/\sqrt{d_i}$ where d_i is the degree of the i-th vertex. On the other hand, the adjacency spectral embedding satisfies

$$\sqrt{n}(\hat{\mathbf{X}}_i - p) \xrightarrow{\mathrm{d}} \mathcal{N}(0, 1 - p^2).$$

As another example, let $\mathbf{A}$ be sampled from a stochastic blockmodel with block probability matrix $\mathbf{B} = \left[\begin{smallmatrix} p^2 & pq \\ pq & q^2 \end{smallmatrix}\right]$ and block assignment probabilities $(\pi, 1-\pi)$. Since $\mathbf{B}$ has rank 1, this model corresponds to a random dot product graph where the latent positions are either p with probability π or q with probability $1-\pi$. Then for each fixed index i, the normalized Laplacian embedding satisfies

$$n\big(\breve{\mathbf{X}}_i - \tfrac{p}{\sqrt{n_1 p^2 + n_2 pq}}\big) \xrightarrow{\mathrm{d}} \mathcal{N}\Big(0, \tfrac{\pi p(1-p^2) + (1-\pi)q(1-pq)}{4(\pi p + (1-\pi)q)^3}\Big) \text{ if } \mathbf{X}_i = p, \tag{18}$$

$$n\big(\breve{\mathbf{X}}_i - \tfrac{q}{\sqrt{n_1 pq + n_2 q^2}}\big) \xrightarrow{\mathrm{d}} \mathcal{N}\Big(0, \tfrac{\pi p(1-pq) + (1-\pi)q(1-q^2)}{4(\pi p + (1-\pi)q)^3}\Big) \text{ if } \mathbf{X}_i = q. \tag{19}$$

where n_1 and $n_2 = n - n_1$ are the number of vertices of $\mathbf{A}$ with latent positions p and q. The adjacency spectral embedding, meanwhile, satisfies

$$\sqrt{n}(\hat{\mathbf{X}}_i - p) \xrightarrow{\text{d}} \mathcal{N}\Big(0, \tfrac{\pi p^4(1-p^2)+(1-\pi)pq^3(1-pq)}{(\pi p^2+(1-\pi)q^2)^2}\Big) \text{ if } \mathbf{X}_i = p, \tag{20}$$

$$\sqrt{n}(\hat{\mathbf{X}}_i - q) \xrightarrow{\text{d}} \mathcal{N}\Big(0, \tfrac{\pi p^3q(1-pq)+(1-\pi)q^4(1-q^2)}{(\pi p^2+(1-\pi)q^2)^2}\Big) \text{ if } \mathbf{X}_i = q. \tag{21}$$

We present a sketch of the proof of Theorem 29 in the Appendix, in Section A.3. Due to the intricacy of the proof, however, even in the Appendix we do not provide full details; we instead refer the reader to Tang and Priebe (2018) for the complete proof. Moving forward, we focus on the important implications of these distributional results for subsequent inference, including a mechanism by which to assess the relative desirability of ASE and LSE, which *vary depending on inference task.*

5. Implications for subsequent inference

The previous sections are devoted to establishing the consistency and asymptotic normality of the adjacency and Laplacian spectral embeddings for the estimation of latent positions in an RDPG. In this section, we describe several subsequent graph inference tasks, all of which depend on this consistency: specifically, nonparametric clustering, semiparametric and nonparametric two-sample graph hypothesis testing, and multi-sample graph inference.

5.1 Nonparametric clustering: a comparison of ASE and LSE via Chernoff information

We now discuss how the limit results of Section 4.2 and Section 4.4 can provide insight into subsequent inference. In a recent pioneering work, the authors of Bickel and Sarkar (2015) analyze, in the context of stochastic blockmodel graphs, how the choice of spectral embedding by either the adjacency matrix or the normalized Laplacian matrix impacts subsequent recovery of the block assignments. In particular, they show that a metric constructed from the average distance between the vertices of a block and its cluster centroid for the spectral embedding can be used as a surrogate measure for the performance of the subsequent inference task: namely, the metric is a surrogate measure for the error rate in recovering the vertices to block assignments using the spectral embedding. It is shown in Bickel and Sarkar (2015) that for a large regime of parameters in a two-block stochastic block model, the normalized Laplacian spectral embedding reduces the within-block variance (occasionally by a factor of four) while preserving the between-block variance, as compared to that of the adjacency spectral embedding. This suggests that for a large region of the parameter space for two-block stochastic blockmodels, the spectral embedding of the Laplacian is preferable to the spectral embedding of the adjacency matrix for subsequent inference. However, we observe that the metric in Bickel and Sarkar (2015) is intrinsically tied to the use of K-means as the clustering procedure: specifically, a smaller value of the metric for the Laplacian spectral embedding as compared to that for the adjacency spectral embedding

only implies that clustering the Laplacian spectral embedding using K-means is possibly better than clustering the adjacency spectral embedding using K-means.

Motivated by the above observation, in Tang and Priebe (2018), we propose a metric that is *independent* of any specific clustering procedure, a metric that characterizes the minimum error achievable by *any* clustering procedure that uses only the spectral embedding for the recovery of block assignments in stochastic blockmodel graphs. For a given embedding method, the metric used in Tang and Priebe (2018) is based on the notion of statistical information between the limiting distributions of the blocks. Roughly speaking, smaller statistical information implies less information to discriminate between the blocks of the stochastic blockmodel. More specifically, the limit result in Section 4.2 and Section 4.4 state that, for stochastic blockmodel graphs, conditional on the block assignments the entries of the scaled eigenvectors corresponding to the few largest eigenvalues of the adjacency matrix and the normalized Laplacian matrix converge to a multivariate normal as the number of vertices increases. Furthermore, the associated covariance matrices are not spherical, so K-means clustering for the adjacency spectral embedding or Laplacian spectral embedding does not yield minimum error for recovering the block assignment. Nevertheless, these limiting results also facilitate comparison between the two embedding methods via the classical notion of Chernoff information (Chernoff, 1952).

We now recall the notion of the Chernoff α-divergences (for $\alpha \in (0,1)$) and Chernoff information. Let F_0 and F_1 be two absolutely continuous multivariate distributions in $\Omega = \mathbb{R}^d$ with density functions f_0 and f_1, respectively. Suppose that $Y_1, Y_2, \ldots, Y_m$ are independent and identically distributed random variables, with Y_i distributed either F_0 or F_1. We are interested in testing the simple null hypothesis $\mathbb{H}_0: F = F_0$ against the simple alternative hypothesis $\mathbb{H}_1: F = F_1$. A test T can be viewed as a sequence of mappings $T_m: \Omega^m \mapsto \{0,1\}$ such that given $Y_1 = y_1, Y_2 = y_2, \ldots, Y_m = y_m$, the test rejects $\mathbb{H}_0$ in favor of $\mathbb{H}_1$ if $T_m(y_1, y_2, \ldots, y_m) = 1$; similarly, the test favors $\mathbb{H}_0$ if $T_m(y_1, y_2, \ldots, y_m) = 0$.

The Neyman-Pearson lemma states that, given $Y_1 = y_1, Y_2 = y_2, \ldots, Y_m = y_m$ and a threshold $\eta_m \in \mathbb{R}$, the likelihood ratio test which rejects $\mathbb{H}_0$ in favor of $\mathbb{H}_1$ whenever

$$\Big(\sum_{i=1}^{m} \log f_0(y_i) - \sum_{i=1}^{m} \log f_1(y_i)\Big) \leq \eta_m$$

is the most powerful test at significance level $\alpha_m = \alpha(\eta_m)$, so that the likelihood ratio test minimizes the Type II error β_m subject to the constraint that the Type I error is at most α_m. Assuming that $\pi \in (0,1)$ is a prior probability that $\mathbb{H}_0$ is true. Then, for a given $\alpha_m^* \in (0,1)$, let $\beta_m^* = \beta_m^*(\alpha_m^*)$ be the Type II error associated with the likelihood ratio test when the Type I error is at most α_m^*. The quantity

$$\inf_{\alpha_m^* \in (0,1)} \pi\alpha_m^* + (1-\pi)\beta_m^*$$

is the Bayes risk in deciding between $\mathbb{H}_0$ and $\mathbb{H}_1$ given the m independent random variables $Y_1, Y_2, \ldots, Y_m$. A classical result of Chernoff (Chernoff, 1952, 1956) states that the Bayes risk is intrinsically linked to a quantity known as the *Chernoff information*. More specifically,

let $C(F_0, F_1)$ be the quantity

$$\begin{aligned} C(F_0, F_1) &= -\log\Big[\inf_{t\in(0,1)} \int_{\mathbb{R}^d} f_0^t(\boldsymbol{x}) f_1^{1-t}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}\Big] \\ &= \sup_{t\in(0,1)} \Big[-\log \int_{\mathbb{R}^d} f_0^t(\boldsymbol{x}) f_1^{1-t}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}\Big]. \end{aligned} \tag{22}$$

Then we have

$$\lim_{m\to\infty} \frac{1}{m} \inf_{\alpha_m^*\in(0,1)} \log(\pi\alpha_m^* + (1-\pi)\beta_m^*) = -C(F_0, F_1). \tag{23}$$

Thus $C(F_0, F_1)$, the Chernoff information between F_0 and F_1, is the *exponential* rate at which the Bayes error $\inf_{\alpha_m^*\in(0,1)} \pi\alpha_m^* + (1-\pi)\beta_m^*$ decreases as $m \to \infty$; we note that the Chernoff information is independent of π. We also define, for a given $t \in (0,1)$ the Chernoff divergence $C_t(F_0, F_1)$ between F_0 and F_1 by

$$C_t(F_0, F_1) = -\log \int_{\mathbb{R}^d} f_0^t(\boldsymbol{x}) f_1^{1-t}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}.$$

The Chernoff divergence is an example of a f-divergence as defined in, for example, the work of Csizár (1967) and Ali and Shelvey (1966). When $t = 1/2$, $C_t(F_0, F_1)$ is the Bhattacharyya distance between F_0 and F_1. Recall that any f-divergence satisfies the Information Processing Lemma and is invariant with respect to invertible transformations (Liese and Vadja, 2006). Therefore, any f-divergence such as the Kullback-Liebler divergence can also be used to compare the two embedding methods; the Chernoff information is particularly attractive because of its explicit relationship with the Bayes risk.

The characterization of Chernoff information in Eq. (23) can be extended to $K + 1 \geq 2$ hypotheses. Let $F_0, F_1, \ldots, F_K$ be distributions on $\mathbb{R}^d$ and suppose that $Y_1, Y_2, \ldots, Y_m$ are independent and identically distributed random variables with Y_i distributed $F \in \{F_0, F_1, \ldots, F_K\}$. We are thus interested in determining the distribution of the Y_i among the $K+1$ hypothesis $\mathbb{H}_0 : F = F_0, \ldots, \mathbb{H}_K : F = F_K$. Suppose also that hypothesis $\mathbb{H}_k$ has *a priori* probability π_k. Then for any decision rule δ, the risk of δ is $r(\delta) = \sum_k \pi_k \sum_{l\neq k} \alpha_{lk}(\delta)$ where $\alpha_{lk}(\delta)$ is the probability of accepting hypothesis $\mathbb{H}_l$ when hypothesis $\mathbb{H}_k$ is true. Then we have, from Leang and Johnson (1997),

$$\inf_{\delta} \lim_{m\to\infty} \frac{r(\delta)}{m} = -\min_{k\neq l} C(F_k, F_l). \tag{24}$$

where the infimum is over all decision rules δ. That is, for any δ, $r(\delta)$ decreases to 0 as $m \to \infty$ at a rate no faster than $\exp(-m \min_{k\neq l} C(F_k, F_l))$. It is also shown in Leang and Johnson (1997) that the *Maximum A Posterior* decision rule achieves this rate.

Finally, if $F_0 = \mathcal{N}(\mu_0, \Sigma_0)$ and $F_1 = \mathcal{N}(\mu_1, \Sigma_1)$, then denoting by $\Sigma_t = t\Sigma_0 + (1-t)\Sigma_1$, the Chernoff information $C(F_0, F_1)$ between F_0 and F_1 is given by

$$C(F_0, F_1) = \sup_{t\in(0,1)} \Big(\frac{t(1-t)}{2}(\mu_1 - \mu_2)^\top \Sigma_t^{-1}(\mu_1 - \mu_2) + \frac{1}{2}\log\frac{|\Sigma_t|}{|\Sigma_0|^t|\Sigma_1|^{1-t}}\Big).$$

Comparison of the performance of the Laplacian spectral embedding and the adjacency spectral embedding for recovering the block assignments now proceeds as follows. Let $\mathbf{B} \in [0,1]^{K\times K}$ and $\boldsymbol{\pi} \in \mathbb{R}^K$ be the matrix of block probabilities and the vector of block assignment probabilities for a K-block stochastic blockmodel. We shall assume that $\mathbf{B}$ is positive semidefinite. Then given an n vertex instantiation of the SBM graph with parameters $(\boldsymbol{\pi}, \mathbf{B})$, for sufficiently large n, the large-sample optimal error rate for recovering the block assignments when adjacency spectral embedding is used as the initial embedding step can be characterized by the quantity $\rho_{\mathrm{A}} = \rho_{\mathrm{A}}(n)$ defined by

$$\rho_{\mathrm{A}} = \min_{k\neq l} \sup_{t\in(0,1)} \frac{1}{2}\log\frac{|\Sigma_{kl}(t)|}{|\Sigma_k|^t|\Sigma_l|^{1-t}} + \frac{nt(1-t)}{2}(\nu_k - \nu_l)^\top \Sigma_{kl}^{-1}(t)(\nu_k - \nu_l) \tag{25}$$

where $\Sigma_{kl}(t) = t\Sigma_k + (1-t)\Sigma_l$; $\Sigma_k = \Sigma(\nu_k)$ and $\Sigma_l = \Sigma(\nu_l)$ are as defined in Theorem 27. We recall Eq. (24), in particular the fact that as ρ_{A} increases, the large-sample optimal error rate decreases. Similarly, the large-sample optimal error rate when Laplacian spectral embedding is used as the pre-processing step can be characterized by the quantity $\rho_{\mathrm{L}} = \rho_{\mathrm{L}}(n)$ defined by

$$\rho_{\mathrm{L}} = \min_{k\neq l} \sup_{t\in(0,1)} \frac{1}{2}\log\frac{|\tilde{\Sigma}_{kl}(t)|}{|\tilde{\Sigma}_k|^t|\tilde{\Sigma}_l|^{1-t}} + \frac{nt(1-t)}{2}(\tilde{\nu}_k - \tilde{\nu}_l)^\top \tilde{\Sigma}_{kl}^{-1}(t)(\tilde{\nu}_k - \tilde{\nu}_l) \tag{26}$$

where $\tilde{\Sigma}_{kl}(t) = t\tilde{\Sigma}_k + (1-t)\tilde{\Sigma}_l$ with $\tilde{\Sigma}_k = \tilde{\Sigma}(\nu_k)$ and $\tilde{\Sigma}_l = \tilde{\Sigma}(\nu_l)$ as defined in Theorem 30, and $\tilde{\nu}_k = \nu_k/(\sum_{k'} \pi_{k'} \nu_k^\top \nu_{k'})^{1/2}$. Note that ρ_{A} and ρ_{L} are similar to a weighted Mahalanobis distance. We emphasize that we have made the simplifying assumption that $n_k = n\pi_k$ in our expression for $\tilde{\nu}_k$ in Eq. (26). This is for ease of comparison between ρ_{A} and ρ_{L} in our subsequent discussion.

The ratio $\rho_{\mathrm{A}}/\rho_{\mathrm{L}}$ is a surrogate measure of the relative large-sample performance of the adjacency spectral embedding as compared to the Laplacian spectral embedding for subsequent inference, at least in the context of stochastic blockmodel graphs. That is to say, for given parameters $\boldsymbol{\pi}$ and $\mathbf{B}$, if $\rho_{\mathrm{A}}/\rho_{\mathrm{L}} > 1$, then adjacency spectral embedding is preferred over Laplacian spectral embedding when n, the number of vertices in the graph, is sufficiently large; similarly, if $\rho_{\mathrm{A}}/\rho_{\mathrm{L}} < 1$, then Laplacian spectral embedding is preferred over adjacency spectral embedding.

As an illustration of the ratio $\rho_{\mathrm{A}}/\rho_{\mathrm{L}}$, we first consider the collection of 2-block stochastic blockmodels where $\mathbf{B} = \left[\begin{smallmatrix} p^2 & pq \\ pq & q^2 \end{smallmatrix}\right]$ for $p, q \in (0,1)$ and $\boldsymbol{\pi} = (\pi_1, \pi_2)$ with $\pi_1 + \pi_2 = 1$. We note that these $\mathbf{B}$ also have rank 1 and thus the Chernoff information can be computed explicitly. Then for sufficiently large n, ρ_{A} is approximately

$$\rho_{\mathrm{A}} \approx \sup_{t\in(0,1)} \frac{nt(1-t)}{2}(p-q)^2(t\sigma_1^2 + (1-t)\sigma_2^2)^{-1}$$

where σ_1 and σ_2 are as specified in Eq. (20) and Eq. (21), respectively. Simple calculations yield

$$\rho_{\mathrm{A}} \approx \frac{n(p-q)^2(\pi_1 p^2 + \pi_2 q^2)^2}{2\Big(\sqrt{\pi_1 p^4(1-p^2) + \pi_2 pq^3(1-pq)} + \sqrt{\pi_1 p^3 q(1-pq) + \pi_2 q^4(1-q^2)}\Big)^2}$$

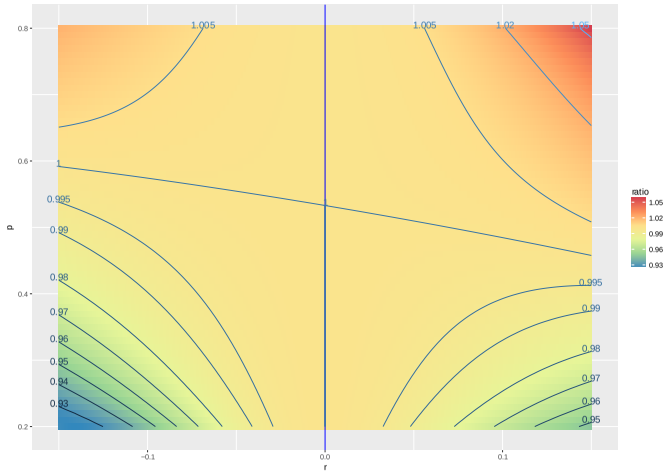


Figure 3: The ratio ρ_A/ρ_L displayed for various values of $p \in [0.2, 0.8]$ and $r = q - p \in [-0.15, 0.15]$. The labeled lines are the contour lines for ρ_A/ρ_L. Figure duplicated from Tang and Priebe (2018).

for sufficiently large n. Similarly, denoting by $\tilde{\sigma}_1^2$ and $\tilde{\sigma}_2^2$ the variances specified in Eq. (18) and Eq. (19), we have

$$\begin{aligned}\rho_L &\approx \sup_{t\in(0,1)} \frac{nt(1-t)}{2}\Big(\frac{p}{\sqrt{\pi_1 p^2+\pi_2 pq}} - \frac{q}{\sqrt{\pi_1 pq+\pi_2 q^2}}\Big)^2 (t\tilde{\sigma}_1^2 + (1-t)\tilde{\sigma}_2^2)^{-1} \\ &\approx \frac{2n(\sqrt{p}-\sqrt{q})^2(\pi_1 p+\pi_2 q)^2}{\Big(\sqrt{\pi_1 p(1-p^2)+\pi_2 q(1-pq)}+\sqrt{\pi_1 p(1-pq)+\pi_2 q(1-q^2)}\Big)^2} \\ &\approx \frac{2n(p-q)^2(\pi_1 p+\pi_2 q)^2}{(\sqrt{p}+\sqrt{q})^2\Big(\sqrt{\pi_1 p(1-p^2)+\pi_2 q(1-pq)}+\sqrt{\pi_1 p(1-pq)+\pi_2 q(1-q^2)}\Big)^2}\end{aligned}$$

for sufficiently large n. Fixing $\boldsymbol{\pi} = (0.6, 0.4)$, we compute the ratio ρ_A/ρ_L for a range of p and q values, with $p \in [0.2, 0.8]$ and $q = p + r$ where $r \in [-0.15, 0.15]$. The results are plotted in Figure 3. The y-axis of Figure 3 denotes the values of p and the x axis are the values of r. We see from the above figure that, in general, neither of the methods, namely adjacency spectral clustering or normalized Laplacian spectral embedding followed by clustering via Gaussian mixture models, dominates over the whole (p, r) parameter space. However, in general, one can easily show that LSE is preferable over ASE whenever the block probability matrix $\mathbf{B}$ becomes sufficiently sparse. Determination of similarly intuitive conditions for which ASE dominates over LSE is considerably more subtle and is the topic of current research. But in general, we observe that ASE dominates over LSE whenever the entries of $\mathbf{B}$ are relatively large.

Finally we consider the collection of stochastic blockmodels with parameters $\boldsymbol{\pi}$ and $\mathbf{B}$ where

$$\mathbf{B} = \begin{bmatrix} p & q & q \\ q & p & q \\ q & q & p \end{bmatrix}, \quad p, q \in (0,1), \text{ and } \boldsymbol{\pi} = (0.8, 0.1, 0.1). \tag{27}$$

First we compute the ratio ρ_A/ρ_L for $p \in [0.3, 0.9]$ and $r = q - p$ with $r \in [-0.2, -0.01]$. The results are plotted in Figure 4, with the y-axis of Figure 4 being the values of p and the x-axis being the values of r. Once again we see that for the purpose of subsequent inference, neither embedding methods dominates over the whole parameter space and that LSE is still preferable to ASE for smaller values of p and q and that ASE is preferable to LSE for larger values of p and q.

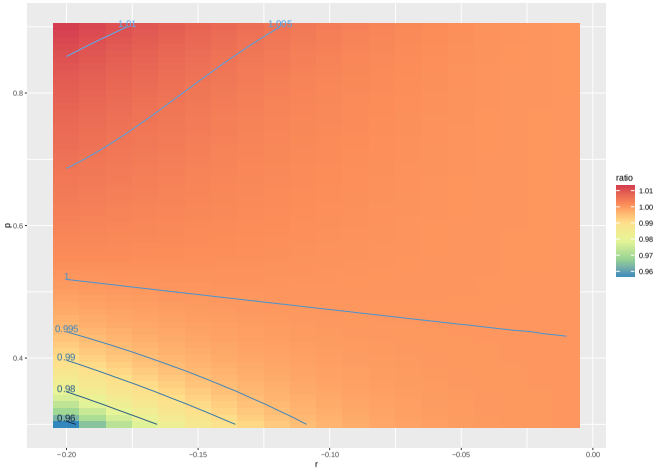


Figure 4: The ratio ρ_A/ρ_L displayed for various values of $p \in [0.2, 0.8]$ and $r = q - p \in [-0.2, -0.01]$ for the 3-block stochastic blockmodel of Eq. (27). The labeled lines are the contour lines for ρ_A/ρ_L. Figure duplicated from Tang and Priebe (2018).

5.2 Hypothesis testing

The field of multi-sample graph inference is comparatively new, and the development of a comprehensive machinery for two-sample hypothesis testing for random graphs is of both theoretical and practical importance. The test procedures in Tang et al. (2017a) and Tang et al. (2017b), both of which leverage the adjacency spectral embedding to test hypotheses of equality or equality in distribution for random dot product graphs, are among the only principled methodologies currently available. In both cases, the accuracy of the adjacency spectral embedding as an estimate of the latent positions is the key to constructing a

test statistic. Specifically, in Tang et al. (2017a), we give a new and improved bound, Theorem 32 below, for the Frobenius norm of the difference between the original latent positions and the estimated latent positions obtained from the embedding. This bound is then used to establish a valid and consistent test for the semiparameteric hypothesis test of equality for latent positions in a pair of vertex-aligned random dot product graphs. In the nonparametric case, in Tang et al. (2017b), we demonstrate how the adjacency spectral embedding can be integrated with a kernel density estimator to accurately estimate the underlying distribution F for in a random dot product graph with i.i.d latent positions.

We reproduce here several important points from Tang et al. (2017a,b). To begin our summary of two-sample hypothesis testing for graphs, we first consider the problem of developing a test for the hypothesis that two random dot product graphs on the same vertex set, with known vertex correspondence, have the same generating latent position or have generating latent positions that are scaled or diagonal transformations of one another, modulo a possible (non-identifiable) orthogonal transformation. This framework includes, as a special case, a test for whether two stochastic blockmodels have the same or related block probability matrices. In this two-sample testing problem, though, the parameter dimension grows as the sample size grows. Therefore, the problem is not precisely analogous to classical two-sample tests for, say, the difference of two parameters belonging to some fixed Euclidean space, in which an increase in data has no effect on the dimension of the parameter. The problem is also not nonparametric, since we view our latent positions as fixed and impose specific distributional requirements on the data—that is, on the adjacency matrices. Indeed, we regard the problem as semiparametric, and Tang et al. (2017a) adapts the traditional definition of consistency to this setting. In particular, for the test procedure we describe, power will increase to one for alternatives in which the difference between the two latent positions grows with the sample size.

Our test procedure is, at first glance, deceptively simple: given a pair of adjacency matrices $\mathbf{A}$ and $\mathbf{B}$ for two d-dimensional random dot product graphs, we generate their adjacency spectral embeddings, denoted $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$, respectively, and compute an appropriately normalized version of the so-called *Procrustes fit* or *Procrustes distance* between the two embeddings:

$$\min_{\mathbf{W}\in\mathcal{O}^{d\times d}} \|\hat{\mathbf{X}} - \hat{\mathbf{Y}}\mathbf{W}\|_F$$

(Recall that such a fit is necessary because of the inherent nonidentifiability of the random dot product model.)

Understanding the limiting distribution of this test statistic is more complicated, however, and appropriately framing the set of null and alternative hypothesis for which the test is valid and consistent (that is, a level α-test with power converging to 1 as $n \to \infty$) is delicate. To that end, we first state the key concentration inequality for $\min_{\mathbf{W}\in\mathcal{O}^{d\times d}} \|\hat{\mathbf{X}} - \hat{\mathbf{Y}}\mathbf{W}\|_F$.

Theorem 32 *Suppose $\mathbf{P} = \mathbf{X}\mathbf{X}^\top$ is an $n \times n$ probability matrix of rank d. Suppose also that there exists $\epsilon > 0$ such that $\delta(\mathbf{P}) > (\log n)^{2+\epsilon}$. Let $c > 0$ be arbitrary but fixed. Then there exists a $n_0(c)$ and a universal constant $C \geq 0$ such that if $n \geq n_0$ and $n^{-c} < \eta < 1/2$, then*

there exists a deterministic $\mathbf{W} \in \mathcal{O}(d)$ such that, with probability at least $1 - 3\eta$,

$$\left| \|\hat{\mathbf{X}} - \mathbf{X}\mathbf{W}\|_F - C(\mathbf{X}) \right| \leq \frac{Cd\log(n/\eta)}{C(\mathbf{X})\sqrt{\gamma^5(\mathbf{P})\delta(\mathbf{P})}} \tag{28}$$

where $C(\mathbf{X})$ is a function of $\mathbf{X}$ given by

$$C(\mathbf{X}) = \sqrt{\operatorname{tr}\, \mathbf{S}_\mathbf{P}^{-1/2}\mathbf{U}_\mathbf{P}^\top \mathbb{E}[(\mathbf{A}-\mathbf{P})^2]\mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{-1/2}} \tag{29}$$

where $\mathbb{E}[(\mathbf{A}-\mathbf{P})^2]$ is taken with respect to $\mathbf{A}$ and conditional on $\mathbf{X}$.

We note that the proof of this theorem consists of two pieces: it is straightforward to show that the Frobenius norm bound of Lemma 50 implies that

$$\|\hat{\mathbf{X}} - \mathbf{X}\mathbf{W}\|_F = \|(\mathbf{A}-\mathbf{P})\mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{-1/2}\|_F + O(d\log(n)\delta^{-1/2}(\mathbf{P})\gamma^{-5/2}(\mathbf{P}))$$

To complete the theorem, then, in Tang et al. (2017a) we provide a concentration inequality for $\|(\mathbf{A}-\mathbf{P})\mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{-1/2}\|_F^2$, showing that

$$\left| \|(\mathbf{A}-\mathbf{P})\mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{-1/2}\|_F^2 - C^2(\mathbf{X}) \right| \leq \frac{14\sqrt{2d}\log(n/\eta)}{\gamma(\mathbf{P})\sqrt{\delta(\mathbf{P})}}. \tag{30}$$

where $C(\mathbf{X})$ is as defined in (29). We do not go into the details of this concentration inequality here, but rather point the reader to Tang et al. (2017a). We observe, however, that this inequality has immediate consequences for two-sample testing for random dot product graphs. For two random dot product graphs with probability matrices $\mathbf{P} = \mathbf{X}\mathbf{X}^\top$ and $\mathbf{Q} = \mathbf{Y}\mathbf{Y}^\top$, consider the null hypothesis $\mathbf{X} = \mathbf{Y}\mathbf{W}$ for some orthogonal $\mathbf{W}$. It can be shown that $\min_{\mathbf{W}\in\mathcal{O}^{d\times d}} \|\hat{\mathbf{X}} - \hat{\mathbf{Y}}\mathbf{W}\|_F$ is the basis for a valid and consistent test. We emphasize, though, that as the graph size n increases, the $n \times d$ matrix of latent positions also increases in size. As a consequence of this, we consider the following notion of consistency in this semiparametric setting. As an aside on the notation, in this section, we consider a sequence of graphs with latent positions, all indexed by n; thus, as we noted in our preliminary remarks on notation, $\mathbf{X}_n$ and $\hat{\mathbf{X}}_n$ refer to the *matrices* of true and estimated latent positions in this sequence. We let

$$\mathbf{X}_n =_W \mathbf{Y}_n$$

denote that $\mathbf{X}_n$ and $\mathbf{Y}_n$ are equivalent up to orthogonal transformation; that is

$$\mathbf{X}_n = \mathbf{Y}_n\mathbf{W}$$

for some orthogonal $\mathbf{W} \in \mathbb{R}^{d\times d}$.

Definition 33 *Let $(\mathbf{X}_n, \mathbf{Y}_n)_{n\in\mathbb{N}}$, be a given sequence of latent positions, where $\mathbf{X}_n$ and $\mathbf{Y}_n$ are both in $\mathbb{R}^{n\times d}$. A test statistic T_n and associated rejection region R_n to test the null hypothesis*

$$H_0^n : \mathbf{X}_n =_W \mathbf{Y}_n \quad \text{against} \quad H_a^n : \mathbf{X}_n \neq_W \mathbf{Y}_n$$

is a consistent, asymptotically level α test *if for any $\eta > 0$, there exists $n_0 = n_0(\eta)$ such that*

(i) If $n > n_0$ and H_a^n is true, then $P(T_n \in R_n) > 1 - \eta$

(ii) If $n > n_0$ and H_0^n is true, then $P(T_n \in R_n) \leq \alpha + \eta$

With this definition of consistency, we obtain the following theorem on two-sample testing for random dot products on the same vertex set and with known vertex correspondence.

Theorem 34 *For each fixed n, consider the hypothesis test*

$$H_0^n : \mathbf{X}_n =_W \mathbf{Y}_n \quad \text{versus} \quad H_a^n : \mathbf{X}_n \neq_W \mathbf{Y}_n$$

where $\mathbf{X}_n$ and $\mathbf{Y}_n \in \mathbb{R}^{n \times d}$ are matrices of latent positions for two random dot product graphs. Let $\hat{\mathbf{X}}_n$ and $\hat{\mathbf{Y}}_n$ be the adjacency spectral embeddings of $\mathbf{A}_n \sim \text{Bernoulli}(\mathbf{X}_n\mathbf{X}_n^\top)$ and $\mathbf{B}_n \sim \text{Bernoulli}(\mathbf{Y}_n\mathbf{Y}_n^\top)$, respectively. Define the test statistic T_n as follows:

$$T_n = \frac{\min_{\mathbf{W} \in \mathcal{O}(d)} \|\hat{\mathbf{X}}_n\mathbf{W} - \hat{\mathbf{Y}}_n\|_F}{\sqrt{d\gamma^{-1}(\mathbf{A}_n)} + \sqrt{d\gamma^{-1}(\mathbf{B}_n)}}. \tag{31}$$

Let $\alpha \in (0,1)$ be given. Then for all $C > 1$, if the rejection region is $R := \{t \in \mathbb{R} : t \geq C\}$, then there exists an $n_1 = n_1(\alpha, C) \in \mathbb{N}$ such that for all $n \geq n_1$, the test procedure with T_n and rejection region R is an at most level α test, that is, for all $n \geq n_1$, if $\mathbf{X}_n =_W \mathbf{Y}_n$, then $\mathbb{P}(T_n \in R) \leq \alpha$. Furthermore, suppose the sequence of latent positions $\{\mathbf{X}_n\}$ and $\{\mathbf{Y}_n\}$, $n \in \mathbb{N}$, adhere to the following requirements:

(i) The latent positions satisfy the eigengap assumptions in Assumption 1;

(ii) With d_n denoting the quantity $d_n := \min_{\mathbf{W} \in \mathcal{O}(d)} \|\mathbf{X}_n\mathbf{W} - \mathbf{Y}_n\|$, suppose $d_n \neq 0$ for infinitely many n. Let $t_1 = \min\{k > 0 : d_k > 0\}$ and sequentially define $t_n = \min\{k > t_{n-1} : d_k > 0\}$. Let $b_n = d_{t_n}$. Suppose $\liminf b_n = \infty$.

Then this test procedure is consistent in the sense of Definition 33 over this sequence of latent positions.

Remark 35 *This result does not require that $\mathbf{A}_n$ and $\mathbf{B}_n$ be independent for any fixed n, nor that the sequence of pairs $(\mathbf{A}_n, \mathbf{B}_n)$, $n \in \mathbb{N}$, be independent. Further, with regard to requirement (ii) above, we note that Theorem 34 is written to emphasize consistency in the sense of Definition 33, even in a case when, for example, the latent position sequence is such that $\mathbf{X}_n =_W \mathbf{Y}_n$ for all even n, but $\mathbf{X}_n$ and $\mathbf{Y}_n$ are sufficiently far apart for odd n. In addition, the insistence that $\liminf b_k = \infty$ can also be relaxed somewhat. Specifically, consistency is achieved as long as*

$$\liminf_{n \to \infty} \Big(\|\mathbf{X}_n\mathbf{W} - \mathbf{Y}_n\|_F - C(\mathbf{X}_n) - C(\mathbf{Y}_n) \Big) > 0.$$

It also possible to construct analogous tests for latent positions related by scaling factors, or, in the case of the degree-corrected stochastic block model, by projection. We summarize these below, beginning with the case of scaling.

For the scaling case, let $\mathcal{C} = \mathcal{C}(\mathbf{Y}_n)$ denote the class of all positive constants c for which all the entries of $c^2\mathbf{Y}_n\mathbf{Y}_n^\top$ belong to the unit interval. We wish to test the null hypothesis

$H_0 : \mathbf{X}_n =_W c_n\mathbf{Y}_n$ for some $c_n \in \mathcal{C}$ against the alternative $H_a : \mathbf{X}_n \neq_W c_n\mathbf{Y}_n$ for any $c_n \in \mathcal{C}$. Recall, again, that our notation denotes equivalence up to $d \times d$ orthogonal transformation $\mathbf{W}$. In what follows below, we will only write $c_n > 0$, but will always assume that $c_n \in \mathcal{C}$, since the problem is ill-posed otherwise. The test statistic T_n is now a simple modification of the one used in Theorem 34: for this test, we compute a Procrustes distance between scaled adjacency spectral embeddings for the two graphs.

Theorem 36 *For each fixed n, consider the hypothesis test*

$$H_0^n : \mathbf{X}_n =_W c_n\mathbf{Y}_n \quad \text{for some } c_n > 0 \quad \text{versus} \quad H_a^n : \mathbf{X}_n \neq_W c_n\mathbf{Y}_n \quad \text{for all } c_n > 0$$

where $\mathbf{X}_n$ and $\mathbf{Y}_n \in \mathbb{R}^{n \times d}$ are latent positions for two random dot product graphs with adjacency matrices $\mathbf{A}_n$ and $\mathbf{B}_n$, respectively. Define the test statistic T_n as follows:

$$T_n = \frac{\min_{\mathbf{W} \in \mathcal{O}(d)} \|\hat{\mathbf{X}}_n\mathbf{W}/\|\hat{\mathbf{X}}_n\|_F - \hat{\mathbf{Y}}_n/\|\hat{\mathbf{Y}}_n\|_F\|_F}{2\sqrt{d\gamma^{-1}(\mathbf{A}_n)}/\|\hat{\mathbf{X}}_n\|_F + 2\sqrt{d\gamma^{-1}(\mathbf{B}_n)}/\|\hat{\mathbf{Y}}_n\|_F}. \tag{32}$$

Let $\alpha \in (0,1)$ be given. Then for all $C > 1$, if the rejection region is $R := \{t \in \mathbb{R} : t \geq C\}$, then there exists an $n_1 = n_1(\alpha, C) \in \mathbb{N}$ such that for all $n \geq n_1$, the test procedure with T_n and rejection region R is an at most level α test. Furthermore, consider the sequence of latent position $\{\mathbf{X}_n\}$ and $\{\mathbf{Y}_n\}$, $n \in \mathbb{N}$, satisfying Assumption 1 and denote by d_n the quantity

$$d_n := \frac{\min_{\mathbf{W} \in \mathcal{O}(d)} \|\mathbf{X}_n\mathbf{W}/\|\mathbf{X}_n\|_F - \mathbf{Y}_n/\|\mathbf{Y}_n\|_F\|_F}{1/\|\mathbf{X}_n\|_F + 1/\|\mathbf{Y}_n\|_F} = \frac{\min_{\mathbf{W} \in \mathcal{O}(d)} \|\mathbf{X}_n\|\mathbf{Y}_n\|_F\mathbf{W} - \mathbf{Y}_n\|\mathbf{X}_n\|_F\|_F}{\|\mathbf{X}_n\|_F + \|\mathbf{Y}_n\|_F} \tag{33}$$

Suppose $d_n \neq 0$ for infinitely many n. Let $t_1 = \min\{k > 0 : d_k > 0\}$ and sequentially define $t_n = \min\{k > t_{n-1} : d_k > 0\}$. Let $b_n = d_{t_n}$. If $\liminf b_n = \infty$, then this test procedure is consistent in the sense of Definition 33 over this sequence of latent positions.

We next consider the case of testing whether the latent positions are related by a diagonal transformation: that is, whether $H_0 : \mathbf{X}_n =_W \mathbf{D}_n\mathbf{Y}_n$ for some diagonal matrix $\mathbf{D}_n$. We proceed analogously to the scaling case, above, by defining the class $\mathcal{E} = \mathcal{E}(\mathbf{Y}_n)$ to be all positive diagonal matrices $\mathbf{D}_n \in \mathbb{R}^{n \times n}$ such that $\mathbf{D}_n\mathbf{Y}_n\mathbf{Y}_n^\top\mathbf{D}_n$ has all entries in the unit interval.

As before, we will always assume that $\mathbf{D}_n$ belongs to $\mathcal{E}$, even if this assumption is not explicitly stated. The test statistic T_n in this case is again a simple modification of the one used in Theorem 34. However, for technical reasons, our proof of consistency requires an additional condition on the minimum Euclidean norm of each row of the matrices $\mathbf{X}_n$ and $\mathbf{Y}_n$. To avoid certain technical issues, we impose a slightly stronger density assumption on our graphs for this test. These assumptions can be weakened, but at the cost of interpretability. The assumptions we make on the latent positions, which we summarize here, are moderate restrictions on the sparsity of the graphs.

Assumption 2 *We assume that there exists $d \in \mathbb{N}$ such that for all n, $\mathbf{P}_n$ is of rank d. Further, we assume that there exist constants $\epsilon_1 > 0$, $\epsilon_2 \in (0,1)$, $c_0 > 0$ and $n_0(\epsilon_1, \epsilon_2, c) \in \mathbb{N}$*

such that for all $n \geq n_0$:

$$\gamma(\mathbf{P}_n) \geq c_0; \qquad \delta(\mathbf{P}_n) \geq (\log n)^{2+\epsilon_1}; \qquad \min_i \|X_i\| > \left(\frac{\log n}{\sqrt{\delta(\mathbf{P}_n)}}\right)^{1-\epsilon_2} \tag{34}$$

We then have the following result.

Theorem 37 *For each fixed n, consider the hypothesis test*

$$H_0^n: \mathbf{X}_n =_W \mathbf{D}_n\mathbf{Y}_n \quad \text{for some } \mathbf{D}_n \in \mathcal{E} \quad \text{versus} \quad H_a^n: \mathbf{X}_n \neq_W \mathbf{D}_n\mathbf{Y}_n \quad \text{for any } \mathbf{D}_n \in \mathcal{E}$$

where $\mathbf{X}_n$ and $\mathbf{Y}_n \in \mathbb{R}^{n\times d}$ are matrices of latent positions for two random dot product graphs. For any matrix $\mathbf{Z} \in \mathbb{R}^{n\times d}$, let $\mathcal{D}(\mathbf{Z})$ be the diagonal matrix whose diagonal entries are the Euclidean norm of the rows of $\mathbf{Z}$ and let $\mathcal{P}(\mathbf{Z})$ be the matrix whose rows are the projection of the rows of $\mathbf{Z}$ onto the unit sphere. We define the test statistic as follows:

$$T_n = \frac{\min_{\mathbf{W}\in\mathcal{O}(d)} \|\mathcal{P}(\hat{\mathbf{X}}_n)\mathbf{W} - \mathcal{P}(\hat{\mathbf{Y}}_n)\|_F}{2\sqrt{d\gamma^{-1}(\mathbf{A})}\|\mathcal{D}^{-1}(\hat{\mathbf{X}}_n)\| + 2\sqrt{d\gamma^{-1}(\mathbf{B}_n)}\|\mathcal{D}^{-1}(\hat{\mathbf{Y}}_n)\|}. \tag{35}$$

where we write $\mathcal{D}^{-1}(\mathbf{Z})$ for $(\mathcal{D}(\mathbf{Z}))^{-1}$. Note that $\|\mathcal{D}^{-1}(\mathbf{Z})\| = 1/(\min_i \|Z_i\|)$.

Let $\alpha \in (0,1)$ be given. Then for all $C > 1$, if the rejection region is $R := \{t \in \mathbb{R} : t \geq C\}$, then there exists an $n_1 = n_1(\alpha, C) \in \mathbb{N}$ such that for all $n \geq n_1$, the test procedure with T_n and rejection region R is an at most level-α test. Furthermore, consider the sequence of latent position $\{\mathbf{X}_n\}$ and $\{\mathbf{Y}_n\}$, $n \in \mathbb{N}$, satisfying Assumption 2 and denote by d_n the quantity

$$d_n := \frac{\min_{\mathbf{W}\in\mathcal{O}(d)} \|\mathcal{P}(\mathbf{X}_n)\mathbf{W} - \mathcal{P}(\mathbf{Y}_n)\|_F}{\|\mathcal{D}^{-1}(\mathbf{X})\| + \|\mathcal{D}^{-1}(\mathbf{Y})\|} = D_{\mathcal{P}}(\mathbf{X}_n, \mathbf{Y}_n) \tag{36}$$

Suppose $d_n \neq 0$ for infinitely many n. Let $t_1 = \min\{k > 0 : d_k > 0\}$ and sequentially define $t_n = \min\{k > t_{n-1} : d_k > 0\}$. Let $b_n = d_{t_n}$. If $\liminf b_n = \infty$, then this test procedure is consistent in the sense of Definition 33 over this sequence of latent positions.

This collection of semiparametric tests has numerous applications in graph comparison; in Section 6 , we describe its use in connectomics and brain scan data. We stress, though, that the Procrustes transformations are rather cumbersome, and they limit our ability to generalize these procedures to graph comparisons involving more than two graphs. As a consequence, it can be useful to consider *joint* or *omnibus* embeddings, in which adjacency matrices for multiple graphs on the same vertex set are jontly embedded into a single (larger-dimensional) space, but *with a distinct representation for each graph.* For an illuminating joint graph inference study on the *C. elegans* connectome that addresses somewhat different questions from semiparametric testing, see Chen et al. (2016). Simultaneously embedding multiple graphs into a shared space allows comparison of graphs without the need to perform pairwise alignments of graph embeddings. Further, a distinct representation of each graph renders the omnibus embedding especially useful for subsequent comparative graph inference.

5.3 Omnibus embedding

In Levin et al. (2017), we show that an omnibus embedding—that is, an embedding of multiple graphs into a single shared space—can yield consistent estimates of underlying latent positions. Moreover, like the adjacency spectral embedding for a single graph, the rows of this omnibus embedding, suitably-scaled, are asymptotically normally distributed. As might be anticipated, the use of multiple independent graphs generated from the same latent positions, as opposed just a single graph, yields a reduction in variance for the estimated latent positions, and since the omnibus embedding provides a distinct representation for each graph, subsequently averaging these estimates reduces the variance further still. Further, the omnibus embedding allows us to compare graphs without cumbersome Procrustes alignments, but with nevertheless very similar empirical power, at least in certain simple cases, as that of the Procrustes-centered semiparametric test decribed earlier. In sum, the omnibus embedding can deliver nearly-optimal estimation of latent positions, and improvements in testing. Finally, when embedding m d-dimensional random dot product graphs, each with n vertices, the omnibus embedding provides a matrix in $\mathbb{R}^{nm\times d}$, with a separate representation in $\mathbb{R}^d$ for each vertex *in each graph.* We surmise that this distinct representation may be especially helpful for inference tasks such as graph regression, estimation of additional vertex covariates, or changepoint detection in a time series of graphs, and the impact of the omnibus embedding on such problems is a topic of curren investigation.

The discussion we give here is condensed and reproduced in significant parts from Levin et al. (2017). To construct the omnibus embedding, we consider a collection of m random dot product graphs, all with the same generating latent positions. This motivates the following definition:

Definition 38 (Conditionally independent, identically distributed joint RDPG) *Let F be a d-dimensional inner product distribution on $\mathbb{R}^d$. We say that random graphs $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(m)}$ are distributed as a* conditionally independent, identically distributed joint random dot product graph (ciJRDPG) *and write*

$$(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(m)}, \mathbf{X}) \sim \text{ciJRDPG}(F, n, m)$$

if $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n]^\top \in \mathbb{R}^{n\times d}$ has its (transposed) rows distributed i.i.d. as $\mathbf{X}_i \sim F$, and we have marginal distributions $(\mathbf{A}^{(k)}, \mathbf{X}) \sim \text{RDPG}(F, n)$ for each $k = 1, 2, \ldots, m$. That is, the $\mathbf{A}^{(k)}$ are conditionally independent given $\mathbf{X}$, with edges independently distributed as $\mathbf{A}^{(k)}_{i,j} \sim \text{Bernoulli}((\mathbf{X}\mathbf{X}^\top)_{ij})$ for all $1 \leq i < j \leq n$ and all $k \in [m]$.

Remark 39 *We observe that the notion of a* joint *random dot product graph can be much more general than this definition alone, and indeed can allow for different mechanisms of dependence across graphs. Here, however, we are interested in the simple case in which the graphs are all on the same vertex set, and latent positions for the graphs have the same distribution; given the latent position matrix, the m graphs are generated independently of one another from the same latent positions.*

Given a set of m adjacency matrices distributed as

$$(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(m)}, \mathbf{X}) \sim \text{ciJRDPG}(F, n, m)$$

for distribution F on $\mathbb{R}^d$, a natural inference task is to recover the n latent positions $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n \in \mathbb{R}^d$ shared by the vertices of the m graphs. To estimate the underlying latent postions from these m graphs, Tang et al. (2016) provides justification for the estimate $\bar{\mathbf{X}} = \text{ASE}(\bar{\mathbf{A}}, d)$, where $\bar{\mathbf{A}}$ is the sample mean of the adjacency matrices $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(m)}$. However, $\bar{\mathbf{X}}$ is ill-suited to any task that requires comparing latent positions across the m graphs, since the $\bar{\mathbf{X}}$ estimate collapses the m graphs into a single set of n latent positions. This motivates the *omnibus embedding*, which still yields a single spectral decomposition, but with a separate d-dimensional representation for each of the m graphs. This makes the omnibus embedding useful for *simultaneous* inference across all m observed graphs.

Definition 40 (Omnibus embedding) *Let $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(m)} \in \mathbb{R}^{n \times n}$ be adjacency matrices of a collection of m undirected graphs. We define the mn-by-mn omnibus matrix of $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(m)}$ by*

$$\mathbf{M} = \begin{bmatrix} \mathbf{A}^{(1)} & \frac{1}{2}(\mathbf{A}^{(1)} + \mathbf{A}^{(2)}) & \ldots & \frac{1}{2}(\mathbf{A}^{(1)} + \mathbf{A}^{(m)}) \\ \frac{1}{2}(\mathbf{A}^{(2)} + \mathbf{A}_2^{(1)}) & \mathbf{A}^{(2)} & \ldots & \frac{1}{2}(\mathbf{A}^{(2)} + \mathbf{A}^{(m)}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{2}(\mathbf{A}^{(m)} + \mathbf{A}^{(1)}) & \frac{1}{2}(\mathbf{A}^{(m)} + \mathbf{A}^{(2)}) & \ldots & \mathbf{A}^{(m)} \end{bmatrix}, \tag{37}$$

and the d-dimensional omnibus embedding *of $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(m)}$ is the adjacency spectral embedding of $\mathbf{M}$:*

$$\text{OMNI}(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(m)}, d) = \text{ASE}(\mathbf{M}, d).$$

where *ASE* is the d-dimensional adjacency spectral embedding. Under the ciJRDPG model, given the latent positions $\mathbf{X}$, the omnibus matrix has expected value

$$\mathbb{E}\mathbf{M} = \tilde{\mathbf{P}} = \mathbf{J}_m \otimes \mathbf{P} = \mathbf{U}_{\tilde{\mathbf{P}}} \mathbf{S}_{\tilde{\mathbf{P}}} \mathbf{U}_{\tilde{\mathbf{P}}}^\top$$

for $\mathbf{U}_{\tilde{\mathbf{P}}} \in \mathbb{R}^{mn \times d}$ having d orthonormal columns and $\mathbf{S}_{\tilde{\mathbf{P}}} \in \mathbb{R}^{d \times d}$ diagonal. Since $\mathbf{M}$ is a reasonable estimate for $\tilde{\mathbf{P}} = \mathbb{E}\mathbf{M}$, the matrix $\hat{\mathbf{Z}} = \text{OMNI}(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(m)}, d)$ is a natural estimate of the mn latent positions collected in the matrix $\mathbf{Z} = [\mathbf{X}^\top \mathbf{X}^\top \ldots \mathbf{X}^\top]^\top \in \mathbb{R}^{mn \times d}$. Here again, as in Remark 7, $\hat{\mathbf{Z}}$ only recovers the true latent positions $\mathbf{Z}$ up to an orthogonal rotation. The matrix

$$\mathbf{Z}^* = \begin{bmatrix} \mathbf{X}^* \\ \mathbf{X}^* \\ \vdots \\ \mathbf{X}^* \end{bmatrix} = \mathbf{U}_{\tilde{\mathbf{P}}} \mathbf{S}_{\tilde{\mathbf{P}}}^{1/2} \in \mathbb{R}^{mn \times d}, \tag{38}$$

provides a reasonable canonical choice of latent positions, so that $\mathbf{Z} = \mathbf{Z}^* \mathbf{W}$ for some suitably-chosen orthogonal matrix $\mathbf{W} \in \mathbb{R}^{d \times d}$; again, just as for a single random dot product graph, spectral embedding of the omnibus matrix is a consistent estimator for the latent positions (up to rotation).

Below, we state precise results on consistency and asymptotic normality of the embedding of the omnibus matrix $\mathbf{M}$. The proofs are similar to, but somewhat more involved than, the aforementioned analogues for the adjacency spectral embedding for one graph. We also demonstrate from simulations that the omnibus embedding can be successfully leveraged for subsequent inference, specifically two-sample testing.

First, Lemma 41 shows that the omnibus embedding provides uniformly consistent estimates of the true latent positions, up to an orthogonal transformation, roughly analogous to Lemma 5 in Lyzinski et al. (2014). Lemma 41 shows consistency of the omnibus embedding under the $2 \to \infty$ norm, implying that all mn of the estimated latent positions are near (a rotation of) their corresponding true positions.

Lemma 41 *With $\tilde{\mathbf{P}}$, $\mathbf{M}$, $\mathbf{U}_{\mathbf{M}}$, and $\mathbf{U}_{\tilde{\mathbf{P}}}$ defined as above, there exists an orthogonal matrix $\tilde{\mathbf{W}} \in \mathbb{R}^{d \times d}$ such that with high probability,*

$$\|\mathbf{U}_{\mathbf{M}} \mathbf{S}_{\mathbf{M}}^{1/2} - \mathbf{U}_{\tilde{\mathbf{P}}} \mathbf{S}_{\tilde{\mathbf{P}}}^{1/2} \tilde{\mathbf{W}}\|_{2 \to \infty} \leq \frac{C m^{1/2} \log mn}{\sqrt{n}}. \tag{39}$$

As with the adjacency spectral embedding, we once again can show the asymptotic normality of the individual rows of the omnibus embedding. Note that the covariance matrix does change with m, and for m large, this results in a nontrivial variance reduction.

Theorem 42 *Let $(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(m)}, \mathbf{X}) \sim \text{ciJRDPG}(F, n, m)$ for some d-dimensional inner product distribution F and let $\mathbf{M}$ denote the omnibus matrix as in (37). Let $\mathbf{Z} = \mathbf{Z}^* \mathbf{W}$ with $\mathbf{Z}^*$ as defined in Equation (38), with estimate $\hat{\mathbf{Z}} = \text{OMNI}(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(m)}, d)$. Let $h = m(s-1) + i$ for $i \in [n], s \in [m]$, so that $\hat{\mathbf{Z}}_h$ denotes the estimated latent position of the i-th vertex in the s-th graph $\mathbf{A}^{(s)}$. That is, $\hat{\mathbf{Z}}_h$ is the column vector formed by transposing the h-th row of the matrix $\hat{\mathbf{Z}} = \mathbf{U}_{\mathbf{M}} \mathbf{S}_{\mathbf{M}}^{1/2} = \text{OMNI}(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(m)}, d)$. Let $\Phi(\mathbf{x}, \mathbf{\Sigma})$ denote the cumulative distribution function of a (multivariate) Gaussian with mean zero and covariance matrix $\mathbf{\Sigma}$, evaluated at $\mathbf{x} \in \mathbb{R}^d$. There exists a sequence of orthogonal d-by-d matrices $(\tilde{\mathbf{W}}_n)_{n=1}^\infty$ such that for all $\mathbf{x} \in \mathbb{R}^d$,*

$$\lim_{n \to \infty} \Pr\left[n^{1/2} \left(\hat{\mathbf{Z}} \tilde{\mathbf{W}}_n - \mathbf{Z} \right)_h \leq \mathbf{x} \right] = \int_{\text{supp } F} \Phi\left(\mathbf{x}, \mathbf{\Sigma}(\mathbf{y})\right) dF(\mathbf{y}),$$

where $\mathbf{\Sigma}(\mathbf{y}) = (m+3) \mathbf{\Delta}^{-1} \tilde{\mathbf{\Sigma}}(\mathbf{y}) \mathbf{\Delta}^{-1} / (4m)$, $\mathbf{\Delta} = \mathbb{E}[\mathbf{X}_1 \mathbf{X}_1^\top]$ and

$$\tilde{\mathbf{\Sigma}}(\mathbf{y}) = \mathbb{E}\left[(\mathbf{y}^\top \mathbf{X}_1 - (\mathbf{y}^\top \mathbf{X}_1)^2) \mathbf{X}_1 \mathbf{X}_1^\top \right].$$

Next, we summarize from Levin et al. (2017) experiments on synthetic data exploring the efficacy of the omnibus embedding described above. If we merely wish to estimate the latent positions $\mathbf{X}$ of a set of m graphs $(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(m)}, \mathbf{X}) \sim \text{ciJRDPG}(F, n, m)$, the estimate $\bar{\mathbf{X}} = \text{ASE}(\sum_{i=1}^m \mathbf{A}^{(i)}/m, d)$, the embedding of the sample mean of the adjacency matrices performs well asymptotically (see Tang et al., 2016). Indeed, all else equal, the embedding $\bar{\mathbf{X}}$ is preferable to the omnibus embedding if only because it requires an eigendecomposition of an n-by-n matrix rather than the much larger mn-by-mn omnibus matrix.

Of course, the omnibus embedding can still be used to to estimate the latent positions, potentially at the cost of increased variance. Figure 5 compares the mean-squared error of various techniques for estimating the latent positions for a random dot product graph. The figure plots the (empirical) mean squared error in recovering the latent positions of a 3-dimensional ciJRDPG as a function of the number of vertices n. Each point in the plot is the empirical mean of 50 independent trials; in each trial, the latent positions are

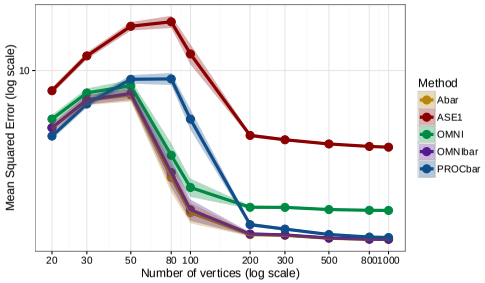


Figure 5: Mean squared error (MSE) in recovery of latent positions (up to rotation) in a 2-graph joint RDPG model as a function of the number of vertices. Performance of ASE applied to a single graph (red), ASE embedding of the mean graph (gold), the Procrustes-based pairwise embedding (blue), the omnibus embedding (green) and the mean omnibus embedding (purple). Each point is the mean of 50 trials; error bars indicate $\pm 2(SE)$. Mean omnibus embedding (OMNIbar) is competitive with $\mathrm{ASE}(\bar{\mathbf{A}}, d)$; Procrustes alignment estimation is notably inferior to the other two-graph techniques for graphs of size between 80 and 200 vertices (note that the gap appears to persist at larger graph sizes, though it shrinks). Figure duplicated from Levin et al. (2017).

drawn i.i.d. from a Dirichlet with parameter $[1, 1, 1]^\top \in \mathbb{R}^3$. Once the latent positions are so obtained, we independently generate two random dot product graphs, $\mathbf{A}^{(1)}, \mathbf{A}^{(2)} \in \mathbb{R}^{n \times n}$ with these latent positions. The figure is interpreted as follows:

1. **ASE1 (red)**: we embed only one of the two observed graphs, and use only the ASE of that graph to estimate the latent positions in $\mathbf{X}$, ignoring entirely the information present in $\mathbf{A}^{(2)}$. This condition serves as a baseline for how much additional information is provided by the second graph $\mathbf{A}^{(2)}$.

2. **Abar (gold)**: we embed the average of the two graphs, $\bar{\mathbf{A}} = (\mathbf{A}^{(1)} + \mathbf{A}^{(2)})/2$ as $\hat{\mathbf{X}} = \mathrm{ASE}(\bar{\mathbf{A}}, 3)$.

3. **OMNI (green)**: We apply the omnibus embedding to obtain $\hat{\mathbf{Z}} = \mathrm{ASE}(\mathbf{M}, 3)$, where $\mathbf{M}$ is as in Equation (37). We then use only the first n rows of $\hat{\mathbf{Z}} \in \mathbb{R}^{2n \times d}$ as our estimate of $\mathbf{X}$. This embedding incorporates information available in both graphs $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$, but does not weight them equally, since the first rows of $\hat{\mathbf{Z}}$ are based primarily on the information contained in $\mathbf{A}^{(1)}$.

4. **OMNIbar (purple)**: We again apply the omnibus embedding to obtain estimated latent positions $\hat{\mathbf{Z}} = \mathrm{ASE}(\mathbf{M}, 3)$, but this time we use all available information by averaging the first n rows and the second n rows of $\hat{\mathbf{Z}}$.

5. **PROCbar (blue)**: We separately embed the graphs $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$, perform a Procrustes alignment between the two embeddings, and average the aligned embeddings to obtain our final estimate of the latent positions.

First, let us note that ASE applied to a single graph (red) lags all other methods, as expected, since this discards all information from the second graph. For very small graphs, the dearth of signal is such that no method will recover the latent positions accurately.

Crucially, however, we see that the OMNIbar estimate (purple) performs nearly identically to the Abar estimate (gold), the natural choice among spectral methods for the estimation latent positions The Procrustes estimate (in blue) provides a two-graph analogue of ASE (red): it combines two ASE estimates via Procrustes alignment, but does not enforce an *a priori* alignment of the estimated latent positions. As predicted by the results in Lyzinski et al. (2014) and Tang et al. (2017a), the Procrustes estimate is competitive with the Abar (gold) estimate for suitably large graphs. The OMNI estimate (in green) serves, in a sense, as an intermediate, since it uses information available from both graphs, but in contrast to Procrustes (blue), OMNIbar (purple) and Abar (gold), it does not make complete use of the information available in the second graph. For this reason, it is noteworthy that the OMNI estimate outperforms the Procrustes estimate for graphs of 80-100 vertices. That is, for certain graph sizes, the omnibus estimate appears to more optimally leverage the information in both graphs than the Procrustes estimate does, despite the fact that the information in the second graph has comparatively little influence on the OMNI embedding.

The omnibus embedding can also be applied to testing the semiparametric hypothesis that two observed graphs are drawn from the same underlying latent positions. Consider a collection of latent positions $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n, \mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_n \in \mathbb{R}^d$. Let the graph G_1 with adjacency matrix $\mathbf{A}^{(1)}$ have edges distributed independently as $\mathbf{A}^{(1)}_{ij} \sim \mathrm{Bernoulli}(\mathbf{X}_i^\top \mathbf{X}_j)$. Similarly, let G_2 have adjacency matrix $\mathbf{A}^{(2)}$ with edges distributed independently as $\mathbf{A}^{(2)}_{ij} \sim \mathrm{Bernoulli}(\mathbf{Y}_i^\top \mathbf{Y}_j)$. Consider the following null hypothesis:

$$H_0 : \mathbf{X} =_W \mathbf{Y} \tag{40}$$

where $\mathbf{X}$ and $\mathbf{Y}$ are the matrices whose rows are the latent positions $\mathbf{X}_i$, $\mathbf{Y}_i$, respectively. The omnibus embedding provides a natural test of this null hypothesis (40) by comparing the first n and last n rows of a lower-dimensional embedding of the omnibus matrix

$$\mathbf{M} = \begin{bmatrix} \mathbf{A}^{(1)} & (\mathbf{A}^{(1)} + \mathbf{A}^{(2)})/2 \\ (\mathbf{A}^{(1)} + \mathbf{A}^{(2)})/2 & \mathbf{A}^{(2)} \end{bmatrix}.$$

Intuitively, when H_0 holds, the distributional result in Theorem 42 holds, and the i-th and $(n+i)$-th rows of $\mathrm{OMNI}(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, d)$ are equidistributed (though they are not independent). On the other hand, when H_0 fails to hold, there exists at least one $i \in [n]$ for which the i-th and $(n+i)$-th rows of $\mathbf{M}$ are *not* identically distributed, and thus the corresponding embeddings are also distributionally distinct. This suggests a test that compares the first n rows of $\mathrm{OMNI}(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, d)$ against the last n rows. That is, our omnibus test statistic simply considers the Frobenius norm of the difference between the top $n \times d$ submatrix of the omnibus embedding and the bottom $n \times d$ submatrix of the omnibus embedding, *with no Procrustes alignment*. Here, we empirically explore the power this test against its Procrustes-based alternative from Tang et al. (2017a).

We draw $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n \in \mathbb{R}^3$ i.i.d. according to a Dirichlet distribution F with parameter $\vec{\alpha} = [1, 1, 1]^\top$. With $\mathbf{X}$ defined as the matrix $\mathbf{X} = [\mathbf{X}_1 \mathbf{X}_2 \ldots \mathbf{X}_n]^\top \in \mathbb{R}^{n \times 3}$, let graph G_1 have

adjacency matrix $\mathbf{A}^{(1)}$, where $\mathbf{A}^{(1)}_{ij} \sim \text{Bernoulli}((\mathbf{X}\mathbf{X}^\top)_{ij})$. We generate a second graph G_2 by first drawing random points $\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_n \overset{\text{i.i.d.}}{\sim} F$. Selecting a set of indices $I \subset [n]$ of size $k < n$ uniformly at random from among all such $\binom{n}{k}$ sets, we let G_2 have latent positions

$$\mathbf{Y}_i = \begin{cases} \mathbf{Z}_i & \text{if } i \in I \\ \mathbf{X}_i & \text{otherwise.} \end{cases}$$

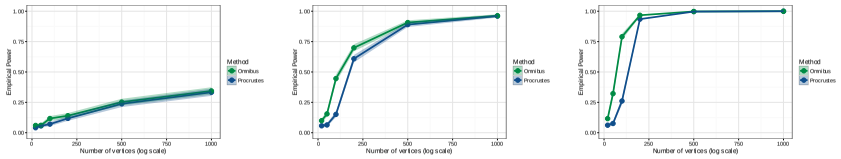
With $\mathbf{Y}$ the matrix $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_n]^\top \in \mathbb{R}^{n\times 3}$, we generate graph G_2 with adjacency matrix $\mathbf{A}^{(2)}$, where $\mathbf{A}^{(2)}_{ij} \sim \text{Bernoulli}((\mathbf{Y}\mathbf{Y}^\top)_{ij})$. We wish to test

$$H_0 : \mathbf{X} =_W \mathbf{Y}. \tag{41}$$

Consider two different tests, one based on a Procrustes alignment of the adjacency spectral embeddings of G_1 and G_2, as in Tang et al. (2017a), and the other based on the omnibus embedding. Both approaches are based on estimates of the latent positions of the two graphs. In both cases we use a test statistic that is some variant of the form $T = \sum_{i=1}^n \|\hat{\mathbf{X}}_i - \hat{\mathbf{Y}}_i\|_F^2$, and accept or reject based on a Monte Carlo estimate of the critical value of T under the null hypothesis, in which $\mathbf{X}_i = \mathbf{Y}_i$ for all $i \in [n]$. In each trial, we use 500 Monte Carlo iterates to estimate the distribution of T.

We note that in the experiments presented here, we assume that the latent positions $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$ of graph G_1 are known for sampling purposes, so that the matrix $\mathbf{P} = \mathbb{E}\mathbf{A}^{(1)}$ is known exactly, rather than estimated from the observed adjacency matrix $\mathbf{A}^{(1)}$. This allows us to sample from the true null distribution. As proved in Lyzinski et al. (2014), the estimated latent positions $\hat{\mathbf{X}}_1 = \text{ASE}(\mathbf{A}^{(1)})$ and $\hat{\mathbf{X}}_2 = \text{ASE}(\mathbf{A}^{(2)})$ recover the true latent positions $\mathbf{X}_1$ and $\mathbf{X}_2$ (up to rotation) to arbitrary accuracy in $(2, \infty)$-norm for suitably large n (Lyzinski et al., 2014). Without using this known matrix $\mathbf{P}$, we would require that our matrices have tens of thousands of vertices before the variance associated with estimating the latent positions would no longer overwhelm the signal present in the few altered latent positions.

Three major factors influence the complexity of testing the null hypothesis in Equation (41): the number of vertices n, the number of changed latent positions $k = |I|$, and the distances $\|\mathbf{X}_i - \mathbf{Y}_i\|_F$ between the latent positions. The three plots in Figure 6 illustrate the first two of these three factors. These three plots show the power of two different approaches to testing the null hypothesis (41) for different sized graphs and for different values of k, the number of altered latent positions. In all three conditions, both methods improve as the number of vertices increases, as expected, especially since we do not require estimation of the underlying expected matrix $\mathbf{P}$ for Monte Carlo estimation of the null distribution of the test statistic. We see that when only one vertex is changed, neither method has power much above 0.25. However, in the case of $k = 5$ and $k = 10$, is it clear that the omnibus-based test achieves higher power than the Procrustes-based test, especially in the range of 30 to 250 vertices. A more detailed examination of the relative impact of these factors in testing is given in Levin et al. (2017). We note that our present power analysis is limited to an empirical study in a few special cases, and a more detailed analysis of theoretical power guarantees for this embedding is ongoing.



(a) (b) (c)

Figure 6: Power of the ASE-based (blue) and omnibus-based (green) tests to detect when the two graphs being testing differ in (a) one, (b) five, and (c) ten of their latent positions. Each point is the proportion of 1000 trials for which the given technique correctly rejected the null hypothesis, and error bars denote two standard errors of this empirical mean in either direction. Figure duplicated from Levin et al. (2017).

In sum, our omnibus embedding provides a natural mechanism for the simultaneous embedding of multiple graphs into a single vector space. This eliminates the need for multiple Procrustes alignments, which were required in previously-explored approaches to multiple-graph testing (see Tang et al., 2017a). In the Procrustes-based approach, each graph is embedded separately, yielding estimates $\mathbf{X}_1$ and $\mathbf{X}_2$, and the test statistic is

$$\min_{\mathbf{W}\in\mathcal{O}_d} \|\hat{\mathbf{X}}_1 - \hat{\mathbf{X}}_2\mathbf{W}\|_F, \tag{42}$$

Under the null hypothesis, a suitable rescaling of this converges as $n \to \infty$. The effect of this Procrustes alignment on subsequent inference is ill-understood; it has the potential to introduce variance, and our simulations results suggest that it negatively impacts performance in both estimation and testing settings. Furthermore, when the matrix $\mathbf{P} = \mathbf{X}\mathbf{X}^\top$ does not have distinct eigenvalues (and is thus not uniquely diagonalizable), this Procrustes step is unavoidable, since the difference $\|\hat{\mathbf{X}}_1 - \hat{\mathbf{X}}_2\|_F$ need not converge at all.

In contrast, our omnibus embedding builds an alignment of the graphs into its very structure. To see this, consider, for simplicity, the $m = 2$ case. Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ be the matrix whose rows are the latent positions of both graphs G_1 and G_2, and let $\mathbf{M} \in \mathbb{R}^{2n\times 2n}$ be their omnibus matrix. Then

$$\mathbb{E}\mathbf{M} = \tilde{\mathbf{P}} = \begin{bmatrix} \mathbf{P} & \mathbf{P} \\ \mathbf{P} & \mathbf{P} \end{bmatrix} = \begin{bmatrix} \mathbf{X} \\ \mathbf{X} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{X} \end{bmatrix}^\top.$$

Suppose now that we wish to factorize $\tilde{\mathbf{P}}$ as

$$\tilde{\mathbf{P}} = \begin{bmatrix} \mathbf{X} \\ \mathbf{X}\mathbf{W}^* \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{X}\mathbf{W}^* \end{bmatrix}^\top = \begin{bmatrix} \mathbf{P} & \mathbf{X}(\mathbf{W}^*)^\top\mathbf{X}^\top \\ \mathbf{X}\mathbf{W}^*\mathbf{X}^\top & \mathbf{P} \end{bmatrix}.$$

That is, we want to consider graphs G_1 and G_2 as being generated from the same latent positions, but in one case, say, under a *different* rotation. This possibility necessitates the Procrustes alignment in the case of separately-embedded graphs. In the case of the omnibus matrix, the structure of the $\tilde{\mathbf{P}}$ matrix implies that $\mathbf{W}^* = \mathbf{I}_d$. In contrast to the Procrustes alignment, the omnibus matrix incorporates an alignment *a priori*. Simulations show

that the omnibus embedding outperforms the Procrustes-based test for equality of latent positions, especially in the case of moderately-sized graphs.

To further illustrate the utility of this omnibus embedding, consider the case of testing whether three different random dot product graphs have the same generating latent positions. The omnibus embedding gives us a *single* canonical representation of all three graphs: Let $\hat{\mathbf{X}}_1^O$, $\hat{\mathbf{X}}_2^O$, and $\hat{\mathbf{X}}_3^O$ be the estimates for the three latent position matrices generated from the omnibus embedding. To test whether any two of these random graphs have the same generating latent positions, we merely have to compare the Frobenius norms of their differences, as opposed to computing three separate Procrustes alignments. In the latter case, in effect, we do not have a canonical choice of coordinates in which to compare our graphs simultaneously.

5.4 Nonparametric graph estimation and testing

The semiparametric and omnibus testing procedures we describe both focus on the estimation of the latent positions themselves. But a very natural concern, for a random dot product graph with distribution F, is the estimation of the *distribution* F. We next address how the adjacency spectral embedding, judiciously integrated with kernel density estimation, can be used for nonparametric estimation and testing in random graphs.

Throughout our discussion on nonparametric estimation, we shall always assume that the distributions of the latent positions satisfy the following distinct eigenvalues assumption. The assumption implies that the estimates of the latent position obtained by the adjacency spectral embedding will, in the limit, be uniquely determined.

Assumption 3 *The distribution F for the latent positions $X_1, X_2, \ldots, \sim F$ is such that the second moment matrix $\mathbb{E}[X_1 X_1^\top]$ has d distinct eigenvalues and d is known.*

We realize that Assumption 3 is restrictive – in particular, it is not satisfied by the stochastic block model with $K > 2$ blocks of equal size and edge probabilities p within communities and q between communities – it is a necessary technical condition for us to obtain the limiting results of Theorem 3. The motivation behind this assumption is as follows: the matrix $\mathbb{E}[X_1 X_1^\top]$ is of rank d with d known so that given a graph $\mathbf{A} \sim \mathrm{RDPG}(F)$, one can construct the adjacency spectral embedding of $\mathbf{A}$ into the "right" Euclidean space. The requirement that $\mathbb{E}[X_1 X_1^\top]$ has d distinct eigenvalues is—once again—due to the intrinsic property of non-identifiability of random dot product graphs. As always, for any random dot product graph $\mathbf{A}$, the latent position $\mathbf{X}$ associated with $\mathbf{A}$ can only be estimated up to some true but unknown orthogonal transformation. Because we are concerned with two-sample hypothesis testing, we must guard against the scenario in which we have two graphs $\mathbf{A}$ and $\mathbf{B}$ with latent positions $\mathbf{X} = \{X_i\}_{i=1}^n \overset{\text{i.i.d}}{\sim} F$ and $\mathbf{Y} = \{Y_k\}_{k=1}^m \overset{\text{i.i.d}}{\sim} F$ but whose estimates $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$ lie in different, incommensurate subspaces of $\mathbb{R}^d$. That is to say, the estimates $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$ satisfy $\hat{\mathbf{X}} \approx \mathbf{X}\mathbf{W}_1$ and $\hat{\mathbf{Y}} \approx \mathbf{Y}\mathbf{W}_2$, but $\|\mathbf{W}_1 - \mathbf{W}_2\|_F$ does not converge to 0 as $n, m \to \infty$. See also Fishkind et al. (2015b) for exposition of a related so-called "incommensurability phenomenon."

Our main point of departure for this subsection compared to Section 5.2 is the assumption that, given a sequence of pairs of random dot product graphs with adjacency matrices $\mathbf{A}_n$ and $\mathbf{B}_n$, the rows of the latent positions $\mathbf{X}_n$ and $\mathbf{Y}_n$ are independent samples from some fixed distributions F and G, respectively. The corresponding tests are therefore tests of equality between F and G. More formally, we consider the following two-sample nonparametric testing problems for random dot product graphs. Let F and G be two inner product distributions. Given $\mathbf{A} \sim \mathrm{RDPG}(F)$ and $\mathbf{B} \sim \mathrm{RDPG}(G)$, we consider the tests:

1. *(Equality, up to orthogonal transformation)*

$$H_0 \colon F \doteq G \quad \text{against} \quad H_A \colon F \not\doteq G,$$

   where $F \doteq G$ denotes that there exists a unitary operator U on $\mathbb{R}^d$ such that $F = G \circ U$ and $F \not\doteq G$ denotes that $F \neq G \circ U$ for any unitary operator U on $\mathbb{R}^d$.

2. *(Equality, up to scaling)*

$$H_0 \colon F \doteq G \circ c \quad \text{for some } c > 0 \quad \text{against} \quad H_A \colon F \not\doteq G \circ c \quad \text{for any } c > 0,$$

   where $Y \sim F \circ c$ if $cY \sim F$.

3. *(Equality, up to projection)*

$$H_0 \colon F \circ \pi^{-1} \doteq G \circ \pi^{-1} \quad \text{against} \quad H_A \colon F \circ \pi^{-1} \not\doteq G \circ \pi^{-1},$$

   where π is the projection $x \mapsto x/\|x\|$; hence $Y \sim F \circ \pi^{-1}$ if $\pi^{-1}(Y) \sim F$.

We note that the above null hypotheses are nested; $F \doteq G$ implies $F \doteq G \circ c$ for $c = 1$ while $F \doteq G \circ c$ for some $c > 0$ implies $F \circ \pi^{-1} \doteq G \circ \pi^{-1}$.

We shall address the above hypothesis testing problem by combining the framework of adjacency spectral embedding and the kernel-based hypothesis testing framework of Gretton et al. (2012). The testing procedure in Gretton et al. (2012) is based on the following notion of the maximum mean discrepancy between distributions. Let Ω be a compact metric space and $\kappa \colon \Omega \times \Omega \mapsto \mathbb{R}$ a continuous, symmetric, and positive definite kernel on Ω. Denote by $\mathcal{H}$ the reproducing kernel Hilbert space associated with κ. Now let F be a probability distribution on Ω. Under mild conditions on κ, the map $\mu[F]$ defined by

$$\mu[F] := \int_\Omega \kappa(\omega, \cdot)\, \mathrm{d}F(\omega)$$

belongs to $\mathcal{H}$. Now, for given probability distributions F and G on Ω, the *maximum mean discrepancy* between F and G with respect to $\mathcal{H}$ is the measure

$$\mathrm{MMD}(F, G; \mathcal{H}) := \|\mu[F] - \mu[G]\|_{\mathcal{H}}.$$

We now summarize some important properties of the maximum mean discrepancy from Gretton et al. (2012).

Theorem 43 *Let $\kappa\colon \mathcal{X}\times\mathcal{X}\mapsto\mathbb{R}$ be a positive definite kernel and denote by $\mathcal{H}$ the reproducing kernel Hilbert space associated with κ. Let F and G be probability distributions on Ω; X and X' independent random variables with distribution F, Y and Y' independent random variables with distribution G, and X is independent of Y. Then*

$$\begin{aligned}\|\mu[F]-\mu[G]\|_{\mathcal{H}}^2 &= \sup_{h\in\mathcal{H}:\|h\|_{\mathcal{H}}\le 1}|\mathbb{E}_F[h]-\mathbb{E}_G[h]|^2\\ &= \mathbb{E}[\kappa(X,X')]-2\mathbb{E}[\kappa(X,Y)]+\mathbb{E}[\kappa(Y,Y')].\end{aligned} \tag{43}$$

Given $\mathbf{X}=\{X_i\}_{i=1}^n$ and $\mathbf{Y}=\{Y_k\}_{k=1}^m$ with $\{X_i\}\overset{\text{i.i.d}}{\sim}F$ and $\{Y_i\}\overset{\text{i.i.d}}{\sim}G$, the quantity $U_{n,m}(\mathbf{X},\mathbf{Y})$ defined by

$$\begin{aligned}U_{n,m}(\mathbf{X},\mathbf{Y}) = &\frac{1}{n(n-1)}\sum_{j\neq i}\kappa(X_i,X_j)-\frac{2}{mn}\sum_{i=1}^n\sum_{k=1}^m\kappa(X_i,Y_k)\\ &+\frac{1}{m(m-1)}\sum_{l\neq k}\kappa(Y_k,Y_l)\end{aligned} \tag{44}$$

is an unbiased consistent estimate of $\|\mu[F]-\mu[G]\|_{\mathcal{H}}^2$. Denote by $\tilde{\kappa}$ the kernel

$$\tilde{\kappa}(x,y)=\kappa(x,y)-\mathbb{E}_z\kappa(x,z)-\mathbb{E}_{z'}\kappa(z',y)+\mathbb{E}_{z,z'}\kappa(z,z')$$

where the expectation is taken with respect to $z,z'\sim F$. Suppose that $\frac{m}{m+n}\to\rho\in(0,1)$ as $m,n\to\infty$. Then under the null hypothesis of $F=G$,

$$(m+n)U_{n,m}(\mathbf{X},\mathbf{Y})\xrightarrow{d}\frac{1}{\rho(1-\rho)}\sum_{l=1}^{\infty}\lambda_l(\chi_{1l}^2-1) \tag{45}$$

where $\{\chi_{1l}^2\}_{l=1}^{\infty}$ is a sequence of independent χ^2 random variables with one degree of freedom, and $\{\lambda_l\}$ are the eigenvalues of the integral operator $\mathcal{I}_{F,\tilde{\kappa}}:\mathcal{H}\mapsto\mathcal{H}$ defined as

$$I_{F,\tilde{\kappa}}(\phi)(x)=\int_{\Omega}\phi(y)\tilde{\kappa}(x,y)dF(y).$$

Finally, if κ is a universal or characteristic kernel (Sriperumbudur et al., 2011; Steinwart, 2001), then μ is an injective map: that is, $\mu[F]=\mu[G]$ if and only if $F=G$.

Remark 44 *A kernel $\kappa\colon\mathcal{X}\times\mathcal{X}\mapsto\mathbb{R}$ is universal if κ is a continuous function of both its arguments and if the reproducing kernel Hilbert space $\mathcal{H}$ induced by κ is dense in the space of continuous functions on $\mathcal{X}$ with respect to the supremum norm. Let $\mathcal{M}$ be a family of Borel probability measures on $\mathcal{X}$. A kernel κ is characteristic for $\mathcal{M}$ if the map $\mu\in\mathcal{M}\mapsto\int\kappa(\cdot,z)\mu(dz)$ is injective. If κ is universal, then κ is characteristic for any $\mathcal{M}$ (Sriperumbudur et al., 2011). As an example, let $\mathcal{X}$ be a finite dimensional Euclidean space and define, for any $q\in(0,2)$, $k_q(x,y)=\frac{1}{2}(\|x\|^q+\|y\|^q-\|x-y\|^q)$. The kernels k_q are then characteristic for the collection of probability distributions with finite second moments (Lyons, 2013; Sejdinovic et al., 2013). In addition, by Eq. (43), the maximum mean discrepancy with reproducing kernel k_q can be written as*

$$\mathrm{MMD}^2(F,Q;k_q)=2\mathbb{E}\|X-Y\|^q-\mathbb{E}\|X-X'\|^q-\mathbb{E}\|Y-Y'\|^q.$$

where X,X' are independent with distribution F, Y,Y' are independent with distribution G, and X,Y are independent. This coincides with the notion of the energy distances of Székely and Rizzo (2013), or, when $q=1$, a special case of the one-dimensional interpoint comparisons of Maa et al. (1996). Finally, we note that $(m+n)U_{n,m}(\mathbf{X},\mathbf{Y})$ under the null hypothesis of $F=G$ in Theorem 43 depends on the $\{\lambda_l\}$ which, in turn, depend on the distribution F; thus the limiting distribution is not distribution-free. Moreover the eigenvalues $\{\lambda_l\}$ can, at best, be estimated; for finite n, they cannot be explicitly determined when F is unknown. In practice, generally the critical values are estimated through a bootstrap resampling or permutation test.

We focus on the nonparametric two-sample hypothesis test of $\mathbb{H}_0\colon F\doteq G$ against $\mathbb{H}_A\colon F\not\doteq G$. For our purposes, we shall assume henceforth that κ is a twice continuously-differentiable radial kernel and that κ is also universal. To justify this assumption on our kernel, we point out that in Theorem 45 below, we show that the test statistic $U_{n,m}(\hat{\mathbf{X}},\hat{\mathbf{Y}})$ based on the estimated latent positions converges to the corresponding statistic $U_{n,m}(\mathbf{X},\mathbf{Y})$ for the true but unknown latent positions. Due to the non-identifiability of the random dot product graph under unitary transformation, *any* estimate of the latent positions is close, only up to an appropriate orthogonal transformation, to $\mathbf{X}$ and $\mathbf{Y}$. For a radial kernel κ, this implies the approximations $\kappa(\hat{X}_i,\hat{X}_j)\approx\kappa(X_i,X_j)$, $\kappa(\hat{Y}_k,\hat{Y}_l)\approx\kappa(Y_k,Y_l)$ and the convergence of $U_{n,m}(\hat{\mathbf{X}},\hat{\mathbf{Y}})$ to $U_{n,m}(\mathbf{X},\mathbf{Y})$. If κ is not a radial kernel, the above approximations might not hold and $U_{n,m}(\hat{\mathbf{X}},\hat{\mathbf{Y}})$ need not converge to $U_{n,m}(\mathbf{X},\mathbf{Y})$. The assumption that κ is twice continuously-differentiable is technical. Finally, the assumption that κ is universal allows the test procedure to be consistent against a large class of alternatives.

Theorem 45 *Let $(\mathbf{X},\mathbf{A})\sim\mathrm{RDPG}(F)$ and $(\mathbf{Y},\mathbf{B})\sim\mathrm{RDPG}(G)$ be independent random dot product graphs with latent position distributions F and G. Furthermore, suppose that both F and G satisfies the distinct eigenvalues condition in Assumption 3. Consider the hypothesis test*

$$H_0\colon F\doteq G\quad\textit{against}\quad H_A\colon F\not\doteq G.$$

Denote by $\hat{\mathbf{X}}=\{\hat{X}_1,\dots,\hat{X}_n\}$ and $\hat{\mathbf{Y}}=\{\hat{Y}_1,\dots,\hat{Y}_m\}$ the adjacency spectral embedding of $\mathbf{A}$ and $\mathbf{B}$, respectively. Let κ be a twice continuously-differentiable radial kernel and $U_{n,m}(\hat{\mathbf{X}},\hat{\mathbf{Y}})$ be defined as

$$U_{n,m}(\hat{\mathbf{X}},\hat{\mathbf{Y}})=\frac{1}{n(n-1)}\sum_{j\neq i}\kappa(\hat{X}_i,\hat{X}_j)-\frac{2}{mn}\sum_{i=1}^n\sum_{k=1}^m\kappa(\hat{X}_i,\hat{Y}_k)+\frac{1}{m(m-1)}\sum_{l\neq k}\kappa(\hat{Y}_k,\hat{Y}_l).$$

Let $\mathbf{W}_1$ and $\mathbf{W}_2$ be $d\times d$ orthogonal matrices in the eigendecomposition $\mathbf{W}_1\mathbf{S}_1\mathbf{W}_1^\top=\mathbf{X}^\top\mathbf{X}$, $\mathbf{W}_2\mathbf{S}_2\mathbf{W}_2=\mathbf{Y}^\top\mathbf{Y}$, respectively. Suppose that $m,n\to\infty$ and $m/(m+n)\to\rho\in(0,1)$. Then under the null hypothesis of $F\doteq G$, the sequence of matrices $\mathbf{W}_{n,m}=\mathbf{W}_2\mathbf{W}_1^\top$ satisfies

$$(m+n)(U_{n,m}(\hat{\mathbf{X}},\hat{\mathbf{Y}})-U_{n,m}(\mathbf{X},\mathbf{Y}\mathbf{W}_{n,m}))\xrightarrow{\text{a.s.}}0. \tag{46}$$

Under the alternative hypothesis of $F\not\doteq G$, the sequence of matrices $\mathbf{W}_{n,m}$ satisfies

$$\frac{m+n}{\log^2(m+n)}(U_{n,m}(\hat{\mathbf{X}},\hat{\mathbf{Y}})-U_{n,m}(\mathbf{X},\mathbf{Y}\mathbf{W}_{n,m}))\xrightarrow{\text{a.s.}}0. \tag{47}$$

Eq.(46) and Eq.(47) state that the test statistic $U_{n,m}(\hat{\mathbf{X}}, \hat{\mathbf{Y}})$ using the *estimated* latent positions is almost identical to the statistic $U_{n,m}(\mathbf{X}, \mathbf{Y}\mathbf{W}_{n,m})$ using the true latent positions, under both the null and alternative hypothesis. If we assume that κ is a universal kernel, then $U_{n,m}(\mathbf{X}, \mathbf{Y}\mathbf{W}_{n,m})$ converges to 0 under the null and converges to a positive number under the alternative. The test statistic $U_{n,m}(\hat{\mathbf{X}}, \hat{\mathbf{Y}})$ therefore yields a test procedure that is consistent against any alternative, provided that both F and G satisfy Assumption 3, namely that the second moment matrices have d distinct eigenvalues.

We next consider the case of testing the hypothesis that the distributions F and G are equal up to scaling or equal up to projection. For the test of equality up to scaling,

$$H_0 \colon F \doteq G \circ c \quad \text{for some } c > 0 \quad \text{against} \quad H_A \colon F \not\doteq G \circ c \quad \text{for any } c > 0,$$

where $Y \sim F \circ c$ if $cY \sim F$, we modified Theorem 45 by first scaling the adjacency spectral embeddings by the norm of the empirical means before computing the kernel test statistic. In particular, let

$$\hat{s}_X = n^{-1/2}\|\hat{\mathbf{X}}\|_F, \quad \hat{s}_Y = m^{-1/2}\|\hat{\mathbf{Y}}\|_F, \quad s_X = n^{-1/2}\|\mathbf{X}\|_F, \quad s_Y = m^{-1/2}\|\mathbf{Y}\|_F,$$

then the conclusions of Theorem 45 hold when we replace $U_{n,m}(\hat{\mathbf{X}}, \hat{\mathbf{Y}})$ and $U_{n,m}(\mathbf{X}, \mathbf{Y}\mathbf{W}_{n,m})$ with $U_{n,m}(\hat{\mathbf{X}}/\hat{s}_X, \hat{\mathbf{Y}}/\hat{s}_Y)$ and $U_{n,m}(\mathbf{X}/s_X, \mathbf{Y}\mathbf{W}_{n,m}/s_Y)$, respectively.

For the test of equality up to projection,

$$H_0 \colon F \circ \pi^{-1} \doteq G \circ \pi^{-1} \quad \text{against} \quad H_A \colon F \circ \pi^{-1} \not\doteq G \circ \pi^{-1},$$

where π is the projection $x \mapsto x/\|x\|$ that maps x onto the unit sphere in $\mathbb{R}^d$, the conclusions of Theorem 45 hold when we replace $U_{n,m}(\hat{\mathbf{X}}, \hat{\mathbf{Y}})$ and $U_{n,m}(\mathbf{X}, \mathbf{Y}\mathbf{W}_{n,m})$ with $U_{n,m}(\pi(\hat{\mathbf{X}}), \pi(\hat{\mathbf{Y}}))$ and $U_{n,m}(\pi(\mathbf{X}), \pi(\mathbf{Y})\mathbf{W}_{n,m})$, respectively, provided that that 0 is not an atom of either F or G (and hence that $F(\{0\}) = G(\{0\}) = 0$). The assumption that 0 is not an atom is necessary, because otherwise the problem is possibly ill-posed: specifically, $\pi(0)$ is undefined. To contextualize the test of equality up to projection, consider the very specific case of the degree-corrected stochastic blockmodel of Karrer and Newman (2011). As mentioned earlier, the degree-corrected stochastic blockmodel can be regarded as a random dot product graph whose latent position X_v for an arbitrary vertex v is of the form $X_v = \theta_v \nu_v$ where ν_v is sampled from a mixture of point masses and θ_v (the degree-correction factor) is sampled from a distribution on $(0, 1]$. Thus, given two degree-corrected stochastic blockmodel graphs, equality up to projection tests whether the underlying mixtures of point masses (that is, the distribution of the ν_v) are the same modulo the distribution of the degree-correction factors θ_v.

6. Applications

We begin this section by first presenting an application of the two-sample semiparametric test procedure in Section 5.2, demonstrating how it can be applied to compare data from a collection of neural images.

Algorithm 1 Bootstrapping procedure for the test $\mathbb{H}_0 \colon \mathbf{X} =_W \mathbf{Y}$.

procedure Bootstrap($\mathbf{X}, T, bs$) ▷ Returns the p-value associated with T.
  $d \leftarrow \text{ncol}(\mathbf{X})$; $\mathcal{S}_X \leftarrow \varnothing$ ▷ Set d to be the number of columns of $\mathbf{X}$.
  for $b \leftarrow 1{:}bs$ **do**
    $\mathbf{A}_b \leftarrow \text{RDPG}(\hat{\mathbf{X}})$; $\mathbf{B}_b \leftarrow \text{RDPG}(\hat{\mathbf{X}})$
    $\hat{\mathbf{X}}_b \leftarrow \text{ASE}(\mathbf{A}_b, d)$; $\hat{\mathbf{Y}}_b \leftarrow \text{ASE}(\mathbf{B}_b, d)$
    $T_b \leftarrow \min_{\mathbf{W}} \|\hat{\mathbf{X}}_b - \hat{\mathbf{Y}}_b\mathbf{W}\|_F$; $\mathcal{S}_X \leftarrow \mathcal{S}_X \cup T_b$
  end for
  return $p \leftarrow (|\{s \in \mathcal{S}_X \colon s \geq T\}| + 0.5)/bs$ ▷ Continuity correction.
end procedure

$\hat{\mathbf{X}} \leftarrow \text{ASE}(\mathbf{A}, d)$; $\hat{\mathbf{Y}} \leftarrow \text{ASE}(\mathbf{B}, d)$ ▷ The embedding dimension d is assumed given.
$T \leftarrow \min_{\mathbf{W}} \|\hat{\mathbf{X}} - \hat{\mathbf{Y}}\mathbf{W}\|_F$
$p_X \leftarrow \text{Bootstrap}(\hat{\mathbf{X}}, T, bs)$ ▷ The number of bootstrap samples bs is assumed given.
$p_Y \leftarrow \text{Bootstrap}(\hat{\mathbf{Y}}, T, bs)$
$p = \max\{p_X, p_Y\}$ ▷ Returns the maximum of the two p-values.

6.1 Semiparametric testing for brain scan data

We consider neural imaging graphs obtained from the test-retest diffusion MRI and magnetization-prepared rapid acquisition gradient echo (MPRAGE) data of Landman et al. (2011). The raw data consist of 42 images: namely, one pair of neural images from each of 21 subjects. These images are generated, in part, for the purpose of evaluating scan-rescan reproducibility of the MPRAGE image protocol. Table 5 from Landman et al. (2011) indicates that the variability of MPRAGE is quite small; specifically, the cortical gray matter, cortical white matter, ventricular cerebrospinal fluid, thalamus, putamen, caudate, cerebellar gray matter, cerebellar white matter, and brainstem were identified with mean volume-wise reproducibility of 3.5%, with the largest variability being that of the ventricular cerebrospinal fluid at 11%.

We use the MIGRAINE pipeline of Roncal et al. (2013) to convert these scans into spatially-aligned graphs, in which each vertex corrresponds to a particular voxel in a reference coordinate system to which the image is registered. We first consider a collection of small graphs on seventy vertices that are generated from an atlas of seventy brain regions and the fibers connecting them. Given these graphs, we proceed to investigate the similarities and dissimilarities between the scans. We first embed each graph into $\mathbb{R}^4$. We then test the hypothesis of equality up to rotation between the graphs. Since Theorem 34 is a large-sample result, the rejection region specified therein might be excessively conservative for the graphs on $n = 70$ vertices in our current context. We remedy this issue by using the rejection region and p-values reported by the parametric bootstrapping procedure presented in Algorithm 1.

The pairwise comparisons between the 42 graphs are presented in Figure 7. Figure 7 indicates that, in general, the test procedure fails to reject the null hypothesis when the two graphs are for the same subject. This is consistent with the reproducibility finding of Landman et al. (2011). Furthermore, this outcome is also intuitively plausible; in addition

to failing to reject when two scans are from the same subject, we also frequently *do* reject the null hypothesis when the two graphs are from scans of different subjects. Note that our analysis is purely exploratory; as such, we do not grapple with issues of multiple comparisons here.

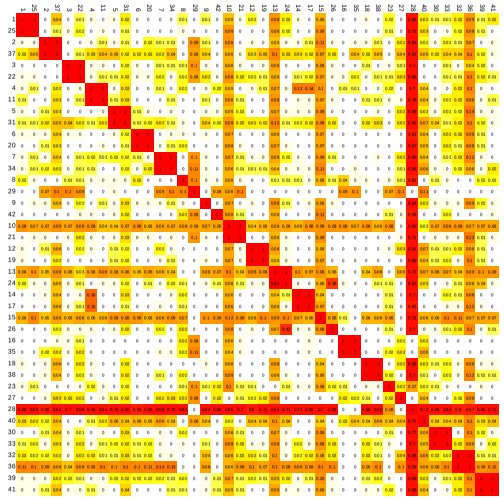


Figure 7: Matrix of p-values (uncorrected) for testing the hypothesis $\mathbb{H}_0: \mathbf{X} =_W \mathbf{Y}$ for the $42 \times 41/2$ pairs of graphs generated from the KKI test-retest dataset Landman et al. (2011). The labels had been arranged so that the pair $(2i-1, 2i)$ correspond to scans from the same subject. The p-values are color coded to vary in intensity from white (p-value of 0) to dark red (p-value of 1). Figure duplicated from Tang et al. (2017a).

Finally, we note that similar results also hold when we consider the large graphs generated from these test-retest data through the MIGRAINE pipeline. In particular, for each magnetic resonance scan, the MIGRAINE pipeline can generates graphs with upto 10^7 vertices and 10^{10} edges with the vertices of all the graphs aligned. Bootstrapping the test statistics for these large graphs present some practical difficulties; one procedure proposed in Tang et al. (2017a) is based on bootstrapping disjoint induced subgraphs of the original graphs using the bootstrapping procedure in Algorithm 1 and combining the resulting p-values using Fisher's combined probability tests (Mosteller and Fisher, 1948).

6.2 Community detection and classification in hierarchical models

In disciplines as diverse as social network analysis and neuroscience, many large graphs are believed to be composed of loosely connected communities and/or smaller graph primitives, whose structure is more amenable to analysis. We emphasize that while the problem of community detection is very well-studied and there are an abundance of community detection algorithms, these algorithms have focused mostly on uncovering the subgraphs. Recently, however, the characterization and further *classification* of these subgraphs into stochastically similar motifs has emerged as an important area of ongoing research The nonparametric two-sample hypothesis testing procedure in Section 5.4 can be used in conjunction with spectral community detection algorithms to yield a robust, scalable, integrated methodology for *community detection* and *community comparison* in graphs (Lyzinski et al., 2017).

The notion of *hierarchical stochastic block model*—namely, a graph consisting of densely connected subcommunities which are themselves stochastic block models, with this structure iterated repeatedly—is precisely formulated in Lyzinski et al. (2017). In that work, a novel angle-based clustering method is introduced, and this clustering method allows us to isolate appropriate subgraphs. We emphasize that the angle-based clustering procedure in Lyzinski et al. (2017) is designed to identify a particular affinity structure within our hierarchical block model graph. Figure 8 illustrates how an angle-based clustering may differ from a k-means clustering.

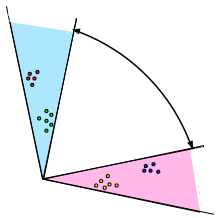


Figure 8: Subgraphs vs. angle-based clustering: Note that if the fraction of points in the pink cone is sufficiently large, K-means clustering (with $K = 2$) will not cluster the vertices in the hierarchical SBM affinity model of Lyzinski et al. (2017) into the appropriate subgraphs. Figure duplicated from Lyzinski et al. (2017).

Our overall community detection algorithm is summarized in Algorithm 2. As an illustrative example of this methodology, we present an analysis of the communities in the Friendster social network. The Friendster social network contains roughly 60 million users and 2 billion connections/edges. In addition, there are roughly 1 million communities at the local scale. Because we expect the social interactions in these communities to inform the function of the different communities, we expect to observe distributional repetition among the graphs associated with these communities.

Algorithm 2 Detecting hierarchical structure for graphs

Input: Adjacency matrix $A \in \{0,1\}^{n\times n}$ for a latent position random graph.
Output: Subgraphs and characterization of their dissimilarity
while Cluster size exceeds threshold **do**
Step 1: Compute the adjacency spectral embedding $\hat{\mathbf{X}}$ of A into $\mathbb{R}^D$;
Step 2: Cluster $\hat{\mathbf{X}}$ to obtain subgraphs $\hat{H}_1, \cdots, \hat{H}_R$ using a novel angle-based clustering procedure given in Lyzinski et al. (2017).
Step 3: For each $i \in [R]$, compute the adjacency spectral embedding for each subgraph $\hat{H}_i$ into $\mathbb{R}^d$, obtaining $\hat{\mathbf{X}}_{\hat{H}_i}$;
Step 4: Compute $\widehat{S} := [U_{\hat{n}_r,\hat{n}_s}(\hat{\mathbf{X}}_{\hat{H}_r}, \hat{\mathbf{X}}_{\hat{H}_s})]$, where U is the test statistic in Theorem 45, producing a pairwise dissimilarity matrix on induced subgraphs;
Step 5: Cluster induced subgraphs into motifs using the dissimilarities given in $\widehat{S}$; for example, use a hierarchical clustering algorithm to cluster the rows of $\widehat{S}$ or the matrix of associated p-values.
Step 6: Recurse on a representative subgraph from each motif (for example, the largest subgraph), embedding into $\mathbb{R}^d$ in Step 1 (not $\mathbb{R}^D$);
end while

Implementing Algorithm 2 on the very large Friendster graph presents several computational challenges and model selection quagmires. To overcome the computational challenge in scalability, we use the specialized SSD-based graph processing engine `FlashGraph` (Zheng et al., 2015), which is designed to analyze graphs with billions of nodes. With `FlashGraph`, we adjacency spectral embed the Friendster adjacency matrix into $\mathbb{R}^{14}$—where $\widehat{D} = 14$ is chosen using universal singular value thresholding on the partial SCREE plot (Chatterjee, 2015). We next cluster the embedded points into $\widehat{R} = 15$ large-scale/coarse-grained clusters ranging in size from 10^6 to 15.6 million vertices (note that to alleviate sparsity concerns, we projected the embedding onto the sphere before clustering); After re-embedding the induced subgraphs associated with these 15 clusters, we use a linear time estimate of the test statistic U to compute $\widehat{S}$, the matrix of estimated pairwise dissimilarities among the subgraphs. See Figure 9 for a heat map depicting $\widehat{S} \in \mathbb{R}^{15\times 15}$. In the heat map, the similarity of the communities is represented on the spectrum between white and red, with white representing highly similar communities and red representing highly dissimilar communities. From the figure, we can see clear repetition in the subgraph distributions; for example, we see a repeated motif including subgraphs $\{\hat{H}_5, \hat{H}_4, \hat{H}_3, \hat{H}_2\}$ and a clear motif including subgraphs $\{\hat{H}_{10}, \hat{H}_{12}, \hat{H}_9\}$.

Formalizing the motif detection step, we employ hierarchical clustering to cluster $\widehat{S}$ into motifs; see Figure 9 for the corresponding hierarchical clustering dendrogram, which suggests that our algorithm does in fact uncover repeated motif structure at the coarse-grained level in the Friendster graph. While it may be difficult to draw meaningful inference from repeated motifs at the scale of hundreds of thousands to millions of vertices, if these motifs are capturing a common HSBM structure within the subgraphs in the motif, then we can employ our algorithm recursively on each motif to tease out further hierarchical structure.

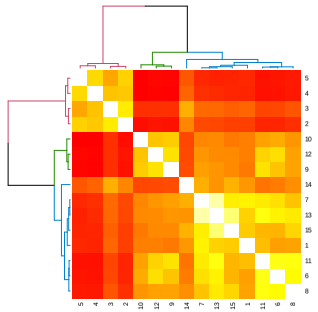


Figure 9: Heat map depiction of the level one Friendster estimated dissimilarity matrix $\widehat{S} \in \mathbb{R}^{15\times 15}$. In the heat map, the similarity of the communities is represented on the spectrum between white and red, with white representing highly similar communities and red representing highly dissimilar communities. In addition, we cluster $\widehat{S}$ using hierarchical clustering and display the associated hierarchical clustering dendrogram. Figure duplicated from Lyzinski et al. (2017).

Exploring this further, we consider three subgraphs $\{\hat{H}_2, \hat{H}_8, \hat{H}_{15}\}$, two of which are in the same motif (8 and 15) and both differing significantly from subgraph 2 according to $\widehat{S}$. We embed these subgraphs into $\mathbb{R}^{26}$ (26 were chosen once again using the universal singular value thresholding of Chatterjee (2015)) perform a Procrustes alignment of the vertex sets of the three subgraphs, cluster each into 4 clusters (4 chosen to optimize silhouette width in k-means clustering), and estimate both the block connection probability matrices,

$$\hat{P}_2 = \begin{bmatrix} 0.000045 & 0.00080 & 0.00056 & 0.00047 \\ 0.00080 & 0.025 & 0.0096 & 0.0072 \\ 0.00057 & 0.0096 & 0.012 & 0.0067 \\ 0.00047 & 0.0072 & 0.0067 & 0.023 \end{bmatrix},$$

$$\hat{P}_8 = \begin{bmatrix} 0.0000022 & 0.000031 & 0.000071 & 0.000087 \\ 0.000031 & 0.0097 & 0.00046 & 0.00020 \\ 0.000071 & 0.00046 & 0.0072 & 0.0030 \\ 0.000087 & 0.00020 & 0.003 & 0.016 \end{bmatrix},$$

$$\hat{P}_{15} = \begin{bmatrix} 0.0000055 & 0.00011 & 0.000081 & 0.000074 \\ 0.00011 & 0.014 & 0.0016 & 0.00031 \\ 0.000081 & 0.0016 & 0.0065 & 0.0022 \\ 0.000074 & 0.00031 & 0.0022 & 0.019 \end{bmatrix},$$

and the block membership probabilities $\hat{\pi}_2$, $\hat{\pi}_8$, $\hat{\pi}_{15}$, for each of the three graphs. We calculate

$$\|\hat{P}_2 - \hat{P}_8\|_F = 0.033; \quad \|\hat{\pi}_2 - \hat{\pi}_8\| = 0.043;$$
$$\|\hat{P}_8 - \hat{P}_{15}\|_F = 0.0058; \quad \|\hat{\pi}_8 - \hat{\pi}_{15}\| = 0.0010;$$
$$\|\hat{P}_2 - \hat{P}_{15}\|_F = 0.027; \quad \|\hat{\pi}_2 - \hat{\pi}_{15}\| = 0.043;$$

which suggests that the repeated structure our algorithm uncovers is *SBM substructure*, thus ensuring that we can proceed to apply our algorithm recursively to the subsequent motifs.

As a final point, we recursively apply Algorithm 2 to the subgraph $\hat{H}_{11}$. We first embed the graph into $\mathbb{R}^{26}$ (again, with 26 chosen via universal singular value thresholding). We then cluster the vertices into $\widehat{R} = 13$ large-scale clusters of sizes ranging from 500K to 2.7M vertices. We then use a linear time estimate of the test statistic T to compute $\widehat{S}$ (see Figure 10), and note that there appear to be clear repeated motifs (for example, subgraphs 8 and 12) among the $\widehat{H}$'s. We run hierarchical clustering to cluster the 13 subgraphs, and note that the associated dendrogram—as shown in Figure 10—shows that our algorithm again uncovered some repeated level-2 structure in the Friendster network. We can, of course, recursively apply our algorithm still further to tease out the motif structure at increasingly fine-grained scale.

Ideally, when recursively running Algorithm 2, we would like to simultaneously embed and cluster all subgraphs in the motif. In addition to potentially reducing embedding variance, being able to efficiently simultaneously embed all the subgraphs in a motif could greatly increase algorithmic scalability in large networks with a very large number of communities at local-scale. In order to do this, we need to understand the nature of the repeated structure within the motifs. This repeated structure can inform an estimation of a motif average (an averaging of the subgraphs within the motif), which can then be embedded into an appropriate Euclidean space in lieu of embedding all of the subgraphs in the motif separately. However, this averaging presents several novel challenges, as these subgraphs may be of very different orders and may be errorfully obtained, which could lead to compounded errors in the averaging step.

6.3 Structure discovery in the *Drosophila* connectome

In this subsection, we address a cutting-edge application of our techniques to neuroscience: structure discovery in the larval Drosophila connectome, comprehensively described in Priebe et al. (2017), and from which significant portions are reproduced here, with permission. This is a first-of-its-kind exploratory data analysis of a newly-available wiring diagram, and although the connectome graph we analyze is directed, weighted, and also of unknown embedding dimension, our statistical techniques can nonetheless be adapted to this setting.

Specifically, we introduce the *latent structure model* (LSM) for network modeling and inference. The LSM is a generalization of the stochastic block model (SBM) in that the latent

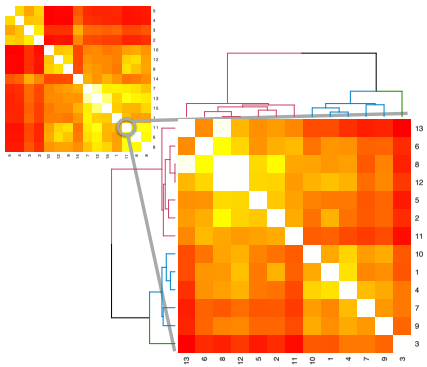


Figure 10: Heat map depiction of the level two Friendster estimated dissimilarity matrix $\widehat{S} \in \mathbb{R}^{13\times 13}$ of $\widehat{H}_{11}$. In the heat map, the similarity of the communities is represented on the spectrum between white and red, with white representing highly similar communities and red representing highly dissimilar communities. In addition, we cluster $\widehat{S}$ using hierarchical clustering and display the associated hierarchical clustering dendrogram. Figure duplicated from Lyzinski et al. (2017).

positions are allowed to lie on a lower-dimensional curve, and this model is amenable to semiparametric Gaussian mixture modeling (GMM) applied to the adjacency spectral embedding (ASE). The resulting *connectome code* derived via semiparametric GMM composed with ASE, which we denote, in shorthand, by $GMM \circ ASE$, captures latent connectome structure and elucidates biologically relevant neuronal properties.

HHMI Janelia has recently reconstructed the complete wiring diagram of the higher order parallel fiber system for associative learning in the larval *Drosophila* brain, the mushroom body (MB). Memories are thought to be stored as functional and structural changes in connections between neurons, but the complete circuit architecture of a higher-order learning center involved in memory formation or storage has not been known in any organism—until now, that is. Our MB connectome was obtained via serial section transmission electron microscopy of an entire larval *Drosophila* nervous system (Ohyama et al., 2015; Schneider-Mizell et al., 2016). This connectome contains the entirety of MB intrinsic neurons, called Kenyon cells, and all of their pre- and post-synaptic partners (Eichler et al., 2017).

We consider the right hemisphere MB. The connectome consists of four distinct types of neurons – Kenyon Cells (KC), Input Neurons (MBIN), Output Neurons (MBON), Projection Neurons (PN) – with directed connectivity illustrated in Figure 11. There are $n = 213$ neurons[2], with $n_{KC} = 100$, $n_{MBIN} = 21$, $n_{MBON} = 29$, and $n_{PN} = 63$. Figure 12 displays the

2. There are 13 isolates, all are KC; removing these isolates makes the (directed) graph one (weakly, but not strongly) connected component with 213 vertices and 7536 directed edges.

observed MB connectome as an adjacency matrix. Note that, in accordance with Figure 11, Figure 12 shows data (edges) in only eight of the 16 blocks.

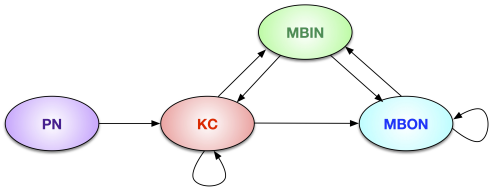


Figure 11: Illustration of the larval *Drosophila* mushroom body connectome as a directed graph on four neuron types. Figure duplicated from Priebe et al. (2017).

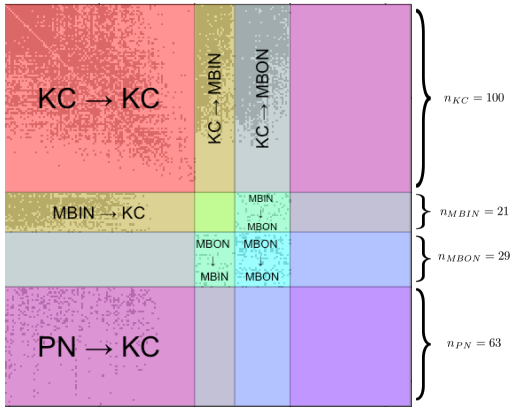


Figure 12: Observed data for our MB connectome as a directed adjacency matrix on four neuron types with 213 vertices ($n_{KC} = 100$, $n_{MBIN} = 21$, $n_{MBON} = 29$, and $n_{PN} = 63$) and 7536 directed edges. (This data matrix is available at `<http://www.cis.jhu.edu/~parky/MBstructure.html>`.) Figure duplicated from Priebe et al. (2017).

Due to its undeniable four-neuron-type connectivity structure, we might think of our MB connectome, to first order, as an observation from a $(K = 4)$-block directed stochastic block model (SBM) (Wang and Wong, 1987) on n vertices. This model is parameterized by (i) a block membership probability vector $\rho = [\rho_1, \cdots, \rho_K]$ such that $\rho_k \geq 0$ for all k and $\sum_k \rho_k = 1$ and (ii) a $K \times K$ block connectivity probability matrix B with entries $B_{k_1,k_2} \in [0, 1]$ governing the probability of directed edges from vertices in block k_1 to vertices in block k_2. For this

model of our MB connectome we have

$$B = \begin{bmatrix} B_{11} & B_{12} & B_{13} & 0 \\ B_{21} & 0 & B_{23} & 0 \\ 0 & B_{32} & B_{33} & 0 \\ B_{41} & 0 & 0 & 0 \end{bmatrix}$$

where the 0 in the B_{31} entry, for example, indicates that there are no directed connections from any MBON neuron to any KC neuron (as seen in Figures 11 and 12).

Theoretical results and methodological advances suggest that Gaussian mixture modeling (see, for example, Fraley and Raftery, 2002) composed with adjacency spectral embedding, denoted $GMM \circ ASE$, can be instructive in analysis of the (directed) SBM.

Since this graph is directed, we adapt our embedding technique just slightly. Given $d \geq 1$, the adjacency spectral embedding (ASE) of a *directed* graph on n vertices employs the singular value decomposition to represent the $n \times n$ adjacency matrix via $\mathbf{A} = \left[\mathbf{U} \mid \mathbf{U}^{\perp}\right]\left[\mathbf{S} \oplus \mathbf{S}^{\perp}\right]\left[\mathbf{V} \mid \mathbf{V}^{\perp}\right]^{\top}$ where $\mathbf{S}$ is the $d \times d$ diagonal matrix of the d largest singular values and $\mathbf{U}$ and $\mathbf{V}$ are the matrix of corresponding left and right singular vectors, and we embed the graph as n points in $\mathbb{R}^{2d}$ via the concatenation

$$\hat{\mathbf{X}} = \left[\mathbf{U}\mathbf{S}^{1/2} \mid \mathbf{V}\mathbf{S}^{1/2}\right] \in \mathbb{R}^{n \times 2d}.$$

(The scaled left-singular vectors $\mathbf{U}\mathbf{S}^{1/2}$ are interpreted as the "out-vector" representation of the digraph, modeling vertices' propensity to originate directed edges; similarly, $\mathbf{V}\mathbf{S}^{1/2}$ are interpreted as the "in-vectors".) Gaussian mixture modeling (GMM) then fits a K-component $2d$-dimensional Gaussian mixture model to the points $\hat{\mathbf{X}}_1, \cdots, \hat{\mathbf{X}}_n$ given by the rows of $\hat{\mathbf{X}}$. If the graph is a stochastic block model, then as we have discussed previously in Section 4.2, clustering the rows of the adjacency spectral embedding via Gaussian mixture modeling gives us consistent estimates for the latent positions.

Applying this procedure to our MB connectome yields the clustered embedding depicted via the pairs plot presented in Figure 13, with the associated cluster confusion matrix with respect to true neuron types presented in Table 2. The clusters are clearly coincident with the four true neuron types. (For ease of illustration, Figure 13 presents just the Out1 vs. Out2 subspace.)

There are two model selection problems inherent in spectral clustering in general, and in obtaining our clustered embedding (Figure 13) in particular: choice of embedding dimension $(\hat{d})$, and choice of mixture complexity $(\hat{K})$. A ubiquitous and principled method for choosing the number of dimensions in eigendecompositions and singular value decompositions is to examine the so-called *scree plot* and look for "elbows" or "knees" defining the cut-off between the top signal dimensions and the noise dimensions. Identifying a "best" method is, in general, impossible, as the bias-variance tradeoff demonstrates that, for small n, subsequent inference may be optimized by choosing a dimension *smaller than* the true signal dimension; see Section 3 of Jain et al. (2000) for a clear and concise illustration of this phenomenon. There are a plethora of variations for automating this singular value thresholding (SVT); Section 2.8 of Jackson (2004) provides a comprehensive discussion in

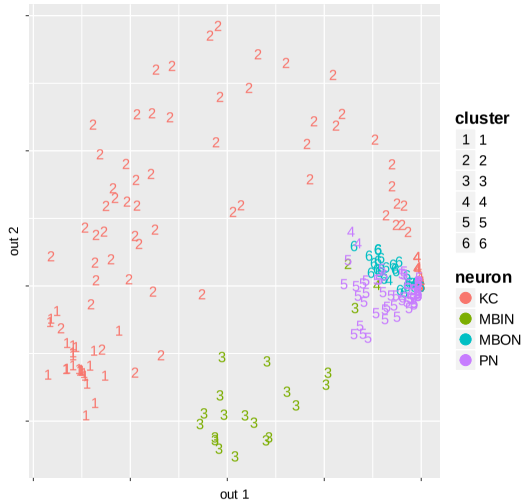


Figure 13: Plot for the clustered embedding of our MB connectome in the Out1 vs. Out2 dimensions. For ease of illustration, we present embedding results in this two-dimensional subspace. Recall that this is a two-dimensional visualization of six-dimensional structure. Figure duplicated from Priebe et al. (2017).

the context of principal components, and Chatterjee (2015) provides a theoretically-justified (but perhaps practically suspect, for small n) universal SVT. Using the profile likelihood SVT method of Zhu and Ghodsi (2006) yields a cut-off at three singular values, as depicted in Figure 14. Because this is a directed graph, we have both left- and right-singular vectors for each vertex; thus the SVT choice of three singular values results in $\hat{d} = 6$.

Similarly, a ubiquitous and principled method for choosing the number of clusters in, for example, Gaussian mixture models, is to maximize a fitness criterion penalized by model complexity. Common approaches include Akaike Information Criterion (AIC) (Akaike, 1974), Bayesian Information Criterion (BIC) (Schwarz, 1978), and Minimum Description Length (MDL) (Rissanen, 1978), to name a few. Again, identifying a "best" method is, in general, impossible, as the bias-variance tradeoff demonstrates that, for small n, inference performance may be optimized by choosing a number of clusters *smaller than* the true cluster complexity. The MCLUST algorithm of Fraley and Raftery (2002), as implemented in `R`, and its associated BIC applied to our MB connectome embedded via ASE into $\mathbb{R}^{\hat{d}=6}$, is maximized at six clusters, as depicted in Figure 15, and hence $\hat{K} = 6$.

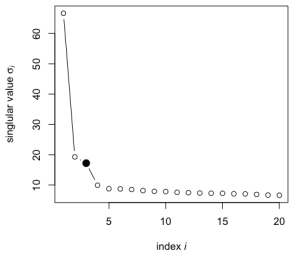


Figure 14: Model Selection: embedding dimension $\hat{d} = 6$ – the top 3 singular values and their associated left- and right-singular vectors – is chosen by SVT. Figure duplicated from Priebe et al. (2017).

	1	2	3	4	5	6
KC	**25**	**57**	0	**16**	2	0
MBIN	0	1	**19**	1	0	0
MBON	0	0	0	1	0	**28**
PN	0	0	0	2	**61**	0

Table 2: $GMM \circ ASE$ for our MB connectome yields $\hat{K} = 6$ clusters. The clusters are clearly coherent with but not perfectly aligned with the four true neuron types, as presented in this confusion matrix. Table duplicated from Priebe et al. (2017).

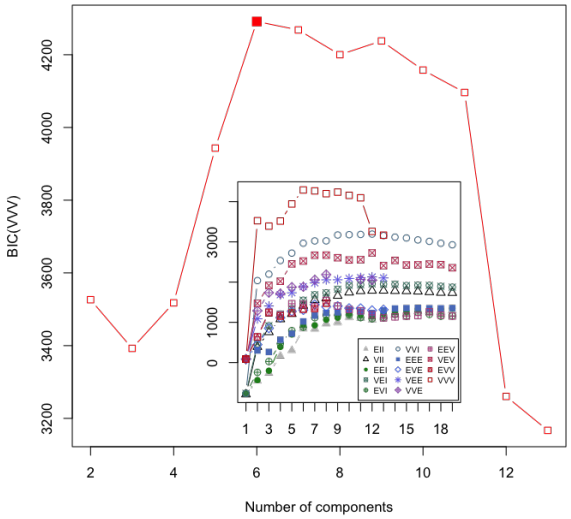


Figure 15: Model Selection: mixture complexity $\hat{K} = 6$ is chosen by BIC. (The inset shows that the main curve – BIC for dimensions 2 through 13 for MCLUST's most general covariance model, in red – dominates all other dimensions and all other models.) Figure duplicated from Priebe et al. (2017).

While BIC chooses $\hat{K} = 6$ clusters, it is natural to ask whether the distribution of KC across multiple clusters is an artifact of insufficiently parsimonious model selection. However, choosing four or five clusters not only (substantially) decreases BIC, but in fact leaves KC distributed across multiple clusters while splitting and/or merging other neuron types. In the direction of less parsimony, Figure 15 suggests that any choice from 7 to 11 clusters is competitive, in terms of BIC, with the maximizer $\hat{K} = 6$. Moreover, any of these choices only slightly decreases BIC, while leaving PN, MBIN, and MBON clustered (mostly) singularly and (mostly) purely and distributing KC across more clusters. Tables 3, 4, and 5 show cluster confusion matrices for other choices of K.

	1	2	3	4
KC	26	56	16	2
MBIN	0	20	1	0
MBON	0	28	1	0
PN	0	0	16	47

Table 3: Cluster confusion matrix for $GMM \circ ASE$ with 4 clusters. Choosing four or five clusters not only (substantially) decreases BIC (compared to $\hat{K} = 6$), but in fact leaves KC distributed across multiple clusters while splitting and/or merging other neuron types.

	1	2	3	4	5
KC	26	56	16	2	0
MBIN	0	20	1	0	0
MBON	0	0	1	0	28
PN	0	0	16	47	0

Table 4: Cluster confusion matrix for $GMM \circ ASE$ with 5 clusters. Choosing four or five clusters not only (substantially) decreases BIC (compared to $\hat{K} = 6$), but in fact leaves KC distributed across multiple clusters while splitting and/or merging other neuron types.

	1	2	3	4	5	6	7
KC	25	42	15	0	16	2	0
MBIN	0	0	1	19	1	0	0
MBON	0	0	0	0	1	0	28
PN	0	0	0	0	2	61	0

Table 5: Cluster confusion matrix for $GMM \circ ASE$ with 7 clusters. Any choice from 7 to 11 clusters only slightly decreases BIC (compared to $\hat{K} = 6$), while leaving PN, MBIN, and MBON clustered (mostly) singularly and (mostly) purely and distributing KC across more clusters. All tables reproduced from Priebe et al. (2017).

We see that our spectral clustering of the MB connectome via $GMM \circ ASE$, with principled model selection for choosing embedding dimension and mixture complexity, yields meaningful results: a single Gaussian cluster for each of MBIN, MBON, and PN, and multiple clusters for KC. That is, we have one substantial revision to Figure 11's illustration of the larval *Drosophila* mushroom body connectome as a directed graph on four neuron types:

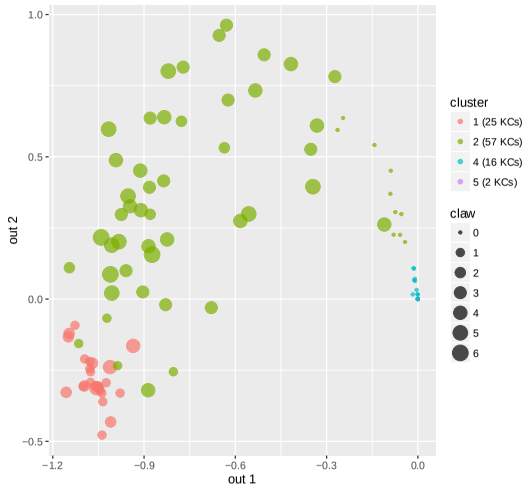


Figure 16: The multiple clusters for the KC neurons are capturing neuron age. Depicted are the first two dimensions for the KC neuron out-vectors, with color representing $\hat{K} = 6$ cluster membership – recall from Table 2 that the $n_{KC} = 100$ KCs are distributed across multiple clusters, with 25 neurons in cluster #1, 57 in #2, 0 in #3, 16 in #4, 2 in #5, and 0 in #6. The size of the dots represent the number of claws associated with the neurons. We see from the scatter plot that the embedded KC neurons arc from oldest (one-claw, lower left, cluster 1, in red), up and younger (more claws) through cluster 2 in green, and back down to youngest (zero-claw, clusters 4 and 5). See also Table 6. Recall that this is a two-dimensional visualization of six-dimensional structure. Figure duplicated from Priebe et al. (2017).

cluster	1	2	3	4	5	6
#KCs	25	57	0	16	2	0
claw: 1 (oldest)	15	4	—	0	0	—
claw: 2	7	4	—	0	0	—
claw: 3	0	15	—	0	0	—
claw: 4	3	13	—	0	0	—
claw: 5	0	8	—	0	0	—
claw: 6	0	3	—	0	0	—
claw: 0 (youngest)	0	10	—	16	2	—

Table 6: The multiple clusters for the KC neurons are capturing neuron age via the number of claws associated with the neuron. We see from the $\hat{K} = 6$ clustering table, for the $n_{KC} = 100$ KC neurons, that cluster 1 captures predominantly older neurons, cluster 2 captures both old and young neurons, and clusters 4 and 5 capture only the youngest neurons. See also Figure 16. Table reproduced from Priebe et al. (2017).

significant *substructure* associated with the KC neurons. Indeed, this hints at the possibility of a continuous, rather than discrete, structure for the KC. The paper Eichler et al. (2017) describes so-called "claws" associated with each KC neuron, and posits that KCs with only 1 claw are the oldest, followed in decreasing age by multi-claw KCs (from 2 to 6 claws), with finally the youngest KCs being those with 0 claws.

Figure 16 and Table 6 use this additional neuronal information to show that the multiple clusters for the KC neurons are capturing neuron age – and in a seemingly coherent geometry. Indeed, precisely because the clusters for the KC neurons are capturing neuron age – a continuous vertex attribute – again, in a seemingly coherent geometry, we define a "latent structure model" (LSM), a generalization of the SBM, together with a principled semiparametric spectral clustering methodology $SemiparGMM \circ ASE$ associated thereto. Specifically, we fit a continuous curve to (the KC subset of) the data in latent space and show that traversal of this curve corresponds monotonically to neuron age. To make this precise, we begin with a directed stochastic block model:

Definition 46 (Directed Stochastic Blockmodel (SBM)) *Let $d_{\text{out}} = d_{\text{in}}$, with $d = d_{\text{out}} + d_{\text{in}}$. We say that an n vertex graph $(\mathbf{A}, \mathbf{X}) \sim \text{RDPG}(F)$ is a directed stochastic blockmodel (SBM) with K blocks if the distribution F is a mixture of K point masses,*

$$dF = \sum_{k=1}^{K} \rho_k \delta_{x_k},$$

with block membership probability vector $\vec{\rho}$ in the unit $(K-1)$-simplex and distinct latent positions given by $\boldsymbol{\nu} = [\nu_1, \nu_2, \ldots, \nu_K]^\top \in \mathbb{R}^{K \times d}$. The first d_{out} entries of each latent position ν_k are the out-vectors, denoted $\xi_k \in \mathbb{R}^{d_{\text{out}}}$, and the remaining d_{in} elements are the in-vectors ζ_k. We write $G \sim SBM(n, \vec{\rho}, \boldsymbol{\xi}\boldsymbol{\zeta}^\top)$, and we refer to $\boldsymbol{\xi}\boldsymbol{\zeta}^\top \in \mathbb{R}^{K \times K}$ as the block connectivity probability matrix for the model.

We model the MB connectome as a four-component latent structure model (LSM), where LSM denotes the "generalized SBM" where each "block" may be generalized from point mass latent position distribution to latent position distribution with support on some curve (with the "block" curves disconnected, as (of course) are SBM's point masses). So LSM does have block structure, albeit not as simple as an SBM; and LSM will exhibit clustering, albeit just as transparently as an SBMs. As such, it is similar to other generalizations of SBMs, including the degree-corrected and hierarchical variants.

Definition 47 (Directed Latent Structure Model (LSM)) *Let $d_{\text{out}} = d_{\text{in}}$, and let F be a distribution on a set $\mathcal{X} = \mathcal{Y} \times \mathcal{Z} \subset \mathbb{R}^{d_{\text{out}}} \times \mathbb{R}^{d_{\text{in}}}$ such that $\langle y, z \rangle \in [0, 1]$ for all $y \in \mathcal{Y}$ and $z \in \mathcal{Z}$. We say that an n vertex graph $(\mathbf{A}, \mathbf{X}) \sim \text{RDPG}(F)$ is a directed latent structure model (LSM) with K "structure components" if the support of distribution F is a mixture of K (disjoint) curves,*

$$dF = \sum_{k=1}^{K} \rho_k dF_k(x),$$

with block membership probability vector $\vec{\rho}$ in the unit $(K-1)$-simplex and F_k supported on $\mathcal{C}_k$ and $\mathcal{C}_1, \cdots, \mathcal{C}_K$ disjoint. We write $G \sim LSM(n, \vec{\rho}, (F_1, \cdots, F_K))$.

We now investigate our MB connectome as an LSM with latent positions $X_i \overset{\text{i.i.d.}}{\sim} F$ where F is no longer a mixture of four point masses with one point mass per neuron type but instead supp(F) is three points and a continuous curve $\mathcal{C}_{KC}$.

Motivated by approximate normality of the adjacency spectral embedding of an RDPG, we consider estimating F via a semiparametric Gaussian mixture model for the $\hat{\mathbf{X}}_i$'s. Let H be a probability measure on a parameter space $\Theta \subset \mathbb{R}^d \times S_{d\times d}$, where $S_{d\times d}$ is the space of d-dimensional covariance matrices, and let $\{\varphi(\cdot;\theta) : \theta \in \Theta\}$ be a family of normal densities. Then the function given by

$$\alpha(\cdot; H) = \int_\Theta \varphi(\cdot;\theta) dH(\theta)$$

is a semiparametric GMM. $H \in \mathcal{M}$ is referred to as the mixing distribution of the mixture, where $\mathcal{M}$ is the class of all probability measures on Θ. If H consists of a finite number of atoms, then $\alpha(\cdot; H)$ is a finite normal mixture model with means, variances and proportions determined by the locations and weights of the point masses, and in Lindsay (1983), the author provides theory for maximum likelihood estimation (MLE) in this context.

Thus (ignoring covariances for presentation simplicity, so that $\theta \in \mathbb{R}^d$ is the component mean vector) we see that the central limit theorem suggests that we estimate the probability density function of the embedded MB connectome $\hat{\mathbf{X}}_1, \cdots, \hat{\mathbf{X}}_{n=213}$, under the LSM assumption, as the semiparametric GMM $\alpha(\cdot; H)$ with $\Theta = \mathbb{R}^6$ and where $H = F$ is supported by three points and a continuous curve $\mathcal{C}_{KC}$. Note that in the general case, where Θ includes both means and covariance matrices, we have $H = H_{F,n}$. The central limit theorem for the adjacency spectral embedding provides a large-sample approximation for $H_{F,n}$, and provides a mean-covariance constraint so that if we knew the latent position distribution F we would have no extra degrees of freedom (though perhaps a more challenging MLE optimization problem). As it is, we do our fitting in the general case, with simplifying constraints on the covariance structure associated with $\mathcal{C}_{KC}$.

Our MLE (continuing to ignore covariances for presentation simplicity) is given by

$$d\hat{H}(\theta) = \sum_{k=1}^{3} \widehat{\rho}_k I\{\theta = \widehat{\theta}_k\} + \left(1 - \sum_{k=1}^{3} \widehat{\rho}_k\right) \widehat{\rho}_{KC}(\theta) I\{\theta \in \widehat{\mathcal{C}}_{KC}\}$$

where $\widehat{\theta}_1, \widehat{\theta}_2, \widehat{\theta}_3$ are given by the means of the $GMM \circ ASE$ Gaussian mixture components for MBIN, MBON, and PN, and $\widehat{\mathcal{C}}_{KC} \subset \mathbb{R}^d$ is a one-dimensional curve. Figure 17 displays the MLE results from an EM optimization for the curve $\widehat{\mathcal{C}}_{KC}$ constrained to be quadratic, as detailed in the Appendix. (Model testing for $\mathcal{C}_{KC}$ in $\mathbb{R}^6$ does yield quadratic: testing the null hypothesis of linear against the alternative of quadratic yields clear rejection ($p < 0.001$), while there is insufficient evidence to favor H_A cubic over H_0 quadratic ($p \approx 0.1$).) That is, (continuing to ignore covariances for presentation simplicity) our structure discovery via $SemiparGMM \circ ASE$ yields an $\mathbb{R}^6$ latent position estimate for the MB connectome – a *connectome code* for the larval *Drosophila* mushroom body – as a semiparametric Gaussian mixture of three point masses and a continuous parameterized curve $\widehat{\mathcal{C}}_{KC}$; the three Gaussians correspond to three of the four neuron types, and the curve corresponds to the fourth neuron type (KC) with the parameterization capturing neuron age (see Figure 18). We note that in Eichler et al. (2017) , the authors suggest distance-to-neuropile δ_i – the

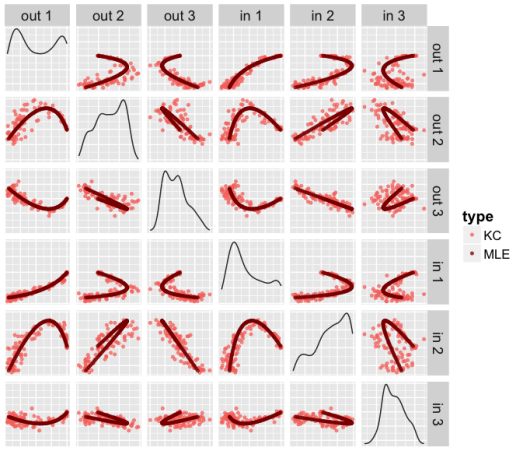


Figure 17: Semiparametric MLE $\widehat{\mathcal{C}}_{KC}$ for the KC latent-space curve in $\mathbb{R}^6$. Figure duplicated from Priebe et al. (2017).

distance to the MB neuropile from the bundle entry point of each KC neuron i – as a proxy for neuron age, and analyzes this distance in terms of number of claws for neuron i (see Figure 19). We now demonstrate that the correlation of this distance with the KC neurons' projection onto the parameterized curve $\widehat{\mathcal{C}}_{KC}$ is highly significant – this semiparametric spectral model captures neuroscientifically important structure in the connectome. To wit, we project each KC neuron's embedding onto our parameterized $\widehat{\mathcal{C}}_{KC}$ and study the relationship between the projection's position on the curve, t_i, and the neuron's age through the distance proxy δ_i (see Figures 20 and 21). We find significant correlation of δ_i with t_i – Spearman's $s = -0.271$, Kendall's $\tau = -0.205$, Pearson's $\rho = -0.304$, with $p < 0.01$ in each case – demonstrating that our semiparametric spectral modeling captures biologically relevant neuronal properties.

In summary, motivated by the results of a spectral clustering investigation of the recently-reconstructed synapse-level larval *Drosophila* mushroom body structural connectome, which demonstrate conclusively that modeling the Kenyon Cells (KC) demands additional latent space structure, we have developed semiparametric spectral modeling. Exploratory data analysis suggests that the MB connectome can be productively approximated by a four-component latent structure model (LSM), and the resulting MB connectome code derived via $SemiparGMM \circ ASE$ captures biologically relevant neuronal properties. Data and code for all our analyses are available at `http://www.cis.jhu.edu/~parky/MBstructure.html`.

Of course, the true connectome code is more elaborate, and cannot be completely encompassed by any simple latent position model – such a model precludes the propensity for transitivity, for example – but our semiparametric spectral modeling provides another step

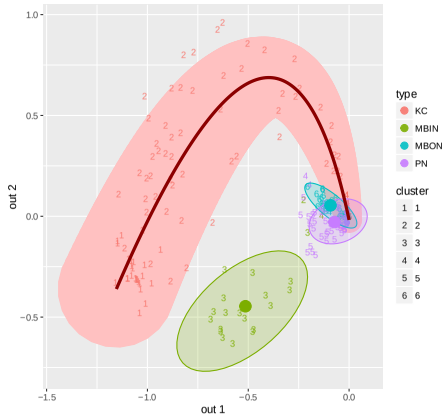


Figure 18: Semiparametric spectral latent space estimate of our MB connectome as three Gaussians and a KC curve: colors distinguish the four neuron types and numbers distinguish the original $\hat{K} = 6$ clusters. Recall that this is a two-dimensional visualization of six-dimensional structure. Figure duplicated from Priebe et al. (2017).

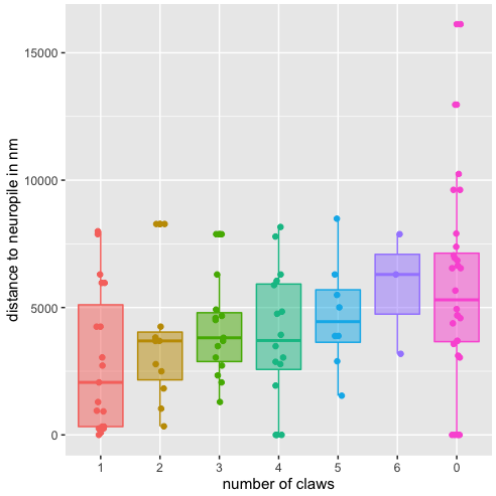


Figure 19: Relationship between number of claws and distance δ_i (a proxy for age) for the KC neurons, from Eichler et al. (2017). Figure duplicated from Priebe et al. (2017).

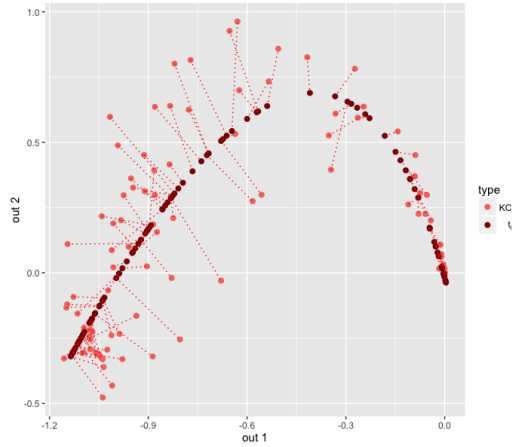


Figure 20: Projection of KC neurons onto the quadratic curve $\widehat{\mathcal{C}}_{KC}$, yielding projection point t_i for each neuron. Recall that this is a two-dimensional visualization of six-dimensional structure. Figure duplicated from Priebe et al. (2017).

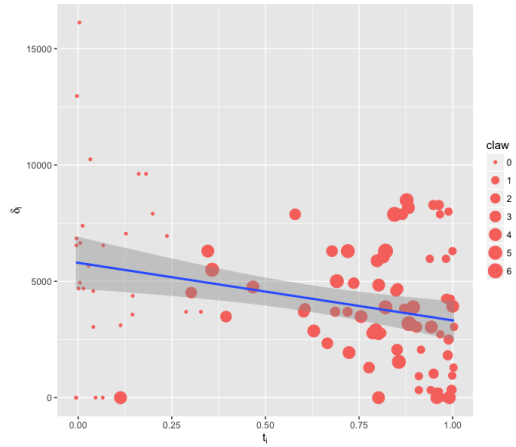


Figure 21: The correlation between the projection points t_i on the quadratic curve $\widehat{\mathcal{C}}_{KC}$ and distance δ_i (a proxy for age) for the KC neurons is highly significant, demonstrating that our semiparametric spectral modeling captures biologically relevant neuronal properties. Figure duplicated from Priebe et al. (2017).

along the path. In terms of a (partial) ladder of biological scales – for example, *C. elegans*, *Drosophila*, zebrafish, mouse, primate, and humans – this works moves us off the first rung for analysis of a complete neurons-as-vertices and synapses-as-edges connectome.

7. Conclusion: complexities, open questions, and future work

Our paradigm for statistical inference on random graphs is anchored by the familiar pillars of classical Euclidean inference. We exhibit estimates for graph parameters that satisfy (uniform) consistency and asymptotic normality, and we demonstrate how these estimates can be exploited in a bevy of subsequent inference tasks: community detection in heterogeneous networks, multi-sample graph hypothesis testing, and exploratory data analysis in connectomics. The random dot product graph model in which we ground our techniques has both linear algebraic transparency and wide applicability, since a hefty class of independent-edge graphs is well-approximated by RDPGs. The lynchpins of our approach are spectral decompositions of adjacency and Laplacian matrices, but many of our results and proof techniques can be applied to more general random matrices. In recent work, for example, we examine eigenvalue concentration for certain classes of random matrices (Cape et al., 2017a), and accurate estimation of covariance matrices (Cape et al., 2017b). As such, our methodology is a robust edifice from which to explore questions in graph inference, data analysis, and random matrix theory.

The results we summarize here are but the tip of the iceberg, though, and there persist myriad open problems in spectral graph inference—for random dot product graphs in particular, of course, but also for random graphs writ large. In this section, we outline some current questions of interest and describe future directions for research.

Our consistency results for the adjacency spectral embedding depend on knowing the correct embedding dimension. In real data, this optimal embedding dimension is typically not only not known, but, since the RDPG model is only an approximation to any true model, may well depend on the desired subsequent inference task. As we have pointed out in Section 6.3, multiple methods exist for estimating embedding dimension, and the universal singular value thresholding of Chatterjee (2015) and other thresholding methods (such as Fishkind et al., 2013) are theoretically justified in the large-n limit. For finite n, however, model selection is considerably trickier. Underestimation of the embedding dimension can markedly—and provably—bias subsequent inference. While we do not address it here, we remark that asymptotic results can be shown for the adjacency spectral embedding of a d-dimensional RDPG when the chosen embedding dimension is $d' < d$. On the other hand, if the embedding dimension is overestimated, no real signal is lost; therefore, most embedding methods continue to perform well, albeit with some loss of efficiency due to increased variance. Precisely quantifying this loss of efficiency is very much an open question and is related to analogous classical questions in random matrix theory (see Tao and Vu, 2012).

In our RDPG model, an important assumption is that the $\mathbf{P}$ matrix be positive semidefinite. While this limits the model, as we mentioned in the introduction, the *generalized random dot product graph model* (gRDPG) of Rubin-Delanchy et al. (2017) is not similarly restricted, but nevertheless shares many important useful properties of the RDPG. Considering a matrix

of latent positions $\mathbf{X} \in \mathbb{R}^{p+q}$, one can set $\mathbf{P} = \mathbf{X}\mathbf{I}_{p,q}\mathbf{X}^\top$ where $\mathbf{I}_{p,q}$ is the diagonal matrix of size $(p+q)\times(p+q)$ with p diagonal entries being 1 and q diagonal entries being -1. Under this generalization, any $\mathbf{P}$ can be obtained, provided that $p+q$ is appropriately chosen. This then implies that any latent position graph model, even a non-positive-semidefinite one, can be approximated arbitrarily closely by this generalized RDPG. We are currently investigation extensions of our results on consistency, distributional limits, and testing for estimates of gRDPGs parameters, including, for instance, suitable reformulations of the two-sample hypothesis tests for, say, mixed membership stochastic blockmodels.

We also wish to adapt our procedures to weighted graphs. For simple weighting, such as Poisson-distributed weights, our existing methodology applies quite naturally to the weighted adjacency matrix. More intricate questions arise when the weights are contaminated or their distribution is heavily skewed. In such cases, one can "pass to ranks"; that, is replace the nonzero weight by its normalized rank among all the edge weights. This mitigates skew and works well in many examples, but a deeper theoretical analysis of this procedure, as well as other approaches to weighted graph inference, remain open.

To address graphs with self-edges, note that the random dot product graph model does not preclude such edges. Indeed, since $\mathbf{P}$ is not constrained to be hollow, the adjacency matrix $\mathbf{A}$ thus generated need not be hollow, either. However, the generative mechanism for self-edges in a graph may be different from the mechanism for edges between two different vertices. One approach to addressing this is to set the diagonal entries of $\mathbf{A}$ to zero and then *augment* the diagonal artificially with imputed values. In fact, even when there are no self loops, such procedures can improve finite-sample inferential performance. In Marchette et al. (2011), it is demonstrated that augmenting the diagonal via $\mathbf{A}_{ii} = d_i/(n+1)$, where d_i is the degree of vertex i, can improve inference in certain cases. Similarly, in Scheinerman and Tucker (2010), we find an iterative procedure to find a diagonal augmentation consistent with the low-rank structure of $\mathbf{P}$. It is still unclear exactly what augmentation of the diagonal, if any, might be optimal for each particular inference task.

In the case when the vertices or edges are corrupted by occlusion or noise, the work of Priebe et al. (2015) and Levin and Lyzinski (2017) consider approaches to vertex classification and graph recovery, demonstrating that spectral embeddings are robust to certain sources of graph error. Violations of the independent edge assumption, though, can lead to more significant hurdles, both theoretical practical, since it is a requirement for a number of the concentration inequalities on which we depend.

For joint graph inference and testing, open problems abound. We mention, for instance, the analysis of the omnibus embedding when the m graphs are correlated, or when some are corrupted; a closer examination of the impact of the Procrustes alignment on power; the development of an analogue to a Tukey test for determining which graphs differ when we test for the equality of distribution for more than two graphs; the comparative efficiency of the omnibus embedding relative to other spectral decompositions; and a quantification of the trade-off for subsequent inference between a large number of independent graphs and large graph size (Tang et al., 2016).

In sum, the random dot product graph is a compact, manageable, and applicable model. The Euclidean nature of the adjacency and Laplacian spectral embedding for a random dot product graph allows us to approach statistical inference in this setting from a familiar Euclidean perspective. Both the adjacency spectral embedding and the Laplacian spectral embedding can be profitably leveraged for latent position estimation and single- and multi-sample graph hypothesis testing. Moreover, our distributional results for these spectral embeddings provide reassuring classical analogues of asymptotic normality for estimators, and in current ongoing work, we consider how to compare asymptotic relative efficiency of different estimators for graph parameters. While spectral methods may not always be optimal for a given task, they are often feasible and can provide a way to accurately initialize more complex procedures. Moreover, these Euclidean representations of graph data render possible the application of techniques for analysis of Euclidean data—clustering, classification, and density estimation, for instance—to graphs. As we have outlined above, while many important theoretical and practical challenges remain, spectral embeddings for random dot product graphs constitute an important piece of the greater puzzle of random graph inference.

Acknowledgments

We would like to acknowledge support for this project from the Johns Hopkins University Human Language Technology Center of Excellence (JHU HLT COE); the XDATA, SIMPLEX, and D3M programs of the Defense Advanced Research Projects Agency (DARPA) administered through contract FA8750-12-2-0303, contract N66001-15-C-4041, and contract FA8750-17-2-0112, respectively; and the Acheson J. Duncan Fund.

Appendix A. Appendix

In the appendix, we provide details for the proofs of our results on consistency and asymptotic normality of the adjacency spectral embedding, as well as an outline of the proof of the central limit theorem for the Laplacian spectral embedding. Our notation remains as stated in Section 4. We begin with a detailed proof of our result on the consistency, in the $2 \to \infty$ norm, of the ASE for latent position recovery in RDPGs.

A.1 Proof of Theorem 26

Let us recall **Theorem 26**: Let $\mathbf{A}_n \sim \text{RDPG}(\mathbf{X}_n)$ for $n \geq 1$ be a sequence of random dot product graphs where the $\mathbf{X}_n$ is assumed to be of rank d for all n sufficiently large. Denote by $\hat{\mathbf{X}}_n$ the adjacency spectral embedding of $\mathbf{A}_n$ and let $(\hat{\mathbf{X}}_n)_i$ and $(\mathbf{X}_n)_i$ be the i-th row of $\hat{\mathbf{X}}_n$ and $\mathbf{X}_n$, respectively. Let E_n be the event that there exists an orthogonal transformation $\mathbf{W}_n \in \mathbb{R}^{d\times d}$ such that

$$\max_i \|(\hat{\mathbf{X}})_i - \mathbf{W}_n(\mathbf{X}_n)_i\| \leq \frac{Cd^{1/2}\log^2 n}{\delta^{1/2}(\mathbf{P}_n)}$$

where $C > 0$ is some fixed constant and $\mathbf{P}_n = \mathbf{X}_n\mathbf{X}_n^\top$. Then E_n occurs asymptotically almost surely; that is, $\Pr(E_n) \to 1$ as $n \to \infty$.

The proof of Theorem 26 will follow from a succession of supporting results. We note that Theorem 50, which deals with the accuracy of spectral embedding estimates in Frobenius norm, may be of independent interest. We begin with the following simple but essential proposition, in which we show that $\mathbf{U}_\mathbf{P}^\top\mathbf{U}_\mathbf{A}$ is close to an orthogonal transformation. For ease of exposition, in the remainder of this subsection we shall suppress the subscript index n from our matrices $\mathbf{X}_n$, $\mathbf{A}_n$ and $\hat{\mathbf{X}}_n$.

Proposition 48 *Let $\mathbf{A} \sim \text{RDPG}(\mathbf{X})$ and let $\mathbf{W}_1\mathbf{\Sigma}\mathbf{W}_2^\top$ be the singular value decomposition of $\mathbf{U}_\mathbf{P}^\top\mathbf{U}_\mathbf{A}$. Then with high probability,*

$$\|\mathbf{U}_\mathbf{P}^\top\mathbf{U}_\mathbf{A} - \mathbf{W}_1\mathbf{W}_2^\top\| = O(\delta^{-1}(\mathbf{P})).$$

Proof Let $\sigma_1, \sigma_2, \ldots, \sigma_d$ denote the singular values of $\mathbf{U}_\mathbf{P}^\top\mathbf{U}_\mathbf{A}$ (the diagonal entries of $\mathbf{\Sigma}$). Then $\sigma_i = \cos(\theta_i)$ where the θ_i are the principal angles between the subspaces spanned by $\mathbf{U}_\mathbf{A}$ and $\mathbf{U}_\mathbf{P}$. Furthermore, by the Davis-Kahan Theorem,

$$\|\mathbf{U}_\mathbf{A}\mathbf{U}_\mathbf{A}^\top - \mathbf{U}_\mathbf{P}\mathbf{U}_\mathbf{P}^\top\| = \max_i |\sin(\theta_i)| \leq \frac{C\sqrt{d}\|\mathbf{A}-\mathbf{P}\|}{\lambda_d(\mathbf{P})}$$

for sufficiently large n. Recall here $\lambda_d(\mathbf{P})$ denotes the d-th largest eigenvalue of $\mathbf{P}$. The spectral norm bound for $\mathbf{A}-\mathbf{P}$ from Theorem 21 along with the assumption that $\lambda_d(\mathbf{P})/\delta(\mathbf{P}) \geq c_0$ for some constant c_0 in Assumption 1 yield

$$\|\mathbf{U}_\mathbf{A}\mathbf{U}_\mathbf{A}^\top - \mathbf{U}_\mathbf{P}\mathbf{U}_\mathbf{P}^\top\| \leq \frac{C\sqrt{d}}{\delta^{1/2}(\mathbf{P})}.$$

We thus have

$$\begin{aligned}\|\mathbf{U}_\mathbf{P}^\top\mathbf{U}_\mathbf{A} - \mathbf{W}_1\mathbf{W}_2^\top\| &= \|\mathbf{\Sigma} - I\| = \max_i|1-\sigma_i| \leq \max_i(1-\sigma_i^2)\\ &= \max_i \sin^2(\theta_i) = \|\mathbf{U}_\mathbf{A}\mathbf{U}_\mathbf{A}^T - \mathbf{U}_\mathbf{P}\mathbf{U}_\mathbf{P}^\top\|^2 = O(\delta^{-1}(\mathbf{P}))\end{aligned}$$

as desired. ■

Denote by $\mathbf{W}^*$ the orthogonal matrix $\mathbf{W}_1\mathbf{W}_2^\top$ as defined in the above proposition. We now establish the following key lemma. The lemma allows us to exchange the order of the orthogonal transformation $\mathbf{W}^*$ and the diagonal scaling transformation $\mathbf{S}_\mathbf{A}$ or $\mathbf{S}_\mathbf{P}$.

Lemma 49 *Let $(\mathbf{A}, \mathbf{X}) \sim \text{RDPG}(F)$ with sparsity factor ρ_n. Then asymptotically almost surely,*

$$\|\mathbf{W}^*\mathbf{S}_\mathbf{A} - \mathbf{S}_\mathbf{P}\mathbf{W}^*\|_F = O(\log n) \tag{48}$$

and

$$\|\mathbf{W}^*\mathbf{S}_\mathbf{A}^{1/2} - \mathbf{S}_\mathbf{P}^{1/2}\mathbf{W}^*\|_F = O((\log n)\,\delta^{-1/2}(\mathbf{P})) \tag{49}$$

Proof Let $\mathbf{R} = \mathbf{U_A} - \mathbf{U_P}\mathbf{U_P}^\top\mathbf{U_A}$. We note that $\mathbf{R}$ is the residual after projecting $\mathbf{U_A}$ orthogonally onto the column space of $\mathbf{U_P}$, and thus

$$\|\mathbf{U_A} - \mathbf{U_P}\mathbf{U_P}^\top\mathbf{U_A}\|_F \le \min_{\mathbf{W}} \|\mathbf{U_A} - \mathbf{U_P}\mathbf{W}\|_F$$

where the minimization is over all orthogonal matrices $\mathbf{W}$. The variant of the Davis-Kahan Theorem given in Eq. (9) then implies

$$\|\mathbf{U_A} - \mathbf{U_P}\mathbf{U_P}^\top\mathbf{U_A}\|_F \le O(\delta^{-1/2}(\mathbf{P})).$$

We next derive that

$$\begin{aligned}
\mathbf{W}^*\mathbf{S_A} &= (\mathbf{W}^* - \mathbf{U_P}^\top\mathbf{U_A})\mathbf{S_A} + \mathbf{U_P}^\top\mathbf{U_A}\mathbf{S_A} = (\mathbf{W}^* - \mathbf{U_P}^\top\mathbf{U_A})\mathbf{S_A} + \mathbf{U_P}^\top\mathbf{A}\mathbf{U_A} \\
&= (\mathbf{W}^* - \mathbf{U_P}^\top\mathbf{U_A})\mathbf{S_A} + \mathbf{U_P}^\top(\mathbf{A}-\mathbf{P})\mathbf{U_A} + \mathbf{U_P}^\top\mathbf{P}\mathbf{U_A} \\
&= (\mathbf{W}^* - \mathbf{U_P}^\top\mathbf{U_A})\mathbf{S_A} + \mathbf{U_P}^\top(\mathbf{A}-\mathbf{P})\mathbf{R} + \mathbf{U_P}^\top(\mathbf{A}-\mathbf{P})\mathbf{U_P}\mathbf{U_P}^\top\mathbf{U_A} + \mathbf{U_P}^\top\mathbf{P}\mathbf{U_A} \\
&= (\mathbf{W}^* - \mathbf{U_P}^\top\mathbf{U_A})\mathbf{S_A} + \mathbf{U_P}^\top(\mathbf{A}-\mathbf{P})\mathbf{R} + \mathbf{U_P}^\top(\mathbf{A}-\mathbf{P})\mathbf{U_P}\mathbf{U_P}^\top\mathbf{U_A} + \mathbf{S_P}\mathbf{U_P}^\top\mathbf{U_A}
\end{aligned}$$

Writing $\mathbf{S_P}\mathbf{U_P}^\top\mathbf{U_A} = \mathbf{S_P}(\mathbf{U_P}^\top\mathbf{U_A} - \mathbf{W}^*) + \mathbf{S_P}\mathbf{W}^*$ and rearranging terms, we obtain

$$\begin{aligned}
\|\mathbf{W}^*\mathbf{S_A} - \mathbf{S_P}\mathbf{W}^*\|_F &\le \|\mathbf{W}^* - \mathbf{U_P}^\top\mathbf{U_A}\|_F(\|\mathbf{S_A}\| + \|\mathbf{S_P}\|) + \|\mathbf{U_P}^\top(\mathbf{A}-\mathbf{P})\mathbf{R}\|_F \\
&\quad + \|\mathbf{U_P}^\top(\mathbf{A}-\mathbf{P})\mathbf{U_P}\mathbf{U_P}^\top\mathbf{U_A}\|_F \\
&\le O(1) + O(1) + \|\mathbf{U_P}^\top(\mathbf{A}-\mathbf{P})\mathbf{U_P}\|_F\|\mathbf{U_P}^\top\mathbf{U_A}\|
\end{aligned}$$

asymptotically almost surely. Now, $\|\mathbf{U_P}^\top\mathbf{U_A}\| \le 1$. Hence we can focus on the term $\mathbf{U_P}^\top(\mathbf{A}-\mathbf{P})\mathbf{U_P}$, which is a $d \times d$ matrix whose ij-th entry is of the form

$$\begin{aligned}
(\mathbf{U_P}^\top(\mathbf{A}-\mathbf{P})\mathbf{U_P})_{ij} &= \sum_{k=1}^{n}\sum_{l=1}^{n}(\mathbf{A}_{kl} - \mathbf{P}_{kl})\mathbf{U}_{ki}\mathbf{U}_{lj} \\
&= 2\sum_{k,l:k<l}(\mathbf{A}_{kl} - \mathbf{P}_{kl})\mathbf{U}_{ki}\mathbf{U}_{lj} - \sum_k \mathbf{P}_{kk}\mathbf{U}_{ki}\mathbf{U}_{kj}
\end{aligned}$$

where $\mathbf{U}_{\cdot i}$ and $\mathbf{U}_{\cdot j}$ are the i-th and j-th columns of $\mathbf{U_P}$. Thus, conditioned on $\mathbf{P}$, $(\mathbf{U_P}^\top(\mathbf{A}-\mathbf{P})\mathbf{U_P})_{ij}$ is a sum of independent mean 0 random variables and a term of order $O(1)$. Now, by Hoeffding's inequality,

$$\Pr\left[\left|\sum_{k,l:k<l} 2(\mathbf{A}_{kl} - \mathbf{P}_{kl})\mathbf{U}_{ki}u_{lj}\right| \ge t\right] \le 2\exp\Big(\frac{-2t^2}{\sum_{k,l:k<l}(2\mathbf{U}_{ki}\mathbf{U}_{lj})^2}\Big) \le 2\exp(-t^2).$$

Therefore, each entry of $\mathbf{U_P}^\top(\mathbf{A}-\mathbf{P})\mathbf{U_P}$ is of order $O(\log n)$ with high probability, and as a consequence, since $\mathbf{U_P}^\top(\mathbf{A}-\mathbf{P})\mathbf{U_P}$ is a $d \times d$ matrix,

$$\|\mathbf{U_P}^\top(\mathbf{A}-\mathbf{P})\mathbf{U_P}\|_F = O(\log n) \tag{50}$$

with high probability. We establish that

$$\|\mathbf{W}^*\mathbf{S_A}^{1/2} - \mathbf{S_P}^{1/2}\mathbf{W}^*\|_F = O((\log n)\lambda_d^{-1/2}(\mathbf{P}))$$

by noting that the ij-th entry of $\mathbf{W}^*\mathbf{S_A}^{1/2} - \mathbf{S_P}^{1/2}\mathbf{W}^*$ can be written as

$$\mathbf{W}^*_{ij}(\lambda_i^{1/2}(\mathbf{A}) - \lambda_j^{1/2}(\mathbf{P})) = \mathbf{W}^*_{ij}\frac{\lambda_i(\mathbf{A}) - \lambda_j(\mathbf{P})}{\lambda_i^{1/2}(\mathbf{A}) + \lambda_j^{1/2}(\mathbf{P})},$$

and the desired bound follows from the above after bounding $\lambda_i(\mathbf{A})$, either by Weyl's inequality and Theorem 21, or, alternatively, by a Kato-Temple inequality from Cape et al. (2017a). ■

We next present Theorem 50, which allows us to write the Frobenius norm difference of the adjacency spectral embedding $\hat{\mathbf{X}}$ and the true latent position $\mathbf{X}$ in terms of the Frobenius norm difference of $\mathbf{A}-\mathbf{P}$ and smaller order terms.

Theorem 50 *Let $\mathbf{A} \sim \mathrm{RDPG}(\mathbf{X})$. Then there exists an orthogonal matrix $\mathbf{W}$ such that, with high probability,*

$$\|\hat{\mathbf{X}} - \mathbf{X}\mathbf{W}\|_F = \|(\mathbf{A}-\mathbf{P})\mathbf{U_P}\mathbf{S_P}^{-1/2}\|_F + O((\log n)\delta^{-1/2}(\mathbf{P}))$$

Proof Let

$$\begin{aligned}
\mathbf{R}_1 &= \mathbf{U_P}\mathbf{U_P}^\top\mathbf{U_A} - \mathbf{U_P}\mathbf{W}^* \\
\mathbf{R}_2 &= (\mathbf{W}^*\mathbf{S_A}^{1/2} - \mathbf{S_P}^{1/2}\mathbf{W}^*).
\end{aligned}$$

We deduce that

$$\begin{aligned}
\hat{\mathbf{X}} - \mathbf{U_P}\mathbf{S_P}^{1/2}\mathbf{W}^* &= \mathbf{U_A}\mathbf{S_A}^{1/2} - \mathbf{U_P}W^*\mathbf{S_A}^{1/2} + \mathbf{U_P}(\mathbf{W}^*\mathbf{S_A}^{1/2} - \mathbf{S_P}^{1/2}W^*) \\
&= (\mathbf{U_A} - \mathbf{U_P}\mathbf{U_P}^\top\mathbf{U_A})\mathbf{S_A}^{1/2} + \mathbf{R}_1\mathbf{S_A}^{1/2} + \mathbf{U_P}\mathbf{R}_2 \\
&= \mathbf{U_A}\mathbf{S_A}^{1/2} - \mathbf{U_P}\mathbf{U_P}^\top\mathbf{U_A}\mathbf{S_A}^{1/2} + \mathbf{R}_1\mathbf{S_A}^{1/2} + \mathbf{U_P}\mathbf{R}_2
\end{aligned}$$

Observe that $\mathbf{U_P}\mathbf{U_P}^\top\mathbf{P} = \mathbf{P}$ and $\mathbf{U_A}\mathbf{S_A}^{1/2} = \mathbf{A}\mathbf{U_A}\mathbf{S_A}^{-1/2}$. Hence

$$\hat{\mathbf{X}} - \mathbf{U_P}\mathbf{S_P}^{1/2}\mathbf{W}^* = (\mathbf{A}-\mathbf{P})\mathbf{U_A}\mathbf{S_A}^{-1/2} - \mathbf{U_P}\mathbf{U_P}^\top(\mathbf{A}-\mathbf{P})\mathbf{U_A}\mathbf{S_A}^{-1/2} + \mathbf{R}_1\mathbf{S_A}^{1/2} + \mathbf{U_P}\mathbf{R}_2$$

Writing

$$\mathbf{R}_3 = \mathbf{U_A} - \mathbf{U_P}\mathbf{W}^* = \mathbf{U_A} - \mathbf{U_P}\mathbf{U_P}^\top\mathbf{U_A} + \mathbf{R}_1,$$

we derive that

$$\begin{aligned}
\hat{\mathbf{X}} - \mathbf{U_P}\mathbf{S_P}^{1/2}\mathbf{W}^* &= (\mathbf{A}-\mathbf{P})\mathbf{U_P}\mathbf{W}^*\mathbf{S_A}^{-1/2} - \mathbf{U_P}\mathbf{U_P}^\top(\mathbf{A}-\mathbf{P})\mathbf{U_P}\mathbf{W}^*\mathbf{S_A}^{-1/2} \\
&\quad + (\mathbf{I} - \mathbf{U_P}\mathbf{U_P}^\top)(\mathbf{A}-\mathbf{P})\mathbf{R}_3\mathbf{S_A}^{-1/2} + \mathbf{R}_1\mathbf{S_A}^{1/2} + \mathbf{U_P}\mathbf{R}_2
\end{aligned}$$

Recalling that $\|\mathbf{U_A} - \mathbf{U_P}\mathbf{U_P}^\top\mathbf{U_A}\|_F = O(\delta^{-1/2}(\mathbf{P}))$ with high probability, we have

$$\|\mathbf{R}_1\|_F = O(\delta^{-1}(\mathbf{P})), \quad \|\mathbf{R}_2\|_F = O((\log n)\,\delta^{-1/2}(\mathbf{P})), \quad \text{and } \|\mathbf{R}_3\|_F = O(\delta^{-1/2}(\mathbf{P}))$$

with high probability. Furthermore, a similar application of Hoeffding's inequality to that in the proof of Lemma 49, along with an application of Weyl's inequality and Theorem 21 to bound $\lambda_i(\mathbf{A})$, ensures that

$$\|\mathbf{U_P}\mathbf{U_P}^\top(\mathbf{A}-\mathbf{P})\mathbf{U_P}\mathbf{W}^*\mathbf{S_A}^{-1/2}\|_F \leq \|\mathbf{U_P}^\top(\mathbf{A}-\mathbf{P})\mathbf{U_P}\|_F\|\mathbf{S_A}^{-1/2}\|_F = O(\log n\delta^{-1/2}(\mathbf{P}))/$$

As a consequence, with high probability

$$\begin{aligned}\|\hat{\mathbf{X}}-\mathbf{U_P}\mathbf{S_P}^{1/2}\mathbf{W}^*\|_F &= \|(\mathbf{A}-\mathbf{P})\mathbf{U_P}\mathbf{W}^*\mathbf{S_A}^{-1/2}\|_F + O((\log n)\,\delta^{-1/2}(\mathbf{P}))\\ &= \|(\mathbf{A}-\mathbf{P})\mathbf{U_P}\mathbf{S_P}^{-1/2}\mathbf{W}^* - (\mathbf{A}-\mathbf{P})\mathbf{U_P}(\mathbf{S_P}^{-1/2}\mathbf{W}^* - \mathbf{W}^*\mathbf{S_A}^{-1/2})\|_F\\ &\quad + O((\log n)\,\delta^{-1/2}(\mathbf{P}))\end{aligned}$$

A very similar argument to that employed in the proof of Lemma 49 implies that

$$\|\mathbf{S_P}^{-1/2}\mathbf{W}^* - \mathbf{W}^*\mathbf{S_A}^{-1/2}\|_F = O((\log n)\,\delta^{-3/2}(\mathbf{P}))$$

with high probability. We thus obtain

$$\begin{aligned}\|\hat{\mathbf{X}}-\mathbf{U_P}\mathbf{S_P}^{1/2}\mathbf{W}^*\|_F &= \|(\mathbf{A}-\mathbf{P})\mathbf{U_P}\mathbf{S_P}^{-1/2}\mathbf{W}^*\|_F + O((\log n)\,\delta^{-1/2}(\mathbf{P}))\\ &= \|(\mathbf{A}-\mathbf{P})\mathbf{U_P}\mathbf{S_P}^{-1/2}\|_F + O((\log n)\,\delta^{-1/2}(\mathbf{P}))\end{aligned} \tag{51}$$

with high probability. Finally, to complete the proof, we note that $\mathbf{X} = \mathbf{U_P}\mathbf{S_P}^{1/2}\tilde{\mathbf{W}}$ for some orthogonal matrix $\mathbf{W}$. Since $\mathbf{W}^*$ is also orthogonal, we conclude that there exists some orthogonal $\mathbf{W}$ for which $\mathbf{X}\mathbf{W} = \mathbf{U_P}\mathbf{S_P}^{1/2}\mathbf{W}^*$, as desired. ■

We are now ready to prove the $2\to\infty$ consistency we assert in Theorem 26.

Proof

To establish Theorem 26, we note that from Theorem 50

$$\|\hat{\mathbf{X}}-\mathbf{X}\mathbf{W}\|_F = \|(\mathbf{A}-\mathbf{P})\mathbf{U_P}\mathbf{S_P}^{-1/2}\|_F + O((\log n)\,\delta^{-1/2}(\mathbf{P}))$$

and hence

$$\begin{aligned}\max_i\|\hat{\mathbf{X}}_i-\rho_n^{1/2}\mathbf{W}\mathbf{X}_i\| &\leq \frac{1}{\lambda_d^{1/2}(\mathbf{P})}\max_i\|((\mathbf{A}-\mathbf{P})\mathbf{U_P})_i\| + O((\log n)\,\delta^{-1/2}(\mathbf{P}))\\ &\leq \frac{d^{1/2}}{\lambda_d^{1/2}(\mathbf{P})}\max_j\|(\mathbf{A}-\mathbf{P})\mathbf{U}_{\cdot j}\|_\infty + O((\log n)\,\delta^{-1/2}(\mathbf{P}))\end{aligned}$$

where $\mathbf{U}_{\cdot j}$ denotes the j-th column of $\mathbf{U_P}$. Now, for a given j and a given index i, the i-th element of the vector $(\mathbf{A}-\mathbf{P})\mathbf{U}_{\cdot j}$ is of the form

$$\sum_k(\mathbf{A}_{ik}-\mathbf{P}_{ik})\mathbf{U}_{kj}$$

and once again, by Hoeffding's inequality, the above term is $O(\log n)$ asymptotically almost surely. Taking the union bound over all indices i and all columns j of $\mathbf{U_P}$, we conclude that with high probability,

$$\begin{aligned}\max_i\|\hat{\mathbf{X}}_i-\rho_n^{1/2}\mathbf{W}\mathbf{X}_i\| &\leq \frac{Cd^{1/2}}{\lambda_d^{1/2}(P)}\log^2 n + O((\log n)\,\delta^{-1/2}(\mathbf{P}))\\ &\leq \frac{Cd^{1/2}\log^2 n}{\delta^{1/2}(\mathbf{P})}\end{aligned}$$

as desired. ■

Now, we move to our distributional results.

A.2 Proof of the central limit theorem for the adjacency spectral embedding

.

Recall **Theorem 27**: Let $(\mathbf{A}_n,\mathbf{X}_n)\sim\text{RDPG}(F)$ be a sequence of adjacency matrices and associated latent positions of a d-dimensional random dot product graph according to an inner product distribution F. Let $\Phi(\mathbf{x},\mathbf{\Sigma})$ denote the cdf of a (multivariate) Gaussian with mean zero and covariance matrix $\mathbf{\Sigma}$, evaluated at $\mathbf{x}\in\mathbb{R}^d$. Then there exists a sequence of orthogonal d-by-d matrices $(\mathbf{W}_n)_{n=1}^\infty$ such that for all $\boldsymbol{z}\in\mathbb{R}^d$ and for any fixed index i,

$$\lim_{n\to\infty}\Pr\left[n^{1/2}\left(\hat{\mathbf{X}}_n\mathbf{W}_n-\mathbf{X}_n\right)_i\leq\boldsymbol{z}\right] = \int_{\text{supp}\,F}\Phi\left(\boldsymbol{z},\mathbf{\Sigma}(\mathbf{x})\right)dF(\mathbf{x}),$$

where

$$\mathbf{\Sigma}(\mathbf{x}) = \Delta^{-1}\mathbb{E}\left[(\mathbf{x}^\top\mathbf{X}_1-(\mathbf{x}^\top\mathbf{X}_1)^2)\mathbf{X}_1\mathbf{X}_1^\top\right]\Delta^{-1};\quad\text{and }\Delta=\mathbb{E}[\mathbf{X}_1\mathbf{X}_1^\top]. \tag{52}$$

To prove this, we begin with the following simple lemma which indicates that when the rows of $\mathbf{X}$ are sampled i.i.d. from some distribution F, that the eigenvalues of $\mathbf{X}$ grows proportionally with n.

Lemma 51 *With notation as above, let F be an inner product distribution and suppose $\mathbf{X}_1,\cdots,\mathbf{X}_n,\mathbf{Y}$ be i.i.d F. Suppose also that $\Delta=\mathbb{E}[\mathbf{X}_1\mathbf{X}_1^\top]$ is of rank d. Then for $1\leq i\leq d$, $\lambda_i(\mathbf{P})=\Omega(n\lambda_i(\Delta))$ almost surely.*

Proof Recall that for any matrix $\mathbf{H}$, the nonzero eigenvalues of $\mathbf{H}^\top\mathbf{H}$ are the same as those of $\mathbf{H}\mathbf{H}^\top$, so $\lambda_i(\mathbf{X}\mathbf{X}^\top)=\lambda_i(\mathbf{X}^\top\mathbf{X})$. In what follows, we remind the reader that $\mathbf{X}$ is a matrix whose rows are the tranposes of the column vectors $\mathbf{X}_i$, and $\mathbf{Y}$ is a d-dimensional vector that is independent from and has the same distribution as that of the $\mathbf{X}_i$. We observe that $(\mathbf{X}^\top\mathbf{X}-n\mathbb{E}\mathbf{Y}\mathbf{Y}^\top)_{ij}=\sum_{k=1}^n(\mathbf{X}_{ki}\mathbf{X}_{kj}-\mathbb{E}\mathbf{Y}_i\mathbf{Y}_j)$ is a sum of n independent zero-mean random variables, each contained in $[-1,1]$. Thus, Hoeffding's inequality yields, for all $i,j\in[d]$,

$$\Pr\left[|(\mathbf{X}^\top\mathbf{X})-n\mathbb{E}\mathbf{Y}\mathbf{Y}^\top|_{ij}\geq 2\sqrt{n\log n}\right]\leq\frac{2}{n^2}.$$

A union bound over all $i, j \in [d]$ implies that $\|\mathbf{X}^\top\mathbf{X} - n\mathbb{E}\mathbf{Y}\mathbf{Y}^\top\|_F^2 \leq 4d^2 n \log n$ with probability at least $1 - 2d^2/n^2$. Taking square roots and noting that the Frobenius norm is an upper bound on the spectral norm, we have that $\|\mathbf{X}^\top\mathbf{X} - n\mathbb{E}(\mathbf{Y}\mathbf{Y}^\top)\| \leq 2d\sqrt{n\log n}$ with probability at least $1 - 2d^2/n^2$, and Weyl's inequality yields that for all $1 \leq i \leq d$, $|\lambda_i(\mathbf{X}\mathbf{X}^\top) - n\lambda_i(\mathbb{E}(\mathbf{Y}\mathbf{Y}^\top)| \leq 2d\sqrt{n\log n}$ with probability at least $1 - 2d^2/n^2$. Of course, since the vector $\mathbf{Y}$ has the same distribution as any of the latent positions $\mathbf{X}_i$, we see that $\mathbb{E}(\mathbf{Y}\mathbf{Y}^\top) = \Delta$. By the reverse triangle inequality, for any $1 \leq i \leq d$, we have

$$\lambda_i(\mathbf{X}\mathbf{X}^\top) \geq \lambda_d(\mathbf{X}\mathbf{X}^\top) \geq |n\lambda_d(\Delta) - 2d\sqrt{n\log n}| = \Omega(n).$$

Multiplying through by ρ_n, we find that there exists some constant C so that for all n sufficiently large, $\lambda_d(\rho_n\mathbf{X}\mathbf{X}^\top) \geq n\rho_n\lambda_d(\Delta)$ with probability at least $1 - 2d^2/n^2$. ■

As described previously, to prove our central limit theorem, we require somewhat more precise control on certain residual terms, which we establish in the following key lemma. In the lemmas and proofs that follow, we frequently suppress the dependence of the sequence of graphs and embeddings on n.

Lemma 52 *Let $\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3$ be defined, as above, by*

$$\begin{aligned}
\mathbf{R}_1 &= \mathbf{U}_\mathbf{P}\mathbf{U}_\mathbf{P}^\top\mathbf{U}_\mathbf{A} - \mathbf{U}_\mathbf{P}\mathbf{W}^* \\
\mathbf{R}_2 &= \mathbf{W}^*\mathbf{S}_\mathbf{A}^{1/2} - \mathbf{S}_\mathbf{P}^{1/2}\mathbf{W}^* \\
\mathbf{R}_3 &= \mathbf{U}_\mathbf{A} - \mathbf{U}_\mathbf{P}\mathbf{U}_\mathbf{P}^\top\mathbf{U}_\mathbf{A} + \mathbf{R}_1 = \mathbf{U}_\mathbf{A} - \mathbf{U}_\mathbf{P}\mathbf{W}^*.
\end{aligned}$$

where, as before, $\mathbf{W}^*$ is the orthogonal transformation $\mathbf{W}^* = \mathbf{W}_1\mathbf{W}_2^\top$ with $\mathbf{W}_1\Sigma\mathbf{W}_2$ being the singular value decomposition of $\mathbf{U}_\mathbf{P}^\top\mathbf{U}_\mathbf{A}$. Then the following convergences in probability hold:

$$\sqrt{n}\Big[(\mathbf{A} - \mathbf{P})\mathbf{U}_\mathbf{P}(\mathbf{W}^*\mathbf{S}_\mathbf{A}^{-1/2} - \mathbf{S}_\mathbf{P}^{-1/2}\mathbf{W}^*)\Big]_h \xrightarrow{P} \mathbf{0}, \tag{53}$$

$$\sqrt{n}\Big[\mathbf{U}_\mathbf{P}\mathbf{U}_\mathbf{P}^\top(\mathbf{A} - \mathbf{P})\mathbf{U}_\mathbf{P}\mathbf{W}^*\mathbf{S}_\mathbf{A}^{-1/2}\Big]_h \xrightarrow{P} \mathbf{0}, \tag{54}$$

$$\sqrt{n}\Big[(\mathbf{I} - \mathbf{U}_\mathbf{P}\mathbf{U}_\mathbf{P}^\top)(\mathbf{A} - \mathbf{P})\mathbf{R}_3\mathbf{S}_\mathbf{A}^{-1/2}\Big]_h \xrightarrow{P} \mathbf{0}, \tag{55}$$

and with high probability,

$$\|\mathbf{R}_1\mathbf{S}_\mathbf{A}^{1/2} + \mathbf{U}_\mathbf{P}\mathbf{R}_2\|_F \leq \frac{C\log n}{n^{1/2}}.$$

Proof We begin by observing that

$$\|\mathbf{R}_1\mathbf{S}_\mathbf{A}^{1/2} + \mathbf{U}_\mathbf{P}\mathbf{R}_2\|_F \leq \|\mathbf{R}_1\|_F\|\mathbf{S}_\mathbf{A}^{1/2}\| + \|\mathbf{R}_2\|_F.$$

Proposition 48 and the trivial upper bound on the eigenvalues of $\mathbf{A}$ ensures that

$$\|\mathbf{R}_1\|_F\|\mathbf{S}_\mathbf{A}^{1/2}\| \leq \frac{C\log n}{n^{1/2}} \text{ w.h.p. },$$

Combining this with Eq. (49) in Lemma 49, we conclude that

$$\|\mathbf{R}_1\mathbf{S}_\mathbf{A}^{1/2} + \mathbf{U}_\mathbf{P}\mathbf{R}_2\|_F \leq \frac{C\log n}{n^{1/2}} \text{ w.h.p.}$$

We will establish (53), (54) and (55) order. To see (53), observe that

$$\sqrt{n}\|(\mathbf{A} - \mathbf{P})\mathbf{U}_\mathbf{P}(\mathbf{W}^*\mathbf{S}_\mathbf{A}^{-1/2} - \mathbf{S}_\mathbf{P}^{-1/2}\mathbf{W}^*)\|_F \leq \sqrt{n}\|(\mathbf{A} - \mathbf{P})\mathbf{U}_\mathbf{P}\|\|\mathbf{W}^*\mathbf{S}_\mathbf{A}^{-1/2} - \mathbf{S}_\mathbf{P}^{-1/2}\mathbf{W}^*\|_F,$$

and Lemma 49 imply that with high probability

$$\sqrt{n}\|(\mathbf{A} - \mathbf{P})\mathbf{U}_\mathbf{P}(\mathbf{W}^*\mathbf{S}_\mathbf{A}^{-1/2} - \mathbf{S}_\mathbf{P}^{-1/2}\mathbf{W}^*)\|_F \leq \frac{C\log n}{\sqrt{n}},$$

which goes to 0 as $n \to \infty$.

To show the convergence in (54), we recall that $\mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{1/2}\mathbf{W} = \mathbf{X}$ for some orthogonal matrix $\mathbf{W}$ and observe that since the rows of the latent position matrix $\mathbf{X}$ are necessarily bounded in Euclidean norm by 1, and since the top d eigenvalues of $\mathbf{P}$ are of order n (recall Lemma 51), it follows that

$$\|\mathbf{U}_\mathbf{P}\|_{2\to\infty} \leq Cn^{-1/2} \text{ w.h.p.} \tag{56}$$

Next, (50) and Lemma 51 imply that

$$\begin{aligned}
\|(\mathbf{U}_\mathbf{P}\mathbf{U}_\mathbf{P}^\top(\mathbf{A} - \mathbf{P})\mathbf{U}_\mathbf{P}\mathbf{W}^*\mathbf{S}_\mathbf{A}^{-1/2})_h\| &\leq \|\mathbf{U}_\mathbf{P}\|_{2\to\infty}\|\mathbf{U}_\mathbf{P}^\top(\mathbf{A} - \mathbf{P})\mathbf{U}_\mathbf{P}\|\|\mathbf{S}_\mathbf{A}^{-1/2}\| \\
&\leq \frac{C\log n}{n} \text{ w.h.p.},
\end{aligned}$$

which implies (54). Finally, to establish (55), we must bound the Euclidean norm of the vector

$$\Big[(\mathbf{I} - \mathbf{U}_\mathbf{P}\mathbf{U}_\mathbf{P}^\top)(\mathbf{A} - \mathbf{P})\mathbf{R}_3\mathbf{S}_\mathbf{A}^{-1/2}\Big]_h, \tag{57}$$

where, as defined above, $\mathbf{R}_3 = \mathbf{U}_\mathbf{A} - \mathbf{U}_\mathbf{P}\mathbf{W}^*$. Let $\mathbf{B}_1$ and $\mathbf{B}_2$ be defined as follows:

$$\begin{aligned}
\mathbf{B}_1 &= (\mathbf{I} - \mathbf{U}_\mathbf{P}\mathbf{U}_\mathbf{P}^\top)(\mathbf{A} - \mathbf{P})(\mathbf{I} - \mathbf{U}_\mathbf{P}\mathbf{U}_\mathbf{P}^\top)\mathbf{U}_\mathbf{A}\mathbf{S}_\mathbf{A}^{-1/2} \\
\mathbf{B}_2 &= (\mathbf{I} - \mathbf{U}_\mathbf{P}\mathbf{U}_\mathbf{P}^\top)(\mathbf{A} - \mathbf{P})\mathbf{U}_\mathbf{P}(\mathbf{U}_\mathbf{P}^\top\mathbf{U}_\mathbf{A} - \mathbf{W}^*)\mathbf{S}_\mathbf{A}^{-1/2}
\end{aligned} \tag{58}$$

Recalling that $\mathbf{R}_3 = \mathbf{U}_\mathbf{A} - \mathbf{U}_\mathbf{P}\mathbf{W}^*$, we have

$$\begin{aligned}
(\mathbf{I} - \mathbf{U}_\mathbf{P}\mathbf{U}_\mathbf{P}^\top)(\mathbf{A} - \mathbf{P})\mathbf{R}_3\mathbf{S}_\mathbf{A}^{-1/2} &= (\mathbf{I} - \mathbf{U}_\mathbf{P}\mathbf{U}_\mathbf{P}^\top)(\mathbf{A} - \mathbf{P})(\mathbf{U}_\mathbf{A} - \mathbf{U}_\mathbf{P}\mathbf{U}_\mathbf{P}^\top\mathbf{U}_\mathbf{A})\mathbf{S}_\mathbf{A}^{-1/2} \\
&\quad + (\mathbf{I} - \mathbf{U}_\mathbf{P}\mathbf{U}_\mathbf{P}^\top)(\mathbf{A} - \mathbf{P})(\mathbf{U}_\mathbf{P}\mathbf{U}_\mathbf{P}^\top\mathbf{U}_\mathbf{A} - \mathbf{U}_\mathbf{P}\mathbf{W}^*)\mathbf{S}_\mathbf{A}^{-1/2} \\
&= \mathbf{B}_1 + \mathbf{B}_2.
\end{aligned}$$

We will bound the Euclidean norm of the h-th row of each of these two matrices on the right-hand side, from which a triangle inequality will yield our desired bound on the quantity in Equation (57). Recall that we use C to denote a positive constant, independent of n and m, which may change from line to line.

Let us first consider $\mathbf{B}_2 = (\mathbf{I} - \mathbf{U_P}\mathbf{U_P}^\top)(\mathbf{A} - \mathbf{P})\mathbf{U_P}(\mathbf{U_P}^\top\mathbf{U_A} - \mathbf{W}^*)\mathbf{S_A}^{-1/2}$. We have

$$\|\mathbf{B}_2\|_F \leq \|(\mathbf{I} - \mathbf{U_P}\mathbf{U_P}^\top)(\mathbf{A} - \mathbf{P})\mathbf{U_P}\|\|\mathbf{U_P}^\top\mathbf{U_A} - \mathbf{W}^*\|_F\|\mathbf{S_A}^{-1/2}\|.$$

By submultiplicativity of the spectral norm and Theorem20, $\|(\mathbf{I} - \mathbf{U_P}\mathbf{U_P}^\top)(\mathbf{A} - \mathbf{P})\mathbf{U_P}\| \leq Cn^{1/2}\log^{1/2} n$ with high probability (and indeed, under our degree assumptions and Theorem 21, the $\log n$ factor can be dropped). From Prop. 48 and Lemma 51, respectively, we have with high probability

$$\|\mathbf{U_P}^\top\mathbf{U_A} - \mathbf{W}^*\|_F \leq Cn^{-1}\log n \quad \text{and} \quad \|\mathbf{S_A}^{-1/2}\| \leq Cn^{-1/2}$$

Thus, we deduce that with high probability,

$$\|\mathbf{B}_2\|_F \leq \frac{C\log^{3/2} n}{n} \tag{59}$$

from which it follows that $\|\sqrt{n}\mathbf{B}_2\|_F \xrightarrow{P} 0$, and hence $\|\sqrt{n}(\mathbf{B}_2)_h\| \xrightarrow{P} 0$.

Turning our attention to $\mathbf{B}_1$, and recalling that $\mathbf{U_A}^\top\mathbf{U_A} = \mathbf{I}$, we note that

$$\begin{aligned}
\|(\mathbf{B}_1)_h\| &= \left\|\left[(\mathbf{I} - \mathbf{U_P}\mathbf{U_P}^\top)(\mathbf{A} - \mathbf{P})(\mathbf{I} - \mathbf{U_P}\mathbf{U_P}^\top)\mathbf{U_A}\mathbf{S_A}^{-1/2}\right]_h\right\| \\
&= \left\|\left[(\mathbf{I} - \mathbf{U_P}\mathbf{U_P}^\top)(\mathbf{A} - \mathbf{P})(\mathbf{I} - \mathbf{U_P}\mathbf{U_P}^\top)\mathbf{U_A}\mathbf{U_A}^\top\mathbf{U_A}\mathbf{S_A}^{-1/2}\right]_h\right\| \\
&\leq \left\|\mathbf{U_A}\mathbf{S_A}^{-1/2}\right\|\left\|\left[(\mathbf{I} - \mathbf{U_P}\mathbf{U_P}^\top)(\mathbf{A} - \mathbf{P})(\mathbf{I} - \mathbf{U_P}\mathbf{U_P}^\top)\mathbf{U_A}\mathbf{U_A}^\top\right]_h\right\|.
\end{aligned}$$

Let $\epsilon > 0$ be a constant. We will show that

$$\lim_{n\to\infty} \Pr\left[\|\sqrt{n}(\mathbf{B}_1)_h\| > \epsilon\right] = 0. \tag{60}$$

For ease of notation, define

$$\mathbf{E}_1 = (\mathbf{I} - \mathbf{U_P}\mathbf{U_P}^\top)(\mathbf{A} - \mathbf{P})(\mathbf{I} - \mathbf{U_P}\mathbf{U_P}^\top)\mathbf{U_A}\mathbf{U_A}^\top.$$

We will show that

$$\lim_{n\to\infty} \Pr\left[\sqrt{n}\,\|[\mathbf{E}_1]_h\| > n^{1/4}\right] = 0, \tag{61}$$

which will imply (60) since, by Lemma 51, $\|\mathbf{U_A}\mathbf{S_A}^{-1/2}\| \leq Cn^{-1/2}$ with high probability. Let $\mathbf{Q} \in \mathbb{R}^{n\times n}$ be any permutation matrix. We observe that

$$\mathbf{Q}\mathbf{U_P}\mathbf{U_P}^\top\mathbf{Q}^\top\mathbf{Q}\mathbf{P}\mathbf{Q}^\top = \mathbf{Q}\mathbf{P}\mathbf{Q}^\top,$$

and thus $\mathbf{Q}\mathbf{U_P}\mathbf{U_P}^\top\mathbf{Q}^\top$ is a projection matrix for $\mathbf{Q}\mathbf{P}\mathbf{Q}^\top$ if and only if $\mathbf{U_P}\mathbf{U_P}^\top$ is a projection matrix for $\mathbf{P}$. A similar argument applies to the matrix $\mathbf{U_A}\mathbf{U_A}^\top$. Combining this with the exchangeability structure of the matrix $\mathbf{A} - \mathbf{P}$, it follows that the Frobenius norms of the rows of $\mathbf{E}_1$ are equidistributed. This row-exchangeability for $\mathbf{E}_1$ implies that $n\mathbb{E}\|(\mathbf{E}_1)_h\|^2 = \|\mathbf{E}_1\|_F^2$. Applying Markov's inequality,

$$\begin{aligned}
\Pr\left[\left\|\sqrt{n}\,[\mathbf{E}_1]_h\right\| > t\right] &\leq \frac{n\mathbb{E}\left\|\left[(\mathbf{I} - \mathbf{U_P}\mathbf{U_P}^\top)(\mathbf{A} - \mathbf{P})(\mathbf{I} - \mathbf{U_P}\mathbf{U_P}^\top)\mathbf{U_A}\mathbf{U_A}^\top\right]_h\right\|^2}{t^2} \\
&= \frac{\mathbb{E}\left\|(\mathbf{I} - \mathbf{U_P}\mathbf{U_P}^\top)(\mathbf{A} - \mathbf{P})(\mathbf{I} - \mathbf{U_P}\mathbf{U_P}^\top)\mathbf{U_A}\mathbf{U_A}^\top\right\|_F^2}{t^2}.
\end{aligned} \tag{62}$$

We will proceed by showing that with high probability,

$$\left\|(\mathbf{I} - \mathbf{U_P}\mathbf{U_P}^\top)(\mathbf{A} - \mathbf{P})(\mathbf{I} - \mathbf{U_P}\mathbf{U_P}^\top)\mathbf{U_A}\mathbf{U_A}^\top\right\|_F \leq C\log n, \tag{63}$$

whence choosing $t = n^{1/4}$ in (62) yields that

$$\lim_{n\to\infty} \Pr\left[\left\|\sqrt{n}\left[(\mathbf{I} - \mathbf{U_P}\mathbf{U_P}^\top)(\mathbf{A} - \mathbf{P})(\mathbf{I} - \mathbf{U_P}\mathbf{U_P}^\top)\mathbf{U_A}\mathbf{U_A}^\top\right]_h\right\| > n^{1/4}\right] = 0,$$

and (60) will follow. We have

$$\left\|(\mathbf{I} - \mathbf{U_P}\mathbf{U_P}^\top)(\mathbf{A} - \mathbf{P})(\mathbf{I} - \mathbf{U_P}\mathbf{U_P}^\top)\mathbf{U_A}\mathbf{U_A}^\top\right\|_F \leq \|\mathbf{A} - \mathbf{P}\|\|\mathbf{U_A} - \mathbf{U_P}\mathbf{U_P}^\top\mathbf{U_A}\|_F\|\mathbf{U_A}\|$$

Theorem 21 and Lemma 51 implies that the first term in this product is at most $Cn^{1/2}$ high probability, and the final term in this product is, trivially, at most 1. To bound the second term, we will follow reasoning similar to that in Lemma 49, combined with the Davis-Kahan theorem. The Davis-Kahan Theorem implies that for a suitable constant $C > 0$,

$$\|\mathbf{U_A}\mathbf{U_A}^\top - \mathbf{U_P}\mathbf{U_P}^\top\| \leq \frac{C\|\mathbf{A} - \mathbf{P}\|}{\lambda_d(\mathbf{P})}.$$

By Theorem 2 in Yu et al. (2015), there exists orthonormal $\mathbf{W} \in \mathbb{R}^{d\times d}$ such that

$$\|\mathbf{U_A} - \mathbf{U_P}\mathbf{W}\|_F \leq C\|\mathbf{U_A}\mathbf{U_A}^\top - \mathbf{U_P}\mathbf{U_P}^\top\|_F.$$

We observe further that the multivariate linear least squares problem

$$\min_{\mathbf{T}\in\mathbb{R}^{d\times d}} \|\mathbf{U_A} - \mathbf{U_P}\mathbf{T}\|_F^2$$

is solved by $\mathbf{T} = \mathbf{U_P}^\top\mathbf{U_A}$. Combining all of the above, we find that

$$\begin{aligned}
\|\mathbf{U_A} - \mathbf{U_P}\mathbf{U_P}^\top\mathbf{U_A}\|_F^2 \leq \|\mathbf{U_A} - \mathbf{U_P}\mathbf{W}\|_F^2 &\leq C\|\mathbf{U_A}\mathbf{U_A}^\top - \mathbf{U_P}\mathbf{U_P}^\top\|_F^2 \\
&\leq C\|\mathbf{U_A}\mathbf{U_A}^\top - \mathbf{U_P}\mathbf{U_P}^\top\|^2 \leq \left(\frac{C\|\mathbf{A} - \mathbf{P}\|}{\lambda_d(\mathbf{P})}\right)^2 \leq \frac{C}{n} \quad \text{w.h.p.}
\end{aligned}$$

We conclude that

$$\begin{aligned}
\left\|(\mathbf{I} - \mathbf{U_P}\mathbf{U_P}^\top)(\mathbf{A} - \mathbf{P})(\mathbf{I} - \mathbf{U_P}\mathbf{U_P}^\top)\mathbf{U_A}\mathbf{U_A}^\top\right\|_F &\leq C\|\mathbf{A} - \mathbf{P}\|\|\mathbf{U_A} - \mathbf{U_P}\mathbf{U_P}^\top\mathbf{U_A}\|_F\|\mathbf{U_A}\|_S \\
&\leq C\log n \text{ w.h.p. ,}
\end{aligned}$$

which implies (63), as required, and thus the convergence in (55) is established, completing the proof. ■

Next, recall the inherent nonidentifiability in the RDPG model: suppose the "true" latent positions are some matrix $\mathbf{X}$. Then $\mathbf{X} = \mathbf{U_P}\mathbf{S_P}^{1/2}\mathbf{W}$ for some suitably-chosen orthogonal matrix $\mathbf{W}$. We now consider the asymptotic distribution of

$$n^{1/2}\mathbf{W}_n^\top\left[(\mathbf{A} - \mathbf{P})\mathbf{U_P}\mathbf{S_P}^{-1/2}\right]_h$$

conditional on $\mathbf{X}_i = \mathbf{x}_i$. Because we can write suitable terms of $\mathbf{A} - \mathbf{P}$ as the sum of independent random variables, we can invoke the Lindeberg-Feller Central Limit Theorem to establish the asymptotic normality of

$$n^{1/2}\mathbf{W}_n^\top \Big[(\mathbf{A}-\mathbf{P})\mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{-1/2}\Big]_h$$

as follows.

Lemma 53 *Fix some $i \in [n]$. Conditional on $\mathbf{X}_i = \mathbf{x}_i \in \mathbb{R}^d$, there exists a sequence of d-by-d orthogonal matrices $\{\mathbf{W}_n\}$ such that*

$$n^{1/2}\mathbf{W}_n^\top \Big[(\mathbf{A}-\mathbf{P})\mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{-1/2}\Big]_h \xrightarrow{\mathcal{L}} \mathcal{N}(0, \mathbf{\Sigma}(\mathbf{x}_i)),$$

where $\mathbf{\Sigma}(\mathbf{x}_i) \in \mathbb{R}^{d\times d}$ is a covariance matrix that depends on $\mathbf{x}_i$.

Proof For each n, choose orthogonal $\mathbf{W}_n \in \mathbb{R}^{d\times d}$ so that $\mathbf{X} = \mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{1/2}\mathbf{W}_n$. At least one such $\mathbf{W}_n$ exists for each value of n, since, as discussed previously, the true latent positions $\mathbf{X}$ are specified only up to some orthogonal transformation. We then have

$$\begin{aligned}
n^{1/2}\mathbf{W}_n^\top \Big[(\mathbf{A}-\mathbf{P})\mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{-1/2}\Big]_i &= n^{1/2}\mathbf{W}_n^\top\mathbf{S}_\mathbf{P}^{-1}\mathbf{W}_n\left[\mathbf{A}\mathbf{X}-\mathbf{P}\mathbf{X}\right]_i \\
&= n^{1/2}\mathbf{W}_n^\top\mathbf{S}_\mathbf{P}^{-1}\mathbf{W}_n\left(\sum_{j=1}^n (\mathbf{A}_{ij}-\mathbf{P}_{ij})\,\mathbf{X}_j\right) \\
&= n^{1/2}\mathbf{W}_n^\top\mathbf{S}_\mathbf{P}^{-1}\mathbf{W}_n\left(\sum_{j\neq i} (\mathbf{A}_{ij}-\mathbf{P}_{ij})\,\mathbf{X}_j\right) - n^{1/2}\mathbf{W}_n^\top\mathbf{S}_\mathbf{P}^{-1}\mathbf{W}_n\mathbf{P}_{ii}\mathbf{X}_i \\
&= \big(n\mathbf{W}_n^\top\mathbf{S}_\mathbf{P}^{-1}\mathbf{W}_n\big)\left[n^{-1/2}\sum_{j\neq i} (\mathbf{A}_{ij}-\mathbf{P}_{ij})\,\mathbf{X}_j\right] - n\mathbf{W}_n^\top\mathbf{S}_\mathbf{P}^{-1}\mathbf{W}_n\frac{\mathbf{P}_{ii}\mathbf{X}_i}{n^{1/2}}.
\end{aligned}$$

Conditioning on $\mathbf{X}_i = \mathbf{x}_i \in \mathbb{R}^d$, we first observe that $\frac{\mathbf{P}_{ii}}{n^{1/2}}\mathbf{X}_i = \frac{\mathbf{x}_i^\top\mathbf{x}_i}{n^{1/2}}x_i \to 0$ almost surely. Furthermore,

$$n^{-1/2}\sum_{j\neq i}(\mathbf{A}_{ij}-\mathbf{P}_{ij})\,\mathbf{X}_j = n^{-1/2}\sum_{j\neq i}\big(\mathbf{A}_{ij}-\mathbf{X}_j^\top\mathbf{x}_i\big)\mathbf{X}_j$$

is a scaled sum of $n-1$ independent 0-mean random variables, each with covariance matrix given by

$$\tilde{\mathbf{\Sigma}}(\mathbf{x}_i) = \mathbb{E}\left[\big(\mathbf{x}_i^\top\mathbf{X}_j - (\mathbf{x}_i^\top\mathbf{X}_j)^2\big)\,\mathbf{X}_j\mathbf{X}_j^\top\right].$$

The multivariate central limit theorem thus implies that

$$n^{-1/2}\sum_{j\neq i}\big(\mathbf{A}_{ij}-\mathbf{X}_j\mathbf{x}_i^\top)\mathbf{X}_j\big) \xrightarrow{\mathcal{L}} \mathcal{N}(\mathbf{0}, \tilde{\mathbf{\Sigma}}(\mathbf{x}_i)). \tag{64}$$

Finally, by the strong law of large numbers,

$$n^{-1}\mathbf{W}_n^\top\mathbf{S}_{\mathbf{P}_n}\mathbf{W}_n = \frac{1}{n}\mathbf{X}^\top\mathbf{X} \to \mathbf{\Delta} \text{ a.s.}$$

and thus $(n\mathbf{W}_n^\top\mathbf{S}_{\mathbf{P}_n}^{-1}\mathbf{W}_n) \to \mathbf{\Delta}^{-1}$ almost surely. Combining this fact with (64), the multivariate version of Slutsky's theorem yields

$$n^{1/2}\mathbf{W}_n^\top\Big[(\mathbf{A}-\mathbf{P})\mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{-1/2}\Big]_h \xrightarrow{\mathcal{L}} \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}(\mathbf{x}_i))$$

where $\mathbf{\Sigma}(\mathbf{x}_i) = \mathbf{\Delta}^{-1}\tilde{\mathbf{\Sigma}}(\mathbf{x}_i)\mathbf{\Delta}^{-1}$. Integrating over the possible values of $\mathbf{x}_i$ with respect to distribution F completes the proof. ■

Lemmas 52 and 53 are the main ingredients in the proof of Theorem 27, whose proof now follows easily:

Proof [Proof of Theorem 27]. We start with the following decomposition that was originally used in the proof of Theorem 50.

$$\begin{aligned}
\sqrt{n}\left(\mathbf{U}_\mathbf{A}\mathbf{S}_\mathbf{A}^{1/2} - \mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{1/2}\mathbf{W}^*\right) &= \sqrt{n}(\mathbf{A}-\mathbf{P})\mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{-1/2}\mathbf{W}^* + \sqrt{n}(\mathbf{A}-\mathbf{P})\mathbf{U}_\mathbf{P}(\mathbf{W}^*\mathbf{S}_\mathbf{A}^{-1/2} - \mathbf{S}_\mathbf{P}^{-1/2}\mathbf{W}^*) \\
&\quad - \sqrt{n}\mathbf{U}_\mathbf{P}\mathbf{U}_\mathbf{P}^\top(\mathbf{A}-\mathbf{P})\mathbf{U}_\mathbf{P}\mathbf{W}^*\mathbf{S}_\mathbf{A}^{-1/2} \\
&\quad + \sqrt{n}(\mathbf{I}-\mathbf{U}_\mathbf{P}\mathbf{U}_\mathbf{P}^\top)(\mathbf{A}-\mathbf{P})\mathbf{R}_3\mathbf{S}_\mathbf{A}^{-1/2} + \sqrt{n}\mathbf{R}_1\mathbf{S}_\mathbf{A}^{1/2} + \sqrt{n}\mathbf{U}_\mathbf{P}\mathbf{R}_2.
\end{aligned} \tag{65}$$

Now given any index i, Lemma 52 can be used to bound the Euclidean norms of the i-th rows of

$$(\mathbf{A}-\mathbf{P})\mathbf{U}_\mathbf{P}(\mathbf{W}^*\mathbf{S}_\mathbf{A}^{-1/2} - \mathbf{S}_\mathbf{P}^{-1/2}\mathbf{W}^*),$$
$$\mathbf{U}_\mathbf{P}\mathbf{U}_\mathbf{P}^\top(\mathbf{A}-\mathbf{P})\mathbf{U}_\mathbf{P}\mathbf{W}^*\mathbf{S}_\mathbf{A}^{-1/2}$$

and

$$(\mathbf{I}-\mathbf{U}_\mathbf{P}\mathbf{U}_\mathbf{P}^\top)(\mathbf{A}-\mathbf{P})\mathbf{R}_3\mathbf{S}_\mathbf{A}^{-1/2}.$$

The Euclidean norms of the i-th row of $\mathbf{R}_1\mathbf{S}_\mathbf{A}^{1/2}$ and $\mathbf{U}_\mathbf{P}\mathbf{R}_2$ can be bounded from above by the bounds on $\|\mathbf{R}_1\mathbf{S}_\mathbf{A}^{1/2}\|_{2\to\infty}$ and $\|\mathbf{U}_\mathbf{P}\mathbf{R}_2\|_{2\to\infty}$. Since

$$\|\mathbf{U}_\mathbf{P}\|_{2\to\infty} = O(n^{-1/2})$$

by Eq. (56), we conclude that

$$\sqrt{n}\|\mathbf{R}_1\mathbf{S}_\mathbf{A}^{1/2}\|_{2\to\infty} = O(n^{-1/2}\log^{1/2} n)$$

and

$$\sqrt{n}\|\mathbf{U}_\mathbf{P}\mathbf{R}_2\|_{2\to\infty} = O(n^{-1/2}\log^{1/2} n)$$

Therefore, for any fixed index i, we have

$$\sqrt{n}\left(\mathbf{U}_\mathbf{A}\mathbf{S}_\mathbf{A}^{1/2} - \mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{1/2}\mathbf{W}^*\right)_i = \sqrt{n}((\mathbf{A}-\mathbf{P})\mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{-1/2}\mathbf{W}^*)_i + O(n^{-1/2}\log^{1/2} n)$$

with high probability. Since $\mathbf{X} = \mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}\mathbf{W}$, we can rewrite the above expression as

$$\sqrt{n}\left(\mathbf{U}_\mathbf{A}\mathbf{S}_\mathbf{A}^{1/2}(\mathbf{W}^*)^\top\mathbf{W} - \mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{1/2}\mathbf{W}\right)_i = \sqrt{n}((\mathbf{A}-\mathbf{P})\mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{-1/2}\mathbf{W})_i + O(n^{-1/2}\log^{1/2} n).$$

Lemma 53 then establishes the asymptotic normality of $\sqrt{n}((\mathbf{A}-\mathbf{P})\mathbf{U}_\mathbf{P}\mathbf{S}_\mathbf{P}^{-1/2}\mathbf{W})_i$ as desired. ■

We now turn out attention a brief sketch of the proof of the central limit theorem for the Laplacian spectral embedding.

A.3 Sketch of proof of Theorem 29

We present in this subsection a sketch of the main ideas in the proof of Theorem 29; detailed proofs are given in Tang and Priebe (2018). We first introduce some additional notation. For $(\mathbf{X}_n, \mathbf{A}_n) \sim \text{RDPG}(F)$, let $\mathbf{T}_n = \text{diag}(\mathbf{P}_n\mathbf{1})$ be the $n \times n$ diagonal matrices whose diagonal entries are the *expected* vertex degrees. Then defining $\tilde{\mathbf{X}}_n = \mathbf{T}_n^{-1/2}\mathbf{X}_n$, and noting that $\tilde{\mathbf{X}}_n\tilde{\mathbf{X}}_n^\top = \mathcal{L}(\mathbf{P}_n) = \mathbf{T}_n^{-1/2}\mathbf{P}_n\mathbf{T}_n^{-1/2}$, Theorem 29 depends on showing that there exists an orthogonal matrix $\mathbf{W}_n$ such that

$$\check{\mathbf{X}}_n\mathbf{W}_n - \tilde{\mathbf{X}}_n = \mathbf{T}_n^{-1/2}(\mathbf{A}_n - \mathbf{P}_n)\mathbf{T}_n^{-1/2}\tilde{\mathbf{X}}_n(\tilde{\mathbf{X}}_n^\top\tilde{\mathbf{X}}_n)^{-1} + \tfrac{1}{2}(\mathbf{I} - \mathbf{D}_n\mathbf{T}_n^{-1})\tilde{\mathbf{X}}_n + \mathbf{R}_n \tag{66}$$

where $\|\mathbf{R}_n\|_F = O(n^{-1})$ with high probability. The motivation behind Eq. (66) is as follows. Given $\tilde{\mathbf{X}}_n$, the entries of the right hand side of Eq. (66), except for the term $\mathbf{R}_n$, can be expressed explicitly in terms of linear combinations of the entries $a_{ij} - p_{ij}$ of $\mathbf{A}_n - \mathbf{P}_n$. This is in contrast with the left hand side of Eq. (66), which depends on the quantities $\mathbf{U}_{\mathcal{L}(\mathbf{A}_n)}$ and $\mathbf{S}_{\mathcal{L}(\mathbf{A}_n)}$ (recall Definition 18). Since the quantities $\mathbf{U}_{\mathcal{L}(\mathbf{A}_n)}$ and $\mathbf{S}_{\mathcal{L}(\mathbf{A}_n)}$ can not be expressed explicitly in terms of the entries of $\mathbf{A}_n$ and $\mathbf{P}_n$, we conclude that the right hand side of Eq. (66) is simpler to analyze.

Once Eq. (66) is established, we can derive Theorem 29 as follows. Let ξ_i denote the i-th row of $n(\check{\mathbf{X}}_n\mathbf{W}_n - \tilde{\mathbf{X}}_n)$ and let r_i denote the i-th row of $\mathbf{R}_n$. Eq. (66) then implies

$$\begin{aligned}
\xi_i &= (\tilde{\mathbf{X}}_n^\top\tilde{\mathbf{X}}_n)^{-1}\frac{n}{\sqrt{t_i}}\Big(\sum_j \frac{a_{ij} - p_{ij}}{\sqrt{t_j}}(\tilde{\mathbf{X}}_n)_j\Big) + \frac{n(t_i - d_i)}{2t_i}(\tilde{\mathbf{X}}_n)_i + nr_i \\
&= (\tilde{\mathbf{X}}_n^\top\tilde{\mathbf{X}}_n)^{-1}\frac{\sqrt{n}}{\sqrt{t_i}}\Big(\sum_j \frac{\sqrt{n\rho_n}(a_{ij} - p_{ij})(\mathbf{X}_n)_j}{t_j}\Big) - \frac{n(\mathbf{X}_n)_i}{2t_i^{3/2}}\sum_j(a_{ij} - p_{ij}) + nr_i \\
&= \frac{\sqrt{n}}{\sqrt{t_i}}\sum_j \frac{(a_{ij} - p_{ij})}{\sqrt{n}}\Big(\frac{(\tilde{\mathbf{X}}_n^\top\tilde{\mathbf{X}}_n)^{-1}(\mathbf{X}_n)_j}{t_j/n} - \frac{(\mathbf{X}_n)_i}{2t_i/n}\Big) + nr_i
\end{aligned}$$

where a_{ij} and p_{ij} are the ij-th entries of $\mathbf{A}$ and $\mathbf{P}$, respectively, and t_i is the i-th diagonal entry of $\mathbf{T}_n$. We can then show that $nr_i \xrightarrow{\text{d}} 0$. Indeed, there are n rows in $\mathbf{R}_n$ and $\|\mathbf{R}_n\|_F = O(n^{-1})$; hence, on average, for each index i, $\|r_i\|^2 = O(n^{-3})$ with high probability (a more precise argument similar to that used in proving Lemma 52 is needed to establish this rigorously). Furthermore,

$$t_i/n = \sum_j (\mathbf{X}_n)_i^\top(\mathbf{X}_n)_j/n \xrightarrow{\text{a.s.}} (\mathbf{X}_n)_i^\top\boldsymbol{\mu}$$

as $n \to \infty$. Finally,

$$\tilde{\mathbf{X}}_n^\top\tilde{\mathbf{X}}_n = \sum_i\big((\mathbf{X}_n)_i(\mathbf{X}_n)_i^\top/(\sum_j (\mathbf{X}_n)_i^\top(\mathbf{X}_n)_j)\big),$$

and this can be shown to converge to $\tilde{\Delta} = \mathbb{E}\big[\frac{\mathbf{X}_1\mathbf{X}_1^\top}{\mathbf{X}_1^\top\boldsymbol{\mu}}\big]$ as $n \to \infty$. We therefore have, after additional algebraic manipulations, that

$$\begin{aligned}
\xi_i &= \frac{\sqrt{n}}{\sqrt{t_i}}\sum_j \frac{(a_{ij} - p_{ij})}{\sqrt{n}}\Big(\frac{\tilde{\Delta}^{-1}(\mathbf{X}_n)_j}{(\mathbf{X}_n)_j^\top\boldsymbol{\mu}} - \frac{(\mathbf{X}_n)_i}{2(\mathbf{X}_n)_i^\top\boldsymbol{\mu}}\Big) + o(1) \\
&= \frac{\sqrt{n}}{\sqrt{t_i}}\sum_j \frac{(a_{ij} - (\mathbf{X}_n)_i^\top(\mathbf{X}_n)_j)}{\sqrt{n}}\Big(\frac{\tilde{\Delta}^{-1}(\mathbf{X}_n)_j}{(\mathbf{X}_n)_j^\top\boldsymbol{\mu}} - \frac{(\mathbf{X}_n)_i}{2(\mathbf{X}_n)_i^\top\boldsymbol{\mu}}\Big) + o(1)
\end{aligned}$$

with high probability. Conditioning on $(\mathbf{X}_n)_i = \boldsymbol{x}$, the above expression for ξ_i is roughly a sum of independent and identically distributed mean 0 random variables. The multivariate central limit theorem can then be applied to the above expression for ξ_i, thereby yielding Theorem 29.

We now sketch the derivation of Eq. (66). For simplicity of notation, we shall ignore the subscript n in the matrices $\mathbf{A}_n$, $\mathbf{X}_n$, $\mathbf{P}_n$ and related matrices. First, consider the following expression.

$$\begin{aligned}
\mathbf{U}_{\mathcal{L}(\mathbf{A})}\mathbf{S}_{\mathcal{L}(\mathbf{A})}^{1/2} - \mathbf{U}_{\mathcal{L}(\mathbf{P})}\mathbf{S}_{\mathcal{L}(\mathbf{P})}^{1/2}\mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top\mathbf{U}_{\mathcal{L}(\mathbf{A})} &= \mathcal{L}(\mathbf{A})\tilde{\mathbf{U}}_{\mathcal{L}(\mathbf{A})}\tilde{\mathbf{S}}_{\mathcal{L}(\mathbf{A})}^{-1/2} - \mathcal{L}(\mathbf{P})\tilde{\mathbf{U}}_{\mathcal{L}(\mathbf{P})}\tilde{\mathbf{S}}_{\mathcal{L}(\mathbf{P})}^{-1/2}\tilde{\mathbf{U}}_{\mathcal{L}(\mathbf{P})}^\top\tilde{\mathbf{U}}_{\mathcal{L}(\mathbf{A})} \\
&= \mathcal{L}(\mathbf{A})\mathbf{U}_{\mathcal{L}(\mathbf{A})}\mathbf{U}_{\mathcal{L}(\mathbf{A})}^\top\mathbf{U}_{\mathcal{L}(\mathbf{A})}\mathbf{S}_{\mathcal{L}(\mathbf{A})}^{-1/2} - \mathcal{L}(\mathbf{P})\mathbf{U}_{\mathcal{L}(\mathbf{P})}\mathbf{S}_{\mathcal{L}(\mathbf{P})}^{-1/2}\mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top\mathbf{U}_{\mathcal{L}(\mathbf{A})}
\end{aligned} \tag{67}$$

Now $\mathcal{L}(\mathbf{A})$ is concentrated around $\mathcal{L}(\mathbf{P})$: namely, in the current setting,

$$\|\mathcal{L}(\mathbf{A}) - \mathcal{L}(\mathbf{P})\| = O(n^{-1/2})$$

with high probability (see Theorem 2 in Lu and Peng, 2013). Since $\|\mathcal{L}(\mathbf{P})\| = \Theta(1)$ and the non-zero eigenvalues of $\mathcal{L}(\mathbf{P})$ are all of order $\Theta(1)$, this again implies, by the Davis-Kahan theorem, that the eigenspace spanned by the d largest eigenvalues of $\mathcal{L}(\mathbf{A})$ is "close" to that spanned by the d largest eigenvalues of $\mathcal{L}(\mathbf{P})$. More precisely, $\|\mathbf{U}_{\mathcal{L}(\mathbf{A})}\mathbf{U}_{\mathcal{L}(\mathbf{A})} - \mathbf{U}_{\mathcal{L}(\mathbf{P})}\mathbf{U}_{\mathcal{L}(\mathbf{P})}\| = O(n^{-1/2})$ with high probability, and

$$\begin{aligned}
\mathbf{U}_{\mathcal{L}(\mathbf{A})}\mathbf{S}_{\mathcal{L}(\mathbf{A})}^{1/2} - \mathbf{U}_{\mathcal{L}(\mathbf{P})}\mathbf{S}_{\mathcal{L}(\mathbf{P})}^{1/2}\mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top\mathbf{U}_{\mathcal{L}(\mathbf{A})} &= \mathcal{L}(\mathbf{A})\mathbf{U}_{\mathcal{L}(\mathbf{P})}\mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top\mathbf{U}_{\mathcal{L}(\mathbf{A})}\mathbf{S}_{\mathcal{L}(\mathbf{A})}^{-1/2} \\
&\quad - \mathcal{L}(\mathbf{P})\mathbf{U}_{\mathcal{L}(\mathbf{P})}\mathbf{S}_{\mathcal{L}(\mathbf{P})}^{-1/2}\mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top\mathbf{U}_{\mathcal{L}(\mathbf{A})} + \mathbf{R}_n
\end{aligned}$$

where $\|\mathbf{R}_n\| = O(n^{-1})$ with high probability. In addition, $\|\mathbf{U}_{\mathcal{L}(\mathbf{A})}\mathbf{U}_{\mathcal{L}(\mathbf{A})} - \mathbf{U}_{\mathcal{L}(\mathbf{P})}\mathbf{U}_{\mathcal{L}(\mathbf{P})}\| = O(n^{-1/2})$ also implies that there exists an orthogonal matrix $\mathbf{W}^*$ such that $\|\mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top\mathbf{U}_{\mathcal{L}(\mathbf{A})} - \mathbf{W}^*\| = O(n^{-1})$ with high probability.

We next consider the terms $\mathbf{S}_{\mathcal{L}(\mathbf{P})}^{-1/2}\mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top\mathbf{U}_{\mathcal{L}(\mathbf{A})}$ and $\mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top\mathbf{U}_{\mathcal{L}(\mathbf{A})}\mathbf{S}_{\mathcal{L}(\mathbf{A})}^{-1/2}$. Note that the both are $d \times d$ matrices; furthermore, since $\mathbf{S}_{\mathcal{L}(\mathbf{A})}$ and $\mathbf{S}_{\mathcal{L}(\mathbf{P})}$ are diagonal matrices, the ij-th entry of $\mathbf{S}_{\mathcal{L}(\mathbf{P})}^{-1/2}\mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top\mathbf{U}_{\mathcal{L}(\mathbf{A})} - \mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top\mathbf{U}_{\mathcal{L}(\mathbf{A})}\mathbf{S}_{\mathcal{L}(\mathbf{A})}^{-1/2}$ can be written as the $\zeta_{ij} \times h_{ij}$ where ζ_{ij} is the ij-th entry of $\mathbf{S}_{\mathcal{L}(\mathbf{P})}\mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top\mathbf{U}_{\mathcal{L}(\mathbf{A})} - \mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top\mathbf{U}_{\mathcal{L}(\mathbf{A})}\mathbf{S}_{\mathcal{L}(\mathbf{A})}$ and the h_{ij} are functions of $\lambda_i(\mathcal{L}(\mathbf{A}))$

and $\lambda_j(\mathcal{L}(\mathbf{P}))$. In particular, $|h_{ij}| < C$ for some positive constant C for all n. We then have that

$$\begin{aligned}\mathbf{S}_{\mathcal{L}(\mathbf{P})}\mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top\mathbf{U}_{\mathcal{L}(\mathbf{A})} - \mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top\mathbf{U}_{\mathcal{L}(\mathbf{A})}\mathbf{S}_{\mathcal{L}(\mathbf{A})} &= \mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top(\mathcal{L}(\mathbf{P}) - \mathcal{L}(\mathbf{A}))\mathbf{U}_{\mathcal{L}(\mathbf{A})} \\ &= \mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top(\mathcal{L}(\mathbf{P}) - \mathcal{L}(\mathbf{A}))\mathbf{U}_{\mathcal{L}(\mathbf{P})}\mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top\mathbf{U}_{\mathcal{L}(\mathbf{A})} \\ &+ \mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top((\mathcal{L}(\mathbf{P}) - \mathcal{L}(\mathbf{A}))(\mathbf{I} - \mathbf{U}_{\mathcal{L}(\mathbf{P})}\mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top)\mathbf{U}_{\mathcal{L}(\mathbf{A})}\end{aligned}$$

Now, conditioning on $\mathbf{P}$, the ij-th entry of $\mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top(\mathcal{L}(\mathbf{P}) - \mathcal{L}(\mathbf{A}))\mathbf{U}_{\mathcal{L}(\mathbf{P})}$ can be written as a linear combination of the entries of $\mathbf{A} - \mathbf{P}$ (which are independent) and the rows of $\mathbf{X}$; hence, it can be bounded using Hoeffding's inequality. Meanwhile, the term

$$\mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top((\mathcal{L}(\mathbf{P}) - \mathcal{L}(\mathbf{A}))(\mathbf{I} - \mathbf{U}_{\mathcal{L}(\mathbf{P})}\mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top)\mathbf{U}_{\mathcal{L}(\mathbf{A})}$$

can be bounded by the Davis-Kahan Theorem and the spectral norm difference of $\|\mathcal{L}(\mathbf{A}) - \mathcal{L}(\mathbf{P})\|$. We therefore arrive at the important fact that

$$\|\mathbf{S}_{\mathcal{L}(\mathbf{P})}\mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top\mathbf{U}_{\mathcal{L}(\mathbf{A})} - \mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top\mathbf{U}_{\mathcal{L}(\mathbf{A})}\mathbf{S}_{\mathcal{L}(\mathbf{A})}\|_F = O(n^{-1})$$

with high probability, and hence

$$\|\mathbf{S}_{\mathcal{L}(\mathbf{P})}^{-1/2}\mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top\mathbf{U}_{\mathcal{L}(\mathbf{A})} - \mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top\mathbf{U}_{\mathcal{L}(\mathbf{A})}\mathbf{S}_{\mathcal{L}(\mathbf{A})}^{-1/2}\| = O(n^{-1})$$

with high probability.

We can juxtapose $\mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top\mathbf{U}_{\mathcal{L}(\mathbf{A})}$ and $\mathbf{S}_{\mathcal{L}(\mathbf{A})}^{-1/2}$ in the expression for Eq. (67) and then replace $\mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top\mathbf{U}_{\mathcal{L}(\mathbf{A})}$ by the orthogonal matrix $\mathbf{W}^*$, thereby obtaining

$$\mathbf{U}_{\mathcal{L}(\mathbf{A})}\mathbf{S}_{\mathcal{L}(\mathbf{A})}^{1/2} - \mathbf{U}_{\mathcal{L}(\mathbf{P})}\mathbf{S}_{\mathcal{L}(\mathbf{P})}^{1/2}\mathbf{W}^* = (\mathcal{L}(\mathbf{A}) - \mathcal{L}(\mathbf{P}))\mathbf{U}_{\mathcal{L}(\mathbf{P})}\mathbf{S}_{\mathcal{L}(\mathbf{P})}^{-1/2}\mathbf{W}^* + \tilde{\mathbf{R}}_n$$

where $\|\tilde{\mathbf{R}}_n\| = O(n^{-1})$ with high probability. Since

$$\tilde{\mathbf{X}}\tilde{\mathbf{X}}^\top = \mathcal{L}(\mathbf{P}) = \mathbf{U}_{\mathcal{L}(\mathbf{P})}\mathbf{S}_{\mathcal{L}(\mathbf{P})}\mathbf{U}_{\mathcal{L}(\mathbf{P})}^\top,$$

we have $\tilde{\mathbf{X}} = \mathbf{U}_{\mathcal{L}(\mathbf{P})}\mathbf{S}_{\mathcal{L}(\mathbf{P})}^{1/2}\tilde{\mathbf{W}}$ for some orthogonal matrix $\tilde{\mathbf{W}}$. Therefore,

$$\begin{aligned}\mathbf{U}_{\mathcal{L}(\mathbf{A})}\mathbf{S}_{\mathcal{L}(\mathbf{A})}^{1/2} - \tilde{\mathbf{X}}\tilde{\mathbf{W}}^\top\mathbf{W}^* &= (\mathcal{L}(\mathbf{A}) - \mathcal{L}(\mathbf{P}))\mathbf{U}_{\mathcal{L}(\mathbf{P})}\mathbf{S}_{\mathcal{L}(\mathbf{P})}^{-1/2}\mathbf{W}^* + \tilde{\mathbf{R}}_n \\ &= (\mathcal{L}(\mathbf{A}) - \mathcal{L}(\mathbf{P}))\mathbf{U}_{\mathcal{L}(\mathbf{P})}\mathbf{S}_{\mathcal{L}(\mathbf{P})}^{1/2}\tilde{\mathbf{W}}\tilde{\mathbf{W}}^\top\mathbf{S}_{\mathcal{L}(\mathbf{P})}^{-1}\tilde{\mathbf{W}}\tilde{\mathbf{W}}^\top\mathbf{W}^* + \tilde{\mathbf{R}}_n \\ &= (\mathcal{L}(\mathbf{A}) - \mathcal{L}(\mathbf{P}))\tilde{\mathbf{X}}(\tilde{\mathbf{X}}^\top\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{W}}^\top\mathbf{W}^* + \tilde{\mathbf{R}}_n.\end{aligned}$$

Equivalently,

$$\mathbf{U}_{\mathcal{L}(\mathbf{A})}\mathbf{S}_{\mathcal{L}(\mathbf{A})}^{1/2}(\mathbf{W}^*)^\top\tilde{\mathbf{W}} - \tilde{\mathbf{X}} = (\mathcal{L}(\mathbf{A}) - \mathcal{L}(\mathbf{P}))\tilde{\mathbf{X}}(\tilde{\mathbf{X}}^\top\tilde{\mathbf{X}})^{-1} + \tilde{\mathbf{R}}_n(\mathbf{W}^*)^\top\tilde{\mathbf{W}}. \tag{68}$$

The right hand side of Eq. (68) — except for the residual term $\tilde{\mathbf{R}}_n(\mathbf{W}^*)^\top\tilde{\mathbf{W}}$ which has norm of order $O(n^{-1})$ with high probability — can now be written explicitly in terms of the entries of $\mathbf{A}$ and $\mathbf{P}$. However, since

$$\mathcal{L}(\mathbf{A}) = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2},$$

the entries of the right hand side of Eq. (68) are neither linear nor affine combinations of the entries of $\mathbf{A} - \mathbf{P}$. Nevertheless, a Taylor-series expansion of the entries of $\mathbf{D}^{-1/2}$ allows us to conclude that

$$\|\mathbf{D}^{-1/2} - \mathbf{T}^{-1/2} - \tfrac{1}{2}\mathbf{T}^{-3/2}(\mathbf{T} - \mathbf{D})\| = O(n^{-3/2})$$

with high probability. Substituting this into Eq. (68) yields, after a few further algebraic simplifications, Eq. (66).

References

E. Abbe and C. Sandon. Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery. In *Proceedings of the 56th Annual Symposium on Foundations of Computer Science*, pages 670–688, 2015.

E. Abbe, A. S. Bandeira, and G. Hall. Exact recovery in the stochastic block model. *IEEE Transactions on Information Theory*, 62:471–487, 2016.

E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *The Journal of Machine Learning Research*, 9:1981–2014, 2008.

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19:716–723, 1974.

S. M. Ali and S. D. Shelvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society, Series B.*, 28:121–132, 1966.

N. Alon and J. Spencer. *The Probabilistic Method*. John Wiley, 3 edition, 2008.

E. Arias-Castro and N. Verzelen. Community detection in dense random networks. *Annals of Statistics*, 42:940–969, 2014.

A. Athreya, V. Lyzinski, D. J. Marchette, C. E. Priebe, D. L. Sussman, and M. Tang. A limit theorem for scaled eigenvectors of random dot product graphs. *Sankhya A*, 78:1–18, 2016.

R. Bhatia. *Matrix Analysis*. Springer, 1997.

P. Bickel and P. Sarkar. Role of normalization for spectral clustering in stochastic blockmodels. *Annals of Statistics*, 43:962–990, 2015.

P. Bickel, D. Choi, X. Chang, and H. Zhang. Asymptotic normality of maximum likelihood and its variational approximation for stochastic blockmodels. *Annals of Statistics*, 41: 1922–1943, 2013.

P. J. Bickel and A. Chen. A nonparametric view of network models and Newman-Girvan and other modularities. *Proceedings of the National Academy of Sciences of the United States of America*, 106:21068–21073, 2009.

P. J. Bickel, A. Chen, and E. Levina. The method of moments and degree distributions for network models. *Annals of Statistics*, 39:38–59, 2011.

B. Bollobás, S. Janson, and O. Riordan. The phase transition in inhomogeneous random graphs. *Random Structures and Algorithms*, 31:3–122, 2007.

J. Cape, M. Tang, and C. E. Priebe. The Kato-Temple inequality and eigenvalue concentration. *Electronic Journal of Statistics*, 11:3954–3978, 2017a.

J. Cape, M. Tang, and C. E. Priebe. The two-to-infinity norm and singular subspace geometry with applications to high-dimensional statistics. Arxiv preprint at `https://arxiv.org/abs/1705.10735`, 2017b.

S. Chatterjee. Matrix estimation by universal singular value thresholding. *Annals of Statistics*, 43:177–214, 2015.

L. Chen, J.T. Vogelstein, V. L. Lyzinski, and C. E. Priebe. A joint graph inference case study: the C. Elegans chemical and electrical connectomes. *Worm*, 5:e1142041, 2016.

H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23:493–507, 1952.

H. Chernoff. Large sample theory: Parametric case. *Annals of Mathematical Statistics*, 27: 1–22, 1956.

F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

K. L. Chung. *A course in probability theory*. Academic Press, 3 edition, 2001.

G. A. Coppersmith. Vertex nomination. *Wiley Interdisciplinary Reviews: Computational Statistics*, 6:144–153, 2014.

I. Csizár. Information-type measures of difference of probability distributions and indirect observations. *Studia Scientiarum Mathematicarum Hungarica*, 2:229–318, 1967.

C. Davis and W. Kahan. The rotation of eigenvectors by a pertubation. III. *SIAM Journal on Numerical Analysis*, 7:1–46, 1970.

C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1:211–218, 1936.

K. Eichler, F. Li, A. L. Kumar, Y. Park, I. Andrade, C. Schneider-Mizell, T. Saumweber, A. Huser, D. Bonnery, B. Gerber, R. D. Fetter, J. W. Truman, C. E. Priebe, L. F. Abbott, A. Thum, M. Zlatic, and A. Cardona. The complete wiring diagram of a high-order learning and memory center, the insect mushroom body. *Nature*, 548:175–182, 2017.

P. Erdős and A. Rényi. On the evolution of random graphs. *Proceedings of the Hungarian Academy of Sciences*, pages 17–61, 1960.

M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23:298–305, 1973.

D. E. Fishkind, D. L. Sussman, M. Tang, J. T. Vogelstein, and Carey E Priebe. Consistent adjacency-spectral partitioning for the stochastic block model when the model parameters are unknown. *SIAM Journal on Matrix Analysis and Applications*, 34:23–39, 2013.

D. E. Fishkind, V. Lyzinski, H. Pao, L. Chen, and C. E. Priebe. Vertex nomination schemes for membership prediction. *Annals of Applied Statistics*, 9:1510–1532, 2015a.

D. E. Fishkind, C. Shen, and C. E. Priebe. On the incommensurability phenomenon. *Journal of Classification*, 33:185–209, 2015b.

S. Fortunato. Community detection in graphs. *Physics Reports*, 486:75–174, 2010.

C. Fraley and A. E. Raftery. MCLUST: Software for model-based cluster analysis. *Journal of Classification*, 16:297–306, 1999.

C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis and density estimation. *Journal of the American Statistical Association*, 97:611–631, 2002.

Z. Füredi and J. Komlós. The eigenvalues of random symmetric matrices. *Combinatorica*, 1:233–241, 1981.

E. N. Gilbert. Random graphs. *Annals of Statistics*, 30:1141–1144, 1959.

A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi. A survey of statistical network models. *Foundations and Trends® in Machine Learning*, 2:129–233, 2010.

A. Gretton, K. M. Borgwadt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

P. Hoff. Modeling homophily and stochastic equivalence in symmetric relational data. In *Proceedings of the 20th Conference on Neural Information Processing Systems*, pages 657–664, 2007.

P. D. Hoff, A. E. Raftery, and M. S. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97:1090–1098, 2002.

P. W. Holland, K. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5:109–137, 1983.

R. Horn and C. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

J. E. Jackson. *A User's Guide to Principal Components*. John Wiley & Sons, Inc., 2004.

A. K. Jain, R. P. W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:4–37, 2000.

B. Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83, 2011.

T. Kato. On the upper and lower bounds of eigenvalues. *Physical Review Letters*, 77: 334–339, 1950.

T. Kawamoto and Y. Kabashima. Limitations in the spectral method for graph partitioning: detectability threshold and localization of eigenvectors. *Physical Review E*, 91, 2015.

E. D. Kolaczyk. *Statistical Analysis of Network Data*. Springer-Verlag, 2009.

F. Krzakala, C. Moore, E. Mossel, J. Neeman, A. Sly, L. Zdeborová, and P. Zhang. Spectral redemption in clustering sparse networks. *Proceedings of the National Academy of Sciences of the United States of America*, 110:20935–20940, 2013.

B. A. Landman, A. J. Huang, A. Gifford, D. S. Vikram, I. A. Lim, J. A. Farrell, et al. Multi-parametric neuroimaging reproducibility: a 3-t resource study. *Neuroimage*, 54: 2854–2866, 2011.

C. Le, E. Levina, and R. Vershynin. Concentration and regularization of random graphs. *Random Structures & Algorithms*, 51:538–561, 2017.

C. C. Leang and D. H. Johnson. On the asymptotics of M-hypothesis bayesian detection. *IEEE Transactions on Information Theory*, 43:280–282, 1997.

J. Lei. A goodness-of-fit test for stochastic block models. *Annals of Statistics*, 44:401–424, 2016.

J. Lei and A. Rinaldo. Consistency of spectral clustering in stochastic blockmodels. *Annals of Statistics*, 43:215–237, 2015.

K. Levin and V. Lyzinski. Laplacian eigenmaps from sparse, noisy similarity measurements. *IEEE Transactions on Signal Processing*, 65:1988–2003, 2017.

K. Levin, A. Athreya, M. Tang, V. Lyzinski, and C. E. Priebe. A central limit theorem for an omnibus embedding of random dot product graphs. *arXiv preprint arXiv:1705.09355*, 2017.

F. Liese and I. Vadja. On divergences and informations in statistics and information theory. *IEEE Transactions on Information Theory*, 52:4394–4412, 2006.

B. G. Lindsay. The geometry of mixture likelihoods: A general theory. *Annals of Statistics*, 11:86–94, 1983.

L. Lu and X. Peng. Spectra of edge-independent random graphs. *Electronic Journal of Combinatorics*, 20, 2013.

R. Lyons. Distance covariance in metric spaces. *Annals of Probability*, 41:3284–3305, 2013.

V. Lyzinski, D. L. Sussman, M. Tang, A. Athreya, and C. E. Priebe. Perfect clustering for stochastic blockmodel graphs via adjacency spectral embedding. *Electronic Journal of Statistics*, 8:2905–2922, 2014.

V. Lyzinski, D. E. Fishkind, M. Fiori, J. T. Vogelstein, C. E. Priebe, and G. Sapiro. Graph matching: Relax at your own risk. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38:60–73, 2016a.

V. Lyzinski, K. Levin, D. Fishkind, and C. E. Priebe. On the consistency of the likelihood maximization vertex nomination scheme: Bridging the gap between maximum likelihood estimation and graph matching. *Journal of Machine Learning Research*, 17, 2016b.

V. Lyzinski, M. Tang, A. Athreya, Y. Park, and C. E. Priebe. Community detection and classification in hierarchical stochastic blockmodels. *IEEE Transactions in Network Science and Engineering*, 4:13–26, 2017.

J.-F. Maa, D. K. Pearl, and R. Bartoszyński. Reducing multidimensional two-sample data to one-dimensional interpoint comparisons. *Annals of Statistics*, 24:1067–1074, 1996.

D. Marchette, C. E. Priebe, and G. Coppersmith. Vertex nomination via attributed random dot product graphs. In *Proceedings of the 57th ISI World Statistics Congress*, 2011.

E. Mossel, J. Neeman, and A. Sly. Belief propagation, robust reconstruction and optimal recovery of block models. In *Proceedings of the 27th Annual Conference on Learning Theory*, pages 356–370, 2014.

E. Mossel, J. Neeman, and A. Sly. A proof of the block model threshold conjecture. *Combinatorica*, 2018. In press.

F. Mosteller and R. A. Fisher. Questions and answer. *The American Statistician*, 2:30–31, 1948.

M. E. J. Newman. Modularity and community structure in networks. *Proccedings of the National Academy of Sciences of the United States of America*, 103:8577–8582, 2006.

T. Ohyama, C. M. Schneider-Mizell, R. D. Fetter J. V. Aleman, R. Franconville, M. Rivera-Alba, B. D. Mensh, K. M. Branson, J. H. Simpson, J. W. Truman, A. Cardona, and M. Zlatic. A multilevel multimodal circuit enhances action selection in Drosophila. *Nature*, 520:633–639, 2015.

R. I. Oliveira. Concentration of the adjacency matrix and of the Laplacian in random graphs with independent edges. `http://arxiv.org/abs/0911.0600`, 2009.

D. Pollard. Strong consistency of k-means clustering. *Annals of Statistics*, 9:135–140, 1981.

C. E. Priebe, D. L. Sussman, M. Tang, and J. T. Vogelstein. Statistical inference on errorfully observed graphs. *Journal of Computational and Graphical Statistics*, 24:930–953, 2015.

C. E. Priebe, Y. Park, M. Tang, A. Athreya, V. Lyzinski, J. T. Vogelstein, Y. Qin, B. Cocanougher, K. Eichler, M. Zlatic, and A. Cardona. Semiparametric spectral modeling of the drosophila connectome. arXiv preprint at `https://arxiv.org/abs/1705.03297`, 2017.

J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

K. Rohe, S. Chatterjee, and B. Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *Annals of Statistics*, 39:1878–1915, 2011.

W. G. Roncal, Z. H. Koterba, D. Mhembere, D. M. Kleissas, J. T. Vogelstein, R. Burns, et al. Migraine: MRI graph reliability analysis and inference for connectomics. Arxiv preprint at `http://arxiv.org/abs/1312.4875`, 2013.

P. Rubin-Delanchy, C. E. Priebe, and M. Tang. The generalised random dot product graph. Arxiv preprint available at `http://arxiv.org/abs/1709.05506`, 2017.

E. R. Scheinerman and K. Tucker. Modeling graphs using dot product representations. *Computational statistics*, 25:1–16, 2010.

C. M. Schneider-Mizell, S. Gerhard, M. Longair, T. Kazimiers, F. Li, M. F. Zwart, A. Champion, F. M. Midgley, R. D. Fetter, S. Saalfeld, and A. Cardona. Quantitative neuroanatomy for connectomics in Drosophila. *Elife*, 5:e12059, 2016.

G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

D. Sejdinovic, B. Sriperumbudur, A. Gretton, and K. Fukumizu. Equivalence of distance-based and RKHS-based statistics in hypothesis testing. *Annals of Statistics*, 41:2263–2291, 2013.

T. A. B. Snijders and K.Nowicki. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14:75–100, 1997.

B. K. Sriperumbudur, K. Fukumizu, and G. R. G. Lanckriet. Universality, characteristic kernels and RKHS embeddings of measures. *Journal of Machine Learning Research*, 12:2389–2410, 2011.

I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93, 2001.

D. L. Sussman, M. Tang, D. E. Fishkind, and C. E. Priebe. A consistent adjacency spectral embedding for stochastic blockmodel graphs. *Journal of the American Statistical Association*, 107:1119–1128, 2012.

D. L. Sussman, M. Tang, and C. E. Priebe. Consistent latent position estimation and vertex classification for random dot product graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36:48–57, 2014.

S. Suwan, D. S Lee, R. Tang, D.L. Sussman, M. Tang, and C.E. Priebe. Empirical Bayes estimation for the stochastic blockmodel. *Electronic Journal of Statistics*, 10:761–782, 2016.

G. J. Székely and M. L. Rizzo. Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference*, 143:1249–1272, 2013.

M. Tang and C. E. Priebe. Limit theorems for eigenvectors of the normalized laplacian for random graphs. *Annals of Statistics*, 2018. In press.

M. Tang, D. L. Sussman, and C. E. Priebe. Universally consistent vertex classification for latent position graphs. *Annals of Statistics*, 41:1406 – 1430, 2013.

M. Tang, A. Athreya, D. L. Sussman, V. Lyzinski, Y. Park, and C. E. Priebe. A semiparametric two-sample hypothesis testing problem for random dot product graphs. *Journal of Computational and Graphical Statistics*, 26:344–354, 2017a.

M. Tang, A. Athreya, D. L. Sussman, V. Lyzinski, and C. E. Priebe. A nonparametric two-sample hypothesis testing problem for random dot product graphs. *Bernoulli*, 23:1599–1630, 2017b.

R. Tang, M. Ketcha, J. T. Vogelstein, C. E. Priebe, and D. L. Sussman. Laws of large graphs. arXiv preprint at `http://arxiv.org/abs/1609.01672`, 2016.

T. Tao and V. Vu. Random matrices: Universal properties of eigenvectors. *Random Matrices: Theory and Applications*, 1, 2012.

J. A. Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends in Machine Learning*, 8:1–230, 2015.

U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, 2007.

Y. J. Wang and G. Y. Wong. Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82:8–19, 1987.

P. J. Wolfe and S. C. Olhede. Nonparametric graphon estimation. arXiv preprint at `http://arxiv.org/abs/1309/5936`, 2013.

S. Young and E. Scheinerman. Random dot product graph models for social networks. In *Proceedings of the 5th international conference on algorithms and models for the web-graph*, pages 138–149, 2007.

Y. Yu, T. Wang, and R. J. Samworth. A useful variant of the Davis-Kahan theorem for statisticians. *Biometrika*, 102:315–323, 2015.

D. Zheng, D. Mhembere, R. Burns, J. T. Vogelstein, C. E. Priebe, and A. S. Szalay. Flashgraph: Processing billion-node graphs on an array of commodity SSDs. In *Proceedings of the 13th USENIX Conference on File and Storage Technologies*, 2015.

M. Zhu and A. Ghodsi. Automatic dimensionality selection from the scree plot via the use of profile likelihood. *Computational Statistics and Data Analysis*, 51:918–930, 2006.

Journal of Machine Learning Research 18 (2018) 1-16 Submitted 12/17; Revised 05/18; Published 6/18

Rate of Convergence of k-Nearest-Neighbor Classification Rule

Maik Döring MAIK.DOERING@UNI-HOHENHEIM.DE
Institute of Applied Mathematics and Statistics
University of Hohenheim, 70599 Stuttgart, Germany,
Max Rubner Institute,76131 Karlsruhe, Germany

László Györfi GYORFI@CS.BME.HU
Department of Computer Science and Information Theory
Budapest University of Technology and Economics
1111 Budapest, Hungary

Harro Walk HARRO.WALK@T-ONLINE.DE
Institute of Stochastic and Applications
University of Stuttgart
70049 Stuttgart, Germany

Editor: John Shawe-Taylor

Abstract

A binary classification problem is considered. The excess error probability of the k-nearest-neighbor classification rule according to the error probability of the Bayes decision is revisited by a decomposition of the excess error probability into approximation and estimation errors. Under a weak margin condition and under a modified Lipschitz condition or a local Lipschitz condition, tight upper bounds are presented such that one avoids the condition that the feature vector is bounded. The concept of modified Lipschitz condition is applied for discrete distributions, too. As a consequence of both concepts, we present the rate of convergence of L_2 error for the corresponding nearest neighbor regression estimate.

Keywords: rate of convergence, classification, error probability, k-nearest-neighbor rule

1. Introduction

Let the feature vector X take values in $\mathbb{R}^d$, and let its label Y be ± 1 valued. If g is an arbitrary decision function then its error probability is denoted by

$$L(g) = \mathbb{P}\{g(X) \neq Y\}.$$

Put

$$D(x) = \mathbb{E}\{Y \mid X = x\},$$

then the Bayes decision g^* minimizes the error probability:

$$g^*(x) = sign\, D(x),$$

where $sign(z) = 1$ for $z > 0$ and $sign(z) = -1$ for $z \leq 0$, and

$$L^* = \mathbb{P}\{g^*(X) \neq Y\}$$

denotes its error probability.

In the standard model of pattern recognition, we are given training labeled samples, which are independent and identical copies of (X, Y):

$$\mathcal{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}.$$

Based on these labeled samples, one can estimate the regression function D by $\tilde{D}$, and the corresponding plug-in classification rule g derived from $\tilde{D}$ is defined by

$$g(x) = sign\, \tilde{D}(x).$$

Then for any plug-in rule g derived from the regression estimate $\tilde{D}$ we have

$$L(g) - L^* = \mathbb{E}\left\{\mathbb{I}_{\{g(X) \neq g^*(X)\}} |D(X)|\right\} = \mathbb{E}\left\{\mathbb{I}_{\{sign\, \tilde{D}(X) \neq sign\, D(X)\}} |D(X)|\right\}, \tag{1}$$

where $\mathbb{I}$ denotes the indicator function (compare Theorem 2.2 in Devroye, Györfi and Lugosi 1996).

In the sequel our focus lies on the rate of convergence of the excess error probability $\mathbb{E}\{L(g_{n,k})\} - L^*$, where $g_{n,k}$ is the k-nearest-neighbor rule defined as follows. We fix $x \in \mathbb{R}^d$, and reorder the data $(X_1, Y_1), \ldots, (X_n, Y_n)$ according to increasing values of $\|X_i - x\|$, where $\|\cdot\|$ denotes the Euclidean norm. The reordered data sequence is denoted by

$$(X_{(n,1)}(x), Y_{(n,1)}(x)), \ldots, (X_{(n,n)}(x), Y_{(n,n)}(x)).$$

$X_{(n,k)}(x)$ is the k-th nearest neighbor of x. In this paper we assume that tie happens with probability 0. For instance when the distribution μ of X has a density f, this assumption is satisfied. In any case, by adding a randomizing component to X one can ensure that this assumption holds. Choose an integer k less than n, then the k-nearest-neighbor estimate of D is

$$D_{n,k}(x) = \frac{1}{k} \sum_{i=1}^{k} Y_{(n,i)}(x),$$

and the k-nearest-neighbor classification rule is

$$g_{n,k}(x) = sign\, D_{n,k}(x).$$

Concerning the properties of k-nearest-neighbor rule and the related literature see Biau and Devroye (2015).

The main aim of this paper is to show tight upper bounds on the excess error probability $\mathbb{E}\{L(g_{n,k})\} - L^*$ of the k-nearest-neighbor classification rule $g_{n,k}$. Given the plug-in classification rule g derived from $\tilde{D}$, (1) implies that

$$\mathbb{E}\{L(g)\} - L^* \leq \mathbb{E}\{|D(X) - \tilde{D}(X)|\}.$$

Therefore we may get an upper bound on the rate of convergence of the excess error probability $\mathbb{E}\{L(g_{n,k})\} - L^*$ via the L_1 rate of convergence of the corresponding regression estimation. Then

$$\mathbb{E}\{L(g_{n,k})\} - L^* \leq \mathbb{E}\{|D(X) - D_{n,k}(X)|\}.$$

Under some smoothness assumptions on D one could further upper bound the L_1 rate. For instance we may assume that D satisfies the *Lipschitz condition*: there is a constant C such that for any $x, z \in \mathbb{R}^d$

$$|D(x) - D(z)| \leq C\|x - z\|.$$

If D is Lipschitz continuous and X is bounded with the diameter M of the support of μ, then

$$\mathbb{E}\{|D(X) - D_{n,k}(X)|^2\} \leq c_1 M^2 (k/n)^{2/d} + c_2/k \tag{2}$$

with $d \geq 2$ (compare Chapter 6 in Györfi et al. 2002 and Liitiäinen, Corona and Lendasse 2010), so for $k = \lfloor c_3 n^{2/(d+2)} \rfloor$,

$$\mathbb{E}\{L(g_{n,k})\} - L^* \leq \sqrt{\mathbb{E}\{|D(X) - D_{n,k}(X)|^2\}} \leq c_4 n^{-1/(d+2)}. \tag{3}$$

However, according to Section 6.7 in Devroye, Györfi and Lugosi (1996) the classification is easier than L_1 regression function estimation, since the rate of convergence of the error probability depends on the behavior of the function D in the neighborhood of the decision boundary

$$B_0 = \{x; D(x) = 0\}. \tag{4}$$

This phenomenon has been discovered and investigated by Mammen and Tsybakov (1999); Tsybakov (2004); Audibert and Tsybakov (2007) and Kohler and Krzyżak (2007), who introduced the (weak) margin condition:

- The *weak margin condition* means that for all $0 < t \leq 1$,

$$\mathbb{P}\{0 < |D(X)| \leq t\} \leq c^* \cdot t^\alpha,$$

  where $\alpha > 0$ and $c^* > 0$.

Denote by

$$B_{0,r} = \left\{ x; \min_{z \in B_0} \|x - z\| \leq r \right\}, \; r > 0,$$

the closed r-neighborhood of the decision boundary B_0 defined by (4). Let λ be the Lebesgue measure and let $M^*(B_0)$ be the outer surface (Minkowski content) of the decision boundary B_0 defined by

$$M^*(B_0) = \lim_{r \downarrow 0} \frac{\lambda(B_{0,r} \setminus B_0)}{r}.$$

If D satisfies the Lipschitz condition, X has a density f, the density f is bounded by f_{max} and $M^*(B_0)$ is finite, then Lemma 2 in Döring, Györfi and Walk (2015) implies that the weak margin condition holds with $\alpha = 1$. Notice that the Lipschitz condition implies $\alpha \leq 1$.

In the analysis of classification rule one may use conditions on the density f of X:

- The *strong density condition* means that for $f(x) > 0$,

$$f(x) \geq f_{min} > 0.$$

The strong density condition implies that the support of the density f has finite Lebesgue measure, and so this assumption is close to the condition that X is bounded. It is a very restrictive condition, excluding important densities like Gaussian densities.

Kohler and Krzyżak (2007) proved that under the margin condition, Lipschitz condition and strong density assumption, for choice

$$k_n = \lfloor (\log n)^2 n^{2/(d+2)} \rfloor,$$

the order of the upper bound is smaller than (3):

$$(\log n)^{\frac{2(1+\alpha)}{d}} n^{-\frac{1+\alpha}{d+2}}.$$

Gadat, Klein and Marteau (2016) (comprehending also some classes of distributions with unbounded support) extended this bound such that under the margin condition, Lipschitz condition and the so called strong minimal mass assumption, for choice

$$k_n = \lfloor n^{2/(d+2)} \rfloor, \tag{5}$$

one has the order

$$n^{-\frac{1+\alpha}{d+2}}. \tag{6}$$

Audibert and Tsybakov (2007) showed that, under the margin condition and the strong density assumption, (6) is the minimax optimal rate of convergence for the class of Lipschitz continuous D, that is, (6) is the lower bound for *any* classifier.

Let $S_{x,r} = \{x' \in \mathbb{R}^d : \|x' - x\| \leq r\}$ and $S^o_{x,r} = \{x' \in \mathbb{R}^d : \|x' - x\| < r\}$ be the closed and open Euclidean ball, respectively, centered at $x \in \mathbb{R}^d$ with radius $r > 0$. In Chaudhuri and Dasgupta (2014) distribution-dependent rates of convergence are provided for the nearest neighbor classification rule in the framework of metric spaces. Therein a smoothness condition with respect to the distribution μ is introduced: For positive constants κ and L it is assumed that

$$\left| D(x) - \frac{1}{\mu(S_{x,r})} \int_{S_{x,r}} D(z)\mu(dz) \right| \leq L\mu(S^o_{x,r})^\kappa$$

for all $r > 0$ and x in the support of μ. Then the regression function D is called (κ, L)-smooth. Chaudhuri and Dasgupta (2014) revisited the order (6) using such a smoothness condition.

For higher order smoothness, one gets better rates of convergence. For weighted nearest neighbor classification including non-weighted k-nearest-neighbor classification, Samworth (2012a,b), with further references, considered the case when X is bounded, D is continuously differentiable with gradient $\nabla D(x) \neq 0$ for $x \in B_0$, the conditional densities of X given Y are twice differentiable and the density f of X satisfies the strong density assumption. Under some additional conditions on B_0, Samworth (2012b) derives the margin condition with $\alpha = 1$ and shows

$$\mathbb{E}\{L(g_{n,k})\} - L^* \leq \frac{c_5}{k} + c_6 (k/n)^{4/d},$$

which implies the order

$$n^{-\frac{4}{d+4}}.$$

In Cannings, Berett and Samworth (2017) the order $n^{-\frac{4}{d+4}}$ is revisited combining tail and smoothness conditions. For the feature vector a moment condition instead of boundedness is required. Further it is assumed that in a neighborhood of the decision boundary the function D and the marginal feature density are twice continuously differentiable and that the latter density allows to control the error of a Taylor approximation even in this region. For feature values away from the decision boundary it is assumed that the marginal feature distribution fulfills the strong minimal mass assumption, see Gadat, Klein and Marteau (2016), and that the function D does not approach the decision boundary too fast:

$$\sup_{x\in\mathbb{R}^d\setminus B_{0,r}:f(x)\geq\delta} |D(x)|^{-1} = o(\delta^{-\tau}) \text{ as } \delta\to 0 \text{ for some } r>0 \text{ and for every } \tau>0.$$

Interestingly, the analysis of empirical error minimization rules can avoid the condition that X is bounded, see Binev, Cohen, Dahmen and DeVore (2014) and Blaschzyk and Steinwart (2018).

Under the margin condition with $\alpha \leq 1$ ($d \geq 2$) and the strong density assumption, Audibert and Tsybakov (2007) showed that the order

$$n^{-\frac{2(1+\alpha)}{d+4}}$$

is the minimax optimal rate of convergence for the class of regression functions D, which have Lipschitz continuous gradients, that is, they are differentiable and the partial derivatives are Lipschitz continuous. Samworth (2012b) showed that under the assumptions together with Lipschitz continuity of the density function f several weighted nearest neighbor classifiers, particularly the non-weighted k-nearest-neighbor classifiers, attain this minimax rate.

2. Rate of Convergence of the Error Probability for k-NN Classifier

For most of the above cited results, the feature vector X is assumed to be bounded. Whenever the strong density assumption is used, it is implicitly assumed that the feature vector is bounded. They exclude the classical parametric discrimination problem, where the conditional distribution of X given Y are multidimensional Gaussian distributions. Next, we revisit these bounds such that our main aim is to avoid the condition that X is bounded and the strong density assumption.

In order to have non-trivial rate of convergence of the classification error probability, one has to assume tail and smoothness conditions. We treat two concepts of combined tail and smoothness condition, under which we get the known minimax rate of convergence.

- The *modified Lipschitz condition* means that there is a constant C^* such that for any $x, z \in \mathbb{R}^d$
$$|D(x) - D(z)| \leq C^* \mu(S^o_{x,\|x-z\|})^{1/d}.$$

Obviously, this definition is equivalent to the corresponding symmetric definition, i.e.,

$$|D(x) - D(z)| \leq C^* \min\{\mu(S^o_{x,\|x-z\|})^{1/d}, \mu(S^o_{z,\|x-z\|})^{1/d}\}.$$

In fact, for (κ, L)-smooth regression function D, Chaudhuri and Dasgupta (2014) already introduced the modified Lipschitz condition, where $\kappa = 1/d$ and $L = C^*$. The modified

Lipschitz condition is a universal condition not assuming the existence of a density, it holds for any pair of distribution μ and function D of practical interest.

The main result (Theorem 1) establishes rate of convergence under the modified Lipschitz condition by a decomposition of the excess error into approximation and estimation errors such that it extends and sharpens the result of Kohler and Krzyżak (2007) by avoiding the use of the strong density assumption. Furthermore, Theorem 7b in Chaudhuri and Dasgupta (2014) is closely related to Theorem 1 below.

Theorem 1 *Assume that tie happens with probability* 0*, D satisfies the weak margin condition with $0 < \alpha \leq 1$ and the modified Lipschitz condition. Then*

$$\mathbb{E}\{L(g_{n,k})\} - L^* = O(1/k^{(1+\alpha)/2}) + O((k/n)^{(\alpha+1)/d}),$$

and the choice (5) yields the order (6).

Because of (1), we have the following decomposition of the excess error probability:

$$\begin{aligned}\mathbb{E}\{L(g_{n,k})\} - L^* &= \mathbb{E}\left\{\int_{\{sign\, D_{n,k}(x)\neq sign\, D(x)\}} |D(x)|\mu(dx)\right\}\\ &\leq \mathbb{E}\left\{\int_{\{|D_{n,k}(x)-D(x)|\geq|D(x)|\}} |D(x)|\mu(dx)\right\}\\ &\leq I_{n,k} + J_{n,k},\end{aligned}$$

where

$$I_{n,k} = \mathbb{E}\left\{\int_{\{|\overline{D}_{n,k}(x)-D(x)|\geq|D(x)|/2\}} |D(x)|\mu(dx)\right\}$$

and

$$J_{n,k} = \mathbb{E}\left\{\int_{\{|D_{n,k}(x)-\overline{D}_{n,k}(x)|\geq|D(x)|/2\}} |D(x)|\mu(dx)\right\}$$

with

$$\overline{D}_{n,k}(x) = \mathbb{E}\{D_{n,k}(x) \mid X_1, \ldots, X_n\} = \frac{1}{k}\sum_{i=1}^{k} D(X_{(n,i)}(x)). \tag{7}$$

$I_{n,k}$ is called *approximation error*, while $J_{n,k}$ is the *estimation error*.

We split Theorem 1 into two lemmas such that Lemma 2 is on the estimation error, while Lemma 3 is on the approximation error.

Lemma 2 *If D satisfies the weak margin condition with $0 < \alpha \leq 1$, then*

$$J_{n,k} = O(1/k^{(1+\alpha)/2}). \tag{8}$$

Lemma 3 *Assume that tie happens with probability* 0*. If D satisfies the weak margin condition with $0 < \alpha \leq 1$ and the modified Lipschitz condition holds, then*

$$I_{n,k} \leq e^{-(1-\log 2)k} + O((k/n)^{(\alpha+1)/d}). \tag{9}$$

The proofs of these lemmas are in Section 4.

The concept of modified Lipschitz condition can be applied for discrete distributions, too. As an example, assume that the values of X are positive integers:

$$\mathbb{P}\{X = j\} = p_j.$$

For the classical example for slow rate of convergence, put $Y = D(X)$, that means the function D takes ± 1 values. Then $L^* = 0$ and D satisfies the Lipschitz condition with $C = 2$. As in the proof of Theorem 7.2 in Devroye, Györfi and Lugosi (1996), for any classifier g_n, the rate of convergence of $\mathbb{E}\{L(g_n)\}$ to zero can be arbitrarily slow by appropriate choice of the distribution $\{p_i\}$ of large tail.

Consider this discrete case with arbitrary function D such that the modified Lipschitz condition has the form

$$|D(j) - D(j')| \le C^* \mu([j - |j - j'| + 1, j + |j - j'| - 1]). \tag{10}$$

Next we show that under (10) and for any distribution $\{p_i\}$, even with large tail, the slow rate of convergence is excluded. Apply the k-NN rule with tie-breaking by indices, such that the k-NN estimate of D has the form

$$D_{n,k}(j) = \frac{1}{k} \sum_{i=1}^{k} Y_{(n,i)}(j).$$

Proposition 4 *Under the modified Lipschitz condition (10),*

$$\mathbb{E}\{L(g_{n,k})\} - L^* \le \frac{2 \max_{\{0 \le z\}} z e^{-z^2/8}}{\sqrt{k}} + e^{-3k/14} + 4C^* k/n.$$

The modified Lipschitz condition is an implicit condition, it is used in the proof of Lemma 3 in Section 4. We show how to extend this proof starting from a second concept such that we avoid the boundedness of X again. One can check that the Lipschitz condition and the strong density condition imply both concepts. However, as we mentioned earlier, the strong density condition is close to the condition, that X is bounded.

In the framework of the second concept we assume that μ has a density f satisfying a mild condition:

- The *weak density condition* means that there exist $c_{min} > 0$ and $\delta > 0$ such that for $f(x) r^d \le \delta^d$,
$$\mu(S_{x,r}) \ge c_{min}^d f(x) r^d.$$

If v_d denotes the volume of the unit ball $S_{0,1}$, then the Lebesgue density theorem implies that for all f and for almost all x with respect to the Lebesgues measure,

$$\lim_{r \downarrow 0} \frac{\mu(S_{x,r})}{v_d r^d} = f(x),$$

therefore, for small $r > 0$,

$$\mu(S_{x,r}) \approx f(x) v_d r^d.$$

Thus, the weak density condition requires a bit more than the Lebesgue density theorem.

- The *local Lipschitz condition* means that there exists a constant $\bar{C}$ such that for any $x, z \in \mathbb{R}^d$ with $f(x) > 0$
$$|D(x) - D(z)| \le \overline{C} f(x)^{1/d} \|x - z\|.$$

For the local Lipschitz condition the Lipschitz factor is proportional to $f(x)^{1/d}$. Thus, the fluctuation of D is small if the density is small. One may have a symmetric definition meaning that there exists a constant $\bar{C}$ such that for $\min\{f(x), f(z)\} > 0$

$$|D(x) - D(z)| \le \overline{C} \min\{f(x), f(z)\}^{1/d} \|x - z\|.$$

However, this definition has no additional advantage, just makes the derivation more involved. Because all three sets $\{(x,z); f(x) = 0\}$, $\{(x,z); f(z) = 0\}$, $\{(x,z); f(x) = 0 or f(z) = 0\}$ have zero product measure $\mu \otimes \mu$, both definitions are equivalent. Notice that the local Lipschitz condition is satisfied for any pair of density f and function D of practical interest.

Theorem 5 below states that under the local Lipschitz condition together with the weak density condition instead of the modified Lipschitz condition, the assertion of Theorem 1 remains valid. It will be shown in Section 4 by a modification of the proof of Lemma 3.

Theorem 5 *Assume that μ has a density such that the weak density condition holds. Furthermore, suppose that D satisfies the weak margin condition with $0 < \alpha \le 1$ and the local Lipschitz condition. Then*

$$\mathbb{E}\{L(g_{n,k})\} - L^* = O(1/k^{(1+\alpha)/2}) + O((k/n)^{(\alpha+1)/d}),$$

and the choice (5) yields the order (6).

On the one hand, note that the conditions of Theorem 1 don't imply the conditions of Theorem 5, because for Theorem 1, even the existence of a density is not required. On the other hand, the local Lipschitz and the weak density conditions imply that the modified Lipschitz condition holds in a neighborhood of x. It means that for all $x, z \in \mathbb{R}^d$ with $f(x) > 0$ and with $\|x - z\| \le \delta f(x)^{-1/d}$, one has

$$|D(x) - D(z)| \le \overline{C} f(x)^{1/d} \|x - z\| \le \overline{C} \mu(S^o_{x,\|x-z\|})^{1/d} / c_{min}.$$

3. Rate of Convergence of the L_2 Error for k-NN Regression Estimator

In this section we summarize the consequences of the previous section for k-NN regression estimation such that we prove (2) without assuming that X is bounded. Usually, (2) is proved such that after applying the Lipschitz condition one investigates

$$\mathbb{E}\{\|X_{(n,k)}(X) - X\|^2\},$$

which involves the condition that X is bounded. Compare Theorem 14.5 in Biau and Devroye (2015), Theorem 6.2 in Györfi et al. (2002) and Theorem 3.2 in Liitiäinen, Corona and Lendasse (2010), also Theorem 2 in Kohler, Krzyżak and Walk (2006), where a moment condition on X is assumed.

Theorem 6 *Put*

$$\sigma^2(x) = \mathbb{E}\left\{(Y - D(X))^2 \mid X = x\right\}.$$

If tie happens with probability 0*, D satisfies the modified Lipschitz condition and $k/n \to 0$, then*

$$\int \mathbb{E}\{(D_{n,k}(x) - D(x))^2\}\mu(dx) \le (\mathbb{E}\{\sigma^2(X)\} + o(1))/k + 2C^{*2}(k/n)^{2/d}.$$

The proof of Theorem 6 is at the end of Section 4. Similarly to Theorem 5, in Theorem 6 the modified Lipschitz condition can be replaced by the local Lipschitz condition together with the weak density condition.

4. Proofs

In this section we present the proofs of Lemmas 2 and 3, hence Theorem 1, and of Proposition 4, and of Theorems 5 and 6.

Proof of Lemma 2. For a fixed x, Proposition 8.1 in Biau and Devroye (2015) says the following: given $X_1, \dots, X_n$, the random pairs

$$(X_{(n,1)}(x), Y_{(n,1)}(x)), \dots, (X_{(n,k)}(x), Y_{(n,k)}(x))$$

are independent, and

$$\mathbb{E}\{Y_{(n,i)}(x) - D(X_{(n,i)}(x)) \mid X_1, \dots, X_n\} = 0.$$

Therefore, the Hoeffding inequality implies that

$$\begin{aligned}
&\mathbb{P}\{|D_{n,k}(x) - \overline{D}_{n,k}(x)| \ge |D(x)|/2 \mid X_1, \dots, X_n\} \\
&= \mathbb{P}\left\{\left|\frac{1}{k}\sum_{i=1}^{k}(Y_{(n,i)}(x) - D(X_{(n,i)}(x)))\right| \ge |D(x)|/2 \mid X_1, \dots, X_n\right\} \\
&\le 2e^{-k|D(x)|^2/8}.
\end{aligned}$$

Thus,

$$J_{n,k} \le 2\int |D(x)|e^{-k|D(x)|^2/8}\mu(dx).$$

The weak margin condition

$$G(t) := \mathbb{P}\{0 < |D(X)| \le t\} \le c^* \cdot t^\alpha,$$

implies by use of partial integration with respect to $G(s)$ that

$$\begin{aligned}
\int |D(x)|e^{-k|D(x)|^2/8}\mu(dx) &= \int_0^1 se^{-ks^2/8}G(ds) \\
&= e^{-k/8} - \int_0^1 e^{-ks^2/8}[1 - ks^2/4]G(s)ds \\
&\le e^{-k/8} + \frac{c^*}{4}\int_0^1 e^{-ks^2/8}ks^{2+\alpha}ds \\
&\le e^{-k/8} + \frac{c^*}{4}k^{-(\alpha+1)/2}\int_0^\infty e^{-u^2/8}u^{2+\alpha}du \\
&= O(k^{-(\alpha+1)/2}).
\end{aligned}$$

Thus, (8) is obtained. ■

Proof of Lemma 3. For $U_1, \dots, U_n$ i.i.d. uniformly distributed on $[0,1]$, let $U_{(1,n)}, \dots, U_{(n,n)}$ denote the corresponding order statistic. If tie happens with probability 0, then for any fixed x, $\mu(S_{x,r})$ is continuous in r, which implies that $\mu(S_{x,\|x-X\|})$ is uniformly distributed on $[0,1]$. From Section 1.2 in Biau and Devroye (2015) for any fixed x we have that

$$\mu(S_{x,\|x-X_{(n,k)}(x)\|}) \stackrel{\mathcal{D}}{=} U_{(k,n)}. \tag{11}$$

Because of

$$\begin{aligned}
|D(x) - \overline{D}_{n,k}(x)| &= \left|D(x) - \frac{1}{k}\sum_{i=1}^{k}D(X_{(n,i)}(x))\right| \\
&\le \frac{1}{k}\sum_{i=1}^{k}|D(x) - D(X_{(n,i)}(x))|
\end{aligned}$$

the modified Lipschitz condition together with (11) implies that

$$\begin{aligned}
&\mathbb{P}\left\{|D(x)|/2 < |D(x) - \overline{D}_{n,k}(x)|\right\} \\
&\le \mathbb{P}\left\{|D(x)|/2 < C^*\frac{1}{k}\sum_{i=1}^{k}\mu(S_{x,\|x-X_{(n,i)}(x)\|})^{1/d}\right\} \\
&\le \mathbb{P}\left\{|D(x)|/2 < C^*\mu(S_{x,\|x-X_{(n,k)}(x)\|})^{1/d}\right\} \\
&= \mathbb{P}\left\{|D(x)|/2 < C^*U_{(k,n)}^{1/d}\right\} \\
&= \mathbb{P}\left\{|D(x)|^d/(2C^*)^d < U_{(k,n)}\right\}.
\end{aligned} \tag{12}$$

Without loss of generality, assume that $C^* \geq 1/2$. Then

$$\begin{aligned}
&\mathbb{P}\left\{|D(x)|/2 < |D(x) - \overline{D}_{n,k}(x)|\right\} \\
&\leq \mathbb{P}\left\{\sum_{i=1}^{n} \mathbb{I}_{\{U_i \leq |D(x)|^d/(2C^*)^d\}} < k\right\} \\
&\leq \mathbb{I}_{\{|D(x)|^d/(2C^*)^d \geq 2k/n\}} \mathbb{P}\left\{\sum_{i=1}^{n} \mathbb{I}_{\{U_i \leq |D(x)|^d/(2C^*)^d\}} < \frac{n}{2}|D(x)|^d/(2C^*)^d\right\} \\
&+ \mathbb{I}_{\{|D(x)|^d/(2C^*)^d < 2k/n\}} \\
&\leq \mathbb{I}_{\{|D(x)|^d/(2C^*)^d \geq 2k/n\}} e^{-\frac{1-\log 2}{2} n|D(x)|^d/(2C^*)^d} + \mathbb{I}_{\{|D(x)|^d/(2C^*)^d < 2k/n\}} \\
&\leq e^{-(1-\log 2)k} + \mathbb{I}_{\{|D(x)|^d/(2C^*)^d < 2k/n\}}, \qquad (13)
\end{aligned}$$

where the third inequality follows from Chernoff's exponential inequality. Applying the weak margin condition, we get (9) by

$$\begin{aligned}
I_{n,k} &= \int |D(x)| \mathbb{P}\left\{|D(x)|/2 < |D(x) - \overline{D}_{n,k}(x)|\right\} \mu(dx) \\
&\leq e^{-(1-\log 2)k} + O((k/n)^{(\alpha+1)/d}). \qquad (14)
\end{aligned}$$

■

Proof of Proposition 4. The Hoeffding inequality implies

$$\begin{aligned}
&\mathbb{P}\{|D_{n,k}(j) - \overline{D}_{n,k}(j)| \geq |D(j)|/2 \mid X_1, \ldots, X_n\} \\
&= \mathbb{P}\left\{\left|\frac{1}{k}\sum_{i=1}^{k}(Y_{(n,i)}(j) - D(X_{(n,i)}(j)))\right| \geq |D(j)|/2 \mid X_1, \ldots, X_n\right\} \\
&\leq 2e^{-k|D(j)|^2/8}, \quad j \in \mathbb{N},
\end{aligned}$$

from which one gets a rough non-trivial upper bound on the estimation error:

$$\begin{aligned}
\mathbb{E}\left\{\int_{\{|D_{n,k}(x) - \overline{D}_{n,k}(x)| \geq |D(x)|/2\}} |D(x)|\mu(dx)\right\} &\leq 2\sum_{j=1}^{\infty} p_j |D(j)| e^{-k|D(j)|^2/8} \\
&\leq \frac{2\max_{\{0 \leq z\}} z e^{-z^2/8}}{\sqrt{k}}.
\end{aligned}$$

Concerning the approximation error, the modified Lipschitz condition (10) implies

$$\begin{aligned}
|\overline{D}_{n,k}(j) - D(j)| &\leq \frac{1}{k}\sum_{i=1}^{k} |D(X_{(n,i)}(j)) - D(j)| \\
&\leq C^* \frac{1}{k}\sum_{i=1}^{k} \mu\left([j - |X_{(n,i)}(j) - j| + 1, j + |X_{(n,i)}(j) - j| - 1]\right) \\
&\leq C^* \mu\left([j - |X_{(n,k)}(j) - j| + 1, j + |X_{(n,k)}(j) - j| - 1]\right).
\end{aligned}$$

Without loss of generality assume that $C^* > 1/2$ and $|D(j)| > 0$. Introduce the notation

$$\ell_j^* := \min\left\{\ell \in \mathbb{N}; \mu\left([j - \ell + 1, j + \ell - 1]\right) \geq \frac{|D(j)|}{2C^*}\right\}.$$

Because of

$$0 < \frac{|D(j)|}{2C^*} < 1,$$

ℓ_j^* is well defined. Put

$$A_j = \left[j - \ell_j^* + 1, j + \ell_j^* - 1\right].$$

Thus

$$\begin{aligned}
&\mathbb{E}\left\{\int_{\{|\overline{D}_{n,k}(x) - D(x)| \geq |D(x)|/2\}} |D(x)|\mu(dx)\right\} \\
&\leq \sum_{j=1}^{\infty} p_j |D(j)| \mathbb{P}\left\{\mu\left([j - |X_{(n,k)}(j) - j| + 1, j + |X_{(n,k)}(j) - j| - 1]\right) \geq \frac{|D(j)|}{2C^*}\right\} \\
&= \sum_{j=1}^{\infty} p_j |D(j)| \mathbb{P}\left\{|X_{(n,k)}(j) - j| \geq \ell_j^*\right\}.
\end{aligned}$$

If μ_n denotes the empirical distribution for $X_1, \ldots, X_n$, then

$$\begin{aligned}
\mathbb{E}\left\{\int_{\{|\overline{D}_{n,k}(x) - D(x)| \geq |D(x)|/2\}} |D(x)|\mu(dx)\right\} &\leq \sum_{j=1}^{\infty} p_j |D(j)| \mathbb{P}\left\{\sum_{i=1}^{n} \mathbb{I}_{|X_i - j| < \ell_j^*} \leq k\right\} \\
&= \sum_{j=1}^{\infty} p_j |D(j)| \mathbb{P}\left\{\mu_n(A_j) \leq k/n\right\}.
\end{aligned}$$

For the decomposition

$$\mathbb{P}\left\{\mu_n(A_j) \leq k/n\right\} \leq \mathbb{I}_{\mu(A_j) \leq 2k/n} + \mathbb{I}_{\mu(A_j) > 2k/n} \mathbb{P}\left\{\mu_n(A_j) \leq k/n\right\},$$

apply the Bernstein inequality:

$$\begin{aligned}
\mathbb{I}_{\mu(A_j) > 2k/n} \mathbb{P}\left\{\mu_n(A_j) \leq k/n\right\} &= \mathbb{I}_{\mu(A_j) > 2k/n} \mathbb{P}\left\{\mu_n(A_j) - \mu(A_j) \leq k/n - \mu(A_j)\right\} \\
&\leq \mathbb{I}_{\mu(A_j) > 2k/n} \mathbb{P}\left\{\mu_n(A_j) - \mu(A_j) \leq -\mu(A_j)/2\right\} \\
&\leq \mathbb{I}_{\mu(A_j) > 2k/n} e^{-\frac{n\mu(A_j)^2/4}{2(\mu(A_j) + \mu(A_j)/6)}} \\
&= \mathbb{I}_{\mu(A_j) > 2k/n} e^{-3n\mu(A_j)/28} \\
&\leq e^{-3k/14}.
\end{aligned}$$

The definition of A_j implies

$$\mu(A_j) \geq \frac{|D(j)|}{2C^*}.$$

Therefore,

$$\sum_{j=1}^{\infty} p_j |D(j)| \mathbb{I}_{\mu(A_j) \le 2k/n} \le \sum_{j=1}^{\infty} p_j |D(j)| \mathbb{I}_{|D(j)|/(2C^*) \le 2k/n} \le 4C^* k/n.$$

These bounds imply the bound on the approximation error:

$$\mathbb{E}\left\{ \int_{\{|\overline{D}_{n,k}(x) - D(x)| \ge |D(x)|/2\}} |D(x)| \mu(dx) \right\} \le e^{-3k/14} + 4C^* k/n.$$

■

Proof of Theorem 5. Again we use the decomposition (7). Lemma 2 with unchanged proof yields (8). Under the local Lipschitz condition and the weak density condition, we have to prove (9), i.e., (14). Let $\delta > 0$ be from the definition of weak density assumption. We have that

$$\begin{aligned}
&\int |D(x)| \mathbb{P}\left\{ |D(x)|/2 < |D(x) - \overline{D}_{n,k}(x)| \right\} \mu(dx) \\
&\le \int |D(x)| \mathbb{P}\left\{ |D(x)|/2 < \overline{C} f(x)^{1/d} \frac{1}{k} \sum_{i=1}^{k} \|x - X_{(n,i)}(x)\| \right\} \mu(dx) \\
&\le \int |D(x)| \mathbb{P}\left\{ |D(x)|/2 < \overline{C} f(x)^{1/d} \|x - X_{(n,k)}(x)\| \right\} \mu(dx) \\
&\le \int |D(x)| \mathbb{P}\left\{ |D(x)|/2 < \overline{C} \mu(S_{x,\|x - X_{(n,k)}(x)\|})^{1/d} / c_{min} \right\} \mu(dx) \\
&+ \int |D(x)| \mathbb{P}\left\{ f(x)^{1/d} \|x - X_{(n,k)}(x)\| > \delta \right\} \mu(dx).
\end{aligned}$$

The first term of the right hand side is

$$e^{-(1-\log 2)k} + O((k/n)^{(\alpha+1)/d})$$

by the weak margin condition according to (12) and (13). For the second term, we note

$$\begin{aligned}
&\mathbb{P}\left\{ f(x)^{1/d} \|x - X_{(n,k)}(x)\| > \delta \right\} \\
&= \mathbb{P}\left\{ \|x - X_{(n,k)}(x)\| > \delta / f(x)^{1/d} \right\} \\
&= \mathbb{P}\left\{ \sum_{i=1}^{n} \mathbb{I}_{\left\{X_i \in S_{x,\delta/f(x)^{1/d}}\right\}} < k \right\} \\
&\le \mathbb{I}_{\left\{\mu(S_{x,\delta/f(x)^{1/d}}) \ge 2k/n\right\}} \mathbb{P}\left\{ \sum_{i=1}^{n} \mathbb{I}_{\left\{X_i \in S_{x,\delta/f(x)^{1/d}}\right\}} < \frac{n}{2} \mu(S_{x,\delta/f(x)^{1/d}}) \right\} \\
&+ \mathbb{I}_{\left\{\mu(S_{x,\delta/f(x)^{1/d}}) < 2k/n\right\}} \\
&\le \mathbb{I}_{\left\{\mu(S_{x,\delta/f(x)^{1/d}}) \ge 2k/n\right\}} e^{-\frac{1-\log 2}{2} n \mu(S_{x,\delta/f(x)^{1/d}})} + \mathbb{I}_{\left\{\mu(S_{x,\delta/f(x)^{1/d}}) < 2k/n\right\}},
\end{aligned}$$

the latter by Chernoff's exponential inequality. The weak density assumption yields

$$\mathbb{I}_{\left\{\mu(S_{x,\delta/f(x)^{1/d}}) < 2k/n\right\}} \le \mathbb{I}_{\left\{c_{min}^d \delta^d < 2k/n\right\}}.$$

Thus the second term is bounded by

$$e^{-(1-\log 2)k} + \mathbb{I}_{\left\{c_{min}^d \delta^d < 2k/n\right\}} = e^{-(1-\log 2)k},$$

as soon as

$$c_{min}^d \delta^d \ge 2k/n.$$

■

Proof of Theorem 6. With the notation (7), we have

$$\mathbb{E}\{(D_{n,k}(x) - D(x))^2\} = \mathbb{E}\{(D_{n,k}(x) - \overline{D}_{n,k}(x))^2\} + \mathbb{E}\{(\overline{D}_{n,k}(x) - D(x))^2\}.$$

We show that

$$\int \mathbb{E}\left\{ \left(D_{n,k}(x) - \overline{D}_{n,k}(x)\right)^2 \right\} \mu(dx) = (\mathbb{E}\{\sigma^2(X)\} + o(1))/k \tag{15}$$

and

$$\mathbb{E}\left\{ \left(\overline{D}_{n,k}(x) - D(x)\right)^2 \right\} \le 2C^{*2}(k/n)^{2/d}, \tag{16}$$

which imply the assertion of the theorem.

In the proof of Lemma 2 we mentioned that for given $X_1, \ldots, X_n$, the random variable $D_{n,k}(x) - \overline{D}_{n,k}(x)$ is an average of independent random variables with mean zero, therefore

$$\begin{aligned}
&\mathbb{E}\left\{ (D_{n,k}(x) - \overline{D}_{n,k}(x))^2 \mid X_1, \ldots, X_n \right\} \\
&= \mathbb{E}\left\{ \left(\frac{1}{k} \sum_{i=1}^{k} (Y_{(n,i)}(x) - D(X_{(n,i)}(x))) \right)^2 \mid X_1, \ldots, X_n \right\} \\
&= \frac{1}{k^2} \sum_{i=1}^{k} \mathbb{E}\left\{ \left(Y_{(n,i)}(x) - D(X_{(n,i)}(x))\right)^2 \mid X_1, \ldots, X_n \right\} \\
&= \frac{1}{k^2} \sum_{i=1}^{k} \sigma^2(X_{(n,i)}(x)).
\end{aligned}$$

Problems 6.3 and 6.4 in Györfi et al. (2002) together with $k/n \to 0$ imply that

$$\int \mathbb{E}\{(D_{n,k}(x) - \overline{D}_{n,k}(x))^2\} \mu(dx) = \frac{\mathbb{E}\{\sigma^2(X)\} + o(1)}{k},$$

and so (15) is verified. Concerning (16), the modified Lipschitz condition implies that

$$\begin{aligned}\left(\overline{D}_{n,k}(x) - D(x)\right)^2 &= \left(\frac{1}{k}\sum_{i=1}^{k}(D(x) - D(X_{(n,i)}(x)))\right)^2 \\ &\leq \left(\frac{1}{k}\sum_{i=1}^{k}|D(x) - D(X_{(n,i)}(x))|\right)^2 \\ &\leq C^{*2}\left(\frac{1}{k}\sum_{i=1}^{k}\mu(S_{x,\|x-X_{(n,i)}(x)\|})^{1/d}\right)^2 \\ &\leq C^{*2}\mu(S_{x,\|x-X_{(n,k)}(x)\|})^{2/d}.\end{aligned}$$

Thus,

$$\mathbb{E}\left\{\left(\overline{D}_{n,k}(x) - D(x)\right)^2\right\} \leq C^{*2}\mathbb{E}\left\{U_{(k,n)}^{2/d}\right\}.$$

If $d \geq 2$, then the Jensen inequality implies

$$\mathbb{E}\left\{U_{(k,n)}^{2/d}\right\} \leq (k/n)^{2/d},$$

while for $d = 1$, one has

$$\mathbb{E}\left\{U_{(k,n)}^{2}\right\} = \mathbb{V}ar(U_{(k,n)}) + \mathbb{E}\left\{U_{(k,n)}\right\}^2 \leq k/n^2 + (k/n)^2.$$

■

Acknowledgments

The authors would like to thank the Editor and Reviewers for careful reading and comments. These comments and suggestions have been very helpful for revising and improving the manuscript.

The research of L. Györfi and of H. Walk was supported through the programme "Research in Pairs" by the Mathematisches Forschungsinstitut Oberwolfach in 2017. L. Györfi was supported by the National University of Public Service under the priority project KÖFOP-2.1.2-VEKOP-15-2016-00001 titled "Public Service Development Establishing Good Governance" in the Ludovika Workshop.

References

Jean-Yves Audibert and Alexandre B. Tsybakov. Fast learning rates for plug-in classifiers. The Annals of Statistics, 35(2):608—633, 2007.

Gérard Biau and Luc Devroye. *Lectures on the Nearest Neighbor Method*. Springer—Verlag, Cham, 2015.

Peter Binev, Albert Cohen, Wofgang Dahmen and Ronald DeVore, Classification algorithms using adaptive partitioning. *Annals of Statistics* 42:2141—2163, 2014.

Iingrid Blaschzyk and Ingo Steinwart, Improved classification rates under refined margin conditions. *Electronic Journal of Statistics*, 12:793—823, 2018.

Timothy I. Cannings, Thomas B. Berrett and Richard J. Samworth. Local nearest neighbor classification with applications to semi-supervised learning. *arXiv: 1704.00642*, 2017.

Kamalika Chaudhuri and Sanjoy Dasgupta. Rates of convergence for nearest neighbor classification. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence and Kilian Q. Weinberger, editors, *Neural Information Processing Systems.* 27:3437—3445,*arXiv: 1407.0067v2*, 2014.

Luc Devroye, László Györfi and Gábor Lugosi. *A Probabilistic Theory of Pattern Recognition.* Springer—Verlag, New York, 1996.

Maik Döring, László Györfi and Harro Walk. Exact rate of convergence of kernel-based classification rule. In Stan Matwin and Jan Mielniczuk, editors, *Challenges in Statistics and Data Mining.* Studies in Computational Intelligence, 605:71—91, Springer, Cham, 2015.

Sébastien Gadat, Thierry Klein and Clément Marteau. Classification with the nearest neighbor rule in general finite dimensional space. *The Annals of Statistics*, 44(3):982—1009, 2016.

László Györfi, Michael Kohler, Adem Krzyżak and Harro Walk. *A Distribution-Free Theory of Nonparametric Regression.* Springer—Verlag, New York, 2002.

Michael Kohler and Adam Krzyżak. On the rate of convergence of local averaging plug-in classification rules under a margin condition. *IEEE Transactions on Information Theory*, 53(5):1735—1742, 2007.

Michael Kohler, Adam Krzyżak and Harro Walk. Rate of convergence for partitioning and nearest neighbor regression estimates with unbounded data. *Journal of Multivariate Analysis*, 97:311—323, 2006.

Elia Liitiäinen, Francesco Corona and Amaury Lendasse. Residual variance estimation using a nearest neighbor statistic. *Journal of Multivariate Analysis*, 101:811—823, 2010.

Enno Mammen and Alexandre B. Tsybakov. Smooth discrimination analysis. *The Annals of Statistics*, 27(6):1808—1829, 1999.

Richard J. Samworth. Optimal weighted nearest neighbor classifiers. *The Annals of Statistics*, 40(5):2733—2763, 2012a.

Richard J. Samworth. Supplement to Samworth (2012a). arXiv: 1101.5783, 2012b.

Alexandre B. Tsybakov. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1):135—166. 2004.

Journal of Machine Learning Research 18 (2018) 1-50 Submitted 6/16; Revised 6/17; Published 7/18

A Theory of Learning with Corrupted Labels

Brendan van Rooyen BRENDAN.VANROOYEN@OUTLOOK.COM

Robert C. Williamson BOB.WILLIAMSON@ANU.EDU.AU
The Australian National University and Data61
Canberra ACT 2601, Australia

Editor: Inderjit Dhillon

Abstract

It is usual in machine learning theory to assume that the training and testing sets comprise of draws from the same distribution. This is rarely, if ever, true and one must admit the presence of corruption. There are many different types of corruption that can arise and as of yet there is no general means to compare the relative ease of learning in these settings. Such results are necessary if we are to make informed economic decisions regarding the acquisition of data.

Here we begin to develop an abstract framework for tackling these problems. We present a generic method for learning from a fixed, known, *reconstructible* corruption, along with an analyses of its statistical properties. We demonstrate the utility of our framework via concrete novel results in solving supervised learning problems wherein the labels are corrupted, such as learning with noisy labels, semi-supervised learning and learning with partial labels.

Keywords: Supervised Learning, Generalized Supervision, Decision Theory, Minimax Bounds, Data Processing, Noise

1. Introduction

The goal of supervised learning is to find,

$$\arg\min_{f\in\mathcal{F}} \mathbb{E}_{(x,y)\sim P}\ell(y, f(x)),$$

where $\mathcal{F}$ is a hypothesis class of functions and ℓ is loss function. Usually the decision maker is not given direct access to P, but rather a training set comprising n iid samples $\{(x_i, y_i)\}_{i=1}^n$ from P. There are many algorithms for solving this problem, for example empirical risk minimization (ERM), and this problem is well understood.

There are many other types of data one could learn from. For example in semi-supervised learning (Chapelle et al., 2010) the decision maker is given n instance label pairs and m instances devoid of labels. In learning with noisy labels (Angluin and Laird, 1988; Kearns, 1998; Natarajan et al., 2013), the decision maker observes instance label pairs where the observed labels have been corrupted by some noise process. There are many variants including, but not limited to, learning with label proportions (Quadrianto et al., 2009), learning with partial labels (Cour et al., 2011), multiple instance learning (Maron and Lozano-Pérez, 1998) as well as combinations of the above.

Abstractly, all of these problems can be understood as corrupted learning problems, in that the "ideal" clean training set is contaminated by a fixed and in general unknown noise process. What is currently lacking is a general theory of learning from corrupted data, as well as means to *compare* the relative usefulness of different data types. Such a theory is required if one wishes to make informed economic decisions on which data sets to acquire.

To make progress, we consider abstract prediction problems that have been contaminated by a fixed, known corruption. We demonstrate a generic method for learning from corrupted data, if the corruption is *reconstructible*. We show the utility of this framework via concrete novel results in supervised learning problems wherein the label has been corrupted, as in learning with label noise, semi-supervised learning and learning with partial labels.

We do not explicitly consider corruption of the instance nor corruption of the label that depends on the instance. While certain examples of these problems are reconstructible, they are not our interest here. We also do not consider non-reconstructible problems such as multiple instance learning/learning with label proportions.

The concrete contributions of this paper are:

- A general method for learning from corrupted labels based on a generalization of the method of unbiased estimators presented in Natarajan et al. (2013) and implicit in the earlier work of Kearns (1998) (theorems 5 and 6).
- Upper and lower bounds on the risk of learning from combinations of corrupted labels, with some analyses of their tightness (theorems 7 and 16). Our results greatly extend the state of the art of Crammer et al. (2005), both in scope and in completeness.
- Demonstration of the computational feasibility of our approach via the preservation of convexity (theorem 29).

Elements of our framework have appeared elsewhere, for example in Crammer et al. (2005); Cid-Sueiro et al. (2014); Blanchard et al. (2016); Natarajan et al. (2013). While not the complete story for *all* problems, the contributions outlined above make progress toward the final goal of informed economic decisions. The end result is a collection of general tools that allow one to *learn from* and *compare* the usefulness of various corrupted labels.

2. Basic Notation

Let $\mathbb{R}_+$ be the set of non-negative real numbers. Let Y^X be the set of functions with domain X and range Y. For a set X define the functions $\mathrm{id}_X(x) = x$, and $\mathbf{1}_X(x) = 1$. For a function $f \in \mathbb{R}^{X\times Y}$ and $y \in Y$, we denote the *partial* function $f(-, y) \in \mathbb{R}^X$, with $f(-, y)(x) = f(x, y)$, with similar notation for fixing the first argument. We denote the *dual space* of $\mathbb{R}^X$, the set of linear maps $\mathbb{R}^X \to \mathbb{R}$, by $(\mathbb{R}^X)^*$. We take the general view that a probability distribution is an element of the dual.

Definition 1 *A* probability distribution *on a set X is an element of $(\mathbb{R}^X)^*$, i.e. a linear function $\langle P, -\rangle : \mathbb{R}^X \to \mathbb{R}$, such that:*

1. $\langle P, \mathbf{1}_X \rangle = 1$.
2. *If* $f(x) \leq g(x),\ \forall x \in X$ *then* $\langle P, f \rangle \leq \langle P, g \rangle$.

The linear function $\langle P, - \rangle$ is called an expectation. For a large class of general topological spaces, this definition is equivalent to the usual one in terms of measures on sigma algebras (Rudin, 1991). If X is finite, a distribution is nothing more than a vector.

At times we use the standard prefix notation, with $\mathbb{E}_P f = \langle P, f \rangle$. Define the set of all distributions on a set X to be $\mathbb{P}(X)$. For any $x \in X$ define the point mass distribution δ_x, with $\langle \delta_x, f \rangle = f(x)$ for all functions f. Finally, for a boolean predicate $p : X \to \{\mathsf{True}, \mathsf{False}\}$, let $[\![p(x)]\!] = 1$ if $p(x)$ is true and 0 otherwise. Other notation will be developed as necessary.

3. The General Decision Problem

Consider the problem faced by a scientist in a laboratory. In front of them is a beaker, containing an unknown substance. Available to them are a myriad of experiments that can be performed to ascertain its identity. The scientist could attempt to ignite it, mix a bit of it with a known substance and see what happens, x-ray a sample, throw some of it at high velocity toward an oncoming beam of electrons and so on. Due to time and budget constraints, only a limited number of experiments can be performed to ascertain the substance's true identity. Therefore the scientist should focus their effort on the "most informative" experiments. Of course, what is informative depends on how the substance is to be *used*. For example, if the scientist wishes to sprinkle some of it on their food to enhance its flavor, misidentifying arsenic as table salt is a very bad idea. However, if they want to sprinkle it on the snails in their garden, this distinction is less important. The focus of this section is the abstract formulation of this problem. We consider the problem of how a decision maker, or scientist, uses observations from experiments to inform their actions.

Let Θ be a set of possible values of some unknown quantity, and A the set of actions available to the decision maker. The consequence of an action is measured by a loss function $L : \Theta \times A \to \mathbb{R}$. A negative loss represents a gain to the decision maker. In light of our previous example, Θ are the possible substances that could be in the beaker, A is what the decision maker can *do* with the substance (eat it, put it on snails and so on), and L measures the consequence of an action to the scientist ($L(\text{arsenic}, \text{eat})$ should be high). The norm of a loss function is given by its largest possible consequence (positive or negative), $\|L\|_\infty = \max_{\theta,a} |L(\theta, a)|$.

Unknown to the decision maker is the *exact* value of θ. To *discover* this, the decision maker is guided by experiments. Let $\mathcal{Z}$ be the set of possible outcomes of an experiment. We will assume that $\mathcal{Z}$ has enough mathematical structure so as to make sense of the theorems. As is standard in machine learning, the reader can assume that $\mathcal{Z}$ is of the form $X \times Y$ where X is a compact subset of $\mathbb{R}^d$ and Y is a finite set. All of the key ideas can be gleaned from assuming $\mathcal{Z}$ *finite*. This places no restrictions whatsoever one the usefulness of the theory in application to computer science problems. After all, computers work perfectly well with finite state spaces. The outcome of the experiment, $z \in \mathcal{Z}$, is assumed related to the unknown, certain outcomes are more strongly linked to certain values of θ. The relationship between the unknown and the outcome of the experiment is modeled by a *transition*.

3.1 Transitions

Definition 2 *A* transition *from a set X to a set Y is a linear map $T : (\mathbb{R}^X)^* \to (\mathbb{R}^Y)^*$.*

While abstract in appearance, we remark that when X and Y are *finite*, a transition is nothing more than a *matrix*. In general, a transition is an integral operator. Denote the set of all transitions from X to Y by $\mathbb{T}(X, Y)$. We call a transition *Markov* if $T(\mathbb{P}(X)) \subseteq \mathbb{P}(Y)$, ie it maps distributions over X to distributions over Y. When X and Y are finite, Markov transitions are represented by column stochastic matrices.

Markov transitions constitute a modern approach to conditional probability (Chang and Pollard, 1997; Torgersen, 1991; Le Cam, 1964; Chentsov, 1982). The distribution,

$$T(x) := T(\delta_x)$$

is how the decision maker summarizes their uncertainty about Y if the true value of X is x. In fact a transition is *completely determined* by its value on point masses. Every function $\phi \in Y^X$ defines a transition with,

$$\langle \phi(\alpha), f \rangle_Y := \langle \alpha, f \circ \phi \rangle_X\,,\ \forall f \in \mathbb{R}^Y,\ \forall \alpha \in (\mathbb{R}^X)^*.$$

Such a transition is called *deterministic*. Transitions can be combined in *series* and in *parallel*.

For transitions $T \in \mathbb{T}(X, Y)$ and $S \in \mathbb{T}(Y, Z)$ we can define $S \circ T \in \mathbb{T}(X, Z)$ by usual function composition. If X, Y and Z are finite, then this is just matrix multiplication. If T and S are Markov, this is just iterated expectation. Intuitively, this can be seen as "marginalizing" over Y in the Markov chain,

$$X \to Y \to Z.$$

Combination of transitions in series models corruptions that are performed one after another.

Transitions can be combined in *parallel*. For $\alpha, \beta \in (\mathbb{R}^X)^*$, denote the product by $\alpha \otimes \beta \in (\mathbb{R}^{X \times X})^*$. If $T_i \in \mathbb{T}(X_i, Y_i)$, $i \in [1; k]$, are transitions then denote,

$$\otimes_{i=1}^k T_i \in \mathbb{T}(\times_{i=i}^k X_i, \times_{i=1}^k Y_i)$$

with $\otimes_{i=1}^k T_i(x) = T_1(x_1) \otimes \cdots \otimes T_k(x_k)$, where $\times$ denotes the Cartesian product and $\otimes$ denotes products of duals. Transitions can also be *replicated*. For any transition $T \in \mathbb{T}(X, Y)$ we denote the *replicated transition* $T_n \in \mathbb{T}(X, Y^n),\ n \in \{1, 2, \dots\}$, with,

$$T_n(x) := \underbrace{T(x) \otimes \cdots \otimes T(x)}_{n \text{ times}} := T(x)^n,$$

the n-fold product of $T(x)$.

Parallel composition of transitions models performing two different experiments as well as the repeated performance of the same experiment.

3.2 Experiments and Risk

An *experiment* is a Markov transition $e \in \mathbb{T}(\Theta, \mathcal{Z})$. We call $\mathcal{Z}$ the observation space of the experiment. The distribution $e(\theta)$ summarizes uncertainty in the observation when θ is the value of the unknown. After observing the results of an experiment, the decision maker is tasked with choosing a suitable action. They do this via a *learning algorithm*.

In our language, a *learning algorithm* is a Markov transition $\mathcal{A} \in \mathbb{T}(\mathcal{Z}, A)$ [1]. $\mathcal{A}(z)$ summarizes the decision makers uncertainty in which action to choose. We define the *risk*,

$$\text{Risk}_L(\theta, e, \mathcal{A}) := \mathbb{E}_{z\sim e(\theta)}\mathbb{E}_{a\sim\mathcal{A}(z)}L(\theta, a) = \langle \mathcal{A} \circ e(\theta), L(\theta, -)\rangle\,.$$

The risk measures the quality of the final action chosen by the decision maker when they use the learning algorithm $\mathcal{A}$, after performing experiment e, assuming θ is the true value of the unknown. The risk does not provide a single number for the comparison of experiments, rather it provides an entire risk profile. To compare experiments directly we use the *minimax risk*,

$$\underline{\text{Risk}}_L(e) := \inf_{\mathcal{A}} \sup_{\theta} \text{Risk}_L(\theta, e, \mathcal{A}),$$

3.3 The Standard Prediction Problem

The preceding framework for defining and measuring the value of different experiments was largely conceived in the field of theoretical statistics, in the works of von Neumann and Morgenstern (1947); Blackwell (1951); DeGroot (1962) and Le Cam (1964). There is a perceived tension between the goals of statistics, that is to *discover* θ, versus the goals of machine learning, that is to *predict* the outcomes of the experiment e. As we now show, this distinction is only superficial. Let $\ell : \mathcal{Z} \times A \to \mathbb{R}$ be a "predictive" loss, that measures how well the action a predicts the outcome of the experiment e. Common examples include:

	Observations $\mathcal{Z}$	Actions A	Loss ℓ
Density Estimation	$X \subseteq \mathbb{R}^d$	Model $\Theta \subseteq P(X)$	$-\log(P_\theta(x))$
Classification	$X \times \{\pm 1\}$	Function class $\mathcal{F} \subseteq \{\pm 1\}^X$	$[\![y = f(x)]\!]$
Regression	$X \times \mathbb{R}$	Function class $\mathcal{F} \subseteq \mathbb{R}^X$	$(y - f(x))^2$
Supervised Learning	$X \times Y$	Function class $\mathcal{F} \subseteq \mathbb{P}(Y)^X$	$\ell(y, f(x))$

Due to the frequency with which we take expectations we overload our notation and define,

$$\ell(P, Q) := \mathbb{E}_{z\sim P}\mathbb{E}_{a\sim Q}\ell(z, a).$$

In learning theory, $\ell(P, Q)$ as defined above is refered to as the *risk*, here we reserve the term *risk* for its more classical statistical definition. Let $\Theta = \mathbb{P}(\mathcal{Z})$, the set of all possible distributions on the observation space. The loss function $L : \mathbb{P}(\mathcal{Z}) \times \mathbb{P}(A) \to \mathbb{R}$ of interest is the *regret*,

$$L(P, Q) := \ell(P, Q) - \inf_{a\in A} \ell(P, a),$$

Let e_n be the experiment that maps each $P \in \mathbb{P}(\mathcal{Z})$ to its n-fold product P^n. Then,

$$\underline{\text{Risk}}_L(e_n) = \inf_{\mathcal{A}} \sup_{P} \mathbb{E}_{S\sim P^n}\ell(P, \mathcal{A}(S)) - \inf_{a\in A} \ell(P, a),$$

1. We adopt the common terminology of algorithm, while not addressing computational issues.

is the minimax regret, a central object in learning theory. Therefore, the *prediction* problem of machine learning can be understood in the more general language of experiments. It is these problems that are of central interest in this paper. For the remainder L will represent the regret for a loss ℓ.

4. Corrupted Prediction Problems

Due to limitations in the measurement apparatus available to the decision maker, rather than observing $z \in \mathcal{Z}$, it is often the case that the decision maker observes a corrupted $\tilde{z}$ in a potentially different observation space $\tilde{\mathcal{Z}}$. We model the corruption process via a Markov transition $T \in \mathbb{T}(\mathcal{Z}, \tilde{\mathcal{Z}})$.

Definition 3 *A* corruption *is a Markov transition* $\mathbb{T}(\mathcal{Z}, \tilde{\mathcal{Z}})$.

For example, we may wish to learn a relationship between measured symptoms and a medical diagnosis, as provided to us by an expert. To do so, rather than access to the expert's data, we are given data from one of their apprentices. Here T models the hypothesized link between the expert's and apprentice's data. The goal of predicting as well as the expert remains.

More concretely, we are interested in supervised learning problems wherein,

$$\mathcal{Z} = X \times Y \text{ and } \tilde{\mathcal{Z}} = X \times \tilde{Y},$$

with T corrupting only the label Y and acting as the identity on the instance X,

$$T = id_X \otimes T_Y.$$

Prominent examples include:

- **Learning with Label Noise**:

$$T_Y = \begin{pmatrix} 1 - \sigma_{-1} & \sigma_1 \\ \sigma_{-1} & 1 - \sigma_1 \end{pmatrix}.$$

- **Semi Supervised Learning**:

$$T_Y = \begin{pmatrix} \sigma_{-1} & 0 \\ 0 & \sigma_1 \\ 1 - \sigma_{-1} & 1 - \sigma_1 \end{pmatrix}$$

- **Learning with partial labels**: We assume that a partial label always includes the true label as one of the possibilities and furthermore that spurious labels are added with probability σ.

$$T_Y = \begin{pmatrix} (1-\sigma)^2 & 0 & 0 \\ 0 & (1-\sigma)^2 & 0 \\ 0 & 0 & (1-\sigma)^2 \\ (1-\sigma)\sigma & (1-\sigma)\sigma & 0 \\ (1-\sigma)\sigma & 0 & (1-\sigma)\sigma \\ 0 & (1-\sigma)\sigma & (1-\sigma)\sigma \\ \sigma^2 & \sigma^2 & \sigma^2 \end{pmatrix}.$$

For convenience we define the corrupted experiment $\tilde{e} = T \circ e$. We order the utility of different corruptions via the minimax risk,

$$\underline{\text{Risk}}_L(\tilde{e}_n) := \min_{\mathcal{A}} \max_{P} \text{Risk}_L(P, \tilde{e}_n, \mathcal{A}).$$

Note that the domain of $\mathcal{A}$ is now $\tilde{\mathcal{Z}}^n$. Ideally we wish to compare $\underline{\text{Risk}}_L(\tilde{e}_n)$ with $\underline{\text{Risk}}_L(e_n)$, the minimum risk of the corrupted and the clean experiments. By the general data processing theorem, $\underline{\text{Risk}}_L(\tilde{e}_n) \geq \underline{\text{Risk}}_L(e_n)$, however this does not allow one to *rank* the utility of *different* T.

Even after many years of directed research, in general we can not compute $\underline{\text{Risk}}_L(e_n)$ exactly, let alone $\underline{\text{Risk}}_L(\tilde{e}_n)$ for general corruptions. Consequently our effort for the remaining turns to upper and lower bounds of $\underline{\text{Risk}}_L(\tilde{e}_n)$.

4.1 Corruption Corrected Losses

When convenient we use the shorthand $T(P) = \tilde{P}$. Natarajan et al. (2013) introduced a method of learning classifiers from data subjected to label noise, called the "method of unbiased estimators". Here we show that this method can be generalized to other corruptions.

Recall that a transition $T \in \mathbb{T}(\mathcal{Z}, \tilde{\mathcal{Z}})$ is a linear map $T : (\mathbb{R}^{\mathcal{Z}})^* \to (\mathbb{R}^{\tilde{\mathcal{Z}}})^*$. Associated with *any* transition is a *dual* or *adjoint* linear map $T^* : \mathbb{R}^{\tilde{\mathcal{Z}}} \to \mathbb{R}^{\mathcal{Z}}$ with,

$$\left\langle \alpha, T^*(\tilde{f}) \right\rangle_{\mathcal{Z}} := \left\langle T(\alpha), \tilde{f} \right\rangle_{\tilde{\mathcal{Z}}}, \ \forall \tilde{f} \in \mathbb{R}^{\tilde{\mathcal{Z}}}, \ \forall \alpha \in (\mathbb{R}^{\mathcal{Z}})^*,$$

In words, T^* "pulls back" functions of the corrupted sample to functions of the clean sample. When T is a matrix, T^* is the *transpose* of T. We wish to go in the other direction, to *transfer* functions of clean samples to those of corrupted samples.

Definition 4 *A transition $T \in \mathbb{T}(\mathcal{Z}, \tilde{\mathcal{Z}})$ is* reconstructible *if T has a left inverse; that is there exists a transition $R \in \mathbb{T}(\tilde{\mathcal{Z}}, \mathcal{Z})$ such that $R \circ T = \text{id}_{(\mathbb{R}^{\mathcal{Z}})^*}$.*

Intuitively, T is reconstructible if there is some transformation that "undoes" the effects of T. Note that R need not be Markov for Markov T. We denote the set of all reconstructible transitions by $\mathbb{T}_{\leftarrow}(\mathcal{Z}, \tilde{\mathcal{Z}})$.

Many forms of corrupted learning are reconstructible, including semi-supervised learning, learning with label noise and learning with partial labels for all but a few pathological cases. Appendix A contains several worked examples.

We call a left inverse of T a *reconstruction*. For concreteness one can always take,

$$R = (T^*T)^{-1}T^*,$$

the Moore-Penrose pseudo inverse of T. For *invertible* T, R is given by the standard inverse. In general it will be useful to consider other reconstructions. In particular, for the proof of theorem 29, we use reconstructions with,

$$R^*(\mathbf{1}_{\mathcal{Z}}) = \mathbf{1}_{\tilde{\mathcal{Z}}},$$

ie those which preserve the *constant* loss function. This condition is *always* satisfied when T is invertible. More broadly, there is no loss in generality working with such reconstructions.

Reconstructible transitions are exactly those where we can *transfer* a function of the clean z to one of the corrupted $\tilde{z}$ while preserving expectations. By properties of adjoints,

$$\langle P, f \rangle = \langle R \circ T(P), f \rangle = \langle T(P), R^*(f) \rangle .$$

In words, to take expectations of f with samples from $\tilde{P}$, we use the *corruption corrected* $\tilde{f} = R^*(f)$. Recall the partial loss function $\ell(-, a) \in \mathbb{R}^{\mathcal{Z}}$. Using R we can reconstruct the partial loss from corrupted examples.

Theorem 5 (Corruption Corrected Loss) *For all reconstructible corruptions $T \in \mathbb{T}(\mathcal{Z}, \tilde{\mathcal{Z}})$ and loss functions $\ell : \mathcal{Z} \times A \to \mathbb{R}$ define the* corruption corrected *loss $\ell_R : \tilde{\mathcal{Z}} \times A \to \mathbb{R}$ with,*

$$\ell_R(-, a) = R^*(\ell(-, a)), \ \forall a \in A.$$

Then for all distributions $P \in \mathbb{P}(\mathcal{Z})$, $\mathbb{E}_{z \sim P} \ell(z, a) = \mathbb{E}_{\tilde{z} \sim \tilde{P}} \ell_R(\tilde{z}, a)$.

4.1.1 USES OF CORRUPTION CORRECTED LOSSES IN SUPERVISED LEARNING

In supervised learning $\mathcal{Z} = X \times Y$ and the goal is to find a function that predicts $y \in Y$ from $x \in X$ with low expected loss. Given a suitable function class $\mathcal{F} \subseteq A^X$ and a loss $\ell : Y \times A \to \mathbb{R}$ one attempts to find,

$$f^* = \arg\min_{f \in \mathcal{F}} \mathbb{E}_{(x,y) \sim P} \ell(y, f(x)).$$

If the labels have been corrupted by $T_Y \in \mathbb{T}(Y, \tilde{Y})$, we can correct for the corruption and find,

$$f^* = \arg\min_{f \in \mathcal{F}} \mathbb{E}_{(x,\tilde{y}) \sim \tilde{P}} \ell_{R_Y}(\tilde{y}, f(x)) = \arg\min_{f \in \mathcal{F}} \mathbb{E}_{(x,y) \sim P} \ell(y, f(x)).$$

4.1.2 A WORKED EXAMPLE: LEARNING WITH SYMMETRIC LABEL NOISE

When learning under symmetric label noise, the decision maker is required to predict a binary label $y \in \{-1, 1\}$, where $y \sim P$. Rather than observing the true y, the decision maker observes $\tilde{y}$, where $\tilde{y} = y$ with probability $1 - \sigma$ and $\tilde{y} = -y$ with probability σ. This process can be modeled by the following corruption and reconstruction respectively;

$$T = \begin{pmatrix} 1-\sigma & \sigma \\ \sigma & 1-\sigma \end{pmatrix}, \ R^* = \frac{1}{1-2\sigma} \begin{pmatrix} 1-\sigma & -\sigma \\ -\sigma & 1-\sigma \end{pmatrix}.$$

Note that R is *not* Markov, as some of the entries of R are negative. For a loss $\ell : \{-1, 1\} \times A \to \mathbb{R}$ we have,

$$\begin{pmatrix} \ell_R(-1, a) \\ \ell_R(1, a) \end{pmatrix} = \frac{1}{1-2\sigma} \begin{pmatrix} 1-\sigma & -\sigma \\ -\sigma & 1-\sigma \end{pmatrix} \begin{pmatrix} \ell(-1, a) \\ \ell(1, a) \end{pmatrix},$$

or more compactly,

$$\ell_R(y, a) = \frac{(1-\sigma)\ell(y, a) - \sigma\ell(-y, a)}{1-2\sigma}.$$

This is equivalent to the "method of unbiased estimators" of Natarajan et al. (2013). Several examples of corruption corrected losses are given in appendix A.

4.1.3 WHEN TO APPEAL TO A SURROGATE LOSS

Many loss functions in machine learning are non convex, the zero one loss a prime example. Rather than minimizing one of these losses directly, it is common to instead minimize a convex surrogate loss function, such as the hinge loss. When learning in the presence of corruption we face a dilemma; when does one appeal to a surrogate? The method of unbiased estimators *first* appeals to a surrogate loss before correcting for noise. The method of label dependent costs (Natarajan et al., 2013) first corrects for noise before appealing to a surrogate.

4.2 Similarities with Other Frameworks

The framework here shares common points with two other broad directions in the literature, in the line of working of Cid-Sueiro et al. (2014) and Blanchard et al. (2016).

Cid-Sueiro et al. (2014); Cid-Sueiro (2012) consider the problem of learning when the corruption is only partially known. They also use transitions (stochastic matrices) and their reconstructions to construct loss functions that "correct for" noise in the labeling process. Their development closely mirrors that of section 11, with subtle differences (see section 11.3).

Blanchard et al. (2016) consider the closely related problem of learning under mutually contaminated distributions. For example, in binary classification many performance measures can be optimized with access to samples from $P_1, P_{-1} \in \mathbb{P}(X)$, the distribution over instances given a positive or negative label respectively. In practice however the samples can be *mixed*, yielding distributions,

$$\tilde{P}_{\pm 1} = (1 - \pi_{\pm 1})P_1 + \pi_{\pm 1}P_{-1}.$$

In general the mixing constants $\pi_{\pm 1}$ are unknown. Under the assumption that P_1 and P_{-1} are *mutually irreducible*, Blanchard et al. (2016) shows that one can reconstruct the clean distributions together with the mixing coefficients, providing means to estimate the corruption. Their framework has been extended to other problems (Katz-Samuels and Scott, 2016).

5. Upper Bounds for Corrupted Learning

In this section we develop upper bounds on the risk of any algorithm that learns from a corrupted sample, in terms of the *sample risk*, $\tilde{S} \sim \tilde{P}^n$. For simplicity we assume that A is finite. This assumption can be removed by PAC-Bayesian bounds (as we do in the appendix), via covering number arguments (Bartlett et al., 1997) or via more refined bounds from empirical process theory (Bartlett and Mendelson, 2006).

By an application of the PAC-Bayes bound (Zhang, 2006), one has for all algorithms $\mathcal{A} \in \mathbb{T}(\tilde{\mathcal{Z}}^n, A)$ and distributions $P \in \mathbb{P}(\mathcal{Z})$,

$$\mathbb{E}_{\tilde{S}\sim\tilde{P}^n}\ell_R(\tilde{P}, \mathcal{A}(\tilde{S})) \leq \mathbb{E}_{\tilde{S}\sim\tilde{P}^n}\ell_R(\tilde{S}, \mathcal{A}(\tilde{S})) + \|\ell_R\|_\infty\sqrt{\frac{2\log(|A|)}{n}}.$$

By the construction of ℓ_R, $\ell_R(\tilde{P}, \mathcal{A}(\tilde{S})) = \ell(P, \mathcal{A}(\tilde{S}))$, and the above bound yields the following theorem.

Theorem 6 *For all reconstructible $T \in \mathbb{T}(\mathcal{Z}, \tilde{\mathcal{Z}})$, algorithms $\mathcal{A} \in \mathbb{T}(\tilde{\mathcal{Z}}^n, A)$, distributions $P \in \mathbb{P}(\mathcal{Z})$ and bounded loss functions ℓ,*

$$\mathbb{E}_{\tilde{S}\sim\tilde{P}^n}\ell(P, \mathcal{A}(\tilde{S})) \leq \mathbb{E}_{\tilde{S}\sim\tilde{P}^n}\ell_R(\tilde{S}, \mathcal{A}(\tilde{S})) + \|\ell_R\|_\infty\sqrt{\frac{2\log(|A|)}{n}}.$$

A similar result also holds with high probability on draws from $\tilde{P}^n$. This bound motivates the following algorithm for learning from corrupted data,

$$\mathcal{A}_{ERM}(\tilde{S}) = \arg\min_{a\in A} \ell_R(\tilde{S}, a).$$

As this algorithm minimizes the loss on the sample,

$$\mathbb{E}_{\tilde{S}\sim\tilde{P}^n}\ell_R(\tilde{S}, \mathcal{A}_{ERM}(\tilde{S})) \leq \mathbb{E}_{\tilde{S}\sim\tilde{P}^n}\ell_R(\tilde{S}, a) = \ell(P, a),\ \forall a \in A.$$

Together with theorem 6 we have,

$$\mathbb{E}_{\tilde{S}\sim\tilde{P}^n}\ell(P, \mathcal{A}_{ERM}(\tilde{S})) \leq \inf_a \ell(P, a) + \|\ell_R\|_\infty\sqrt{\frac{2\log(|A|)}{n}},\ \forall P$$

yielding,

$$\underline{\text{Risk}}_L(\tilde{e}_n) \leq \|\ell_R\|_\infty\sqrt{\frac{2\log(|A|)}{n}}.$$

Similarly,

$$\underline{\text{Risk}}_L(e_n) \leq \|\ell\|_\infty\sqrt{\frac{2\log(|A|)}{n}}.$$

Therefore, the ratio $\frac{\|\ell_R\|_\infty}{\|\ell\|_\infty}$ measures the relative difficulty of corrupted versus clean learning, as judged solely by our upper bound.

5.1 Upper Bounds for Combinations of Corrupted Data

Recall that our final goal is to quantify the utility of a data set comprising different corrupted data. For example in learning with noisy labels out of n data, there could be n_1 clean, n_2 slightly noisy and n_3 very noisy samples and so on. More generally we assume access to a corrupted sample $\tilde{S}$, made up of k different types of corrupted data, with $\tilde{S}_i \sim \tilde{P}^{n_i}$, $i \in [1; k]$.

Theorem 7 *Let $T_i \in \mathbb{T}(\mathcal{Z}, \tilde{\mathcal{Z}}_i)$ be a collection of k reconstructible corruptions. Let $\tilde{P} = \otimes_{i=i}^k \tilde{P}_i^{n_i}$, $\tilde{\mathcal{Z}} = \times_{i=1}^k \tilde{\mathcal{Z}}_i^{n_i}$, $n = \sum_{i=1}^k n_i$ and $r_i = \frac{n_i}{n}$. Then for all algorithms $\mathcal{A} \in \mathbb{T}(\tilde{\mathcal{Z}}, A)$, distributions $P \in \mathbb{P}(\mathcal{Z})$ and bounded loss functions ℓ,*

$$\mathbb{E}_{\tilde{S}\sim\tilde{P}}\ell(P, \mathcal{A}(\tilde{S})) \leq \mathbb{E}_{\tilde{S}\sim\tilde{P}}\sum_{i=1}^k r_i\ell_{R_i}(\tilde{S}_i, \mathcal{A}(\tilde{S})) + K\sqrt{\frac{2\log(|A|)}{n}},$$

where $K = \sqrt{\sum_{i=1}^k r_i\|\ell_{R_i}\|_\infty^2}$.

A similar result also holds with high probability on draws from $\tilde{P}$. Theorem 7 is a generalization of the final bound appearing in Crammer et al. (2005) that only pertains to symmetric label noise and binary classification. Theorem 7 suggests the following means of choosing n_i examples from each of the corrupted experiments. Let c_i be the cost of acquiring data corrupted by T_i. First, choose data from the T_i with lowest $c_i \|\ell_{R_i}\|_\infty^2$ until picking more violates the budget constraint. Then choose data from the second lowest and so on.

One must be careful when comparing upper bounds, as there may exist alternate methods for learning from the corrupted sample with better properties. In the next section we present arguments indicating this is not the case.

6. Lower Bounds for Corrupted Learning

Thus far we have developed upper bounds for ERM algorithms. In particular we have found that reconstructible corruption does not affect the *rate* at which learning occurs, it only affects constants in the upper bound. Can we do better? Are these constants *tight*? To answer this question we develop lower bounds for corrupted learning.

Here we review Le Cam's method (see Yu (1997) for a more detailed account), a powerful technique for generating lower bounds for decision problems that very often gives the correct rate and dependence on constants (including being able to reproduce the standard VC dimension lower bounds for classification presented in Massart and Nédélec (2006)). In recent times it has been used to establish lower bounds for: differentially private learning (Duchi et al., 2013), learning in a distributed set up (Zhang et al., 2013), function evaluations required in convex optimization (Agarwal et al., 2012) as well as generic lower bounds in statistical estimation problems (Yang and Barron, 1999). We show how this method can be extended using the strong data processing theorem (Cohen and Kempermann, 1998) to provide a general tool for lower bounding the possible performance attainable in corrupted prediction problems.

We stress that these techniques apply to general experiments $e \in \mathbb{T}(\Theta, \mathcal{Z})$, and general loss functions $L : \Theta \times A \to \mathbb{R}$, and not only the predictive problems of interest here.

6.1 Le Cam's Method and Minimax Lower Bounds

Le Cam's method proceeds by reducing a general decision problem to an easier binary classification problem, before relating the best possible performance on this classification problem to the minimax risk. Let Θ be a set of unknowns, $e \in \mathbb{T}(\Theta, \mathcal{Z})$ an experiment and $L : \Theta \times A \to \mathbb{R}$ a loss. We assume further that $\inf_a L(\theta, a) = 0$ for all $\theta \in \Theta$. Define the *separation* $\rho : \Theta \times \Theta \to \mathbb{R}$,

$$\rho(\theta_1, \theta_2) := \inf_a L(\theta_1, a) + L(\theta_2, a).$$

The separation measures how hard it is to act well against both θ_1 and θ_2 simultaneously. Furthermore define the *variational divergence*,

$$V(P, Q) := \sup_{f \in [-1,1]^{\mathcal{Z}}} \mathbb{E}_P f - \mathbb{E}_Q f, \ \forall P, Q \in \mathbb{P}(\mathcal{Z}),$$

The variational divergence measures how hard to is to distinguish two distributions P and Q and is deeply related to binary classification.

Lemma 8 *For all experiments $e \in \mathbb{T}(\Theta, \mathcal{Z})$ and loss functions L,*

$$\underline{\text{Risk}}_L(e) \geq \frac{1}{4} \sup_{\theta_1, \theta_2 \in \Theta} \rho(\theta_1, \theta_2) \left(1 - V\left(e(\theta_1), e(\theta_2)\right)\right).$$

A compact proof is given in appendix B. This lower bound is a trade off between distances measured by ρ and statistical distances measured by the variational divergence. A decision problem is easy if proximity in variational divergence of $e(\theta_1)$ and $e(\theta_2)$ (hard to distinguish θ_1 and θ_2 statistically) implies proximity of θ_1 and θ_2 in ρ (hard to distinguish θ_1 and θ_2 with actions).

Lemma 8 suggests that to *rank* the difficulty of various experiments, one should work with their *Le Cam function.*

Definition 9 *Let e be an experiment and L a loss. The* Le Cam *function of e is,*

$$\text{Le Cam}_L(e, \gamma) = \frac{1}{4} \sup_{\theta_1, \theta_2 \in \Theta} \rho(\theta_1, \theta_2) \left(1 - \gamma V\left(e(\theta_1), e(\theta_2)\right)\right), \ \forall \gamma \geq 0.$$

Lemma 8 can be restated as,

$$\underline{\text{Risk}}_L(e) \geq \text{Le Cam}_L(e, 1).$$

REPLICATION AND RATES

We wish to lower bound how the risk decreases as n, the number of times the experiment is replicated, grows. The following lemma provides a simple way to do this.

Lemma 10 *For all collections of distributions $P_i, Q_i \in \mathbb{P}(\mathcal{Z}_i)$, $i \in [1; k]$,*

$$V(\otimes_{i=1}^k P_i, \otimes_{i=1}^k Q_i) \leq \sum_{i=1}^k V(P_i, Q_i).$$

We make use of the specific case where $P_i = P$ and $Q_i = Q$ for all i. Lemma 8 and lemma 10 yield the following.

Lemma 11 *For all experiments $e \in \mathbb{T}(\Theta, \mathcal{Z})$, loss functions L and n,*

$$\underline{\text{Risk}}_L(e_n) \geq \text{Le Cam}_L(e_n, 1) \geq \text{Le Cam}_L(e, n).$$

OTHER METHODS FOR OBTAINING MINIMAX LOWER BOUNDS

There are many other techniques for constructing lower bounds in terms of functions of pairwise KL divergences (Yu, 1997) as well as functions of pairwise f-divergences (Guntuboyina, 2011). Ultimately these methods replace the Variational divergence and separation in lemma 11 with a more general function of the experiment. While such methods are often required to get tighter lower bounds, all of what follows can be applied to these more intricate techniques. For the sake of conceptual clarity we proceed with Le Cam's method.

6.2 Measuring the Amount of Corruption

Rather than the experiment e, in corrupted learning we work with the corrupted experiment $\tilde{e}$. The data processing theorem for f-divergences states that,

$$V(T(P), T(Q)) \leq V(P, Q), \ \forall P, Q \in \mathbb{P}(\mathcal{Z}).$$

Thus any lower bound achieved by Le Cam's method for e can be directly transferred to one for $\tilde{e}$. However, this provides us with no means to rank different T. For some T, the data processing theorem can be *strengthened.*

Definition 12 *The* Clarity *of a Markov transitions $T \in \mathbb{T}(\mathcal{Z}, \tilde{\mathcal{Z}})$ is,*

$$\text{Clarity}(T) := \sup_{P,Q \in \mathbb{P}(\mathcal{Z})} \frac{V(T(P), T(Q))}{V(P, Q)}.$$

One has,

$$V(T(P), T(Q)) \leq \text{Clarity}(T) V(P, Q), \ \forall P, Q \in \mathbb{P}(\mathcal{Z}).$$

$\text{Clarity}(T)$ measures how much T corrupts. For example, if T is constant and maps all P to the same distribution, then $\text{Clarity}(T) = 0$. If T is an invertible function, then $\text{Clarity}(T) = 1$. When $\mathcal{Z}$ and $\tilde{\mathcal{Z}}$ are finite,

$$\text{Clarity}(T) = \sup_{v \in \Omega} \frac{\|T(v)\|_1}{\|v\|_1},$$

where $\Omega = \{v : \sum v_i = 0, v \neq 0\}$. Hence $\text{Clarity}(T)$ is the operator 1-norm of T when restricted to Ω. Clarity behaves as expected under composition.

Lemma 13 *For all transitions $T_1 \in \mathbb{T}(\mathcal{Z}, \tilde{\mathcal{Z}}_1)$ and $T_2 \in \mathbb{T}(\tilde{\mathcal{Z}}_1, \tilde{\mathcal{Z}}_2)$,*

$$\text{Clarity}(T_2 \circ T_1) \leq \text{Clarity}(T_2)\text{Clarity}(T_1) \leq \min(\text{Clarity}(T_2), \text{Clarity}(T_1)).$$

Hence $T_2 \circ T_1$ is at least as corrupt as either of the T_i.

$\text{Clarity}(T)$ was first used by Dobrushin (1956), where it is called the coefficient of ergodicity and is used to prove rates of convergence of Markov chains to their stationary distribution.

6.3 Lower bounds Relative to the Amount of Corruption

Together with lemma 11, the Clarity leads to meaningful lower bounds that allow the comparison of different corrupted experiments.

Theorem 14 *For all experiments e, loss functions L, corruptions $T \in \mathbb{T}(\mathcal{Z}, \tilde{\mathcal{Z}})$ and $\gamma \geq 0$,*

$$\text{Le Cam}_L\left(T \circ e, \gamma\right) \geq \text{Le Cam}_L\left(e, \text{Clarity}(T)\gamma\right).$$

The proof is a simple application of the strong data processing theorem. Together with lemma 11, the above lemma yields the following corollary.

Corollary 15 *For all experiments e, loss functions L, corruptions $T \in \mathbb{T}(\mathcal{Z}, \tilde{\mathcal{Z}})$ and n,*

$$\underline{\text{Risk}}_L(\tilde{e}_n) \geq \text{Le Cam}_L\left(e, \text{Clarity}(T)n\right)$$

In particular if Le Cam's method yields a lower bound of $\frac{C}{\sqrt{n}}$ for the clean problem, as is usual for many machine learning problems, theorem 15 yields a lower bound of $\frac{C}{\sqrt{\text{Clarity}(T)n}}$ for the corrupted problem. The *rate* at which one learns is unaffected, only the constants. A penalty of $\text{Clarity}(T)$ is unavoidable no matter what learning algorithm is used.

6.4 Lower Bounds for Combinations of Corrupted Data

As in section 5.1 we present lower bounds for combinations of corrupted data. For example in learning with noisy labels out of n data, there could be n_1 clean, n_2 slightly noisy and n_3 very noisy samples and so on.

Theorem 16 *Let $T_i \in \mathbb{T}(\mathcal{Z}, \tilde{\mathcal{Z}}_i)$, $i \in [1; k]$, be k corruptions. Let $T = \otimes_{i=1}^k T_i^{n_i}$ with $n = \sum_{i=1}^k n_k$. Then,*

$$\text{Le Cam}_L\left(T \circ e_n, \gamma\right) \geq \text{Le Cam}_L\left(e, \left(\sum_{i=1}^n n_i \text{Clarity}(T_i)\right)\gamma\right).$$

Furthermore,

$$\underline{\text{Risk}}_L(T \circ e_n) \geq \text{Le Cam}_L\left(e, \left(\sum_{i=1}^n n_i \text{Clarity}(T_i)\right)\right).$$

As in section 5.1, this bound suggests means of choosing data sets via the following integer program,

$$\arg\max_{n_1, n_2 \ldots n_k} \sum_{i=1}^k \text{Clarity}(T_i) n_i \ \text{ subject to } \sum_{i=1}^k c_i n_i \leq C,$$

where c_i is the cost of acquiring data corrupted by T_i and C is the maximum total cost. This is exactly the unbounded knapsack problem (Dantzig, 1957) which admits the following near optimal greedy algorithm. First, choose data from the T_i with highest $\frac{\text{Clarity}(T_i)}{c_i}$ until picking more violates the constraints. Then pick from the second highest and so on.

6.5 The Generality of Clarity

In light of section 6.1, it is often the case that more complicated lower bounding techniques based on pairwise f-divergences are required to produce *tight* lower bounds. Recall the definition of an f-divergence (Ali and Silvey, 1966).

Definition 17 *Let $f : \mathbb{R}_+ \to \mathbb{R}$ be a convex function with $f(1) = 0$. For all distributions $P, Q \in \mathbb{P}(\mathcal{Z})$ the* f-divergence *between P and Q is,*

$$D_f(P, Q) := \mathbb{E}_P f\left(\frac{dQ}{dP}\right),$$

if P and Q are absolutely continuous*, and is infinite otherwise.*

Both the variational and KL divergence are examples of f divergences. Following on from the reasoning of section 6.2, we seek a $\text{Clarity}_f(T)$ such that,

$$D_f(T(P), T(Q)) \leq \text{Clarity}_f(T) D_f(P, Q) \ \forall P, Q, f.$$

On the surface the choice of f matters. However, the clarity as we have defined is *generic*.

Theorem 18 (Strong Data Processing(Theorem 4.1 of Cohen et al. (1993))) *Let $T \in \mathbb{T}(\mathcal{Z}, \tilde{\mathcal{Z}})$ be a Markov transition. Then for all P, Q, f,*

$$D_f(T(P), T(Q)) \leq \text{Clarity}(T) D_f(P, Q).$$

7. Measuring the Tightness of the Upper Bounds and Lower Bounds

In the previous sections we have shown upper bounds that depend on $\|\ell_R\|_\infty$ as well as lower bounds that depend on $\text{Clarity}(T)$. Here we compare these bounds.

Recall from theorem 5, $\ell_R(-, a) = R^*(\ell(-, a))$. The worst case ratio $\frac{\|\ell_R\|_\infty}{\|\ell\|_\infty}$ is determined by the *operator norm* of R^*. For a linear map $R : \mathbb{R}^X \to \mathbb{R}^Y$ define,

$$\|R\|_1 := \sup_{v \in \mathbb{R}^X} \frac{\|Rv\|_1}{\|v\|_1} \text{ and } \|R\|_\infty := \sup_{v \in \mathbb{R}^X} \frac{\|Rv\|_\infty}{\|v\|_\infty}$$

which are two operator norms of R. They are equal to the maximum absolute column and row sum of R respectively (Bernstein, 2009). Hence $\|R\|_1 = \|R^*\|_\infty$.

Lemma 19 *For all losses ℓ, $T \in \mathbb{T}(\mathcal{Z}, \tilde{\mathcal{Z}})$ and reconstructions R, $\frac{\|\ell_R\|_\infty}{\|\ell\|_\infty} \leq \|R^*\|_\infty$.*

Lemma 20 *If $T \in \mathbb{T}(\mathcal{Z}, \tilde{\mathcal{Z}})$ is reconstructible, with reconstruction R, then,*

$$\frac{1}{\text{Clarity}(T)} \leq 1 / \left(\inf_{u \in \mathbb{R}^X} \frac{\|Tu\|_1}{\|u\|_1} \right) \leq \|R^*\|_\infty.$$

Note that for lower bounds we look at the *best* case separation of columns of T, for upper bounds we essentially use the *worst*. We also get the following compositional theorem.

Lemma 21 *If $T_1 \in \mathbb{T}(\mathcal{Z}, \tilde{\mathcal{Z}}_1)$ and $T_2 \in \mathbb{T}(\tilde{\mathcal{Z}}_1, \tilde{\mathcal{Z}}_2)$ are reconstructible, with reconstructions R_1 and R_2, then $T_2 \circ T_1$ is reconstructible with reconstruction $R_1 \circ R_2$. Furthermore,*

$$\frac{1}{\text{Clarity}(T_1)\text{Clarity}(T_2)} \leq \|R_1 \circ R_2\|_1 \leq \|R_1\|_1 \|R_2\|_1.$$

Proof The first statement is obvious. For the first inequality simply use lemma 20 followed by lemma 13. The second inequality is an easy to prove property of operator norms. ■

7.1 Comparing Theorems 6 and 15

We have shown the following implication,

$$\frac{C_1}{\sqrt{n}} \leq \underline{\text{Risk}}_L(e_n) \leq \frac{C_2 \|\ell\|_\infty}{\sqrt{n}} \Rightarrow \frac{C_1}{\sqrt{\text{Clarity}(T) n}} \leq \underline{\text{Risk}}_L(\tilde{e}_n) \leq \frac{C_2 \|\ell_R\|_\infty}{\sqrt{n}},$$

for all reconstructible T. By lemma 20, in the worst case $\|\ell_R\|_\infty \geq \frac{\|\ell\|_\infty}{\text{Clarity}(T)}$. Thus in the worst case over all losses, we arrive at upper and lower bounds for the corrupted problem that are at least

factor of $\sqrt{\text{Clarity}(T)}$ apart. We do not know if this is the fault of our upper or lower bounding techniques. However, for *specific* ℓ and T this gap can be smaller.

For example, in the problem of learning with symmetric label noise discussed in section 4.1.2, with misclassification loss ℓ_{01},

$$\text{Clarity}(T) = 1 - 2\sigma \text{ and } \|\ell_{01,T}\| = \frac{1 - \sigma}{1 - 2\sigma},$$

respectively. The worst case ratio of upper and lower bounds over all losses is of order $\frac{1}{\sqrt{1-2\sigma}}$. For misclassification loss the actual ratio is $\frac{1-\sigma}{\sqrt{1-2\sigma}}$. For all $\sigma \in [0, \frac{2}{10}]$, i.e. up to 0.2 flip probability, this ratio is never larger than $\frac{4}{\sqrt{15}} \approx 1.03$.

7.2 Comparing Theorems 7 and 16

Let $\text{Cost}(T)$ be the cost of acquiring data corrupted by T. Theorem 7 prefers corruptions with low $\|\ell_R\|_\infty^2 \text{Cost}(T)$, or equivalently those with high,

$$\frac{1}{\|\ell_R\|_\infty \text{Cost}(T)}$$

Theorem 16 prefers corruptions with high,

$$\frac{\text{Clarity}(T)}{\text{Cost}(T)} \approx \frac{\|\ell\|_\infty}{\|\ell_R\|_\infty \text{Cost}(T)}.$$

In theorems 16 and 7 we have, respectively, best case and a worst case loss specific method for choosing data sets. Theorem 7 combined with 1emma 19 provides a worst case loss insensitive method for choosing data sets.

8. Corrupted Learning when Clean Learning is Fast

The contents of this paper largely solve the problem of learning from data with corrupted labels, when learning on the original problem occurs at the *standard* $\frac{1}{\sqrt{n}}$ rate. There are many conditions under which clean learning is fast, here we focus on the Bernstein condition presented in Bartlett and Mendelson (2006); van Erven et al. (2012).

Definition 22 *Let $P \in \mathbb{P}(\mathcal{Z})$, ℓ a loss and $a_P = \arg\min_a \mathbb{E}_{z \sim P} \ell(z, a)$. A pair (ℓ, P) satisfies the* Bernstein condition *with constant K if for all $a \in A$,*

$$\mathbb{E}_{z \sim P}(\ell(z, a) - \ell(z, a_P))^2 \leq K \, \mathbb{E}_{z \sim P} \ell(z, a) - \ell(z, a_P)$$

When A is finite, such a condition leads to $\frac{1}{n}$ rates of convergence for empirical risk minimization (ERM).

Theorem 23 *Let $\mathcal{A}$ be ERM with A finite. If (ℓ, P) satisfies the Bernstein condition then for some constant $C > 0$,*

$$\mathbb{E}_{S \sim P^n} \ell(P, \mathcal{A}(S)) - \ell(P, a_P) \leq \frac{C \log(|A|)}{n}.$$

Furthermore with probability at least $1-\delta$ *on a draw from* P^n *one has,*

$$\ell(P, \mathcal{A}(S)) - \ell(P, a_P) \leq \frac{C\left(\log(|A|) + \log\left(\frac{1}{\delta}\right)\right)}{n}.$$

While our lower bounding techniques will turn a lower bound of $\frac{1}{n}$ for clean learning to one of $\frac{1}{\alpha(T)n}$ for corrupted learning, it may be the case that this bound is too optimistic, there may be no algorithm that gives a $\frac{1}{n}$ rate of convergence.

A natural question to ask is when using the ERM algorithm for the loss ℓ_R converges quickly from samples drawn from $\tilde{P}$. Here we ask the simpler question: when does $(\ell_R, \tilde{P})$ satisfy the Bernstein condition?

Lemma 24 *If* $(\ell_R, \tilde{P})$ *satisfies the Bernstein condition with constant* K *then* (ℓ, P) *also satisfies the Bernstein condition with the same constant.*

This theorem (almost) rules out pathological behavior where ERM learns quickly from corrupted data and yet slowly for clean data. The converse of lemma 24 is not true, for example consider the case of PAC learning versus PAC learning with arbitrary instance dependent noise. In some cases the Bernstein condition can be transfered from the clean problem to the corrupted problem, as we now explore.

Definition 25 *Let* $T \in \mathbb{T}(\mathcal{Z}, \tilde{\mathcal{Z}})$*be a corruption and* ℓ *a loss. A pair* (ℓ, T) *are* η-compatible *if for all* $z \in \mathcal{Z}$ *and* $a_1, a_2 \in A$*,*

$$\mathbb{E}_{\tilde{z}\sim T(z)}(\ell_R(\tilde{z}, a_1) - \ell_R(\tilde{z}, a_2))^2 \leq \eta(\ell(z, a_1) - \ell(z, a_2))^2.$$

Theorem 26 *If the pair* (ℓ, P) *satisfies the Bernstein condition with constant* K *and the pair* (ℓ, T) *are* η*-compatible then* $(\ell_R, \tilde{P})$ *satisfies the Bernstein condition with constant* ηK*.*

While by no means the final word on fast corrupted learning, this theorem does allow one to prove interesting results in the binary classification setting.

Theorem 27 *Let* T *be label noise,* $T = \begin{pmatrix} 1-\sigma_{-1} & \sigma_1 \\ \sigma_{-1} & 1-\sigma_1 \end{pmatrix}$*, then the pair* (ℓ_{01}, T) *is* η*-compatible with* $\eta = \left(\frac{1+|\sigma_{-1}-\sigma_1|}{1-\sigma_{-1}-\sigma_1}\right)^2$*.*

One very useful example of a pair (P, ℓ) satisfying the Bernstein condition with constant 1 is when P is separable, ℓ is 01 loss and the Bayes optimal classifier is in the function class. Theorem 27 guarantees that as long as $\sigma_{-1} + \sigma_1 \neq 1$ (i.e. it is possible to learn from noisy labels), one learns at a fast rate from noisy examples.

9. Canonical Losses and Convexity

Thus far we have focused on the statistical properties of corrected loss minimization procedures. However, performing our correction may incur a computational penalty. Even if the loss ℓ is convex, there is no guarantee that the corrected loss remains so. Here we show that a large and useful class of loss functions remain convex when corrected.

Recall the constant function, $\mathbf{1}_{\mathcal{Z}}(z)$ and define,

$$\mathbf{1}_{\mathcal{Z}}^{\perp} := \left\{ v \in \mathbb{R}^{\mathcal{Z}} : \sum_{z\in\mathcal{Z}} v(z) = 0 \right\},$$

those functions that are orthogonal to $\mathbf{1}_{\mathcal{Z}}$. Proper losses and the closely related canonical losses form a large class of loss functions that provide means of attacking prediction problems.

Definition 28 *A loss* $\mathcal{L} : \mathcal{Z} \times A \to \mathbb{R}$ *is* canonical *if* A *is a convex subset of* $\mathbf{1}_{\mathcal{Z}}^{\perp}$ *and,*

$$\mathcal{L}(z, v) = -\langle \delta_z, v\rangle + \Psi(v),$$

for some convex function $\Psi : A \to \mathbb{R}$*.*

Intuitively, minimizing a canonical loss reduces to "lining up" with the average observed label, with "over confident" predictions penalized by the function Ψ. In appendix C we show that all losses are essentially canonical losses in disguise (theorem 49). Note that canonical losses split into two terms, one convex in v and unaffected by the observation, and a term linear in v. These losses are particular easy to correct for corruption, as only the linear term needs to be corrected.

Theorem 29 (Correcting Canonical Losses) *Let* $\mathcal{L} : \mathcal{Z} \times A \to \mathbb{R}$ *be a canonical loss. For all reconstructible corruptions* $T \in \mathbb{T}(\mathcal{Z}, \tilde{\mathcal{Z}})$ *there exists a reconstruction* R *such that,*

$$\mathcal{L}_R(\tilde{z}, v) = -\langle \delta_{\tilde{z}}, R^*(v)\rangle + \Psi(v)$$

Furthermore R *can be calculated efficiently via Gaussian elimination.*

We develop the necessary representation results needed to prove this theorem in appendix C.

Theorem 29 extends results presented in Natarajan et al. (2013) concerning when losses used for learning binary classifiers remain convex when corrected for asymmetric label noise. Therefore we need not abandon the framework of convex surrogates when the corruption is known.

9.1 Comparison with Natarajan et al. (2013)

Recall in the problem of learning with label noise,

$$T = \begin{pmatrix} 1-\sigma_{-1} & \sigma_1 \\ \sigma_{-1} & 1-\sigma_1 \end{pmatrix},$$

where σ_y is the probability of label y being flipped. For this problem, the corrected loss is,

$$\ell_R(y,a) = \frac{(1-\sigma_{-y})\ell(y,a) - \sigma_y \ell(-y,a)}{1-\sigma_{-1}-\sigma_1}.$$

If $\ell : \{\pm 1\} \times \mathbb{R} \to \mathbb{R}$ is convex, smooth and satisfies,

$$\ell''(y,a) = \ell''(-y,a),$$

then lemma 4 of Natarajan et al. (2013) guarantees that ℓ_R is convex. Integrating twice yields,

$$\ell(y,a) = b_y + c_y v + g(a),$$

For some convex function g. Rearranging one has,

$$\ell(y,a) = \left(\frac{b_{-1} - b_1 + (c_1 - c_{-1})\,a}{2}\right) y + \underbrace{\frac{b_1 + b_{-1} + (c_1 + c_{-1})\,a}{2} + g(a)}_{\text{convex in } a}.$$

For canonical losses, corollary 50 states that any binary canonical loss is of the form $\mathcal{L} : \{\pm 1\} \times I \to \mathbb{R}$,

$$\mathcal{L}(y,v) = -yv + \Psi(v),\ v \in I,$$

where I is a convex subset of $\mathbb{R}$ and $\Psi : I \to \mathbb{R}$ a convex function. Letting,

$$\nu = -\left(\frac{b_{-1} - b_1 + (c_1 - c_{-1})\,a}{2}\right),$$

one can see that the losses considered in Natarajan et al. (2013) are affine re-parametrizations of canonical losses. Theorem 29 is therefore a generalization of lemma 4 of Natarajan et al. (2013).

9.2 Comparison with Cid-Sueiro et al. (2014)

Section 5 of Cid-Sueiro et al. (2014) provides some discussion of the convexity of corruption corrected losses. They state, without proof, that a requirement for the preservation of convexity is that the reconstruction have *non negative* entries. Theorem 29 shows that this is not the case, one can *always* preserve convexity as long as the correct reconstruction and parametrization of the loss function are used.

10. Learning when the Corruption Process is Partially Known

Thus far we have considered the problem of learning when T is known. Here we consider the problem of when T lies in a subset $\mathcal{C} \subset \mathbb{T}_{\leftarrow}(\mathcal{Z}, \tilde{\mathcal{Z}})$. For example when learning classifiers under symmetric label noise (Angluin and Laird, 1988), the corruption is of the form,

$$T_\sigma = \begin{pmatrix} 1-\sigma & \sigma \\ \sigma & 1-\sigma \end{pmatrix},$$

where $\sigma \neq \frac{1}{2}$. There are three ways in which one can proceed.

If we assume access to a "gold standard" sample $S \sim P^n$ as well as a corrupted sample $\tilde{S}$, we can use methods akin to those in Kearns (1998). One covers the set $\mathcal{C}$ to some tolerance ϵ with a finite cover $\{T_i\}_{i=1}^k$. For each T_i in the cover, estimate an action a_i using ℓ_{R_i} and the corrupted sample. Finally, choose the a_i that best predicts the gold standard sample. Using theorem 6, we know that for a large enough corrupted sample, one of the a_i has performance close to that of the optimal a.

One can attempt to *estimate* T from the corrupted sample. Under certain distributional assumptions (such as separability and mutual irreducibility), Menon et al. (2015) surveys methods for estimating T for the problem of learning under asymmetric label noise. Blanchard et al. (2016) gives theory for these estimators. There is a growing literature extending these estimates to other problems as well as new, computational efficient estimates (Blanchard and Scott, 2014; Katz-Samuels and Scott, 2016; Ramaswamy et al., 2016). We believe these methods can be extended to general corruptions.

In both of the above methods, operator norms can provide suitable losses/metrics that can guide their use.

Lemma 30 *Let $T, T' \in \mathbb{T}_{\leftarrow}(\mathcal{Z}, \tilde{\mathcal{Z}})$. Then,*

$$\|\ell_R - \ell_{R'}\|_\infty \leq \left\|R - R'\right\|_1 \|\ell\|_\infty.$$

The quantity $\|R - R'\|_1$ is a statistically motivated distance that can be used when covering $\mathcal{C}$. Furthermore, it can be used when designing loss functions for estimating T.

Finally, one can look for loss functions that are "invariant" to $\mathcal{C}$. This approach is explored further in section 11 and in van Rooyen et al. (2017).

11. Corruption Invariant Loss Functions

While exact knowledge of $T \in \mathcal{C} \subset \mathbb{T}_{\leftarrow}(\mathcal{Z}, \tilde{\mathcal{Z}})$ is required to estimate the *expected loss* from a corrupted distribution, in certain situations this is unnecessary for estimating optimal actions and *any* reconstruction will suffice. In this section we formalize this notion and *characterize* when exact knowledge of T is unnecessary. We assume for each $T \in \mathcal{C}$ an explicit reconstruction R, ie a function,

$$\mathrm{Rec} : \mathcal{C} \to \mathbb{T}(\tilde{\mathcal{Z}}, \mathcal{Z}),$$

with $\mathrm{Rec}(T)T = \mathrm{id}_{\mathcal{Z}}$. As in theorem 29, we will assume that,

$$\mathrm{Rec}(T)^*(\mathbf{1}_{\mathcal{Z}}) = \mathbf{1}_{\tilde{\mathcal{Z}}},\ \forall T \in \mathcal{C}.$$

We focus on the problem of prediction. Let $\ell : \mathcal{Z} \times A \to \mathbb{R}$ be a loss. ℓ provides an *ordering* on $\mathbb{P}(\mathcal{Z} \times A)$. $P_1 \leq_\ell P_2$ if,

$$\mathbb{E}_{(z,a)\sim P_1} \ell(z,a) \leq \mathbb{E}_{(z,a)\sim P_2} \ell(z,a).$$

In words, P_1 is making better decisions than P_2. One can think of P_1 and P_2 being the output of a learning algorithm, although exactly how they came to be is of no concern in this section.

Let $T \in \mathcal{C}$ be the true corruption and $T_0 \in \mathcal{C}$ the assumed corruption. By properties of adjoints,

$$\langle T(P), \mathrm{Rec}(T_0)^* \ell(-,a) \rangle = \langle P, (\mathrm{Rec}(T_0)T)^* \ell(-,a) \rangle.$$

When using the wrong reconstruction the decision maker is effectively using the loss,

$$\tilde{\ell}_T(-,a) := (\mathrm{Rec}(T_0)T)^* \ell(-,a),$$

in place of ℓ. In general there is no guarantee,

$$P_1 \leq_\ell P_2 \Leftrightarrow P_1 \leq_{\tilde{\ell}_T} P_2.$$

In words, assuming the *wrong* corruption may lead to the *wrong* ordering. Corruption immune losses are precisely those where the ordering is *unaltered.*

The reader may wonder why we focus on preserving *order* rather than preserving only the optimal action. Invariably in machine learning we make approximations, be it via working with a finite simple, or a restricted function class. What is key therefore is preserving what is *better* and not only what is *best.*

Definition 31 (Order Equivalence) *Let $\ell, \ell' : \mathcal{Z} \times A \to \mathbb{R}$ be loss functions. ℓ is* order equivalent *to ℓ' if for all $P_1, P_2 \in \mathbb{P}(A \times \mathcal{Z})$,*

$$P_1 \leq_\ell P_2 \Leftrightarrow P_1 \leq_{\ell'} P_2.$$

The lemma below *characterizes* when losses are order equivalent.

Lemma 32 (Theorem 2, Section 7.9 of DeGroot (1962)) *ℓ is order equivalent to ℓ' if and only if there exists a constants $\alpha > 0$ and β such that,*

$$\ell(z,a) = \alpha \ell'(z,a) + \beta,\ \forall z \in \mathcal{Z},\ \forall a \in A.$$

Definition 33 *Let $\mathcal{C} \subset \mathbb{T}(\mathcal{Z}, \tilde{\mathcal{Z}})$ be a set of reconstructible transitions. A loss ℓ is* immune to $\mathcal{C}$ *if for all $T \in \mathcal{C}$, ℓ is order equivalent to $\tilde{\ell}_T$.*

Losses that are immune to $\mathcal{C}$ are precisely those where exact knowledge of the corruption is unnecessary, assuming *any* $T \in \mathcal{C}$ and using its corresponding reconstruction is sufficient.

Theorem 34 *Fix $T_0 \in \mathcal{C}$. A loss ℓ is* immune to $\mathcal{C} \subset \mathbb{T}_\leftarrow(\mathcal{Z}, \tilde{\mathcal{Z}})$ *if and only if for all $T \in \mathcal{C}$.*

$$(\mathrm{Rec}(T_0)T)^* \ell(-,a) = \alpha(T)\ell(-,a) + \beta(T)\mathbf{1}_{\mathcal{Z}},\ \forall a \in A,$$

for functions $\alpha : \mathcal{C} \to \mathbb{R}_+$ and $\beta : \mathcal{C} \to \mathbb{R}$.

The proof follows directly from the definition and lemma 32. The operator $(\mathrm{Rec}(T_0)T)$ measures the effect of reconstructing incorrectly.

11.1 Constructing Corruption Immune Loss Functions

Theorem 34 provides means to *test* when a loss function is immune to $\mathcal{C}$. Here we show how to *construct* such losses. Immune losses arise from the *persistent eigenvectors* of $(\mathrm{Rec}(T_0)T)^*$.

Let $S \subset \mathbb{T}(\mathcal{Z}, \mathcal{Z})$. We call v a *persistent eigenvector* of S if,

$$Mv = \lambda(M)v,\ \forall M \in S,$$

ie, v is an eigenvector *for all* $M \in S$, albeit with differing eigenvalue. We call the function $\lambda : S \to \mathbb{R}$ a *persistent eigenvalue*. Much like normal eigenspaces, we define *persistent eigenspaces* as subspaces of $\mathbb{R}^{\mathcal{Z}}$ with the same persistent eigenvalue,

$$E_{\lambda,S} = \left\{ v \in \mathbb{R}^{\mathcal{Z}} : Mv = \lambda(M)v,\ \forall T \in S \right\}.$$

Persistent eigenspaces provide an alternative statement of theorem 34.

Corollary 35 *Let $\mathcal{C} \subset \mathbb{T}_\leftarrow(\mathcal{Z}, \tilde{\mathcal{Z}})$, with $T_0 \in \mathcal{C}$. Let,*

$$S = \left\{ (\mathrm{Rec}(T_0)T)^* : T \in \mathcal{C} \right\}.$$

and let $\alpha : S \to \mathbb{R}_+$ be a persistent eigenvalue of S. Any loss ℓ of the form,

$$\ell(-,a) = v(a) + \gamma \mathbf{1}_{\mathcal{Z}},$$

where $v(a) \in E_{\alpha,S}\ \forall a \in A$ and $\gamma \in \mathbb{R}$ is immune to $\mathcal{C}$.

Therefore the search for losses that are immune to $\mathcal{C}$ reduces to the calculation of the persistent eigenspaces of $\{(\mathrm{Rec}(T_0)T)^* : T \in \mathcal{C}\}$.

11.2 Examples of Corruption Immune Losses

Here we show how to apply corollary 35 to find losses that are immune to families of corruptions.

11.2.1 SYMMETRIC LABEL NOISE AND THE LINEAR LOSS

We proceed as in van Rooyen et al. (2017). In the problem of symmetric label noise $\mathcal{Z} = \{-1, +1\}$ with,

$$T = \begin{pmatrix} 1-\sigma & \sigma \\ \sigma & 1-\sigma \end{pmatrix} \text{ and } R = \frac{1}{1-2\sigma} \begin{pmatrix} 1-\sigma & -\sigma \\ -\sigma & 1-\sigma \end{pmatrix},$$

for $\sigma \neq \frac{1}{2}$. Let $\mathcal{C}$ be all such T with $\sigma \neq \frac{1}{2}$. Define the *linear* loss function,

$$\ell(y,v) = -yv,\ v \in \mathbb{R},$$

or in partial form,

$$\ell(-,v) = \begin{pmatrix} v \\ -v \end{pmatrix}.$$

We can quickly verify that the linear loss is immune to $\mathcal{C}$. Let $T_0 = \mathrm{id}_{\mathcal{Z}}$. As $\begin{pmatrix} 1 \\ -1 \end{pmatrix}$ is an eigenvector of T^* with eigenvalue of $1 - 2\sigma$ we have,

$$T^*\ell(-, v) = (1 - 2\sigma)\,\ell(-, v),\ \forall v \in \mathbb{R},\ \forall \sigma \neq \frac{1}{2}.$$

Therefore the linear loss is immune to $\mathcal{C}$.

11.2.2 MULTI-CLASS IMMUNE LOSSES

Here we consider the three-class generalization of linear loss with $\mathcal{Z} = \{1, 2, 3\}$ and,

$$\ell(-, v) = v, v \in \mathbf{1}_{\mathcal{Z}}^{\perp},$$

with,

$$v = v_1 \begin{pmatrix} -\frac{2}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{pmatrix} + v_2 \begin{pmatrix} \frac{1}{3} \\ -\frac{2}{3} \\ \frac{1}{3} \end{pmatrix},\ v_1, v_2 \in \mathbb{R}.$$

We consider two classes of corruption, firstly three-class symmetric noise, secondly symmetric partial label noise.

For three-class symmetric noise,

$$T = \begin{pmatrix} 1-\sigma & \frac{\sigma}{2} & \frac{\sigma}{2} \\ \frac{\sigma}{2} & 1-\sigma & \frac{\sigma}{2} \\ \frac{\sigma}{2} & \frac{\sigma}{2} & 1-\sigma \end{pmatrix} \text{ and } R = \begin{pmatrix} \frac{2-\sigma}{2-3\sigma} & \frac{-\sigma}{2-3\sigma} & \frac{-\sigma}{2-3\sigma} \\ \frac{-\sigma}{2-3\sigma} & \frac{2-\sigma}{2-3\sigma} & \frac{-\sigma}{2-3\sigma} \\ \frac{-\sigma}{2-3\sigma} & \frac{-\sigma}{2-3\sigma} & \frac{2-\sigma}{2-3\sigma} \end{pmatrix},$$

for $\sigma \neq \frac{2}{3}$. Fix $T_0 = \mathrm{id}_{\mathcal{Z}}$. Both,

$$v = v_1 \begin{pmatrix} -\frac{2}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{pmatrix} \text{ and } \begin{pmatrix} \frac{1}{3} \\ -\frac{2}{3} \\ \frac{1}{3} \end{pmatrix},$$

are eigenvalues of T^* with eigenvalue $\frac{2-3\sigma}{2}$. Therefore,

$$T^*\ell(-, v) = \frac{2 - 3\sigma}{2}\ell(-, v),\ \forall v \in \mathbf{1}_{\mathcal{Z}}^{\perp},\ \forall \sigma \neq \frac{2}{3},$$

Hence linear loss is immune to three-class symmetric noise. Note that general losses are *not* immune to symmetric three-class label noise.

In learning under symmetric partial label noise,

$$T = \begin{pmatrix} (1-\sigma)^2 & 0 & 0 \\ 0 & (1-\sigma)^2 & 0 \\ 0 & 0 & (1-\sigma)^2 \\ (1-\sigma)\sigma & (1-\sigma)\sigma & 0 \\ (1-\sigma)\sigma & 0 & (1-\sigma)\sigma \\ 0 & (1-\sigma)\sigma & (1-\sigma)\sigma \\ \sigma^2 & \sigma^2 & \sigma^2 \end{pmatrix} \text{ and } R = \begin{pmatrix} 1 & 0 & 0 & \frac{3-2\sigma}{3-3\sigma} & \frac{3-2\sigma}{3-3\sigma} & -\frac{\sigma-3}{3(\sigma-1)} & \frac{1}{3} \\ 0 & 1 & 0 & \frac{3-2\sigma}{3-3\sigma} & -\frac{\sigma-3}{3(\sigma-1)} & \frac{3-2\sigma}{3-3\sigma} & \frac{1}{3} \\ 0 & 0 & 1 & -\frac{\sigma-3}{3(\sigma-1)} & \frac{3-2\sigma}{3-3\sigma} & \frac{3-2\sigma}{3-3\sigma} & \frac{1}{3} \end{pmatrix},$$

for $\sigma \neq 1$. Taking,

$$T_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

ie $\sigma = 0$, one has,

$$(RT_1)^* = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \forall \sigma \neq 1.$$

This means that *all* losses are immune to symmetric partial label noise.

11.3 Comparison with Cid-Sueiro et al. (2014)

The development here closely mirrors that of Cid-Sueiro et al. (2014) with one key exception. While we look for *losses* with,

$$(\mathrm{Rec}(T_0)T)^*\ell(-, a) \cong \ell(-, a),\ \forall T \in \mathcal{C},$$

Cid-Sueiro et al. (2014) looks for classes of losses $\ell : \mathcal{Z} \times \mathbb{R}^{\mathcal{Z}} \to \mathbb{R}$ that are closed under the action of $(\mathrm{Rec}(T_0)T)$. In particular they consider strictly proper and classification calibrated (Bartlett et al., 2006) loss functions. They show (theorem 2) that if,

$$(\mathrm{Rec}(T_0)T)^* = \alpha(T)\mathrm{id}_{\mathcal{Z}},\ \forall T \in \mathcal{C},$$

for $\alpha(T) > 0$, then proper losses are mapped to proper losses by $(\mathrm{Rec}(T_0)T)^*$. By theorem 34 this would guarantee,

$$(\mathrm{Rec}(T_0)T)^*\ell(-, a) = \alpha(T)\ell(-, a),\ \forall T \in \mathcal{C},\ \forall a \in A,\ \forall \ell,$$

ie *all* losses are immune to $\mathcal{C}$. We saw an example of this in section 11.2.2. They also show (theorem 5) that if,

$$(\mathrm{Rec}(T_0)T)^* = \alpha(T)\mathrm{id}_{\mathcal{Z}} + v(T) \otimes \mu,\ \forall T \in \mathcal{C},$$

for $\alpha(T) > 0$ and $v(T) \in \mathbb{R}^{\mathcal{Z}}$, where μ is the uniform distribution over the outcomes, then classification calibrated losses are mapped to classification calibrated losses. This yields,

$$(\text{Rec}(T_0)T)^*\ell(-,a) = \alpha(T)\ell(-,a) + \left(\sum_{z\in\mathcal{Z}} \ell(z,a)\right) v(T),\ \forall T \in \mathcal{C},\ \forall a \in A,\ \forall \ell,$$

or in expectation,

$$\langle P, (\text{Rec}(T_0)T)^*\ell(-,a)\rangle = \alpha(T)\ell(P,a) + \langle P, v(T)\rangle\, \ell(\mu,a),$$

ie the effect of reconstructing incorrectly is to add a distribution dependent multiple of $\ell(\mu,a)$ to the loss.

The virtue of our approach is the identification of loss functions with *stronger* robustness properties over standard classification calibrated losses. We direct the reader to van Rooyen et al. (2017) for an in depth discussion.

12. Summary and a Guide to the Practitioner

Real world data sets are amalgamations of data of variable quality and type. Understanding how to *learn from* and *compare* different corrupted data sets is therefore a problem of great practical importance. Theorem 7, Appendix C and theorem 29 yield powerful means to do this. In concert, they provide the following framework to the practitioner:

1. Identify the appropriate loss function for the problem at hand.
2. Acquire data in accordance with theorem 7, and any relevant financial constraints.
3. Correct for any noise present in the data, and solve for,

$$\arg\min_{f\in\mathcal{F}} \sum_{i=1}^{n} \ell_{R_i}(y_i, f(x_i)).$$

As long as the decision maker is willing to work with *linear* and *kernelized* function classes, this is a convex problem with *many* available algorithmic solutions.

Acknowledgments

This work was completed while Brendan van Rooyen was with NICTA, The Australian National University, ACEMS and The Queensland University of Technology. The work was partially supported by the Australian Research Council. Many thanks go to Aditya Menon for many interesting conversations and providing some of the initial impetus for the paper. Thanks to Clayton Scott for providing detailed feedback.

Appendix A. Examples

Here we show examples of common corrupted machine learning problems.

A.1 Noisy Labels

We consider the problem of learning from noisy binary labels (Angluin and Laird, 1988; Natarajan et al., 2013). Here σ_i is the probability that class i is has its label flipped. We have,

$$T = \begin{pmatrix} 1-\sigma_{-1} & \sigma_1 \\ \sigma_{-1} & 1-\sigma_1 \end{pmatrix} \text{ and } R = \frac{1}{1-\sigma_{-1}-\sigma_1}\begin{pmatrix} 1-\sigma_1 & -\sigma_1 \\ -\sigma_{-1} & 1-\sigma_{-1} \end{pmatrix},$$

This yields,

$$\ell_R(y,a) = \frac{(1-\sigma_{-y})\ell(y,a) - \sigma_y\ell(-y,a)}{1-\sigma_{-1}-\sigma_1}.$$

The above loss is lemma 1 in Natarajan et al. (2013). Interestingly, even if ℓ is positive, ℓ_R can be negative. If the noise is symmetric with $\sigma_{-1} = \sigma_1 = \sigma$ and ℓ is 01 loss then,

$$\ell_R(y,a) = \frac{\ell_{01}(y,a) - \sigma}{1-2\sigma},$$

which is just a rescaled and shifted version of 01 loss. If we work in the realizable setting, i.e. there is some $f \in \mathcal{F}$ with,

$$\mathbb{E}_{(x,y)\sim P}\ell_{01}(y, f(x)) = 0,$$

then the above provides an interesting correspondence between learning with symmetric label noise and learning under distributions with large Tsybakov margin (Audibert and Tsybakov, 2007). Taking $\sigma = \frac{1}{2} - h$ with P *separable* in turn implies $\tilde{P}$ has Tsybakov margin h. This means bounds developed for this setting, such as in Massart and Nédélec (2006), can be transferred to the setting of learning with symmetric label noise. Our lower bound reproduces the results of Massart and Nédélec (2006).

Below is a table of the relevant parameters for learning with noisy binary labels. These results directly extend those presented in Kearns (1998) that considered only the case of symmetric label noise.

Learning with Label Noisy (Binary)

T	$\begin{pmatrix} 1-\sigma_{-1} & \sigma_1 \\ \sigma_{-1} & 1-\sigma_1 \end{pmatrix}$
R^*	$\frac{1}{1-\sigma_{-1}-\sigma_1}\begin{pmatrix} 1-\sigma_1 & -\sigma_1 \\ -\sigma_{-1} & 1-\sigma_{-1} \end{pmatrix}$
$\text{Clarity}(T)$	$\lvert 1-\sigma_{-1}-\sigma_1\rvert$
$\lVert R^*\rVert_\infty$	$\frac{1}{\lvert 1-\sigma_{-1}-\sigma_1\rvert}\max(1-\sigma_{-1}+\sigma_1, 1-\sigma_1+\sigma_{-1})$
$\lVert \ell_{01,T}\rVert_\infty$	$\frac{1}{\lvert 1-\sigma_{-1}-\sigma_1\rvert}\max(1-\sigma_{-1}, 1-\sigma_1, \sigma_{-1}, \sigma_1)$

We see that as long as $\sigma_{-1} + \sigma_1 \neq 1$, T is reconstructible. The pattern we see in this table is quite common. $\|R^*\|_\infty$ tends to be marginally greater than $\frac{1}{\alpha(T)}$, with $\|\ell_{01,T}\|_\infty$ less than both. In the symmetric case our lower bound reproduces that of Aslam and Decatur (1996).

Finally, when working with symmetric label noise ($\sigma_{-1} = \sigma_1 = \sigma$),

$$\|R_\sigma - R_{\sigma'}\|_1 = \frac{2|\sigma - \sigma'|}{|1 - 2\sigma||1 - 2\sigma'|}.$$

For fixed true noise rate σ, the presence of a factor $|1-2\sigma'|$ in the denominator means that underestimating σ is preferred to overestimating. Hence when designing estimates for σ, those with negative bias might perform better than those that are unbiased or are positively biased. Furthermore, when covering noise rates, as per the discussion in 10, more focus should be given to higher noise rates than to lower.

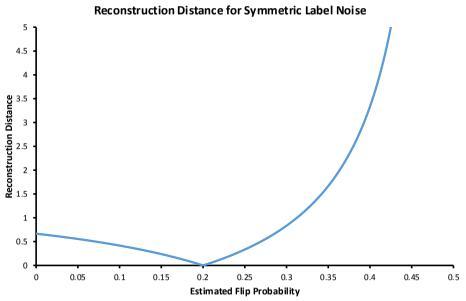


Figure 1: Plot of $\|R_\sigma - R_{\sigma'}\|_1$ for $\sigma = 0.2$. $\|R_\sigma - R_{\sigma'}\|_1$ is a measure of how far apart two corruptions are. This distance measure can be used when constructing estimators for the corruption process T. See text.

A.2 Semi-Supervised Learning

We consider the problem of semi-supervised learning (Chapelle et al., 2010). Here $1 - \sigma_i$ is the probability class i has a missing label. We consider the symmetric case where $\sigma_{-1} = \sigma_1 = \sigma$.

Symmetric Semi-Supervised Learning

T	$\begin{pmatrix} \sigma & 0 \\ 0 & \sigma \\ 1-\sigma & 1-\sigma \end{pmatrix}$
R^*	$\begin{pmatrix} \frac{1-2\sigma+2\sigma^2}{1-3\sigma+5\sigma^2-3\sigma^3} & \frac{-\sigma^2}{1-3\sigma+5\sigma^2-3\sigma^3} \\ \frac{-\sigma^2}{1-3\sigma+5\sigma^2-3\sigma^3} & \frac{1-2\sigma+2\sigma^2}{1-3\sigma+5\sigma^2-3\sigma^3} \\ \frac{\sigma}{1-2\sigma+3\sigma^2} & \frac{\sigma}{1-2\sigma+3\sigma^2} \end{pmatrix}$
$\text{Clarity}(T)$	σ
$\|R^*\|_\infty$	$\frac{1}{\sigma}$
$\|\ell_{01,T}\|_\infty$	$\frac{1-2\sigma+2\sigma^2}{2\sigma+3\sigma-5\sigma^2}$

Once again $\|\ell_{01,T}\|_\infty \leq \frac{1}{\text{Clarity}(T)}$. Our lower bound confirms that in general unlabelled data does not help (Balcan and Blum, 2010). Rather than using the method of unbiased estimators, one could simply throw away the unlabelled data leaving behind σn labelled instances on average. To make further progress in this problem, as noted elsewhere (Balcan and Blum, 2010), normally one assumes some form of compatibility between the marginal distribution of instances and the optimal classifier. In principle, restricted versions of Le Cams method and the strong data processing inequality could be used to give lower bounds under these different assumptions. As our interest here are minimax bounds, we do not pursue these methods.

A.3 Three Class Symmetric Label Noise

Here we present parameters for the three class variant of symmetric label noise. We have $\tilde{Y} = Y = \{1, 2, 3\}$ with $P(\tilde{Y} = \tilde{y}|Y = y) = 1 - \sigma$, if $y = \tilde{y}$ and $\frac{\sigma}{2}$ otherwise.

Learning with Symmetric Label Noisy (Multiclass)

T	$\begin{pmatrix} 1-\sigma & \frac{\sigma}{2} & \frac{\sigma}{2} \\ \frac{\sigma}{2} & 1-\sigma & \frac{\sigma}{2} \\ \frac{\sigma}{2} & \frac{\sigma}{2} & 1-\sigma \end{pmatrix}$
R^*	$\begin{pmatrix} \frac{2-\sigma}{2-3\sigma} & \frac{-\sigma}{2-3\sigma} & \frac{-\sigma}{2-3\sigma} \\ \frac{-\sigma}{2-3\sigma} & \frac{2-\sigma}{2-3\sigma} & \frac{-\sigma}{2-3\sigma} \\ \frac{-\sigma}{2-3\sigma} & \frac{-\sigma}{2-3\sigma} & \frac{2-\sigma}{2-3\sigma} \end{pmatrix}$
$\text{Clarity}(T)$	$\lvert 1 - \frac{3}{2}\sigma \rvert$
$\|R^*\|_\infty$	$\frac{2+\sigma}{\lvert 2-3\sigma \rvert}$
$\|\ell_{01,T}\|_\infty$	$\frac{2}{\lvert 2-3\sigma \rvert} \max(\sigma, 1-\sigma)$

We see that as long as $\sigma \neq \frac{2}{3}$, T is reconstructible. Once again $\|\ell_{01,T}\|_\infty \leq \frac{1}{\alpha(T)}$.

A.4 Partial Labels

Here we follow Cour et al. (2011) with $Y = \{1, 2, 3\}$ and $\tilde{Y} = \{0, 1\}^Y$ the set of partial labels. A partial label of $(0, 1, 1)$ indicates that the true label is either 2 or 3 but not 1. We assume that a

partial label always includes the true label as one of the possibilities and furthermore that spurious labels are added with probability σ.

Learning with Partial Labels

T	$\begin{pmatrix} (1-\sigma)^2 & 0 & 0 \\ 0 & (1-\sigma)^2 & 0 \\ 0 & 0 & (1-\sigma)^2 \\ (1-\sigma)\sigma & (1-\sigma)\sigma & 0 \\ (1-\sigma)\sigma & 0 & (1-\sigma)\sigma \\ 0 & (1-\sigma)\sigma & (1-\sigma)\sigma \\ \sigma^2 & \sigma^2 & \sigma^2 \end{pmatrix}$
R	$\begin{pmatrix} 1 & 0 & 0 & \frac{3-2\sigma}{3-3\sigma} & \frac{3-2\sigma}{3-3\sigma} & -\frac{\sigma-3}{3(\sigma-1)} & \frac{1}{3} \\ 0 & 1 & 0 & \frac{3-2\sigma}{3-3\sigma} & -\frac{\sigma-3}{3(\sigma-1)} & \frac{3-2\sigma}{3-3\sigma} & \frac{1}{3} \\ 0 & 0 & 1 & -\frac{\sigma-3}{3(\sigma-1)} & \frac{3-2\sigma}{3-3\sigma} & \frac{3-2\sigma}{3-3\sigma} & \frac{1}{3} \end{pmatrix}$
$\mathrm{Clarity}(T)$	$1-\sigma$
$\|R^*\|_\infty$	$\frac{\lvert 3-\sigma\rvert + 2\lvert 3-2\sigma\rvert}{3\lvert 1-\sigma\rvert}$
$\|\ell_{01,T}\|_\infty$	$\frac{6-4\sigma}{3-3\sigma}$

We see that as long as $\sigma \neq 1$, T is reconstructible. In this case $\|\ell_{01,T}\|_\infty$ and $\|R^*\|_\infty$ are given by more complicated expressions (however they are both available in closed form). We display their interrelation in a graph in below. To the best of our knowledge, no upper and lower bounds are present in the literature for this problem.

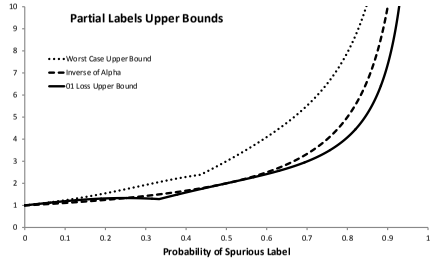


Figure 2: Upper and lower bounds for the problem of learning from partial labels, see text.

Appendix B. Le Cam's Method and Minimax Lower Bounds

The development here closely follows Duchi et al. (2013) with some streamlining. We consider a general decision problem with unknowns Θ, observation space $\mathcal{Z}$ and loss $L : \Theta \times A \to \mathbb{R}$, with $\inf_a L(\theta, a) = 0$. For any learning algorithm $\mathcal{A} \in \mathbb{T}(\mathcal{Z}, A)$ we wish to lower bound the minimax risk,

$$\underline{\mathrm{Risk}}_L(e) = \inf_{\mathcal{A}} \sup_{\theta} \mathbb{E}_{z\sim e(\theta)} L(\theta, \mathcal{A}(z)).$$

The method proceeds by reducing a general decision problem to an easier binary classification problem, by considering a supremum of the risk over a restricted set $\{\theta_1, \theta_2\}$. Using Markov's inequality we then relate this to the minimum 01 loss in a particular binary classification problem. Finally one finds a lower bound for this quantity. With $\theta\sim\{\theta_1, \theta_2\}$ meaning θ is drawn uniformly at random from the set $\{\theta_1, \theta_2\}$, we have,

$$\begin{aligned}
\sup_{\theta} \mathbb{E}_{z\sim e(\theta)} \mathbb{E}_{a\sim\mathcal{A}(z)} L(\theta, a) &\geq \sup_{\{\theta_1,\theta_2\}} \mathbb{E}_{z\sim e(\theta)} \mathbb{E}_{a\sim\mathcal{A}(z)} L(\theta, a) \\
&\geq \mathbb{E}_{\theta\sim\{\theta_1,\theta_2\}} \mathbb{E}_{z\sim e(\theta)} \mathbb{E}_{a\sim\mathcal{A}(z)} L(\theta, a) \\
&\geq \delta \mathbb{E}_{\theta\sim\{\theta_1,\theta_2\}} \mathbb{E}_{z\sim e(\theta)} \mathbb{E}_{a\sim\mathcal{A}(z)} [\![L(\theta, a) \geq \delta]\!].
\end{aligned}$$

Recall the *separation* $\rho : \Theta\times\Theta \to \mathbb{R}$, $\rho(\theta_1, \theta_2) = \inf_a L(\theta_1, a) + L(\theta_2, a)$. The separation measures how hard it is to act well against both θ_1 and θ_2 simultaneously. We now assume $\rho(\theta_1, \theta_2) > 2\delta$. Define $f : A \to \{\theta_1, \theta_2, \text{error}\}$ where $f(a) = \theta_i$ if $L(\theta_i, a) < \delta$ and error otherwise. This function is well defined as if there exists an action a with $L(\theta_1, a) < \delta$ and $L(\theta_2, a) < \delta$ then $\rho(\theta_1, \theta_2) < 2\delta$, a contradiction. Let $\hat{\mathcal{A}}$ be the classifier that first draws $a\sim\mathcal{A}(z)$ and then outputs $f(a)$. We have,

$$\begin{aligned}
\sup_{\theta} \mathbb{E}_{z\sim e(\theta)} \mathbb{E}_{a\sim\mathcal{A}(z)} L(\theta, a) &\geq \delta \mathbb{E}_{\theta\sim\{\theta_1,\theta_2\}} \mathbb{E}_{z\sim e(\theta)} \mathbb{E}_{\theta'\sim\hat{\mathcal{A}}(z)} [\![\theta \neq \theta']\!] \\
&\geq \delta \inf_{\hat{\mathcal{A}}\in\mathbb{T}(\mathcal{Z},\Theta)} \mathbb{E}_{\theta\sim\{\theta_1,\theta_2\}} \mathbb{E}_{z\sim e(\theta)} \mathbb{E}_{\theta'\sim\hat{\mathcal{A}}(z)} [\![\theta \neq \theta']\!] \\
&= \delta \left(\frac{1}{2} - \frac{1}{2} V(e(\theta_1), e(\theta_2)) \right),
\end{aligned}$$

where the first line is a rewriting of of the previous in terms of the classifier $\hat{\mathcal{A}}$, the second takes an infimum over all classifiers and the final line is a standard result in theoretical statistics (see Reid and Williamson (2011) for a modern treatment). Taking $\delta = \frac{\rho(\theta_1,\theta_2)}{2}$ yields lemma 8.

B.1 Extension of Le Cam's Method to Bayesian Risk

Rather than lower bounding $\sup_\theta \mathbb{E}_{z\sim e(\theta)} L(\theta, \mathcal{A}(z))$, a Bayesian with some knowledge about the unknown, given in the form of a prior $\pi \in \mathbb{P}(\Theta)$, wishes to lower bound the Bayesian risk,

$$\underline{\mathrm{Risk}}^{\pi}_L(e) := \inf_{\mathcal{A}} \mathbb{E}_{\theta\sim\pi} \mathbb{E}_{z\sim e(\theta)} L(\theta, \mathcal{A}(z)).$$

Following from the second line of the derivation of Le Cam's method, we have a lower bound,

$$\begin{aligned}
\mathbb{E}_{\theta\sim\{\theta_1,\theta_2\}} \mathbb{E}_{z\sim e(\theta)} \mathbb{E}_{a\sim\mathcal{A}(z)} L(\theta, a) &= \frac{1}{2}\mathrm{Risk}_L(\theta_1, e, \mathcal{A}) + \frac{1}{2}\mathrm{Risk}_L(\theta_2, e, \mathcal{A}) \\
&\geq \rho(\theta_1, \theta_2) \left(\frac{1}{4} - \frac{1}{4} V(e(\theta_1), e(\theta_2)) \right).
\end{aligned}$$

Let $\mu \in \mathbb{P}(\Theta \times \Theta)$ be any distribution with *both* marginals over Θ equal to π. Averaging over this distribution we have,

$$\mathbb{E}_{\theta\sim\pi}\text{Risk}_L(\theta, e, \mathcal{A}) \geq \mathbb{E}_{(\theta_1,\theta_2)\sim\mu}\rho(\theta_1, \theta_2)\left(\frac{1}{4} - \frac{1}{4}V(e(\theta_1), e(\theta_2))\right).$$

This insight leads to a Bayesian version of lemma 8.

Lemma 36 *Let $\mu \in \mathbb{P}(\Theta \times \Theta)$ be any distribution with* both *marginals over Θ equal to π. Then for all experiments e and loss functions ℓ,*

$$\underline{\text{Risk}}^{\pi}_L(e) \geq \mathbb{E}_{(\theta_1,\theta_2)\sim\mu}\rho(\theta_1, \theta_2)\left(\frac{1}{4} - \frac{1}{4}V(e(\theta_1), e(\theta_2))\right).$$

Using this in place of lemma 8 leads to Bayesian versions of theorems 15 and 16.

Appendix C. Canonical Loss Functions and their Convexification

Here we develop *general* representations of loss functions $L : \Theta \times A \to \mathbb{R}$. We assume that the set Θ is finite.

In many statistical problems, it is natural for the space of actions A to be the set of distributions over unknowns $\mathbb{P}(\Theta)$.

Definition 37 *A loss $L : \Theta \times \mathbb{P}(\Theta) \to \mathbb{R}$ is* proper *if for all distributions $P \in \mathbb{P}(\Theta)$,*

$$P \in \operatorname*{arg\,min}_{Q\in\mathbb{P}(\Theta)} \langle P, L(-, Q)\rangle_\Theta .$$

It is strictly proper *if P is the* unique *minimizer.*

A proper loss takes a prediction $Q \in \mathbb{P}(\Theta)$, and then penalizes the decision maker according to how much weight their prediction assigned to the unknown θ. Intuitively properness ensures that if the decision maker *knows* P, then they minimize their expected loss by *reporting* P. Proper losses constitute a well studied class of loss functions, that provide suitable surrogates for decision problems (Brier, 1950; Grünwald and Dawid, 2004; Zhang, 2004; Gneiting and Raftery, 2007; Dawid, 2007; Reid and Williamson, 2009a; Dawid, 2007; Ávila Pires et al., 2013).

As will be shown, all "sensible" losses are essentially re-parametrized proper losses. We show how to *construct* proper losses from their entropies. Furthermore, we show how to render any proper loss convex through a canonical re-parametrization. This allows the use of tools from convex analysis (Boyd and Vandenberghe, 2004; Lucchetti, 2006) to aid in calculating optimal actions.

C.1 Entropy from Loss

Rather than working with probability distributions, we take the route of Williamson (2014) and work with un-normalized distributions. Denote the set of all unnormalized distributions on Θ by $\mathbb{P}^+(\Theta)$. For any loss function L, define the *entropy* $\underline{L} : \mathbb{P}^+(\Theta) \to \mathbb{R}$,

$$\underline{L}(\mu) = \min_{a\in A}\langle \mu, L(-, a)\rangle_\Theta.$$

$\underline{L}(P)$ measures the uncertainty of the optimal action for the distribution P. The entropy is also called an *uncertainty function*, a *Bayes risk* or a *support function* (DeGroot, 1962; Williamson, 2014). It is concave and 1-homogeneous.

Definition 38 *A function $f : \mathbb{P}^+(\Theta) \to \mathbb{R}$ is 1-homogeneous if for all $x \in \mathbb{P}^+(\Theta)$ and for all $\lambda > 0$,*

$$f(\lambda x) = \lambda f(x).$$

C.2 Loss from Entropy

All loss functions give rise to an entropy. Conversely, the entropy encodes much information of its associated loss through its *super-gradients*, which include all the *Bayes* actions for the underlying loss.

C.2.1 Bayes Actions and Super-gradients

For any distribution P, define the *Bayes actions* for P as the set of minimizers,

$$A_P = \operatorname*{arg\,min}_{a\in A} \langle P, L(-, a)\rangle .$$

For any $a_P \in A_P$ we have $\underline{L}(P) = \langle P, L(-, a_P)\rangle$.

Definition 39 (Super-gradient of a concave function) *Let $f : \mathbb{P}^+(\Theta) \to \mathbb{R}$ be a concave function. $v \in \mathbb{R}^\Theta$ is a* super-gradient *of f at the point x if for all $y \in \mathbb{P}^+(\Theta)$,*

$$\langle y - x, v\rangle + f(x) \geq f(y).$$

Denote the set of all super-gradients at a point x by $\partial f(x)$, and the set of all super-gradients by $\partial f = \cup_x \partial f(x)$. For differentiable concave functions, super-gradients are the same as regular gradients (Lucchetti, 2006). 1-homogeneous functions afford a very simple representation via their super-gradients.

Theorem 40 (Generalized Euler's Homogeneous Function Theorem) *Let $f : \mathbb{P}^+(\Theta) \to \mathbb{R}$ be a concave 1-homogeneous function. Then for all x and for all $v \in \partial f(x)$,*

$$f(x) = \langle x, v\rangle.$$

Furthermore, $v \in \partial f(x) \implies v \in \partial f(\lambda x)$ for all $\lambda > 0$.

We include a simple proof of this theorem for completeness.
Proof Firstly, for all x and all $\lambda > 0$,

$$\langle \lambda x - x, v\rangle + f(x) \geq \lambda f(x),$$

which follows directly from the definition of a super-gradient at x and the 1-homogeneity of f. Re-arranging yields, $(1 - \lambda)(f(x) - \langle x, v\rangle) \geq 0$. Letting $\lambda \to 0^+$ yields $f(x) \geq \langle x, v\rangle$. Similarly, for all x and all $\lambda > 0$,

$$\langle x - \lambda x, v\rangle + \lambda f(x) \geq f(x),$$

which follows directly from the definition of a super-gradient at λx and the 1-homogeneity of f. Re-arranging yields, $(1-\lambda)(f(x)-\langle x,v\rangle)\leq 0$. Letting $\lambda\to 0^+$ yields $f(x)\leq\langle x,v\rangle$, therefore $f(x)=\langle x,v\rangle$.

To prove the second claim, we have for all y and $\lambda>0$,

$$\langle y-x,v\rangle+f(x)\geq f(y)$$
$$\langle \lambda y-\lambda x,v\rangle+f(\lambda x)\geq f(\lambda y),$$

where the first line is by definition, and the second is by 1-homogeneity. As y is arbitrary, the claim is proved.

■

This theorem provides a corollary, that shows the super-gradients of a 1-homogeneous function have a property similar to properness.

Corollary 41 *Let $f:\mathbb{P}^+(\Theta)\to\mathbb{R}$ be a concave 1-homogeneous function. Then for all $x,y\in\mathbb{P}^+(\Theta)$ and for all $v_x\in\partial f(x)$, $v_y\in\partial f(y)$,*

$$\langle x,v_y\rangle\geq\langle x,v_x\rangle\,.$$

We now show that the partial loss of a Bayes action is a super-gradient of $\underline{L}$.

Theorem 42 *For all loss functions L and distributions P, $a_P\in A_P\Leftrightarrow L(-,a_P)\in\partial\underline{L}(P)$.*

Proof For $a_P\in A_P$ we have for all $\mu\in\mathbb{P}^+(\Theta)$,

$$\langle\mu-P,L(-,a_P)\rangle+\underline{L}(P)=\langle\mu,L(-,a_P)\rangle\geq\min_{a\in A}\langle\mu,L(-,a)\rangle=\underline{L}(\mu).$$

Hence $L(-,a_P)\in\partial\underline{L}(P)$. For the converse, if $L(-,a_P)\in\partial\underline{L}(P)$ then,

$$\underline{L}(P)=\langle P,L(-,a_P)\rangle=\min_{a\in A}\langle P,L(-,a)\rangle\,,$$

meaning a is Bayes.

■

Therefore, once non-Bayes actions are discarded, we can identify a loss with a subset of $\partial\underline{L}$. Rather than working with a subset $\partial\underline{L}$, it is advantageous to consider *all* of $\partial\underline{L}$.

Definition 43 (Canonical Loss (Preliminary)) *Let $\underline{L}:\mathbb{P}^+(\Theta)\to\mathbb{R}$ be a concave, 1-homogeneous function. Then its* canonical loss, *$\mathcal{L}:\Theta\times\partial\underline{L}\to\mathbb{R}$ is given by, $\mathcal{L}(\theta,\zeta)=\zeta(\theta)$.*

As will be shown, canonical losses can always be convexified. Furthermore, they maintain all of the properties of L needed for assessing the quality of decisions.

C.2.2 THE BAYES SUPER PREDICTION SET

The process of *canonising* a loss, i.e. going from,

$$L\to\underline{L}\to\mathcal{L},$$

can create *extra* partial losses/actions that were not originally available to the decision maker. However, they gain no benefit from these extra actions. From any entropy define the *Bayes super prediction set*,

$$\mathcal{S}_{\underline{L}}:=\left\{\zeta\in\mathbb{R}^\Theta:\langle\mu,\zeta\rangle\geq\underline{L}(\mu),\ \forall\mu\in\mathbb{R}^\Theta_+\right\}.$$

By the definition,

$$\min_{a\in A}\langle P,L(-,a)\rangle=\min_{\zeta\in\mathcal{S}_{\underline{L}}}\langle P,\zeta\rangle\,,\ \forall P\in\mathbb{P}(\Theta).$$

The Bayes super prediction set is precisely those partial losses that the decision maker need not use over the actions available to them, no matter the distribution P. The super prediction set is convex. Furthermore, the Bayes actions for $\mathcal{L}$ are the lower boundary of the super prediction set.

Lemma 44 *Let $\underline{L}:\mathbb{P}^+(\Theta)\to\mathbb{R}$ be a concave, 1-homogeneous function. Then $\zeta\in\partial\underline{L}$ if and only if,*

$$\langle\mu,\zeta\rangle\geq\underline{L}(\mu),\ \forall\mu\in\mathbb{P}^+(\Theta),$$

with equality holding for at least one μ.

The proof is a straightforward application of 1-homogeneity and super-gradients.

Canonical losses use *all* super gradients of $\underline{L}$. Proper losses use some.

Corollary 45 (Loss from Entropy) *Let $\underline{L}:\mathbb{P}^+(\Theta)\to\mathbb{R}$ be a concave, 1-homogeneous function and let $\nabla\underline{L}:\mathbb{P}^+(\Theta)\to\mathbb{R}^\Theta$ be a super-gradient function, $\nabla\underline{L}(\mu)\in\partial\underline{L}(\mu)$, $\forall\mu$. Then,*

$$L(\theta,Q)=\mathcal{L}(\theta,\nabla\underline{L}(Q)),$$

is a proper loss. Furthermore if $\underline{L}$ is strictly concave then L is strictly proper.

C.3 Convexification of Losses in Canonical Form

The preceding shows how to *construct* losses, we begin with a concave 1-homogeneous function and take super-gradients. Focus now turns to their convexification. Once convexified, the decision maker gains access to the large and ever growing literature on the minimization of convex functions to aid in the calculation of optimal actions. We closely follow Dawid (2007), with a focus on canonical losses. This streamlines the development. For example, for some proper losses lemma 47 fails to hold. Furthermore, our result on convexification of canonical losses (theorem 49), is to the best of our knowledge novel.

Recall $\mathbf{1}_\Theta\in\mathbb{R}^\Theta$ is the function that always returns 1, and define $\mathbf{1}_\Theta^\perp$ to be its orthogonal complement in $\mathbb{R}^\Theta$, i.e. the functions $v\in\mathbb{R}^\Theta$ with,

$$\langle\mathbf{1}_\Theta,v\rangle=\sum_{z\in\Theta}v(z)=0.$$

Define,

$$\Gamma_{\underline{L}} = \{(\gamma, v) \in \mathbb{R} \times \mathbf{1}_\Theta^\perp : \gamma \mathbf{1}_\Theta + v \in \partial \underline{L}\}.$$

Lemma 46 *Let $(\gamma, v) \in \Gamma_{\underline{L}}$. Then γ is uniquely determined by v.*

Proof Fix v and suppose there exists γ_1 and γ_2 with $\gamma_1 < \gamma_2$ and $\gamma_1 \mathbf{1}_\Theta + v, \gamma_2 \mathbf{1}_\Theta + v \in \partial \underline{L}$. By assumption, $\gamma_2 \mathbf{1}_\Theta + v$ is Bayes for some distribution P. But,

$$\langle P, \gamma_1 \mathbf{1}_\Theta + v \rangle = \gamma_1 + \langle P, v \rangle < \gamma_2 + \langle P, v \rangle = \langle P, \gamma_2 \mathbf{1}_\Theta + v \rangle,$$

a contradiction. ■

Thus we lose nothing by working with projections of losses onto $\mathbf{1}_\Theta^\perp$. Define,

$$\hat{\Gamma}_{\underline{L}} = \mathrm{proj}_{\mathbf{1}_\Theta^\perp}(\partial \underline{L}) \subseteq \mathbf{1}_\Theta^\perp.$$

By lemma 46 $\hat{\Gamma}_{\underline{L}}$ is in 1-1 correspondence with $\partial \underline{L}$.

Lemma 47 *$\hat{\Gamma}_{\underline{L}}$ is a convex set.*

Proof To show $\hat{\Gamma}_{\underline{L}}$ is convex, we are required to show that for all $\zeta_1, \zeta_2 \in \partial \underline{L}$ and all $\lambda \in [0, 1]$ there is a constant γ such that,

$$\lambda \zeta_1 + (1 - \lambda)\zeta_2 - \gamma \mathbf{1}_\Theta \in \partial \underline{L}.$$

By lemma 44, this is equivalent to,

$$\underbrace{\lambda \langle P, \zeta_1 \rangle + (1 - \lambda) \langle P, \zeta_2 \rangle - \underline{L}(P)}_{\gamma(P)} - \gamma = \gamma(P) - \gamma \geq 0, \ \forall P \in \mathbb{P}(\Theta),$$

with equality holding for one P. Let $\gamma^* = \min_P \gamma(P)$, with P^* the distribution that achieves the minimum. Clearly $\gamma(P) - \gamma^* \geq 0$. Therefore,

$$\lambda \langle P, \zeta_1 \rangle + (1 - \lambda) \langle P, \zeta_2 \rangle - \gamma^* \geq \underline{L}(P), \ \forall P \in \mathbb{P}(\Theta),$$

with equality for P^*. Therefore by lemma 44, $\lambda \zeta_1 + (1 - \lambda)\zeta_2 - \gamma^* \mathbf{1}_\Theta \in \partial \underline{L}$. ■

Define the function $\Psi : \hat{\Gamma}_{\underline{L}} \to \mathbb{R}$ such that,

$$v + \Psi(v) \mathbf{1}_\Theta \in \partial \underline{L}.$$

By lemma 46, Ψ is well defined.

Lemma 48 *Ψ is a convex function.*

Proof Let $v_1, v_2 \in \hat{\Gamma}_{\underline{L}}$ with $v_\lambda = \lambda v_1 + (1 - \lambda) v_2$. Let their partial losses be,

$$\begin{aligned}
\zeta_1 &= v_1 + \Psi(v_1) \mathbf{1}_\Theta \\
\zeta_2 &= v_2 + \Psi(v_2) \mathbf{1}_\Theta \\
\zeta_\lambda &= \lambda v_1 + (1 - \lambda) v_2 + \Psi(\lambda v_1 + (1 - \lambda) v_2) \mathbf{1}_\Theta,
\end{aligned}$$

respectively. By assumption, for all $\lambda \in [0, 1]$ there exists a distribution P_λ such that,

$$\langle P_\lambda, \zeta_\lambda \rangle \leq \langle P_\lambda, \zeta \rangle, \ \forall \zeta \in \partial \underline{L}.$$

Assume there is a λ^* such that,

$$\lambda^* \Psi(v_1) + (1 - \lambda^*) \Psi(v_2) < \Psi(\lambda^* v_1 + (1 - \lambda^*) v_2).$$

But then,

$$\langle P_{\lambda^*}, \lambda^* \zeta_1 + (1 - \lambda^*) \zeta_2 \rangle < \langle P_{\lambda^*}, \zeta_{\lambda^*} \rangle,$$

a contradiction. ■

This gives the following representation theorem for canonical losses.

Theorem 49 (Representation of Canonical Losses) *Let $\underline{L} : \mathbb{P}^+(\Theta) \to \mathbb{R}$ be a concave, 1-homogeneous function. Then its canonical loss $\mathcal{L}$ can be represented as $\mathcal{L} : \Theta \times C \to \mathbb{R}$, with $C \subseteq \mathbf{1}_\Theta^\perp$ a convex set and,*

$$\mathcal{L}(\theta, v) = -\langle \delta_\theta, v \rangle + \Psi(v),$$

for a convex function Ψ.

The situation is particularly simple for binary problems.

Corollary 50 *Let $\Theta = \{\pm 1\}$. Then all canonical losses can be written as $\mathcal{L} : \{\pm 1\} \times I \to \mathbb{R}$,*

$$\mathcal{L}(\theta, v) = -\theta v + \Psi(v),$$

where I is a convex subset of $\mathbb{R}$ and $\Psi : I \to \mathbb{R}$ a convex function.

The proof comes from the observation that,

$$\mathbf{1}_\Theta^\perp = \mathrm{Span}((-1, 1)).$$

C.4 Illustrative Example: Binary Decisions and Square Loss

In binary problems, $\Theta = \{\pm 1\}$. An often used loss is the square or Brier loss. $L(\theta, \hat{\theta}) = \left(\theta - \hat{\theta}\right)^2$, where $\hat{\theta} \in [-1, 1]$. We plot this loss in figure 3. The partial losses are given by the red curve, the super prediction set in gray. The loss on negatives is plotted on the x-axis. In figure 4 we show geometrically how to produce canonical coordinates.

We seek to decompose,

$$L(-, \hat{\theta}) = \left((-1 - \hat{\theta})^2, (1 - \hat{\theta})^2\right) = \alpha \begin{pmatrix} 1 \\ -1 \end{pmatrix} + \beta \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Solving for α and β in terms of $\hat{\theta}$ gives,

$$\alpha = 2\hat{\theta} \text{ and } \beta = 1 + \hat{\theta}^2.$$

These equations can be easily solved for $\hat{\theta}$ giving,

$$\beta = \Psi(\alpha) = 1 + \frac{\alpha^2}{4}.$$

Therefore the canonical form of square loss is,

$$\mathcal{L}(\theta, \alpha) = -\theta\alpha + 1 + \frac{\alpha^2}{4}.$$

Notice that the only dependence on θ is via the "linear" term $-\theta\alpha$.

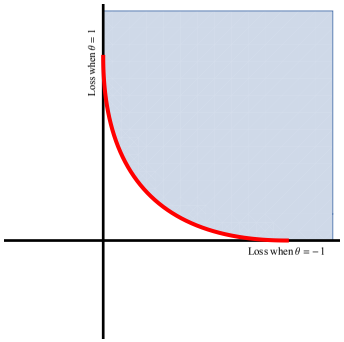


Figure 3: Plot of super prediction set and its lower boundary for square loss, see text.

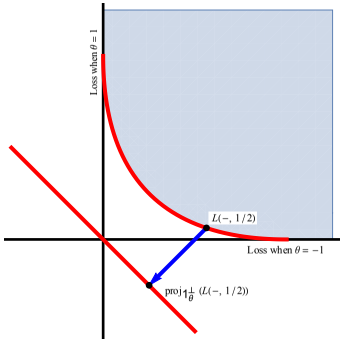


Figure 4: Construction of canonical coordinates for square loss, see text.

C.5 Illustrative Example: Hinge Loss

Hinge loss $L : \{-1, 1\} \times \mathbb{R} \to \mathbb{R}_+$,

$$L(\theta, \nu) = \max(0, 1 - \theta\nu),$$

is an often used, non-differentiable, loss function. Its minimization forms the basis of the support vector machine learning algorithm (Cortes and Vapnik, 1995; Schölkopf and Smola, 2002; Steinwart and Christmann, 2008), a popular method for learning classifiers. Here we show that the canonical form of hinge loss is the *linear loss*,

$$\mathcal{L}(\theta, \nu) = -\theta\nu, \ \nu \in [-1, 1].$$

Figure 5 is a plot of hinge loss and its super prediction set. Shown in orange are actions with $|\nu| \geq 1$. These actions are *inadmissible* and therefore not Bayes for any distribution over Θ. Figure 6 shows the projection of the Bayes actions for hinge loss unto $\mathbf{1}_\Theta^\perp$. As can be seen from the figure, linear loss is the result of "canonizing" hinge loss.

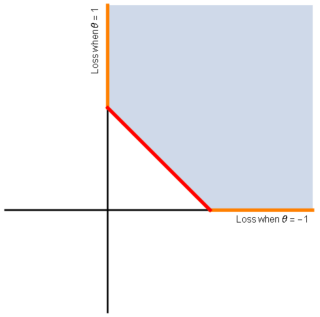


Figure 5: Plot of super prediction set and its lower boundary for hinge loss, see text (best viewed in color).

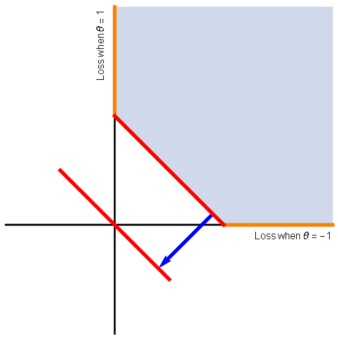


Figure 6: Construction of canonical coordinates for square loss, see text.

C.6 Proof of Theorem 29

Proof As $\mathcal{L}$ is canonical, its partial loss function is given by $\mathcal{L}(-,v) = v + \Psi(v)\mathbf{1}_{\mathcal{Z}}$. By definition,

$$\mathcal{L}_R(-,v) = R^*(\mathcal{L}(-,v)) = R^*(v) + \Psi(v)R^*(\mathbf{1}_{\mathcal{Z}}).$$

If $|\mathcal{Z}| = |\tilde{\mathcal{Z}}|$, then T is reconstructible if and only if T is invertible. As T is column stochastic,

$$\mathbf{1}_{\mathcal{Z}} = T^*(\mathbf{1}_{\tilde{\mathcal{Z}}}).$$

This yields,

$$R^*(\mathbf{1}_{\mathcal{Z}}) = R^*T^*(\mathbf{1}_{\tilde{\mathcal{Z}}}) = \mathbf{1}_{\tilde{\mathcal{Z}}}.$$

For the more general case where $|\mathcal{Z}| < |\tilde{\mathcal{Z}}|$, we have for all T and all $v \in \mathbf{1}_{\tilde{\mathcal{Z}}}^{\perp}$,

$$\begin{aligned}\left\langle T(v), \mathbf{1}_{\tilde{\mathcal{Z}}}\right\rangle &= \left\langle v, T^*(\mathbf{1}_{\tilde{\mathcal{Z}}})\right\rangle \\ &= \langle v, \mathbf{1}_{\mathcal{Z}}\rangle \\ &= 0.\end{aligned}$$

Therefore $T(\mathbf{1}_{\tilde{\mathcal{Z}}}^{\perp}) \subseteq \mathbf{1}_{\tilde{\mathcal{Z}}}^{\perp}$. As left inverses are not unique, we can further restrict R to those with $R(\mathbf{1}_{\tilde{\mathcal{Z}}}^{\perp}) \subseteq \mathbf{1}_{\tilde{\mathcal{Z}}}^{\perp}$, or dually those with $R^*(\mathbf{1}_{\mathcal{Z}}) = \mathbf{1}_{\tilde{\mathcal{Z}}}$. There is always such an R, as the restriction of T to $\mathbf{1}_{\tilde{\mathcal{Z}}}^{\perp}$ is also left invertible. Furthermore, $T(\mathbf{1}_{\mathcal{Z}}) \notin \mathbf{1}_{\tilde{\mathcal{Z}}}^{\perp}$, as $T(\mathbf{1}_{\mathcal{Z}})$ nonnegative entries. Therefore, we can take R restricted to $\mathbf{1}_{\tilde{\mathcal{Z}}}^{\perp}$ to be the left inverse of T restricted to $\mathbf{1}_{\tilde{\mathcal{Z}}}^{\perp}$, with $RT(\mathbf{1}_{\mathcal{Z}}) = \mathbf{1}_{\mathcal{Z}}$. Such an R can then be extended to all of $\mathbb{R}^{\tilde{\mathcal{Z}}}$. Finally, by definition, $\mathcal{L}_R(\tilde{z},v) = \langle \delta_{\tilde{z}}, \mathcal{L}_R(-,v)\rangle$, yielding,

$$\begin{aligned}\mathcal{L}_R(\tilde{z},v) &= \langle \delta_{\tilde{z}}, R^*(v)\rangle + \langle \delta_{\tilde{z}}, R^*(\mathbf{1}_{\mathcal{Z}})\rangle \Psi(v) \\ &= \langle R(\delta_{\tilde{z}}), v\rangle + \Psi(v),\end{aligned}$$

where the last line is by properties of adjoints. This function is the sum of two functions, one linear in v the other convex and is therefore convex in v. ■

C.7 Illustrative Example: Multi-class Logistic Loss and its Corrections

Here we show by example how to apply theorem 29 to construct losses for multiclass corrupted learning problems. We take $\Theta = \{1,2,3\}$, with L the log loss,

$$L(-,P) = (-\log(P_1), -\log(P_2), -\log(P_3))\,.$$

We express $v \in \mathbf{1}_{\Theta}^{\perp}$ as,

$$v = v_1 \begin{pmatrix} -\frac{2}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{pmatrix} + v_2 \begin{pmatrix} \frac{1}{3} \\ -\frac{2}{3} \\ \frac{1}{3} \end{pmatrix}.$$

Note that these basis vectors are the projections onto $\mathrm{id}_{\Theta}^{\perp}$ of the partial losses,

$$\begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \text{ and } \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix},$$

respectively. We have,

$$P = \left(\frac{e^{v_1}}{1+e^{v_1}+e^{v_2}}, \frac{e^{v_2}}{1+e^{v_1}+e^{v_2}}, \frac{1}{1+e^{v_1}+e^{v_2}}\right).$$

with,

$$\begin{aligned}\mathcal{L}(-,v) &= \begin{pmatrix} -v_1 + \log(1+e^{v_1}+e^{v_2}) \\ -v_2 + \log(1+e^{v_1}+e^{v_2}) \\ \log(1+e^{v_1}+e^{v_2}) \end{pmatrix} \\ &= v + \underbrace{\left(-\frac{1}{3}(v1+v2) + \log(1+e^{v_1}+e^{v_2})\right)}_{\Psi(v)} \mathbf{1}_{\Theta}.\end{aligned} \tag{1}$$

(1) is the usual form of multiclass logistic loss. For the case of symmetric noise,

$$T = \begin{pmatrix} 1-\sigma & \frac{\sigma}{2} & \frac{\sigma}{2} \\ \frac{\sigma}{2} & 1-\sigma & \frac{\sigma}{2} \\ \frac{\sigma}{2} & \frac{\sigma}{2} & 1-\sigma \end{pmatrix} \text{ and } R = \begin{pmatrix} \frac{2-\sigma}{2-3\sigma} & \frac{-\sigma}{2-3\sigma} & \frac{-\sigma}{2-3\sigma} \\ \frac{-\sigma}{2-3\sigma} & \frac{2-\sigma}{2-3\sigma} & \frac{-\sigma}{2-3\sigma} \\ \frac{-\sigma}{2-3\sigma} & \frac{-\sigma}{2-3\sigma} & \frac{2-\sigma}{2-3\sigma} \end{pmatrix},$$

respectively. All $v \in \mathbf{1}_{\Theta}^{\perp}$ are eigenvalues of R, with eigenvalue $\frac{2}{2-3\sigma}$, giving a corruption corrected loss of,

$$\mathcal{L}_R(-,v) = \frac{2}{2-3\sigma}v + \Psi(v)\mathrm{id}_{\Theta}.$$

Recall for the problem of learning partial labels $\tilde{\Theta} = \{0,1\}^{\Theta}$ the set of all partial labellings. Under the assumption that the partial label always includes the correct underlying label, and that spurious

labels are added with probability σ,

$$T = \begin{pmatrix} (1-\sigma)^2 & 0 & 0 \\ 0 & (1-\sigma)^2 & 0 \\ 0 & 0 & (1-\sigma)^2 \\ (1-\sigma)\sigma & (1-\sigma)\sigma & 0 \\ (1-\sigma)\sigma & 0 & (1-\sigma)\sigma \\ 0 & (1-\sigma)\sigma & (1-\sigma)\sigma \\ \sigma^2 & \sigma^2 & \sigma^2 \end{pmatrix}.$$

For this problem, there are several different alternatives for R. The reconstruction,

$$R = \begin{pmatrix} 1 & 0 & 0 & \frac{3-2\sigma}{3-3\sigma} & \frac{3-2\sigma}{3-3\sigma} & \frac{\sigma-3}{3(1-\sigma)} & \frac{1}{3} \\ 0 & 1 & 0 & \frac{3-2\sigma}{3-3\sigma} & \frac{\sigma-3}{3(1-\sigma)} & \frac{3-2\sigma}{3-3\sigma} & \frac{1}{3} \\ 0 & 0 & 1 & \frac{\sigma-3}{3(1-\sigma)} & \frac{3-2\sigma}{3-3\sigma} & \frac{3-2\sigma}{3-3\sigma} & \frac{1}{3} \end{pmatrix}, \ \sigma \neq 1,$$

is a left inverse for T and satisfies the requirements of theorem 29. For this reconstruction one has,

$$R^* \begin{pmatrix} -\frac{2}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{pmatrix} = \begin{pmatrix} -\frac{2}{3} \\ \frac{1}{3} \\ \frac{1}{3} \\ \frac{\sigma-2}{3(1-\sigma)} \\ \frac{\sigma-2}{3(1-\sigma)} \\ \frac{2(2-\sigma)}{3(1-\sigma)} \\ 0 \end{pmatrix} \text{ and } R^* \begin{pmatrix} \frac{1}{3} \\ -\frac{2}{3} \\ \frac{1}{3} \end{pmatrix} = \begin{pmatrix} \frac{1}{3} \\ -\frac{2}{3} \\ \frac{1}{3} \\ \frac{\sigma-2}{3(1-\sigma)} \\ \frac{2(2-\sigma)}{3(1-\sigma)} \\ \frac{\sigma-2}{3(1-\sigma)} \\ 0 \end{pmatrix},$$

leading to the loss,

$$\mathcal{L}_R(-, v) = v_1 \begin{pmatrix} -\frac{2}{3} \\ \frac{1}{3} \\ \frac{1}{3} \\ \frac{\sigma-2}{3(1-\sigma)} \\ \frac{\sigma-2}{3(1-\sigma)} \\ \frac{2(2-\sigma)}{3(1-\sigma)} \\ 0 \end{pmatrix} + v_2 \begin{pmatrix} \frac{1}{3} \\ -\frac{2}{3} \\ \frac{1}{3} \\ \frac{\sigma-2}{3(1-\sigma)} \\ \frac{2(2-\sigma)}{3(1-\sigma)} \\ \frac{\sigma-2}{3(1-\sigma)} \\ 0 \end{pmatrix} + \Psi(v)\mathrm{id}_{\tilde{\Theta}}.$$

Appendix D. Proofs of Theorems in Main Text

D.1 Proof of Theorem 7

We actually prove a more general theorem, that works for infinite action sets.

Theorem 51 *Let $T_i \in \mathbb{T}(\mathcal{Z}, \tilde{\mathcal{Z}}_i)$ be a collection of k reconstructible corruptions. Let $\tilde{P} = \otimes_{i=i}^k \tilde{P}_i^{n_i}$, $\tilde{\mathcal{Z}} = \times_{i=1}^k \tilde{\mathcal{Z}}_i^{n_i}$, $n = \sum_{i=1}^k n_i$ and $r_i = \frac{n_i}{n}$. Then for all algorithms $\mathcal{A} \in \mathbb{T}(\tilde{\mathcal{Z}}, A)$, priors $\pi \in \mathbb{P}(A)$, distributions $P \in \mathbb{P}(\mathcal{Z})$ and bounded loss functions ℓ,*

$$\mathbb{E}_{\tilde{S}\sim\tilde{P}}\ell(P, \mathcal{A}(\tilde{S})) \leq \mathbb{E}_{\tilde{S}\sim\tilde{P}} \sum_{i=1}^k r_i \ell_{R_i}(\tilde{S}_i, \mathcal{A}(\tilde{S})) + K\sqrt{\frac{2\mathbb{E}_{S\sim P^n} D_{KL}(\mathcal{A}(\tilde{S}), \pi)}{n}}.$$

where $K = \sqrt{\sum_{i=1}^k r_i \|\ell_{R_i}\|_\infty^2}$.

Proof Define $L(\tilde{S}, a) = \sum_{i=1}^k \sum_{\tilde{z}\in\tilde{S}_i} \ell_{R_i}(z_i, a)$, the sum of the corrupted losses on the sample. By theorem 2.1 of Zhang (2006) for $\beta > 0$,

$$\mathbb{E}_{\tilde{S}\sim Q}\mathbb{E}_{a\sim\mathcal{A}(\tilde{S})} - \frac{1}{\beta}\log(\mathbb{E}_{\tilde{S}'\sim Q} e^{-\beta L(\tilde{S}', a)}) \leq \mathbb{E}_{\tilde{S}\sim Q}\left[L(\tilde{S}, \mathcal{A}(\tilde{S})) + \frac{D_{KL}(\mathcal{A}(\tilde{S}), \pi)}{\beta}\right]$$

$$\sum_{i=1}^k n_i \mathbb{E}_{\tilde{S}\sim Q}\mathbb{E}_{a\sim\mathcal{A}(\tilde{S})} - \frac{1}{\beta}\log(\mathbb{E}_{\tilde{z}\sim\tilde{P}_i} e^{-\beta\ell_{R_i}(\tilde{z}, a)}) \leq \mathbb{E}_{\tilde{S}\sim Q}\left[L(\tilde{S}, \mathcal{A}(\tilde{S})) + \frac{D_{KL}(\mathcal{A}(\tilde{S}), \pi)}{\beta}\right]$$

where the first line follows from the theorem and the second from properties of the cumulant generating function. Invoking lemma A.1 of Cesa-Bianchi and Lugosi (2006) yields,

$$\sum_{i=1}^k n_i \left(\mathbb{E}_{\tilde{S}\sim Q}\ell_{R_i}(\tilde{P}_i, \mathcal{A}(\tilde{S})) - \frac{\|\ell_{R_i}\|_\infty^2 \beta}{2}\right) \leq \mathbb{E}_{\tilde{S}\sim Q}\left[L(\tilde{S}, \mathcal{A}(\tilde{S})) + \frac{D_{KL}(\mathcal{A}(\tilde{S}), \pi)}{\beta}\right].$$

As the T_i are reconstructible,

$$\mathbb{E}_{\tilde{S}\sim Q}\ell(P, \mathcal{A}(\tilde{S})) \leq \frac{1}{n}\mathbb{E}_{\tilde{S}\sim Q}\left[L(\tilde{S}, \mathcal{A}(\tilde{S})) + \frac{D_{KL}(\mathcal{A}(\tilde{S}), \pi)}{\beta}\right] + \frac{\left(\sum_{i=1}^k r_i \|\ell_{R_i}\|_\infty^2\right)\beta}{2}.$$

Optimizing over β yields the desired result. ■

Theorem 7 is recovered by taking A finite, π uniform on A and upper bounding $D_{KL}(\mathcal{A}(S), \pi) \leq \log(|A|)$.

D.2 Proof of Lemma 10

Proof Firstly V is a *metric* on $\mathbb{P}(\times_{n=1}^k \mathcal{Z}_i)$ (Reid and Williamson, 2009b). Thus,

$$\begin{aligned} V(\otimes_{i=1}^k P_i, \otimes_{i=1}^k Q_i) &= V(P_1 \otimes (\otimes_{i=2}^k P_i), Q_1 \otimes (\otimes_{i=2}^k Q_i)) \\ &\leq V(P_1 \otimes (\otimes_{i=2}^k P_i), Q_1 \otimes (\otimes_{i=2}^k P_i)) + V(Q_1 \otimes (\otimes_{i=2}^k P_i), Q_1 \otimes (\otimes_{i=2}^k Q_i)) \\ &= V(P_1, Q_1) + V(\otimes_{i=2}^k P_i, \otimes_{i=2}^k Q_i), \end{aligned}$$

where the first line is by definition, the second as V is a metric and the third is easily verified from the definition of V. To complete the proof proceed inductively. ■

D.3 Proof of Theorem 16

Proof Let

$$T = \otimes_{i=i}^{k} T_i^{n_i} = \underbrace{T_1 \otimes \cdots \otimes T_1}_{n_1 \text{ times}} \otimes \underbrace{T_2 \otimes \cdots \otimes T_2}_{n_2 \text{ times}} \otimes \cdots \otimes \underbrace{T_k \otimes \cdots \otimes T_k}_{n_k \text{ times}}.$$

One has $T(e_n(\theta)) = T_1(e(\theta))^{n_1} \otimes T_2(e(\theta))^{n_2} \otimes \cdots \otimes T_k(e(\theta))^{n_k}$. By lemma 10,

$$\begin{aligned} V(T(e_n(\theta_1)), T(e_n(\theta_2)) &\leq \sum_{i=1}^{k} n_i V(T_i(e(\theta_1)), T_i(e(\theta_2))) \\ &\leq \underbrace{\left(\sum_{i=1}^{k} \text{Clarity}(T_i) n_i \right)}_{K} V(e(\theta_1), e(\theta_2)). \end{aligned}$$

Rearranging gives,

$$\rho(\theta_1, \theta_2) \left(1 - \gamma V(T(e_n(\theta_1), T(e_n(\theta_2))\right) \geq \rho(\theta_1, \theta_2) \left(1 - K\gamma V(T(e(\theta_1), T(e(\theta_2))\right), \ \forall \gamma > 0.$$

Taking supremum's over θ_1, θ_2 yields,

$$\text{Le Cam}_L \left(T \circ e_n, \gamma\right) \geq \text{Le Cam}_L \left(e, K\gamma\right)$$

Applying lemma 11 yields the result. ■

D.4 Proof of Lemma 13

Proof

$$\begin{aligned} \text{Clarity}(T_2 T_1) &= \sup_{P,Q \in \mathbb{P}(\mathcal{Z})} \frac{\|T_2 \circ T_1(P) - T_2 \circ T_1(Q)\|_1}{\|P - Q\|_1} \\ &= \sup_{P,Q \in \mathbb{P}(\mathcal{Z})} \frac{\|T_2 \circ T_1(P) - T_2 \circ T_1(Q)\|_1}{\|T_1(P) - T_2(Q)\|_1} \frac{\|T_1(P) - T_2(Q)\|_1}{\|P - Q\|_1} \\ &\leq \sup_{P,Q \in \mathbb{P}(\mathcal{Z})} \frac{\|T_2 \circ T_1(P) - T_2 \circ T_1(Q)\|_1}{\|T_1(P) - T_2(Q)\|_1} \sup_{P,Q \in \mathbb{P}(\mathcal{Z})} \frac{\|T_1(P) - T_2(Q)\|_1}{\|P - Q\|_1} \\ &\leq \sup_{P,Q \in \mathbb{P}(\tilde{\mathcal{Z}}_1)} \frac{\|T_2(P) - T_2(Q)\|_1}{\|P - Q\|_1} \sup_{P,Q \in \mathbb{P}(\mathcal{Z})} \frac{\|T_1(P) - T_2(Q)\|_1}{\|P - Q\|_1} \\ &= \text{Clarity}(T_2)\text{Clarity}(T_1), \end{aligned}$$

where the first line follows from the definitions, the second follows if $T_1(P) \neq T_2(Q)$ and the rest are simple rearrangements. For the final inequality, remember that $\text{Clarity}(T) \leq 1$. ■

D.5 Proof of Lemma 19

Proof By definition $\|\tilde{\ell}\|_\infty = \sup_{z,a} |\tilde{\ell}(z,a)| = \sup_a \|\tilde{\ell}_a\|_\infty$. Hence,

$$\begin{aligned} \|\tilde{\ell}\|_\infty &= \sup_a \|\tilde{\ell}_a\|_\infty \\ &\leq \sup_a \|R^*\|_\infty \|\ell_a\|_\infty \\ &= \|R^*\|_\infty \|\ell\|_\infty, \end{aligned}$$

where the second line follows from the definition of the operator norm $\|R^*\|_\infty$. ■

D.6 Proof of Lemma 20

Proof Firstly $\|R\|_1 = \|R^*\|_\infty$ (Bernstein, 2009). From the definition of $\|R\|_1$ we have,

$$\begin{aligned} \|R\|_1 &= \sup_{v \in \mathbb{R}^Y} \frac{\|Rv\|_1}{\|v\|_1} \\ &\geq \sup_{u \in \mathbb{R}^X} \frac{\|RTu\|_1}{\|Tu\|_1} \\ &= \sup_{u \in \mathbb{R}^X} \frac{\|u\|_1}{\|Tu\|_1} \\ &= 1 / \left(\inf_{u \in \mathbb{R}^X} \frac{\|Tu\|_1}{\|u\|_1} \right). \end{aligned}$$

This proves the first inequality. Recall,

$$\text{Clarity}(T) = \sup_{v \in \Omega} \frac{\|T(v)\|_1}{\|v\|_1},$$

where $\Omega = \{v \in \mathbb{R}^X : \sum v_i = 0, v \neq 0\}$. Hence $\inf_{u \in \mathbb{R}^X} \frac{\|Tu\|_1}{\|u\|_1} \leq \text{Clarity}(T)$. ■

D.7 Proof of Theorem 23

Proof First, define $\ell_P(z,a) = \ell(z,a) - \ell(z,a_P)$. ℓ_P measures the loss relative to the best action for the distribution P. It is easy to verify that for bounded ℓ, $\|\ell_P\|_\infty \leq 2\|\ell\|_\infty$. We now utilize theorem 2.1 of Zhang (2006), together with a lower bound presented in the appendix, with ℓ_P and π uniform on A. This yields,

$$\mathbb{E}_{S \sim P^n} \left[\ell_P(P, \mathcal{A}(S)) - \gamma \mathbb{E}_{z \sim P} \ell_P^2(z, \mathcal{A}(S)) \right] \leq \frac{1}{n} \mathbb{E}_{S \sim P^n} \left[\ell_P(S, \mathcal{A}(S)) + \|\ell_P, \|_\infty \left(\frac{\log(|A|)}{\beta} \right) \right],$$

with $\gamma = \frac{(e^\beta - 1 - \beta)}{\beta \|\ell_P\|_\infty}$. Firstly ERM minimizes the right hand side of the bound meaning,

$$\frac{1}{n} \mathbb{E}_{S \sim P^n} \left[\ell_P(S, \mathcal{A}(S)) + \|\ell_P\|_\infty \left(\frac{\log(|A|)}{\beta} \right) \right] \leq \frac{1}{n} \left[\|\ell_P\|_\infty \left(\frac{\log(|A|)}{\beta} \right) \right].$$

To see this consider the algorithm that always outputs a_P, this algorithm generalizes very well however it may be suboptimal on the sample. Secondly (ℓ, P) satisfies the Bernstein condition with constant K. Therefore,

$$(1-\gamma K)\mathbb{E}_{S\sim P^n}\ell_P(P,\mathcal{A}(S)) \leq \frac{1}{n}\left[\|\ell_P\|_\infty\left(\frac{\log(|A|)}{\beta}\right)\right].$$

Finally chose β small enough so that $\gamma K \leq 1$. This can always be done as $\gamma \to 0$ as $\beta \to 0_+$. The high probability version proceeds in a similar way.

■

D.8 Proof of Lemma 24

Proof

$$\begin{aligned}
K\mathbb{E}_{z\sim P}\ell(z,a) - \ell(z,a_P) &= K\mathbb{E}_{\tilde{z}\sim\tilde{P}}\ell_R(z,a) - \ell_R(z,a_P)\\
&\geq \mathbb{E}_{\tilde{z}\sim\tilde{P}}(\ell_R(\tilde{z},a) - \ell_R(\tilde{z},a_P))^2\\
&= \mathbb{E}_{z\sim P}\mathbb{E}_{\tilde{z}\sim T(z)}(\ell_R(\tilde{z},a) - \ell_R(\tilde{z},a_P))^2\\
&\geq \mathbb{E}_{z\sim P}(\mathbb{E}_{\tilde{z}\sim T(z)}\ell_R(\tilde{z},a) - \mathbb{E}_{\tilde{z}\sim T(z)}\ell_R(\tilde{z},a_P))^2\\
&= \mathbb{E}_{z\sim P}(\ell(z,a) - \ell(z,a_P))^2,
\end{aligned}$$

where the first line follows from the definition of ℓ and because $a_P = a_{\tilde{P}}$, the second as $(\ell_R, \tilde{P})$ satisfies the Bernstein condition and finally we have used the convexity of $f(x) = x^2$.

■

D.9 Proof of Theorem 26

Proof

$$\begin{aligned}
\mathbb{E}_{\tilde{z}\sim\tilde{P}}(\ell_R(\tilde{z},a) - \ell_R(\tilde{z},a_P))^2 &= \mathbb{E}_{z\sim P}\mathbb{E}_{\tilde{z}\sim T(z)}(\ell_R(\tilde{z},a) - \ell_R(\tilde{z},a_P))^2\\
&\leq \eta\mathbb{E}_{z\sim P}(\ell(z,a) - \ell(z,a_P))^2\\
&\leq \eta K\mathbb{E}_{z\sim P}\ell(z,a) - \ell(z,a_P)\\
&= \eta K\mathbb{E}_{\tilde{z}\sim\tilde{P}}\ell_R(\tilde{z},a) - \ell_R(\tilde{z},a_P),
\end{aligned}$$

where we have first used η-compatibility, then the fact that (ℓ, P) satisfies the Bernstein condition with constant K and finally the definition of ℓ_R.

■

D.10 Proof of Theorem 27

Proof Due to the symmetry of the left and right hand sides of the Bernstein condition, one only needs to check the case where $a_1 = 1$, $a_2 = -1$. Recall,

$$\begin{aligned}
\ell_{01,T}(\tilde{y},a) &= \frac{(1-\sigma_{-y})\ell_{01}(\tilde{y},a) - \sigma_y\ell_{01}(-\tilde{y},a)}{1-\sigma_{-1}-\sigma_1}\\
&= \frac{(1-\sigma_{-y}+\sigma_y)\ell_{01}(\tilde{y},a) - \sigma_y}{1-\sigma_{-1}-\sigma_1}.
\end{aligned}$$

For $y = 1$ it is easy to confirm $(\ell_{01}(1,1) - \ell_{01}(1,-1))^2 = 1$. We have,

$$\begin{aligned}
\ell_{01,T}(\tilde{y},1) - \ell_{01,T}(\tilde{y},-1) &= \frac{(1-\sigma_{-y}+\sigma_y)(\ell_{01}(\tilde{y},1) - \ell_{01}(\tilde{y},-1))}{1-\sigma_{-1}-\sigma_1}\\
&= \frac{-\tilde{y}(1-\sigma_{-y}+\sigma_y)}{1-\sigma_{-1}-\sigma_1}.
\end{aligned}$$

Squaring, taking maxima and finally expectations yields the desired result.

■

D.11 Proof of Corollary 35

Proof Let,

$$\ell(-,a) = v(a) + \gamma\mathbf{1}_{\mathcal{Z}},$$

where $v(a) \in E_{\alpha,S}\ \forall a \in A$ and $\gamma \in \mathbb{R}$. Therefore for all $M \in S$ and $\forall a \in A$,

$$\begin{aligned}
M\ell(-,a) &= Mv(a) + \gamma M\mathbf{1}_{\mathcal{Z}}\\
&= \alpha(M)v(a) + \gamma\mathbf{1}_{\mathcal{Z}}\\
&= \underbrace{\alpha(M)\left(v(a) + \gamma\mathbf{1}_{\mathcal{Z}}\right)}_{\alpha(M)\ell(-,a)} + \underbrace{\left(\gamma - \gamma\alpha(M)\right)\mathbf{1}_{\mathcal{Z}}}_{\beta(M)\mathbf{1}_{\mathcal{Z}}},
\end{aligned}$$

where the second line follows as by assumption $M\mathbf{1}_{\mathcal{Z}} = \mathbf{1}_{\mathcal{Z}}$ for all $M \in S$.

■

References

Alekh Agarwal, Peter L. Bartlett, Pradeep Ravikumar, and Martin J. Wainwright. Information-Theoretic Lower Bounds on the Oracle Complexity of Stochastic Convex Optimization. *IEEE Transactions on Information Theory*, 58(5):3235, 2012.

Syed Mumtaz Ali and Samuel D. Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 131–142, 1966.

Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

Javed A. Aslam and Scott E. Decatur. On the sample complexity of noise-tolerant learning. *Information Processing Letters*, 57(4):189–195, 1996.

Jean-Yves Audibert and Alexandre B. Tsybakov. Fast learning rates for plug-in classifiers. *The Annals of Statistics*, 35(2):608–633, 2007.

Bernardo Ávila Pires, Csaba Szepesvari, and Mohammad Ghavamzadeh. Cost-sensitive multiclass classification risk bounds. In *Proceedings of The 30th International Conference on Machine Learning*, pages 1391–1399, 2013.

Maria-Florina Balcan and Avrim Blum. A discriminative model for semi-supervised learning. *Journal of the ACM*, 57(3):19, 2010.

Peter L. Bartlett and Shahar Mendelson. Empirical minimization. *Probability Theory and Related Fields*, 135(3):311–334, 2006.

Peter L. Bartlett, Sanjeev R. Kulkarni, and S. Eli Posner. Covering numbers for real-valued function classes. *IEEE transactions on information theory*, 43(5):1721–1724, 1997.

Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Dennis S. Bernstein. *Matrix mathematics: Theory, Facts and Formulas*. Princeton University Press, 2009.

David Blackwell. Comparison of experiments. In *Second Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 93–102, 1951.

Gilles Blanchard and Clayton Scott. Decontamination of Multually Contaminated Models. In *AISTATS*, 2014.

Gilles Blanchard, Marek Flaska, Gregory Handy, Sara Pozzi, and Clayton Scott. Classification with asymmetric label noise: Consistency and maximal denoising. *Electronic Journal of Statistics*, 10 (2):2780–2824, 2016.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning and Games*. Cambridge University Press Cambridge, 2006.

Joseph T. Chang and D. Pollard. Conditioning as disintegration. *Statistica Neerlandica*, 51(3): 287–317, November 1997.

Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. *Semi-Supervised Learning*. MIT Press, 2010.

Nikolai Nikolaevich Chentsov. *Statistical decision rules and optimal inference*. American Mathematical Society, 1982.

J Cid-Sueiro. Proper losses for learning from partial labels. *Advances in Neural Information Processing System*, pages 1–9, 2012.

Jesús Cid-Sueiro, Darío García-García, and Raúl Santos-Rodríguez. Consistency of Losses for Learning from Weak Labels. In *Machine Learning and Knowledge Discovery in Databases*, pages 197–210. Springer, 2014.

Joel E. Cohen and Johannes H. B. Kempermann. *Comparisons of Stochastic Matrices with Applications in Information Theory, Statistics, Economics and Population Sciences*. Springer, 1998.

Joel E Cohen, Yoh Iwasa, Gh Rautu, Mary Beth Ruskai, Eugene Seneta, and Gh Zbaganu. Relative entropy under mappings by stochastic matrices. *Linear algebra and its applications*, 179:211–235, 1993.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Timothee Cour, Benn Sapp, and Ben Taskar. Learning from partial labels. *Journal of Machine Learning Research*, 12:1501–1536, 2011.

Koby Crammer, Michael Kearns, and Jennifer Wortman. Learning from data of variable quality. In *Advances in Neural Information Processing Systems*, pages 219–226, 2005.

George B. Dantzig. Discrete-variable extremum problems. *Operations research*, 5(2):266–288, 1957.

A. Phillip Dawid. The geometry of proper scoring rules. *Annals of the Institute of Statistical Mathematics*, (April 2006):77–93, 2007.

Morris H. DeGroot. Uncertainty, information, and sequential experiments. *The Annals of Mathematical Statistics*, 33(2):404–419, 1962.

Roland L. Dobrushin. Central limit theorem for nonstationary Markov chains. I. *Theory of Probability and its Applications*, 1(1):65–80, 1956.

John C. Duchi, Michael Jordan, and Martin J. Wainwright. Local privacy and statistical minimax rates. In *IEEE Symposium on the Foundations of Computer Science (FOCS)*, pages 429–438. IEEE, 2013.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Peter D. Grünwald and A. Philip Dawid. Game theory, maximum entropy, minimum discrepancy and robust Bayesian decision theory. *The Annals of Statistics*, 32(4):1367–1433, 2004.

Adityanand Guntuboyina. *Minimax Lower Bounds*. PhD thesis, Yale, 2011.

Julian Katz-Samuels and Clayton Scott. A mutual contamination analysis of mixed membership and partial label models. *arXiv preprint arXiv:1602.06235*, 2016.

Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45 (6):983–1006, 1998.

Lucien Le Cam. Sufficiency and approximate sufficiency. *The Annals of Mathematical Statistics*, 35(4):1419–1455, 1964.

Roberto Lucchetti. *Convexity and well-posed problems*. Springer, 2006.

Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. *Advances in neural information processing systems*, pages 570–576, 1998.

Pascal Massart and Élodie Nédélec. Risk bounds for statistical learning. *The Annals of Statistics*, 34:2326–2366, 2006.

Aditya Menon, Brendan van Rooyen, Cheng Soon Ong, and Robert C. Williamson. Learning from Corrupted Binary Labels via Class-Probability Estimation. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 125–134, 2015.

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep D Ravikumar, and Ambuj Tewari. Learning with Noisy Labels. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1196–1204, 2013.

Novi Quadrianto, Alex J Smola, Tibério S Caetano, and Quoc V Le. Estimating Labels from Label Proportions. *Journal of Machine Learning Research*, 10:2349–2374, December 2009.

Harish Ramaswamy, Clayton Scott, and Ambuj Tewari. Mixture proportion estimation via kernel embeddings of distributions. In *International Conference on Machine Learning*, pages 2052–2060, 2016.

Mark D. Reid and Robert C. Williamson. Surrogate regret bounds for proper losses. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 897–904, 2009a.

Mark D. Reid and Robert C. Williamson. Information, divergence and risk for binary experiments. *The Journal of Machine Learning Research*, 12:731–817, 2011.

MD Reid and RC Williamson. Generalised pinsker inequalities. *arXiv preprint arXiv:0906.1244*, 2009b. URL http://arxiv.org/abs/0906.1244.

Walter Rudin. *Functional Analysis*. McGraw-Hill, 1991.

Bernhard Schölkopf and Alexander J. Smola. *Learning with kernels*, volume 129. MIT Press, 2002.

Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer, 2008.

Erik Torgersen. *Comparison of Statistical Experiments*. Cambridge University Press, 1991.

Tim van Erven, Peter Grünwald, Mark D. Reid, and Robert C. Williamson. Mixability in statistical learning. In *Advances in Neural Information Processing Systems*, pages 1691–1699, 2012.

Brendan van Rooyen, Aditya K. Menon, and Bob Williamson. An average classification algorithm. http://www.mlunhinged.online/Content/AnAverageClassificationAlgorithm.pdf, 2017.

John von Neumann and Oskar Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1947.

Robert C. Williamson. Geometry of Losses. *Proceedings of the 27th Annual Conference on Learning Theory*, pages 1078–1108, 2014.

Yuhong Yang and Andrew Barron. Information-theoretic determination of minimax rates of convergence. *The Annals of Statistics*, 27(5):1564–1599, 1999.

Bin Yu. Assouad, Fano, and Le Cam. In *Festschrift for Lucien Le Cam*, pages 423–435. Springer, 1997.

Tong Zhang. Statistical analysis of some multi-category large margin classification methods. *The Journal of Machine Learning Research*, 5:1225–1251, 2004.

Tong Zhang. Information-theoretic upper and lower bounds for statistical estimation. *IEEE Transactions on Information Theory*, 52(4):1307–1321, 2006.

Yuchen Zhang, John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Information-theoretic lower bounds for distributed statistical estimation with communication constraints. In *Advances in Neural Information Processing Systems*, pages 2328–2336, 2013.

Journal of Machine Learning Research 18 (2018) 1-39 Submitted 8/2016; Revised 1/2018; Published 6/2018

Interactive Algorithms: Pool, Stream and Precognitive Stream

Sivan Sabato SABATOS@CS.BGU.AC.IL
Tom Hess TOMHE@POST.BGU.AC.IL
Department of Computer Science
Ben-Gurion University of the Negev
Beer Sheva 8410501, Israel.

Editor: Csaba Szepesvari

Abstract

We consider interactive algorithms in the pool-based setting, and in the stream-based setting. Interactive algorithms observe suggested elements (representing actions or queries), and interactively select some of them and receive responses. Pool-based algorithms can select elements at any order, while stream-based algorithms observe elements in sequence, and can only select elements immediately after observing them. We further consider an intermediate setting, which we term *precognitive stream*, in which the algorithm knows in advance the identity of all the elements in the sequence, but can select them only in the order of their appearance. For all settings, we assume that the suggested elements are generated independently from some source distribution, and ask what is the stream size required for emulating a pool algorithm with a given pool size, in the stream-based setting and in the precognitive stream setting. We provide algorithms and matching lower bounds for general pool algorithms, and for utility-based pool algorithms. We further derive nearly matching upper and lower bounds on the gap between the two settings for the special case of active learning for binary classification.

Keywords: interactive algorithms, active learning, pool-based, stream-based

1. Introduction

Interactive algorithms are algorithms which are presented with input in the form of suggested elements (representing actions or queries), and iteratively select elements, getting a response for each selected element. The reward of the algorithm, which is application-specific, is a function of the final set of selected elements along with their responses. Interactive algorithms are used in many application domains, including, for instance, active learning (McCallum and Nigam, 1998), interactive sensor placement (Golovin and Krause, 2011), summarization (Singla et al., 2016) and promotion in social networks (Guillory and Bilmes, 2010). As a specific motivating example, consider an application in which elements represent web users, and the algorithm should select up to q users to present with a free promotional item. For each selected user, the response is the observed behavior of the user after having received the promotion, such as the next link that the user clicked on. The final reward of the algorithm depends on the total amount of promotional impact it obtained, as measured by some function of the set of selected users and their observed responses. Note that the algorithm can use responses from previous selected users when deciding on the next user to select.

We consider two main interaction settings for interactive algorithms: The *pool-based* setting and the *stream-based* setting. In the pool-based setting, the entire set of suggested elements is provided in advance to the algorithm, which can then select any of the elements at any order. For instance, in the web promotion example, there might be a set of users who use the website for an extended period of time, and any of them can be approached with a promotion. In the stream-based setting, elements are presented to the algorithm in sequence, and the algorithm must decide immediately after observing an element, whether to select it or not. In the web promotion example, this is consistent with a setting where users access the website for single-page sessions, and so any promotion must be decided on immediately when the user is observed. Note that in the stream-based setting considered in this work, the only direct restriction is on the timing of selecting elements. We do not place restrictions on storage space or other resources.

The stream-based setting is in general weaker than the pool-based setting. Nonetheless, it is important and useful: In many real-life scenarios, it is not possible to postpone selection of elements, for instance due to storage and retrieval constraints, or because of timing constraints. This is especially pertinent when the data stream is real-time in nature, such as in streaming document classification (Bouguelia et al., 2013), in spam filtering (Chu et al., 2011), in web streams such as Twitter (Smailović et al., 2014), in video surveillance (Loy et al., 2012) and with active sensors (Krishnamurthy, 2002).

In this work, our goal is to study the relationship between these two important settings: the pool-based setting and the stream-based setting. Both of these settings have been widely studied in many contexts. In active learning, both the pool-based and the stream-based setting have been studied in classic works (Cohn et al., 1994; Lewis and Gale, 1994). Works that address mainly the stream-based setting include, for instance, Balcan et al. (2009); Hanneke (2011); Dasgupta (2012); Balcan and Long (2013); Sabato and Munos (2014). Some theoretical results hold for both the stream-based and the pool-based settings (e.g., Balcan and Long, 2013; Hanneke and Yang, 2015). Several near-optimal algorithms have been developed for the pool-based setting (Dasgupta, 2005; Golovin and Krause, 2011; Golovin et al., 2010b; Hanneke, 2007; Sabato et al., 2013; Gonen et al., 2013; Cuong et al., 2014). The pool-based setting is also heavily studied in various active learning applications (e.g., Tong and Koller, 2002; Tong and Chang, 2001; Mitra et al., 2004; Gosselin and Cord, 2008; Cebron and Berthold, 2009; Guo et al., 2013). General interactive algorithms have also been studied in both a pool-based setting (e.g., Golovin and Krause, 2011; Guillory and Bilmes, 2010; Deshpande et al., 2014) and in stream-based settings (e.g., Demaine et al., 2014; Arlotto et al., 2016; Streeter and Golovin, 2009; Golovin et al., 2010a).

To facilitate this study, we introduce a third setting, which we term the *precognitive stream-based* setting. This is an intermediate setting, which is weaker than the pool-based setting but stronger than the stream-based setting. In the precognitive setting, as in the standard stream-based setting, there is a sequence of elements and the algorithm must make a decision immediately after the item is presented to it. However, unlike the standard stream-based setting, in the precognitive setting the algorithm knows in advance the identity of items that will be presented in the future. This intermediate setting is of interest, since it has only one of the disadvantages of the standard stream-based setting: the requirement that elements are selected in order of their presentation in the sequence. On the other hand, it does not have the second disadvantage: the lack of knowledge of future possible items. Thus, studying this setting allows distinguishing the effect of the two issues on the performance of stream-based algorithms.

To study the relationship between the pool-based setting and the stream-based setting, as well as its precognitive variant, we assume that in all settings the suggested elements, along with their hidden responses, are drawn i.i.d. from some unknown source distribution. We then ask under what conditions, and at what cost, can a stream-based algorithm obtain the same output distribution as

a given black-box pool algorithm. Such an exact emulation is advantageous, as it allows direct application of methods and results developed for the pool-based setting, in the stream-based setting. Especially, if a pool-based algorithm succeeds in practice, but its analysis is unknown or limited, exact emulation guarantees that this success is reproduced in the streaming setting as well.

For discrete source distributions, any pool-based algorithm can be emulated in a stream-based setting, simply by waiting long enough, until the desired element shows up again. The challenge for stream-based interactive algorithms is thus to achieve the same output distribution as a pool-based algorithm, while observing as few suggested elements as possible. Clearly, there are many cases in which it is desirable to require less suggested elements, as this could save resources such as time, money, and communication. In active learning as well, while examples are usually assumed cheap, they are not usually completely free in all respects.

We study emulation of pool-based algorithms in two regimes. First, we consider the fully general case, of emulating some unrestricted pool algorithm. We provide a stream algorithm that can emulate any given black-box pool algorithm, and uses a uniformly bounded expected number of observed elements. The bound on the expected number of observed elements is exponential in the number of selected elements. We further prove a lower bound which indicates that this exponential dependence is necessary. We also study the precognitive stream-based setting and show the following: On the one hand, it can require a significantly smaller number of observed elements than the standard stream-based setting. On the other hand, its worst-case performance is also exponential in the number of selected elements, similarly to the stream-based setting. We conclude that while knowing the sequence in advance can be helpful, it does not improve the general performance of stream-based emulation.

Second, we consider *utility-based* interactive algorithm for the pool setting. We provide a stream algorithm that emulates such pool algorithms, using repeated careful applications of solutions of the well known "Secretary Problem" (Dynkin, 1963; Gilbert and Mosteller, 1966; Ferguson, 1989). The expected number of observed elements for this algorithm is only linear in the number of selected elements. In this case too we prove matching lower bounds. These results hold also for the precognitive stream setting. Our analysis shows a tradeoff between the number of observed elements and the number of selected elements, in cases where the stream algorithm is allowed to select extra elements over what the pool-algorithm selects.

Finally, we consider the special case of active learning for binary classification. We give nearly-matching upper and lower bounds for stream emulation in this setting. From the lower bound, we conclude that even in this well-studied setting, there are cases in which there exists a significant gap between the best pool-based algorithm and the best stream-based algorithm. This result generalizes a previous observation of Gonen et al. (2013) on the sub-optimality of CAL (Cohn et al., 1994), the classical stream-based active learning algorithm, compared to pool algorithms.

This paper is structured as follows: In Section 2 formal definitions and notations are provided. Section 3 discusses natural but suboptimal solutions. Section 4 considers emulating general pool algorithms, and Section 5 addresses the case of utility-based pool algorithms. In Section 6 we study active learning for binary classification. We conclude in Section 7.

2. Definitions

For a predicate p, denote by $\mathbb{I}[p]$ the indicator function which is 1 if p holds and zero otherwise. For an integer k, denote $[k] := \{1, \ldots, k\}$ and $[k]_0 = \{0, \ldots, k\}$. For a sequence S, $S(i)$ is the

i'th member of the sequence. Denote concatenation of sequences by $\circ$. Denote by Π_k the set of permutations over $[k]$.

For A, B which are both sequences, or one is a set and one a sequence, we use $A =_\pi B$ and $A \subseteq_\pi B$ to denote equality or inclusion on the unordered sets of elements in B and in A. We sometimes omit the π when no other interpretation is possible.

Let $\mathcal{X}$ be a measurable domain of elements, and let $\mathcal{Y}$ be a measurable domain of responses. A pool-based (or just pool) interactive algorithm $\mathcal{A}_p$ receives as input an integer $q \leq m$, and a pool of elements $(x_1, \ldots, x_m) \in \mathcal{X}^m$. We assume that for each x_i there is a response $y_i \in \mathcal{Y}$, which is initially hidden from $\mathcal{A}_p$. Denote $S = ((x_i, y_i))_{i \in [m]}$. We will assume throughout this work that S is drawn i.i.d. from a distribution over $\mathcal{X} \times \mathcal{Y}$. For a given S, S_X denotes the pool $(x_1, \ldots, x_m)$. At each round, $\mathcal{A}_p$ selects one of the elements i_t that have not been selected yet, and receives its response y_{i_t}. After q rounds, $\mathcal{A}_p$ terminates. Its output is the set $\{(x_{i_1}, y_{i_1}), \ldots, (x_{i_q}, y_{i_q})\}$. For a pool algorithm $\mathcal{A}_p$, denote by $\mathsf{sel}_p(S, t)$ the element that $\mathcal{A}_p$ selects at round t, if S is the pool it interacts with. $\mathsf{sel}_p(S, t)$, which can be random, can depend on S_X and on y_{i_k} for $k < t$. Denote by $\mathsf{sel}_p(S, [t])$ the sequence of elements selected by $\mathcal{A}_p$ in the first t rounds. $\mathsf{pairs}_p(S, t)$ and $\mathsf{pairs}_p(S, [t])$ similarly denote the selected elements along with their responses. The final output of $\mathcal{A}_p$ is the set of pairs in the sequence $\mathsf{pairs}_p(S, [q])$. We assume that $S \mapsto \mathsf{pairs}_p(S, [q])$ is measurable.

We assume for simplicity that the pool algorithm is permutation invariant. That is, for any $S, S' \subseteq (\mathcal{X} \times \mathcal{Y})^m$, if S' is a permutation of S then $\mathsf{sel}_p(S, [q]) = \mathsf{sel}_p(S', [q])$, or if $\mathcal{A}_p$ is randomized then the output distributions are the same. Since the pool S is drawn i.i.d. this does not lose generality.

A stream-based (or just stream) interactive algorithm $\mathcal{A}_s$ receives as input an integer q. We assume an infinite stream $S \subseteq (\mathcal{X} \times \mathcal{Y})^\infty$, where $S(t) = (x_t, y_t)$. We will assume that this stream is also an i.i.d. sample from a distribution over $\mathcal{X} \times \mathcal{Y}$. At iteration t, $\mathcal{A}_s$ observes x_t, and may select one of the following actions:

- Do nothing
- Select x_t and observe y_t
- Terminate.

At termination, the algorithm outputs a subset of size q of the set of pairs (x_t, y_t) it observed. Denote by $\mathsf{sel}_s(S, t)$ the t'th element that $\mathcal{A}_s$ selects and is also in the output set. Denote by $\mathsf{sel}_s(S, [t])$ the sequence of first t elements that $\mathcal{A}_s$ selects and are also in the output set. Use pairs_s to denote these elements along with their responses. The output of $\mathcal{A}_s$ when interacting with S is the set of the pairs in the sequence $\mathsf{pairs}_s(S, [q])$. We assume $S \mapsto \mathsf{pairs}_s(S, [q])$ is measurable. The total number of elements selected by $\mathcal{A}_s$ when interacting with S (including discarded elements) is denoted $N_{\mathsf{sel}}(\mathcal{A}_s, S, q)$. The number of iterations (observed elements) until $\mathcal{A}_s$ terminates is denoted $N_{\mathsf{iter}}(\mathcal{A}_s, S, q)$.

We would like to have stream algorithms that emulate pool algorithms, under the assumption that both the pool and the stream are drawn from the same distribution. We define an equivalence between a stream algorithm and a pool algorithm as follows.

Definition 1 *Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$ and let q be an integer. Let $S \sim \mathcal{D}^m$, $S' \sim \mathcal{D}^\infty$. A pool algorithm $\mathcal{A}_p$ and a stream algorithm $\mathcal{A}_s$ are* $(q, \mathcal{D})$-equivalent*, if the total-variation distance between the distributions of $\mathsf{pairs}_p(S, [q])$ and $\mathsf{pairs}_s(S', [q])$ is zero.*

In the standard stream setting defined above, the algorithm can only make decisions based on data observed in the past, that is its decision at iteration t can depend on $x_1, \ldots, x_t$, as well as the responses for the elements it selected until iteration $t-1$. Thus, the stream setting is harder than the pool setting in two separate aspects:

- It must make decisions without knowing what elements will show up after the current iteration; and
- It must select elements in the order of their appearance in the stream.

We introduce an additional setting, which we term *precognitive stream*. A precognitive stream algorithm follows the same protocol as the standard stream algorithm defined above, but we assume that it knows the elements in the entire stream, including those in future iterations. Formally, its decisions in iteration t for an input stream $S \subseteq (\mathcal{X} \times \mathcal{Y})^\infty$ may depend on the full (infinite) sequence elements S_X.

Denote by $\mathcal{D}_X$ the marginal of $\mathcal{D}$ on $\mathcal{X}$. For a given distribution $\mathcal{D}_X$ over $\mathcal{X}$, let $\textsc{ds}(\mathcal{D}_X)$ be the set of distributions over $\mathcal{X} \times \mathcal{Y}$ such that their marginal over $\mathcal{X}$ is equal to $\mathcal{D}_X$. Below, unless specified otherwise, we assume that the probability under $\mathcal{D}_X$ of observing any single $x \in \mathcal{X}$ is zero. This does not lose generality, since if this is not the case, $\mathcal{D}_X$ can be replaced by the distribution $\mathcal{D}_X \times \mathrm{Unif}[0,1]$, with the interactive algorithms ignoring the second element in the pair.

In some of the proofs below we use the following lemma, proved in Appendix A. This lemma captures a relationship between the expected number of observed Bernoulli trials and the probability of success in the last trial.

Lemma 2 *Let $\alpha \in (0,1), p \in (0, \alpha^4/2)$. Let $X_1, X_2, \ldots$ be independent Bernoulli random variables with $\mathbb{P}[X_i = 1] \leq p$. Let I be a random integer, which can be dependent on the entire sequence $X_1, X_2, \ldots$. Suppose that $\mathbb{P}[X_I = 1] \geq \alpha$. Then $\mathbb{E}[I] \geq \frac{\alpha^2}{4p}$.*

3. Simple Equivalent Stream Algorithms

Algorithm 1 Algorithm $\mathcal{A}_{\text{wait}}$

In the first m iterations, observe $x_1, \ldots, x_m$ and do nothing.
$S \leftarrow ((x_1, \star), \ldots, (x_m, \star))$
$j \leftarrow 1, t \leftarrow m+1$
repeat
  In iteration t, observe element x_t
  if $x_t = \mathsf{sel}_p(S, j)$ **then**
    Select x_t and observe y_t
    $S(i) \leftarrow (x_t, y_t)$
    $j \leftarrow j + 1$
  end if
  $t \leftarrow t + 1$
until $j = q + 1$
Return the set of all the pairs (x, y) in S with $y \neq \star$.

Emulating a pool algorithm in a streaming setting can be naively done using two extremely simple approaches, which we discuss below. However, each of these approaches is wasteful, in either the number of iterations it requires, or the number of selections it makes.

For the first approach, let $\mathcal{A}_p$ be a pool algorithm. For any discrete distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, and any q, it is easy to define a stream algorithm which is $(q, \mathcal{D})$-equivalent to $\mathcal{A}_p$. Let "$\star$" be some value not in $\mathcal{Y}$, and define $\mathcal{A}_{\text{wait}}$ as in Alg. 1. $\mathcal{A}_{\text{wait}}$ first draws a pool of size m, and then replicates the selections that a pool algorithm would have done for this pool, by each time waiting until the requested element is observed again in the stream. This stream algorithm is clearly $(q, \mathcal{D})$ equivalent to $\mathcal{A}_p$ for any discrete distribution $\mathcal{D}$, and it has $N_{\mathsf{sel}}(\mathcal{A}_{\text{wait}}, S', q) = q$ for all $S' \in (\mathcal{X} \times \mathcal{Y})^\infty$. However, $\mathbb{E}_{S' \sim \mathcal{D}^\infty}[N_{\mathsf{iter}}(\mathcal{A}_{\text{wait}}, S', q)]$ is not uniformly bounded, even for the class of discrete distributions. For instance, if the probability of observing any $x \in \mathcal{X}$ is some small p, then every selection that the algorithm makes would require observing $1/p$ elements in expectation.

In the second approach, one can simply select the first m elements observed by the stream, as done in the stream algorithm $\mathcal{A}_{\text{nowait}}$ defined in Alg. 2. This algorithm is also $(q, \mathcal{D})$ equivalent to $\mathcal{A}_p$, and it requires the smallest possible number of observed elements, since $N_{\mathsf{iter}}(\mathcal{A}_{\text{wait}}, S', q) = m$ for all $S' \in (\mathcal{X} \times \mathcal{Y})^\infty$, exactly the same as for the pool algorithm. However, in this case the number of selections is large, since $N_{\mathsf{sel}}(\mathcal{A}_{\text{nowait}}, S', q) = m > q$, regardless of q. This trivial algorithm is thus again unsatisfying.

Algorithm 2 Algorithm $\mathcal{A}_{\text{nowait}}$

input Pool size m, Black-box pool algorithm $\mathcal{A}_p$.
In each iteration $t \in [m]$, select x_t and observe y_t.
Return the pairs in $\mathsf{pairs}_p(S, q)$.

In the next section we consider the general pool emulation problem, and show that it is possible to have a uniform upper bound on the number of iterations, regardless of the distribution, and without selecting more than q elements. In addition, in Section 5, we show for utility-based pool algorithms, that there is a tradeoff between the number of additional selected elements and the expected number of observed elements. This tradeoff is evident also in the two simple algorithms shown above.

4. General Pool Algorithms

In this section we consider emulating general pool algorithms. In Section 4.1 we present a stream algorithm which can emulate any pool-based algorithm, using only black-box access to the pool algorithm. We prove a distribution-free upper bound on its expected number of iterations. In Section 4.2 we show that the number of iterations required by the proposed algorithm cannot be significantly improved. In Section 4.3, we study the precognitive stream setting and compare its guarantees to the standard stream setting.

4.1 Stream Emulation for General Pool Algorithms

The stream algorithm $\mathcal{A}_{\text{gen}}$, listed in Alg. 3, emulates any pool based algorithm $\mathcal{A}_p$ using only black-box access to $\mathcal{A}_p$. The algorithm emulates a general pool algorithm by making sure that in each iteration, its probability of selecting an element is identical to the conditional probability of the pool algorithm selecting the same element, conditioned on the history of elements and responses

selected and observed so far. This is achieved by repeatedly drawing the remaining part of the pool, and keeping it only if it is consistent with the elements that were already selected. The algorithm further uses the partial pool draw only if the element to be selected happens to have been observed last.

Algorithm 3 Algorithm $\mathcal{A}_{\text{gen}}$

input Original pool size m, budget $q < m$, black-box pool algorithm $\mathcal{A}_p$.

$S_0 \leftarrow ()$
for $i = 1 : q$ **do**
  repeat
    Draw $m - i + 1$ elements, denote them $\bar{x}_{i,i}, \ldots, \bar{x}_{i,m}$.
    $S'_i \leftarrow ((\bar{x}_{i,i}, \star), \ldots, (\bar{x}_{i,m}, \star))$.
  until $\mathsf{pairs}_p(S_{i-1} \circ S'_i, [i-1]) =_\pi S_{i-1}$ and $\mathsf{sel}_p(S_{i-1} \circ S'_i, i) = \bar{x}_{i,m}$.
  Select $\bar{x}_{i,m}$, get the response $\bar{y}_{i,m}$.
  $S_i \leftarrow S_{i-1} \circ ((\bar{x}_{i,m}, \bar{y}_{i,m}))$.
end for
Output S_q.

Below we show that $\mathcal{A}_{\text{gen}}$ improves over the two stream algorithms presented above, in that it selects exactly q elements, and has a uniform upper bound on the expected number of iterations, for any source distribution. First, we prove that $\mathcal{A}_{\text{gen}}$ indeed emulates any pool-based algorithm.

Theorem 3 *For any pool algorithm $\mathcal{A}_p$, any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, any integer m and $q \le m$, $\mathcal{A}_s := \mathcal{A}_{gen}(\mathcal{A}_p)$ is $(q, \mathcal{D})$-equivalent to $\mathcal{A}_p$.*

Proof For simplicity of presentation, we prove the result for discrete distributions. The proof for continuous distribution is analogous. Consider the probability space defined by the infinite sequence $S' \sim \mathcal{D}^\infty$ which generates the input to the stream algorithm, and an independent sequence $S \sim \mathcal{D}^m$ which is the input to the pool algorithm. For $z_1, \ldots, z_q \in \mathcal{X} \times \mathcal{Y}$, denote $Z_j = \{z_1, \ldots, z_j\}$. We have, for every $i \in [q]$,

$$\mathbb{P}[\mathsf{pairs}_p(S, [i]) =_\pi Z_i] = \sum_{j=1}^{i} \mathbb{P}[(\mathsf{pairs}_p(S, i) = z_j) \wedge (\mathsf{pairs}_p(S, [i-1]) =_\pi Z_i \setminus \{z_j\})]$$

$$= \sum_{j=1}^{i} \mathbb{P}[\mathsf{pairs}_p(S, i) = z_j \mid \mathsf{pairs}_p(S, [i-1]) =_\pi Z_i \setminus \{z_j\}] \cdot \mathbb{P}[\mathsf{pairs}_p(S, [i-1]) =_\pi Z_i \setminus \{z_j\}].$$

The same holds for $\mathsf{pairs}_s(S', \cdot)$. To show the equivalence it thus suffices to show that for all $z_1, \ldots, z_q \in \mathcal{X} \times \mathcal{Y}, i \in [q]$,

$$\mathbb{P}[\mathsf{pairs}_s(S', i) = z_i \mid \mathsf{pairs}_s(S', [i-1]) =_\pi Z_{i-1}] = \mathbb{P}[\mathsf{pairs}_p(S, i) = z_i \mid \mathsf{pairs}_p(S, [i-1]) =_\pi Z_{i-1}].$$

From the definition of $\mathcal{A}_s$ we have

$$\begin{aligned}
&\mathbb{P}[\mathsf{pairs}_s(S', i) = z_i \mid \mathsf{pairs}_s(S', [i-1]) =_\pi Z_{i-1}] \\
&= \mathbb{P}[\mathsf{pairs}_p(S_{i-1} \circ S'_i, i) = z_i \mid S_{i-1} =_\pi Z_{i-1} \wedge \mathsf{pairs}_p(S_{i-1} \circ S'_i, [i-1]) =_\pi Z_{i-1}] \\
&= \mathbb{P}[\mathsf{pairs}_p(S, i) = z_i \mid \mathsf{pairs}_p(S, [i-1]) =_\pi Z_{i-1}].
\end{aligned}$$

The last equality follows since $\mathcal{A}_p$ is permutation invariant and never selects the same index twice. This proves the equivalence. ■

The next theorem provides an upper bound on the expected number of elements observed by $\mathcal{A}_{\text{gen}}$. Unlike $\mathcal{A}_{\text{wait}}$, this upper bound holds uniformly for all source distributions.

Theorem 4 *For any pool algorithm $\mathcal{A}_p$, any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, any integer m and $q \le m$, if $\mathcal{A}_s := \mathcal{A}_{gen}(\mathcal{A}_p)$, $N_{\mathsf{sel}}(\mathcal{A}_s, S, q) = q$ for any $S \in (\mathcal{X} \times \mathcal{Y})^\infty$, and*

$$\mathbb{E}_{S \sim \mathcal{D}^\infty}[N_{\mathsf{iter}}(\mathcal{A}_s, S, q)] \le m^2 \left(\frac{em}{q-1}\right)^{q-1}.$$

Proof First, clearly $N_{\mathsf{sel}}(\mathcal{A}_s, S, q) = q$ for any $S \sim \mathcal{D}^\infty$. We now prove the upper bound on the expected number of iterations of $\mathcal{A}_s$. Let $S \sim \mathcal{D}^m$. For $i \ge 1$, $z_1, \ldots, z_{i-1} \in \mathcal{X}$, denote $Z_j = \{z_1, \ldots, z_j\}$, and let

$$p_i(z_1, \ldots, z_i) := \mathbb{P}[\mathsf{sel}_p(S, [i]) =_\pi Z_i \mid Z_i \subseteq_\pi S_X].$$

Suppose that $(S_{i-1})_X =_\pi Z_{i-1}$. The expected number of times that steps 3 to 6 are repeated for index i is the inverse of the probability that the condition in 6 holds. This condition, in our notation, is that $\mathsf{sel}_p(S_{i-1} \circ S'_i, [i-1]) =_\pi Z_{i-1}$ and $\mathsf{sel}_p(S_{i-1} \circ S'_i, i) = \bar{x}_{i,m}$. We have, from the permutation invariance of $\mathcal{A}_p$,

$$\mathbb{P}[\mathsf{sel}_p(S_{i-1} \circ S'_i, [i-1]) =_\pi Z_{i-1} \mid (S_{i-1})_X =_\pi Z_{i-1}] = p_{i-1}(z_1, \ldots, z_{i-1}).$$

In addition, for every draw of S'_i,

$$\mathbb{P}[\mathsf{sel}_p(S_{i-1} \circ S'_i, i) = \bar{x}_{i,m} \mid \mathsf{sel}_p(S_{i-1} \circ S'_i, [i-1]) =_\pi Z_{i-1} \wedge (S_{i-1})_X =_\pi Z_{i-1}] = \frac{1}{m-i+1}.$$

This is since under the conditional, one of the elements in S'_i must be selected by $\mathcal{A}_p$ in round i. Therefore, the probability that the condition in step 6 holds is $p_{i-1}(z_1, \ldots, z_{i-1})/(m-i+1)$. The expected number of times that steps 3 to 6 are repeated for index i is the inverse of that, and in each round $m-i+1$ elements are observed. Therefore the expected number of elements observed until selection i is made conditioned on $z_1, \ldots, z_{i-1}$ is $(m-i+1)^2/p_{i-1}(z_1, \ldots, z_{i-1})$. The unconditional expected number of elements observed until selection i is $(m-i+1)^2 \cdot \mathbb{E}[1/p_{i-1}(\mathsf{sel}_s(S', [i-1]))]$. For a set of indices J, denote $S|_J = \{S(j) \mid j \in J\}$. For simplicity of presentation we give the following derivation for discrete distributions, the proof for continuous distributions is analogous.

$$\begin{aligned}
\mathbb{E}[1/p_i(\mathsf{sel}_s(S', [i]))] &= \mathbb{E}[1/p_i(\mathsf{sel}_p(S, [i]))] \\
&= \sum_{\{z_1, \ldots, z_i\} \subseteq \mathcal{X} \times \mathcal{Y}} \mathbb{P}[\mathsf{sel}_p(S, [i]) =_\pi Z_i] \cdot \frac{1}{p_i(z_1, \ldots, z_i)} \\
&= \sum_{\{z_1, \ldots, z_i\} \subseteq \mathcal{X} \times \mathcal{Y}} \mathbb{P}[Z_i \subseteq_\pi S_X].
\end{aligned}$$

Hence

$$\begin{aligned}\mathbb{E}[1/p_i(\mathsf{sel}_s(S',[i])] &\leq \sum_{\{z_1,..,z_i\}\subseteq\mathcal{X}\times\mathcal{Y}} \sum_{J\subseteq[m],|J|=i} \mathbb{P}[(S|_J)_X = Z_i] \\ &= \sum_{J\subseteq[m],|J|=i} \sum_{\{z_1,..,z_i\}\subseteq\mathcal{X}\times\mathcal{Y}} \mathbb{P}[(S|_J)_X = Z_i] \\ &= \sum_{J\subseteq[m],|J|=i} 1 = \binom{m}{i}.\end{aligned}$$

It follows that the expected number of elements observed after the $i-1$'th selection and until selection i is at most $(m-i+1)^2\binom{m}{i-1}$. We conclude that

$$\mathbb{E}[N_{\mathsf{iter}}(\mathcal{A}_s,S,q)] \leq \sum_{i=0}^{q-1}(m-i)^2\binom{m}{i} \leq m^2\left(\frac{em}{q-1}\right)^{q-1}.$$

This completes the proof. ■

From the existence of $\mathcal{A}_{\text{gen}}$ we can conclude that the pool-based and the stream-based setting are essentially equivalent, up to the number of observed elements. However, the expected number of observed elements is exponential in q. In the next sections we show that this exponential dependence cannot be avoided when emulating general pool algorithms in a stream setting.

4.2 A Lower Bound for the Standard Stream Setting

The upper bound in Theorem 4 is exponential in q. We next show that this dependence cannot be eliminated in the general case for a standard stream algorithm. This result indicates that $\mathcal{A}_{\text{gen}}$ is close to optimal in terms of expected number of iterations. The lower bound holds for a standard stream algorithm, which does not know the identity of future elements in the stream. We consider precognitive stream algorithms in Section 4.3.

The lower bound is proved using a construction in which some of the elements represent base elements, and some represent permutations over base elements. The pool algorithm selects permutation elements that are consistent with the base elements it selected. Since the ranking of elements depends on the elements in the input pool, the same permutation can be consistent with different selections in different pools. The following lemma, which is used the proof of the lower bound, shows that nonetheless, once certain base elements have been selected, the set of permutations that are likely consistent with them is considerably limited. The lemma is also later used in a lower bound for the precognitive stream setting.

Lemma 5 *Let $t \leq t' \leq l$ be integers such that $l \geq 2t$. Let $Z = z_1,\ldots,z_t$ a set of values in $[0,1]$. Let X be l random values sampled i.i.d. from the uniform distribution on $[0,1]$. Denote $X = x_1,\ldots,x_l$ where $x_1 \leq x_2 \leq \ldots x_l$. Let $A_\sigma(Z,X) \subseteq \Pi_l$ be the set of permutations σ such that $\{x_{\sigma(1)},\ldots,x_{\sigma(t')}\} \supseteq \{z_1,\ldots,z_t\}$. For any $d \leq l-t$, there exists a set of permutations $\Phi(Z) \subseteq \Pi_l$, such that*

$$|\Phi(Z)|/|\Pi_l| \leq (4dt'/l)^t,$$

and

$$\mathbb{P}[A_\sigma(Z,X) \subseteq \Phi(Z) \mid Z \subseteq X] \geq 1 - 2t\exp(-2d^2/(l-t)),$$

Where the probability is over the randomness of X. Moreover, let $L(\sigma) := \cup_{Z:\sigma\in\Phi(Z)}\Phi(Z)$, then for all $\sigma \in \Pi_l$,

$$|L(\sigma)|/|\Pi_l| \leq (8dt'/l)^{2t}.$$

Proof Fix $d \leq l-t$. Denote the expected number of elements that are smaller than z_i in X, conditioned on $Z \subseteq X$, by $n_i := (l-t)z_i + \sum_{j=1}^t \mathbb{I}[z_j < z_i]$, and let

$$\Phi(Z) := \{\sigma \in \Pi_l \mid \exists f : [t] \to [t'] \text{ s.t. } f \text{ is one-to-one and } \forall i \in [t], |\sigma(f(i)) - n_i| < d\}. \quad (1)$$

These are the permutations such that the first t elements according to the permutation are mapped from elements with ranks in $[n_i - d, n_i + d]$. It is easy to see that

$$|\Phi(Z)|/|\Pi_l| \leq (t')^t \frac{(2d)^t}{\prod_{i=0}^{t-1}(l-i)} \leq \left(\frac{4dt'}{l}\right)^t,$$

where the last inequality follows since $l \geq 2t$. This proves the first part of the lemma. We now show the second part of the lemma. For $x \in X$, let $r(x)$ be its rank in X. By Hoeffding's inequality, for any $i \leq t$,

$$\mathbb{P}[|r(z_i) - n_i| \geq d \mid Z \subseteq X] \leq 2\exp(-2d^2/(l-t)).$$

Therefore,

$$\mathbb{P}[\forall i \leq t, |r(z_i) - n_i| < d \mid Z \subseteq X] \geq 1 - 2t\exp(-2d^2/(l-t)).$$

For any $\sigma \in A_\sigma$ we have that for each $i \leq t$, $z_i = x_{\sigma(j)}$ for some $j \leq t'$, that is $r(z_i) = \sigma(j)$. Moreover, different values of i are mapped to different values of j. Therefore there is some one-to-one function $f : [t] \to [t']$ such that for all $i \leq t$, $z_i = x_{\sigma(f(i))}$. Conditioned on the event $\forall i \leq t, |r(z_i) - n_i| < d$, it follows that $\forall i \leq t, |\sigma(f(i)) - n_i| < d$. Thus $\sigma \in \Phi(Z)$, which proves the second claim of the lemma.

To see the last claim, observe that if $\sigma, \sigma' \in \phi(Z)$ for some Z, then there are functions $f, g : [t] \to [t']$ such that for $i \in [t]$, $|\sigma(f(i)) - \sigma'(g(i))| < 2d$. Therefore,

$$L(\sigma) \subseteq \{\sigma' \in \Pi_l \mid \exists f, g : [t] \to [t'], \forall i \in [t], |\sigma(f(i)) - \sigma'(g(i))| < 2d\}.$$

The bound on the size of $L(\sigma)$ in the claim directly follows, similarly to the bound on $|\Phi(Z)|$. ■

The lower bound for the standard stream setting is provided below. It shows that for some pool algorithm, any equivalent stream algorithm has an expected number of observed elements which is exponential in q. The lower bound is worst-case over all source distributions $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, where the pool and the stream, as above, are drawn i.i.d. from the distribution. The proof involves constructing a pool-based algorithm in which the last selected element is significantly constrained by the identity of the previously selected elements, using the permutation construction. This type of constraint is not an issue in a pool setting, since the algorithm has advance knowledge of all the available elements. In a streaming setting, however, this requires a possibly long wait to obtain the matching last element. Because the stream algorithm is allowed to select elements in a different order than the pool algorithm, additional care is taken to make sure that in this construction, it is not possible to circumvent the problem this way.

Theorem 6 *There is an integer q_0, such that for $q \geq q_0$ and $m \geq 16q^2\log(4q)+1$, there exist a pool algorithm $\mathcal{A}_p$ and a marginal $\mathcal{D}_X$, such that any stream algorithm $\mathcal{A}_s$ which is $(q,\mathcal{D})$ equivalent to $\mathcal{A}_p$ for all $\mathcal{D} \in \text{DS}(\mathcal{D}_X)$, and selects only q elements, has*

$$\exists \mathcal{D} \in \text{DS}(\mathcal{D}_X), \mathbb{E}_{S\sim\mathcal{D}^\infty}[N_{\mathsf{iter}}(\mathcal{A}_s,S,q)] \geq \frac{1}{1{,}000}\left(\frac{m-1}{8q^2\log(4q)}\right)^{\frac{q-1}{2}}.$$

Proof Let the domain of elements be $\mathcal{X} = [0,2)$ and assume responses in $\mathcal{Y} = \{0,1\}$. Denote $\texttt{base} := [0,1]$ and $\texttt{perm} := (1,2)$. Call a pool S_X in which exactly one element in the pool is in $\texttt{perm}$ and the rest are in $\texttt{base}$ a "good pool".

We now define a pool algorithm as follows. On bad pools, $\mathcal{A}_p$ always selects only elements in $\texttt{base}$ or only elements in $\texttt{perm}$. In a good pool, denote for simplicity the single element from $\texttt{perm}$ by z_m, and the elements from $\texttt{base}$ by $z_1,\ldots,z_{m-1}$, where $z_{i-1} < z_i$ for $i \in [m-1]$. Define a mapping $\psi : \texttt{perm} \to \Pi_{m-1}$, such that if z_m is uniform over $\texttt{perm}$, then for $\psi(z_m)$ all permutations in the range are equally likely. Let $\sigma = \psi(z_m)$. On a good pool, $\mathcal{A}_p$ behaves as follows: The first $q-1$ elements that $\mathcal{A}_p$ selects are $z_{1+\sigma(1)},\ldots,z_{1+\sigma(q-1)}$. The last element that it selects is z_m if the response for all previous elements was 0, and z_1 otherwise.

Define the marginal $\mathcal{D}_X$ over $\mathcal{X}$ such that for $X \sim \mathcal{D}_X$, $\mathbb{P}[X \in \texttt{base}] = 1-1/m$, $\mathbb{P}[X \in \texttt{perm}] = 1/m$, and in each range $\texttt{base},\texttt{perm}$, X is uniform. The probability of a good pool under $\mathcal{D} \in \text{DS}(\mathcal{D}_X)$ is $(1-1/m)^{m-1} \geq 1/e^2 =: c_1$. We now show a lower bound on the expected number of iterations of a stream algorithm which is $(q,\mathcal{D})$-equivalent to any $\mathcal{D} \in \text{DS}(\mathcal{D}_X)$. Let $\mathcal{D}_0$ be the distribution over $\mathcal{X}\times\mathcal{Y}$ such that for $(X,Y) \sim \mathcal{D}_0$, $X \sim \mathcal{D}_X$ and $Y = 0$ with probability 1. Let $S \sim \mathcal{D}_0^m$ be the input to $\mathcal{A}_p$.

We apply Lemma 5 with $t = t' = q-1, l = m-1, d = \sqrt{(m-q)\log(4q)/2}$. By this lemma, for any set $Z = \{z_1,\ldots,z_{q-1}\} \subseteq \texttt{base}$, there exists a set of permutations $\Phi(Z) \subseteq \Pi_{m-1}$ such that

$$|\Phi(Z)|/|\Pi_{m-1}| \leq \left(\frac{8q^2(m-q)\log(4q)}{(m-1)^2}\right)^{(q-1)/2} \leq \left(\frac{8q^2\log(4q)}{m-1}\right)^{\frac{q-1}{2}}, \tag{2}$$

and also

$$\mathbb{P}[\psi(z_m) \in \Phi(Z) \mid \mathsf{sel}_p(S,[q-1]) =_\pi Z \wedge S \text{ is good}] \geq 1 - 2(q-1)\exp(-\log(4q)) \geq \frac{1}{2}. \tag{3}$$

The theorem follows from the following two claims:

1. When $\mathcal{A}_s$ emulates a good pool, it selects an element from $\texttt{perm}$ only after selecting $q-1$ elements from $\texttt{base}$.
2. Therefore, when $\mathcal{A}_s$ emulates a good pool, the expected number of observed elements until selecting the last element is lower bounded, and so the overall expected number is lower bounded.

We first prove claim 1. Consider a stream algorithm which is $(q,\mathcal{D})$-equivalent to $\mathcal{A}_p$ for any $\mathcal{D} \in \text{DS}(\mathcal{D}_X)$. Consider runs of $\mathcal{A}_s$ with input $S' \sim \mathcal{D}_0^\infty$. Denote by E_g the event that the output of $\mathcal{A}_s$ is equal to a possible output of $\mathcal{A}_p$ on a good pool with $S \sim \mathcal{D}_0^m$. Then $\mathbb{P}[E_g] \geq c_1$. Claim 1 is that

$$\mathbb{P}[\mathsf{sel}_s(S',[q-1]) \subseteq_\pi \texttt{base} \mid E_g] = 1. \tag{4}$$

In other words, when simulating a good pool, the elements in $\texttt{base}$ are all selected before the element in $\texttt{perm}$.

To show claim 1, note that by the definition of $\mathcal{A}_p$, for any source distribution over $\mathcal{X}\times\mathcal{Y}$, if $\mathcal{A}_p$ outputs a set with elements both in $\texttt{base}$ and in $\texttt{perm}$, then there is exactly one element in $\texttt{perm}$ in the output, and all the responses in the output for elements in $\texttt{base}$ are 0 with probability 1.

Now, suppose that $\mathbb{P}[\mathsf{sel}_s(S',[q-1]) \subseteq_\pi \texttt{base} \mid E_g] < 1$. Then $\mathbb{P}[\mathsf{sel}_s(S',q) \in \texttt{base} \mid E_g] > 0$, since there can be only one element in $\texttt{perm}$ in the output of a good pool. But, consider running $\mathcal{A}_s$ with a source distribution $\mathcal{D}' \in \text{DS}(\mathcal{D}_X)$ such that for $(X,Y) \sim \mathcal{D}'$, $X \sim \mathcal{D}_X$ and $\mathbb{P}_{\mathcal{D}'}[Y = 0|X = x] = \frac{1}{2}$ for all x. There is a positive probability that in the first $q-1$ selected elements all the responses are 0, just as for $\mathcal{D}_0$. Therefore, also for $S'' \sim \mathcal{D}'^\infty$, $\mathbb{P}[\mathsf{sel}_s(S'',q) \in \texttt{base} \mid E_g] > 0$. But then there is a positive probability that the response for the last element, which is in $\texttt{base}$, is 1, contradicting the $(q,\mathcal{D}')$-equivalence of the pool and $\mathcal{A}_s$. This proves claim 1.

We now show claim 2, which completes the proof. From claim 1 in Eq. (4), we conclude that $\mathbb{P}[\mathsf{sel}_s(S',q) \in \texttt{perm} \mid E_g] = 1$. Therefore, from Eq. (3), for any $Z \subseteq \texttt{base}$ with $|Z| = q-1$,

$$\mathbb{P}[\psi(\mathsf{sel}_s(S',q)) \in \Phi(\mathsf{sel}_s(S',[q-1])) \mid E_g] \geq 1/2.$$

Therefore

$$\mathbb{P}[\psi(\mathsf{sel}_s(S',q)) \in \Phi(\mathsf{sel}_s(S',[q-1]))] \geq \mathbb{P}[E_g]/2 \geq c_1/2.$$

Now, let $X_i \sim \mathcal{D}_X$ be the i'th element observed after selecting the first $q-1$ elements, and let $B_i = \mathbb{I}[\psi(X_i) \in \Phi(Z)]$, where Z is the set of $q-1$ selected elements. B_i are independent Bernoulli random variables, each with a probability of success at most p, where by Eq. (2)

$$p \leq \frac{|\Phi(Z)|}{|\Pi_{m-1}|} \leq \left(\frac{8q^2\log(4q)}{m-1}\right)^{\frac{q-1}{2}}.$$

Let I be the number of elements that $\mathcal{A}_s$ observes after selecting Z, until selecting element q. We have $\mathbb{P}[B_I = 1] \geq c_1/2$. From the assumption in the theorem statement that $m \geq 16q^2\log(4q)+1$, we have that for sufficiently large q, $p \leq c_1^4/32$. By Lemma 2, it follows that $\mathbb{E}[I] \geq c_1^2/(16p) \geq (1000p)^{-1}$. Hence

$$\mathbb{E}[I] \geq \frac{1}{1000}\left(\frac{m-1}{8q^2\log(4q)}\right)^{\frac{q-1}{2}}.$$

Since $\mathbb{E}[N_{\mathsf{iter}}(\mathcal{A},S,q)] \geq \mathbb{E}[I]$, this completes claim 2 and finalizes the proof. ■

The lower bound, as well as the upper bound in Theorem 4, both show an exponential dependence on q. However, the exponent in the lower bound is about half that of the upper bound. Some of this gap might be an artifact of the fact that in the lower bound, only a fixed marginal distribution is considered. Closing this gap remains an open problem, which we leave for future work.

4.3 Emulation with a Precognitive Stream Algorithm

We next consider the precognitive setting. In the precognitive setting, the stream algorithm has the advantage that it can plan ahead, since it knows in advance the identity of elements that will be available for selection from the entire stream. This breaks the construction used in the lower bound of Theorem 6, thus showing that in some cases there is a significant gap between standard stream algorithms and precognitive stream algorithms. We prove this in the following theorem.

Theorem 7 *There is an integer q_0, such that for $q \geq q_0$ and $m \geq 16q^2\log(4q)+1$, there exist a pool algorithm $\mathcal{A}_p$ and a marginal $\mathcal{D}_X$, such that any stream algorithm $\mathcal{A}_s$ which is $(q, \mathcal{D})$ equivalent to $\mathcal{A}_p$ for all $\mathcal{D} \in \mathrm{DS}(\mathcal{D}_X)$, and selects only q elements, requires an expected number of iterations of $\Omega\left(\left(\frac{m}{q^2\log(q)}\right)^{q/2}\right)$, while there exists a precognitive stream algorithm that satisfies the same equivalence, and requires only m^2 iterations in expectation.*

Proof Consider the same construction as in the proof of Theorem 6: the same $\mathcal{D}_X$, and the same behavior of the pool algorithm on "good pools", defined as in that proof. Assume further, consistently with the definition of $\mathcal{A}_p$ in that proof, that on bad pools, the pool algorithm checks if there are more `base` element or more `perm` elements in the pool, and then selects the first q elements in the pool that belong to the set with the larger number (in case of a tie, `base` is selected). Denote `base`, `perm`, ψ as in the proof of Theorem 6.

From Theorem 6 we have that the expected number of iterations of a standard stream algorithm on this problem is $\Omega\left(\left(\frac{m}{q^2\log(q)}\right)^{q/2}\right)$.

Now, consider the following precognitive stream algorithm: The algorithm checks, before the start of the iterations, whether the first m stream elements constitute a good pool. If they do not, it checks whether there are more `base` or more `perm` elements in the first m elements. Then it selects the first q elements in the stream that belong to the majority set (tie-breaking with preference to `base` as in the pool algorithm). Thus, for bad pools, the expected number of iterations by the precognitive stream algorithm is at most m.

Now, if the first m elements in the stream constitute a good pool, then the precognitive stream algorithm finds an integer t such that x_t is the smallest element from `base` among the previous $m-1$ `base` elements, using the following procedure:

1. Let $t_0 \leftarrow 1$; $t_1 \leftarrow m$.
2. Let A be the set of $m-1$ elements from `base` in $x_{t_0}, \ldots, x_{t_1}$.
3. If x_{t_1} is the minimum of A, set $t \leftarrow t_1$ and terminate.
4. Otherwise, set $t_0 \leftarrow t_1 + 1$, and $t_1 \leftarrow$ the smallest integer such that $x_{t_0}, \ldots, x_{t_1}$ has exactly $m-1$ elements from `base`. Go to 2

After setting t, the algorithm identifies the earliest iteration $t' > t$ such that $x_{t'} \in$ `perm`. The algorithm then ranks the $m-1$ elements from `base` that immediately precede x_t, denoting them $z_2 < \ldots < z_{m-1}$, and denotes $z_1 := x_t$.

The algorithm then selects $z_{1+\sigma(1)}, \ldots, z_{1+\sigma(q-1)}$ in the order of their appearance in the stream, where $\sigma := \psi(x_{t'})$. Then, if all of the responses to these elements were 0, it selects $x_{t'}$. Otherwise, it selects z_1.

It is easy to see that this algorithm is equivalent to the pool algorithm defined above. Also, the expected number of iterations of this algorithm is

$$\mathbb{E}[t'] = \mathbb{E}[t' - t] + \mathbb{E}[t].$$

Now, $\mathbb{E}[t'-t]$ is the expected wait time until observing an element from `perm`. Since for $X \sim \mathcal{D}_X$, $\mathbb{P}[X \in \texttt{perm}] = 1/m$, we have $\mathbb{E}[t'-t] = m$. In addition, by Wald's identity, $\mathbb{E}[t]$ is equal to the expected number of rounds of the procedure for setting t above, which is $m-1$, times the expected size

of $t_1 - t_0 + 1$ in each round, which is $\mathbb{P}[X \in \texttt{base}] \cdot (m-1)$. Thus, $\mathbb{E}[t] = (m-1)^3/m \leq (m-1)m$. It follows that the expected number of iterations of the precognitive stream algorithm is $\mathbb{E}[t'] \leq m^2$. ■

This result shows that the precognitive setting has a significant advantage in some cases. Nonetheless, we show a lower bound, indicating that its worst-case performance cannot be much better than the worst-case performance of a standard stream algorithm. The proof of the lower bound shares some techniques with the proof of Theorem 6, especially the use of permutations. It additionally relies on the observation that while the stream algorithm knows in advance what elements will be available for selection and when, it is still constrained to selecting the elements in the order that they are provided by the stream. It also must keep to a certain order of selections to emulate the pool algorithm, since the pool algorithm makes decisions based on previously observed responses. Thus, in the proof of the lower bound, we construct a pool algorithm that might select two permutation elements, but the order in which they would be selected depends on the responses to selected base elements. As a result, the precognitive algorithm cannot tell in advance in which order the two elements must be selected. The probability of both orders appearing early enough in the stream is low, so that even when the whole stream is taken into account, an exponential dependence cannot be avoided, though this dependence is weaker than the one shown in Theorem 6 for standard stream algorithms.

The proof of the lower bound employs the following lemma, which states that after partially observing two sequences of fair coin tosses, there is a positive probability that both sequences share the majority outcome, as well as a positive probability that their majority outcome is different. In the lower bound below, this lemma is used to show that the precognitive algorithm must commit to a small set of permutations before finding out in which order they must be selected.

Lemma 8 *Consider n i.i.d. tosses of two fair coins, where n is an odd number. Denote the tosses of the first coin by $A_1, \ldots, A_n \in \{0,1\}$, and the second by $B_1, \ldots, B_n \in \{0,1\}$. Let $K \in \{n/2, \ldots, n\}$ be a stopping time for the sequence $B_1, \ldots, B_n$, which can depend also on $\bar{A} := (A_1, \ldots, A_{n/2})$. Denote $\bar{B}(K) := (B_1, \ldots, B_K)$.*

There exists some integer n_0 and constants $c = 0.68$, $c' = 0.15$ such that for all odd $n > n_0$, with a probability at least c over the values of $\bar{A}, K, \bar{B}(K)$,

$$\mathbb{P}\left[\mathbb{I}[\sum_{i=1}^{n} A_i > n/2] = \mathbb{I}[\sum_{i=1}^{n} B_i > n/2] \;\middle|\; \bar{A}, K, \bar{B}(K)\right] \in [c', 1-c']$$

Proof Denote for brevity $A_j^l := \sum_{i=j}^{l} A_i$ and similarly for B_j^l. First note that

$$\begin{aligned}&\mathbb{P}\left[\mathbb{I}[A_1^n > n/2] = \mathbb{I}[B_1^n > n/2] \;\middle|\; K, \bar{A}, \bar{B}(K)\right] \\ &= \mathbb{P}[B_1^n > n/2 \mid K, \bar{B}(K)] \cdot \mathbb{P}[A_1^n > n/2 \mid \bar{A}] + \mathbb{P}[B_1^n \leq n/2 \mid K, \bar{B}(K)] \cdot \mathbb{P}[A_1^n \leq n/2 \mid \bar{A}]\end{aligned}$$

Thus it suffices to prove that with a probability at least c over the values of $\bar{A}$, $\mathbb{P}[A_1^n > n/2 \mid \bar{A}] \in [c', 1-c']$, which implies the same for $\mathbb{P}[A_1^n \leq n/2 \mid \bar{A}]$, leading to the claim of the lemma.

We first consider the limit case $n \to \infty$. In this case $A_1^{n/2} \sim N(n/4, \sqrt{n/8})$. Thus, with a probability of $\mathrm{erf}(1/\sqrt{2}) > 0.68 =: c$,

$$A_1^{n/2} \in [n/4 - \sqrt{n/8}, n/4 + \sqrt{n/8}].$$

If this is the case, then

$$A^n_{n/2+1} < n/4 - \sqrt{n/8} \quad \Longrightarrow \quad A^n_1 < n/2, \text{ and}$$
$$A^n_{n/2+1} > n/4 + \sqrt{n/8} \quad \Longrightarrow \quad A^n_1 > n/2.$$

Since $A^n_{n/2+1} \sim N(n/4, \sqrt{n/8})$, each of the above events occurs with a probability at least $\frac{1}{2} - \mathrm{erf}(1/\sqrt{2})/2 > 0.15 =: c'$. Thus, with a probability at least c over the values of $\bar{A}$,

$$\mathbb{P}[A^n_1 > n/2 \mid \bar{A}], \mathbb{P}[A^n_1 \leq n/2 \mid \bar{A}] \in [c', 1-c'].$$

The same holds for any large enough finite value of n. The statement of the lemma directly follows. ■

The lower bound for the precognitive setting is given below. It is exponential in q, like the lower bound for the standard stream setting in Theorem 6, however the dependence is slightly weaker.

Theorem 9 *There is an integer q_0 such that for any $q \geq q_0$ and $m \geq 2q+1$ there exist a pool algorithm $\mathcal{A}_p$ and a distribution $\mathcal{D}$, such that any precognitive stream algorithm $\mathcal{A}_s$ which is $(q, \mathcal{D})$ equivalent to $\mathcal{A}_p$ and selects only q elements, has*

$$\mathbb{E}_{S\sim\mathcal{D}^\infty}[N_{\mathsf{iter}}(\mathcal{A}_s, S, q)] \geq \frac{1}{600}\left(\frac{m}{8q^2\log(2q)}\right)^{q/8}.$$

Proof Assume for simplicity that 4 divides q and m. Let the domain of elements be $\mathcal{X} = [0,4) \cup \{\star\}$ and assume responses in $\mathcal{Y} = \{0,1\}$. Denote $\texttt{base1} := [0,1]$, $\texttt{base2} := [2,3]$, $\texttt{perm1} := (1,2)$ and $\texttt{perm2} := (3,4)$. Denote $\texttt{base} = \texttt{base1} \cup \texttt{base2}$ and $\texttt{perm} = \texttt{perm1} \cup \texttt{perm2}$.

We define the pool algorithm $\mathcal{A}_p$ as follows. Call a pool S_X a "good pool" if it includes at least $m/4$ elements from each of $\texttt{base1}$ and $\texttt{base2}$, and at least one element from each of $\texttt{perm1}, \texttt{perm2}, \star$. In a good pool S_X, the pool algorithm $\mathcal{A}_p$ behaves as follows. For $i \in \{1,2\}$, let $X_i \subseteq S_X \cap \texttt{base}(i)$ be a subset of size $m/4$, selected uniformly at random from $S_X \cap \texttt{base}(i)$. Let $x_{p(i)}$ be a random element from $S_X \cap \texttt{perm}(i)$. Let $\sigma_1 = \psi(x_{p(1)} - 1)$, $\sigma_2 = \psi(x_{p(2)} - 3)$. For $j \in [m/4]$, let x^i_j be the j-th largest value in X_i. Define a mapping $\psi : (0,1) \to \Pi_{m/4}$, such that if Z is uniform over $(0,1)$, then for $\psi(Z)$ all permutations in the range are equally likely.

The first $q-2$ elements that $\mathcal{A}_p$ selects are $\bar{X}_i := x^i_{\sigma_i(1)}, \ldots, x^i_{\sigma_i(q/2-1)}$, for each of $i \in \{1,2\}$. The last 2 elements that $\mathcal{A}_p$ selects are determined as follows: Let $r_i \in \{0,1\}$ be the value of the majority of responses received for the elements in $\bar{X}_i$. Denote $s = 1 + \mathbb{I}[r_1 = r_2]$ and $\bar{s} = 1 + \mathbb{I}[r_1 \neq r_2]$. The second-to-last element that $\mathcal{A}_p$ selects is $x_{p(s)}$. The last element that $\mathcal{A}_p$ selects is $x_{p(\bar{s})}$ if the response for $x_{p(s)}$ is 0, and $\star$ otherwise. See illustration in Figure 1.

Define the distribution $\mathcal{D}$ such that its marginal $\mathcal{D}_X$ over $\mathcal{X}$ satisfies, for $X \sim \mathcal{D}_X$, $\mathbb{P}[X \in \texttt{base1}] = \mathbb{P}[X \in \texttt{base2}] = 1/3$, $\mathbb{P}[X \in \texttt{perm1}] = \mathbb{P}[X \in \texttt{perm2}] = \mathbb{P}[X = \star] = 1/9$, and in each range $\texttt{base1}, \texttt{base2}, \texttt{perm1}, \texttt{perm2}$, X is conditionally uniform. In addition, for $(X,Y) \sim \mathcal{D}$, $\mathbb{P}[Y = 0 | X = x] = \frac{1}{2}$ for all $x \in \mathcal{X}$. Assume a large enough m such that the probability of a good pool under $\mathcal{D}_X$ is at least $1-\epsilon$ for some very small $\epsilon > 0$. We set $\epsilon = 0$ below for simplicity, noting that any small enough choice will lead to the same lower bound due to rounding down of constants. Thus we assume all pools are good pools.

Consider running the $(q, \mathcal{D})$-equivalent algorithm $\mathcal{A}_s$ with the source distribution $\mathcal{D}$. We derive a lower bound on $\mathbb{E}_{S\sim\mathcal{D}^\infty}[N_{\mathsf{iter}}(\mathcal{A}_s, S, q)]$ via the following sequence of arguments.

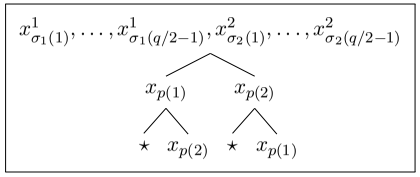


Figure 1: A tree describing the possible selections made by the pool algorithm on a good pool in the proof of Theorem 9.

1. We show that $\mathcal{A}_s$ selects at least $q/4$ elements from each of $\texttt{base1}$ and $\texttt{base2}$ before it selects any element from $\texttt{perm}$.

2. Using the previous claim, we show that there is a positive probability over runs of $\mathcal{A}_s$ on $S \sim D^\infty$ that there is an iteration in the run such that the following hold:

   - Before this iteration at least $q/4$ elements from each of $\texttt{base1}$ and $\texttt{base2}$ have been selected.
   - Conditioned on the run until this iteration, and on the entire S_X, the probability that at some time after this iteration an element from $\texttt{perm1}$ is selected, and an element from $\texttt{perm2}$ is selected at some time after that, is larger than a constant.
   - The same property holds for the reverse order: selecting first an element from $\texttt{perm2}$ and then an element from $\texttt{perm1}$.

3. We conclude that the expected number of iterations of $\mathcal{A}_s$ depends on the probability to have, in a single stream S_X, two elements that are both mapped to permutations in a single small subset. We then show that this implies a lower bound on the number of iterations of $\mathcal{A}_s$.

We start by showing the claim in item 1. Consider a run of $\mathcal{A}_s$ with input $S \sim \mathcal{D}^\infty$. Since the output must be equivalent to an output of $\mathcal{A}_p$ on a good pool, this implies that for $i \in \{1,2\}$, an element from $\texttt{perm}(i)$ is selected if and only if $s = i$ (where s is defined as done above for the pool algorithm) or the response for an element selected from $\texttt{perm}(\bar{s})$ is 0. Denote the iteration in which an element from $\texttt{perm}$ is first selected by I. Suppose for contradiction that before iteration I, less than $q/4$ elements from either $\texttt{base1}$ or $\texttt{base2}$ have been selected. Thus, for at least one of $\texttt{base1}, \texttt{base2}$, less than half of the elements to be selected from this set during this run are selected until iteration I. It follows that at iteration I the majority of the responses for at least one of the sets $\bar{X}_i$ is yet unknown. Therefore the value of s is also unknown and there is a positive probability that it will be each of $1,2$. There is also a positive probability that, if an additional element from $\texttt{perm}$ is selected later, its response will be 1. Thus, there is a positive probability that the output of this run does not correspond to any output of $\mathcal{A}_p$, contradicting the equivalence of $\mathcal{A}_s$ and $\mathcal{A}_p$. This proves the claim in item 1.

We now prove the claim in item 2. Let T be the smallest integer such that until iteration T (non-inclusive), at least $q/4$ elements from each of $\texttt{base1}$ and $\texttt{base2}$ have been selected, and no

element from $\texttt{perm}$ has been selected yet. T exists by item 1 above. An element from $\texttt{perm}(s)$, where s depends on the majority of responses in each of $\bar{X}_i$, must be selected before an element from $\texttt{perm}(\bar{s})$ is selected (if at all). Consider the distribution of s, conditioned on the run of $\mathcal{A}_s$ until iteration T and on S_X. We have $s = 2$ if the majority of responses on $\bar{X}_1$ is equal to the majority of responses on $\bar{X}_2$. Out of these responses, at least $q/4 - 1$ of at least one of $\bar{X}_1, \bar{X}_2$, are yet unknown at iteration T. Thus, by Lemma 8, for a large enough q we have that for any S_X, there is a probability of at least $c = 0.68$ over the responses until iteration T that

$$\mathbb{P}[s = 2 \mid \text{the responses until iteration } T, S_X] \in [c', 1 - c'],$$

where $c' = 0.15$. The same holds for $s = 1$.

To finalize the proof of the claim in item 2, observe that if the response for the element selected from $\texttt{perm}(s)$ is 0, then an element from $\texttt{perm}(\bar{s})$ will also be selected later. This occurs with a probability of $\frac{1}{2}$. Consider the event $B_1 :=$ "an element from $\texttt{perm}(1)$ is selected at some point in the run and an element from $\texttt{perm}(2)$ is selected some time later". Let B_2 be the symmetric event. We conclude that with a probability of at least c,

$$\forall i \in \{1, 2\}, \quad \mathbb{P}[B_i \mid \text{the responses until iteration } T, S_X] \geq c'/2. \tag{5}$$

This proves the claim in item 2. Denote the event that Eq. (5) holds by E_p.

We now turn to the third and last part of the proof. Denote by Z_i, for $i \in \{1, 2\}$, the first $q/4$ elements selected by $\mathcal{A}_s$ from $\texttt{base}(i)$. We apply Lemma 5 with $t = q/4, t' = q/2 - 1, l = m/4, d = \sqrt{m \log(q/c)/8}$, to obtain that if the pool algorithm has $Z_i \subseteq \bar{X}_i$ for $i \in \{1, 2\}$, then there is a probability of at least $1 - \frac{q}{2} \exp(-8d^2/(m-q)) \geq 1 - c/2$ that the element that $\mathcal{A}_p$ selects from $\texttt{perm}(i)$ for $i \in \{1, 2\}$ (if it exists) is in a given set of permutations $\Phi(Z_i)$ that satisfies[1]

$$\frac{|\Phi(Z_i)|}{|\Pi_{m/4}|} \leq \left(\frac{8dq}{m}\right)^{q/4}.$$

Thus, this holds also for $\mathcal{A}_s$. Since this holds with probability $1 - c/2$, and Eq. (5) holds with a probability at least c, there is a probability at least $c/2$ that both events hold simultaneously. That is, denoting the event $G_1 :=$ "an element from $\phi(Z_1)$ is selected at some point in the run and an element from $\phi(Z_2)$ is selected some time later" and symmetrically for G_2, we have that with a probability at least $c/2$ over the responses until iteration T and S_X,

$$\forall i \in \{1, 2\}, \quad \mathbb{P}[G_i \mid \text{the responses until iteration } T, S_X] \geq c'/2.$$

Denote the event that this holds E_G. Abbreviate $N_{\mathsf{iter}} := N_{\mathsf{iter}}(\mathcal{A}_s, S, q)$. It follows that

$$\begin{aligned}\mathbb{E}[N_{\mathsf{iter}} \mid S_X, E_G] &\geq c'/2 \cdot \left(\mathbb{E}_{Z_1,Z_2}\Big[\mathbb{E}[N_{\mathsf{iter}} \mid G_1, E_G, S_X, Z_1, Z_2] + \mathbb{E}[N_{\mathsf{iter}} \mid G_2, E_G, S_X, Z_1, Z_2]\Big]\right) \\ &\geq c'/2 \min_{Z_1,Z_2} \left(\mathbb{E}[N_{\mathsf{iter}} \mid G_1, E_G, S_X, Z_1, Z_2] + \mathbb{E}[N_{\mathsf{iter}} \mid G_2, E_G, S_X, Z_1, Z_2]\right)\end{aligned}$$

Letting $S_X = x_1, x_2, \ldots$, denote by $N_1 \equiv N_1(S_X, Z_1, Z_2)$ the length of the shortest prefix of $x_T, x_{T+1}, \ldots$ that includes an element from $\Phi(Z_1)$ and at some later point an element from $\Phi(Z_2)$, and by $N_2 \equiv N_2(S_X, Z_1, Z_2)$ the length of the shortest prefix that includes the reverse order. Clearly, the number of iterations under G_i is at least N_i. Therefore

$$\mathbb{E}[N_{\mathsf{iter}} \mid S_X, E_G] \geq c'/2 \min_{Z_1,Z_2} (N_1(S_X, Z_1, Z_2) + N_2(S_X, Z_1, Z_2)).$$

Let $N \equiv N(S_X, Z_1, Z_2) := \max(N_1(S_X, Z_1, Z_2), N_2(S_X, Z_1, Z_2))$. Observe that in the subsequence $x_T, \ldots, x_{T+N}$, there is an element from $\Phi(Z_1)$ followed by one from $\Phi(Z_2)$ some time later, and also the reverse order. It follows there are at least two elements from $\Phi(Z_i)$ for at least one of $i \in \{1, 2\}$ in this subsequence. Let $N' \equiv N'(S_X, Z_1, Z_2)$ be the smallest integer such that $x_T, \ldots, x_{T+N'}$ includes at least one of the two options. Then $N' \leq N$, and so

$$\mathbb{E}[N_{\mathsf{iter}} \mid S_X, E_G] \geq c'/2 \min_{Z_1,Z_2} (N'(S_X, Z_1, Z_2)).$$

Taking expectation over S_X conditioned on E_G, it follows that

$$\mathbb{E}[N_{\mathsf{iter}} \mid E_G] \geq c'/2 \cdot \mathbb{E}_{S_X}[\min_{Z_1,Z_2} (N'(S_X, Z_1, Z_2)) \mid E_G].$$

therefore, since $\mathbb{P}[E_G] \geq c/2$,

$$\mathbb{E}[N_{\mathsf{iter}}] \geq c' \cdot c/4 \cdot \mathbb{E}_{S_X}[\min_{Z_1,Z_2} (N'(S_X, Z_1, Z_2)) \mid \mathbb{E}_G]. \tag{6}$$

We now lower-bound the RHS of this inequality. Let K be an integer. We have

$$\mathbb{E}_{S_X}[\min_{Z_1,Z_2} N' \mid \mathbb{E}_G] \geq K(\mathbb{P}[\min_{Z_1,Z_2} N' \geq K] - (1 - \mathbb{P}[E_G])) \geq K(c/2 - \mathbb{P}[\min_{Z_1,Z_2} N' < K]). \tag{7}$$

We have left to upper-bound $\mathbb{P}[\min_{Z_1,Z_2} N' < K]$ for a non-trivial value of K. Denoting $S_X = x_1, x_2, \ldots$, we have

$$\begin{aligned}\mathbb{P}[\min_{Z_1,Z_2} N' < K] &= \mathbb{P}[\exists i \in \{1, 2\}, Z_i \subseteq \texttt{base}(i), j < j' < K, \text{ s.t. } |Z_i| = q/4, x_j, x_{j'} \in \Phi(Z_i)] \\ &\leq 2\mathbb{P}[\exists Z_1 \subseteq \texttt{base1}, j < j' < K, \text{ s.t. } |Z_1| = q/4, x_j, x_{j'} \in \Phi(Z_1)].\end{aligned}$$

The last inequality follows from a union bound and the symmetry between the cases $i = 1$, $i = 2$. For $x \in \texttt{perm1}$, denote $L(x) := \cup_{Z \subseteq \texttt{base1}: |Z| = q/4, x \in \Phi(Z))} \Phi(Z)$. For $x \notin \texttt{perm1}$, let $L(x) := \emptyset$. We have

$$\begin{aligned}&\mathbb{P}[Z_1 \subseteq \texttt{base1}, j < j' < K, \text{ s.t. } |Z_i| = q/4, x_j, x_{j'} \in \Phi(Z_1)] \\ &\leq \mathbb{P}[\exists j < j' < K \text{ s.t. } x_{j'} \in L(x_j)] \\ &\leq K^2 \max_{x \in \texttt{perm1}} \mathbb{P}_{X \sim \mathcal{D}_X}[X \in L(x)] \\ &\leq K^2 \max_{x \in \texttt{perm1}} |L(x)|/|\Pi_{m/4}| \\ &\leq K^2 (8dq/m)^{q/2},\end{aligned}$$

where the last inequality follows from the last part of Lemma 5. Substituting for the value of d, it follows that

$$\mathbb{P}[N < K] \leq K^2 \left(8q^2 \log(q/c)/m\right)^{q/4}.$$

1. We abuse notation and let Φ denote a mapping from $\texttt{base}(i)$ to $\texttt{perm}(i)$ for both of $i \in \{1, 2\}$, using the obvious isomorfisms.

Setting $K = \sqrt{c}/2 \cdot \left(\frac{m}{8q^2 \log(q/c)}\right)^{q/8}$, we get that $\mathbb{P}[N < K] \leq c/4$, therefore by Eq. (6) and Eq. (7),

$$\mathbb{E}[N_{\mathsf{iter}}] \geq c' \cdot c/4 \cdot K(c/4) \geq c' \cdot c^{5/2}/32 \cdot \left(\frac{m}{8q^2 \log(q/c)}\right)^{q/8}.$$

Substituting $c' = 0.15$, $c = 0.68$ and rounding downward, we get the statement of the lemma. ■

The lower bounds above indicate that to improve the dependence on q, one must consider a more restricted class of pool algorithms. This is the topic of the next section.

5. Utility-Based Pool Algorithms

In Section 4 we provided an algorithm with a uniform guarantee on the expected number of iterations. However, this guarantee was exponential q, and as our lower bounds show, this dependence cannot be removed for a general pool algorithm. We now consider a more restricted class of pool algorithms, which we term *utility-based* pool algorithms, and show that it allows stream emulation with an expected number of iterations linear in q. Utility-based pool algorithms are defined in Section 5.1. In Section 5.2 an algorithm that emulates utility-based pool algorithms in a streaming setting is proposed, and it is shown that for this algorithm, the expected number of iterations is at most linear in q, in contrast to the exponential dependence required in the general case. In Section 5.3 two lower bounds are proved, showing that a quasi-linear dependence of the number of iterations on q cannot be avoided when emulating utility-based pool algorithms.

5.1 Defining Utility-Based Algorithms

A common approach for designing pool-based interactive algorithms is to define a utility function that scores each element depending on the history of selected elements and their responses so far (e.g., Seung et al., 1992; Lewis and Gale, 1994; Tong and Koller, 2002; Guo and Greiner, 2007; Golovin et al., 2010b; Guillory and Bilmes, 2010; Golovin and Krause, 2011; Gonen et al., 2013; Cuong et al., 2014). In each round, the algorithm selects the element that maximizes the current utility function. For example, the score can estimate the marginal benefit of selecting an element based on the current information on the source distribution, as gleaned from the previous elements and their responses. We consider black-box emulation for this class of pool-based algorithms.

Formally, a utility-based interactive pool algorithm is defined by a utility function $\mathcal{U}$, of the form $\mathcal{U} : \cup_{n=0}^{\infty} \mathcal{X} \times (\mathcal{X} \times \mathcal{Y})^n \to \mathbb{R}_{>0}$. The value of $\mathcal{U}(x, S_{t-1})$ represents the score of element x given history S_{t-1}. The pool algorithm selects, in each round, the element that is assigned the maximal score by the utility function given the history. We assume for simplicity that there are no ties in $\mathcal{U}$ and that all its outputs are positive. The general form of a utility-based interactive pool algorithm for $\mathcal{U}$, denoted $\mathcal{A}_p^{\mathcal{U}}$, is given in Alg. 4.

Our goal when emulating a pool algorithm is not to maximize $\mathcal{U}$ on the selected elements, but to exactly emulate the behavior of the pool algorithm $\mathcal{A}_p^{\mathcal{U}}$. This is because we do not assume any specific relationship between the value of the utility function and the reward of the algorithm. For instance, the utility-based pool algorithm might be empirically successful, while its analysis is not fully understood (e.g. Tong and Koller, 2002).

Algorithm 4 $\mathcal{A}_p^{\mathcal{U}}$

input Elements $x_1, \ldots, x_m$, budget $q < m$.

$S_0 \leftarrow ()$
$M_0 \leftarrow [m]$
for $t = 1 : q$ **do**
  $i_t \leftarrow \operatorname{argmax}_{j \in M_{t-1}} \mathcal{U}(x_j, S_{t-1})$.
  Select x_{i_t}, get y_{i_t}.
  $S_t \leftarrow S_{t-1} \circ (x_{i_t}, y_{i_t})$.
  $M_t \leftarrow M_{t-1} \setminus \{i_t\}$.
end for
Output the set of all pairs in S_q.

5.2 Stream Emulation for Utility-Based Pool Algorithms

We propose a stream algorithm that emulates a utility-based pool algorithm using a black-box solution to the well-known *secretary problem* (Dynkin, 1963; Gilbert and Mosteller, 1966; Ferguson, 1989). We first present this classical riddle and discuss its possible solutions in Section 5.2.1. The stream algorithm and its analysis are presented in Section 5.2.2.

5.2.1 THE SECRETARY PROBLEM WITH A BI-CRITERION

In the classical formulation of the secretary problem, an algorithm sequentially observes a stream of n different real numbers, and selects a single number. The goal of the algorithm is to select the unique maximal number out of the n numbers, but it can only select a number immediately after it is observed, before observing the rest of the numbers. It is assumed that the n numbers in the stream are unknown and selected by an adversary, but their order of appearance is uniformly random.

The classical goal is to select the maximal number with a maximal probability, where n is known to the algorithm. This task can be optimally solved by a simple deterministic algorithm, achieving a success probability which approaches $1/e$ in the limit of $n \to \infty$ (Dynkin, 1963; Gilbert and Mosteller, 1966; Ferguson, 1989). The optimal algorithm observes the first $t(n)$ numbers, and then selects the next observed number which is at least as large as the maximal in the first $t(n)$. The limit of $t(n)/n$ for $n \to \infty$ is $1/e$.

In the next section we show how any secretary problem strategy can be used to emulate a utility-based pool algorithm in a streaming setting. However, while the classical formulation of the secretary problem is concerned only with the probability of success, in our context an additional criterion is important. Thus for a given strategy, we consider the following two criteria:

1. The probability of success—the probability that the strategy selects the maximal element out of n, assuming a random ordering of the input sequence. This is the single criterion of the classical secretary problem.

2. The *success/select ratio*—the conditional probability that a number is maximal, given that the strategy selected it.

For the second criterion to be meaningful, we allow the strategy to decide not to select any number. For instance, in the classical optimal solution to the secretary problem, the strategy should not select

any number if after the first $t(n)$ numbers, all the other numbers are smaller than the maximal in the first $t(n)$. As we see in Section 5.2.2, we would optimally like the strategy to have both a high success probability and a high success/select ratio. This is because a high success probability leads to a lower number of observed elements N_{iter} required by the stream algorithm, while a high success/select ratio leads to a lower number of selections of elements N_{sel}.

The classical solution maximizes the success probability, but does not optimize for the success/select ratio. Another extreme can be found in the following strategy, which sets the success/select ratio at one, at the expense of the success probability: Wait for the last number, and then select it only if it is the maximal out of all the observed items. This strategy never selects a non-maximal number, hence its success/select ratio is one, however its success probability is only $1/n$. Denote this strategy SecPr[last].

In general, consider strategies of the following form: For a given $M \geq 1$, observe the first $M-1$ numbers. Then select the first number which is larger than the maximal in the first $M-1$ numbers. If no such number is observed, avoid selecting a number. Denote this strategy SecPr$[M]$. Denote by $p_s(M)$ the success probability of this strategy, and by $\mathrm{sr}(M)$ its success/select ratio. The analysis of $p_s(M)$ (see Ferguson, 1989) gives $p_s(1) = 1/n$, and for $M > 1$,

$$p_s(M) = \frac{M-1}{n}(H_{n-1} - H_{M-1}),$$

where H_i is the i'th harmonic number $H_i := \sum_{j=1}^{i} \frac{1}{i}$.

To calculate $\mathrm{sr}(M)$, note that the event that some number is selected occurs exactly when the maximal number is in the last $n-(M-1)$ observed numbers. Therefore the probability of selecting a number is $1 - \frac{M-1}{n}$, and so

$$\mathrm{sr}(M) = \frac{p_s(M)}{1 - \frac{M-1}{n}}.$$

Setting $\alpha := \frac{M-1}{n}$ and considering large n, we have $H_{n-1} - H_{M-1} \approx \ln(\frac{n-1}{M-1}) \approx \ln(1/\alpha)$. Therefore

$$p_s(M) \approx \alpha \log(1/\alpha), \quad \text{and} \quad \mathrm{sr}(M) \approx \frac{\alpha \log(1/\alpha)}{1-\alpha}.$$

For $\alpha \in (0,1)$, $p_s(M)$ is concave with a single maximum at $\alpha = 1/e$, while $\mathrm{sr}(M)$ is monotonic increasing. Since we wish for both criteria to be large, the Pareto frontier includes only and all the solutions with $\alpha \geq 1/e$. See Figure 2 and Figure 3 for the trade-off between $p_s(M)$ and $\mathrm{sr}(M)$ as α changes between $1/e$ and 1.

In the implementation of the stream algorithm shown in the following section, the secretary problem strategy is executed multiple times, for sequences of different lengths. Given any secretary problem strategy for length m, we apply it for any length up to m, using the procedure in Alg. 5.

When running SecPrVar, the subset I should be drawn uniformly at random from all possible subsets. This way, clearly the success probability and the success/select ratio for SecPrVar, for any sequence of positive numbers of length $m' \leq m$, are at least as high as those for SecPr with a sequence of length m.

We also consider stream emulation using precognitive stream algorithms. In this case, the secretary problem can easily be perfectly solved: it may be assumed that the algorithm knows in advance all the numbers in the sequence and thus it can select the maximal number with probability 1. We

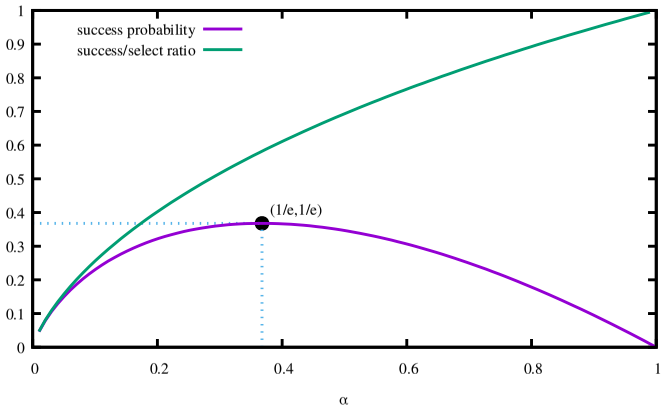


Figure 2: The success probability and the success/select ratio for large n, as a function of α.

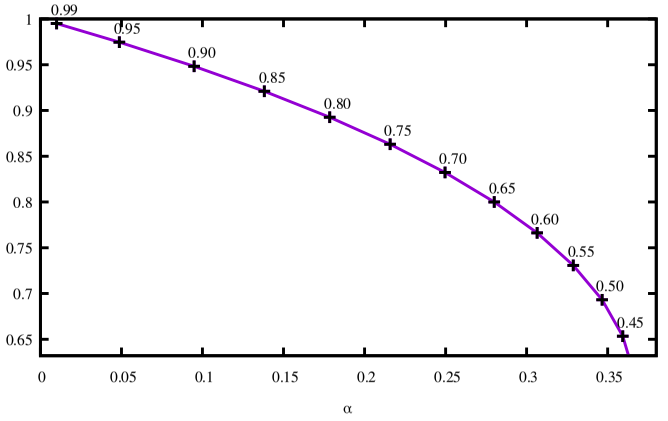


Figure 3: The trade-off between the success probability and the success/select ratio for large n, in the Pareto frontier of α values, for strategies of the form SecPr$[\alpha n]$. On the Pareto frontier α ranges between $1/e$ (largest success probability) and 1 (smallest success probability). The labels on the points indicate the value of α.

Algorithm 5 SecPrVar(m, m', I,SecPr): Secretary problem for sequences of variable length

input Integers $m, m' \leq m$, $I \subseteq [m]$, a secretary problem strategy SecPr for sequence length m.

$R \leftarrow ()$
for $i = 1 : m$ **do**
  if $i \notin I$ **then**
    $R \leftarrow R \circ (0)$ # Add a dummy zero to sequence
  else
    Get next number r from input
    $R \leftarrow R \circ (r)$
    if r should be selected according to SecPr on R **then**
      Select r and terminate.
    end if
  end if
end for

will denote this strategy $SecPr[\text{precog}]$. It has success probability and success/select ratio both equal to 1.

We next present and analyze the stream emulation algorithm for utility-based pool algorithms, which uses a secretary problem strategy via SecPrVar.

5.2.2 THE STREAM EMULATION ALGORITHM

We propose the stream emulation algorithm $\mathcal{A}_s^{\mathcal{U}}$, listed in Alg. 6. This algorithm repeatedly applies a secretary problem strategy to decide which elements to select. Because the secretary problem strategy might fail to select the maximal number, repeated applications of the strategy might be necessary. This trial-and-error approach means that $\mathcal{A}_s^{\mathcal{U}}$ might select more than q elements. However, the expected number of selected elements is a constant factor over q. The trade-off between the number of iterations and number of selected elements is controlled by the probability of success and the success/select ratio of the selected strategy SecPr, which is provided to $\mathcal{A}_s^{\mathcal{U}}$ as input.

$\mathcal{A}_s^{\mathcal{U}}$ is equivalent to $\mathcal{A}_p^{\mathcal{U}}$, as Theorem 10 below shows. In order to guarantee this equivalence, $\mathcal{A}_s^{\mathcal{U}}$ never selects an element that could not have been in a pool in which the previous elements were selected. This is achieved by discarding such elements in each round: The set $\mathcal{X}_i$ is the set of elements that are allowed in round i, and is defined to include only elements that could not have been selected by the pool algorithm before round i. To bound the expected number of iterations, we show in Theorem 11 that the probability mass of $\mathcal{X}_i$ can be controlled in expectation, which leads to a bound on the expected number of discarded elements.

Theorem 10 *For any utility function $\mathcal{U}$, any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, any integer m and $q \leq m$, $\mathcal{A}_s^{\mathcal{U}}$ is $(q, \mathcal{D})$-equivalent to $\mathcal{A}_p^{\mathcal{U}}$.*

Proof Consider the probability space defined by $S \sim \mathcal{D}^m$ and $S' \sim \mathcal{D}^\infty$, where S, S' are independent. We prove the equivalence by showing that for any $j \in [q]$ and $L_j = ((x_{i,k_i}, y_{i,k_i}))_{i \in [j]}$ that could have been selected by the pool algorithm,

$$d\mathbb{P}[\mathsf{pairs}_p(S, j+1) \mid \mathsf{pairs}_p(S, [j]) = L_j] = d\mathbb{P}[\mathsf{pairs}_s(S', j+1) \mid \mathsf{pairs}_s(S', [j]) = L_j].$$

Algorithm 6 $\mathcal{A}_s^{\mathcal{U}}$ Stream-emulation for utility-based pool algorithms

input Integers $m, q \leq m$, a secretary problem strategy SecPr for sequences of length m.

$L_0 \leftarrow ()$
$\mathcal{X}_1 = \mathcal{X}$
for $i = 1 : q$ **do**
  repeat
    Draw a random subset $I \subseteq [m]$ of size m'
    for $j = 1 : m - i + 1$ **do**
      Repeatedly draw elements from $\mathcal{D}_X$, until drawing an element in $\mathcal{X}_i$.
      Denote it $x_{i,j}$, and let $r_{i,j} \leftarrow \mathcal{U}(x_{i,j}, L_{i-1})$.
      if SecPrVar($m, m-i+1, I$,SecPr) would select the last element in the sequence prefix $r_{i,1}, \ldots, r_{i,j}$ **then**
        $k \leftarrow j$
        Select $x_{i,k}$, get its response $y_{i,k}$.
      end if
    end for
  until $r_{i,k} = \max\{r_{i,1}, \ldots, r_{i,m-i+1}\}$.
  $k_i \leftarrow k$
  $L_i \leftarrow L_{i-1} \circ (x_{i,k_i}, y_{i,k_i})$.
  $\mathcal{X}_{i+1} \leftarrow \{x \in \mathcal{X}_i \mid \mathcal{U}(x, L_{i-1}) < \mathcal{U}(x_{i,k_i}, L_{i-1})\}$
end for
Output the set of pairs in L_q.

For a given L_j, denote by $\mathcal{D}_{j+1}$ the distribution generated by drawing $(X, Y) \sim \mathcal{D}$ conditioned on $X \in \mathcal{X}_{j+1}$, where $\mathcal{X}_{j+1}$ depends on L_j. Denote by $\mathcal{G}$ all the finite sequences of pairs such that when the optimal secretary problem solution is applied to the sequence, it succeeds. That is, the optimal value under the score $(x, y) \to \mathcal{U}(x, L_j)$ is indeed selected. From the definition of $\mathcal{A}_{\mathcal{U}}^s$, we have

$$d\mathbb{P}[\mathsf{pairs}_s(S', j+1) \mid \mathsf{pairs}_s(S', [j]) = L_j] = d\mathbb{P}_{\bar{S} \sim \mathcal{D}_{j+1}^{m-j}}[\operatorname*{argmax}_{(x,y) \in \bar{S}} \mathcal{U}(x, L_j) \mid \bar{S} \in \mathcal{G}].$$

For a given sequence $\bar{S} = ((\bar{x}_i, y_i))_{i \in [m-j]}$, let $\sigma(\bar{S}) : [m-j] \to [m-j]$ be a permutation such that for all $i \leq m - j$, $\bar{x}_{\sigma(i)} \leq \bar{x}_{\sigma(i+1)}$. The success of the optimal secretary problem algorithm depends only on the ordering of ranks in its input sequence, hence there is a set of permutations $\mathcal{G}'$ such that $\bar{S} \in \mathcal{G}$ if and only if $\sigma(\bar{S}) \in \mathcal{G}'$. Now, $\operatorname{argmax}_{(x,y) \in \bar{S}} \mathcal{U}(x, L_j)$ depends only on the identity of pairs in $\bar{S}$, while $\sigma(\bar{S})$ depends only on their order. Since the elements in $\bar{S}$ are i.i.d., these two properties are independent. Therefore

$$d\mathbb{P}_{\bar{S} \sim \mathcal{D}_{j+1}^{m-j}}[\operatorname*{argmax}_{(x,y) \in \bar{S}} \mathcal{U}(x, L_j) \mid \bar{S} \in \mathcal{G}] = d\mathbb{P}_{\bar{S} \sim \mathcal{D}_{j+1}^{m-j}}[\operatorname*{argmax}_{(x,y) \in \bar{S}} \mathcal{U}(x, L_j)].$$

Therefore

$$\begin{aligned}
&d\mathbb{P}[\mathsf{pairs}_s(S', j+1) \mid \mathsf{pairs}_s(S', [j]) = L_j] \\
&= d\mathbb{P}_{\bar{S}\sim\mathcal{D}^{m-j}_{j+1}}[\operatorname*{argmax}_{(x,y)\in\bar{S}} \mathcal{U}(x, L_j)] \\
&= d\mathbb{P}_{\hat{S}\sim\mathcal{D}^{m-j}}[\operatorname*{argmax}_{(x,y)\in\hat{S}} \mathcal{U}(x, L_j) \mid \hat{S} \subseteq (\mathcal{X}_{j+1} \times \mathcal{Y})^{m-j}] \\
&= d\mathbb{P}_{\hat{S}\sim\mathcal{D}^{m-j}}[\operatorname*{argmax}_{(x,y)\in\hat{S}} \mathcal{U}(x, L_j) \mid \forall (x,y) \in \hat{S}, i \in [j],\ \mathcal{U}(x, L_{i-1}) < \mathcal{U}(x_{i,k_i}, L_{i-1})] \\
&= d\mathbb{P}_{\hat{S}\sim\mathcal{D}^{m-j}}[\operatorname*{argmax}_{(x,y)\in\hat{S}} \mathcal{U}(x, L_j) \mid \mathsf{pairs}_p(L_j \circ \hat{S}, [j]) = L_j] \\
&= d\mathbb{P}_{S\sim\mathcal{D}^{m}}[\operatorname*{argmax}_{(x,y)\in S\setminus L_j} \mathcal{U}(x, L_j) \mid \mathsf{pairs}_p(S, [j]) = L_j] \\
&= d\mathbb{P}_{S\sim\mathcal{D}^{m}}[\mathsf{pairs}_p(S)(j+1) \mid \mathsf{pairs}_p(S, [j]) = L_j].
\end{aligned}$$

Here L_i is the prefix of length i of L_j. Since this equality holds for all $j \in [q-1]$, $d\mathbb{P}[\mathsf{pairs}_s(S', [q])] = d\mathbb{P}[\mathsf{pairs}_p(S, [q])]$. ■

The following theorem gives the expected number of selected elements and the expected number of observed elements used by $\mathcal{A}^{\mathcal{U}}_s$. Both of these values depend on the properties of the secretary problem strategy SecPr that $\mathcal{A}^{\mathcal{U}}_s$ receives as input.

Theorem 11 *Suppose that $\mathcal{A}^{\mathcal{U}}_s$ is run with $m, q \leq m$ and a secretary problem strategy SecPr for sequences of size m with success probability p_s and success/select ratio* sr. *Then, for any utility function $\mathcal{U}$ and any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$*

$$\mathbb{E}_{S\sim\mathcal{D}^\infty}[N_{\mathsf{sel}}(\mathcal{A}^{\mathcal{U}}_s, S, q)] = q/\mathrm{sr}$$

and

$$\mathbb{E}_{S\sim\mathcal{D}^\infty}[N_{\mathsf{iter}}(\mathcal{A}^{\mathcal{U}}_s, S, q)] = mq/p_s.$$

To observe the implications of these upper bounds, consider for instance the classical secretary problem strategy SecPr$[t(n)]$, as defined in Section 5.2.1, which has $p_s \approx 1/e$ and $\mathrm{sr} \approx \frac{1}{e(1-1/e)}$ for large m. It follows that for $m \to \infty$, the expected number of selected elements by $\mathcal{A}^{\mathcal{U}}_s$ is eq, and the expected number of observed elements is eqm. Other choices of parameters for the secretary problem lead to other points on the Pareto frontier in Figure 3. All the points on the Pareto frontier give a constant factor over q for the expected number of selected elements, and a factor linear in q over m for the expected number of iterations. To get an algorithm that selects exactly q elements, one can use the strategy SecPr[last], which gives $p_s = 1/m$ and $\mathrm{sr} = 1$, leading to an expected number of iterations of qm^2. In the precognitive setting, one can use $SecPr$[precog], which gives an expected number of selections exactly q, and an expected number of iterations mq.

Proof [of Theorem 11] Call a full run of the loop starting at step 6 an attempt for the i'th element. In each attempt for the i'th element, $m - i + 1$ elements from $\mathcal{X}_i$ are observed. The expected number of element selection attempts for each successful element selection is $\frac{1}{\mathrm{sr}}$. Thus,

$$\mathbb{E}_{S\sim\mathcal{D}^\infty}[N_{\mathsf{sel}}(\mathcal{A}^{\mathcal{U}}_s, S, q)] = \frac{1}{\mathrm{sr}} \cdot q.$$

For the second part of the statement, we need to bound $\mathbb{E}_{S\sim\mathcal{D}^\infty}[N_{\mathsf{iter}}(\mathcal{A}^{\mathcal{U}}_s, S, q)]$. The total expected number of attempts of running the secretary problem for each successful selection of an element is $\frac{1}{p_s}$, and the expected number of elements from $\mathcal{X}_i$ observed in each attempt is $m - i + 1$. Thus the expected number of elements from $\mathcal{X}_i$ observed until x_i is selected is $\frac{1}{p_s} \cdot (m - i + 1)$. However, drawing a single element from $\mathcal{X}_i$ might require several draws from $\mathcal{X}$. To bound the total expected number of observed elements, we now consider the definition of $\mathcal{X}_i$.

Denote by f_i the utility function $\mathcal{U}(\cdot, L_{i-1})$. Let $x_i := x_{i,k_i}$, be the i'th element added to L_i. Then $\mathcal{X}_i = \{x \in \mathcal{X}_{i-1} \mid f_{i-1}(x) \leq f_{i-1}(x_{i-1})\}$. Consider the probability space defined by the input to the stream algorithm $S \sim \mathcal{D}^\infty$, and let $Z_i, Z'_i \sim \mathcal{D}_X$ for $i \in [q]$ such that these random variables and S are all independent. Denote

$$p(\alpha, i) := \mathbb{P}[f_i(Z_i) \leq \alpha \mid Z_i \in \mathcal{X}_i].$$

$p(\alpha, i)$ is a random variable since $\mathcal{X}_i$ depends on S. Let $U_i := p(f_i(Z'_i), i)$. Since we assume no ties in $\mathcal{U}$, and no single x has a positive probability in $\mathcal{D}_X$, then conditioned on $\mathcal{X}_i$, U_i is distributed uniformly in $[0, 1]$. Hence $U_1, \ldots, U_q$ are statistically independent.

For $i > 1$, define the random variable $M_i := p(f_{i-1}(x_{i-1}), i-1)$. Then $M_i = \mathbb{P}[\mathcal{X}_i]/\mathbb{P}[\mathcal{X}_{i-1}]$. The expected number of elements that need to be drawn from $\mathcal{D}$ to get a single element from $\mathcal{X}_i$ is $1/\mathbb{P}[\mathcal{X}_i] = (\prod_{j=1}^{i} M_j)^{-1}$. Therefore,

$$\mathbb{E}[N_{\mathsf{iter}}(\mathcal{A}^{\mathcal{U}}_s, S, q) \mid M_2, \ldots, M_q] = \sum_{i=1}^{q} \frac{p_s^{-1} \cdot (m - i + 1)}{\prod_{j=1}^{i} M_j}.$$

The element x_i maximizes the function $x \mapsto f_i(x)$ over $m-i+1$ independent draws of elements x from $\mathcal{D}_X$ conditioned on $x \in \mathcal{X}_i$, hence it also maximizes $x \mapsto p(f_i(x), i)$. Therefore, for $i > 1$, M_i is the maximum of $m - i + 2$ independent copies of U_i, hence $\mathbb{P}[M_i \leq p] = p^{m-i+2}$. Hence

$$d\mathbb{P}[M_2, \ldots, M_q](p_2, \ldots, p_q)/dp_2 \cdot \ldots \cdot dp_q = \prod_{i=2}^{q} d\mathbb{P}[M_i \leq p_i]/dp_i = \prod_{i=2}^{q} (m - i + 2) p_i^{m-i+1}.$$

We have

$$\begin{aligned}
\mathbb{E}[N_{\mathsf{iter}}(\mathcal{A}^{\mathcal{U}}_s, S, q)] &= \int_{M_2=0}^{1} \cdots \int_{M_q=0}^{1} \mathbb{E}[N_{\mathsf{iter}}(\mathcal{A}^{\mathcal{U}}_s, S, q) \mid M_1, \ldots, M_q] d\mathbb{P}[M_1, \ldots, M_q] \\
&= p_s^{-1} \int_{M_2=0}^{1} \cdots \int_{M_q=0}^{1} \sum_{i=1}^{q} \frac{m - i + 1}{\prod_{j=1}^{i} M_j} \prod_{l=2}^{q} (m - l + 2) M_l^{m-l+1} dM_l \\
&= p_s^{-1} \sum_{i=1}^{q} (m - i + 1) \int_{M_2=0}^{1} \cdots \int_{M_q=0}^{1} \prod_{l=2}^{i} (m - l + 2) M_l^{m-l} dM_l \\
&\qquad \cdot \prod_{l=i+1}^{q} (m - l + 2) M_l^{m-l+1} dM_l,
\end{aligned}$$

Therefore

$$\mathbb{E}[N_{\mathsf{iter}}(\mathcal{A}_s^{\mathcal{U}}, S, q)] = p_s^{-1} \sum_{i=1}^{q} (m-i+1) \prod_{l=2}^{i} \frac{m-l+2}{m-l+1}$$
$$= p_s^{-1} \sum_{i=1}^{q} (m-i+1) \cdot \frac{m}{m-i+1} = mq/p_s.$$

This concludes the proof. ■

5.3 Lower Bounds

The following lower bounds show that the number of iterations required by any stream emulator for utility-based algorithms must be at least quasi-linear in q. We provide two results. The first result considers a stream algorithm that selects exactly q elements, and it allows q to be large with respect to m. The second result considers stream algorithms that are allowed to select more than q elements. In this case, the lower bound is smaller, but it is reduced only linearly in the number of selections. It follows that a constant factor increase in the number of selections cannot overcome the quasi-linear dependence on q in the number of iterations. The proofs of the lower bounds are based on constructing a utility function which in effect allows only one set of selected elements for a given distribution, and forces the stream algorithm to select them in the same order as the pool algorithm. The first lower bound considers stream emulation with exactly q selections. The bound holds for both standard streaming algorithms and precognitive streaming algorithms.

Theorem 12 *For any integer m, $q \leq m/2$, there exists a utility-based pool algorithm, and a marginal $\mathcal{D}_X$, such that any stream algorithm $\mathcal{A}_s$ which is $(q, \mathcal{D})$ equivalent to the pool algorithm for all $\mathcal{D} \in \mathrm{DS}(\mathcal{D}_X)$, and selects only q elements, has*

$$\exists \mathcal{D} \in \mathrm{DS}(\mathcal{D}_X), \mathbb{E}_{S\sim\mathcal{D}^\infty}[N_{\mathsf{iter}}(\mathcal{A}_s, S, q)] \geq \frac{q}{32} \left\lfloor \frac{m}{2\log(4q)} \right\rfloor .$$

The result holds even if $\mathcal{A}_s$ is a precognitive stream algorithm.

Proof Let $n = \left\lfloor \frac{m}{2\log(4q)} \right\rfloor$, and let $\mathcal{D}_X$ be a uniform distribution over $\mathcal{X} = \{a_i \mid i \in [n]\}$. Assume $\mathcal{Y} = \{0,1\}$. A pool of size m then includes all elements in $A := \{a_i \mid i \in [2q-1]\}$ with a probability of at least $\alpha \geq 1 - (2q-1)\exp(-m/n) \geq 1 - \frac{1}{4q}$.

Consider a utility function $\mathcal{U}$ such that given a history of the form $((a_1, y_1), \ldots, (a_t, y_t))$ for $t \in [q-1]$, assigns a maximal score in $\mathcal{X}$ to a_{t+1} if $\forall i \leq t, y_i = 0$, and a maximal score to a_{q+t} if $\exists i \leq t, y_i = 1$. In addition, if the history is $((a_1, y_1), \ldots, (a_t, y_t), (a_{q+t}, y_{t+1}), \ldots, (a_{q+t'}, y_{t'+1}))$ for some $t \in [q-1], t' \in [q-2]$, then $\mathcal{U}$ assigns a maximal score to $a_{q+t'+1}$. Then, in a pool that includes all elements $a_1, \ldots, a_{2q-1}$, the pool algorithm based on $\mathcal{U}$ behaves as follows: In every round, if all selected elements so far received the response 0, it selects at round t the element a_t. Otherwise, it selects the element a_{q+t} and then continues to select $a_{q+t+1}, \ldots, a_{2q-1}$.

Let $\mathcal{D}_0$ be a distribution in which the response is deterministically zero. If the distribution is $\mathcal{D}_0$, $\mathcal{A}_s$ selects $Z_0 = \{a_1, \ldots, a_q\}$ with a probability at least α. Denote $\mathcal{D}_t$ for $t \in [q]$, in which the response is deterministically zero for $X \in \{a_1, \ldots, a_q\} \setminus \{a_t\}$ and 1 for a_t. For this distribution, the algorithm must select the elements in $Z_t = \{a_1, \ldots, a_t, a_{q+t}, \ldots, a_{2q-1}\}$ with a probability at least α.

Consider the probability space defined by the input sequence $S \sim \mathcal{D}_0^\infty$ and the randomness of $\mathcal{A}_s$. Let E_0 be the event in this space that $\mathcal{A}_s$ selects $\{a_1, \ldots, a_q\}$, and let $\bar{E}_0$ be the event in this space that $\mathcal{A}_s$ selects $(a_1 \ldots, a_q)$ in order. We have $\mathbb{P}[E_0] \geq \alpha$. Denote $\beta = \mathbb{P}[\bar{E}_0]$. We show a lower bound on β.

Let T be a random variable in the same probability space, such that $T = 0$ if E_0 does not hold. If E_0 does hold, then T is the smallest round in which the algorithm selects some $a_{t'}$, for $t' > T$, or $T = 0$ if no such round exists. If $\mathcal{A}_s$ selects $\{a_1, \ldots, a_q\}$ but not in order, then $T \in [q]$. Therefore, $\mathbb{P}[T \in [q]] \geq \alpha - \beta$, hence there exists some $t^* \in [q]$ such that $\mathbb{P}[T = t^*] \geq (\alpha - \beta)/q$. Now, consider the distribution $\mathcal{D}_{t^*}$. Define a sequence of pairs $\gamma(S)$ such that S and $\gamma(S)$ have the same elements in the same order, and the responses in $\gamma(S)$ are determined by $\mathcal{D}_{t^*}$ instead of by $\mathcal{D}_0$. Clearly, $\gamma(S)$ is distributed according to $\mathcal{D}_{t^*}^\infty$. Consider a run of the algorithm on S in which E_0 holds, and a parallel run (with the same random bits) on $\gamma(S)$. The algorithm selects the same elements for both sequences until the T'th selection, inclusive. By the definition of T, the T'th selection is some element in $\{a_{T+1}, \ldots, a_q\}$. If $T = t^*$, then an element not in Z_{t^*} is selected in round T. But this same element would be selected by $\mathcal{A}_s$ in the parallel run on $\gamma(S)$. Since under $\gamma(S) \sim \mathcal{D}_{t^*}^\infty$, $\mathcal{A}_s$ selects exactly the set Z_{t^*} with a probability of at least α, and so we have $\mathbb{P}[T = t^*] \leq 1 - \alpha$. This holds also for the precognitive stream algorithm since it is required from the emulation of the pool algorithm. It follows that $(\alpha - \beta)/q \leq \mathbb{P}[T = t^*] \leq 1 - \alpha$. Hence $\beta \geq \frac{1}{2}$.

Let W_i be the number of elements that $\mathcal{A}_s$ observes after selecting element $i-1$, until observing the next element. Let $X_1, X_2, \ldots$ be the sequence of elements observed after selecting the first $i-1$ elements, and for integers j, let $B_j = \mathbb{I}[X_j = a_i]$. $B_1, B_2, \ldots$ are independent Bernoulli random variables with $\mathbb{P}[B_j = 1] = 1/n$, and $\mathbb{P}[B_{W_i} = 1] \geq \mathbb{P}[\bar{E}_0] = \beta \geq \frac{1}{2}$. By Lemma 2, if $\frac{1}{n} \leq \frac{1}{32}$, $\mathbb{E}[W_i] \geq \frac{n}{16}$. It follows that the expected number of iterations for making q selections is at least $\frac{qn}{16}$ if $n \geq 32$. Since it is also at least q, a lower bound of $\frac{qn}{32} = \frac{q}{32}\left\lfloor \frac{m}{2\log(4q)} \right\rfloor$ holds for all n. This analysis holds also for the precognitive stream algorithm, since it also must select the same set of elements in the given order. ■

The second lower bound considers stream emulation with possibly more than q selections. This lower bound also holds for both standard and precognitive stream algorithms.

Theorem 13 *For any integers m, q such that $m \geq 2^{q+1}\log(2q)$ there exists a utility-based pool algorithm, and a marginal $\mathcal{D}_X$, such that for any $\beta \geq 1$, any stream algorithm $\mathcal{A}_s$ which is $(q, \mathcal{D})$ equivalent to the pool algorithm for all $\mathcal{D} \in \mathrm{DS}(\mathcal{D}_X)$, and selects at most βq elements, has*

$$\exists \mathcal{D} \in \mathrm{DS}(\mathcal{D}_X), \mathbb{E}_{S\sim\mathcal{D}^\infty}[N_{\mathsf{iter}}(\mathcal{A}_s, S, q)] \geq \frac{q}{256\beta} \cdot \left\lfloor \frac{m}{\log(2q)} \right\rfloor .$$

This result holds also for precognitive stream algorithms.

Proof Let $n = \left\lfloor \frac{m}{\log(2q)} \right\rfloor$, and let $\mathcal{D}_X$ be a uniform distribution over $\mathcal{X} = \{a_v \mid v \in \{0,1\}^t, t \in [q-1]_0\} \cup \mathcal{X}'$, where $\mathcal{X}'$ includes arbitrary elements so that $|\mathcal{X}| = n$. Note that $\mathcal{X}$ includes $a_()$, which stands for a zero-length vector. Assume $\mathcal{Y} = \{0,1\}$. Let $v \in \{0,1\}^q$ be a binary vector of length

q, and let $v(1:t)$ be its prefix of length $t \in [q-1]_0$. Consider distributions $\mathcal{D}_{v,f}$ for $v \in \{0,1\}^q$ and $f : \cup_{t\in[q-1]_0}\{0,1\}^t \to \{0,1\}$, such that $\mathcal{D}_v$ has a uniform marginal over $\mathcal{X}$, and for every $t \in [q-1]_0$, $\mathbb{P}[Y = v(t+1) \mid X = a_{v(1:t)}] = 1$. In other words, the label of any element which is a prefix of v is the next bit in v. In addition, for any other element a_b, $\mathbb{P}[Y = 1 \mid X = a_b] = f(a_b)$. Denote the elements representing prefixes of v by $A_v := \{a_{v(1:t)} \mid t \in [q-1]_0\}$.

Assume a utility function $\mathcal{U}$ such that for any $b \in \{0,1\}^t$ of length $t < q$, if the last element selected so far was a_b and its response was y, the function assigns the maximal score to $a_{b\circ y}$. Under this utility function, if the distribution is $\mathcal{D}_{v,f}$ and $A_v \subseteq_\pi S_X$, then the pool algorithm selects exactly the elements in A_v. This happens with probability at least $1 - q\exp(-m/n) \geq \frac{1}{2}$.

Let $\mathcal{A}_s$ be a stream-based algorithm which is equivalent to the pool algorithm $\mathcal{A}_p^{\mathcal{U}}$. Let V be a random variable drawn uniformly at random from $\{0,1\}^q$, and let F be a random variable drawn uniformly over all functions $f : \cup_{t\in[q-1]_0}\{0,1\}^t \to \{0,1\}$. Consider the probability space defined by $V, F, S \sim \mathcal{D}_{V,F}^{\infty}$, and the run of $\mathcal{A}_s$ on S. Let I_t be a random variable that is equal to 1 if the t'th selection of $\mathcal{A}_s$ is an element from A_V that has not been selected in previous rounds. Since $\mathcal{A}_s$ is equivalent to $\mathcal{A}_p^{\mathcal{U}}$, then $\mathbb{P}[\sum_{t=1}^{\beta q} I_t = q] \geq \frac{1}{2}$. By the reverse Markov's inequality, there are at least $q/4$ rounds t such that $\mathbb{E}[I_t] \geq \frac{1}{4\beta}$. This holds also for a precognitive stream algorithm due to its emulation of the pool algorithm.

Consider one such round t. Let $Z = \mathsf{pairs}_s(S, [t-1])$. We have $\frac{1}{4\beta} \leq \mathbb{E}[I_t] = \mathbb{E}_Z[\mathbb{E}[I_t \mid Z]]$. By the reverse Markov's inequality, with a probability of at least $\frac{1}{8\beta}$, the value of Z is some z that satisfies $\mathbb{E}[I_t \mid Z = z] \geq \frac{1}{8\beta}$. Let z such that $\mathbb{E}[I_t \mid Z = z] \geq \frac{1}{8\beta}$. Denote by B_t the (random) index of the element a_{B_t} selected in round t. By the reverse Markov's inequality, the probability that $B_t = b$ for some b such that $\mathbb{P}[I_t = 1 \mid Z = z, B_t = b] \geq \frac{1}{16\beta}$ is at least $\frac{1}{16\beta}$. In other words,

$$P(b,t) := \mathbb{P}[a_b \in A_V \mid Z = z, B_t = b] \geq \frac{1}{16\beta}.$$

We now upper bound the number of vectors b such that this inequality holds and they do not already appear in Z. Call such vectors "admissible".

Let Z', z' be the prefixes of length $t-2$ of Z, z respectively, and denote $P(b,t-1) := \mathbb{P}[a_b \in A_V \mid Z' = z']$. Let (w,y) be the last pair in z, and assume without loss of generality that (w,y) is absent from z'. The relationship between $P(b,t-1)$ and $P(b,t)$ can be described based on the relationships between w, y, b, by distinguishing into cases.

1. b is a (weak) prefix of w.
2. $w \circ y$ is a (weak) prefix of b.
3. $w \circ (1-y)$ is a (weak) prefix of b.
4. Neither of the cases above hold. In this case, b and w are incompatible.

In cases 1 and 4, the distribution of y is uniform on $\{0,1\}$ conditioned on b being a prefix of V. It is uniform also conditioned on b not being a prefix of V. Hence, in these cases conditioning on (w,y) does not affect the probability that b is a prefix of V, hence $P(b,t) = P(b,t-1)$. In case 3, $P(b,t) = 0$. We are left with case 2, in which $w \circ y$ is a (weak) prefix of b. In this case we have that $P(b,t) = 2P(b,t-1)$. By induction, we can conclude that if $P(b,t) \neq 0$, then $P(b,t) = 2^{-|b|} \cdot 2^{N(b)}$, where $N(b)$ is the number of pairs in Z of the form (w,y) where $w \circ y$ is a prefix of b.

To bound the number of admissible vectors, let us show a one-to-one mapping between such vectors and binary vectors with a bounded length. For any such b, we have $N(b) - |b| = \log_2(P(b,t)) \geq \log_2(\frac{1}{16\beta})$. Denote $k = \lceil \log_2(16\beta) \rceil$. Then $k \geq |b| - N(b)$. b can be encoded as a vector of length at most k as follows: Go over the bits in b from first to last. For each bit i in b, copy it to the output vector in the next available location only if the prefix $b(1:(i-1))$ does not match some w such that $(w,y) \in Z$. Clearly, $N(b)$ bits are skipped this way, so that the total number of bits in the output is no more than k. Moreover, the mapping is one-to-one, since the only possible ambiguity in decoding is how many bits to decode; but whenever $(w,y) \in Z$, b cannot be equal to w, since w has appeared in Z. Therefore the decoding should stop only when it is not possible to infer more bits. It follows that the number of admissible vectors is at most $2^{k+1} - 1 \leq 64\beta$.

Therefore, the probability of observing such an element in each iteration after round $t-1$ is at most $64\beta/|\mathcal{X}|$. Since there are $q/4$ such iterations, the expected total number of iterations is at least $\frac{qn}{256\beta} = \frac{q}{256\beta} \cdot \left\lfloor \frac{m}{\log(2q)} \right\rfloor$. This result holds also for a precognitive stream algorithm, since it also must select admissible vectors due to the same analysis. ■

6. Active Learning for Binary Classification

Lastly, we consider a special case of interactive algorithms, active learning for binary classification. Recent works provide for this setting relatively tight label complexity bounds, that hold for both the stream-based and the pool-based settings. In Balcan and Long (2013), tight upper and lower bounds for active learning of homogeneous linear separators under isotropic log-concave distributions are provided. The bounds hold for both the stream-based and the pool-based setting, and assume the same budget of unlabeled examples. In Hanneke and Yang (2015), tight minimax label complexity bounds for active learning are provided for several classes of distributions. These bounds also hold for both the stream-based and the pool-based setting. No explicit restriction is placed on the number of unlabeled examples.

The upper bound in Theorem 11 for utility-based pool algorithms can be applied to several pool-based active-learning algorithms which use a utility function (e.g., Golovin and Krause, 2011; Gonen et al., 2013; Cuong et al., 2014). The upper bound shows that when the label budget q is relatively small, the gap between the stream and the pool settings is not significant. For instance, consider an active learning problem in which a utility-based pool active learner achieves a label complexity close to the information-theoretic lower bound for the realizable setting (Kulkarni et al., 1993), so that $q \in \Theta(\log(1/\epsilon))$. The passive learning sample complexity in the realizable setting is at most $m \in \Theta(1/\epsilon)$. Therefore, a stream-based active learner with the same properties needs at most $O(\log(1/\epsilon)/\epsilon)$ unlabeled examples. Therefore, in this case the difference between the pool-based setting and the stream-based setting can be seen as negligible.

Here, we study the general question: what is the possible gap between pool-based and stream based active learning? We study this question in the realizable setting, where the distribution is consistent with some hypothesis in a given finite hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$. We give an upper bound and a lower bound on this gap, which hold for a large class of pool-based algorithms.

First, we show the upper bound. This upper bound is derived by showing that any pool-based active learner for the realizable setting with a finite hypothesis class can be approximated by a utility-based pool algorithm. This in turn implies, using Theorem 11, that a stream algorithm with

similar guarantees can emulate the pool-based algorithm with a bounded approximation factor on the number of iterations and queries.

To show that any pool-based active learner can be approximated by a utility-based pool algorithm, we use the machinery of submodular function maximization for active learning. The challenge is that a successful pool-based active learner might not fully identify the true hypothesis consistent with the distribution, since it suffices that it identifies a hypothesis which is ϵ-close to the true one. Thus, we define a submodular function that allows taking this into account. Our construction combines ideas from Guillory and Bilmes (2010); Golovin et al. (2010b); Dasgupta (2006).

Let $f : \cup_{i=0}^{\infty}(\mathcal{X} \times \mathcal{Y})^n \to \mathbb{N}$ be an objective function which maps any set of labeled examples to a non-negative value. For a given pool of examples, let OPT be the smallest number of queries that are required (in the worst-case over all possible labelings of the pool that are consistent with $\mathcal{H}$) to obtain an S such that $f(S) = Q$ for some fixed integer Q. Guillory and Bilmes (2010) show that if f is monotone non-decreasing and submodular, then there exists a utility-based active learning algorithm which requires at most $O(\log(Q|\mathcal{H}|)) \cdot \mathrm{OPT}$ queries in the worst case to obtain an S such that $f(S) = Q$.

We define a function f which will allow proving the existence of a utility-based pool active learner in our setting. For a given set of labeled examples $S \subseteq \mathcal{X} \times \mathcal{Y}$, the version space induced by S on $\mathcal{H}$ is defined by $\mathrm{VS}(S) := \{h \in \mathcal{H} \mid \forall (x, y) \in S, h(x) = y\}$. Define a fixed set of pairs of hypotheses $E \subseteq \mathcal{H} \times \mathcal{H}$. Let $E(S) := (\mathrm{VS}(S) \times \mathrm{VS}(S)) \cap E$. This is the set of pairs from E such that none of the hypotheses in the pair are disqualified by the labels in S.

Given E, define the objective function f by

$$f(S) := |E| - |E(S)|.$$

Let $Q = |E|$. The following lemma shows that the objective-function requirements above hold for this definition of f.

Lemma 14 *For any $E \subseteq \mathcal{H} \times \mathcal{H}$, f as defined above is monotone and submodular.*

Proof Monotonicity is trivial here since if $S \subseteq S'$ then $|E(S)| \leq |E(S')|$. f is submodular (see e.g., Fujishige 2005) if and only if for any S, S' such that $S \subseteq S'$, and any $(x, y) \notin S'$,

$$f(S \cup \{(x, y)\}) - f(S) \geq f(S' \cup \{(x, y)\}) - f(S'). \tag{8}$$

Denote $S_+ := S \cup \{(x, y)\}$ for brevity, and similarly for S'_+. It is easy to see that $\mathrm{VS}(S) \supseteq \mathrm{VS}(S')$. In addition, $\mathrm{VS}(S) \setminus \mathrm{VS}(S_+) \supseteq \mathrm{VS}(S') \setminus \mathrm{VS}(S'_+)$, since any $h \in \mathrm{VS}(S') \setminus \mathrm{VS}(S'_+)$ is consistent with all the pairs in S' but not with (x, y), implying that $h \in \mathrm{VS}(S) \setminus \mathrm{VS}(S_+)$.

It follows that

$$\mathrm{VS}(S) \times \mathrm{VS}(S) \setminus (\mathrm{VS}(S_+) \times \mathrm{VS}(S_+)) \supseteq \mathrm{VS}(S') \times \mathrm{VS}(S') \setminus (\mathrm{VS}(S'_+) \times \mathrm{VS}(S'_+)).$$

Therefore,

$$(\mathrm{VS}(S) \times \mathrm{VS}(S) \cap E) \setminus ((\mathrm{VS}(S_+) \times \mathrm{VS}(S_+)) \cap E) \supseteq (\mathrm{VS}(S') \times \mathrm{VS}(S') \cap E) \setminus ((\mathrm{VS}(S'_+) \times \mathrm{VS}(S'_+)) \cap E).$$

It follows that $|E(S)| - |E(S_+)| \geq |E(S')| - |E(S'_+)|$, which implies Eq. (8). ■

We use this construction in the proof of the upper bound on the gap between pool-based and stream-based active learning, which we presently state and prove.

A somewhat technical issue is that the results of Guillory and Bilmes (2010) hold only under the assumption that whenever the optimal pool-based algorithm succeeds, it knows that it has succeeded. In other words, it has "proof" that its answer is ϵ-good. This is known as a *self-certifying* algorithm Golovin et al. (2010b). Our results thus apply to pool-based active learning algorithm that have this property as well. This property is related, though not completely equivalent, to the concept of "verifiable active learning algorithms" studied in Balcan et al. (2010). In that work it is shown that non-verifiable active learning algorithms sometime require fewer labels than verifiable active learning algorithms. Characterizing the gap between pool-based and stream-based algorithms for non-verifiable active learning algorithms is an open question which we leave for future work.

For a fixed distribution $\mathcal{D}_X$ over $\mathcal{X}$, denote $\Delta(h, h') := \mathbb{P}[h(X) \neq h'(X)]$.

Theorem 15 *Let $\mathcal{X}$ be a finite instance domain and let $\mathcal{Y}$ be a finite label domain. Let $\mathcal{D}_X$ be a marginal distribution over $\mathcal{X}$. Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a hypothesis class. Suppose that there is a pool-based active learning algorithm such that for any $\mathcal{D} \in \mathrm{DS}(\mathcal{D}_X)$ which is consistent with some $h^* \in \mathcal{H}$, with a probability at least $1 - \delta$ over i.i.d. pools of size m, it outputs $\hat{h} \in \mathcal{H}$ such that $\Delta(\hat{h}, h^*) < \epsilon$ for $(X, Y) \sim \mathcal{D}$, using q label queries. We also assume that the pool algorithm is self-certifying: the algorithm also outputs an indicator $I \in \{0, 1\}$ where $\mathbb{P}[\Delta(\hat{h}, h^*) < \epsilon \mid I = 1] = 1$, and $\mathbb{P}[I = 1] \geq 1 - \delta$.*

Under these assumptions, there exists a stream-based active learner that with a probability of at least $1 - |\mathcal{H}|\delta$ outputs a hypothesis $\hat{h}$ such that $\Delta(\hat{h}, h^*) \leq 2\epsilon$. The expected number of queries requested by the stream-based active learner is at most $O(\log(|\mathcal{H}|)) \cdot q$, and its expected number of iterations is at most $O(\log(|\mathcal{H}|)) \cdot mq$.

Proof Define $E := \{(h, h') \mid \Delta(h, h') \geq 2\epsilon\}$. Denote an ϵ-ball around a hypothesis $h \in \mathcal{H}$ by

$$B(h, \epsilon) := \{h' \in \mathcal{H} \mid \Delta(h, h') < \epsilon\}.$$

Fix a $\mathcal{D} \in \mathrm{DS}(\mathcal{D}_X)$ which is consistent with some $h^* \in \mathcal{H}$. Any pool algorithm with the guarantees assumed by the theorem statement, outputs an $I = 1$ with a probability at least $1 - \delta$, and has $\mathbb{P}[\hat{h} \in B(h^*, \epsilon) \mid I = 1] = 1$. Thus, for any pool with $I = 1$, $h^* \in B(\hat{h}, \epsilon)$. Let S be the set of labeled examples that the pool algorithm obtained during its entire run. Since any of the hypotheses in $\mathrm{VS}(S)$ could be the true h^*, it follows that for any $h \in \mathrm{VS}(S)$, $\Delta(\hat{h}, h) < \epsilon$, hence for any $h, h' \in \mathrm{VS}(S)$, $\Delta(h, h') \leq 2\epsilon$. Therefore, $(h, h') \notin E$. It follows that $E(S) = \emptyset$.

There are $|\mathcal{H}|$ possible distributions $\mathcal{D} \in \mathrm{DS}(\mathcal{D}_X)$ which are consistent with some $h^* \in \mathcal{H}$. It follows that with a probability of $1 - |\mathcal{H}|\delta$ over pools, at the end of the run $E(S) = \emptyset$. For any such pool, we have that q queries suffice to certifiably obtain $f(S) = Q$, thus $\mathrm{OPT} \leq q$.

We now apply the results cited above from Guillory and Bilmes (2010), which imply that there exists a utility-based pool algorithm that obtains $f(S) = Q$ with at most $O(\log(|\mathcal{H}|)q)$ queries. By Theorem 11, we conclude that there exists a stream algorithm which requires at most $O(\log(|\mathcal{H}|)) \cdot q$ queries and $O(\log(|\mathcal{H}|)) \cdot mq$ iterations to obtain $f(S) = Q$.

Since $f(S) = Q$ at the end of the run of the stream algorithm, then $E(S) = \emptyset$, which implies that for any (h, h') such that $\Delta(h, h') \geq 2\epsilon$, at least one of h, h' is not in $\mathrm{VS}(S)$. Therefore, the diameter of $\mathrm{VS}(S)$ is at most 2ϵ. The stream algorithm may therefore return any $\hat{h} \in \mathrm{VS}(S)$, and it holds that $\Delta(\hat{h}, h^*) \leq 2\epsilon$. ■

Next, we provide a lower bound, showing that in the general case $\tilde{\Omega}(mq)$ iterations are required for emulating pool-based active learning in a stream setting. The lower bound, which holds also for the precognitive setting, employs the following example and is presented in Theorem 16.

Example 1 *For given integers m and $q \leq m$, and $T \leq q$, define $\mathcal{X} = \{a_{k,j} \mid k \in [q], j \in \{0, \ldots, 2^{\min(k,T)-1} - 1\}\} \cup \mathcal{X}'$, where $\mathcal{X}'$ includes arbitrary elements so that $|\mathcal{X}| = n$, for some $n \geq q2^T/2$. Define the following hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$.*

$$\mathcal{H} := \{h_i \mid i \in \{0, \ldots, 2^q - 1\}\}, \text{ where } h_i(a_{k,j}) = \begin{cases} \mathbb{I}[i \bmod 2^k = j] & k \leq T, \\ \mathbb{I}[\lfloor i/2^{T-k} \rfloor \bmod 2^T = j]. & k > T. \end{cases} \quad (9)$$

Essentially, for $k \leq T$, $h_i(a_{k,j}) = 1$ if the k least significant bits in the binary expansion of i are equal to the binary expansion of j to T bits. For $k \geq T$, $h_i(a_{k,j}) = 1$ if T consecutive bits in i, starting from bit $T - k$, are equal to the binary expansion of j.

Theorem 16 *Let $q \geq 22$ and $m \geq 8 \log(2q) q^2$ be integers. Consider Example 1 with m, q, setting $T = \lceil \log_2(q) \rceil$ and $n = \lfloor m/7 \log(2q) \rfloor$. Consider $\mathcal{H}$ as defined in Eq. (9). There exist $\delta, \epsilon \in (0, 1)$ such that there is a pool-based active learning algorithm that uses a pool of m unlabeled examples and q labels, such that for any distribution $\mathcal{D}$ which is consistent with some $h^* \in \mathcal{H}$ and has a uniform marginal over $\mathcal{X}$, with a probability of at least $1 - \delta$, $\mathbb{P}[\hat{h}(X) \neq h^*(X)] \leq \epsilon$. On the other hand, for $q > 22$, any stream-based active learning algorithm with the same guarantee, including a precognitive stream algorithm, requires at least $\frac{q}{512} \left\lfloor \frac{m}{7 \log(2q)} \right\rfloor$ unlabeled examples in expectation.*

Proof Let $\mathcal{D}_X$ be uniform over $\mathcal{X}$. Let E be the event that $\mathcal{X} \not\subseteq_\pi S_X$, and define $\delta := \mathbb{P}_{S \sim \mathcal{D}_X^m}[E]$. Define $\epsilon = 1/n$, so that $\mathbb{P}[\hat{h}(X) \neq h^*(X)] < \epsilon$ if and only if $\hat{h} = h^*$. Let i^* such that $h^* = h_{i^*}$.

First, a pool-based algorithm can achieve the required accuracy as follows: Let $j_t := i^* \bmod 2^t$ for $t \leq T$, and $j_t := \lfloor i^*/2^{T-t} \rfloor \bmod 2^T$ for $t \geq T$. If E holds, then t'th element selected by the pool algorithm is $a_{t,j}$, where j is obtained as follows: If $t \leq T$, $j = j_{t-1}$. If $t > T$, $j = \lfloor j_{t-1}/2 \rfloor$. In round 1, $j = 0$ and the selected element is $a_{1,0}$. Inductively, in this strategy the algorithm finds the t'th least significant bit in the binary expansion of i^* in round t, thus it can use j_{t-1} to set j for round t. Under E, after q labels i^* is identified exactly. This happens with a probability of $1 - \delta$ for any $\mathcal{D}$ with the uniform marginal $\mathcal{D}_X$.

Now, let $\mathcal{D}_h$ be a distribution with a uniform marginal over $\mathcal{X}$ with labels consistent with $h \in \mathcal{H}$. Consider a stream-based algorithm $\mathcal{A}_s$ which is equivalent to the pool algorithm. Denote its output by $\bar{h}$ and its input by $S \sim \mathcal{D}_{h^*}^\infty$. Let I be a random variable drawn uniformly at random from $\{0, \ldots, 2^q - 1\}$. Let $H = h_I$ be a hypothesis chosen uniformly at random from $\mathcal{H}$. Consider the probability space defined by $I, S \sim \mathcal{D}_H^\infty$, and the run of $\mathcal{A}_s$ on S. Let $(Z_1, Y_1), \ldots, (Z_q, Y_q)$ be the examples that $\mathcal{A}_s$ receives and the labels it gets, in order. Let $Y = (Y_1, \ldots, Y_q)$. Let $\alpha = \mathbb{P}[Z_1 = a_{1,0} \mid S_X]$. If $Z_1 = a_{1,0}$, then $\mathbb{P}[Y_1 = 0 \mid S_X] = \frac{1}{2}$. If $Z_1 \neq a_{1,0}$, then $\mathbb{P}[Y_1 = 0 \mid S_X] \geq 3/4$. Let $\mathbb{H}$ be the base-2 entropy, and $\mathbb{H}_b$ be the binary entropy. Then $\mathbb{H}_b(Y_1 \mid S_X) = \mathbb{H}_b((\alpha + 1)/4)$, and so

$$\begin{aligned} \mathbb{H}(H \mid Y, S_X) &= \mathbb{H}(H, Y \mid S_X) - \mathbb{H}(Y_1 \mid S_X) - \mathbb{H}(Y_1, \ldots, Y_q \mid Y_1, S_X) \\ &\geq q - \mathbb{H}_b((\alpha + 1)/4) - (q - 1) \\ &= 1 - \mathbb{H}_b((\alpha + 1)/4). \end{aligned}$$

From the Taylor expansion of the binary entropy around $1/2$, $\mathbb{H}_b(p) \leq 1 - (1 - 2p)^2/2$, therefore $\mathbb{H}(H \mid Y, S_X) \geq (1 - \alpha)^2/8$. Since $\mathcal{A}_s$ is equivalent to $\mathcal{A}_p^{\mathcal{U}}$, we have $\mathbb{P}[\bar{h} \neq H] \leq \delta$, hence $\mathbb{P}_{S_X}[\mathbb{P}[\bar{h} \neq H \mid S_X] \leq 2\delta] \geq \frac{1}{2}$. By Fano's inequality, for any S_X such that $\mathbb{P}[\bar{h} \neq H \mid S_X] \leq 2\delta$,

$$(1 - \alpha)^2/8 \leq \mathbb{H}(H \mid Y, S_X) \leq \mathbb{H}_b(2\delta) + 2\delta q \leq 2\delta(\log_2(\frac{1}{2\delta}) + 2 + q).$$

Where the last inequality follows from $\mathbb{H}_b(p) \leq p \log_2(1/p) + 2p$. From the definition of δ, we have $\delta \leq |\mathcal{X}| \exp(-m/n)$. Setting $T = \lceil \log_2(q) \rceil$, and noting that $|\mathcal{X}| \leq q2^T/2 \leq q^2$ and so $m \geq n \log(128q^3|\mathcal{X}|)$, we have $\delta \leq \frac{1}{128q^3}$. Therefore, for $q \geq 22$, $1 - \alpha \leq \frac{1}{2q}$. It follows that $\mathbb{P}_{S_X}[\mathbb{P}[Z_1 \neq a_{1,0} \mid S_X] \leq \frac{1}{2q}] \geq 1/2$.

Now, the same argument holds for any round t conditioned on $I \bmod 2^t = 0$ and $Z_1 = a_{1,0}, \ldots, Z_t = a_{t,0}$, since in this case after t labels, the algorithm has $q - t$ queries left, and needs to select from $\mathcal{H}'$, which is equivalent to $\mathcal{H}$, with $q - t$ instead of q. Moreover, $\mathbb{P}[\bar{h} = H \mid I \bmod 2^t = 0] \leq 1 - \delta$ as well, since this holds for every H individually. We conclude that for every $t \leq q$, with a probability at least $\frac{1}{2}$ over S_X,

$$\mathbb{P}[Z_t \neq a_{t,0} \mid S_X, H = h_0] \leq \frac{1}{2q}.$$

It follows that with a probability at least $\frac{1}{2}$ over S_X, $\mathbb{P}[Z_1 = a_{1,0}, \ldots, Z_q = a_{q,0} \mid S_X, H = h_0] \geq 1/2$. Hence $\mathbb{P}[Z_1 = a_{1,0}, \ldots, Z_q = a_{q,0} \mid H = h_0] \geq 1/4$.

Finally, suppose $H = h_0$. Let W_t be the number of elements that $\mathcal{A}_s$ observes after selecting element $t - 1$, until observing the next element. Let $X_j \sim \mathcal{D}_X$ be the j'th element observed after selecting the first $t - 1$ elements, and let $B_j = \mathbb{I}[X_j = a_{t,0}]$. B_j are independent Bernoulli random variables with $\mathbb{P}[B_j = 1] = 1/n$, and $\mathbb{P}[B_{W_t} = 1] \geq \mathbb{P}[E] = \beta \geq \frac{1}{4}$. By Lemma 2, if $\frac{1}{n} \leq \frac{1}{512}$, then $\mathbb{E}[W_t] \geq \frac{n}{64}$. It follows that the expected number of iterations over q selections is at least $\frac{qn}{64}$ if $n \geq 512$. Since it is also at least q, a lower bound of $\frac{qn}{512}$ holds for all n. ■

This lower bound shows that a gap between the stream-based and the pool-based settings exists not only for general interactive algorithms, but also specifically for active learning for binary classification. The gap is the most significant when $q = \tilde{\Theta}(\sqrt{m})$, in which case the stream algorithm requires $\tilde{\Omega}(m^{3/2})$ unlabeled examples, compared to only m for the pool algorithm. It has been previously observed (Gonen et al., 2013) that in some cases, the ALuMA algorithm, which is a pool-based active learning algorithm for halfspaces, is superior to the classical stream-based algorithm CAL (Cohn et al., 1994). Theorem 16 shows that this is not a limitation specifically of CAL, but of any stream-based active learning algorithm. On the other hand, Theorem 15 shows that this gap cannot be very large, at least for self-certifying active learning algorithms.

7. Conclusions

In this work we studied the relationship between the stream-based and the pool-based interactive settings, by designing algorithms that emulate pool-based behavior in a stream-based setting, and proving upper and lower bounds on the stream sizes required for such emulation. Our results concern mostly the case where the budget of the stream algorithm is similar or identical to that of the pool algorithm. We expect that as the budget grows, there should be a smooth improvement in

the expected stream length, which should approach m as the budget approaches m. It is an open problem to quantify this tradeoff in the various settings we considered.

Acknowledgments

This work was supported by the Israel Science Foundation (grant No. 555/15).

Appendix A. Proof of Lemma 2

The proof of Lemma 2, defined in Section 2, is provided below.

Proof [of Lemma 2] $\mathbb{E}[I]$ is minimized under the constraint when $\mathbb{P}[X_i = 1] = p$. Therefore assume this equality holds. Let W be the random variable whose value is the smallest integer such that $X_W = 1$. Let T be the largest integer such that $\mathbb{P}[W \leq T] \leq \alpha$.

The expectation of I is subject to $\mathbb{P}[X_I = 1] \geq \alpha$ is smallest if I is defined as follows: $\mathbb{P}[I = W \mid W \leq T] = 1$, $\mathbb{P}[I = W \mid W = T + 1] = \alpha - \mathbb{P}[W \leq T]$, and in all other cases, $I = 0$. Therefore,

$$\mathbb{E}[I] \geq \mathbb{E}[W \cdot \mathbb{I}[W \leq T]].$$

We have

$$\begin{aligned}\frac{1}{p} = \mathbb{E}[W] &= \mathbb{E}[W \cdot \mathbb{I}[W \leq T]] + \mathbb{E}[W \cdot \mathbb{I}[W > T]] \\ &= \mathbb{E}[W \cdot \mathbb{I}[W \leq T]] + (\frac{1}{p} + T)(1-p)^T.\end{aligned}$$

Therefore

$$\mathbb{E}[I] \geq \mathbb{E}[W \cdot \mathbb{I}[W \leq T]] = \frac{1}{p} - (\frac{1}{p} + T)(1-p)^T.$$

From the definition of T, T is the largest integer such that $1 - (1-p)^T \leq \alpha$. Hence $1 - \alpha \leq \exp(-pT)$, so $T \leq \frac{\log(1/(1-\alpha))}{p}$. In addition, since $1 - (1-p)^{T+1} \geq \alpha$, we have $(1-p)^T \leq (1-\alpha)/(1-p)$. Therefore

$$\mathbb{E}[I] \geq \frac{1}{p} - \left(\frac{1}{p} + \frac{\log(1/(1-\alpha))}{p}\right)\frac{1-\alpha}{1-p} = \frac{1}{p}\left(1 - \frac{(1-\alpha)(1+\log(1/(1-\alpha)))}{1-p}\right).$$

To show that $\mathbb{E}[I] \geq \alpha^2/(4p)$, for $p \leq \alpha^4/2$ and $\alpha \in (0,1)$, we show that

$$1 - \frac{(1-\alpha)(1+\log(1/(1-\alpha)))}{1-p} \geq \alpha^2/4. \tag{10}$$

It suffices to show that

$$1 - \alpha^2/4 - \frac{(1-\alpha)(1+\log(1/(1-\alpha)))}{1-\alpha^4/2} \geq 0.$$

Multiplying the LHS by $1 - \alpha^4/2$, we have

$$\begin{aligned}A(\alpha) &:= (1-\alpha^2/4)(1-\alpha^4/2) - (1-\alpha)(1+\log(1/(1-\alpha))) \\ &= \alpha - \alpha^2/4 - \alpha^4/2 + \alpha^6/8 - (1-\alpha)\log(1/(1-\alpha)).\end{aligned}$$

By the Taylor expansion of the natural logarithm around $1 - \alpha$,

$$(1-\alpha)\log(1/(1-\alpha)) = \sum_{n=1}^{\infty}\frac{\alpha^n}{n} - \sum_{n=1}^{\infty}\frac{\alpha^{n+1}}{n} = \alpha + \sum_{n=2}^{\infty}\alpha^n(\frac{1}{n} - \frac{1}{n-1}).$$

Therefore

$$\begin{aligned}A(\alpha) &= \alpha - \alpha^2/4 - \alpha^4/2 + \alpha^6/8 - \alpha + \sum_{n=2}^{\infty}\alpha^n(\frac{1}{n-1} - \frac{1}{n}) \\ &= \sum_{n=2}^{\infty}\alpha^n(\frac{1}{n-1} - \frac{1}{n}) - \alpha^2/4 - \alpha^4/2 + \alpha^6/8 \\ &= \sum_{n=3}^{\infty}\alpha^n(\frac{1}{n-1} - \frac{1}{n}) + (\alpha^2/4 - \alpha^4/2) + \alpha^6/4 \geq 0.\end{aligned}$$

It follows that $A(\alpha)/(1-\alpha^4/2) \geq 0$, implying Eq. (10), and concluding that $\mathbb{E}[I] \geq \alpha^2/(4p)$. ■

References

Alessandro Arlotto, Elchanan Mossel, and J Michael Steele. Quickest online selection of an increasing subsequence of specified size. *Random Structures & Algorithms*, 49(2):235–252, 2016.

M.-F. Balcan and Phil Long. Active and passive learning of linear separators under log-concave distributions. In *Proceedings of the Twenty-Sixth Annual Conference on Computational Learning Theory (COLT)*, pages 288–316, 2013.

M.-F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. *Journal of Computer and System Sciences*, 75(1):78–89, 2009.

Maria-Florina Balcan, Steve Hanneke, and Jennifer Wortman Vaughan. The true sample complexity of active learning. *Machine Learning*, 80(2-3):111–139, 2010.

M. Bouguelia, Y. Belaid, and A. Belaïd. A stream-based semi-supervised active learning approach for document classification. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 611–615. IEEE, 2013.

N. Cebron and M. R. Berthold. Active learning for object classification: from exploration to exploitation. *Data Mining and Knowledge Discovery*, 18(2):283–299, 2009.

W. Chu, M. Zinkevich, L. Li, A. Thomas, and B. Tseng. Unbiased online active learning in data streams. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 195–203. ACM, 2011.

D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15:201–221, 1994.

N.V. Cuong, W.S. Lee, and N. Ye. Near-optimal adaptive pool-based active learning with general loss. In *30th conference on Uncertainty in Artificial Intelligence (UAI)*, pages 122–131, 2014.

S. Dasgupta. Analysis of a greedy active learning strategy. *Advances in Neural Information Processing Systems 18 (NIPS)*, 17:337–344, 2005.

S. Dasgupta. Coarse sample complexity bounds for active learning. *Advances in Neural Information Processing Systems 19 (NIPS)*, 18:235, 2006.

Sanjoy Dasgupta. Consistency of nearest neighbor classification under selective sampling. In *Proceedings of the 25th Annual Conference on Learning Theory (COLT)*, pages 18.1–18.15, 2012.

E. D. Demaine, P. Indyk, S. Mahabadi, and A. Vakilian. On streaming and communication complexity of the set cover problem. In *Distributed Computing*, pages 484–498. Springer, 2014.

A. Deshpande, L. Hellerstein, and D. Kletenik. Approximation algorithms for stochastic boolean function evaluation and stochastic submodular set cover. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1453–1467. SIAM, 2014.

E. B. Dynkin. The optimum choice of the instant for stopping a markov process. In *Soviet Math. Dokl*, volume 4, pages 627–629, 1963.

T. S. Ferguson. Who solved the secretary problem? *Statistical Science*, 4(3):282–289, 1989.

Satoru Fujishige. *Submodular functions and optimization*, volume 58. Elsevier, 2005.

J. P. Gilbert and F. Mosteller. Recognizing the maximum of a sequence. *Journal of the American Statistical Association*, 61(313):35–73, 1966.

D. Golovin and A. Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42:427–486, 2011.

D. Golovin, M. Faulkner, and A. Krause. Online distributed sensor selection. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 220–231. ACM, 2010a.

D. Golovin, A. Krause, and D. Ray. Near-optimal bayesian active learning with noisy observations. In *Advances in Neural Information Processing Systems 23 (NIPS)*, pages 766–774, 2010b.

A. Gonen, S. Sabato, and S. Shalev-Shwartz. Efficient active learning of halfspaces: an aggressive approach. *Journal of Machine Learning Research*, 14:2487–2519, 2013.

P. H. Gosselin and M. Cord. Active learning methods for interactive image retrieval. *IEEE Transactions on Image Processing*, 17(7):1200–1211, 2008.

A. Guillory and J. A. Bilmes. Interactive submodular set cover. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 415–422, 2010.

Y. Guo and R. Greiner. Optimistic active-learning using mutual information. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI)*, volume 7, pages 823–829, 2007.

Y. Guo, I. Silins, U. Stenius, and A. Korhonen. Active learning-based information structure analysis of full scientific articles and two applications for biomedical literature review. *Bioinformatics*, 29 (11):1440–1447, 2013.

S. Hanneke. Teaching dimension and the complexity of active learning. In *Proceedings of the Twentieth Annual Conference on Computational Learning Theory (COLT)*, 2007.

S. Hanneke. Rates of convergence in active learning. *The Annals of Statistics*, 39(1):333–361, 2011.

S. Hanneke and L. Yang. Minimax analysis of active learning. *Journal of Machine Learning Research*, 16:3487–3602, 2015.

V. Krishnamurthy. Algorithms for optimal scheduling and management of hidden markov model sensors. *IEEE Transactions on Signal Processing*, 50(6):1382–1397, 2002.

S. R. Kulkarni, S. K. Mitter, and J. N. Tsitsiklis. Active learning using arbitrary binary valued queries. *Machine Learning*, 11(1):23–35, 1993.

D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc., 1994.

C. C. Loy, T. M. Hospedales, T. Xiang, and S. Gong. Stream-based joint exploration-exploitation active learning. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1560–1567, June 2012.

A. K. McCallum and K. Nigam. Employing em and pool-based active learning for text classification. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML)*, pages 350–358, 1998.

P. Mitra, C.A. Murthy, and S. K. Pal. A probabilistic active support vector learning algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):413–418, 2004.

S. Sabato and R. Munos. Active regression by stratification. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27 (NIPS)*, pages 469–477, 2014.

S. Sabato, A D Sarwate, and N Srebro. Auditing: Active learning with outcome-dependent query costs. In *Advances in Neural Information Processing Systems 26*, pages 512–520, 2013.

H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on computational learning theory*, pages 287–294. ACM, 1992.

A. Singla, S. Tschiatschek, and A. Krause. Noisy submodular maximization via adaptive sampling with applications to crowdsourced image collection summarization. In *Conference on Artificial Intelligence (AAAI)*, 2016.

J. Smailović, M. Grčar, N. Lavrač, and M. Žnidaržič. Stream-based active learning for sentiment analysis in the financial domain. *Information Sciences*, 285(0):181 – 203, 2014.

M. Streeter and D. Golovin. An online algorithm for maximizing submodular functions. In *Advances in Neural Information Processing Systems 22 (NIPS)*, pages 1577–1584, 2009.

S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *Proceedings of the Ninth ACM International Conference on Multimedia*, MULTIMEDIA '01, pages 107–118. ACM, 2001.

S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research (JMLR)*, 2:45–66, 2002.

Journal of Machine Learning Research 18 (2018) 1-49 Submitted 10/16; Published 7/18

CoCoA: A General Framework for Communication-Efficient Distributed Optimization

Virginia Smith SMITHV@STANFORD.EDU
Department of Computer Science
Stanford University
Stanford, CA 94305, USA

Simone Forte SIMONE.FORTE@GESS.ETHZ.CH
Department of Computer Science
ETH Zürich
8006 Zürich, Switzerland

Chenxin Ma CHM514@LEHIGH.EDU
Martin Takáč TAKAC.MT@GMAIL.COM
Industrial and Systems Engineering Department
Lehigh University
Bethlehem, PA 18015, USA

Michael I. Jordan JORDAN@CS.BERKELEY.EDU
Division of Computer Science and Department of Statistics
University of California
Berkeley, CA 94720, USA

Martin Jaggi MARTIN.JAGGI@EPFL.CH
School of Computer and Communication Sciences
EPFL
1015 Lausanne, Switzerland

Editor: Yoram Singer

Abstract

The scale of modern datasets necessitates the development of efficient distributed optimization methods for machine learning. We present a general-purpose framework for distributed computing environments, CoCoA, that has an efficient communication scheme and is applicable to a wide variety of problems in machine learning and signal processing. We extend the framework to cover general non-strongly-convex regularizers, including L1-regularized problems like lasso, sparse logistic regression, and elastic net regularization, and show how earlier work can be derived as a special case. We provide convergence guarantees for the class of convex regularized loss minimization objectives, leveraging a novel approach in handling non-strongly-convex regularizers and non-smooth loss functions. The resulting framework has markedly improved performance over state-of-the-art methods, as we illustrate with an extensive set of experiments on real distributed datasets.

Keywords: Convex optimization, distributed systems, large-scale machine learning, parallel and distributed algorithms

1. Introduction

Distributed computing architectures have come to the fore in modern machine learning, in response to the challenges arising from a wide range of large-scale learning applications. Distributed architectures offer the promise of scalability by increasing both computational and storage capacities. A critical challenge in realizing this promise of scalability is to develop efficient methods for communicating and coordinating information between distributed machines, taking into account the specific needs of machine-learning algorithms.

On most distributed systems, the communication of data between machines is vastly more expensive than reading data from main memory and performing local computation. Moreover, the optimal trade-off between communication and computation can vary widely depending on the dataset being processed, the system being used, and the objective being optimized. It is therefore essential for distributed methods to accommodate flexible communication-computation profiles while still providing convergence guarantees.

Although numerous distributed optimization methods have been proposed, the mini-batch optimization approach has emerged as one of the most popular paradigms for tackling this communication-computation tradeoff (e.g., Dekel et al., 2012; Shalev-Shwartz and Zhang, 2013b; Shamir and Srebro, 2014; Qu et al., 2015; Richtárik and Takáč, 2016). Mini-batch methods are often developed by generalizing stochastic methods to process multiple data points at a time, which helps to alleviate the communication bottleneck by enabling more distributed computation per round of communication. However, while the need to reduce communication would suggest large mini-batch sizes, the theoretical convergence rates of these methods tend to degrade with increased mini-batch size, reverting to the rates of classical (batch) gradient methods. Empirical results corroborate these theoretical rates, and in practice, mini-batch methods have limited flexibility to adapt to the communication-computation tradeoffs that would maximally leverage parallel execution. Moreover, because mini-batch methods are typically derived from a specific single-machine solver, these methods and their associated analyses are often tailored to specific problem instances and can suffer both theoretically and practically when applied outside of their restricted setting.

In this work, we propose a framework, CoCoA[1], that addresses these two fundamental limitations. First, we allow arbitrary local solvers to be used on each machine in parallel. This allows the framework to directly incorporate state-of-the-art, application-specific single-machine solvers in the distributed setting. Second, the framework shares information between machines through a highly flexible communication scheme. This allows the amount of communication to be easily tailored to the problem and system at hand, in particular allowing for the case of significantly reduced communication in the distributed environment.

A key step in providing these features in the framework is to first define meaningful subproblems for each machine to solve in parallel, and to then combine updates from the subproblems in an efficient manner. Our method and convergence results rely on noting that, depending on the distribution of the data (e.g., by feature or by training point), and whether we solve the problem in the primal or the dual, certain machine learning objectives

1. CoCoA-v1 (Jaggi et al., 2014) and CoCoA$^+$ (Ma et al., 2017b, 2015a) are predecessors of this work. We continue to use the name CoCoA for the more general framework proposed here, and show how earlier work can be derived as a special case (Section 4). Portions of this newer work additionally appear in SF's master's thesis (Forte, 2015) and Smith et al. (2015).

can be more easily decomposed into subproblems in the distributed setting. In particular, we categorize common machine learning objectives into several cases, and use duality to help decompose these objectives. Using primal-dual information in this manner not only allows for efficient methods (achieving, e.g., up to 50x speedups compared to state-of-the-art), but also allows for strong primal-dual convergence guarantees and practical benefits such as computation of the duality gap for use as an accuracy certificate and stopping criterion.

1.1 Contributions

General framework. We develop a communication-efficient primal-dual framework that is applicable to a broad class of convex optimization problems. Notably, in contrast to earlier work of Yang (2013), Jaggi et al. (2014), Ma et al. (2017b) and Ma et al. (2015a), our generalized, cohesive framework: (1) specifically incorporates difficult cases of L_1 regularization and other non-strongly-convex regularizers; (2) allows for the flexibility of distributing the data by either feature or training point; and (3) can be run in either a primal or dual formulation, which we show to have significant theoretical and practical implications.

Flexible communication and local solvers. Two key advantages of the proposed framework are its communication efficiency and ability to employ off-the-shelf single-machine solvers internally. On real-world systems, the cost of communication versus computation can vary widely, and it is thus advantageous to permit a flexible amount of communication depending on the setting at hand. Our framework provides exactly such control. Moreover, we allow arbitrary solvers to be used on each machine, which permits the reuse of existing code and the benefits from multi-core or other optimizations therein. We note that beyond the selection of the local solver and communication vs. computation profile, there are no required hyperparameters to tune; the provided default parameters ensure convergence and are used throughout our experiments to achieve state-of-the-art performance.

Convergence guarantees. We derive convergence rates for the framework, guaranteeing, e.g., a $\mathcal{O}(1/t)$ rate of convergence in terms of communication rounds for convex objectives with Lipschitz continuous losses, and a faster linear rate for strongly convex losses. Importantly, our convergence guarantees do not degrade with the number of machines, K, and allow for subproblems to be solved to arbitrary accuracies, which allows for highly flexible computation vs. communication profiles. Additionally, we leverage a novel approach in the analysis of primal-dual rates for non-strongly-convex regularizers. The proposed technique is an improvement over simple smoothing techniques used in, e.g., Nesterov (2005), Shalev-Shwartz and Zhang (2014) and Zhang and Lin (2015) that enforce strong convexity by adding a small L_2 term to the objective. Our results include primal-dual rates and certificates for the general class of linear regularized loss minimization, and we show how earlier work can be derived as a special case of our more general approach.

Experimental comparison. The proposed framework yields order-of-magnitude speedups (as much as 50× faster) compared to state-of-the-art methods for large-scale machine learning. We demonstrate these gains with an extensive experimental comparison on real-world distributed datasets. We additionally explore properties of the framework itself, including the effect of running the framework in the primal or the dual, and the impact of subproblem accuracy on convergence. All algorithms for comparison are implemented in Apache Spark and run on Amazon EC2 clusters. Our code is available at: `gingsmith.github.io/cocoa/`.

2. Background and Setup

In this paper we develop a general framework for minimizing problems of the following form:

$$\ell(\mathbf{u}) + r(\mathbf{u})\,, \tag{I}$$

for convex functions ℓ and r. Frequently the term ℓ is a loss function, taking the form $\sum_i \ell_i(\mathbf{u})$, and the term r is a regularizer, e.g., $r(\mathbf{u}) = \lambda\|\mathbf{u}\|_p$. This formulation includes many popular methods in machine learning and signal processing, such as support vector machines, linear and logistic regression, lasso and sparse logistic regression, and many others.

2.1 Definitions

The following standard definitions will be used throughout the paper.

Definition 1 (L-Lipschitz Continuity). *A function $h : \mathbb{R}^m \to \mathbb{R}$ is* L-Lipschitz continuous *if $\forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^m$, we have*

$$|h(\mathbf{u}) - h(\mathbf{v})| \leq L\|\mathbf{u} - \mathbf{v}\|\,. \tag{1}$$

Definition 2 (L-Bounded Support). *A function $h : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ has* L-bounded support *if its effective domain is bounded by L, i.e.,*

$$h(\mathbf{u}) < +\infty \;\Rightarrow\; \|\mathbf{u}\| \leq L\,. \tag{2}$$

Definition 3 ($(1/\mu)$-Smoothness). *A function $h : \mathbb{R}^m \to \mathbb{R}$ is* $(1/\mu)$-smooth *if it is differentiable and its derivative is $(1/\mu)$-Lipschitz continuous, or equivalently*

$$h(\mathbf{u}) \leq h(\mathbf{v}) + \langle \nabla h(\mathbf{v}), \mathbf{u} - \mathbf{v}\rangle + \frac{1}{2\mu}\|\mathbf{u} - \mathbf{v}\|^2 \qquad \forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^m\,. \tag{3}$$

Definition 4 (μ-Strong Convexity). *A function $h : \mathbb{R}^m \to \mathbb{R}$ is* μ-strongly convex *for $\mu \geq 0$ if*

$$h(\mathbf{u}) \geq h(\mathbf{v}) + \langle s, \mathbf{u} - \mathbf{v}\rangle + \frac{\mu}{2}\|\mathbf{u} - \mathbf{v}\|^2 \qquad \forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^m\,, \tag{4}$$

for any $s \in \partial h(\mathbf{v})$, where $\partial h(\mathbf{v})$ denotes the subdifferential of h at $\mathbf{v}$.

2.2 Primal-Dual Setting

Numerous methods have been proposed to solve (I), and these methods generally fall into two categories: *primal methods*, which run directly on the primal objective, and *dual methods*, which instead run on the dual formulation of the primal objective. In developing our framework, we present an abstraction that allows for either a primal or a dual variant of our framework to be run. In particular, to solve the input problem (I), we consider mapping the problem to one of the following two general problems:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \quad \left[\; \mathcal{O}_A(\boldsymbol{\alpha}) := \quad f(A\boldsymbol{\alpha}) \;+\; g(\boldsymbol{\alpha}) \qquad\qquad \right] \tag{A}$$

$$\min_{\mathbf{w} \in \mathbb{R}^m} \quad \left[\; \mathcal{O}_B(\mathbf{w}) := \quad f^*(\mathbf{w}) \;+\; g^*(-A^\top \mathbf{w}) \qquad \right]. \tag{B}$$

In general, our aim will be to compute a minimizer of the problem (A) in a distributed fashion; the main difference will be whether we initially map the primal (I) to (A) or (B).

Here $\boldsymbol{\alpha} \in \mathbb{R}^n$ and $\mathbf{w} \in \mathbb{R}^m$ are parameter vectors, $A := [\mathbf{x}_1; \ldots; \mathbf{x}_n] \in \mathbb{R}^{m \times n}$ is a data matrix with column vectors $\mathbf{x}_i \in \mathbb{R}^m$, $i \in \{1, \ldots, n\}$, and the functions f^* and g_i^* are the *convex conjugates* of f and g_i, respectively.

The dual relationship between problems (A) and (B) is known as Fenchel-Rockafellar duality (Borwein and Zhu, 2005, Theorem 4.4.2). We provide a self-contained derivation of the duality in Appendix B. Note that while dual problems are typically presented as a pair of (min, max) problems, we have equivalently reformulated (A) and (B) to both be minimization problems in accordance with their roles in our framework.

Given $\boldsymbol{\alpha} \in \mathbb{R}^n$ in the context of (A), a corresponding vector $\mathbf{w} \in \mathbb{R}^m$ for problem (B) is obtained by:

$$\mathbf{w} = \mathbf{w}(\boldsymbol{\alpha}) := \nabla f(A\boldsymbol{\alpha}) \,. \tag{5}$$

This mapping arises from first-order optimality conditions on the f-part of the objective. The duality gap, given by:

$$G(\boldsymbol{\alpha}) := \mathcal{O}_A(\boldsymbol{\alpha}) - [-\mathcal{O}_B(\mathbf{w}(\boldsymbol{\alpha}))] \,, \tag{6}$$

is always non-negative, and under strong duality, the gap will reach zero only for an optimal pair $(\boldsymbol{\alpha}^\star, \mathbf{w}^\star)$. The duality gap at any point provides a practically computable upper bound on the unknown primal as well as dual optimization error (suboptimality), since

$$\mathcal{O}_A(\boldsymbol{\alpha}) \geq \mathcal{O}_A(\boldsymbol{\alpha}^\star) \geq -\mathcal{O}_B(\mathbf{w}^\star) \geq -\mathcal{O}_B(\mathbf{w}(\boldsymbol{\alpha})) \;.$$

In developing the proposed framework, noting the duality between (A) and (B) has many benefits, including the ability to compute the duality gap, which acts as a certificate of the approximation quality. It is also useful as an analysis tool, helping us to present a cohesive framework and relate this work to the prior work of Yang (2013), Jaggi et al. (2014) and Ma et al. (2015a, 2017b). As a word of caution, note that we avoid prescribing the name "primal" or "dual" directly to either of the problems (A) or (B), as we demonstrate below that their role as primal or dual can change depending on the application problem of interest.

2.3 Assumptions and Problem Cases

Assumptions. Our main assumptions on problem (A) are that f is $(1/\tau)$-smooth, and the function g is separable, i.e., $g(\boldsymbol{\alpha}) = \sum_i g_i(\alpha_i)$, with each g_i having L-bounded support. Given the duality between the problems (A) and (B), this can be equivalently stated as assuming that in problem (B), f^* is τ-strongly convex, and the function $g^*(-A^\top \mathbf{w}) = \sum_i g_i^*(-\mathbf{x}_i^\top \mathbf{w})$ is separable with each g_i^* being L-Lipschitz.

Problem cases. Suppose, as in equation (I), we would like to find a minimizer of the general objective $\ell(\mathbf{u}) + r(\mathbf{u})$. Depending on the smoothness of the function ℓ and the strong convexity of the function r, we will be able to map the input function (I) to one (or both) of the objectives (A) and (B) based on our assumptions. In particular, we outline three separate cases: Case I, in which the function ℓ is smooth and the function r is strongly convex; case II, in which ℓ is smooth, and r is non-strongly convex and separable; and case III, in which ℓ is non-smooth and separable, and r is strongly convex. These cases are summarized in Table 1. Note that the union of these cases captures most commonly-used applications of linear regularized loss minimization problems.

In Section 3, we will see that different variants of the framework may be realized depending on which of these three cases we consider when solving the input problem (I).

Table 1: Criteria for Objectives (A) and (B).

	Smooth ℓ	Non-smooth, separable ℓ
Strongly convex r	Case I: Obj (A) or (B)	Case III: Obj (B)
Non-strongly convex, separable r	Case II: Obj (A)	–

2.4 Running Examples

To illustrate the cases in Table 1, we consider several examples below. *Those interested in details of the framework itself may skip to Section 3.* These applications will serve as running examples throughout the paper, and we will revisit them in our experiments (Section 6). For further applications and details, see Section 5.

1. *Elastic Net Regression (Case I: map to either (A) or (B)).* We can map elastic-net regularized least squares regression,

$$\min_{\mathbf{u} \in \mathbb{R}^p} \; \tfrac{1}{2}\|A\mathbf{u} - \mathbf{b}\|_2^2 + \eta\lambda\|\mathbf{u}\|_1 + (1-\eta)\frac{\lambda}{2}\|\mathbf{u}\|_2^2 \,, \tag{7}$$

   to either objective (A) or (B). To map to objective (A), we let: $f(A\boldsymbol{\alpha}) = \frac{1}{2}\|A\boldsymbol{\alpha} - \mathbf{b}\|_2^2$ and $g(\boldsymbol{\alpha}) = \sum_i g_i(\alpha_i) = \sum_i \eta\lambda|\alpha_i| + (1-\eta)\frac{\lambda}{2}\alpha_i^2$, setting n to be the number of features and m the number of training points. To map to (B), we let: $g(-A^\top \mathbf{w}) = \sum_i g_i^*(-\mathbf{x}_i^\top \mathbf{w}) = \sum_i \frac{1}{2}(\mathbf{x}_i^\top \mathbf{w} - b_i)^2$ and $f^*(\mathbf{w}) = \eta\lambda\|\mathbf{w}\|_1 + (1-\eta)\frac{\lambda}{2}\|\mathbf{w}\|_2^2$, setting m to be the number of features and n the number of training points. We discuss in Section 3 how the choice of mapping to either (A) or to (B) can have implications on the distribution scheme and overall performance of the framework.

2. *Lasso (Case II: map to (A)).* We can represent L_1-regularized least squares regression by mapping the model:

$$\min_{\mathbf{u} \in \mathbb{R}^p} \; \tfrac{1}{2}\|A\mathbf{u} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{u}\|_1 \tag{8}$$

   to objective (A), letting $f(A\boldsymbol{\alpha}) = \frac{1}{2}\|A\boldsymbol{\alpha} - \mathbf{b}\|_2^2$ and $g(\boldsymbol{\alpha}) = \sum_i g_i(\alpha_i) = \sum_i \lambda|\alpha_i|$. In this mapping, n represents the number of features, and m the number of training points. Note that we cannot map the lasso objective to (B) directly, as f^* must be τ-strongly convex and the L_1-norm is non-strongly convex.

3. *Support Vector Machine (Case III: map to (B)).* We can represent a hinge loss support vector machine (SVM) by mapping the model:

$$\min_{\mathbf{u} \in \mathbb{R}^p} \; \frac{1}{m}\sum_{i=1}^m \max\left\{0, 1 - y_i(\mathbf{x}_i^\top \mathbf{u})\right\} + \tfrac{\lambda}{2}\|\mathbf{u}\|_2^2 \,, \tag{9}$$

   to objective (B), letting $g^*(-A^\top \mathbf{w}) = \sum_i g_i^*(-\mathbf{x}_i^\top \mathbf{w}) = \sum_i \frac{1}{n}\max\{0, 1 - y_i\mathbf{x}_i^\top \mathbf{w}\}$ and $f^*(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|_2^2$. In this mapping, m represents the number of features, and n the number of training points. Note that we cannot map the hinge loss SVM primal to objective (A) directly, as f must be $(1/\tau)$-smooth and the hinge loss is non-smooth.

2.5 Data Partitioning

In this work, we are interested in the setting where the dataset at hand is distributed across multiple machines. We assume that the dataset A is distributed over K machines according to a partition $\{\mathcal{P}_k\}_{k=1}^K$ of the *columns* of $A \in \mathbb{R}^{m\times n}$. We denote the size of the partition on machine k by $n_k = |\mathcal{P}_k|$. For machine $k \in \{1,\dots,K\}$ and weight vector $\boldsymbol{\alpha} \in \mathbb{R}^n$, we define $\boldsymbol{\alpha}_{[k]} \in \mathbb{R}^n$ as the n-vector with elements $(\boldsymbol{\alpha}_{[k]})_i := \alpha_i$ if $i \in \mathcal{P}_k$ and $(\boldsymbol{\alpha}_{[k]})_i := 0$ otherwise. Analogously, we write $A_{[k]}$ for the corresponding group of columns of A, and zeros elsewhere (note that columns can correspond to either training examples or features, depending on the application). We discuss these distribution schemes in greater detail in Section 3.

3. The CoCoA Method

In the following sections, we describe the proposed framework, CoCoA, at a high level, and then discuss two approaches for using the framework in practice: CoCoA in the primal, where we consider (A) to be the primal objective and run the framework on this problem directly, and CoCoA in the dual, where we instead consider (B) to be the primal objective, and then run the framework on the dual (A).

Note that in both approaches, the aim will be to compute a minimizer of the problem (A) in a distributed fashion; the main difference will be whether we view (A) as the primal objective or as the dual objective.

3.1 The Generalized Framework

The goal of the CoCoA framework is to find a global minimizer of the objective (A), while distributing computation based on the partitioning of the dataset A across machines (Section 2.5). As a first step, note that distributing the update to the function g in objective (A) is straightforward, as we have required that this term is separable according to the partitioning of our data, i.e., $g(\boldsymbol{\alpha}) = \sum_{i=1}^n g_i(\alpha_i)$. However, the same does not hold for the term $f(A\boldsymbol{\alpha})$. To minimize this part of the objective in a distributed fashion, we propose minimizing a quadratic approximation of the function, which allows the minimization to separate across machines. We make this approximation precise in the following subsection.

Data-local quadratic subproblems. In the general CoCoA framework (Algorithm 1), we distribute computation by defining a data-local subproblem of the optimization problem (A) for each machine. This simpler problem can be solved on machine k and only requires accessing data which is already available locally, i.e., the columns $A_{[k]}$. More formally, each machine k is assigned the following local subproblem, which depends only on the previous shared vector $\mathbf{v} := A\boldsymbol{\alpha} \in \mathbb{R}^m$, and the local data $A_{[k]}$:

$$\min_{\Delta\boldsymbol{\alpha}_{[k]}\in\mathbb{R}^n} \ \mathcal{G}_k^{\sigma'}(\Delta\boldsymbol{\alpha}_{[k]}; \mathbf{v}, \boldsymbol{\alpha}_{[k]})\,, \tag{10}$$

where

$$\mathcal{G}_k^{\sigma'}(\Delta\boldsymbol{\alpha}_{[k]}; \mathbf{v}, \boldsymbol{\alpha}_{[k]}) := \frac{1}{K} f(\mathbf{v}) + \mathbf{w}^\top A_{[k]}\Delta\boldsymbol{\alpha}_{[k]} + \frac{\sigma'}{2\tau}\left\|A_{[k]}\Delta\boldsymbol{\alpha}_{[k]}\right\|^2 + \sum_{i\in\mathcal{P}_k} g_i(\alpha_i + \Delta\boldsymbol{\alpha}_{[k]_i}),$$

Algorithm 1 Generalized CoCoA Distributed Framework

Input: Data matrix A distributed column-wise according to partition $\{\mathcal{P}_k\}_{k=1}^K$, aggregation parameter $\gamma \in (0,1]$, and parameter σ' for the local subproblems $\mathcal{G}_k^{\sigma'}(\Delta\boldsymbol{\alpha}_{[k]}; \mathbf{v}, \boldsymbol{\alpha}_{[k]})$. Starting point $\boldsymbol{\alpha}^{(0)} := \mathbf{0} \in \mathbb{R}^n$, $\mathbf{v}^{(0)} := \mathbf{0} \in \mathbb{R}^m$.
for $t = 0, 1, 2, \dots$ **do**
  for $k \in \{1, 2, \dots, K\}$ **in parallel over computers do**
    call local solver, returning a Θ-approximate solution $\Delta\boldsymbol{\alpha}_{[k]}$ of the local subproblem (10)
    update local variables $\boldsymbol{\alpha}_{[k]}^{(t+1)} := \boldsymbol{\alpha}_{[k]}^{(t)} + \gamma\,\Delta\boldsymbol{\alpha}_{[k]}$
    return updates to shared state $\Delta\mathbf{v}_k := A_{[k]}\Delta\boldsymbol{\alpha}_{[k]}$
  end for
  reduce $\mathbf{v}^{(t+1)} := \mathbf{v}^{(t)} + \gamma\sum_{k=1}^K \Delta\mathbf{v}_k$
end for

and $\mathbf{w} := \nabla f(\mathbf{v})$. Here we let $\Delta\boldsymbol{\alpha}_{[k]}$ denote the change of local variables α_i for indices $i \in \mathcal{P}_k$, and we set $(\Delta\boldsymbol{\alpha}_{[k]})_i := 0$ for all $i \notin \mathcal{P}_k$. It is important to note that the subproblem (10) is simple in the sense that it is always a quadratic objective (apart from the g_i term). The subproblem does not depend on the function f itself, but only its linearization at the fixed shared vector $\mathbf{v}$. This property additionally simplifies the task of the local solver, especially for cases of complex functions f.

Framework parameters γ and σ'. There are two parameters that must be set in the framework: γ, the aggregation parameter, which controls how the updates from each machine are combined, and σ', the subproblem parameter, which is a data-dependent term measuring the difficulty of the data partitioning $\{\mathcal{P}_k\}_{k=1}^K$. These terms play a crucial role in the convergence of the method, as we demonstrate in Section 4. In practice, we provide a simple and robust way to set these parameters: For a given aggregation parameter $\gamma \in (0,1]$, the subproblem parameter σ' will be set as $\sigma' := \gamma K$, but can also be improved in a data-dependent way as we discuss below. *In general, as we show in Section 4, setting $\gamma := 1$ and $\sigma' := K$ will guarantee convergence while delivering our fastest convergence rates.*

Definition 5 (Data-dependent aggregation parameter)**.** *In Algorithm 1, the aggregation parameter γ controls the level of adding ($\gamma := 1$) versus averaging ($\gamma := \frac{1}{K}$) of the partial solutions from all machines. For our convergence results (Section 4) to hold, the subproblem parameter σ' must be chosen not smaller than*

$$\sigma' \geq \sigma'_{min} := \gamma \max_{\boldsymbol{\alpha}\in\mathbb{R}^n} \frac{\|A\boldsymbol{\alpha}\|^2}{\sum_{k=1}^K \|A_{[k]}\boldsymbol{\alpha}_{[k]}\|^2}\,. \tag{11}$$

The simple choice of $\sigma' := \gamma K$ is valid for (11), i.e.,

$$\gamma K \geq \sigma'_{min}\,.$$

In some cases, it will be possible to give a better (data-dependent) choice for σ', closer to the actual bound given in σ'_{min}.

Subproblem interpretation. Here we provide further intuition behind the data-local subproblems (10). The local objective functions $\mathcal{G}_k^{\sigma'}$ are defined to closely approximate the global objective in (A) as the "local" variable $\Delta\boldsymbol{\alpha}_{[k]}$ varies, which we will see in the analysis (Appendix D, Lemma 1). In fact, if the subproblem were solved exactly, this could be interpreted as a data-dependent, block-separable proximal step, applied to the f part of the objective (A) as follows:

$$\sum_{k=1}^{K} \mathcal{G}_k^{\sigma'}(\Delta\boldsymbol{\alpha}_{[k]}; \mathbf{v}, \boldsymbol{\alpha}_{[k]}) = R + f(\mathbf{v}) + \nabla f(\mathbf{v})^\top A\Delta\boldsymbol{\alpha} + \frac{\sigma'}{2\tau}\Delta\boldsymbol{\alpha}^\top \begin{bmatrix} A_{[1]}^\top A_{[1]} & & 0 \\ & \ddots & \\ 0 & & A_{[K]}^\top A_{[K]} \end{bmatrix} \Delta\boldsymbol{\alpha}, \quad (12)$$

where $R = \sum_{i\in[n]} g_i(-\alpha_i - \Delta\alpha_i)$.

However, note that in contrast to traditional proximal methods, CoCoA does *not* assume that this subproblem is solved to high accuracy, as we instead allow the use of local solvers of any approximation quality Θ.

Reusability of existing single-machine solvers. The local subproblems (10) have the appealing property of being very similar in structure to the global problem (A), with the main difference being that they are defined on a smaller (local) subset of the data, and are simpler because they are not dependent on the shape of f. For a user of CoCoA, this presents a significant advantage in that existing single machine-solvers can be directly re-used in our distributed framework (Algorithm 1) by employing them on the subproblems $\mathcal{G}_k^{\sigma'}$.

Therefore, problem-specific tuned solvers which have already been developed, along with associated speed improvements (such as multi-core implementations), can be easily leveraged in the distributed setting. We quantify the dependence on local solver performance with the following assumption and remark, and relate this performance to our global convergence rates in Section 4.

Assumption 1 (Θ-approximate solution)**.** *We assume that there exists $\Theta \in [0, 1)$ such that $\forall k \in [K]$, the local solver at any outer iteration t produces a (possibly) randomized approximate solution $\Delta\boldsymbol{\alpha}_{[k]}$, which satisfies*

$$\frac{\mathbb{E}\left[\mathcal{G}_k^{\sigma'}(\Delta\boldsymbol{\alpha}_{[k]}; \mathbf{v}, \boldsymbol{\alpha}_{[k]}) - \mathcal{G}_k^{\sigma'}(\Delta\boldsymbol{\alpha}_{[k]}^\star; \mathbf{v}, \boldsymbol{\alpha}_{[k]})\right]}{\mathcal{G}_k^{\sigma'}(\ \mathbf{0}\ ; \mathbf{v}, \boldsymbol{\alpha}_{[k]}) - \mathcal{G}_k^{\sigma'}(\Delta\boldsymbol{\alpha}_{[k]}^\star; \mathbf{v}, \boldsymbol{\alpha}_{[k]})} \leq \Theta\,, \quad (13)$$

where

$$\Delta\boldsymbol{\alpha}_{[k]}^\star \in \arg\min_{\Delta\boldsymbol{\alpha}\in\mathbb{R}^n} \mathcal{G}_k^{\sigma'}(\Delta\boldsymbol{\alpha}_{[k]}; \mathbf{v}, \boldsymbol{\alpha}_{[k]}), \quad \forall k \in [K]\,. \quad (14)$$

Remark 1. *In practice, the time spent solving the local subproblems in parallel should be chosen comparable to the time of a communication round, for best overall efficiency on a given system. We study this trade-off in theory (Section 4) and experiments (Section 6).*

Remark 2. *Note that the accuracy parameter Θ does not have to be chosen* a priori*: Our convergence results (Section 4) are valid if Θ is an upper bound on the actual empirical values Θ in Algorithm 1. This allows for some of the K machines to at times deliver better or worse accuracy Θ (e.g., this would allow a slow local machine to be stopped early during a specific round in order to avoid stragglers). See (Smith et al., 2017) for more details.*

Remark 3. *From a theoretical perspective, the multiplicative notion of accuracy is advantageous over classical additive accuracy as existing convergence results for first- and second-order optimization methods typically appear in multiplicative form, i.e., relative to the error at the initialization point (here $\Delta\boldsymbol{\alpha}_{[k]} = \mathbf{0}$). This accuracy notion Θ is also useful beyond the distributed setting (see, e.g., Karimireddy et al., 2018a,b). We discuss local solvers and associated rates to achieve accuracy Θ for particular applications in Section 5.*

With this general framework in place, we next discuss two variants of our framework, CoCoA-Primal and CoCoA-Dual. In running either the primal or dual variant of the framework, the goal will always be to solve objective (A) in a distributed fashion. The main difference will be whether this objective is viewed as the primal or dual of the input problem (I). We make this mapping technique precise and discuss its implications in the following subsections (Sections 3.2–3.4).

3.2 Primal Distributed Optimization

In the primal distributed version of the framework (Algorithm 2), the framework is run by mapping the initial problem (I) directly to objective (A) and then applying the generalized CoCoA framework described in Algorithm 1. In other words, we view problem (A) as the primal objective, and solve this problem directly.

From a theoretical perspective, viewing (A) as the primal will allow us to consider non-strongly convex regularizers, since we allow the terms g_i to be non-strongly convex. This setting was not covered in earlier work of Yang (2013); Jaggi et al. (2014); Ma et al. (2015a); and Ma et al. (2017b), and we discuss it in detail in Section 4, as additional machinery must be introduced to develop primal-dual rates for this setting.

Running the primal version of the framework has important practical implications in the distributed setting, as it typically implies that the data is distributed by feature rather than by training point. In this setting, the amount of communication at every outer iteration will be $\mathcal{O}(\#$ of training points). When the number of features is high (as is common when using sparsity-inducing regularizers) this can help to reduce communication and improve overall performance, as we demonstrate in Section 6.

Algorithm 2 CoCoA-Primal (Mapping Problem (I) to (A))

1: **Map:** Input problem (I) to objective (A)
2: **Distribute:** Dataset A by columns (here typically features) according to partition $\{\mathcal{P}_k\}_{k=1}^K$
3: **Run:** Algorithm 1 with aggregation parameter γ and subproblem parameter σ'

3.3 Dual Distributed Optimization

In the dual distributed version of the framework (Algorithm 3), we run the framework by mapping the original problem (I) to objective (B), and then solve the problem by running Algorithm 1 on the dual (A). In other words, we view problem (B) as the primal, and solve this problem via the dual (A).

This version of the framework will allow us to consider non-smooth losses, such as the hinge loss or absolute deviation loss, since the terms g_i^* can be non-smooth. From a practical perspective, this version of the framework will typically imply that the data is distributed by training point, and for a vector $\mathcal{O}(\#$ of features) to be communicated at every outer iteration. This variant may therefore be preferable when the number of training points exceeds the number of features.

Algorithm 3 CoCoA-Dual (Mapping Problem (I) to (B))

1: **Map:** Input problem (I) to objective (B)
2: **Distribute:** Dataset A by columns (here typically training points) according to partition $\{\mathcal{P}_k\}_{k=1}^K$
3: **Run:** Algorithm 1 with aggregation parameter γ and subproblem parameter σ'

3.4 Primal vs. Dual

In Table 2, we revisit the three cases from Section 2, showing how the primal and dual variants of CoCoA can be applied to various input problems $\ell(\mathbf{u}) + r(\mathbf{u})$, depending on properties of the functions ℓ and r. In particular, in the setting where ℓ is smooth and r is strongly convex, the user may choose whether to run the framework in the primal (Algorithm 2), or in the dual (Algorithm 3).

Intuitively, Algorithm 2 will be preferable as r loses strong convexity, and Algorithm 3 will be preferable as ℓ loses smoothness. However, there are also systems-related aspects to consider. In Algorithm 2, we typically distribute the data by feature, and in Algorithm 3, by training point (this distribution depends on how the terms n and m are defined in our mapping; see Section 5). Depending on whether the number of features or number of training points is the dominating term, we may chose to run Algorithm 2 or Algorithm 3, respectively, in order to reduce communication costs. We validate these ideas empirically in Section 6 by comparing the performance of each variant (primal vs. dual) on real distributed datasets.

Table 2: Criteria for Running Algorithms 2 vs. 3.

	Smooth ℓ	Non-smooth and separable ℓ
Strongly convex r	Case I: Alg. 2 or 3	Case III: Alg. 3
Non-strongly convex and separable r	Case II: Alg. 2	–

In the following subsection, we provide greater insight into the CoCoA framework and its relation to prior work. An extended discussion on related work is available in Section 7.

3.5 Interpretations of CoCoA

There are numerous methods that have been developed to solve (A) and (B) in parallel and distributed environments. We describe related work in detail in Section 7, and here briefly position CoCoA and in relation to other widely-used parallel and distributed methods.

CoCoA in the context of classical parallelization schemes. We first contrast CoCoA with common distributed mini-batch and batch methods, such as mini-batch stochastic gradient descent or coordinate descent, gradient descent, and quasi-Newton methods.

CoCoA is similar to these methods in that they are all *iterative*, i.e., they make progress towards the optimal solution by updating the parameter vector $\boldsymbol{\alpha}$ according to some function h: $\mathbb{R}^n \to \mathbb{R}^n$ at each iteration t: $\boldsymbol{\alpha}^{(t+1)} = h(\boldsymbol{\alpha}^{(t)})$, $t = 0, 1, \ldots$, until convergence is reached. From a coordinate-wise perspective, two approaches to update $\boldsymbol{\alpha}$ iteratively include the Jacobi "all-at-once" and Gauss-Seidel "one-at-a-time" methods (Bersekas and Tsitsiklis, 1989):

$$\text{Jacobi:} \quad \alpha_i^{(t+1)} = h_i(\alpha_1^{(t)}, \ldots, \alpha_n^{(t)}), \quad i = 1, \ldots, n,$$
$$\text{Gauss-Seidel:} \quad \alpha_i^{(t+1)} = h_i(\alpha_1^{(t+1)}, \ldots, \alpha_{i-1}^{(t+1)}, \alpha_i^{(t)}, \ldots, \alpha_n^{(t)}), \quad i = 1, \ldots, n.$$

The Jacobi method does not require information from the other coordinates to update coordinate i, which makes this style of method well-suited for parallelization. However, the sequential Gauss-Seidel-style method tends to converge faster in terms of iterations, as it is able to incorporate information from the updates of other coordinates more quickly. This difference is well-known and evident in single machine solvers, where stochastic methods (benefiting from fresh updates) tend to outperform their batch counterparts.

Typical mini-batch methods, e.g., mini-batch coordinate descent, perform a Jacobi-style update on a subset of the coordinates at each iteration. This makes these methods amenable to high levels of parallelization. However, they are unable to incorporate information as quickly as their serial counterparts in terms of number of data points accessed, as synchronization is required before updating the coordinates. As the size of the mini-batch grows, this can increase the runtime and even lead to divergence (Richtárik and Takáč, 2016).

CoCoA instead aims to combine attractive properties of both of these update paradigms. In CoCoA, Jacobi-style updates are applied in parallel to *blocks* of the coordinates of $\boldsymbol{\alpha}$ to distribute the method, while allowing for (though not necessarily requiring) faster Gauss-Seidel-style updates on each machine. This change in parallelization scheme is one of the key reasons for improved performance over simpler mini-batch or batch style methods.

Two extremes: from distributed CD to one-shot communication. In addition to the parallel block-Jacobi updating scheme described above, CoCoA incorporates an additional level of flexibility by allowing for an *arbitrary number* of sequential Gauss-Seidel iterations (or any local solver for that matter) to be performed locally on each machine. This flexibility is critical in the distributed setting, as one of the key indicators of parallel efficiency is the time spent on local computation vs. communication. In particular, the flexibility to solve each subproblem to arbitrary accuracy, Θ, allows CoCoA to scale from low-communication environments, where more iterations can be performed before communicating, to high communication environments, where fewer local iterations are necessary.

In comparison with other distributed methods, this flexibility also affords an explanation of CoCoA as a method that can freely move between two extremes. On one extreme, if the subproblems (10) are solved exactly, CoCoA recovers block coordinate descent, where the coordinate updates are applied as part of a block-separable proximal step (12). If only one outer round of communication is performed, this is similar in spirit to one-shot communication schemes, which attempt to completely solve for and then combine locally-computed models (see, e.g., Mann et al., 2009; Zhang et al., 2013; Heinze et al., 2016). While

these one-shot communication schemes are ideal in terms of reducing communication, they are, in contrast to CoCoA, generally not guaranteed to converge to the optimal solution.

On the other extreme, if just a single update (i.e., with respect to one coordinate α_i) is performed at each communication round, this recovers traditional distributed coordinate descent. In comparison to CoCoA, vanilla distributed coordinate descent can suffer from a high communication bottleneck due to the low relative amount of local computation. Even in the case of mini-batch coordinate descent, the most amount of work that can be performed locally at each round includes a single pass through the data, whereas CoCoA has the flexibility to take multiple passes. We empirically compare to mini-batch distributed coordinate descent in Section 6 to demonstrate the effect of this issue in practice.

Comparison to ADMM. Finally, we provide a direct comparison between CoCoA and the alternating direction method of multipliers (ADMM), a well-established framework for distributed optimization (Boyd et al., 2010). Similar to CoCoA, ADMM defines a subproblem for each machine to solve in parallel, rather than parallelizing a mini-batch update. ADMM also leverages duality structure, similar to that presented in Section 2. For consensus ADMM (Mota et al., 2013), (B) is decomposed with a re-parameterization:

$$\max_{\mathbf{w}_1,\dots\mathbf{w}_K,\mathbf{w}} \quad \sum_{k=1}^{K}\sum_{i\in\mathcal{P}_k} g^*(-\mathbf{x}_i^\top\mathbf{w}_k) + f^*(\mathbf{w}) \quad \text{s.t. } \mathbf{w}_k = \mathbf{w},\ k=1,\dots,K.$$

This problem is then solved by constructing the augmented Lagrangian, which yields the following decomposable updates:

$$\begin{aligned}
\mathbf{w}_k^{(t+1)} &= \arg\min_{\mathbf{w}_k} \sum_{i\in\mathcal{P}_k} g_i^*(-\mathbf{x}_i^\top\mathbf{w}_k) + \rho\mathbf{u}_k^{(t)^\top}(\mathbf{w}_k - \mathbf{w}^{(t)}) + \frac{\rho}{2}\|\mathbf{w}_k - \mathbf{w}^{(t)}\|^2,\\
\mathbf{w}^{(t+1)} &= \arg\min_{\mathbf{w}} \ f^*(\mathbf{w}) + \frac{\rho K}{2}\|\mathbf{w} - (\bar{\mathbf{w}}_k^{(t+1)} + \bar{\mathbf{u}}_k^{(t)})\|^2,\\
\mathbf{u}_k^{(t+1)} &= \mathbf{u}_k^{(t)} + \mathbf{w}_k^{(t+1)} - \mathbf{w}^{(t+1)},
\end{aligned}$$

where ρ is a penalty parameter that must be tuned for best performance. It can be shown that the update to $\mathbf{w}_k$ can be reformulated in terms of the conjugate functions g_i as:

$$\arg\min_{\boldsymbol{\alpha}_{[k]}} \sum_{i\in\mathcal{P}_k} g_i(\boldsymbol{\alpha}_{[k]_i}) + (\mathbf{w}^{(t)} - \mathbf{u}_k^{(t)})^\top A_{[k]}\boldsymbol{\alpha}_{[k]} + \frac{1}{2\rho}\|A_{[k]}\boldsymbol{\alpha}_{[k]}\|^2\,. \tag{15}$$

Thus, we see that the update to $\mathbf{w}_k$ closely matches the CoCoA subproblem (10), where $\rho := \frac{\tau}{\sigma'}$. This is intuitive as both methods use proximal steps to derive the subproblem, and a similar result can be shown when applying ADMM to the (A) formulation, which can be seen as an instantiation of the sharing variant of ADMM (Boyd et al., 2010, Section 7.3).

However, there remain major differences between the methods despite this connection. First, CoCoA has a more direct and simplified scheme for updating the global weight vector $\mathbf{w}$, as the additional proximal step is not required. Second, in CoCoA, there is no need to tune any parameters such as ρ, as the method can be run simply using $\sigma'{=}K$. Finally, in the CoCoA method and theory, the subproblem can be solved approximately to any accuracy Θ, rather than requiring a full batch update as in ADMM. We will see in our experiments that these differences have a substantial impact in practice (Section 6). We provide a full derivation of the comparison to ADMM for reference in Appendix C.

4. Convergence Analysis

In this section, we provide convergence rates for the proposed framework and introduce a key theoretical technique in analyzing non-strongly convex terms in the primal-dual setting.

For simplicity of presentation, we assume in the analysis that the data partitioning is balanced, i.e., $n_k = n/K$ for all k. Furthermore, we assume that the columns of A satisfy $\|\mathbf{x}_i\| \le 1$ for all $i \in [n]$, and $\mathcal{O}_B$ contains the average term $\frac{1}{n}\sum_{i=1}^n g_i^*(\cdot)$, as is common in ERM-type problems. We present rates for the case where $\gamma := 1$ in Algorithm 1, and where the subproblems (10) are defined using the corresponding safe bound $\sigma' := K$. This case will guarantee convergence while delivering our fastest rates in the distributed setting, which in particular do not degrade as the number of machines K increases and n remains fixed. More general rates and all proof details can be found in the appendix.

4.1 Proof Strategy: Relating Subproblem Approximation to Global Progress

To guarantee convergence, it is critical to show how progress made on the local subproblems (10) relates to the global objective $\mathcal{O}_A$. Our first lemma provides exactly this information. In particular, we see that if the aggregation and subproblem parameters are selected according to Definition 5, the sum of the subproblem objectives, $\sum_{k=1}^K \mathcal{G}_k^{\sigma'}$, will form a block-separable upper bound on the global objective $\mathcal{O}_A$.

Lemma 1. *For any weight vector $\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha} \in \mathbb{R}^n$, $\mathbf{v} = \mathbf{v}(\boldsymbol{\alpha}) := A\boldsymbol{\alpha}$, and real values γ, σ' satisfying* (11)*, it holds that*

$$\mathcal{O}_A\Big(\boldsymbol{\alpha} + \gamma\sum_{k=1}^K \Delta\boldsymbol{\alpha}_{[k]}\Big) \le (1-\gamma)\mathcal{O}_A(\boldsymbol{\alpha}) + \gamma\sum_{k=1}^K \mathcal{G}_k^{\sigma'}(\Delta\boldsymbol{\alpha}_{[k]}; \mathbf{v}, \boldsymbol{\alpha}_{[k]})\,. \tag{16}$$

A proof of Lemma 1 is provided in Appendix D. We use this main lemma, in combination with our measure of quality of the subproblem approximations (Assumption 1), to deliver global convergence rates.

4.2 Rates for General Convex g_i, L-Lipschitz g_i^*

Our first main theorem provides convergence guarantees for objectives with general convex g_i (or, equivalently, L-Lipschitz g_i^*), including models with non-strongly convex regularizers such as lasso and sparse logistic regression, or models with non-smooth losses, such as the hinge loss support vector machine.

Theorem 2. *Consider Algorithm 1 with $\gamma := 1$, and let Θ be the quality of the local solver as in Assumption 1. Let g_i have L-bounded support, and let f be $(1/\tau)$-smooth. Then after T iterations, where*

$$\begin{aligned}
T &\ge T_0 + \max\{\Big\lceil\frac{1}{1-\Theta}\Big\rceil, \frac{4L^2}{\tau\epsilon_G(1-\Theta)}\}\,, \\
T_0 &\ge t_0 + \Big\lceil\frac{2}{1-\Theta}\left(\frac{8L^2}{\tau\epsilon_G} - 1\right)\Big\rceil_+\,, \\
t_0 &\ge \max(0, \Big\lceil\tfrac{1}{(1-\Theta)}\log\Big(\tfrac{\tau n(\mathcal{O}_A(\boldsymbol{\alpha}^{(0)}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star))}{2L^2K}\Big)\Big\rceil)\,,
\end{aligned} \tag{17}$$

we have that the expected duality gap satisfies

$$\mathbb{E}\big[\mathcal{O}_A(\overline{\boldsymbol{\alpha}}) - (-\mathcal{O}_B(\mathbf{w}(\overline{\boldsymbol{\alpha}})))\big] \leq \epsilon_G\,,$$

where $\overline{\boldsymbol{\alpha}}$ *is the averaged iterate:* $\frac{1}{T-T_0}\sum_{t=T_0+1}^{T-1}\boldsymbol{\alpha}^{(t)}$.

Providing primal-dual rates and globally defined primal-dual accuracy certificates for these objectives may require a theoretical technique that we introduce below, in which we show how to satisfy the notion of L-bounded support for g_i, as stated in Definition 2.

4.2.1 Bounded support modification

Additional work is necessary if Theorem 2 is to be applied to non-strongly convex regularizers such as the L_1 norm, which do not have L-bounded support for each g_i, and thus violate the main assumptions. Note for example that the conjugate function of $g_i = |\cdot|$, which is the indicator function of an interval, is not defined globally over $\mathbb{R}$, and thus (without further modification) the duality gap $G(\boldsymbol{\alpha})$:=$\mathcal{O}_A(\boldsymbol{\alpha})$-(-$\mathcal{O}_B(\mathbf{w}(\boldsymbol{\alpha}))$) is not defined at all points $\boldsymbol{\alpha}$.

Smoothing. To address this problem, existing approaches typically use a simple smoothing technique (e.g., Nesterov, 2005; Shalev-Shwartz and Zhang, 2014): by adding a small amount of L_2 regularization, the functions g_i become strongly convex. Following this change, the methods are run on the dual instead of the original primal problem. While this modification satisfies the necessary assumptions for convergence, this smoothing technique is often undesirable in practice, as it changes the iterates, the algorithms at hand, the convergence rate, and the tightness of the resulting duality gap compared to the original objective. Further, the amount of smoothing can be difficult to tune and has a large impact on empirical performance. We perform experiments to highlight these issues in practice in Section 6.

Bounded support modification. In contrast to smoothing, our approach preserves all solutions of the original objective, leaves the iterate sequence unchanged, and allows for direct reusability of existing solvers for the original g_i objectives (such as L_1 solvers). It also removes the need for tuning a smoothing parameter. To achieve this, we modify the function g_i by imposing an additional weak constraint that is inactive in our region of interest. Formally, we replace $g_i(\alpha_i)$ by the following modified function:

$$\bar{g}_i(\alpha_i) := \begin{cases} g_i(\alpha_i) & : \alpha_i \in [-B, B] \\ +\infty & : \text{otherwise.} \end{cases} \tag{18}$$

For large enough B, this problem yields the same solution as the original objective. Note also that this only affects convergence theory, in that it allows us to present a strong primal-dual rate (Theorem 2 for L=B). The modification of g_i does not affect the algorithms for the original problems. Whenever a monotone optimizer is used, we will never leave the level set defined by the objective at the starting point.

Using the resulting modified function will allow us to apply the results of Theorem 2 for general convex functions g_i. This technique can also be thought of as "Lipschitzing" the dual g_i^*, because of the general result that g_i^* is L-Lipschitz if and only if g_i has L-bounded support (Rockafellar, 1997, Corollary 13.3.3). We derive the conjugate function $\bar{g}_i^*$ for completeness in Appendix B (Lemma 6). In Section 5, we show how to leverage this technique for a variety of application input problems. See also Dünner et al. (2016) for a follow-up discussion of this technique in the non-distributed case.

4.3 Rates for Strongly Convex g_i, Smooth g_i^*

For the case of objectives with strongly convex g_i (or, equivalently, smooth g_i^*), e.g., elastic net regression or logistic regression, we obtain the following faster *linear* convergence rate.

Theorem 3. *Consider Algorithm 1 with* $\gamma := 1$*, and let* Θ *be the quality of the local solver as in Assumption 1. Let* g_i *be* μ*-strongly convex* $\forall i \in [n]$*, and let* f *be* $(1/\tau)$*-smooth. Then after* T *iterations where*

$$T \geq \tfrac{1}{(1-\Theta)}\tfrac{\mu\tau+1}{\mu\tau}\log\tfrac{1}{\epsilon_{\mathcal{O}_A}}\,, \tag{19}$$

it holds that

$$\mathbb{E}\big[\mathcal{O}_A(\boldsymbol{\alpha}^{(T)}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star)\big] \leq \epsilon_{\mathcal{O}_A}\,.$$

Furthermore, after T *iterations with*

$$T \geq \tfrac{1}{(1-\Theta)}\tfrac{\mu\tau+1}{\mu\tau}\log\Big(\tfrac{1}{(1-\Theta)}\tfrac{\mu\tau+1}{\mu\tau}\tfrac{1}{\epsilon_G}\Big)\,,$$

we have the expected duality gap

$$\mathbb{E}\big[\mathcal{O}_A(\boldsymbol{\alpha}^{(T)}) - (-\mathcal{O}_B(\mathbf{w}(\boldsymbol{\alpha}^{(T)}))\big] \leq \epsilon_G\,.$$

We provide proofs of both Theorem 2 and Theorem 3 in Appendix D.

4.4 Convergence Cases

Revisiting Table 1 from Section 2, we summarize our convergence guarantees for the three cases of input problems (I) in the following table. In particular, we see that for cases II and III, we obtain a sublinear convergence rate, whereas for case I we can obtain a faster linear rate, as provided in Theorem 3.

Table 3: Applications of Convergence Rates.

	Smooth ℓ	Non-smooth, separable ℓ
Strongly convex r	Case I: Theorem 3	Case III: Theorem 2
Non-strongly convex, separable r	Case II: Theorem 2	–

4.5 Recovering Earlier Work as a Special Case

As a special case, the proposed framework and rates directly apply to L_2-regularized loss-minimization problems, including those presented in the earlier work of Jaggi et al. (2014) and Ma et al. (2015a).

Remark 4. *If we run Algorithm 3 (mapping* (I) *to* (B)*) and restrict* $f^*(\cdot) := \frac{\lambda}{2}\|\cdot\|^2$ *(so that* $\tau = \lambda$*), Theorem 2 recovers as a special case the* CoCoA$^+$ *rates for general L-Lipschitz* ℓ_i^* *losses (see Ma et al., 2015a, Corollary 9). The earlier work of* CoCoA*-v1 (Jaggi et al., 2014) did not provide rates for L-Lipschitz* ℓ_i^* *losses.*

Remark 5. *If we run Algorithm 3 (mapping* (I) *to* (B)*) and restrict* $f^*(\cdot) := \frac{\lambda}{2}\|\cdot\|^2$ *(so that* $\tau = \lambda$*), Theorem 3 recovers as a special case the* CoCoA$^+$ *rates for* $(1/\ell_i^*)$*-smooth losses (see Ma et al., 2015a, Corollary 11). The earlier rates of* CoCoA*-v1 can be obtained by setting* $\gamma{:=}\frac{1}{K}$ *and* $\sigma'{=}1$ *in Algorithm 1 (Jaggi et al., 2014, Theorem 2).*

These cases follow since g_i^* is L-Lipschitz if and only if g_i has L-bounded support (Rockafellar, 1997, Corollary 13.3.3), and g_i^* is μ-strongly convex if and only if g_i is $(1/\mu)$-smooth (Hiriart-Urruty and Lemaréchal, 2001, Theorem 4.2.2).

5. Applications

In this section we detail example applications that can be cast within the general CoCoA framework. For each example, we describe the primal-dual setup and algorithmic details, discuss the convergence properties our framework for the application, and include practical concerns such as information on state-of-the-art local solvers. We discuss examples according to the three cases defined in Table 1 of Section 2 for finding a minimizer of the general objective $\ell(\mathbf{u}) + r(\mathbf{u})$, and provide a summary of these common examples in Table 4.

Table 4: Common Losses and Regularizers.

(i) Losses

Loss	Obj	f / g^*
Least Squares	(A)	$f{=}\frac{1}{2}\|A\boldsymbol{\alpha} - \mathbf{b}\|_2^2$
	(B)	$g^*{=}\frac{1}{2}\|A^\top\mathbf{w} - \mathbf{b}\|_2^2$
Logistic Reg.	(A)	$f{=}\frac{1}{m}\sum_j \log(1{+}\exp(b_j\mathbf{x}_j^\top\boldsymbol{\alpha}))$
	(B)	$g^*{=}\frac{1}{n}\sum_i \log(1{+}\exp(b_i\mathbf{x}_i^\top\mathbf{w}))$
SVM	(B)	$g^*{=}\frac{1}{n}\sum_i \max(0, 1{-}y_i\mathbf{x}_i^\top\mathbf{w})$
Absolute Dev.	(B)	$g^* = \frac{1}{n}\sum_i \|\mathbf{x}_i^\top\mathbf{w} - y_i\|$

(ii) Regularizers

Regularizer	Obj	g / f^*
Elastic Net	(A)	$g{=}\lambda(\eta\|\boldsymbol{\alpha}\|_1{+}\frac{1-\eta}{2}\|\boldsymbol{\alpha}\|_2^2)$
	(B)	$f^*{=}\lambda(\eta\|\mathbf{w}\|_1{+}\frac{1-\eta}{2}\|\mathbf{w}\|_2^2)$
L_2	(A)	$g{=}\frac{\lambda}{2}\|\boldsymbol{\alpha}\|_2^2$
	(B)	$f^*{=}\frac{\lambda}{2}\|\mathbf{w}\|_2^2$
L_1	(A)	$g{=}\lambda\|\boldsymbol{\alpha}\|_1$
Group Lasso	(A)	$g{=}\lambda\sum_p\|\boldsymbol{\alpha}_{\mathcal{I}_p}\|_2$, $\mathcal{I}_p \subseteq [n]$

5.1 Case I: Smooth ℓ, Strongly convex r

For input problems (I) with smooth ℓ and strongly convex r, Theorem 3 from Section 4 gives a global linear (geometric) convergence rate. Smooth loss functions can be mapped either to the function f in objective (A), or g^* in (B). Similarly, strongly convex regularizers can be mapped either to function g in objective (A), or f^* in (B). To illustrate the role of f as a smooth loss function and g as a strongly convex regularizer in objective (A), contrasting with their traditional roles in prior work (Yang, 2013; Jaggi et al., 2014; Ma et al., 2015a, 2017b), we consider the following examples. Note that mapping to objective (B) instead will follow trivially assuming that the loss is separable across training points (see Table 4).

For the examples in this subsection, we use m to represent the number of training points and n the number of features. Note that these definitions may change in the following subsections: this flexibility is useful so that we can present both the primal and dual variations of our framework (Algorithms 2 and 3) via a single abstract method (Algorithm 1).

Smooth ℓ: least squares loss. Let $\mathbf{b} \in \mathbb{R}^m$ be labels or response values, and consider the least squares objective, $f(\mathbf{v}) := \frac{1}{2}\|\mathbf{v} - \mathbf{b}\|_2^2$, which is 1-smooth. We obtain the familiar least-squares regression objective in our optimization problem (A), using

$$f(A\boldsymbol{\alpha}) := \tfrac{1}{2}\|A\boldsymbol{\alpha} - \mathbf{b}\|_2^2 \,. \tag{20}$$

Observing that the gradient of f is $\nabla f(\mathbf{v}) = \mathbf{v}{-}\mathbf{b}$, the primal-dual mapping is given by: $\mathbf{w}(\boldsymbol{\alpha}) := A\boldsymbol{\alpha}{-}\mathbf{b}$, which is well known as the *residual vector* in least-squares regression.

Smooth ℓ: logistic regression loss. For classification, we consider a logistic regression model with m training examples, $\mathbf{y}_j \in \mathbb{R}^n$ for $j \in [m]$, collected as the rows of the data matrix A. For each training example, we are given a binary label, which we collect in the vector $\mathbf{b} \in \{-1, 1\}^m$. Formally, the objective is defined as $f(\mathbf{v}) := \sum_{j=1}^m \log\left(1 + \exp\left(-b_j v_j\right)\right)$, which is again a separable function. The classifier loss is given by

$$f(A\boldsymbol{\alpha}) := \sum_{j=1}^m \log\left(1 + \exp\left(-b_j \mathbf{y}_j^\top\boldsymbol{\alpha}\right)\right), \tag{21}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^n$ is the parameter vector. It is not hard to show that f is 1-smooth if the labels satisfy $b_j \in [-1, 1]$. The primal-dual mapping is given by $w_j(\boldsymbol{\alpha}) := \frac{-b_j}{1+\exp\left(b_j \mathbf{y}_j^\top\boldsymbol{\alpha}\right)}$.

Strongly convex r: elastic net regularizer. An application we can consider for a strongly convex regularizer, g in (A) or f^* in (B), is elastic net regularization, $\eta\lambda\|\mathbf{u}\|_1 + (1-\eta)\frac{\lambda}{2}\|\mathbf{u}\|_2^2$, for fixed parameter $\eta \in (0, 1]$. This can be obtained in (A) by setting

$$g(\boldsymbol{\alpha}) = \sum_{i=1}^n g_i(\alpha_i) := \sum_{i=1}^n \eta\lambda|\alpha_i| + (1-\eta)\tfrac{\lambda}{2}\alpha_i^2. \tag{22}$$

For the special case $\eta = 1$, we obtain the L_1-norm, and for $\eta = 0$, we obtain the L_2-norm. The conjugate of g_i is given by: $g_i^*(x) := \frac{1}{2(1-\eta)}\left(\left[|x| - \eta\right]_+\right)^2$, where $[.]_+$ is the positive part operator, $[s]_+ = s$ for $s > 0$, and zero otherwise.

5.2 Case II: Smooth ℓ, Non-Strongly Convex Separable r

In case II, we consider mapping the input problem (I) to objective (A), where ℓ is assumed to be smooth, and r non-strongly convex and separable. For smooth losses in (A), we can consider as examples those provided in Subsection 5.1, e.g., the least squares loss or logistic loss. For an example of a non-strongly convex regularizer, we consider the important case of L_1 regularization below. Again, we note that this application cannot be realized by objective (B), where it is assumed that the regularization term f^* is strongly convex.

Non-strongly convex r: L_1 regularizer. L_1 regularization is obtained in objective (A) by letting $g_i(\cdot) := \lambda|\cdot|$. However, an additional modification is necessary to obtain primal-dual convergence and certificates for this setting. In particular, we employ the modification introduced in Section 4, which will guarantee L-bounded support. Formally, we replace $g_i(\cdot) = |\cdot|$ by

$$\bar{g}(\alpha) := \begin{cases} |\alpha| & : \alpha \in [-B, B], \\ +\infty & : \text{otherwise.} \end{cases}$$

For large enough B, this problem yields the same solution as the original L_1-regularized objective. Note that this only affects convergence theory, in that it allows us to present a strong primal-dual rate (Theorem 2 for L=B). With this modified L_1 regularizer, the optimization problem (A) with regularization parameter λ becomes

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \; f(A\boldsymbol{\alpha}) + \lambda \sum_{i=1}^{n} \bar{g}(\alpha_i)\,. \tag{23}$$

For large enough choice of the value B, this problems yields the same solution as the original objective: $f(A\boldsymbol{\alpha}) + \lambda\sum_{i=1}^{n}|\alpha_i|$. The modified $\bar{g}$ is simply a constrained version of the absolute value to the interval $[-B, B]$. Therefore by setting B to a large enough value that the values of α_i will never reach it, $\bar{g}^*$ will be continuous and at the same time make (23) equivalent to the original objective.

Formally, a simple way to obtain a large enough value of B, so that all solutions are unaffected, is the following: If we start the algorithm at $\boldsymbol{\alpha} = \mathbf{0}$, for every point encountered during execution of a monotone optimizer, the objective values will never become worse than $\mathcal{O}_A(\mathbf{0})$. Formally, under the assumption that f is non-negative, we will have that (for each i):

$$\lambda|\alpha_i| \leq f(\mathbf{0}) = \mathcal{O}_A(\mathbf{0}) \implies |\alpha_i| \leq \frac{f(\mathbf{0})}{\lambda}.$$

We can therefore safely set the value of B as $\frac{f(\mathbf{0})}{\lambda}$. For the modified $\bar{g}_i$, the conjugate $\bar{g}_i^*$ is given by:

$$\bar{g}_i^*(x) := \begin{cases} 0 & : x \in [-1, 1], \\ B(|x| - 1) & : \text{otherwise.} \end{cases}$$

We provide a proof of this in Appendix B (Lemma 6).

Non-strongly convex r: group lasso. The group lasso penalty can be mapped to objective (A), with:

$$g(\boldsymbol{\alpha}) := \lambda \sum_{p=1}^{P} \|\boldsymbol{\alpha}_{\mathcal{I}_p}\|_2 \quad \text{with} \quad \bigcup_{p=1}^{P} \mathcal{I}_p = \{1, \ldots, n\}\,, \tag{24}$$

where the disjoint sets $\mathcal{I}_p \subseteq \{1, \ldots, n\}$ represent a partitioning of the total set of variables. This penalty can be viewed as an intermediate between a pure L_1 or L_2 penalty, performing variable selection only at the group level. The term $\boldsymbol{\alpha}_{\mathcal{I}_p} \in \mathbb{R}^{|\mathcal{I}_p|}$ denotes part of the vector $\boldsymbol{\alpha}$ with indices $\mathcal{I}_p$. The conjugate is given by:

$$g^*(\mathbf{w}) = I_{\{\mathbf{w} \mid \max_{\mathcal{I}_p \in [n]} \|\boldsymbol{\alpha}_{\mathcal{I}_p}\|_2 \leq \lambda\}}(\mathbf{w}).$$

For details, see, e.g., Dünner et al. (2016) or Boyd and Vandenberghe (2004, Example 3.26).

5.3 Case III: Non-Smooth Separable ℓ, Strongly Convex r

Finally, in case III, we consider mapping the input problem (I) to objective (B), where ℓ is assumed to be non-smooth and separable, and r strongly convex. We discuss two common cases of general non-smooth losses ℓ, including the the hinge loss for classification and

absolute deviation loss for regression. When paired with a strongly convex regularizer, the regularizer via f gives rise to the primal-dual mapping, and Theorem 2 provides a sublinear convergence rate for objectives of this form. We note that these losses cannot be realized directly by objective (A), where it is assumed that the loss term f is smooth.

Non-smooth ℓ: hinge loss. For classification problems, we can consider a hinge loss support vector machine model, on n training points in $\mathbb{R}^m$, given with the loss:

$$g^*(-A^\top \mathbf{w}) = \sum_{i=1}^{n} g_i^*(-\mathbf{x}_i^\top \mathbf{w}) := \frac{1}{n}\sum_{i=1}^{n} \max\{0, 1 - y_i \mathbf{x}_i^\top \mathbf{w}\}. \tag{25}$$

The conjugate function of the hinge loss $\phi(a) = \max\{0, 1 - b\}$ is given by $\phi^*(b) = \{b$ if $b \in [-1, 0]$, else $\infty\}$. When using the L_2 norm for regularization in this problem: $f^*(\mathbf{w}) := \lambda\|\mathbf{w}\|_2^2$, a primal-dual mapping is given by: $\mathbf{w}(\boldsymbol{\alpha}) := \frac{1}{\lambda n} A\boldsymbol{\alpha}$.

Non-smooth ℓ: absolute deviation loss. The absolute deviation loss, used, e.g., in quantile regression or least absolute deviation regression, can be realized in objective (B) by setting:

$$g^*(-A^\top \mathbf{w}) = \sum_{i=1}^{n} g_i^*(-\mathbf{x}_i^\top \mathbf{w}) := \frac{1}{n}\sum_{i=1}^{n} \left|\mathbf{x}_i^\top \mathbf{w} - y_i\right|. \tag{26}$$

The conjugate function of the absolute deviation loss $\phi(a) = |a - y_i|$ is given by $\phi^*(-b) = -by_i$, with $b \in [-1, 1]$.

5.4 Local Solvers

As discussed in Section 3, the subproblems solved on each machine in the CoCoA framework are appealing in that they are very similar in structure to the global problem (A), with the main difference being that they are defined on a smaller (local) subset of the data, and have a simpler dependence on the term f. Therefore, solvers which have already proven their value in the single machine or multicore setting can be easily leveraged within the framework. We discuss some specific examples of local solvers below, and point the reader to Ma et al. (2017b) for an empirical exploration of these choices.

Local solvers for Algorithm 2. In the primal setting (Algorithm 2), the local subproblem (10) becomes a simple quadratic problem on the local data, with regularization applied only to local variables $\boldsymbol{\alpha}_{[k]}$. For the L_1-regularized examples discussed, existing fast L_1-solvers for the single-machine case, such as GLMNET variants (Friedman et al., 2010) or BLITZ (Johnson and Guestrin, 2015) can be directly applied to each local subproblem $\mathcal{G}_k^{\sigma'}(\,\cdot\,; \mathbf{v}, \boldsymbol{\alpha}_{[k]})$ within Algorithm 1. The sparsity induced on the subproblem solutions of each machine naturally translates into the sparsity of the global solution, since the local variables $\boldsymbol{\alpha}_{[k]}$ will be concatenated.

In terms of the approximation quality parameter Θ for the local problems (Assumption 1), we can apply existing recent convergence results from the single machine case. For example, for randomized coordinate descent (as part of GLMNET), Lu and Xiao (2013, Theorem 1) gives a $\mathcal{O}(1/t)$ approximation quality for any separable regularizer, including L_1 and elastic net; see also Tappenden et al. (2015) and Shalev-Shwartz and Tewari (2011).

Local solvers for Algorithm 3. In the dual setting (Algorithm 3) for the discussed examples, the losses are applied only to local variables $\boldsymbol{\alpha}_{[k]}$, and the regularizer is approximated via a quadratic term. Current state of the art for the problems of the form in (B) are variants of randomized coordinate ascent—Stochastic Dual Coordinate Ascent (SDCA) (Shalev-Shwartz and Zhang, 2013a). This algorithm and its variants are increasingly used in practice (Wright, 2015), and extensions such as accelerated and parallel versions can directly be applied (Shalev-Shwartz and Zhang, 2014; Fan et al., 2008) in our framework. For non-smooth losses such as SVMs, the analysis of Shalev-Shwartz and Zhang (2013a) provides a $\mathcal{O}(1/t)$ rate, and for smooth losses, a faster linear rate. There have also been recent efforts to derive a linear convergence rate for problems like the hinge-loss support vector machine that could be applied, e.g., by using error bound conditions (Necoara and Nedelcu, 2014; Wang and Lin, 2014), weak strong convexity conditions (Ma et al., 2015b; Necoara, 2015) or by considering Polyak-Łojasiewicz conditions (Karimi et al., 2016).

6. Experiments

In this section we demonstrate the empirical performance of CoCoA in the distributed setting. We first compare CoCoA to competing methods for two common machine learning applications: lasso regression (Section 6.1) and support vector machine (SVM) classification (Section 6.2). We then explore the performance of CoCoA in the primal versus the dual directly by solving an elastic net regression model with both variants (Section 6.3). Finally, we illustrate general properties of the CoCoA method empirically in Section 6.4.

Experimental setup. We compare CoCoA to numerous state-of-the-art general-purpose methods for large-scale optimization, including:

- Mb-SGD: Mini-batch stochastic gradient. For our experiments with lasso, we compare against Mb-SGD with an L_1-prox.
- GD: Full gradient descent. For lasso we use the proximal version, Prox-GD.
- L-BFGS: Limited-memory quasi-Newton method. For lasso, we use OWL-QN (orthant-wise limited quasi-Newton).
- ADMM: Alternating direction method of multipliers. We use conjugate gradient internally for the lasso experiments, and SDCA for SVM experiments.
- Mb-CD: Mini-batch parallel coordinate descent. For SVM experiments, we implement Mb-SDCA (mini-batch stochastic dual coordinate ascent).

The first three methods are optimized and implemented in Apache Spark's MLlib (v1.5.0) (Meng et al., 2016). We test the performance of each method in large-scale experiments fitting lasso, elastic net regression, and SVM models to the datasets shown in Table 5. In comparing to other methods, we plot the distance to the optimal primal solution. This optimal value is calculated by running all methods for a large number of iterations (until progress has stalled), and then selecting the smallest primal value amongst the results. All code is written in Apache Spark and experiments are run on public-cloud Amazon EC2 m3.xlarge machines with one core per machine. Our code is publicly available at `gingsmith.github.io/cocoa/`.

Table 5: Datasets for Empirical Study.

Dataset	Training Size	Feature Size	Sparsity
url	2 M	3 M	3.5e-5
epsilon	400 K	2 K	1.0
kddb	19 M	29 M	9.8e-7
webspam	350 K	16 M	2.0e-4

We carefully tune each competing method in our experiments for best performance. ADMM requires the most tuning, both in selecting the penalty parameter ρ and in solving the subproblems. Solving the subproblems to completion for ADMM is prohibitively slow, and we thus use an iterative method internally and improve performance by allowing early stopping. We also use a varying penalty parameter ρ — practices described in Boyd et al. (2010, Sections 4.3, 8.2.3, 3.4.1). For Mb-SGD, we tune the step size and mini-batch size parameters. For Mb-CD and Mb-SDCA, we scale the updates at each round by $\frac{\beta}{b}$ for mini-batch size b and $\beta \in [1, b]$, and tune both parameters b and β. Further implementation details for all methods are given in Section 6.5. For simplicity of presentation and comparison, in all of the following experiments, we restrict CoCoA to only use simple coordinate descent as the local solver. We note that even stronger empirical results for CoCoA could be obtained by plugging in state-of-the-art local solvers for each application at hand.

6.1 CoCoA in the Primal: An Application to Lasso Regression

We first demonstrate the performance of CoCoA in the primal (Algorithm 2) by applying CoCoA to a lasso regression model (8) fit to the datasets in Table 5. We use stochastic coordinate descent as a local solver for CoCoA, and select the number of local iterations H (a proxy for subproblem approximation quality, Θ) from several options with best performance.

We compare CoCoA to the general methods listed above, including Mb-SGD with an L_1-prox, Prox-GD, OWL-QN, ADMM, and Mb-CD. We provide a comparison with Shotgun (Bradley et al., 2011), a popular method for solving L_1-regularized problems in the multicore environment, as an extreme case to highlight the detrimental effects of frequent communication in the distributed environment. For Mb-CD, Shotgun, and CoCoA in the primal, datasets are distributed by feature, whereas for Mb-SGD, Prox-GD, OWL-QN and ADMM they are distributed by training point.

In analyzing the performance of each algorithm (Figure 1), we measure the improvement to the primal objective given in (A) ($\mathcal{O}_A(\boldsymbol{\alpha})$) in terms of wall-clock time in seconds. We see that both Mb-SGD and Mb-CD are slow to converge, and come with the additional burden of having to tune extra parameters (though Mb-CD makes clear improvements over Mb-SGD). As expected, naively distributing Shotgun (single coordinate updates per machine) does not perform well, as it is tailored to shared-memory systems and requires communicating too frequently. OWL-QN performs the best of all compared methods, but is still much slower to converge than CoCoA, and converges, e.g., 50× more slowly for the webspam dataset. The optimal performance of CoCoA is particularly evident in datasets

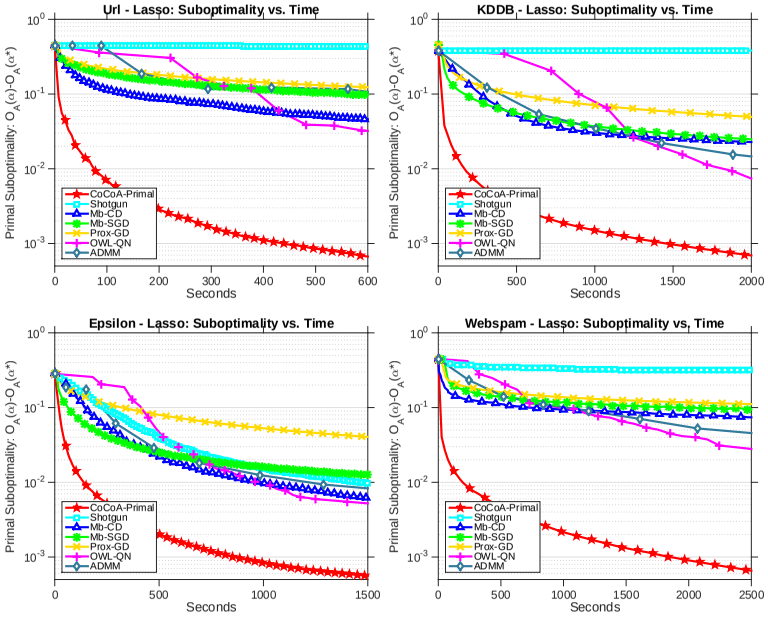


Figure 1: Suboptimality in terms of $\mathcal{O}_A(\boldsymbol{\alpha})$ for fitting a lasso regression model to four datasets: url (K=4, λ=1E-4), kddb (K=4, λ=1E-6), epsilon (K=8, λ=1E-5), and webspam (K=16, λ=1E-5) datasets. CoCoA applied to the primal formulation converges more quickly than all other compared methods in terms of the time in seconds.

with large numbers of features (e.g., url, kddb, webspam), which are exactly the datasets where L_1 regularization would most typically be applied.

Results are shown for regularization parameters λ such that the resulting weight vector $\boldsymbol{\alpha}$ is sparse. However, our results are robust to varying values of λ as well as to various problem settings, as we illustrate in Figure 2.

A case against smoothing. We additionally motivate the use of CoCoA in the primal by showing how it improves upon CoCoA in the dual (Yang, 2013; Jaggi et al., 2014; Ma et al., 2015a, 2017b) for non-strongly convex regularizers. First, CoCoA in the dual cannot be included in the set of experiments in Figure 1 because it cannot be directly applied to the lasso objective (recall that Algorithm 3 only allows for strongly convex regularizers).

To get around this requirement, previous work has suggested implementing the smoothing technique used in, e.g., Shalev-Shwartz and Zhang (2014); Zhang and Lin (2015) — adding a small amount of strong convexity $\delta\|\boldsymbol{\alpha}\|_2^2$ to the objective for lasso regression. In Figure 3 we demonstrate the issues with this approach, comparing CoCoA in the primal

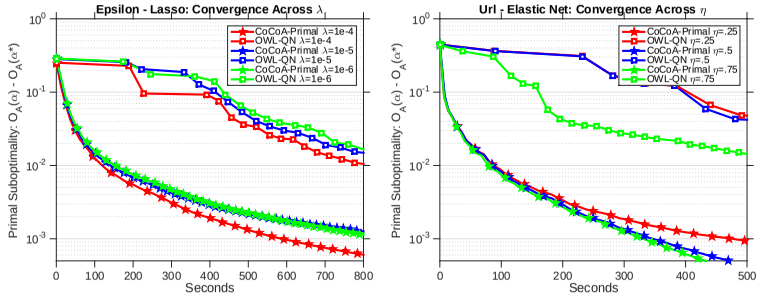


Figure 2: Suboptimality in terms of $\mathcal{O}_A(\boldsymbol{\alpha})$ for solving lasso for the epsilon dataset (left, K=8) and elastic net for the url dataset, (right, K=4, λ=1E-4). Speedups are robust over different regularizers λ (left), and across problem settings, including varying η parameters of elastic net regularization (right).

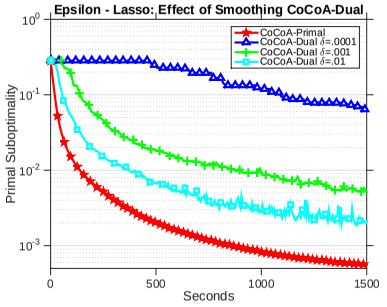


Table 6: Sparsity of Final Iterates.

Method	Sparsity
CoCoA-Primal	0.6030
CoCoA-Dual: $\delta = 0.0001$	0.6035
CoCoA-Dual: $\delta = 0.001$	0.6240
CoCoA-Dual: $\delta = 0.01$	0.6465

Figure 3 & Table 6: For pure L_1 regularization, smoothing is not an effective option for CoCoA in the dual. It either modifies the solution (Figure 3) or slows convergence (Table 6). This motivates running CoCoA instead on the primal for these problems.

on a pure L_1-regularized regression problem to CoCoA in the dual for decreasing levels of δ. The smaller we set δ, the less smooth the problem becomes. As δ decreases, the final sparsity of running CoCoA in the dual starts to match that of running pure L_1 (Table 6), but the performance also degrades (Figure 3). We note that by using CoCoA in the primal with the modification presented in Section 4, we can deliver strong rates without having to make any compromises in terms of the training speed or accuracy.

6.2 CoCoA in the Dual: An Application to SVM Classification

Next we present results on CoCoA in the dual against competing methods, for a hinge loss support vector machine model (9) on the datasets in Table 5. We use stochastic dual

coordinate ascent (SDCA) as a local solver for CoCoA in this setting, again selecting the number of local iterations H from several options with best performance. We compare CoCoA to the general methods listed above, including Mb-SGD, GD, L-BFGS, ADMM, and Mb-SDCA. All datasets are distributed by training point for these methods.

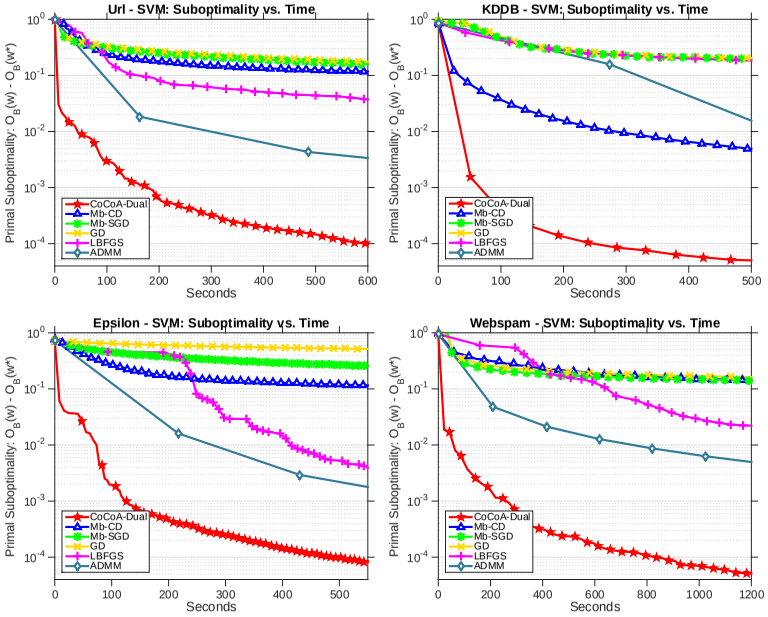


Figure 4: Suboptimality in terms of $\mathcal{O}_B(\mathbf{w})$ for solving a hinge-loss support vector machine model: url (K=4, λ=1e-4), kddb (K=4, λ=1e-6), epsilon (K=8, λ=1e-5), and webspam (K=16, λ=1e-5) datasets. CoCoA applied to the dual formulation converges more quickly than all other compared methods in terms of the time in seconds.

In comparing methods in this setting (Figure 4), we measure the improvement to the primal objective $\mathcal{O}_B(\mathbf{w})$ in terms of wall-clock time in seconds. We see again that Mb-SGD and Mb-CD are slow to converge, and come with the additional burden of having to tune extra parameters. ADMM performs the best of the methods other than CoCoA, followed by L-BFGS. However, both are still much slower to converge than CoCoA in the dual. ADMM in particular is affected by the fact that many internal iterations of SDCA are necessary in order to guarantee convergence. In contrast, CoCoA can incorporate arbitrary amounts of work locally and still converge. We note that although CoCoA, ADMM, and Mb-SDCA run in the dual, Figure 4 tracks progress towards the primal objective, $\mathcal{O}_B(\mathbf{w})$.

6.3 Primal vs. Dual: An Application to Elastic Net Regression

To compare primal vs. dual optimization for CoCoA, we explore both variants by fitting an elastic net regression model (7) to two datasets. We use coordinate descent (with closed-form updates) as the local solver in both variants. From the results in Figure 5, we see that CoCoA in the dual tends to perform better on datasets with a large number of training points (relative to the number of features), and that the performance deteriorates as the strong convexity in the problem disappears. In contrast, CoCoA in the primal performs well on datasets with a large number of features relative to training points, and is robust to changes in strong convexity. These changes in performance are to be expected, as we have discussed that CoCoA in the primal is more suited for non-strongly convex regularizers (Section 6.1), and that the feature size dominates communication for CoCoA in the dual, as compared to the training point size for CoCoA in the primal (Section 3.4).

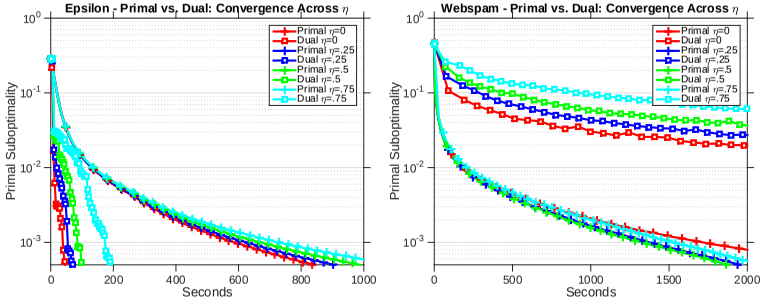


Figure 5: The convergence of CoCoA in the primal versus dual for various values of η in an elastic net regression model. CoCoA in dual performs better on the Epsilon dataset, where the training point size is the dominating term, and CoCoA in the primal performs better on the Webspam dataset, where the feature size is the dominating term. For both datasets, CoCoA in the dual is susceptible to changes in strong convexity—converging more quickly as the problem becomes more strongly convex ($\eta \to 0$), whereas CoCoA in the primal remains robust to changes in strong convexity.

6.4 General Properties: Effect of Communication

Finally, we note that in contrast to the compared methods from Sections 6.1 and 6.2, CoCoA comes with the benefit of having only a single parameter to tune: the subproblem approximation quality, Θ, which we control in our experiments via the number of local subproblem iterations, H, for the example of local coordinate descent. We further explore the effect of this parameter in Figure 6, and provide a general guideline for choosing it in practice (see Remark 1). In particular, we see that while increasing H always results in better performance in terms of the number of communication rounds, smaller or larger values of H may result in better performance in terms of wall-clock time, depending on the cost of communication and computation. The flexibility to fine-tune H is one of the reasons for CoCoA's significant performance gains.

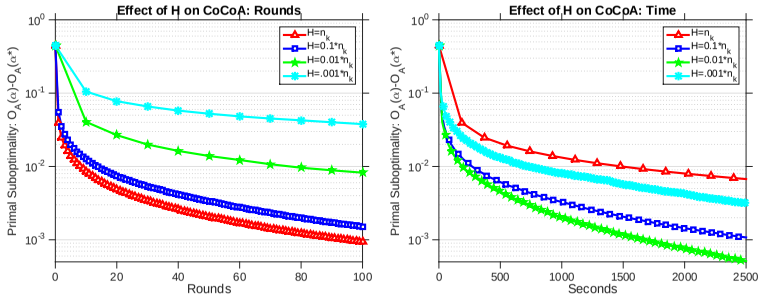


Figure 6: Suboptimality in terms of $\mathcal{O}_A(\boldsymbol{\alpha})$ for solving lasso for the webspam dataset (K=16, λ=1e-5). Here we illustrate how the work spent in the local subproblem (given by H) affects the total performance of CoCoA in terms of number of rounds as well as wall time.

6.5 Experiment Details

In this subsection we provide thorough details on the experimental setup and methods used in our comparison. All experiments are run on Amazon EC2 clusters of m3.xlarge machines, with one core per machine. The code for each method is written in Apache Spark, v1.5.0. Our code is open source and publicly available at `gingsmith.github.io/cocoa/`.

ADMM. ADMM (Boyd et al., 2010) is a popular method that lends itself naturally to the distributed environment. For lasso regression, implementing ADMM for the problems of interest requires solving a large linear system $C\mathbf{x} = \mathbf{d}$ on each machine, where $C \in \mathbb{R}^{n\times n}$ with n scaling beyond 10^7 for the datasets in Table 5, and with C being possibly dense. It is prohibitively slow to solve this directly on each machine, and we therefore employ conjugate gradient with early stopping (see, e.g., Boyd et al., 2010, Section 4.3). For SVM classification, we use stochastic dual coordinate ascent as an internal optimizer, which is shown in Zhang et al. (2012) to have superior performance. We further improve performance with a varying rather than constant penalty parameter, as suggested in Boyd et al., 2010, Section 3.4.1.

Mini-batch SGD and proximal GD. Mini-batch SGD is a standard and widely used method for parallel and distributed optimization. We use the optimized code provided in Spark's machine learning library, MLlib, v1.5.0 (Meng et al., 2016). We tune both the size of the mini-batch and SGD step size using grid search. For lasso, we use the proximal version of the method. Full gradient descent can be seen as a specific setting of mini-batch SGD, where the mini-batch size is equal to the total number of training points. We thus also use the implementation in MLlib for full GD, and tune the step size parameter using grid search.

Mini-batch CD and SDCA. Mini-batch CD (for lasso) and SDCA (for SVM) aim to improve mini-batch SGD by employing coordinate descent, which has theoretical and practical justifications (Shalev-Shwartz and Tewari, 2011; Takáč et al., 2013; Fercoq and Richtárik, 2015). We implement mini-batch CD and SDCA in Spark and scale the updates made at each round by $\frac{\beta}{b}$ for mini-batch size b and $\beta \in [1, b]$, tuning both parameters b and

β via grid search. For the case of lasso regression, we implement Shotgun (Bradley et al., 2011), which is a popular method for parallel optimization. Shotgun can be seen an extreme case of mini-batch CD where the mini-batch is set to K, i.e., there is a single update made by each machine per round. We see in the experiments that communicating this frequently becomes prohibitively slow in the distributed environment.

OWL-QN. OWN-QN (Yu et al., 2010) is a quasi-Newton method optimized in Spark's spark.ml package (Meng et al., 2016). Outer iterations of OWL-QN make significant progress towards convergence, but the iterations themselves can be slow as they require processing the entire dataset. CoCoA, the mini-batch methods, and ADMM with early stopping all improve on this by allowing the flexibility to process only a subset of the dataset at each iteration. CoCoA and ADMM have even greater flexibility by allowing internal methods to process the dataset more than once. CoCoA makes this approximation quality explicit, both in theoretical convergence rates and via guidelines for setting the parameter.

CoCoA. We implement CoCoA with coordinate descent as the local solver. We note that since the framework and theory allow any internal solver to be used, CoCoA could benefit even beyond the results shown, e.g., by using existing fast L_1-solvers for the single-machine case, such as glmnet variants (Friedman et al., 2010) or blitz (Johnson and Guestrin, 2015) or SVM solvers like liblinear (Fan et al., 2008). The only parameter influencing the overall performance of CoCoA is the level of approximation quality, which we parameterize in the experiments through H, the number of local iterations of the iterative method run locally. Our theory relates local approximation quality to global convergence (Section 4), and we provide a guideline for how to choose this value in practice that links the parameter to the systems environment at hand (Remark 1).

7. Related Work

There exist myriad optimization methods for the distributed setting; the following section is not meant to be wholly comprehensive, but to provide an overview of the most prevalent and related approaches. We additionally note that many new methods have been proposed since the time of submission of this manuscript in October 2016, including several extensions of the presented CoCoA framework—e.g., for federated learning (Smith et al., 2017), computing over heterogeneous systems (Dünner et al., 2017), second-order algorithm extensions (Gargiani, 2017; Lee and Chang, 2017; Dünner et al., 2018; Lee et al., 2018), and accelerated methods (Ma et al., 2017a; Zheng et al., 2017). We defer the readers to these follow-up works for the most current literature review.

Single-machine coordinate solvers. For strongly convex regularizers, state-of-the-art for empirical loss minimization is randomized coordinate ascent on the dual (SDCA) (Shalev-Shwartz and Zhang, 2013a) and accelerated variants (e.g., Shalev-Shwartz and Zhang, 2014). In contrast to primal stochastic gradient descent (SGD) methods, the SDCA family is often preferred as it is free of learning rate parameters and has faster (geometric) convergence guarantees. Interestingly, a similar trend in coordinate solvers exists in recent lasso literature, but with the roles of primal and dual reversed. For those problems, primal-based coordinate descent methods are state-of-the-art, as in glmnet (Friedman et al., 2010) and extensions (Yuan et al., 2012); see, e.g., the overview in Yuan et al. (2010). However, primal-

dual rates for unmodified coordinate methods have to our knowledge only been obtained for strongly convex regularizers to date (Shalev-Shwartz and Zhang, 2014; Zhang and Lin, 2015).

Coordinate descent on L_1-regularized problems (i.e., (A) with $g(\cdot) = \lambda\|\cdot\|_1$) can be interpreted as the iterative minimization of a quadratic approximation of the smooth part of the objective, followed by a shrinkage step. In the single-coordinate update case, this is at the core of GLMNET (Friedman et al., 2010; Yuan et al., 2010), and widely used in, e.g., solvers based on the primal formulation of L_1-regularized objectives (Shalev-Shwartz and Tewari, 2011; Yuan et al., 2012; Bian et al., 2013; Fercoq and Richtárik, 2015; Tappenden et al., 2015). When changing more than one coordinate at a time, again employing a quadratic upper bound on the smooth part, this results in a two-loop method as in GLMNET for the special case of logistic regression. In the distributed setting, when the set of active coordinates coincides with the ones on the local machine, these single-machine approaches closely resemble the distributed framework proposed here.

Parallel methods. For the general regularized loss minimization problems of interest, methods based on stochastic subgradient descent (SGD) are well-established. Several variants of SGD have been proposed for parallel computing, many of which build on the idea of asynchronous communication (Niu et al., 2011; Duchi et al., 2013). Despite their simplicity and competitive performance on shared-memory systems, the downside of this approach in the distributed environment is that the amount of required communication is equal to the amount of data read locally, since one data point is accessed per machine per round (e.g., mini-batch SGD with a batch size of one per worker). These variants are in practice not competitive with the more communication-efficient methods considered in this work, which allow more local updates per communication round.

For the specific case of L_1-regularized objectives, parallel coordinate descent (with and without using mini-batches) was proposed in Bradley et al. (2011) (Shotgun) and generalized in Bian et al. (2013) , and is among the best performing solvers in the parallel setting. Our framework reduces to Shotgun as a special case when the internal solver is a single-coordinate update on the subproblem (10), $\gamma = 1$, and for a suitable σ'. However, Shotgun is not covered by our convergence theory, since it uses a potentially unsafe upper bound of β instead of σ', which isn't guaranteed to satisfy our condition for convergence (11). We compare empirically with Shotgun in Section 6 to highlight the detrimental effects of running this high-communication method in the distributed environment.

One-shot communication schemes. At the other extreme, there are distributed methods that use only a single round of communication, such as Mann et al. (2009); Zinkevich et al. (2010); Zhang et al. (2013); McWilliams et al. (2014); and Heinze et al. (2016). These methods require additional assumptions on the partitioning of the data, which are usually not satisfied in practice if the data are distributed "as is", i.e., if we do not have the opportunity to distribute the data in a specific way beforehand. Furthermore, some cannot guarantee convergence rates beyond what could be achieved if we ignored data residing on all but a single computer, as shown in Shamir et al. (2014). Additional relevant lower bounds on the minimum number of communication rounds necessary for a given approximation quality are presented in Balcan et al. (2012) and Arjevani and Shamir (2015).

Mini-batch methods. Mini-batch methods (which use updates from several training points or features per round) are more flexible and lie within the two extremes of parallel and one-shot communication schemes. However, mini-batch versions of both SGD and coordinate descent (CD) (e.g., Dekel et al., 2012; Takáč et al., 2013; Shalev-Shwartz and Zhang, 2013b; Shamir and Srebro, 2014; Qu et al., 2015; Richtárik and Takáč, 2016; Défossez and Bach, 2017) suffer from their convergence rate degrading towards the rate of batch gradient descent as the size of the mini-batch is increased. This follows because mini-batch updates are made based on the outdated previous parameter vector $\mathbf{w}$, in contrast to methods that allow immediate local updates like CoCoA.

Another disadvantage of mini-batch methods is that the aggregation parameter is more difficult to tune, as it can lie anywhere in the order of mini-batch size. The optimal choice is often either unknown or too challenging to compute in practice. In the CoCoA framework there is no need to tune parameters, as the aggregation parameter and subproblem parameters can be set directly using the safe bound discussed in Section 3 (Definition 5).

Batch solvers. ADMM (Boyd et al., 2010), gradient descent, and quasi-Newton methods such as L-BFGS and are also often used in distributed settings because of their relatively low communication requirements. However, they require at least a full (distributed) batch gradient computation at each round, and therefore do not allow the gradual trade-off between communication and computation provided by CoCoA. In Section 6, we include experimental comparisons with ADMM, gradient descent, and L-BFGS variants, including orthant-wise limited memory quasi-Newton (OWL-QN) for the L_1 setting (Andrew and Gao, 2007).

Finally, we note that while the convergence rates provided for CoCoA mirror the convergence class of classical batch gradient methods in terms of the number of outer rounds, existing batch gradient methods come with a weaker theory, as they do not allow general inexactness Θ for the local subproblem (10). In contrast, our convergence rates incorporate this approximation directly, and, moreover, hold for arbitrary local solvers of much cheaper cost than batch methods (where in each round, every machine has to process exactly a full pass through the local data). This makes CoCoA more flexible in the distributed setting, as it can adapt to varied communication costs on real systems. We have seen in Section 6 that this flexibility results in significant performance gains over the competing methods.

Distributed solvers. By making use of the primal-dual structure in the line of work of Yu et al. (2012); Pechyony et al. (2011); Yang (2013); Yang et al. (2013) and Lee and Roth (2015), the CoCoA-v1 and CoCoA$^+$ frameworks (which are special cases of the presented framework, CoCoA) are the first to allow the use of any local solver—of weak local approximation quality—in each round in the distributed setting. The practical variant of the DisDCA (Yang, 2013), called DisDCA-p, allows for additive updates in a similar manner to CoCoA, but is restricted to coordinate decent (CD) being the local solver, and was initially proposed without convergence guarantees. DisDCA-p, CoCoA-v1, and CoCoA$^+$ are all limited to strongly convex regularizers, and therefore are not as general as the CoCoA framework discussed in this work.

In the L_1-regularized setting, an approach related to our framework includes distributed variants of GLMNET as in Mahajan et al. (2017). Inspired by GLMNET and Yuan et al. (2012), the works of Bian et al. (2013) and Mahajan et al. (2017) introduced the idea of a block-diagonal Hessian upper approximation in the distributed L_1 context. The later work of Trofimov and Genkin (2014, 2016) specialized this approach to sparse logistic regression.

If hypothetically each of our quadratic subproblems $\mathcal{G}_k^{\sigma'}(\Delta\boldsymbol{\alpha}_{[k]})$ as defined in (10) were to be minimized exactly, the resulting steps could be interpreted as block-wise Newton-type steps on each coordinate block k, where the Newton-subproblem is modified to also contain the L_1-regularizer (Mahajan et al., 2017; Yuan et al., 2012; Qu et al., 2016). While Mahajan et al. (2017) allows a fixed accuracy for these subproblems, but not arbitrary approximation quality Θ as in our framework, the works of Trofimov and Genkin (2016); Yuan et al. (2012); and Yen et al. (2015) assume that the quadratic subproblems are solved exactly. Therefore, these methods are not able to freely trade off communication and computation. Also, they do not allow the re-use of arbitrary local solvers. On the theoretical side, the convergence rate results provided by Mahajan et al. (2017); Trofimov and Genkin (2016); and Yuan et al. (2012) are not explicit convergence rates but only asymptotic, as the quadratic upper bounds are not explicitly controlled for safety as with our σ'.

8. Discussion

To enable large-scale machine learning and signal processing, we have developed, analyzed, and evaluated a general-purpose framework for communication-efficient primal-dual optimization in the distributed environment. Our framework, CoCoA, takes a unique approach by using duality to derive subproblems for each machine to solve in parallel. These subproblems closely match the global problem of interest, which allows for state-of-the-art single-machine solvers to easily be re-used in the distributed setting. Further, by allowing the local solvers to find solutions of arbitrary approximation quality to the subproblems on each machine, our framework permits a highly flexible communication scheme. In particular, as the local solvers make updates directly to their local parameters, the need to communicate reduces and can be adapted to the system at hand, which helps to manage the communication bottleneck in the distributed setting.

We analyzed the impact of the local solver approximation quality and derived global primal-dual convergence rates for our framework that are agnostic to the specifics of the local solvers. We have taken particular care in extending our framework to the case of non-strongly convex regularizers, where we introduced a bounded-support modification technique to provide robust convergence guarantees. Finally, we demonstrated the efficiency of our framework in an extensive experimental comparison with state-of-the-art distributed solvers. Our framework achieves up to a 50× speedup over other widely-used methods on real-world distributed datasets.

Acknowledgments

We thank Michael P. Friedlander, Matilde Gargiani, Sai Praneeth Karimireddy, Jakub Konečný, Ching-pei Lee, and Peter Richtárik for their help and for fruitful discussions. We are additionally grateful to the reviewers for their valuable comments. We wish to acknowledge support from the U.S. National Science Foundation, under award number NSF:CCF:1618717, NSF:CMMI:1663256 and NSF:CCF:1740796; the Swiss National Science Foundation, under grant number 175796; and the Mathematical Data Science program of the Office of Naval Research, under grant number N00014-15-1-2670.

Appendix A. Convex Conjugates

The convex conjugate of a function $f : \mathbb{R}^m \to \mathbb{R}$ is defined as

$$f^*(\mathbf{v}) := \max_{\mathbf{u}\in\mathbb{R}^m} \mathbf{v}^\top\mathbf{u} - f(\mathbf{u})\,. \tag{27}$$

Below we list several useful properties of conjugates (see, e.g., Boyd and Vandenberghe, 2004, Section 3.3.2):

- Double conjugate: $(f^*)^* = f$ if f is closed and convex.
- Value Scaling: (for $\alpha > 0$) $f(\mathbf{v}) = \alpha g(\mathbf{v}) \quad \Rightarrow \quad f^*(\mathbf{w}) = \alpha g^*(\mathbf{w}/\alpha)\,.$
- Argument Scaling: (for $\alpha \neq 0$) $f(\mathbf{v}) = g(\alpha\mathbf{v}) \quad \Rightarrow \quad f^*(\mathbf{w}) = g^*(\mathbf{w}/\alpha)\,.$
- Conjugate of a separable sum: $f(\mathbf{v}) = \sum_i \phi_i(v_i) \quad \Rightarrow \quad f^*(\mathbf{w}) = \sum_i \phi_i^*(w_i)\,.$

Lemma 4 (Duality between Lipschitzness and L-Bounded Support, (Rockafellar, 1997, Corollary 13.3.3)). *Given a proper convex function f, it holds that f is L-Lipschitz if and only if f^* has L-bounded support.*

Lemma 5 (Duality between Smoothness and Strong Convexity, (Hiriart-Urruty and Lemaréchal, 2001, Theorem 4.2.2)). *Given a closed convex function f, it holds that f is μ strongly convex w.r.t. the norm $\|\cdot\|$ if and only if f^* is $(1/\mu)$-smooth w.r.t. the dual norm $\|\cdot\|_*$.*

Appendix B. Proofs of Primal-Dual Relationships

In the following subsections we provide derivations of the primal-dual relationship of the general objectives (A) and (B), and then show how to derive the conjugate of the modified L_1-norm, as an example of the bounded-support modification introduced in Section 4.

B.1 Primal-Dual Relationship

The relation of the original formulation (A) to its dual formulation (B) is standard in convex analysis. Using the linear map A as in our case, the relationship is an instance of Fenchel-Rockafellar Duality, see e.g. Borwein and Zhu (2005, Theorem 4.4.2) or Bauschke and Combettes (2011, Proposition 15.18). For completeness, we illustrate this correspondence with a self-contained derivation of the duality.

Starting with the original formulation (A), we introduce an auxiliary vector $\mathbf{v} \in \mathbb{R}^m$ representing $\mathbf{v} = A\boldsymbol{\alpha}$. Then optimization problem (A) becomes:

$$\min_{\boldsymbol{\alpha},\mathbf{v}} \quad f(\mathbf{v}) + g(\boldsymbol{\alpha}) \quad \text{such that } \mathbf{v} = A\boldsymbol{\alpha}\,. \tag{28}$$

Introducing Lagrange multipliers $\mathbf{w} \in \mathbb{R}^m$, the Lagrangian is given by:

$$L(\boldsymbol{\alpha}, \mathbf{v}; \mathbf{w}) := f(\mathbf{v}) + g(\boldsymbol{\alpha}) + \mathbf{w}^\top (A\boldsymbol{\alpha} - \mathbf{v})\,.$$

The dual problem of (A) follows by taking the infimum with respect to both $\boldsymbol{\alpha}$ and $\mathbf{v}$:

$$\begin{aligned}\inf_{\boldsymbol{\alpha},\mathbf{v}} L(\mathbf{w},\boldsymbol{\alpha},\mathbf{v}) &= \inf_{\mathbf{v}}\left\{f(\mathbf{v}) - \mathbf{w}^\top\mathbf{v}\right\} + \inf_{\boldsymbol{\alpha}}\left\{g(\boldsymbol{\alpha}) + \mathbf{w}^\top A\boldsymbol{\alpha}\right\}\\ &= -\sup_{\mathbf{v}}\left\{\mathbf{w}^\top\mathbf{v} - f(\mathbf{v})\right\} - \sup_{\boldsymbol{\alpha}}\left\{(-\mathbf{w}^\top A)\boldsymbol{\alpha} - g(\boldsymbol{\alpha})\right\}\\ &= -f^*(\mathbf{w}) - g^*(-A^\top\mathbf{w})\,. \end{aligned} \tag{29}$$

We change signs and turn the maximization of the dual problem (29) into a minimization, thereby arriving at the dual formulation (B) as claimed:

$$\min_{\mathbf{w}\in\mathbb{R}^m} \quad \Big[\ \mathcal{O}_B(\mathbf{w}) := f^*(\mathbf{w}) + g^*(-A^\top\mathbf{w})\ \Big]\,.$$

B.2 Continuous Conjugate Modification for Indicator Functions

Lemma 6 (Conjugate of the modified L_1-norm)**.** *The convex conjugate of the bounded support modification of the L_1-norm, as defined in* (18)*, is:*

$$\bar{g}_i^*(x) := \begin{cases} 0 & : x \in [-1,1],\\ B(|x|-1) & : \textit{otherwise,}\end{cases}$$

and is B-Lipschitz.

Proof. We start by applying the definition of convex conjugate:

$$\bar{g}_i(\alpha) = \sup_{x\in\mathbb{R}}\left[\alpha x - \bar{g}_i^*(x)\right]\,.$$

We begin by looking at the case in which $\alpha \geq B$; in this case it's easy to see that when $x \to +\infty$, we have:

$$\alpha x - B(|x|-1) = (\alpha - B)x - B \to +\infty\,,$$

as $\alpha - B \geq 0$. The case $\alpha \leq -B$ holds analogously. We'll now look at the case $\alpha \in [0,B]$; in this case it is clear we must have $x^\star \geq 0$. It also must hold that $x^\star \leq 1$, since

$$\alpha x - B(x-1) < \alpha x\,,$$

for every $x > 1$. Therefore the maximization becomes

$$\bar{g}_i(\alpha) = \sup_{x\in[0,1]} \alpha x\,,$$

which has maximum α at $x = 1$. The remaining $\alpha \in [-B,0]$ case follows in similar fashion.

Lipschitz continuity of $\bar{g}_i^*$ follows directly, or alternatively also from the general result that g_i^* is L-Lipschitz if and only if g_i has L-bounded support (Rockafellar, 1997, Corollary 13.3.3) or (Dünner et al., 2016, Lemma 5). □

Appendix C. Comparison to ADMM

C.1 ADMM Applied to the $\mathcal{O}_B(\cdot)$ Formulation

Here we compare consensus ADMM (Mota et al., 2013) applied to the problem (B) to the CoCoA framework, as discussed in Section 3.5. For consensus ADMM, the objective $\mathcal{O}_B(\cdot)$ can be decomposed using the following re-parameterization:

$$\begin{aligned}\min_{\mathbf{w}_1,\ldots\mathbf{w}_K,\mathbf{w}} \quad & \sum_{k=1}^K\sum_{i\in\mathcal{P}_k} g_i^*(-\mathbf{x}_i^\top\mathbf{w}_k) + f^*(\mathbf{w})\\ s.t. \quad & \mathbf{w}_k = \mathbf{w},\ k = 1,\ldots,K.\end{aligned}$$

To solve this problem, we construct the augmented Lagrangian:

$$\begin{aligned}L_\rho(\mathbf{w}_1,\ldots,\mathbf{w}_k,\mathbf{w},\mathbf{u}_1,\ldots,\mathbf{u}_k) := &\sum_{k=1}^K\sum_{i\in\mathcal{P}_k} g_i^*(-\mathbf{x}_i^\top\mathbf{w}_k)\\ &+ f^*(\mathbf{w}) + \sum_{k=1}^K\mathbf{u}_k^\top(\mathbf{w}_k - \mathbf{w}) + \frac{\rho}{2}\sum_{k=1}^K\|\mathbf{w}_k - \mathbf{w}\|^2\,,\end{aligned}$$

which yields the following decomposable updates:

$$\begin{aligned}\mathbf{w}_k^{(t+1)} &= \arg\min_{\mathbf{w}_k} \sum_{i\in\mathcal{P}_k} g_i^*(-\mathbf{x}_i^\top\mathbf{w}_k) + \mathbf{u}_k^{(t)\top}(\mathbf{w}_k - \mathbf{w}^{(t)}) + \frac{\rho}{2}\|\mathbf{w}_k - \mathbf{w}^{(t)}\|^2,\\ \mathbf{w}^{(t+1)} &= \arg\min_{\mathbf{w}} f^*(\mathbf{w}) + \sum_{k=1}^K\mathbf{u}_k^{(t)\top}(\mathbf{w}_k^{(t+1)} - \mathbf{w}) + \frac{\rho}{2}\sum_{k=1}^K\|\mathbf{w}_k^{(t+1)} - \mathbf{w}\|^2,\\ \mathbf{u}_k^{(t+1)} &= \mathbf{u}_k^{(t)} + \rho(\mathbf{w}_k^{(t+1)} - \mathbf{w}^{(t+1)}).\end{aligned}$$

These updates can be further simplified by using the scaled form of $\mathbf{u}_k$ and combining terms for $\mathbf{w}$ using the averages $\bar{\mathbf{w}}_k$ and $\bar{\mathbf{u}}_k$:

$$\begin{aligned}\mathbf{w}_k^{(t+1)} &= \arg\min_{\mathbf{w}_k} \sum_{i\in\mathcal{P}_k} g_i^*(-\mathbf{x}_i^\top\mathbf{w}_k) + \rho\mathbf{u}_k^{(t)\top}(\mathbf{w}_k - \mathbf{w}^{(t)}) + \frac{\rho}{2}\|\mathbf{w}_k - \mathbf{w}^{(t)}\|^2,\\ \mathbf{w}^{(t+1)} &= \arg\min_{\mathbf{w}} f^*(\mathbf{w}) + \frac{\rho K}{2}\|\mathbf{w} - (\bar{\mathbf{w}}_k^{(t+1)} + \bar{\mathbf{u}}_k^{(t)})\|^2,\\ \mathbf{u}_k^{(t+1)} &= \mathbf{u}_k^{(t)} + \mathbf{w}_k^{(t+1)} - \mathbf{w}^{(t+1)}.\end{aligned}$$

To compare this to the CoCoA subproblems (10), we will derive the dual form of the update to $\mathbf{w}_k$. Suppressing the iteration counter for simplicity, the minimization is of the form:

$$\begin{aligned}&\min_{\mathbf{w}_k} \sum_{i\in\mathcal{P}_k} g_i^*(-\mathbf{x}_i^\top\mathbf{w}_k) + \rho\mathbf{u}_k{}^\top(\mathbf{w}_k - \mathbf{w}) + \frac{\rho}{2}\|\mathbf{w}_k - \mathbf{w}\|^2\\ =&\min_{\mathbf{w}_k} \sum_{i\in\mathcal{P}_k} \max_{\alpha_i} -\mathbf{x}_i^\top\mathbf{w}_k\alpha_i - g_i(\alpha_i) + \rho\mathbf{u}_k{}^\top(\mathbf{w}_k - \mathbf{w}) + \frac{\rho}{2}\|\mathbf{w}_k - \mathbf{w}\|^2\\ =&\max_{\boldsymbol{\alpha}_{[k]}}\min_{\mathbf{w}_k} -\mathbf{w}_k^\top A_{[k]}\boldsymbol{\alpha}_{[k]} - \sum_{i\in\mathcal{P}_k} g_i(\boldsymbol{\alpha}_{[k]i}) + \rho\mathbf{u}_k{}^\top(\mathbf{w}_k - \mathbf{w}) + \frac{\rho}{2}\|\mathbf{w}_k - \mathbf{w}\|^2\,.\end{aligned}$$

Solving the minimization yields: $\mathbf{w}_k = \frac{1}{\rho} A_{[k]}\boldsymbol{\alpha}_{[k]} - \mathbf{u}_k + \mathbf{w}$. Plugging this back in, we have:

$$\begin{aligned}
&\max_{\boldsymbol{\alpha}_{[k]}} \sum_{i\in\mathcal{P}_k} -g_i(\boldsymbol{\alpha}_{[k]_i}) - (\frac{1}{\rho}A_{[k]}\boldsymbol{\alpha}_{[k]} - \mathbf{u}_k + \mathbf{w})^\top A_{[k]}\boldsymbol{\alpha}_{[k]} + \rho\mathbf{u}_k{}^\top(\frac{1}{\rho}A_{[k]}\boldsymbol{\alpha}_{[k]} - \mathbf{u}_k) + \frac{\rho}{2}\|\frac{1}{\rho}A_{[k]}\boldsymbol{\alpha}_{[k]} - \mathbf{u}_k\|^2 \\
=&\max_{\boldsymbol{\alpha}_{[k]}} \sum_{i\in\mathcal{P}_k} -g_i(\boldsymbol{\alpha}_{[k]_i}) + \mathbf{u}_k^\top A_{[k]}\boldsymbol{\alpha}_{[k]} - \mathbf{w}^\top A_{[k]}\boldsymbol{\alpha}_{[k]} - \frac{1}{2\rho}\|A_{[k]}\boldsymbol{\alpha}_{[k]}\|^2 \\
=&\min_{\boldsymbol{\alpha}_{[k]}} \sum_{i\in\mathcal{P}_k} g_i(\boldsymbol{\alpha}_{[k]_i}) + (\mathbf{w} - \mathbf{u}_k)^\top A_{[k]}\boldsymbol{\alpha}_{[k]} + \frac{1}{2\rho}\|A_{[k]}\boldsymbol{\alpha}_{[k]}\|^2 .
\end{aligned}$$

We therefore see that the update to $\mathbf{w}_k$ has a similar form to the CoCoA subproblem (10), where $\rho := \frac{\tau}{\sigma'}$.

C.2 ADMM Applied to the $\mathcal{O}_A(\cdot)$ Formulation

We can also compare CoCoA to ADMM as applied to the (A) problem. For consensus ADMM, the objective $\mathcal{O}_A(\cdot)$ can be decomposed using the following re-parametrization, which introduces local copies $\boldsymbol{\alpha}_k$ of the global variable $\boldsymbol{\alpha}$, and a set of consensus constraints to achieve the equality between them:

$$\begin{aligned}
\min_{\boldsymbol{\alpha},\boldsymbol{\alpha}_1,\dots\boldsymbol{\alpha}_K} \quad & f(A\boldsymbol{\alpha}) + \sum_{k=1}^K \sum_{i\in\mathcal{P}_k} g_i(\boldsymbol{\alpha}_{k,i}) \\
s.t. \quad & A_{[k]}\boldsymbol{\alpha} = A_{[k]}\boldsymbol{\alpha}_k,\ k = 1,\dots,K.
\end{aligned}$$

To solve this problem, we construct the augmented Lagrangian with a penalty parameter ρ:

$$\begin{aligned}
L_\rho(\tilde{\boldsymbol{\alpha}}, \boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_K, \mathbf{w}_1, \dots, \mathbf{w}_k) := f(A\tilde{\boldsymbol{\alpha}}) + \\
\sum_{k=1}^K \left[\sum_{i\in\mathcal{P}_k} g_i(\boldsymbol{\alpha}_{k,i}) + \mathbf{w}_k^\top A_{[k]}(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}) + \frac{\rho}{2}\|A_{[k]}(\boldsymbol{\alpha}_k - \boldsymbol{\alpha})\|_2^2 \right]
\end{aligned}$$

which yields to the following decomposable updates:

$$\begin{aligned}
\boldsymbol{\alpha}_k^{(t)} &= \arg\min_{\boldsymbol{\alpha}_k} \sum_{i\in\mathcal{P}_k} g_i(\boldsymbol{\alpha}_{k,i}) + \mathbf{w}_k^{(t-1)\top} A_{[k]}(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}^{(t-1)}) + \frac{\rho}{2}\|A_{[k]}(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}^{(t-1)})\|_2^2, \\
\boldsymbol{\alpha}^{(t)} &= \arg\min_{\boldsymbol{\alpha}} f(A\boldsymbol{\alpha}) + \sum_{k=1}^K \left[\mathbf{w}_k^{(t-1)\top} A_{[k]}(\boldsymbol{\alpha}_k^{(t)} - \boldsymbol{\alpha}) + \frac{\rho}{2}\|A_{[k]}(\boldsymbol{\alpha}_k^{(t)} - \boldsymbol{\alpha})\|_2^2 \right], \\
\mathbf{w}_k^{(t)} &= \mathbf{w}_k^{(t-1)} + \rho A_{[k]}(\boldsymbol{\alpha}_k^{(t)} - \boldsymbol{\alpha}^{(t)}).
\end{aligned}$$

The first minimization is solved locally in a distributed manner by the K partitions. By setting $\rho := \frac{\sigma'}{\tau}$ and applying the change of variables $\Delta\boldsymbol{\alpha}_{[k]} = (\boldsymbol{\alpha}_k - \boldsymbol{\alpha}^{(t-1)})$, we can obtain the CoCoA local subproblems (10):

$$\Delta\boldsymbol{\alpha}_{[k]}{}^{(t)} = \arg\min_{\Delta\boldsymbol{\alpha}_{[k]}} \sum_{i\in\mathcal{P}_k} g(\alpha_i^{(t-1)} + \Delta\boldsymbol{\alpha}_{[k]_i}) + \mathbf{w}^{(t-1)\top} A_{[k]}\Delta\boldsymbol{\alpha}_{[k]} + \frac{\sigma'}{2\tau}\|A_{[k]}\Delta\boldsymbol{\alpha}_{[k]}\|^2$$

Note that the update to $\boldsymbol{\alpha}$ in its current form is not separable and must be solved via some distributed optimization procedure. The comparison to the $\mathcal{O}_B(\cdot)$ formulation is more natural in this sense, as it captures a setting and formulation in which distributed ADMM would more commonly be applied. However, the above formulation is closely related to the *sharing* variant of ADMM (Boyd et al., 2010, Section 7.3), where data for canonical regularized loss minimization problems is assumed to be distributed via features (see, e.g., Boyd et al., 2010, Section 8.3).

Appendix D. Convergence Proofs

In this section we provide proofs of our main convergence results. The arguments follow the reasoning in Ma et al. (2015a, 2017b), but where we have generalized them to be applicable directly to (A). We provide full details of Lemma 1 as a proof of concept, but omit details in later proofs that can be derived using the arguments in Ma et al. (2015a) or earlier work of Shalev-Shwartz and Zhang (2013a), and instead outline the proof strategy and highlight sections where the theory deviates.

D.1 Approximation of $\mathcal{O}_A(\cdot)$ by the Local Subproblems $\mathcal{G}_k^{\sigma'}(\cdot)$

Our first lemma in the overall proof of convergence helps to relate progress on the local subproblems to the global objective $\mathcal{O}_A(\cdot)$.

Lemma' 1. *For any dual variables $\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha} \in \mathbb{R}^n$, $\mathbf{v} = \mathbf{v}(\boldsymbol{\alpha}) := A\boldsymbol{\alpha}$, and real values γ, σ' satisfying* (11)*, it holds that*

$$\mathcal{O}_A\Big(\boldsymbol{\alpha} + \gamma\sum_{k=1}^K \Delta\boldsymbol{\alpha}_{[k]}\Big) \le (1-\gamma)\mathcal{O}_A(\boldsymbol{\alpha}) + \gamma\sum_{k=1}^K \mathcal{G}_k^{\sigma'}(\Delta\boldsymbol{\alpha}_{[k]}; \mathbf{v}, \boldsymbol{\alpha}_{[k]}) . \tag{30}$$

Proof. In this proof we follow the line of reasoning in Ma et al. (2015a, Lemma 4) with a more general $(1/\tau)$ smoothness assumption on $f(\cdot)$. An outer iteration of CoCoA performs the following update:

$$\mathcal{O}_A(\boldsymbol{\alpha} + \gamma\sum_{k=1}^K \Delta\boldsymbol{\alpha}_{[k]}) = \underbrace{f(\mathbf{v}(\boldsymbol{\alpha} + \gamma\sum_{k=1}^K \Delta\boldsymbol{\alpha}_{[k]}))}_{A} + \underbrace{\sum_{i=1}^n g_i(\alpha_i + \gamma(\sum_{k=1}^K \Delta\boldsymbol{\alpha}_{[k]})_i)}_{B} . \tag{31}$$

We bound A and B separately. First we bound A using $(1/\tau)$-smoothness of f:

$$\begin{aligned}
A &= f\Big(\mathbf{v}(\boldsymbol{\alpha} + \gamma\sum_{k=1}^K \Delta\boldsymbol{\alpha}_{[k]})\Big) = f\Big(\mathbf{v}(\boldsymbol{\alpha}) + \gamma\sum_{k=1}^K \mathbf{v}(\Delta\boldsymbol{\alpha}_{[k]})\Big) \\
&\overset{\text{smoothness of } f \text{ as in (3)}}{\le} f(\mathbf{v}(\boldsymbol{\alpha})) + \sum_{k=1}^K \gamma\nabla f(\mathbf{v}(\boldsymbol{\alpha}))^\top \mathbf{v}(\Delta\boldsymbol{\alpha}_{[k]}) + \frac{\gamma^2}{2\tau}\|\sum_{k=1}^K \mathbf{v}(\Delta\boldsymbol{\alpha}_{[k]})\|^2 \\
&\overset{\text{definition of } \mathbf{w} \text{ as in (5)}}{=} f(\mathbf{v}(\boldsymbol{\alpha})) + \sum_{k=1}^K \gamma\mathbf{v}(\Delta\boldsymbol{\alpha}_{[k]})^\top \mathbf{w}(\boldsymbol{\alpha}) + \frac{\gamma^2}{2\tau}\|\sum_{k=1}^K \mathbf{v}(\Delta\boldsymbol{\alpha}_{[k]})\|^2
\end{aligned}$$

$$\overset{\text{safe choice of } \sigma' \text{ as in (11)}}{\leq} f(\mathbf{v}(\boldsymbol{\alpha})) + \sum_{k=1}^{K} \gamma \mathbf{v}(\Delta\boldsymbol{\alpha}_{[k]})^\top \mathbf{w}(\boldsymbol{\alpha}) + \frac{1}{2\tau}\gamma\sigma' \sum_{k=1}^{K} \|\mathbf{v}(\Delta\boldsymbol{\alpha}_{[k]})\|^2 .$$

Next we use Jensen's inequality to bound B:

$$B = \sum_{k=1}^{K} \left(\sum_{i\in\mathcal{P}_k} g_i(\alpha_i + \gamma(\Delta\boldsymbol{\alpha}_{[k]})_i) \right) = \sum_{k=1}^{K} \left(\sum_{i\in\mathcal{P}_k} g_i((1-\gamma)\alpha_i + \gamma(\alpha_i + (\Delta\boldsymbol{\alpha}_{[k]})_i)) \right)$$
$$\leq \sum_{k=1}^{K} \left(\sum_{i\in\mathcal{P}_k} (1-\gamma) g_i(\alpha_i) + \gamma g_i(\alpha_i + (\Delta\boldsymbol{\alpha}_{[k]})_i) \right) .$$

Plugging A and B back into (31) yields:

$$\begin{aligned}
\mathcal{O}_A\Big(\boldsymbol{\alpha} + \gamma \sum_{k=1}^{K} \Delta\boldsymbol{\alpha}_{[k]}\Big) \leq\ & f(\mathbf{v}(\boldsymbol{\alpha})) \pm \gamma f(\mathbf{v}(\boldsymbol{\alpha})) + \sum_{k=1}^{K} \gamma \mathbf{v}(\Delta\boldsymbol{\alpha}_{[k]})^\top \mathbf{w}(\boldsymbol{\alpha}) + \frac{1}{2\tau}\gamma\sigma' \sum_{k=1}^{K} \|\mathbf{v}(\Delta\boldsymbol{\alpha}_{[k]})\|^2 \\
& + \sum_{k=1}^{K} \sum_{i\in\mathcal{P}_k} (1-\gamma) g_i(\alpha_i) + \gamma g_i(\alpha_i + (\Delta\boldsymbol{\alpha}_{[k]})_i) \\
=\ & \underbrace{(1-\gamma) f(\mathbf{v}(\boldsymbol{\alpha})) + \sum_{k=1}^{K} \left(\sum_{i\in\mathcal{P}_k} (1-\gamma) g_i(\alpha_i) \right)}_{(1-\gamma)\mathcal{O}_A(\boldsymbol{\alpha})} \\
& + \gamma \sum_{k=1}^{K} \left(\frac{1}{K} f(\mathbf{v}(\boldsymbol{\alpha})) + \mathbf{v}(\Delta\boldsymbol{\alpha}_{[k]})^\top \mathbf{w}(\boldsymbol{\alpha}) + \frac{\sigma'}{2\tau} \|\mathbf{v}(\Delta\boldsymbol{\alpha}_{[k]})\|^2 + \sum_{i\in\mathcal{P}_k} g_i(\alpha_i + (\Delta\boldsymbol{\alpha}_{[k]})_i) \right) \\
\overset{(10)}{=}\ & (1-\gamma)\mathcal{O}_A(\boldsymbol{\alpha}) + \gamma \sum_{k=1}^{K} \mathcal{G}_k^{\sigma'}(\Delta\boldsymbol{\alpha}_{[k]}; \mathbf{v}, \boldsymbol{\alpha}_{[k]}) ,
\end{aligned}$$

where the last equality is by the definition of the subproblem objective $\mathcal{G}_k^{\sigma'}(.)$ as in (10). □

D.2 Proof of Main Convergence Result (Theorem 2)

Before proving the main convergence results, we introduce several useful quantities, and establish the following lemma, which characterizes the effect of iterations of Algorithm 1 on the duality gap for any chosen local solver of approximation quality Θ.

Lemma 7. *Let g_i be strongly convex [2] with convexity parameter $\mu \geq 0$ with respect to the norm $\|\cdot\|$, $\forall i \in [n]$. Then at each iteration of Algorithm 1 under Assumption 1, and any $s \in [0,1]$, it holds that*

$$\mathbb{E}[\mathcal{O}_A(\boldsymbol{\alpha}^{(t)}) - \mathcal{O}_A(\boldsymbol{\alpha}^{(t+1)})] \geq \gamma(1-\Theta)\Big(sG(\boldsymbol{\alpha}^{(t)}) - \frac{\sigma' s^2}{2\tau} R^{(t)} \Big) , \tag{32}$$

2. Note that the case of weakly convex $g_i(.)$ is explicitly allowed here as well, as the Lemma holds for the case $\mu = 0$.

where

$$R^{(t)} := -\tfrac{\tau\mu(1-s)}{\sigma' s} \|\mathbf{u}^{(t)} - \boldsymbol{\alpha}^{(t)}\|^2 + \textstyle\sum_{k=1}^{K} \|A_{[k]}(\mathbf{u}^{(t)} - \boldsymbol{\alpha}^{(t)})_{[k]}\|^2 , \tag{33}$$

for $\mathbf{u}^{(t)} \in \mathbb{R}^n$ with

$$u_i^{(t)} \in \partial g_i^*(-\mathbf{x}_i^\top \mathbf{w}(\boldsymbol{\alpha}^{(t)})) . \tag{34}$$

Proof. This proof is motivated by Shalev-Shwartz and Zhang (2013a, Lemma 19) and follows Ma et al. (2015a, Lemma 5), with a difference being the extension to our generalized subproblems $\mathcal{G}_k^{\sigma'}(\cdot; \mathbf{v}, \boldsymbol{\alpha}_{[k]})$ along with the mappings $\mathbf{w}(\boldsymbol{\alpha}) := \nabla f(\mathbf{v}(\boldsymbol{\alpha}))$ with $\mathbf{v}(\boldsymbol{\alpha}) := A\boldsymbol{\alpha}$.

For simplicity, we write $\boldsymbol{\alpha}$ instead of $\boldsymbol{\alpha}^{(t)}$, $\mathbf{v}$ instead of $\mathbf{v}(\boldsymbol{\alpha}^{(t)})$, $\mathbf{w}$ instead of $\mathbf{w}(\boldsymbol{\alpha}^{(t)})$ and $\mathbf{u}$ instead of $\mathbf{u}^{(t)}$. We can estimate the expected change of the objective $\mathcal{O}_A(\boldsymbol{\alpha})$ as follows. Starting from the definition of the update $\boldsymbol{\alpha}^{(t+1)} := \boldsymbol{\alpha}^{(t)} + \gamma \sum_k \Delta\boldsymbol{\alpha}_{[k]}$ from Algorithm 1, we apply Lemma 1, which relates the local approximation $\mathcal{G}_k^{\sigma'}(\boldsymbol{\alpha}; \mathbf{v}, \boldsymbol{\alpha}_{[k]})$ to the global objective $\mathcal{O}_A(\boldsymbol{\alpha})$, and then bound this using the notion of quality of the local solver (Θ), as in Assumption 1. This gives us:

$$\begin{aligned}
\mathbb{E}\big[\mathcal{O}_A(\boldsymbol{\alpha}^{(t)}) - \mathcal{O}_A(\boldsymbol{\alpha}^{(t+1)})\big] &= \mathbb{E}\Big[\mathcal{O}_A(\boldsymbol{\alpha}) - \mathcal{O}_A\Big(\boldsymbol{\alpha} + \gamma \sum_{k=1}^{K} \Delta\boldsymbol{\alpha}_{[k]}\Big)\Big] \\
&\geq \gamma(1-\Theta) \left(\underbrace{\mathcal{O}_A(\boldsymbol{\alpha}) - \sum_{k=1}^{K} \mathcal{G}_k^{\sigma'}(\Delta\boldsymbol{\alpha}_{[k]}^\star; \mathbf{v}, \boldsymbol{\alpha}_{[k]})}_{C} \right) .
\end{aligned} \tag{35}$$

We next upper bound the C term, denoting $\Delta\boldsymbol{\alpha}^\star = \sum_{k=1}^{K} \Delta\boldsymbol{\alpha}_{[k]}^\star$. We first plug in the definition of the objective $\mathcal{O}_A$ in (A) and the local subproblems (10), and then substitute $s(u_i - \alpha_i)$ for $\Delta\alpha_i^\star$ and apply the μ-strong convexity of the g_i terms. This gives us:

$$\begin{aligned}
C &= \sum_{i=1}^{n} (g_i(\alpha_i) - g_i(\alpha_i + \Delta\alpha_i^\star)) - (A\Delta\boldsymbol{\alpha}^\star)^\top \mathbf{w}(\boldsymbol{\alpha}) - \sum_{k=1}^{K} \frac{\sigma'}{2\tau} \Big\| A_{[k]} \Delta\boldsymbol{\alpha}_{[k]}^\star \Big\|^2 \\
&= \sum_{i=1}^{n} (g_i(\alpha_i) - g_i(su_i + (1-s)\alpha_i)) - A(s(\mathbf{u} - \boldsymbol{\alpha}))^\top \mathbf{w}(\boldsymbol{\alpha}) - \sum_{k=1}^{K} \frac{\sigma'}{2\tau} \Big\| A_{[k]} s(\mathbf{u} - \boldsymbol{\alpha})_{[k]} \Big\|^2 \\
&\geq \sum_{i=1}^{n} \Big(sg_i(\alpha_i) - sg_i(u_i) + \frac{\mu}{2}(1-s)s(u_i - \alpha_i)^2 \Big) \\
&\quad - A(s(\mathbf{u} - \boldsymbol{\alpha}))^\top \mathbf{w}(\boldsymbol{\alpha}) - \sum_{k=1}^{K} \frac{\sigma'}{2\tau} \Big\| A_{[k]} (s(\mathbf{u} - \boldsymbol{\alpha})_{[k]}) \Big\|^2 .
\end{aligned} \tag{36}$$

From the definition of the optimization problems (A) and (B), and definition of convex conjugates, we can write the duality gap as:

$$\begin{aligned} G(\boldsymbol{\alpha}) := \mathcal{O}_A(\boldsymbol{\alpha}) - (-\mathcal{O}_B(\mathbf{w}(\boldsymbol{\alpha})) &\overset{\text{(A),(B)}}{=} \sum_{i=1}^n \left(g_i^*(-\mathbf{x}_i^\top \mathbf{w}(\boldsymbol{\alpha})) + g_i(\alpha_i) \right) + f^*(\mathbf{w}(\boldsymbol{\alpha})) + f(A\boldsymbol{\alpha})) \\ &= \sum_{i=1}^n \left(g_i^*(-\mathbf{x}_i^\top \mathbf{w}(\boldsymbol{\alpha})) + g_i(\alpha_i) \right) + (A\boldsymbol{\alpha})^\top \mathbf{w}(\boldsymbol{\alpha}) \\ &= \sum_{i=1}^n \left(g_i^*(-\mathbf{x}_i^\top \mathbf{w}(\boldsymbol{\alpha})) + g_i(\alpha_i) + \alpha_i \mathbf{x}_i^\top \mathbf{w}(\boldsymbol{\alpha}) \right) . \end{aligned} \tag{37}$$

The convex conjugate maximal property from (34) implies that

$$g_i(u_i) = u_i(-\mathbf{x}_i^\top \mathbf{w}(\boldsymbol{\alpha})) - g_i^*(-\mathbf{x}_i^\top \mathbf{w}(\boldsymbol{\alpha})) . \tag{38}$$

Using (38) and (37), we therefore have:

$$\begin{aligned} C &\overset{(38)}{\geq} \sum_{i=1}^n \left(s g_i(\alpha_i) - s u_i(-\mathbf{x}_i^\top \mathbf{w}(\boldsymbol{\alpha})) + s g_i^*(-\mathbf{x}_i^\top \mathbf{w}(\boldsymbol{\alpha})) + \frac{\mu}{2}(1-s)s(u_i - \alpha_i)^2 \right) \\ &\quad - A(s(\mathbf{u}-\boldsymbol{\alpha}))^\top \mathbf{w}(\boldsymbol{\alpha}) - \sum_{k=1}^K \frac{\sigma'}{2\tau} \left\| A_{[k]}(s(\mathbf{u}-\boldsymbol{\alpha})_{[k]}) \right\|^2 \\ &= \sum_{i=1}^n \left[s g_i(\alpha_i) + s g_i^*(-\mathbf{x}_i^\top \mathbf{w}(\boldsymbol{\alpha})) + s \mathbf{x}_i^\top \mathbf{w}(\boldsymbol{\alpha}) \alpha_i \right] - \sum_{i=1}^n \left[s \mathbf{x}_i^\top \mathbf{w}(\boldsymbol{\alpha})(\alpha_i - u_i) - \frac{\mu}{2}(1-s)s(u_i - \alpha_i)^2 \right] \\ &\quad - A(s(\mathbf{u}-\boldsymbol{\alpha}))^\top \mathbf{w}(\boldsymbol{\alpha}) - \sum_{k=1}^K \frac{\sigma'}{2\tau} \left\| A_{[k]}(s(\mathbf{u}-\boldsymbol{\alpha})_{[k]}) \right\|^2 \\ &\overset{(37)}{=} s G(\boldsymbol{\alpha}) + \frac{\mu}{2}(1-s)s\|\mathbf{u}-\boldsymbol{\alpha}\|^2 - \frac{\sigma' s^2}{2\tau} \sum_{k=1}^K \|A_{[k]}(\mathbf{u}-\boldsymbol{\alpha})_{[k]}\|^2 . \end{aligned} \tag{39}$$

The claimed improvement bound (32) then follows by plugging (39) into (35). □

The following Lemma provides a uniform bound on $R^{(t)}$.

Lemma 8. *If g_i^* are L-Lipschitz continuous for all $i \in [n]$, then*

$$\forall t : R^{(t)} \leq 4L^2 \underbrace{\sum_{k=1}^K \sigma_k n_k}_{=:\sigma} , \tag{40}$$

where

$$\sigma_k := \max_{\boldsymbol{\alpha}_{[k]} \in \mathbb{R}^n} \frac{\|A_{[k]}\boldsymbol{\alpha}_{[k]}\|^2}{\|\boldsymbol{\alpha}_{[k]}\|^2} . \tag{41}$$

Proof. (Ma et al., 2015a, Lemma 6). For general convex functions, the strong convexity parameter is $\mu = 0$, and hence the definition (33) of the complexity constant $R^{(t)}$ becomes

$$R^{(t)} = \sum_{k=1}^K \|A_{[k]}(\mathbf{u}^{(t)} - \boldsymbol{\alpha}^{(t)})_{[k]}\|^2 \overset{(41)}{\leq} \sum_{k=1}^K \sigma_k \|(\mathbf{u}^{(t)} - \boldsymbol{\alpha}^{(t)})_{[k]}\|^2 \leq \sum_{k=1}^K \sigma_k |\mathcal{P}_k| 4L^2 .$$

Here the last inequality follows from (Shalev-Shwartz and Zhang, 2013a, Lemma 21), which shows that for $g_i^* : \mathbb{R} \to \mathbb{R}$ being L-Lipschitz, it holds that for any real value a with $|a| > L$ one has that $g_i(a) = +\infty$. □

Remark 6. *(Ma et al., 2015a, Remark 7) If the data points $\mathbf{x}_i$ are normalized such that $\|\mathbf{x}_i\| \leq 1$, $\forall i \in [n]$, then $\sigma_k \leq |\mathcal{P}_k| = n_k$. Furthermore, if we assume that the data partition is balanced, i.e., that $n_k = n/K$ for all k, then $\sigma \leq n^2/K$. This can be used to bound the constants $R^{(t)}$, above, as $R^{(t)} \leq \frac{4L^2 n^2}{K}$.*

Theorem 9. *Consider Algorithm 1, using a local solver of quality Θ (See Assumption 1). Let $g_i^*(\cdot)$ be L-Lipschitz continuous, and $\epsilon_G > 0$ be the desired duality gap (and hence an upper-bound on suboptimality $\epsilon_{\mathcal{O}_A}$). Then after T iterations, where*

$$T \geq T_0 + \max\{ \left\lceil \frac{1}{\gamma(1-\Theta)} \right\rceil, \left\lceil \frac{4L^2 \sigma\sigma'}{\tau \epsilon_G \gamma(1-\Theta)} \right\rceil \} , \tag{42}$$

$$T_0 \geq t_0 + \left\lceil \frac{2}{\gamma(1-\Theta)} \left(\frac{8L^2\sigma\sigma'}{\tau\epsilon_G} - 1 \right) \right\rceil_+ , \quad t_0 \geq \max(0, \left\lceil \tfrac{1}{\gamma(1-\Theta)} \log\left(\tfrac{\tau(\mathcal{O}_A(\boldsymbol{\alpha}^{(0)}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star))}{2L^2\sigma\sigma'} \right) \right\rceil) ,$$

we have that the expected duality gap satisfies

$$\mathbb{E}[\mathcal{O}_A(\overline{\boldsymbol{\alpha}}) - (-\mathcal{O}_B(\mathbf{w}(\overline{\boldsymbol{\alpha}})))] \leq \epsilon_G$$

at the averaged iterate

$$\overline{\boldsymbol{\alpha}} := \tfrac{1}{T-T_0} \textstyle\sum_{t=T_0}^{T-1} \boldsymbol{\alpha}^{(t)} . \tag{43}$$

Proof. We begin by estimating the expected change of feasibility for $\mathcal{O}_A$. We can bound this above by using Lemma 7 and the fact that the $\mathcal{O}_B(\cdot)$ is always a lower bound for $-\mathcal{O}_A(\cdot)$, and then applying (40) to find:

$$\begin{aligned} \mathbb{E}[\mathcal{O}_A(\boldsymbol{\alpha}^{(t+1)}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star)] &= \mathbb{E}[-\mathcal{O}_A(\boldsymbol{\alpha}^\star) + \mathcal{O}_A(\boldsymbol{\alpha}^{(t+1)}) - \mathcal{O}_A(\boldsymbol{\alpha}^{(t)}) + \mathcal{O}_A(\boldsymbol{\alpha}^{(t)})] \\ &\leq (1 - \gamma(1-\Theta)s) \left(\mathcal{O}_A(\boldsymbol{\alpha}^{(t)}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star) \right) + \gamma(1-\Theta)\tfrac{\sigma' s^2}{2\tau} 4L^2\sigma . \end{aligned} \tag{44}$$

Using (44) recursively we have

$$\mathbb{E}[\mathcal{O}_A(\boldsymbol{\alpha}^{(t)}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star)] \leq (1 - \gamma(1-\Theta)s)^t \left(\mathcal{O}_A(\boldsymbol{\alpha}^{(0)}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star) \right) + s\frac{4L^2\sigma\sigma'}{2\tau} . \tag{45}$$

Choosing $s = 1$ and $t = t_0 := \max\{0, \lceil \frac{1}{\gamma(1-\Theta)} \log(2(\mathcal{O}_A(\boldsymbol{\alpha}^{(0)}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star))/(4L^2\sigma\sigma')) \rceil\}$ leads to

$$\mathbb{E}[\mathcal{O}_A(\boldsymbol{\alpha}^{(t)}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star)] \leq (1 - \gamma(1-\Theta))^{t_0} \left(\mathcal{O}_A(\boldsymbol{\alpha}^{(0)}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star) \right) + \frac{4L^2\sigma\sigma'}{2\tau} \leq \frac{4L^2\sigma\sigma'}{\tau} . \tag{46}$$

Next, we show inductively that

$$\forall t \geq t_0 : \mathbb{E}[\mathcal{O}_A(\boldsymbol{\alpha}^{(t)}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star)] \leq \frac{4L^2\sigma\sigma'}{\tau(1+\frac{1}{2}\gamma(1-\Theta)(t-t_0))}\,. \tag{47}$$

Clearly, (46) implies that (47) holds for $t = t_0$. Assuming that it holds for any $t \geq t_0$, we show that it must also hold for $t+1$. Indeed, using

$$s = \frac{1}{1+\frac{1}{2}\gamma(1-\Theta)(t-t_0)} \in [0,1]\,, \tag{48}$$

we obtain

$$\mathbb{E}[\mathcal{O}_A(\boldsymbol{\alpha}^{(t+1)}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star)] \leq \frac{4L^2\sigma\sigma'}{\tau}\underbrace{\left(\frac{1+\frac{1}{2}\gamma(1-\Theta)(t-t_0)-\frac{1}{2}\gamma(1-\Theta)}{(1+\frac{1}{2}\gamma(1-\Theta)(t-t_0))^2}\right)}_{D}\,,$$

by applying the bounds (44) and (47), plugging in the definition of s (48), and simplifying. We upper bound the term D using the fact that geometric mean is less or equal to arithmetic mean:

$$\begin{aligned} D &= \frac{1}{1+\frac{1}{2}\gamma(1-\Theta)(t+1-t_0)}\underbrace{\frac{(1+\frac{1}{2}\gamma(1-\Theta)(t+1-t_0))(1+\frac{1}{2}\gamma(1-\Theta)(t-1-t_0))}{(1+\frac{1}{2}\gamma(1-\Theta)(t-t_0))^2}}_{\leq 1} \\ &\leq \frac{1}{1+\frac{1}{2}\gamma(1-\Theta)(t+1-t_0)}. \end{aligned}$$

If $\overline{\boldsymbol{\alpha}}$ is defined as (43), we apply the results of Lemma 7 and Lemma 8 to obtain

$$\begin{aligned} \mathbb{E}[G(\overline{\boldsymbol{\alpha}})] &= \mathbb{E}\left[G\left(\sum_{t=T_0}^{T-1}\tfrac{1}{T-T_0}\boldsymbol{\alpha}^{(t)}\right)\right] \leq \tfrac{1}{T-T_0}\mathbb{E}\left[\sum_{t=T_0}^{T-1} G\left(\boldsymbol{\alpha}^{(t)}\right)\right] \\ &\leq \frac{1}{\gamma(1-\Theta)s}\frac{1}{T-T_0}\mathbb{E}\left[\mathcal{O}_A(\boldsymbol{\alpha}^{(T_0)}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star)\right] + \tfrac{4L^2\sigma\sigma' s}{2\tau}\,. \end{aligned} \tag{49}$$

If $T \geq \lceil \frac{1}{\gamma(1-\Theta)} \rceil + T_0$ such that $T_0 \geq t_0$ we have

$$\begin{aligned} \mathbb{E}[G(\overline{\boldsymbol{\alpha}})] &\overset{(49),(47)}{\leq} \frac{1}{\gamma(1-\Theta)s}\frac{1}{T-T_0}\left(\frac{4L^2\sigma\sigma'}{\tau(1+\frac{1}{2}\gamma(1-\Theta)(T_0-t_0))}\right) + \frac{4L^2\sigma\sigma' s}{2\tau} \\ &= \frac{4L^2\sigma\sigma'}{\tau}\left(\frac{1}{\gamma(1-\Theta)s}\frac{1}{T-T_0}\frac{1}{1+\frac{1}{2}\gamma(1-\Theta)(T_0-t_0)} + \frac{s}{2}\right). \end{aligned} \tag{50}$$

Choosing

$$s = \frac{1}{(T-T_0)\gamma(1-\Theta)} \in [0,1] \tag{51}$$

gives us

$$\mathbb{E}[G(\overline{\boldsymbol{\alpha}})] \overset{(50),(51)}{\leq} \frac{4L^2\sigma\sigma'}{\tau}\left(\frac{1}{1+\frac{1}{2}\gamma(1-\Theta)(T_0-t_0)} + \frac{1}{(T-T_0)\gamma(1-\Theta)}\frac{1}{2}\right). \tag{52}$$

To have right hand side of (52) smaller then ϵ_G it is sufficient to choose T_0 and T such that

$$\frac{4L^2\sigma\sigma'}{\tau}\left(\frac{1}{1+\frac{1}{2}\gamma(1-\Theta)(T_0-t_0)}\right) \leq \frac{1}{2}\epsilon_G\,, \tag{53}$$

$$\frac{4L^2\sigma\sigma'}{\tau}\left(\frac{1}{(T-T_0)\gamma(1-\Theta)}\frac{1}{2}\right) \leq \frac{1}{2}\epsilon_G\,. \tag{54}$$

Hence if $T_0 \geq t_0 + \frac{2}{\gamma(1-\Theta)}\left(\frac{8L^2\sigma\sigma'}{\tau\epsilon_G} - 1\right)$ and $T \geq T_0 + \frac{4L^2\sigma\sigma'}{\tau\epsilon_G\gamma(1-\Theta)}$ then (53) and (54) are satisfied. □

The following main theorem simplifies the results of Theorem 9 and is a generalization of Ma et al. (2015a, Corollary 9) for general $f^*(\cdot)$ functions:

Theorem' 2. *Consider Algorithm 1 with $\gamma := 1$, using a local solver of quality Θ (see Assumption 1). Let $g_i^*(\cdot)$ be L-Lipschitz continuous, and assume that the columns of A satisfy $\|\mathbf{x}_i\| \leq 1$, $\forall i \in [n]$ and g_i^* is of the form $\frac{1}{n}g_i^*$, as is common in ERM-type problems. Let $\epsilon_G > 0$ be the desired duality gap (and hence an upper-bound on primal sub-optimality). Then after T iterations, where*

$$\begin{aligned} T &\geq T_0 + \max\{\left\lceil \frac{1}{1-\Theta}\right\rceil, \frac{4L^2}{\tau\epsilon_G(1-\Theta)}\}\,, \\ T_0 &\geq t_0 + \left\lceil \frac{2}{1-\Theta}\left(\frac{8L^2}{\tau\epsilon_G} - 1\right)\right\rceil_+\,, \\ t_0 &\geq \max(0, \left\lceil \tfrac{1}{(1-\Theta)}\log\left(\tfrac{\tau n(\mathcal{O}_A(\boldsymbol{\alpha}^{(0)}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star))}{2L^2K}\right)\right\rceil)\,, \end{aligned} \tag{55}$$

we have that the expected duality gap satisfies

$$\mathbb{E}[\mathcal{O}_A(\overline{\boldsymbol{\alpha}}) - (-\mathcal{O}_B(\mathbf{w}(\overline{\boldsymbol{\alpha}})))] \leq \epsilon_G\,,$$

where $\overline{\boldsymbol{\alpha}}$ is the averaged iterate returned by Algorithm 1.

Proof. Plug in parameters $\gamma := 1$, $\sigma' := \gamma K = K$, $\tilde{L} := \frac{1}{n}L$ to the results of Theorem 9, and note that for balanced datasets with $g_i^* := \frac{1}{n}g_i^*$ we have $\sigma \leq \frac{n}{K}$ (see Remark 6). We can further simplify the rate by noting that $\tau = 1$ for the 1-smooth losses (least squares and logistic) given as examples in this work. □

D.3 Proof of Convergence Result for Strongly Convex g_i

Our second main theorem follows reasoning in Shalev-Shwartz and Zhang (2013a) and is a generalization of Ma et al. (2015a, Corollary 11). We first introduce a lemma to simplify the proof.

Lemma 10. *Assume that $g_i(0) \in [0,1]$ for all $i \in [n]$, then for the zero vector $\boldsymbol{\alpha}^{(0)} := \mathbf{0} \in \mathbb{R}^n$, we have*

$$\mathcal{O}_A(\boldsymbol{\alpha}^{(0)}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star) = \mathcal{O}_A(\mathbf{0}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star) \leq n\,. \tag{56}$$

Proof. Since $-\mathcal{O}_A(\cdot)$ is always a lower bound on $\mathcal{O}_B(\cdot)$, and by definition of the objectives $\mathcal{O}_A$ and $\mathcal{O}_B$ given in (A) and (B) respectively, for $\boldsymbol{\alpha} := \mathbf{0} \in \mathbb{R}^n$,

$$\begin{aligned} 0 \leq \mathcal{O}_A(\boldsymbol{\alpha}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star) \leq \mathcal{O}_A(\mathbf{0}) - (-\mathcal{O}_B(\mathbf{w}(\mathbf{0}))) &\overset{(A),(B)}{=} \\ \overset{(A),(B)}{=} f(\mathbf{0}) + f^*(\mathbf{w}(\mathbf{0})) + g(\mathbf{0}) + g^*(-A^\top \mathbf{w}(\mathbf{0}))\,. \end{aligned}$$

Since $f^*(\mathbf{w}(\mathbf{0})) = f^*(\nabla f(\mathbf{0})) = \mathbf{0}^\top \nabla f(\mathbf{0}) - f(\mathbf{0}) = -f(\mathbf{0})$, and given our initial assumption on $g(\mathbf{0})$, the duality gap reduces to:

$$0 \leq g(\mathbf{0}) + g^*(-A^\top \mathbf{w}(\mathbf{0})) \leq n. \qquad \square$$

Theorem 11. *Assume that g_i are μ-strongly convex $\forall i \in [n]$. We define $\sigma_{\max} = \max_{k\in[K]} \sigma_k$. Then after T iterations of Algorithm 1, with*

$$T \geq \tfrac{1}{\gamma(1-\Theta)} \tfrac{\mu\tau + \sigma_{\max}\sigma'}{\mu\tau} \log \tfrac{n}{\epsilon_{\mathcal{O}_A}}\,,$$

it holds that

$$\mathbb{E}[\mathcal{O}_A(\boldsymbol{\alpha}^{(T)}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star)] \leq \epsilon_{\mathcal{O}_A}\,.$$

Furthermore, after T iterations with

$$T \geq \tfrac{1}{\gamma(1-\Theta)} \tfrac{\mu\tau + \sigma_{\max}\sigma'}{\mu\tau} \log\left(\tfrac{1}{\gamma(1-\Theta)} \tfrac{\mu\tau + \sigma_{\max}\sigma'}{\mu\tau} \tfrac{n}{\epsilon_G}\right)\,,$$

we have the expected duality gap

$$\mathbb{E}[\mathcal{O}_A(\boldsymbol{\alpha}^{(T)}) - (-\mathcal{O}_B(\mathbf{w}(\boldsymbol{\alpha}^{(T)})))] \leq \epsilon_G\,.$$

Proof. Given that $g_i(.)$ is μ-strongly convex, we can apply (33) and the definition of σ_k to find:

$$\begin{aligned} R^{(t)} &\leq -\tfrac{\tau\mu(1-s)}{\sigma' s}\|\mathbf{u}^{(t)} - \boldsymbol{\alpha}^{(t)}\|^2 + \sum\nolimits_{k=1}^K \sigma_k \|\mathbf{u}^{(t)} - \boldsymbol{\alpha}_{[k]}^{(t)}\|^2 \\ &\leq \left(-\tfrac{\tau\mu(1-s)}{\sigma' s} + \sigma_{\max}\right)\|\mathbf{u}^{(t)} - \boldsymbol{\alpha}^{(t)}\|^2\,, \end{aligned} \tag{57}$$

where $\sigma_{\max} = \max_{k\in[K]} \sigma_k$. If we plug the following value of s

$$s = \frac{\tau\mu}{\tau\mu + \sigma_{\max}\sigma'} \in [0,1] \tag{58}$$

into (57) we obtain that $\forall t : R^{(t)} \leq 0$. Putting the same s into (32) will give us

$$\begin{aligned} \mathbb{E}[\mathcal{O}_A(\boldsymbol{\alpha}^{(t)}) - \mathcal{O}_A(\boldsymbol{\alpha}^{(t+1)})] &\overset{(32),(58)}{\geq} \gamma(1-\Theta)\frac{\tau\mu}{\tau\mu + \sigma_{\max}\sigma'} G(\boldsymbol{\alpha}^{(t)}) \\ &\geq \gamma(1-\Theta)\frac{\tau\mu}{\tau\mu + \sigma_{\max}\sigma'}(\mathcal{O}_A(\boldsymbol{\alpha}^{(t)}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star))\,. \end{aligned} \tag{59}$$

Using the fact that $\mathbb{E}[\mathcal{O}_A(\boldsymbol{\alpha}^{(t)}) - \mathcal{O}_A(\boldsymbol{\alpha}^{(t+1)})] = \mathbb{E}[\mathcal{O}_A(\boldsymbol{\alpha}^\star) - \mathcal{O}_A(\boldsymbol{\alpha}^{(t+1)})] + \mathcal{O}_A(\boldsymbol{\alpha}^{(t)}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star)$ we have

$$\mathbb{E}[\mathcal{O}_A(\boldsymbol{\alpha}^\star) - \mathcal{O}_A(\boldsymbol{\alpha}^{(t+1)})] + \mathcal{O}_A(\boldsymbol{\alpha}^{(t)}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star) \overset{(59)}{\geq} \gamma(1-\Theta)\frac{\tau\mu}{\tau\mu + \sigma_{\max}\sigma'}(\mathcal{O}_A(\boldsymbol{\alpha}^{(t)}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star))\,,$$

which is equivalent to

$$\mathbb{E}[\mathcal{O}_A(\boldsymbol{\alpha}^{(t+1)}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star)] \leq \left(1 - \gamma(1-\Theta)\frac{\tau\mu}{\tau\mu + \sigma_{\max}\sigma'}\right)(\mathcal{O}_A(\boldsymbol{\alpha}^{(t)}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star))\,. \tag{60}$$

Therefore if we denote $\epsilon_{\mathcal{O}_A}^{(t)} = \mathcal{O}_A(\boldsymbol{\alpha}^{(t)}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star)$ we have recursively that

$$\begin{aligned} \mathbb{E}[\epsilon_{\mathcal{O}_A}^{(t)}] &\overset{(60)}{\leq} \left(1 - \gamma(1-\Theta)\frac{\tau\mu}{\tau\mu + \sigma_{\max}\sigma'}\right)^t \epsilon_{\mathcal{O}_A}^{(0)} \overset{(56)}{\leq} \left(1 - \gamma(1-\Theta)\frac{\tau\mu}{\tau\mu + \sigma_{\max}\sigma'}\right)^t n \\ &\leq \exp\left(-t\gamma(1-\Theta)\frac{\tau\mu}{\tau\mu + \sigma_{\max}\sigma'}\right) n\,. \end{aligned}$$

The right hand side will be smaller than some $\epsilon_{\mathcal{O}_A}$ if

$$t \geq \frac{1}{\gamma(1-\Theta)}\frac{\tau\mu + \sigma_{\max}\sigma'}{\tau\mu}\log\frac{n}{\epsilon_{\mathcal{O}_A}}\,.$$

Moreover, to bound the duality gap, we have

$$\gamma(1-\Theta)\frac{\tau\mu}{\tau\mu + \sigma_{\max}\sigma'} G(\boldsymbol{\alpha}^{(t)}) \overset{(59)}{\leq} \mathbb{E}[\mathcal{O}_A(\boldsymbol{\alpha}^{(t)}) - \mathcal{O}_A(\boldsymbol{\alpha}^{(t+1)})] \leq \mathbb{E}[\mathcal{O}_A(\boldsymbol{\alpha}^{(t)}) - \mathcal{O}_A(\boldsymbol{\alpha}^\star)]\,.$$

Thus, $G(\boldsymbol{\alpha}^{(t)}) \leq \frac{1}{\gamma(1-\Theta)}\frac{\tau\mu + \sigma_{\max}\sigma'}{\tau\mu}\epsilon_{\mathcal{O}_A}^{(t)}$. Hence if $\epsilon_{\mathcal{O}_A} \leq \gamma(1-\Theta)\frac{\tau\mu}{\tau\mu + \sigma_{\max}\sigma'}\epsilon_G$ then $G(\boldsymbol{\alpha}^{(t)}) \leq \epsilon_G$. Therefore after

$$t \geq \frac{1}{\gamma(1-\Theta)}\frac{\tau\mu + \sigma_{\max}\sigma'}{\tau\mu}\log\left(\frac{1}{\gamma(1-\Theta)}\frac{\tau\mu + \sigma_{\max}\sigma'}{\tau\mu}\frac{n}{\epsilon_G}\right)$$

iterations we have obtained a duality gap less than ϵ_G. $\square$

Theorem' 3. *Consider Algorithm 1 with $\gamma := 1$, using a local solver of quality Θ (see Assumption 1). Let $g_i(\cdot)$ be μ-strongly convex, $\forall i \in [n]$, and assume that the columns of A satisfy $\|\mathbf{x}_i\| \leq 1\ \forall i \in [n]$ and and g_i^* is of the form $\frac{1}{n}g_i^*$, as is common in ERM-type problems. Then we have that T iterations are sufficient for suboptimality $\epsilon_{\mathcal{O}_A}$, with*

$$T \geq \tfrac{1}{(1-\Theta)}\tfrac{\tau\mu+1}{\tau\mu}\log\tfrac{1}{\epsilon_{\mathcal{O}_A}}\,.$$

Furthermore, after T iterations with

$$T \geq \tfrac{1}{(1-\Theta)}\tfrac{\tau\mu+1}{\tau\mu}\log\left(\tfrac{1}{(1-\Theta)}\tfrac{\tau\mu+1}{\tau\mu}\tfrac{1}{\epsilon_G}\right)\,,$$

we have the expected duality gap

$$\mathbb{E}[\mathcal{O}_A(\boldsymbol{\alpha}^{(T)}) - (-\mathcal{O}_B(\mathbf{w}(\boldsymbol{\alpha}^{(T)})))] \leq \epsilon_G\,.$$

Proof. Plug in parameters $\gamma := 1$, $\sigma' := \gamma K = K$, $\tilde{\mu} = n\mu$ to the results of Theorem 11 and note that for balanced datasets with $g_i^* := \frac{1}{n}g_i^*$ we have $\sigma_{\max} \leq \frac{n}{K}$ (see Remark 6). We can further simplify the rate by noting that $\tau = 1$ for the 1-smooth losses (least squares and logistic) given as examples in this work. $\square$

References

G. Andrew and J. Gao. Scalable training of L1-regularized log-linear models. In *International Conference on Machine Learning*, 2007.

Y. Arjevani and O. Shamir. Communication complexity of distributed convex learning and optimization. In *Neural Information Processing Systems*, 2015.

M.-F. Balcan, A. Blum, S. Fine, and Y. Mansour. Distributed learning, communication complexity and privacy. In *Conference on Learning Theory*, 2012.

H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer Science & Business Media, New York, NY, 2011.

D. P. Bersekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall, Englewood Cliffs, NJ, 1989.

Y. Bian, X. Li, Y. Liu, and M.-H. Yang. Parallel coordinate descent Newton method for efficient ℓ_1-regularized minimization. *arXiv.org*, 2013.

J. M. Borwein and Q. Zhu. *Techniques of Variational Analysis*. Springer Science & Buisness Media, New York, NY, 2005.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010.

J. K. Bradley, A. Kyrola, D. Bickson, and C. Guestrin. Parallel coordinate descent for l1-regularized loss minimization. In *International Conference on Machine Learning*, 2011.

A. Défossez and F. Bach. Adabatch: Efficient gradient aggregation rules for sequential and parallel stochastic gradient methods. *arXiv.org*, 2017.

O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao. Optimal Distributed Online Prediction Using Mini-Batches. *Journal of Machine Learning Research*, 13:165–202, 2012.

J. Duchi, M. I. Jordan, and B. McMahan. Estimation, optimization, and parallelism when data is sparse. *Neural Information Processing Systems*, 2013.

C. Dünner, S. Forte, M. Takáč, and M. Jaggi. Primal-dual rates and certificates. In *International Conference on Machine Learning*, 2016.

C. Dünner, T. Parnell, and M. Jaggi. Efficient use of limited-memory accelerators for linear learning on heterogeneous systems. In *Neural Information Processing Systems*, 2017.

C. Dünner, A. Lucchi, M. Gargiani, A. Bian, T. Hofmann, and M. Jaggi. A Distributed Second-Order Algorithm You Can Trust. In *International Conference on Machine Learning*, 2018.

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

O. Fercoq and P. Richtárik. Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015.

S. Forte. Distributed Optimization for Non-Strongly Convex Regularizers. Master's thesis, ETH Zürich, 2015.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

M. Gargiani. Hessian-CoCoA: a general parallel and distributed framework for non-strongly convex regularizers. Master's thesis, ETH Zurich, 2017.

C. Heinze, B. McWilliams, and N. Meinshausen. DUAL-LOCO: Distributing statistical estimation using random projections. In *International Conference on Artificial Intelligence and Statistics*, 2016.

J.-B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of convex analysis*. Springer–Verlag, Berlin, 2001.

M. Jaggi, V. Smith, M. Takáč, J. Terhorst, S. Krishnan, T. Hofmann, and M. I. Jordan. Communication-efficient distributed dual coordinate ascent. In *Neural Information Processing Systems*, 2014.

T. Johnson and C. Guestrin. Blitz: A principled meta-algorithm for scaling sparse optimization. In *International Conference on Machine Learning*, 2015.

H. Karimi, J. Nutini, and M. Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak-łojasiewicz condition. In *European Conference on Machine Learning*, 2016.

S. P. Karimireddy, S. U. Stich, and M. Jaggi. Adaptive balancing of gradient and update computation times using global geometry and approximate subproblems. In *International Conference on Artificial Intelligence and Statistics*, 2018a.

S. P. Karimireddy, S. U. Stich, and M. Jaggi. Global linear convergence of Newton's method without strong-convexity or Lipschitz gradients. *arXiv.org*, 2018b.

C.-p. Lee and K.-W. Chang. Distributed block-diagonal approximation methods for regularized empirical risk minimization. *arXiv.org*, 2017.

C.-P. Lee and D. Roth. Distributed box-constrained quadratic optimization for dual linear SVM. In *International Conference on Machine Learning*, 2015.

C.-p. Lee, C. H. Lim, and S. J. Wright. A distributed quasi-newton algorithm for empirical risk minimization with nonsmooth regularization. In *ACM International Conference on Knowledge Discovery and Data Mining*, 2018.

Z. Lu and L. Xiao. On the complexity analysis of randomized block-coordinate descent methods. *arXiv.org*, 2013.

C. Ma, V. Smith, M. Jaggi, M. I. Jordan, P. Richtárik, and M. Takáč. Adding vs. averaging in distributed primal-dual optimization. In *International Conference on Machine Learning*, 2015a.

C. Ma, R. Tappenden, and M. Takáč. Linear convergence of the randomized feasible descent method under the weak strong convexity assumption. *arXiv.org*, 2015b.

C. Ma, M. Jaggi, F. E. Curtis, N. Srebro, and M. Takáč. An accelerated communication-efficient primal-dual optimization framework for structured machine learning. *arXiv.org*, 2017a.

C. Ma, J. Konecny, M. Jaggi, V. Smith, M. I. Jordan, P. Richtárik, and M. Takáč. Distributed optimization with arbitrary local solvers. *Optimization Methods and Software*, Feb. 2017b.

D. Mahajan, S. S. Keerthi, and S. Sundararajan. A distributed block coordinate descent method for training l 1 regularized linear classifiers. *Journal of Machine Learning Research*, 18(91):1–35, 2017.

G. Mann, R. McDonald, M. Mohri, N. Silberman, and D. D. Walker. Efficient large-scale distributed training of conditional maximum entropy models. *Neural Information Processing Systems*, 2009.

B. McWilliams, C. Heinze, N. Meinshausen, G. Krummenacher, and H. P. Vanchinathan. LOCO: Distributing ridge regression with random projections. *arXiv.org*, 2014.

X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen, D. Xin, R. Xin, M. J. Franklin, R. Zadeh, M. Zaharia, and A. Talwalkar. MLlib: Machine learning in apache spark. *Journal of Machine Learning Research*, 17(34):1–7, 2016.

J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Puschel. D-ADMM: A Communication-Efficient Distributed Algorithm for Separable Optimization. *IEEE Transactions on Signal Processing*, 61(10):2718–2723, 2013.

I. Necoara. Linear convergence of first order methods under weak nondegeneracy assumptions for convex programming. *arXiv.org*, 2015.

I. Necoara and V. Nedelcu. Distributed dual gradient methods and error bound conditions. *arXiv.org*, 2014.

Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

F. Niu, B. Recht, C. Ré, and S. J. Wright. Hogwild!: A lock-free approach to parallelizing stochastic gradient descent. In *Neural Information Processing Systems*, 2011.

D. Pechyony, L. Shen, and R. Jones. Solving large scale linear SVM with distributed block minimization. In *International Conference on Information and Knowledge Management*, 2011.

Z. Qu, P. Richtárik, and T. Zhang. Quartz: Randomized dual coordinate ascent with arbitrary sampling. In *Neural Information Processing Systems*, 2015.

Z. Qu, P. Richtárik, M. Takáč, and O. Fercoq. SDNA: Stochastic dual Newton ascent for empirical risk minimization. In *International Conference on Machine Learning*, 2016.

P. Richtárik and M. Takáč. Distributed coordinate descent method for learning with big data. *Journal of Machine Learning Research*, 17:1–25, 2016.

R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, NJ, 1997.

S. Shalev-Shwartz and A. Tewari. Stochastic methods for l_1-regularized loss minimization. *Journal of Machine Learning Research*, 12:1865–1892, 2011.

S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14:567–599, 2013a.

S. Shalev-Shwartz and T. Zhang. Accelerated mini-batch stochastic dual coordinate ascent. In *Neural Information Processing Systems*, 2013b.

S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Mathematical Programming*, Series A:1–41, 2014.

O. Shamir and N. Srebro. Distributed Stochastic Optimization and Learning. In *Allerton Conference*, 2014.

O. Shamir, N. Srebro, and T. Zhang. Communication-efficient distributed optimization using an approximate newton-type method. In *International Conference on Machine Learning*, 2014.

V. Smith, S. Forte, M. I. Jordan, and M. Jaggi. L1-Regularized Distributed Optimization: A Communication-Efficient Primal-Dual Framework. *arXiv.org*, 2015.

V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar. Federated multi-task learning. In *Neural Information Processing Systems*, 2017.

M. Takáč, A. Bijral, P. Richtárik, and N. Srebro. Mini-batch primal and dual methods for SVMs. In *International Conference on Machine Learning*, 2013.

R. Tappenden, M. Takáč, and P. Richtárik. On the complexity of parallel coordinate descent. *arXiv.org*, 2015.

I. Trofimov and A. Genkin. Distributed coordinate descent for l1-regularized logistic regression. *arXiv.org*, 2014.

I. Trofimov and A. Genkin. Distributed Coordinate Descent for Generalized Linear Models with Regularization. *arXiv.org*, 2016.

P.-W. Wang and C.-J. Lin. Iteration complexity of feasible descent methods for convex optimization. *Journal of Machine Learning Research*, 15(1):1523–1548, 2014.

S. J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.

T. Yang. Trading computation for communication: Distributed stochastic dual coordinate ascent. In *Neural Information Processing Systems*, 2013.

T. Yang, S. Zhu, R. Jin, and Y. Lin. Analysis of distributed stochastic dual coordinate ascent. *arXiv.org*, Dec. 2013.

I. E.-H. Yen, S.-W. Lin, and S.-D. Lin. A dual augmented block minimization framework for learning with limited memory. In *Neural Information Processing Systems*, 2015.

H.-F. Yu, C.-J. Hsieh, K.-W. Chang, and C.-J. Lin. Large linear classification when data cannot fit in memory. *ACM Transactions on Knowledge Discovery from Data*, 5(4):1–23, 2012.

J. Yu, S. Vishwanathan, S. Günter, and N. N. Schraudolph. A quasi-Newton approach to nonsmooth convex optimization problems in machine learning. *Journal of Machine Learning Research*, 11:1145–1200, 2010.

G.-X. Yuan, K.-W. Chang, C.-J. Hsieh, and C.-J. Lin. A comparison of optimization methods and software for large-scale l1-regularized linear classification. *Journal of Machine Learning Research*, 11:3183–3234, 2010.

G.-X. Yuan, C.-H. Ho, and C.-J. Lin. An improved GLMNET for L1-regularized logistic regression. *Journal of Machine Learning Research*, 13:1999–2030, 2012.

C. Zhang, H. Lee, and K. G. Shin. Efficient distributed linear classification algorithms via the alternating direction method of multipliers. In *International Conference on Artificial Intelligence and Statistics*, 2012.

Y. Zhang and X. Lin. Stochastic primal-dual coordinate method for regularized empirical risk minimization. In *International Conference on Machine Learning*, 2015.

Y. Zhang, J. C. Duchi, and M. J. Wainwright. Communication-efficient algorithms for statistical optimization. *Journal of Machine Learning Research*, 14:3321–3363, 2013.

S. Zheng, J. Wang, F. Xia, W. Xu, and T. Zhang. A general distributed dual coordinate optimization framework for regularized loss minimization. *Journal of Machine Learning Research*, 18:1–52, 2017.

M. A. Zinkevich, M. Weimer, A. J. Smola, and L. Li. Parallelized stochastic gradient descent. *Neural Information Processing Systems*, 2010.

Journal of Machine Learning Research 18 (2018) 1-46 Submitted 1/17; Revised 5/18; Published 6/18

Concentration inequalities for empirical processes of linear time series

Likai Chen lkchen@galton.uchicago.edu
Wei Biao Wu wbwu@galton.uchicago.edu
Department of Statistics
The University of Chicago
Chicago, IL 60637, USA

Editor: Mehryar Mohri

Abstract

The paper considers suprema of empirical processes for linear time series indexed by functional classes. We derive an upper bound for the tail probability of the suprema under conditions on the size of the function class, the sample size, temporal dependence and the moment conditions of the underlying time series. Due to the dependence and heavy-tailness, our tail probability bound is substantially different from those classical exponential bounds obtained under the independence assumption in that it involves an extra polynomial decaying term. We allow both short- and long-range dependent processes. For empirical processes indexed by half intervals, our tail probability inequality is sharp up to a multiplicative constant.

Keywords: martingale decomposition, tail probability, heavy tail, MA(∞)

1. Introduction

Concentration inequalities for suprema of empirical processes play a fundamental role in statistical learning theory. They have been extensively studied in the literature; see for example Vapnik (1998), Ledoux (2001), Massart (2007), Boucheron et al. (2013) among others. To fix the idea, let $(\Omega, \mathcal{F}, \mathbb{P})$ be the probability space on which a sequence of random variables (X_i) is defined, $\mathcal{A}$ be a set of real-valued measurable functions. For a function g, denote $S_n(g) = \sum_{i=1}^n g(X_i)$. We are interested in studying the tail probability

$$T(z) := \mathbb{P}(\Delta_n \geq z), \text{ where } \Delta_n = \sup_{g\in\mathcal{A}} |S_n(g) - \mathbb{E}S_n(g)|. \tag{1}$$

When $\mathcal{A}$ is uncountable, $\mathbb{P}$ is understood as the outer probability (van der Vaart (1998)). In the special case in which $X_1, \ldots, X_n$ are independent and identically distributed (i.i.d.) random variables and $\mathcal{A} = \{\mathbf{1}_{(-\infty,t]}, t \in \mathbb{R}\}$ is the set of indicator functions of half intervals, the Dvoretzky-Kiefer-Wolfowitz-Massart (Dvoretzky et al. (1956); Massart (1990)) theorem asserts that for all $z \geq 0$,

$$T(z) \leq 2e^{-2z^2/n}. \tag{2}$$

Talagrand (1994) obtained a concentration inequality with $\mathcal{A} = \{\mathbf{1}_A, A \in \mathcal{C}\}$, where $\mathcal{C}$ is a VC class; cf Vapnik and Chervonenkis (1971). For empirical processes of independent random variables, a substantial theory has been developed and various powerful techniques have been invented; see Talagrand (1995, 1996), Ledoux (1997), Massart (2000), Boucheron et al. (2003), Klein and Rio (2005) and the monograph Boucheron et al. (2013).

In this paper we shall consider tail probability inequalities for temporally dependent data which are commonly encountered in economics, engineering, finance, geography, physics and other fields. It is considerably more challenging to deal with dependent data. Previous results include uniform laws of large numbers and central limit theorems; see, for example, Adams and Nobel (2012), Levental (1988), Arcones and Yu (1994), Kontorovich and Brockwell (2014) and Yu (1994). Various uniform deviation results have been derived for mixing processes, Markov chains and their variants; see Marton (1996, 1998), Samson (2000), Kontorovich and Ramanan (2008), Adamczak (2008), Kontorovich and Weiss (2014), Kontorovich and Raginsky (2017), Kuznetsov and Mohri (2014, 2015) and Agarwal and Duchi (2013) among others. In many of the aforementioned papers, exponentially decaying tail bounds have been obtained which are similar to those obtained under independence.

Here we shall consider the widely used linear or moving average (MA) process

$$X_i = \sum_{k\geq 0} a_k \epsilon_{i-k}, \tag{3}$$

where innovations $\epsilon_i, i \in \mathbb{Z}$, are i.i.d random variables with mean 0 and $a_k, k \geq 0$, are real numbers such that X_i is a proper random variable. Assume that $\epsilon_i \in \mathcal{L}^q$, $q \geq 1$, namely $\mu_q := \|\epsilon_i\|_q = (\mathbb{E}|\epsilon_i|^q)^{1/q} < \infty$ and coefficients $a_k = O(k^{-\beta})$, $\beta > 1/q$. Namely there exists a constant $C > 0$ such that $|a_k| \leq Ck^{-\beta}$ holds for all large k. Then by Kolmogorov's three-series theorem (Chow and Teicher (1997)), the sum in (3) exists and X_i is well-defined. If $q \geq 2$ and $1/2 < \beta < 1$, then the auto-covariances of the process (X_i) may not be summable, suggesting that the process is long-memory or long-range dependent (LRD). When $\beta > 1$, the process is short-range dependent (SRD). The linear or MA(∞) process (3) is very widely used in practice and it includes many important time series models such as the autoregressive moving average (ARMA) process

$$(1 - \sum_{j=1}^p \theta_j B^j) X_i = X_i - \sum_{j=1}^p \theta_j X_{i-j} = \sum_{k=0}^q \phi_k \epsilon_{i-k}, \tag{4}$$

where θ_j and ϕ_k are real coefficients such that the roots to the equation $1 - \sum_{j=1}^p \theta_j u^j = 0$ are all outside the unit disk and B is the backshift operator, and the fractional autoregressive integrated moving average (FARIMA) (cf. Granger and Joyeux (1980); Hosking (1981))

$$(1 - B)^d (X_i - \sum_{j=1}^p \theta_j X_{i-j}) = \sum_{k=0}^q \phi_k \epsilon_{i-k}, \tag{5}$$

where the fractional integration index $d \in (0, 1/2)$. For (4), the corresponding coefficients $|a_i| = O(\rho^i)$ for some $\rho \in (0, 1)$. While for (5) under suitable causality and invertibility conditions the limit $\lim_{i\to\infty} i^{1-d} a_i = c \neq 0$ exists (Granger and Joyeux (1980); Hosking (1981)). Hence $a_i \sim ci^{-\beta}$ with $\beta = 1 - d$.

The primary goal of the paper is to establish a concentration inequality for $T(z)$ in (1) for the linear process (3). Our theory allows both short- and long-range dependence and

heavy-tailed innovations. Heavy-tailed distributions have been substantially studied in the literature. For instance, Mandelbrot (1963) documented evidence of power-law behavior in asset prices. Rachev and Mittnik (2000) showed long memory and heavy tails in the high frequency asset return data. Recently researchers extended tail probability inequalities to independent heavy-tailed random variables. Lederer and van de Geer (2014) applied the truncation method to develop bounds for an envelope of functions with finite moment assumptions on the envelope. Based on the robust M-estimator introduced in Catoni (2012), Brownlees et al. (2015) proposed a risk minimization procedure using the generic chaining method. The case with both dependence and heavy tails is more challenging. Jiang (2009) introduced a triplex inequality to handle unbounded and dependent situations. Mohri and Rostamizadeh (2010) considered φ-mixing and β-mixing processes. It is generally not easy to verify that a process is strong mixing and computation of mixing coefficients can be very difficult. Some simple and widely used AR processes are not strong mixing (Andrews (1984)).

In the present paper, we propose a martingale approximation based method. An intuitive illustration is given in Section 6.2. Our tail probability bound is a combination of an exponential term and a polynomial term (cf. Theorems 4 and 8), whose order depends on both β and q, which quantify the dependence and the moment condition, respectively. Larger β or q implies thinner tails. Our tail inequality allows both short- and long- range dependent processes and can also be adapted to discontinuous function classes including empirical distribution functions, which is fundamental and is of independent interest. Our theorem implies that, if the innovation ϵ_0 has tail

$$\mathbb{P}(|\epsilon_0| \geq x) = O(\log^{-r_0}(x)x^{-q}), \quad \text{as } x \to \infty, \tag{6}$$

where $r_0 > 1$ and $q > 1$ signifies heaviness of the tail, namely there exists a constant $C > 0$ such that $\mathbb{P}(|\epsilon_0| \geq x) \leq C\log^{-r_0}(x)x^{-q}$ holds for all large x, and the coefficients

$$a_k = O(k^{-\beta}),\ \beta > 1 \text{ and } q\beta \geq 2, \tag{7}$$

where β quantifies the dependence with larger β implying weaker dependence, then for $z \geq \sqrt{n}\log(n)$, the tail probability

$$\mathbb{P}\Big(\sup_{t\in\mathbb{R}}\Big|\sum_{i=1}^{n}[\mathbf{1}_{X_i\leq t} - F(t)]\Big| > z\Big) \lesssim \frac{n}{z^{q\beta}\log^{r_0}(z)}, \tag{8}$$

where the constant in $\lesssim$ is independent of n and z, $F(t) = \mathbb{P}(X_i \leq t)$ is the cumulative distribution function (c.d.f.) for X_i. Note that the bound (8) involves both the dependence parameter β and the tail heaviness parameter q. In comparison with the sub-Gaussian bound $e^{-2z^2/n}$ in (2), the polynomial bound (8) is much larger. On the other hand, however, it turns out that the polynomial bound (8) is *sharp* and it is essentially not improvable. For example, let $F_\epsilon(t) = \mathbb{P}(\epsilon_0 \leq t)$ be the c.d.f. of ϵ_0, and assume that the innovation ϵ_i has a symmetric regularly varying tail: for some $r_0 > 1$,

$$F_\epsilon(-x) = 1 - F_\epsilon(x) \sim \log^{-r_0}(x)x^{-q} \text{ as } x \to \infty, \tag{9}$$

namely $\lim_{x\to\infty}(1 - F_\epsilon(x))\log^{r_0}(x)x^q = 1$, and that the coefficients

$$a_k = (k \vee 1)^{-\beta},\ \beta > 1. \tag{10}$$

Then by Theorem 14, when $n/\log^{\alpha_0}(n) \geq z \geq \sqrt{n}\log(n)$ for some $\alpha_0 > 0$, we can have the precise order of the tail probability

$$\mathbb{P}\Big(\sum_{i=1}^{n}[\mathbf{1}_{X_i\leq t} - F(t)] > z\Big) = C_1\frac{n}{z^{q\beta}\log^{r_0}(z)}(1 + o(1)), \quad n \to \infty,$$

and

$$\mathbb{P}\Big(\sum_{i=1}^{n}[\mathbf{1}_{X_i\leq t} - F(t)] < -z\Big) = C_2\frac{n}{z^{q\beta}\log^{r_0}(z)}(1 + o(1)), \quad n \to \infty,$$

where the constants C_1, C_2 are independent of z and n. Hence the bound in (8) is sharp up to a multiplicative constant.

On the technical side, to establish inequality (8) and more generally, a tail probability inequality for empirical processes indexed by function classes, we need to develop new approaches so that the two main challenges posed by dependence and heavy tails can be dealt with. Techniques developed for empirical processes with independent random variables are not directly applicable. Here, we apply the martingale approximation method, together with the Fuk-Nagaev inequalities for high-dimensional vectors recently obtained by Chernozhukov et al. (2017), projection techniques and martingale inequalities, so that an optimal bound can be derived. Intuitions are given in the proof of Theorem 4 in Section 6.2. As a result, we can allow short- and long-range dependent, and light- and heavy-tailed linear processes.

The remainder of the paper is organized as follows. Section 2 states the theoretical results: Subsections 2.1 and 2.2 show the tail probabilities for short- and long- range dependence situations respectively with heavy tailness, Subsection 2.3 presents results for light tailed innovations. In Section 3, we apply the concentration inequality to empirical distribution functions as an important special case. We also derive an exact order of decay speed under certain settings, which demonstrates the sharpness of our upper bound. Sections 4 and 5 present applications in kernel density estimation and empirical risk minimization, respectively. Detailed proofs are provided in Section 6.

We now introduce some notation. For a random variable X and $q > 0$, we write $X \in \mathcal{L}^q$ if $\|X\|_q := \mathbb{E}(|X|^q)^{1/q} < \infty$. Write $\|\cdot\| = \|\cdot\|_2$. For a function g, define $|g|_\infty := \sup_x |g(x)|$. Let $x \vee y = \max\{x, y\}$ and $x \wedge y = \min\{x, y\}$. For two sequences of positive numbers (a_n) and (b_n), write $a_n \lesssim b_n$ (resp. $a_n \ll b_n$, $a_n \asymp b_n$, $a_n \sim b_n$) if there exists a positive constant C such that $a_n/b_n \leq C$ for all large n (resp. $\lim_{n\to\infty} a_n/b_n = 0$, $1/C \leq a_n/b_n \leq C$ for all large n, $\lim_{n\to\infty} a_n/b_n = 1$). Denote by F_ϵ (resp. F) the c.d.f. of the innovation ϵ_i (resp. X_i) and by $f_\epsilon = F_\epsilon'$ (resp. $f = F'$) the probability density function (p.d.f.) of ϵ_i (resp. X_i).

2. Main results

Recall (3) for the MA(∞) process (X_i), where $\epsilon_j \in \mathcal{L}^q,\ j \in \mathbb{Z}$, are i.i.d. with c.d.f. F_ϵ and p.d.f. f_ϵ. Assume $a_0 \neq 0$ and without loss of generality, let $a_0 = 1$.

For a function class $\mathcal{A}$ of bounded functions, define the covering number

$$\mathcal{N}_{\mathcal{A}}(\delta) := \min\Big\{m :\ \text{there exist } g_1, \ldots, g_m \in \mathcal{A} \text{ such that } \sup_{g\in\mathcal{A}}\min_{1\leq j\leq m}|g - g_j|_\infty \leq \delta\Big\}. \tag{11}$$

Let $H_{\mathcal{A}}(\delta) := \log(\mathcal{N}_{\mathcal{A}}(\delta))$ be the metric entropy.

Before stating the main theorems, we shall introduce some assumptions.

(A) (Smoothness) For any $g \in \mathcal{A}$, g', g'' exist and $|g|, |g'|, |g''|$ are uniformly bounded, without loss of generality set the bound to be 1.

(A') Functions in $\mathcal{A}$ are uniformly bounded in $|\cdot|_\infty$ with $\sup_{g\in\mathcal{A}}|g|_\infty \le 1$. Assume that $f'_\epsilon, f''_\epsilon$ exist and the integrals $\int_{-\infty}^{\infty}|f'_\epsilon(x)|\mathrm{d}x, \int_{-\infty}^{\infty}|f''_\epsilon(x)|\mathrm{d}x$ are bounded by 1.

(B) (Algebraically Decaying Coefficients) For some $\gamma, \beta > 0$, $|a_k| \le \gamma k^{-\beta}$ holds for all $k \ge 1$.

(B') (Exponentially Decaying Coefficients) For some $\gamma > 0, 0 < \rho < 1$, $|a_k| \le \gamma\rho^k$ holds for all $k \ge 1$.

(D) (Exponential Class) For some constants $N, C, \theta > 0$, the covering number $\mathcal{N}_{\mathcal{A}}(\delta) \le N\exp(C\delta^{-\theta})$ holds for all $0 < \delta \le 1$.

(D') (Algebraical Class) For some constants $N, \theta > 0$, the covering number $\mathcal{N}_{\mathcal{A}}(\delta) \le N\delta^{-\theta}$ holds for all $0 < \delta \le 1$.

Remark 1 *Assumption (A) requires that functions in $\mathcal{A}$ have up to second order derivatives. This is relaxed in (A'), where an extra differentiability condition of f_ϵ is imposed. It holds for many commonly used distributions such as Gaussian and t distributions.*

Remark 2 *Assumption (B) specifies the decay rate of the MA(∞) coefficients to be at most polynomial. The parameter β controls the dependence strength, with larger β implying weaker dependence. By Theorem 4(v) in Chen and Wu (2016), the AR(∞) process*

$$X_t = \sum_{i\ge1} b_i X_{t-i} + \epsilon_t \tag{12}$$

with coefficients $|b_i| = O(i^{-\beta})$, $\beta > 1$, and $\sum_{i\ge1}|b_i| < 1$, can also be rewritten as an MA(∞) process with coefficients (a_i) decaying at the same polynomial rate. Assumption (B') allows ARMA processes (4).

Remark 3 *Assumptions (D) and (D') quantify the magnitudes of the class $\mathcal{A}$. They are satisfied for many function classes; see van der Vaart and Wellner (1996) and Kosorok (2008). For example, the former holds for Hölder or Sobolev classes, while the latter holds for VC classes.*

In the MA(∞) model described in (3), the parameter β controls the dependence: if $\beta > 1$, the covariances $\mathrm{Cov}(X_i, X_0)$, $i \ge 1$, are absolutely summable and the process (X_i) is short-range dependent; if $1/2 < \beta < 1$, then the covariances may not be absolutely summable and the process exhibits long-range dependence. The two cases are dealt with in Subsections 2.1 and 2.2, respectively. Subsection 2.3 deals linear processes with sub-exponential innovations.

2.1 Short-range dependent linear processes

We first consider the short-range dependence case with $\beta > 1$ in model (3). Recall (1) for Δ_n. Assume throughout the paper that $n \ge 2$. Let $q' := q \wedge 2$ and

$$c(n,q) = \begin{cases} n^{1/q'}, & \text{if } q > 2 \text{ or } 1 < q < 2, \\ n^{1/2}\log^{1/2}(n), & \text{if } q = 2. \end{cases} \tag{13}$$

Theorems 4 and 7 concern algebraically and exponentially decaying coefficients, respectively. In the statements of our theorems we use the notation $C_{\alpha,\beta,\gamma,\ldots}$ to denote constants that only depend on subscripts $\alpha, \beta, \gamma, \ldots$. Since $|g|_\infty \le 1$, we have $T(z) = 0$ if $z > n$ and thus assume throughout the paper that $z \le n$.

Theorem 4 (Algebraically decaying coefficients) *Assume (A) and (B), $\beta > 1, q > 1$ and $q\beta \ge 2$. Then there exist positive constants $C_q, C_{\beta,q,\gamma}$ and $C_{\beta,\gamma}$ such that for all $z > 0$,*

$$\begin{aligned}&\mathbb{P}\Big(\Delta_n \ge C_q a_* \mu_q c(n,q) + z\Big) \\ \le& C_{\beta,q,\gamma}\mu_q^q \frac{n}{z^{q\beta}} + 3\exp\Big(-\frac{z^2}{C_{\beta,\gamma}\mu_{q'}^{q'} n} + H_{\mathcal{A}}(z/(4n))\Big) + 2\exp\Big(-\frac{z^v}{8\mu_q^{v'}} + H_{\mathcal{A}}(z/(4n))\Big),\end{aligned} \tag{14}$$

where $\mu_q = (\mathbb{E}|\epsilon_i|^q)^{1/q}$, $a_* = \sum_{i=0}^{\infty}|a_i|$, and

$$v = v_{q,\beta} = (q'\beta - 1)(3q'\beta - 1)^{-1}, \quad v' = 2q'(3q'\beta - 1)^{-1}. \tag{15}$$

The specific values of the constants $C_q, C_{\beta,q,\gamma}$ and $C_{\beta,\gamma}$ will be given in Remark 25 (Section 6.2). The bound (14) is a combination of exponential and polynomial terms. For z relatively small, the exponential term contributes more, while for z relatively large, the polynomial term $n/z^{q\beta}$ dominates. Note that $0 < v < 1/3$. Comparing the last two terms in (14), if $n^{1/(2-v)} \lesssim z$, then the last term dominates, and vise versa.

In Theorem 4, under Assumption (A), the class $\mathcal{A}$ consists of differentiable functions. To incorporate non-continuous functions, we can impose Assumption (A'), which requires differentiability of f_ϵ; cf Proposition 5. Corollary 6 follows from Theorem 4 and Proposition 5.

Proposition 5 *Assume (A') and (B), $\beta > 1, q > 1$ and $q\beta \ge 2$. Then there exist positive constants $C_q, C_{\beta,q,\gamma}$ and $C_{\beta,\gamma}$ such that for all $z > 0$,*

$$\begin{aligned}&\mathbb{P}\Big(\Delta_n \ge C_q a_* \mu_q c(n,q) + z\Big) \\ \le& C_{\beta,q,\gamma}\mu_q^q \frac{n}{z^{q\beta}} + 5\exp\Big(-\frac{z^2}{C_{\beta,\gamma}(\mu_{q'}^{q'} \vee 1)n} + H_{\mathcal{A}}(z/(4n))\Big) + 2\exp\Big(-\frac{z^v}{8\mu_q^{v'}} + H_{\mathcal{A}}(z/(4n))\Big),\end{aligned}$$

where $c(n,q)$ is defined in (13) *and v, v' are defined in* (15).

Corollary 6 *Assume (A) (or (A')) and (B). Let $\beta > 1, q > 1$ and $q\beta \ge 2$. Define $c(n,q)$ and v as in* (13) *and* (15)*, respectively. If either (i) Assumption (D) holds, $\alpha = \max\{\theta/(\theta+$*

$2), (\theta - v)/(\theta + v)\}/2$, *and* $z \geq cn^{1/2+\alpha}$ *for a sufficiently large* c*; or (ii) for some* $N, \theta > 0$, *Assumption* (D') *holds and* $z \geq cn^{1/2}\log^{1/2}(n)$ *for a sufficiently large* c*, then we have*

$$\mathbb{P}\Big(\Delta_n \geq C_q a_* \mu_q c(n,q) + z\Big) \leq C\mu_q^q \frac{n}{z^{q\beta}}, \tag{16}$$

where the constant C *only depends on* $\beta, q, \gamma, \theta, c$ *and* N.

Observe that in (16), when $q > 2$, the term $C_q a_* \mu_q c(n,q) + z$ can actually be replaced by z by choosing a larger constant C at the right hand side of (16), since $z \geq cn^{1/2+\alpha}$ or $z \geq cn^{1/2}\log^{1/2}(n)$ for a sufficiently large c, under (i) or (ii), respectively. The tail bound depends on both the dependence parameter β and the moment q.

If the coefficients (a_k) decay exponentially (cf Assumption (B')), then the process is very weakly dependent. It turns out that the polynomial term can be removed and an exponential upper bound can be derived; cf Theorem 7. Note that the bound in Theorem 7 explicitly involves ρ, with larger ρ indicating stronger dependence. We emphasize that the constants C_q, $C_{q,\gamma}$ and $C'_{q,\gamma}$ in (17) does not depend on ρ and they are given in Remark 26 (Section 6.3). Concentration inequality of this form is useful in situations in which one needs to deal with the dependence on ρ.

Theorem 7 (Exponentially decaying coefficients) *Assume that the coefficients* (a_k) *of* (X_i) *defined in* (3) *satisfy* (B') *and* $\mu_q = \|\epsilon_i\|_q < \infty$, $q > 1$. *Let* $\mathcal{A} = \{g : \mathbb{R} \mapsto \mathbb{R}, |g|_\infty \leq 1, |g'|_\infty \leq 1\}$. *Then*

$$\mathbb{P}(\Delta_n \geq C_q \mu_q c^*(n,\rho,q) + z) \leq C_{q,\gamma}\frac{\exp\{-q(1-\rho)n\}\mu_q^q}{z^q(1-\rho)^{q+q/q'}} + \exp\Big\{-C'_{q,\gamma}\frac{z^2(1-\rho)^2}{n(\mu_q^q \vee 1)}\Big\}, \tag{17}$$

where $q' = \min\{q, 2\}$,

$$c^*(n,\rho,q) = \begin{cases} n^{1/q'}(1-\rho)^{-1-1/q'}, & \text{if } q \neq 2, \\ \sqrt{n}(1-\rho)^{-3/2}\log(n(1-\rho)^{-1}), & \text{if } q = 2. \end{cases}$$

2.2 Long-range dependent linear processes

The phenomenon of long-range dependence has been observed in various fields including economics, finance, hydrology, geophysics etc; see, for example, Beran (1994), Baillie (1996). This subsection considers $1/2 < \beta < 1$, the long-range dependence case in model (3). Weak convergence for empirical processes for long-memory time series was studied by Ho and Hsing (1996) and Wu (2003) among others. Under suitable conditions on the class $\mathcal{A}$, by Corollary 1 in Wu (2003), one has $\mathbb{E}(\Delta_n^2) \lesssim n^{3-2\beta}$, which by Markov's inequality implies

$$\mathbb{P}(\Delta_n \geq z) \leq \frac{\mathbb{E}(\Delta_n^2)}{z^2} \lesssim \frac{n^{3-2\beta}}{z^2}.$$

Here we shall derive a much sharper and more general bound; cf Theorem 8, which allows strong dependence with non-summable algebraically decaying coefficients since $\beta < 1$. In comparison the coefficients (a_k) in Theorem 4 are summable, since $\beta > 1$, and the process is weakly dependent. Proposition 9 is an analogous version of Proposition 5 which allows discontinuous functions. Corollary 10 provides an explicit upper bound under certain conditions on the bracketing numbers and it follows from Theorem 8 and Proposition 9.

Theorem 8 *Assume* (A) *and* (B), $q > 2$, $1/2 < \beta < 1$. *Then there exist positive constants* $C'_{\beta,q,\gamma}$, $C_{\beta,q,\gamma}$ *and* $C_{\beta,\gamma}$ *such that for all* $z > 0$,

$$\begin{aligned}&\mathbb{P}\Big(\Delta_n \geq C'_{\beta,q,\gamma}\mu_q n^{3/2-\beta} + z\Big) \\ \leq& C_{\beta,q,\gamma}(\mu_q^{2q} \vee \mu_q^q)\frac{n^{1+(1-\beta)q}}{z^q}\Big(1 + \frac{[H_{\mathcal{A}}(z/4n) + \log(n)]^q}{\tilde{c}^q(n,\beta)}\Big) + 3\exp\Big(-\frac{z^2}{C_{\beta,\gamma}n^{3-2\beta}\mu_2^2} + H_{\mathcal{A}}(z/(4n))\Big),\end{aligned} \tag{18}$$

where

$$\tilde{c}(n,\beta) = \begin{cases} n^{1/4-|3/4-\beta|} & \text{if } \beta \neq 3/4, \\ n^{1/4}/\log(n) & \text{if } \beta = 3/4. \end{cases} \tag{19}$$

Values of constants $C'_{\beta,q,\gamma}$, $C_{\beta,q,\gamma}$ and $C_{\beta,\gamma}$ in Theorem 8 are given in Remark 29 (Section 6.4). In comparison with the bound $nz^{-q\beta}$ in the short-range dependence case Theorem 4, the bound $n^{1+(1-\beta)q}z^{-q}$ in (18) of Theorem 8 is larger since $nz^{-q\beta} \leq n^{1+(1-\beta)q}z^{-q}$ and $n \geq z$.

Proposition 9 *Assume* (A') *and* (B), $q > 2$, $1/2 < \beta < 1$. *Recall* (19) *for* $\tilde{c}(n,q)$. *Then there exist positive constants* $C'_{\beta,q,\gamma}$, $C_{\beta,q,\gamma}$ *and* $C_{\beta,\gamma}$ *such that for all* $z > 0$,

$$\begin{aligned}\mathbb{P}\Big(\Delta_n \geq C'_{\beta,q,\gamma}\mu_q n^{3/2-\beta} + z\Big) \;\leq\; & C_{\beta,q,\gamma}(\mu_q^{2q} \vee \mu_q^q)\frac{n^{1+(1-\beta)q}}{z^q}\Big(1 + \frac{[H_{\mathcal{A}}(z/4n) + \log(n)]^q}{\tilde{c}^q(n,\beta)}\Big) \\ & + 5\exp\Big(-\frac{z^2}{C_{\beta,\gamma}n^{3-2\beta}(\mu_2^2 \vee 1)} + H_{\mathcal{A}}(z/(4n))\Big).\end{aligned}$$

Corollary 10 *Assume* (A) *(or* (A')*) and* (B). *Let* $q > 2$, $1/2 < \beta < 1$. *If either (i) for some* $N, \theta > 0$, *Assumption* (D) *holds and* $z \geq cn^{3/2-\beta+\alpha}$ *for* $\alpha = (\beta - 1/2)\theta/(\theta + 2)$ *and a sufficiently large* c *or (ii) for some* $N, \theta > 0$, *Assumption* (D') *holds and* $z \geq cn^{3/2-\beta}\log^{1/2}(n)$ *for a sufficiently large* c. *Then there exists a constant* $C'_{q,\beta,\gamma}$ *such that*

$$\mathbb{P}\Big(\Delta_n \geq C'_{q,\beta,\gamma}\mu_q n^{3/2-\beta} + z\Big) \lesssim \frac{n^{1+(1-\beta)q}}{z^q}(\mu_q^{2q} \vee \mu_q^q)\Big(1 + \frac{t_n^q}{\tilde{c}^q(n,\beta)}\Big), \tag{20}$$

where $t_n = n^{\theta(\beta-1/2-\alpha)}$ *and* $\log(n)$ *for* (i) *and* (ii) *respectively, and the constant in* $\lesssim$ *only depends on* $q, \beta, \gamma, \theta, c$ *and* N.

2.3 Linear processes with sub-exponential innovations

In this subsection, we shall consider concentration inequalities for linear processes with innovations having very light tails. In particular, we assume that innovations ϵ_i have sub-exponential tails. In this case for both short- and long-range dependent processes we have exponentially decaying tail probabilities, with different norming sequences.

Theorem 11 *Let* $\mathcal{G} = \{g : |g|_\infty \leq 1, |g'|_\infty \leq 1\}$. *Assume* (B) *and there exist constants* $c_0 > 0, f_* > 0$ *such that* $|f_\epsilon|_\infty \leq f_*$, *where* f_ϵ *is the p.d.f of* ϵ_0, *and* $\mu_e := \mathbb{E}(e^{c_0|\epsilon_0|}) < \infty$. *Then there exist constants* C_1, C_2, C_3 *and* C_4 *such that*

(a) for SRD case ($\beta > 1$), we have for all $z > 0$,

$$\mathbb{P}(\Delta_n \geq C_1\sqrt{n} + z) \leq 2e^{-C_2 z^2/n},$$

(b) for LRD case ($1/2 < \beta < 1$), we have for all $z > 0$,

$$\mathbb{P}(\Delta_n \geq C_3 n^{3/2-\beta} + z) \leq 2e^{-C_4 z^2/n^{3-2\beta}}.$$

Here the constants C_1 and C_3 only depend on $f_*, \beta, \gamma, c_0, \mu_e$, constants C_2, C_4 only depend on $\beta, \gamma, c_0, \mu, \mu_e$ and their values are given in Remark 30 (Section 6.5). Note that Theorem 11(a) implies $\mathbb{P}(\Delta \geq z) \leq 2e^{-C_5 z^2/n}$ for all $z > 0$, where constant C_5 depends on $f_*, \beta, \gamma, c_0, \mu$ and μ_e. A similar claim can be made for case (b).

In comparison with the results in Theorem 4 and Theorem 8, due to the light tails of the innovations, we do not encounter the polynomial terms $n/z^{q\beta}$ or $n^{3-2\beta}/z^{q\beta}$ here.

3. Empirical distribution functions

In this section we shall consider the important class of indicators indexed by half intervals. Let

$$S_n(t) = n[\hat{F}_n(t) - F(t)] = \sum_{i=1}^{n} [\mathbf{1}_{X_i \leq t} - F(t)]. \tag{21}$$

In Massart (1990)'s result (2), X_i are i.i.d. In Theorem 12, we present a concentration inequality for dependent and possibly heavy-tailed random variables, which has a very different upper bound that involves a polynomial decaying tail. Theorem 14 provides a lower bound for the deviation with regularly varying innovations. That lower bound assures the sharpness of Theorem 12: the polynomial decaying tail is unavoidable. Recall F_ϵ is the c.d.f. of ϵ_0 and f_ϵ its p.d.f. The values of constants in Theorem 12 are given in Remark 31 (Section 6.6). Following assumption states the boundedness of $|f_\epsilon|_\infty$ and $|f'_\epsilon|_\infty$.

(A_1) Let $f_* := \max(1, |f_\epsilon|_\infty, |f'_\epsilon|_\infty)$. Assume $f_* < \infty$.

Theorem 12 *Assume (A_1) and (B). Recall $c(n,q)$ and v, v' in (13) and (15) respectively.*

(i). Let $\beta > 1, q > 1$ (SRD case) and $q\beta \geq 2$. Then there exist constants C_0, C_1, C_2, C_3 such that

$$\begin{aligned}
&\mathbb{P}\left(\sup_{t\in\mathbb{R}} |S_n(t)|/f_* > C_0 a_* \mu_q c(n,q) + z\right) \\
\leq& C_1 \mu_q^q \frac{n}{z^{q\beta}} + 4\exp\Big\{ - C_2 \frac{z^2}{n(\mu_{q'}^{q'} \vee 1)} + C_3 \log(n\mu_q) \Big\} \\
&+ 2\exp\Big\{ - \frac{z^v}{2^{3+2v}\mu_q^{v'}} + C_3 \log(n\mu_q) \Big\},
\end{aligned}$$

In particular, if $z \geq cn^{1/2}\log^{1/2}(n)$, where c is a sufficiently large constant, then the above upper bound becomes $2C_1\mu_q^q n/z^{q\beta}$.

(ii). If $1/2 < \beta < 1$ (LRD case) and $q > 2$, then there exist constants C_0', C_1', C_2', C_3' such that

$$\begin{aligned}
&\mathbb{P}\left(\sup_{t\in\mathbb{R}} |S_n(t)|/f_* > C_0' \mu_q n^{3/2-\beta} + z\right) \\
\leq& C_1'(\mu_q^{2q} \vee \mu_q^q) \frac{n^{1+(1-\beta)q}}{z^q} + 4\exp\Big\{ - C_2' \frac{z^2}{n^{3-2\beta}(\mu_2^2 \vee 1)} + C_3' \log(n\mu_q) \Big\},
\end{aligned}$$

If $z \geq cn^{3/2-\beta}\log^{1/2}(n)$ for a sufficiently large c, then the above upper bound becomes $2C_1'(\mu_q^{2q} \vee \mu_q^q) n^{1+(1-\beta)q}/z^q$.

In (i) the constant C_0 only depends on q, C_1, C_3 only depend on β, q, γ and C_2 only depends on β, γ; In (ii) the constants C_0', C_1', C_3' only depend on β, q, γ and C_2' only depends on β, γ, their specific values can be found in Remark 31 (Section 6.6).

Under certain forms of tail probability of the innovations, we can have a more refined result.

Proposition 13 *Assume (A_1), (B), $\beta > 1$ and $q > 2$. Assume for any $x > 1$, $\mathbb{P}(|\epsilon_0| > x) \leq L\log^{-r_0}(x)x^{-q}$, with some constants $r_0 > 1, L > 0$. If $z \geq c\sqrt{n}\log^\alpha(n)$, $\alpha > 1/2$, then*

$$\mathbb{P}\left(\sup_{t\in\mathbb{R}} |S_n(t)|/f_* > z\right) \lesssim \frac{\mu_q^q n}{z^{q\beta}\log^{r_0}(z)},$$

where the constant in $\lesssim$ only depends on $\beta, q, \gamma, r_0, L, c$ and α.

To appreciate the sharpness of the upper bound in Proposition 13, we derive an exact decay rate when $a_k = (k \vee 1)^{-\beta}$ and ϵ_0 is symmetric with a regularly varying tail.

Theorem 14 *Assume (A_1), (B) with coefficients $a_k = (k \vee 1)^{-\beta}$, $k \geq 0$, and that ϵ_0 is symmetric with tail distribution*

$$\mathbb{P}(\epsilon_0 \geq x) \sim \log^{-r_0}(x)x^{-q}, \text{ as } x \to \infty, \tag{22}$$

where $r_0 > 1$ is a constant. Let $\beta > 1$, $q > 2$ and $\alpha > 1/2$. Then there exists a constant $\Gamma > 0$ such that for all z with $\sqrt{n}\log^\alpha(n) \leq z \leq n/\log^\Gamma(n)$,

$$\mathbb{P}\left(S_n(t) > z\right) = (1 + o(1))C_1 \frac{n}{\log^{r_0}(z)z^{q\beta}}, \tag{23}$$

and

$$\mathbb{P}\left(S_n(t) < -z\right) = (1 + o(1))C_2 \frac{n}{\log^{r_0}(z)z^{q\beta}}, \tag{24}$$

where the constants C_1, C_2 only depend on q, β, r_0, t and F.

Values of C_1 and C_2 are given in Lemma 34, and the constant Γ can be found in Remark 35 (Section 6.7). The asymptotic expressions (23) and (24) in Theorem 14 precisely depict the magnitude of the tail probability $\mathbb{P}\left(S_n(t) > z\right)$ and $\mathbb{P}\left(S_n(t) < -z\right)$. It asserts that the upper bound order in Proposition 13 is optimal within the range $\sqrt{n}\log^\alpha(n) \leq z \leq n/\log^\Gamma(n)$. Thus the polynomial $n/z^{q\beta}$ in Theorems 4 and 12 is sharp up to a multiplicative logarithmic term.

4. Kernel density estimation

Let (X_i) be a stationary sequence satisfying (3) with the marginal p.d.f. f. Given the observations $X_1, \ldots, X_n$, the kernel density estimator of f is

$$\hat{f}_n(x) = \frac{1}{n}\sum_{j=1}^{n} K_b(x - X_j), \quad K_b(\cdot) = b^{-1}K(\cdot/b),$$

where the bandwidth $b = b_n$ satisfies the natural condition $b_n \to 0$ and $nb_n \to \infty$. Wu and Mielniczuk (2002) established an asymptotic distribution theory for $A_n(\hat{f}_n(x) - \mathbb{E}\hat{f}_n(x))$ for both short- and long-range dependent processes, where A_n is a proper norming sequence. In this section we shall derive a bound for the tail probability

$$\mathbb{P}\Big(\sup_{x\in\mathbb{R}} n|\hat{f}_n(x) - \mathbb{E}\hat{f}_n(x)| \geq z\Big).$$

Such a bound is useful for constructing non-asymptotic confidence bounds. Giné and Guillou (2002) and Giné and Nickl (2010) considered the latter problem for i.i.d. data. Einmahl and Mason (2005) derived uniform in bandwidth consistency result for kernel-type function estimators. Hang et al. (2016) studied consistency properties for observations generated by certain dynamical systems under mixing conditions. Rinaldo et al. (2012), Chen et al. (2016) and Arias-Castro et al. (2016) applied such bounds in clustering problem. Liu et al. (2011) and Lafferty et al. (2012) used it in forest density estimation. Here, we shall provide a polynomial decay bound for linear time series.

Corollary 15 *Assume (B), the kernel K is symmetric with support $[-1,1]$, $\max(|K|_\infty, |K'|_\infty) \leq K_*$ and $\max(1, |f_\epsilon|_\infty, |f'_\epsilon|_\infty, |f''_\epsilon|_\infty) \leq f_*$ for some constants $K_*, f_* > 0$.*

(a) In the SRD case with $\beta > 1, q > 1, q\beta \geq 2$, if $z \geq c(n/b_n)^{1/2}\log^{1/2}(n)$ for a sufficiently large c, then

$$\mathbb{P}\Big(\sup_{x\in\mathbb{R}} n|\hat{f}_n(x) - \mathbb{E}\hat{f}_n(x)| \geq \max(f_*, K_*)z\Big) \lesssim \mu_q^q n/z^{q\beta}, \tag{25}$$

where the constant in $\lesssim$ only depends on β, q, γ and c.

(b) In the LRD case with $1/2 < \beta < 1, q > 2$, if $z \geq c\max\{n^{3/2-\beta}, (n/b_n)^{1/2}\}\log^{1/2}(n)$ holds for a sufficiently large c, then

$$\mathbb{P}\Big(\sup_{x\in\mathbb{R}} n|\hat{f}_n(x) - \mathbb{E}\hat{f}_n(x)| \geq \max(f_*, K_*)z\Big) \lesssim (\mu_q^{2q} \vee \mu_q^q)\frac{n^{1+(1-\beta)q}}{z^q}, \tag{26}$$

where the constant in $\lesssim$ only depends on β, q, γ and c.

5. Empirical risk minimization

Empirical risk minimization is of fundamental importance in the statistical learning theory and it is studied in various contexts including classification, regression and clustering among others. To fix the notation, let (X, Y) be a random vector taking values in the space $\mathcal{X} \times \mathcal{Y}$

and $\mathcal{H}$ be a class of measurable functions $h : \mathcal{X} \to \mathcal{Y}$. For a function $h \in \mathcal{H}$, define the risk $R(h) = \mathbb{E}[L(X, Y, h(X))]$, where L is a loss function. Let $h^* = \mathrm{argmin}_{h\in\mathcal{H}} R(h)$. Based on the observations $(X_1, Y_1), \ldots, (X_n, Y_n)$ which are identically distributed as (X, Y), consider the empirical risk minimizer

$$\hat{h} = \mathrm{argmin}_{h\in\mathcal{H}} R_n(h), \text{ where } R_n(h) = n^{-1}\sum_{i=1}^{n} L(X_i, Y_i, h(X_i)) \tag{27}$$

is the empirical risk. Since $R_n(h^*) \geq R_n(\hat{h})$, it follows (cf. Devroye et al. (1996)) that

$$0 \leq R(\hat{h}) - R(h^*) \leq 2\Psi_n, \text{ where } \Psi_n = \sup_{h\in\mathcal{H}} |R_n(h) - R(h)|. \tag{28}$$

A primary goal in statistical learning theory is to bound the uniform deviation Ψ_n. The latter problem has been widely studied when (X_i, Y_i) are assumed to be i.i.d.; see, for example, Caponnetto and Rakhlin (2006), Vapnik (1998, 2000) and Gottlieb et al. (2017). In recent years various dependent processes have been considered; see Modha and Masry (1996), Guo and Shi (2011), Zou and Li (2007), Zou et al. (2009), Alquier and Wintenberger (2012), Mohri and Rostamizadeh (2010), Steinwart and Christmann (2009), Hang and Steinwart (2014, 2016), Shalizi and Kontorovich (2013) among others.

Here we shall provide an upper bound for Ψ_n with (X_i) being the MA(∞) process (3) and the regression model

$$Y_i = H_0(X_i, \eta_i),$$

where $\eta_i, i \in \mathbb{Z}$, are i.i.d. random errors independent of (ϵ_i) and H_0 is an unknown measurable function. Denote $\mathcal{A} = \{g(x, y) = L(x, y, h(x)), h \in \mathcal{H}\}$ and

$$\mathcal{N}_{\mathcal{A}}(\delta) = \min\{m : \text{there exist } g_1, \ldots, g_m \in \mathcal{A}, \text{such that } \sup_{g\in\mathcal{A}} \min_{1\leq j\leq m} |g - g_j|_\infty \leq \delta\},$$

where $|g|_\infty = \sup_{x,y} |g(x, y)|$. Assume that the loss function L take values in $[0, 1]$. Here for the sake of presentational clarity we do not seek the fullest generality but as an illustration on how to apply our main results. Recall that f_ϵ is the density function of ϵ_i.

Corollary 16 *Assume (B), the density $f_\epsilon \in \mathcal{C}^2(\mathbb{R})$ with $f_* := \max(\int_{-\infty}^{\infty} |f'_\epsilon(x)|\mathrm{d}x, \int_{-\infty}^{\infty} |f''_\epsilon(x)|\mathrm{d}x, 1)$. Under conditions (i) or (ii) in Corollary 6 on the function class $\mathcal{H}$, $q, \beta > 1$ and $q\beta \geq 2$ (resp. conditions (i) or (ii) in Corollary 10 on $\mathcal{H}$, $q > 2$ and $1/2 < \beta < 1$), we have* (16) *(resp.* (20)*) holds with Δ_n therein replaced by $n\Psi_n/f_*$.*

Remark 17 In literature, many concentration inequalities for time series are derived under various mixing conditions (see, for example, Mohri and Rostamizadeh (2010)). Since mixing and our model (3) cover different ranges of processes, our results are not directly comparable with theirs. Here we consider an example in which our result and Corollary 21 in Mohri and Rostamizadeh (2010) can be compared. Let $X_i = \sum_{k\geq 0} a_k\epsilon_{i-k}$, where ϵ_t are i.i.d. with finite qth moment, $q > 2$ and $a_0 = 1$, $a_k \asymp k^{-\alpha}$, $\alpha > 2 + 1/q$. Assume the p.d.f. of ϵ_i satisfies $\int_{x\in\mathbb{R}} |f'_\epsilon(x)|\mathrm{d}x < \infty$ and $\int_{x\in\mathbb{R}} |f''_\epsilon(x)|\mathrm{d}x < \infty$. By Theorem 2.1 in Pham and Tran (1985), X_i is β-mixing and its β-mixing coefficient $\beta(k) = O(k^{1-(\alpha-1)q/(1+q)})$.

Assume functions $h \in \mathcal{H}$ are bounded and the function class $\mathcal{H}$ satisfies condition (D'). Also assume that a $\hat{\beta}$-stable algorithm yields an estimate $\hat{h}_S$ with $\hat{\beta} = O(n^{-1})$ where the definition for $\hat{\beta}$-stable can be found in Definition 4 of Mohri and Rostamizadeh (2010).

Let $K = 1/4 - (q+1)/(2(\alpha-1)q)$. By Corollary 21 in Mohri and Rostamizadeh (2010), there exists a constant $C > 0$ such that for $\delta > n^{-K}$,

$$\mathbb{P}(n|R_n(\hat{h}) - R(\hat{h})| \geq Cz_\delta) \leq \delta, \text{ where } z_\delta = n^{1-K}(\log(\delta - n^{-K})^{-1})^{1/2}. \tag{29}$$

By our Corollary 17,

$$\mathbb{P}(\sup_{h\in\mathcal{H}} n|R_n(h) - R(h)| \geq Cz_\delta) \lesssim \frac{n}{z_\delta^{q\alpha}}. \tag{30}$$

Note that, if $\delta > n^{-K}$, $nz_\delta^{-q\alpha} = O(n^{1-(1-K)q\alpha})$, which is of order $o(n^{-K})$ since $1-(1-K)q\alpha < -K$. To give a numeric example, let $\alpha = 4$, $q = 4$. Then $K = 1/24$, $1-(1-K)q\alpha = -43/3$. So (30) gives a much smaller upper bound $O(n^{-43/3})$, while (29) leads to the bound $O(n^{-1/24})$. The latter phenomenon could be explained by the sharpness of our upper bounds.

6. Proofs

In this section we shall provide proofs for results stated in the previous sections. We shall first introduce some notation. For $k \geq 1$ define the functions

$$g_k(x) := \mathbb{E}\Big[g(\sum_{i=0}^{k-1} a_i\epsilon_{-i} + x)\Big], \quad g_\infty(x) := \mathbb{E}[g(X_0 + x)]. \tag{31}$$

Since $a_0 = 1$, $g_1(x) = \mathbb{E}g(\epsilon_0 + x)$. Write $g_0(\cdot) = g(\cdot)$. Define projection operator P_k, $k \in \mathbb{Z}$, by $P_k\cdot = \mathbb{E}(\cdot|\mathcal{F}_k) - \mathbb{E}(\cdot|\mathcal{F}_{k-1})$, where $\mathcal{F}_i = (\epsilon_i, \epsilon_{i-1}, \ldots)$, that is $P_k f = \mathbb{E}(f(X)|\mathcal{F}_k) - \mathbb{E}(f(X)|\mathcal{F}_{k-1})$. For $j \leq i$, let

$$X_{i,j} = \sum_{k\geq 0} a_{i-j+k}\epsilon_{j-k}$$

be the truncated process. Then $X_{i,j} = \mathbb{E}(X_i|\mathcal{F}_j)$ and $g_{i-j}(X_{i,j}) = \mathbb{E}(g(X_i)|\mathcal{F}_j)$.

Let $\lfloor x \rfloor = \max\{i \in \mathbb{Z}, i \leq x\}$ and $\lceil x \rceil = \min\{i \in \mathbb{Z}, i \geq x\}$. Recall $\mu_q = (\mathbb{E}|\epsilon_0|^q)^{1/q}$ and let $\mu = \mu_1$.

In Section 6.1 we shall first present some inequalities and lemmas that will be extensively used. Theorem 4 and Proposition 5 (resp. Theorem 8 and Proposition 9) are proved in Section 6.2 (resp. Section 6.4). Theorem 7 (resp. Theorem 11, Theorem 14) is shown in Section 6.3 (resp. Section 6.5, Section 6.7). Section 6.6 gives proofs of Theorem 12 and Proposition 13. Proofs of Corollaries 15 and 16 are provided in Section 6.8.

6.1 Some useful lemmas

Lemma 18 is a maximal form of Freedman's martingale inequality (cf Freedman (1975)) and it is a simple modified version of Lemma 1 in Haeusler (1984). Lemma 19 is Burkholder's martingale inequality for moments (Burkholder (1988)). Lemma 20 is a Fuk-Nagaev inequality for high dimensional vectors (Chernozhukov et al. (2017)).

Lemma 18 *Let $\mathcal{A}$ be an index set with $|\mathcal{A}| < \infty$. For each $a \in \mathcal{A}$, let $\{\xi_{a,i}\}_{i=1}^n$ be a martingale difference sequence with respect to the filtration $\{\mathcal{F}_i\}_{i=1}^n$. Let $M_a = \sum_{i=1}^n \xi_{a,i}$ and $V_a = \sum_{i=1}^n \mathbb{E}[\xi_{a,i}^2|\mathcal{F}_{i-1}]$. Then for all $z, u, v > 0$*

$$\mathbb{P}\Big(\max_{a\in\mathcal{A}} |M_a| \geq z\Big) \leq \sum_{i=1}^n \mathbb{P}\Big(\max_{a\in\mathcal{A}} |\xi_{a,i}| \geq u\Big) + 2\mathbb{P}\Big(\max_{a\in\mathcal{A}} V_a \geq v\Big) + 2|\mathcal{A}|e^{-z^2/(2zu+2v)}.$$

Lemma 19 (Burkholder (1988), Rio (2009)) *Let $q > 1$, $q' = \min\{q, 2\}$. Let $M_T = \sum_{t=1}^T \xi_t$, where $\xi_t \in \mathcal{L}^q$ are martingale differences. Then*

$$\|M_T\|_q^{q'} \leq K_q^{q'} \sum_{t=1}^T \|\xi_t\|_q^{q'}, \text{ where } K_q = \max((q-1)^{-1}, \sqrt{q-1}).$$

Lemma 20 (A Fuk-Nagaev type inequality) *Let $X_1, \ldots, X_n$ be independent mean 0 random vectors in $\mathbb{R}^p$ and $\sigma^2 = \max_{1\leq j\leq p} \sum_{i=1}^n \mathbb{E}(X_{i,j}^2)$. Then for every $s > 1$ and $t > 0$,*

$$\mathbb{P}\Big(\max_{1\leq j\leq p} |\sum_{i=1}^n X_{i,j}| \geq 2\mathbb{E}(\max_{1\leq j\leq p} |\sum_{i=1}^n X_{i,j}|) + t\Big) \leq e^{-t^2/(3\sigma^2)} + \frac{K_s}{t^s}\sum_{i=1}^n \mathbb{E}(\max_{1\leq j\leq p} |X_{i,j}|^s),$$

where K_s is a constant depending only on s.

Lemma 21 *Assume that function g has second order derivative and $|g|, |g'|, |g''|$ are all bounded by $M < \infty$. Then $g_k, k \geq 1$, and g_∞ also have second order derivatives and $|g_k|, |g_k'|, |g_k''|$, $|g_\infty|, |g_\infty'|, |g_\infty''|$ are all bounded by M, where g_k and g_∞ are defined in* (31).

Proof Since $|g'|$ is bounded by M, by the dominated convergence theorem,

$$\lim_{\delta\to 0} \mathbb{E}\Big(\frac{g(\sum_{i=0}^{k-1} a_i\epsilon_{-i} + x + \delta) - g(\sum_{i=0}^{k-1} a_i\epsilon_{-i} + x)}{\delta}\Big) = \mathbb{E}\Big(g'(\sum_{i=0}^{k-1} a_i\epsilon_{-i} + x)\Big).$$

Since $g_k(x) = \mathbb{E}g(\sum_{i=0}^{k-1} a_i\epsilon_{-i} + x)$, $g_k'(x)$ exists and equals to $\mathbb{E}(g'(\sum_{i=0}^{k-1} a_i\epsilon_{-i} + x))$ with $|g_k'| \leq M$. Similarly g_k'' exists and $|g_k''|_\infty \leq M$. Note that $g_\infty(x) = \mathbb{E}g(\sum_{i=0}^{\infty} a_i\epsilon_{-i} + x)$. Hence same arguments lead to the existence of g_∞' and g_∞'', and they are also bounded in absolute value by M. ∎

Lemma 22 *Let $\lambda > 0$, $\beta > 1$ and $G(y) = \sum_{k=0}^\infty \min\{\lambda, (k \vee 1)^{-\beta} y\}$, $y > 0$. Then for all $y > 0$, $G(y) \leq K_{\beta,\lambda} \min\{y, y^{1/\beta}\}$, where $K_{\beta,\lambda} = \max\{(\beta-1)^{-1}, \lambda\} + 2$.*

Proof Clearly $G(y) \leq \sum_{k=0}^\infty (k \vee 1)^{-\beta} y \leq (2 + (\beta-1)^{-1})y$. If $y \geq 1$, we have $y^{1/\beta} \leq y$ and

$$\begin{aligned} G(y) &\leq \sum_{k=0}^{\lceil y^{1/\beta}\rceil} \lambda + \sum_{k=\lceil y^{1/\beta}\rceil+1}^\infty k^{-\beta} y \leq \lambda(y^{1/\beta} + 2) + (\beta-1)^{-1} y^{(1-\beta)/\beta} y \\ &\leq \max\{(\lambda+2), (\beta-1)^{-1}\} y^{1/\beta}. \end{aligned}$$

So the lemma follows by considering two cases $0 < y < 1$ and $y \geq 1$ separately. ∎

6.2 Proof of Theorem 4 and Proposition 5

The proof of Theorem 4 is quite involved. Here we shall first provide intuitive ideas of our martingale approximation approach. Recall the projection operator $P_k\cdot = \mathbb{E}(\cdot|\mathcal{F}_k) - \mathbb{E}(\cdot|\mathcal{F}_{k-1})$ and (31) for g_k and g_∞. Then $P_k g(X_i) = 0$ if $k > i$. Note that $\phi_j(g) := P_j S_n(g)$, $j = \ldots, n-1, n$, are martingale differences. Since $g_{i-j}(X_{i,j}) = \mathbb{E}(g(X_i)|\mathcal{F}_j)$, $j \le i$,

$$S_n(g) - \mathbb{E}S_n(g) = \sum_{j\le n}\phi_j(g), \text{ where } \phi_j(g) = \sum_{i=1\vee j}^{n}(g_{i-j}(X_{i,j}) - g_{i-j+1}(X_{i,j-1})). \tag{32}$$

Let $\epsilon_i, \epsilon_j', \epsilon_k'', i, j, k \in \mathbb{Z}$ be i.i.d. Since $g_{i-j+1}(x) = \mathbb{E}(g_{i-j}(x + a_{i-j}\epsilon_j))$, $g_{i-j}(x + a_{i-j}\epsilon_j) - g_{i-j+1}(x) = \mathbb{E}(\int_{a_{i-j}\epsilon_j'}^{a_{i-j}\epsilon_j} g_{i-j}'(x+t)\mathrm{d}t|\mathcal{F}_j)$. Note that $X_{i,j} - X_{i,j-1} = a_{i-j}\epsilon_j$. Then

$$g_{i-j}(X_{i,j}) - g_{i-j+1}(X_{i,j-1}) = \mathbb{E}\Big(\int_{a_{i-j}\epsilon_j'}^{a_{i-j}\epsilon_j} g_{i-j}'(x + X_{i,j-1})\mathrm{d}x|\mathcal{F}_j\Big). \tag{33}$$

Let $X_{i,j}'' = \sum_{k\ge 0} a_{i-j+k}\epsilon_{j-k}''$. Then $g_\infty(x) = \mathbb{E}(g_{i-j}(X_{i,j}'' + x)) = \mathbb{E}(g_{i-j}(X_{i,j}'' + x)|\mathcal{F}_j)$ and

$$g_\infty(a_{i-j}\epsilon_j) - \mathbb{E}g_\infty(a_{i-j}\epsilon_j) = \mathbb{E}\Big(\int_{a_{i-j}\epsilon_j'}^{a_{i-j}\epsilon_j} g_{i-j}'(x + X_{i,j}'')\mathrm{d}x|\mathcal{F}_j\Big). \tag{34}$$

Since $\|X_{i,j}\|_q \to 0$ as $j \to \infty$, intuitively we have $g_{i-j}'(x+X_{i,j-1}) \approx g_{i-j}'(x) \approx g_{i-j}'(x+X_{i,j}'')$. These relations (33) and (34) motivate us to approximate $S_n(g) - \mathbb{E}S_n(g)$ by

$$T_n(g) = \sum_{j\le n}\tilde{\phi}_j(g), \text{ where } \tilde{\phi}_j(g) = \sum_{i=1\vee j}^{n}\big(g_\infty(a_{i-j}\epsilon_j) - \mathbb{E}g_\infty(a_{i-j}\epsilon_j)\big). \tag{35}$$

Note that $\tilde{\phi}_j(g), j \le n$, are independent random variables. Hence we can apply corresponding inequalities. In Lemma 23 a Fuk-Nagaev type inequality for $T_n(g)$ is derived. Lemma 24 concerns the closeness of $S_n(g) - \mathbb{E}S_n(g)$ and $T_n(g)$. Similar arguments are also applied in the proofs of other theorems in the paper.

Proof We now proceed with the formal argument. By (11), there exists a set A_n such that for any $g \in \mathcal{A}$, $\min_{h\in A_n}|h-g|_\infty \le z/(4n)$ and $|A_n| = \mathcal{N}_{\mathcal{A}}(z/(4n))$. Then

$$\sup_{g\in\mathcal{A}}\Big|\sum_{i=1}^{n}\big[(g-\tau_n(g))(X_i) - \mathbb{E}(g-\tau_n(g))(X_i)\big]\Big| \le z/2,$$

where $\tau_n(g) := \mathrm{argmin}_{h\in A_n}|h-g|_\infty$. Hence $\Delta_n \le z/2 + \max_{g\in A_n}|S_n(g) - \mathbb{E}S_n(g)|$ and

$$\Delta_n \le \frac{z}{2} + \max_{g\in A_n}|S_n(g) - \mathbb{E}S_n(g) - T_n(g)| + \max_{g\in A_n}|T_n(g)| =: \frac{z}{2} + \Omega_n + U_n. \tag{36}$$

For $U_n = \max_{g\in A_n}|T_n(g)|$, by Lemma 23, we have

$$\mathbb{P}\Big(U_n \ge C_q a_* \mu_q c(n,q) + \frac{z}{4}\Big) \le \exp\Big(-\frac{z^2}{C_{\beta,\gamma}\mu_{q'}^{q'}n}\Big) + C_{\beta,q,\gamma,1}\mu_q^q\frac{n}{z^{q\beta}}. \tag{37}$$

For the difference term $\Omega_n = \max_{g\in A_n}|S_n(g) - \mathbb{E}S_n(g) - T_n(g)|$, by Lemma 24,

$$\mathbb{P}(\Omega_n \ge \frac{z}{4}) \le C_{\beta,q,\gamma,2}\mu_q^q\frac{n}{z^{q\beta}} + 2|A_n|\exp\Big(-\frac{z^2}{C_{\beta,\gamma}\mu_{q'}^{q'}n}\Big) + 2|A_n|\exp\Big(-\frac{z^v}{8\mu_q^{v'}}\Big), \tag{38}$$

where $C_{\beta,q,\gamma,1}$ and $C_{\beta,q,\gamma,2}$ are constants only depending on β, q, γ and $C_{\beta,q,\gamma} = C_{\beta,q,\gamma,1} + C_{\beta,q,\gamma,2}$. Combining (36), (37) and (38), we complete the proof. ■

Lemma 23 *Recall the definitions of $\tilde{\phi}_j(g)$ and $T_n(g)$ in* (32) *and* (35) *respectively. Under assumptions of Theorem 4, we have* (37).

Proof Recall $U_n = \max_{g\in A_n}|T_n(g)|$. The proof contains two parts:

(i). Apply the Fuk-Nagaev type inequality (Lemma 20) to bound $\mathbb{P}(U_n - 2\mathbb{E}U_n \ge z/4)$.

(ii). Show that $2\mathbb{E}U_n \le C_q a_* \mu_q c(n,q)$.

Part (i): For $g \in A_n$, since $|g|, |g'|$ are bounded by 1, by Lemma 21, $|g_\infty|$ and $|g_\infty'|$ are also bounded by 1. Then

$$|\tilde{\phi}_j(g)| = \Big|\sum_{i=1\vee j}^{n}\mathbb{E}\big(\int_{a_{i-j}\epsilon_j'}^{a_{i-j}\epsilon_j} g_\infty'(x)\mathrm{d}x|\mathcal{F}_j\big)\Big| \le \sum_{i=1\vee j}^{n}\min\big\{|a_{i-j}|(|\epsilon_j| + \mu),\, 2\big\}. \tag{39}$$

Therefore for $j < -n$ and any $g \in A_n$, by (39),

$$|\tilde{\phi}_j(g)| \le \min\{\gamma n(-j)^{-\beta}(|\epsilon_j| + \mu), 2n\}, \tag{40}$$

for $-n \le j \le n$ and any $g \in A_n$, by Lemma 22 and (39),

$$|\tilde{\phi}_j(g)| \le \gamma K_{\beta,2/\gamma}(|\epsilon_j| + \mu)^{1/\beta}. \tag{41}$$

Denote $V = \max_{g\in A_n}\sum_{j\le n}\mathbb{E}\tilde{\phi}_j^2(g)$. Hence by (40) and (41),

$$\begin{aligned} V &\le \sum_{j<-n}(\gamma n(-j)^{-\beta})^{q'}\mathbb{E}(|\epsilon_0| + \mu)^{q'}(2n)^{2-q'} + (\gamma K_{\beta,2/\gamma})^2\sum_{-n\le j\le n}\mathbb{E}[(|\epsilon_j| + \mu)^{2/\beta}] \\ &\le \Big(\frac{4\gamma^2}{\beta-1} + 2^{1+2/\beta}(\gamma K_{\beta,2/\gamma})^2\Big)n\mu_{q'}^{q'}. \end{aligned} \tag{42}$$

By (40),

$$\sum_{j<-n}\mathbb{E}\big(\max_{g\in A_n}|\tilde{\phi}_j|^{q\beta}\big) \le \sum_{j<-n}(2n)^{q\beta-q}(\gamma n(-j)^{-\beta})^q\mathbb{E}[(|\epsilon_j| + \mu)^q] \le \frac{2^{q\beta}\gamma^q}{q\beta-1}n\mu_q^q. \tag{43}$$

By (41),

$$\sum_{-n\le j\le n}\mathbb{E}\big(\max_{g\in A_n}|\tilde{\phi}_j|^{q\beta}\big) \le 2n(\gamma K_{\beta,2/\gamma})^{q\beta}\mathbb{E}[(|\epsilon_j| + \mu)^q] \le 2^{q+1}(\gamma K_{\beta,2/\gamma})^{q\beta}n\mu_q^q. \tag{44}$$

Inserting the bounds (42), (43) and (44) into Lemma 20, we obtain

$$\begin{aligned}\mathbb{P}(U_n - 2\mathbb{E}U_n \geq z/4) \leq & e^{-z^2/(48V)} + \frac{4^{q\beta}K_{q\beta}}{z^{q\beta}}\sum_{j\leq n}\mathbb{E}\big(\max_{g\in A_n}|\tilde{\phi}_j|^{q\beta}\big) \\ \leq & \exp\Big(-\frac{z^2}{C_{\beta,\gamma}\mu_{q'}^{q'}n}\Big) + C_{\beta,q,\gamma,1}\mu_q^q\frac{n}{z^{q\beta}}, \end{aligned} \tag{45}$$

where $C_{\beta,\gamma} = 48(4\gamma^2/(\beta-1) + 2^{1+2/\beta}(\gamma K_{\beta,2/\gamma})^2)$ and $C_{\beta,q,\gamma,1} = 4^{q\beta}K_{q\beta}(2^{q\beta}\gamma^q/(q\beta-1) + 2^{q+1}(\gamma K_{\beta,2/\gamma})^{q\beta})$.

Part (ii): Recall $a_* = \sum_{k=0}^{\infty}|a_k|$. Note that $T_n(g)$ can be rewritten as

$$\begin{aligned}T_n(g) = & \sum_{j\leq n}\tilde{\phi}_j(g) = \sum_{k\geq 0}\sum_{i=1}^{n}\{g_\infty(a_k\epsilon_{i-k}) - \mathbb{E}g_\infty(a_k\epsilon_{i-k})\} \\ = & \sum_{k\geq 0}\int_{-\infty}^{\infty}\sum_{i=1}^{n}\big(\mathbf{1}_{a_k\epsilon_{i-k}\geq x} - \mathbb{P}(a_k\epsilon_{i-k}\geq x)\big)g_\infty'(x)\mathrm{d}x.\end{aligned}$$

Let $W_n(x) = \sum_{i=1}^{n}\big(\mathbf{1}_{\epsilon_i\geq x} - \mathbb{P}(\epsilon_i\geq x)\big)$. By Lemma 21, $|g_\infty'(x)|\leq 1$. Then

$$\begin{aligned}\mathbb{E}\Big[\max_{g\in A_n}|T_n(g)|\Big] \leq & \sum_{k\geq 0}\int_{-\infty}^{\infty}\mathbb{E}|\sum_{i=1}^{n}\big(\mathbf{1}_{a_k\epsilon_{i-k}\geq x} - \mathbb{P}(a_k\epsilon_{i-k}\geq x)\big)|\mathrm{d}x \\ = & \sum_{k\geq 0}\int_{-\infty}^{\infty}\mathbb{E}|W_n(x/a_k)|\mathrm{d}x = a_*\int_{-\infty}^{\infty}\mathbb{E}|W_n(y)|\mathrm{d}y, \end{aligned} \tag{46}$$

where the last equality is obtained by change of variables $y = x/a_k$ and $a_* = \sum_{k=0}^{\infty}|a_k|$. Let $T_F(x) = \mathbb{P}(|\epsilon_0|\geq|x|)$. Note that $\mathbb{E}|\mathbf{1}_{\epsilon_i\geq x} - \mathbb{P}(\epsilon_i\geq x)| = 2F_\epsilon(x)(1-F_\epsilon(x)) \leq 2T_F(x)$, and $\mathbb{E}(\mathbf{1}_{\epsilon_i\geq x} - \mathbb{P}(\epsilon_i\geq x))^2 = F_\epsilon(x)(1-F_\epsilon(x)) \leq T_F(x)$. Hence

$$\mathbb{E}|W_n(x)| \leq \min\{\|W_n(x)\|, 2nT_F(x)\} \leq \min\{\sqrt{n}T_F(x)^{1/2}, 2nT_F(x)\}. \tag{47}$$

We have different bounds for (46) when $q>2$, $1<q<2$ and $q=2$. By Markov's inequality,

$$T_F(x) \leq \min\{|x|^{-q}\mu_q^q, 1\}. \tag{48}$$

When $q>2$, we have

$$\int_{-\infty}^{\infty}T_F(x)^{1/2}\mathrm{d}y \leq 2\Big(\int_0^{\mu_q}1\mathrm{d}x + \int_{\mu_q}^{\infty}|x|^{-q/2}\mu_q^{q/2}\mathrm{d}x\Big) = q/(q/2-1)\mu_q.$$

Inserting above into (46) and (47), we obtain

$$\mathbb{E}U_n \leq a_*\int_{-\infty}^{\infty}\mathbb{E}|W_n(x)|\mathrm{d}x \leq q/(q/2-1)a_*\mu_q\sqrt{n}. \tag{49}$$

When $1<q<2$, by (47) and (48),

$$\begin{aligned}\int_{-\infty}^{\infty}\mathbb{E}|W_n(x)|\mathrm{d}x &\leq 2\Big(\int_0^{n^{1/q}\mu_q}\sqrt{n}x^{-q/2}\mu_q^{q/2}\mathrm{d}x + \int_{n^{1/q}\mu_q}^{\infty}2nx^{-q}\mu_q^q\mathrm{d}x\Big) \\ &\leq 4(1/(2-q)+1/(q-1))\mu_q n^{1/q}.\end{aligned}$$

When $q=2$, $\mathrm{I}_1 := \int_{|x|\leq\mu_2}\sqrt{n}T_F(x)^{1/2}\mathrm{d}x \leq 2\mu_2\sqrt{n}$. By (48), $\mathrm{I}_2 := \int_{|x|>n\mu_2}2nT_F(x)\mathrm{d}x \leq 4\int_{n\mu_2}^{\infty}n\mu_2^2x^{-2}\mathrm{d}x = 4\mu_2$. By the Cauchy-Schwarz inequality,

$$\begin{aligned}\mathrm{I}_3^2 &:= \Big[\int_{\mu_2<|x|\leq n\mu_2}\sqrt{n}T_F(x)^{1/2}\mathrm{d}x\Big]^2 \leq 4n\int_{\mu_2}^{n\mu_2}xT_F(x)\mathrm{d}x\int_{\mu_2}^{n\mu_2}x^{-1}\mathrm{d}x \\ &\leq 4n\int_0^{\infty}x\mathbb{P}(|\epsilon_0|\geq x)\mathrm{d}x(\log n) = 2\mathbb{E}(\epsilon_0^2)n\log(n) = 2\mu_2^2n\log(n).\end{aligned}$$

Then by (47), $\int_{-\infty}^{\infty}\mathbb{E}|W_n(x)|\mathrm{d}x \leq \mathrm{I}_1+\mathrm{I}_2+\mathrm{I}_3 \leq 2\mu_2\sqrt{n} + 4\mu_2 + \mu_2(2n\log n)^{1/2}$. Combining the three cases $q>2$, $1<q<2$ and $q=2$, by (46), we have $\mathbb{E}U_n \leq c_qa_*\mu_qc(n,q)$. where $c_q = \max\{q/(q/2-1), 4(1/(2-q)+1/(q-1)), 6+\sqrt{2}\}$. ■

Lemma 24 *Recall the definitions of $\phi_j(g)$, $\tilde{\phi}_j(g)$ and $T_n(g)$ in* (32) *and* (35). *Under conditions of Theorem 4, we have* (38).

Proof Since $S_n(g) - \mathbb{E}S_n(g) - T_n(g)$ is the sum of martingale differences $\phi_j(g) - \tilde{\phi}_j(g)$, $j\leq n$, we can apply Lemma 18 to bound the tail probability. To this end, we shall:

(i). Derive the upper bound for $\mathrm{I}_1 = \sum_{j\leq n}\mathbb{P}(\max_{g\in A_n}|\phi_j(g) - \tilde{\phi}_j(g)|\geq u)$.

(ii). Bound the term $\mathrm{I}_2 = \max_{g\in A_n}\sum_{j\leq n}\mathbb{E}[(\phi_j(g) - \tilde{\phi}_j(g))^2|\mathcal{F}_{j-1}]$.

First we derive some inequalities that will be used for I_1 and I_2. Let $\epsilon_i, \epsilon_j', \epsilon_k'', i,j,k\in\mathbb{Z}$, be i.i.d. and $X_{i,j}'' = \sum_{k\geq 0}a_{i-j+k}\epsilon_{j-k}''$. Write $\phi_j(g) - \tilde{\phi}_j(g) = \sum_{i=1\vee j}^{n}d_{i,j}(g)$, where

$$\begin{aligned}d_{i,j}(g) =& g_{i-j}(X_{i,j}) - g_{i-j+1}(X_{i,j-1}) - g_\infty(a_{i-j}\epsilon_j) + \mathbb{E}g_\infty(a_{i-j}\epsilon_j) && (D_1)\\ =& \mathbb{E}\Big[\int_{a_{i-j}\epsilon_j'}^{a_{i-j}\epsilon_j}\big(g_{i-j}'(x+X_{i,j-1}) - g_{i-j}'(x+X_{i,j}'')\big)\mathrm{d}x\big|\mathcal{F}_j\Big] && (D_2)\\ =& \mathbb{E}\Big[\int_{X_{i,j}''}^{X_{i,j-1}}\big(g_{i-j}'(x+a_{i-j}\epsilon_j) - g_{i-j}'(x+a_{i-j}\epsilon_j')\big)\mathrm{d}x\big|\mathcal{F}_j\Big] && (D_3)\\ =& \mathbb{E}\Big[\int_{a_{i-j}\epsilon_j'}^{a_{i-j}\epsilon_j}\int_{X_{i,j}''}^{X_{i,j-1}}g_{i-j}''(x+y)\mathrm{d}y\mathrm{d}x\big|\mathcal{F}_j\Big]. && (D_4)\end{aligned}$$

By Lemma 21, $|g_j|, |g_j'|$ and $|g_j''|$ are bounded by 1. Hence by (D_1)-(D_4), we have

$$\begin{aligned}&\max_{g\in A_n}|d_{i,j}(g)| \\ \leq& \min\Big\{4, 2|a_{i-j}|(|\epsilon_j|+\mu), 2(|X_{i,j-1}| + \mathbb{E}|X_{i,j}|), |a_{i-j}|(|\epsilon_j|+\mu)(|X_{i,j-1}| + \mathbb{E}|X_{i,j}|)\Big\} \\ =& \min\Big\{|a_{i-j}|(|\epsilon_j|+\mu), 2\Big\}\min\Big\{(|X_{i,j-1}| + \mathbb{E}|X_{i,j}|), 2\Big\}. \end{aligned} \tag{50}$$

Part (i): Recall $q' = \min(q, 2)$. For $i > j$, by Lemma 19,

$$\|X_{i,j-1}\|_q^{q'} \le K_q^{q'} \sum_{k\ge 1} (|a_{i-j+k}|\|\epsilon_{j-k}\|_q)^{q'} \le \big(K_q^{q'}\gamma^{q'}(\beta q'-1)^{-1}\big)(i-j)^{-q'\beta+1}\mu_q^{q'}. \quad (51)$$

Let $r = (q'\beta - 1)/(2q')$, by Markov's inequality,

$$\mathrm{I}_1 \le \sum_{-n\le j\le n} u^{-q(\beta+r)}\mathbb{E}[\max_{g\in A_n}|\phi_j(g) - \tilde\phi_j(g)|^{q(\beta+r)}] + \sum_{j<-n} u^{-q}\mathbb{E}[\max_{g\in A_n}|\phi_j(g) - \tilde\phi_j(g)|^q]. \quad (52)$$

We shall consider the two cases $-n \le j \le n$ and $j < -n$ separately. For $-n \le j \le n$, by (50) and since ϵ_j and $X_{i,j-1}$ are independent,

$$\begin{aligned}
&\|\max_{g\in A_n}|\phi_j(g) - \tilde\phi_j(g)|\|_{q(\beta+r)} \\
&\le \sum_{i=j\vee 1}^{n} \big\|\min\{|a_{i-j}|(|\epsilon_j|+\mu), 2\}\big\|_{q(\beta+r)} \big\|\min\{|X_{i,j-1}| + \mathbb{E}|X_{i,j}|, 2\}\big\|_{q(\beta+r)} \\
&\le \sum_{i=j\vee 1}^{n} \Big(|a_{i-j}|^q\mathbb{E}(|\epsilon_j|+\mu)^q 2^{q(\beta+r)-q}\Big)^{1/q(\beta+r)} \Big(\mathbb{E}(|X_{i,j-1}| + \mathbb{E}|X_{i,j}|)^q 2^{q(\beta+r)-q}\Big)^{1/q(\beta+r)}.
\end{aligned}$$

By (51) and $2\beta q' - 1 > (\beta + r)q'$, above inequality is further bounded by

$$\|\max_{g\in A_n}|\phi_j(g) - \tilde\phi_j(g)|\|_{q(\beta+r)} \le c_1 \sum_{i=j\vee 1}^{n} ((i-j)\vee 1)^{\frac{-2\beta q'+1}{(\beta+r)q'}}\mu_q^{2/(\beta+r)} \le c_2\mu_q^{2/(\beta+r)}, \quad (53)$$

where $c_1 = (K_q\gamma(\beta q'-1)^{-1}2^{\beta+r})^{1/(\beta+r)}$ and $c_2 = 4(2\beta q'-1)(\beta q'-1)^{-1}c_1$.

For $j < -n$, again by (50) and the independence between ϵ_j and $X_{i,j-1}$,

$$\begin{aligned}
\|\max_{g\in A_n}|\phi_j(g) - \tilde\phi_j(g)|\|_q &\le \sum_{i=1}^{n} |a_{i-j}|\||\epsilon_j| + \mu\|_q\|X_{i,j-1}| + \mathbb{E}|X_{i,j}|\|_q \\
&\le \big(4\gamma(\beta q'-1)^{1/q'}\big)n(-j)^{-\frac{2\beta q'-1}{q'}}\mu_q^2, \quad (54)
\end{aligned}$$

where the last inequality is due to (51).

Applying (53) and (54) to (52), we have

$$\mathrm{I}_1 \le c_3\mu_q^{2q}nu^{-\beta(q+r)}, \text{ where } c_3 = 2c_2^{q(\beta+r)} + (4\gamma(\beta q'-1)^{1/q'})^q.$$

Part (ii): We shall bound $\max_{g\in A_n}|\phi_j(g) - \tilde\phi_j(g)|$ for $-n \le j \le n$ and $j < -n$ seperately. For $-n \le j \le n$, by (50) and Lemma 22,

$$\max_{g\in A_n}|\phi_j(g) - \tilde\phi_j(g)| \le \sum_{i=1\vee j}^{n} \min\{|a_{i-j}|(|\epsilon_j|+\mu), 2\} \le \gamma K_{\beta,\gamma/2}(|\epsilon_j|+\mu)^{1/\beta},$$

Since ϵ_j is independent of $\mathcal{F}_{j-1}$, we have

$$\begin{aligned}
\mathrm{I}_{21} &:= \sum_{-n\le j\le n} \mathbb{E}\big[\max_{g\in A_n}(\phi_j(g) - \tilde\phi_j(g))^2|\mathcal{F}_{j-1}\big] \\
&\le \sum_{-n\le j\le n} (\gamma K_{\beta,\gamma/2})^2\mathbb{E}[(|\epsilon_j|+\mu)^{2/\beta}] \le \big(2^{1+2/\beta}(\gamma K_{\beta,\gamma/2})^2\big)n\mu_{q'}^{q'}.
\end{aligned}$$

For $j < -n$, by Lemma 22,

$$\max_{g\in A_n}|\phi_j(g) - \tilde\phi_j(g)| \le n\min\{\gamma(-j)^{-\beta}(|\epsilon_j|+\mu), 2\}.$$

Since ϵ_j is independent of $\mathcal{F}_{j-1}$, we have

$$\begin{aligned}
\mathrm{I}_{22} &:= \sum_{j<-n} \mathbb{E}\big[\max_{g\in A_n}(\phi_j(g) - \tilde\phi_j(g))^2|\mathcal{F}_{j-1}\big] \\
&\le n^2\sum_{j<-n} 2^{2-q'}\gamma^{q'}(-j)^{-q'\beta}\mathbb{E}[(|\epsilon_j|+\mu)^{q'}] \le \big(4\gamma^2/(\beta-1)\big)n\mu_{q'}^{q'},
\end{aligned}$$

Hence we have $\mathrm{I}_2 = \mathrm{I}_{21} + \mathrm{I}_{22} \le c_4 n\mu_{q'}^{q'}$, where $c_4 = 2^{1+2/\beta}(\gamma K_{\beta,\gamma/2})^2 + 4\gamma^{q'}/(q'\beta-1)$.

Inserting the bounds for I_1 and I_2 into Lemma 18 leads to

$$\mathbb{P}(\Omega_n \ge z/4) \le c_3 n\mu_q^{2q}u^{-q(\beta+r)} + 2|A_n|\exp\Big(-\frac{z^2}{32c_4\mu_{q'}^{q'}n}\Big) + 2|A_n|\exp\Big(-\frac{z^2}{8zu}\Big). \quad (55)$$

Take $u = z^{\beta/(\beta+r)}\mu_q^{1/(\beta+r)}$ and we complete the proof. ■

Remark 25 *Let $K_{q\beta}$ (resp. K_q and $K_{\beta,2/\gamma}$) be the constant in Lemma 20 (resp. Lemma 19 and Lemma 22). With a careful check of the proofs of Theorem 4, Lemmas 23 and 24, we can choose constants in Theorem 4 as follows:*

- $C_q = 2\max\{q/(q/2-1), 4(1/(2-q) + 1/(q-1)), 6+\sqrt{2}\}$.
- $C_{\beta,q,\gamma} = C_{\beta,q,\gamma,1} + C_{\beta,q,\gamma,2}$, *where* $C_{\beta,q,\gamma,1} = 4^{q\beta}K_{q\beta}(2^{q\beta}\gamma^q/(q\beta-1) + 2^{q+1}(\gamma K_{\beta,2/\gamma})^{q\beta})$ *and* $C_{\beta,q,\gamma,2} = 2c_2^{q(\beta+r)} + (4\gamma(\beta q'-1)^{1/q'})^q$ *with* $r = (q'\beta-1)/(2q')$, $c_1 = (K_q\gamma(\beta q'-1)^{-1}2^{\beta+r})^{1/(\beta+r)}$ *and* $c_2 = 4(2\beta q'-1)(\beta q'-1)^{-1}c_1$.
- $C_{\beta,\gamma} = 48\big(4\gamma^2/(\beta-1) + 2^{1+2/\beta}(\gamma K_{\beta,2/\gamma})^2\big)$.

Proof [Proof of Proposition 5] Construct A_n as in the proof of Theorem 4. Recall (31) for the function g_k. Note that $g_1(X_{i,i-1}) = \mathbb{E}[g(X_i)|\mathcal{F}_{i-1}]$. By (36), we have

$$\begin{aligned}
\mathbb{P}(|\Delta_n| \ge a+z) \le&\mathbb{P}\Big(\max_{g\in A_n}|S_n(g) - \mathbb{E}S_n(g)| \ge a + z/2\Big) \\
\le&\mathbb{P}\Big(\max_{g\in A_n}\Big|\sum_{i=1}^{n}\big(g_1(X_{i,i-1}) - \mathbb{E}g_1(X_{i,i-1})\big)\Big| \ge a + z/4\Big) \\
&+ \sum_{g\in A_n}\mathbb{P}\Big(\Big|\sum_{i=1}^{n}(g(X_i) - \mathbb{E}[g(X_i)|\mathcal{F}_{i-1}])\Big| \ge z/4\Big) =: \mathrm{I}_1 + \mathrm{I}_2,
\end{aligned}$$

where $a = C_q a_* \mu_q c(n,q)$.

Since $|g| \le 1$ and $g(X_i) - \mathbb{E}[g(X_i)|\mathcal{F}_{i-1}]$, $1 \le i \le n$, are martingale differences, by Azuma's inequality, $\mathrm{I}_2 \le 2|A_n|\exp\{-z^2/(64n)\}$. For I_1, notice

$$g_1(x) = \int_{-\infty}^{\infty} g(x+y) f_\epsilon(y)\mathrm{d}y = \int_{-\infty}^{\infty} g(y) f_\epsilon(y-x)\mathrm{d}y.$$

By Assumption (A'), $\sup_{g\in\mathcal{A}}|g_1|_\infty$, $\sup_{g\in\mathcal{A}}|g_1'|_\infty$ and $\sup_{g\in\mathcal{A}}|g_1''|_\infty$ are all bounded by 1. Thus in the I_1 part, the function g_1 satisfies Assumption (A) and can be dealt with by Theorem 4. Combining I_1 and I_2, we complete the proof. ■

6.3 Proof of Theorem 7

Proof [Proof of Theorem 7] Recall the projection operator $P_k\cdot = \mathbb{E}(\cdot|\mathcal{F}_k) - \mathbb{E}(\cdot|\mathcal{F}_{k-1})$. Let $D_k = P_k\Delta_n$, $k \le n$. Then $\Delta_n - \mathbb{E}\Delta_n = \sum_{k\le n} D_k$ and

$$\mathbb{P}(\Delta_n - \mathbb{E}\Delta_n \ge z) \le \mathbb{P}\Big(\sum_{k\le -n} D_k \ge z/2\Big) + \mathbb{P}\Big(\sum_{-n<k\le n} D_k \ge z/2\Big) =: \mathrm{I}_1 + \mathrm{I}_2. \tag{56}$$

Then the theorem follows from the following three claims which will be proved in the sequel:

(i). $\mathrm{I}_1 \le C_{q,\gamma} e^{-qn(1-\rho)} \mu_q^q (z^q(1-\rho)^{q+q/q'})^{-1}$.

(ii). $\mathrm{I}_2 \le \exp\{-C'_{q,\gamma} z^2(1-\rho)^2((\mu_q^q \vee 1)n)^{-1}\}$.

(iii). $\mathbb{E}\Delta_n \le C_q \mu_q c^*(n,\rho,q)$.

To prove (i) and (ii), we need to apply coupling. Let $\epsilon_i, \epsilon'_j, i, j \in \mathbb{Z}$, be i.i.d. For a random variable $Z = H(\varepsilon)$, where H is a measurable function and $\varepsilon = (\epsilon_i)_{i\in\mathbb{Z}}$, we define the coupled version $Z_{\{j\}} = H(\varepsilon'_{\{j\}})$, where $\varepsilon'_{\{j\}} = (...,\epsilon_{j-1},\epsilon'_j,\epsilon_{j+1},...)$. We shall now derive an upper bound for $|D_k|$. Since $|g|, |g'|$ are bounded by 1, for any $k \le i$,

$$\mathbb{E}\big(\sup_{g\in\mathcal{A}} |g(X_i) - g(X_{i,\{k\}})|\big|\mathcal{F}_k\big) \le \mathbb{E}\big(|X_i - X_{i,\{k\}}|\big|\mathcal{F}_k\big) \le |a_{i-k}|(|\epsilon_k| + \mu). \tag{57}$$

Note $\mathbb{E}(\Delta_n|\mathcal{F}_{k-1}) = \mathbb{E}(\Delta_{n,\{k\}}|\mathcal{F}_k)$, thus $D_k = \mathbb{E}(\Delta_n - \Delta_{n,\{k\}}|\mathcal{F}_k)$ and by (57),

$$|D_k| \le \mathbb{E}\Big(\sup_{g\in\mathcal{A}}\Big|\sum_{i=1}^n [g(X_i) - g(X_{i,\{k\}})]\Big|\Big|\mathcal{F}_k\Big) \le \sum_{i=1\vee k}^n \min\{|a_{i-k}|(|\epsilon_k|+\mu), 2\}. \tag{58}$$

Part (i): Since D_k are martingale differences, by Lemma 19,

$$\mathrm{I}_1 \le (z/2)^{-q}\|\sum_{k\le -n} D_k\|_q^q \le K_q^q (z/2)^{-q}\Big(\sum_{k\le -n} \|D_k\|_q^{q'}\Big)^{q/q'}. \tag{59}$$

Since (58) implies $|D_k| \le \gamma\rho^{-k}(1-\rho)^{-1}(|\epsilon_k|+\mu)$ for any $k \le -n$, we further obtain from (59) and the elementary inequality $\log(\rho^{-1}) \ge 1-\rho$ that

$$\mathrm{I}_1 \le (4K_q\gamma)^q \frac{\rho^{nq}\mu_q^q}{z^q(1-\rho)^q(1-\rho^{q'})^{q/q'}} \le (4K_q\gamma)^q \frac{e^{-nq(1-\rho)}\mu_q^q}{z^q(1-\rho)^{q+q/q'}}. \tag{60}$$

Part (ii): Note for any $y \ge 1$, since $\log(\rho^{-1}) \ge 1-\rho$,

$$\begin{aligned}\sum_{i\ge 0}\min(\rho^i y, 1) &\le \sum_{i\ge -\log_\rho y} \rho^i y + (-\log_\rho y)\\ &\le (1-\rho)^{-1} - \log_\rho y \le (1-\rho)^{-1}[1+\log(y)].\end{aligned} \tag{61}$$

Hence for $k > -n$, by (58) and (61),

$$|D_k| \le (2\vee\gamma)(1-\rho)^{-1}\big[1 + \log(|\epsilon_k|+\mu)\mathbf{1}_{\{|\epsilon_k|+\mu\ge 1\}}\big]. \tag{62}$$

Let $h^* := (2\vee\gamma)^{-1}(1-\rho)q$. Since $\epsilon_k \in \mathcal{L}^q$, for any $0 < h \le h^*$, $\mathbb{E}(e^{D_k h}) < \infty$. Note $\mathbb{E}(D_k|\mathcal{F}_{k-1}) = 0$, then

$$\begin{aligned}\mathbb{E}(e^{D_k h}|\mathcal{F}_{k-1}) &= 1 + \mathbb{E}(e^{D_k h} - D_k h - 1|\mathcal{F}_{k-1})\\ &\le 1 + \mathbb{E}\Big[\frac{e^{|D_k|h} - |D_k|h - 1}{h^2(1-\rho)^{-2}}\Big|\mathcal{F}_{k-1}\Big]h^2(1-\rho)^{-2}\end{aligned} \tag{63}$$

in view of $e^x - x \le e^{|x|} - |x|$ for any x. Note that for any fixed $x > 0$, $(e^{tx} - tx - 1)/t^2$ is increasing on $t \in (0,\infty)$. Applying the upper bound of D_k in (62), we have

$$\begin{aligned}\mathbb{E}\Big[\frac{e^{|D_k|h} - |D_k|h - 1}{h^2(1-\rho)^{-2}}\Big|\mathcal{F}_{k-1}\Big] &\le \mathbb{E}\Big[\frac{e^{|D_k|h^*} - |D_k|h^* - 1}{h^{*2}(1-\rho)^{-2}}\Big|\mathcal{F}_{k-1}\Big]\\ &\le \mathbb{E}\Big[\frac{e^{q[1+\log(|\epsilon_k|+\mu)1_{\{|\epsilon_k|+\mu>1\}}]}}{h^{*2}(1-\rho)^{-2}}\Big|\mathcal{F}_{k-1}\Big] \le c_1\mu_q^q,\end{aligned} \tag{64}$$

where $c_1 = 2^q e^q (2\vee\gamma)^2 q^{-2}$. Hence for any $h \le h^*$,

$$\mathbb{E}(e^{D_k h}|\mathcal{F}_{k-1}) \le 1 + c_1\mu_q^q h^2(1-\rho)^{-2}. \tag{65}$$

By Markov's inequality we have $\mathrm{I}_2 \le e^{-zh/2}\mathbb{E}[\exp(\sum_{-n<k\le n} D_k h)]$. Then by recursively applying (65), let $h = z(1-\rho)^2[8c_1(\mu_q^q\vee 1)n]^{-1} \le h^*$, we further obtain

$$\begin{aligned}\mathrm{I}_2 \le& e^{-zh/2}\mathbb{E}\Big(e^{\sum_{k=-n+1}^{n-1} D_k h}\mathbb{E}(e^{D_n h}|\mathcal{F}_{n-1})\Big) \le e^{-zh/2}(1 + c_1\mu_q^q h^2/(1-\rho)^2)^{2n}\\ \le& \exp\big(-zh/2 + 2nc_1\mu_q^q h^2/(1-\rho)^2\big) \le \exp\Big(-\frac{z^2(1-\rho)^2}{32c_1(\mu_q^q\vee 1)n}\Big),\end{aligned} \tag{66}$$

where the third inequality is due to $1 + x \le e^x$ for $x > 0$.

Part (iii): Note

$$\begin{aligned}g(X_i) - \mathbb{E}g(X_i) &= \sum_{j\ge 0}(g_j(X_{i,i-j}) - g_{j+1}(X_{i,i-j-1}))\\ &= \sum_{j\ge 0}\int_{-\infty}^{\infty} g_j'(x)\big(\mathbf{1}_{x\le X_{i,i-j}} - \mathbb{E}(\mathbf{1}_{x\le X_{i,i-j}}|\mathcal{F}_{i-j-1})\big)\mathrm{d}x.\end{aligned}$$

By above inequality and that $|g_j'|$ are bounded by 1,

$$\mathbb{E}(\Delta_n) \le \sum_{j\ge 0}\int_{-\infty}^{\infty}\mathbb{E}\Big|\sum_{i=1}^{n}\big(\mathbf{1}_{x\le X_{i,i-j}} - \mathbb{E}(\mathbf{1}_{x\le X_{i,i-j}}|\mathcal{F}_{i-j-1})\big)\Big|\mathrm{d}x. \tag{67}$$

Let $H_j(x) = \mathbb{P}(|X_{0,-j}| \ge |x|)$. Since for any fixed j, $\mathbf{1}_{x\le X_{i,i-j}} - \mathbb{E}(\mathbf{1}_{x\le X_{i,i-j}}|\mathcal{F}_{i-j-1})$, $i = 1,\dots,n$, are martingale differences, by the same arguments as for (47), we have

$$\mathbb{E}\Big|\sum_{i=1}^{n}\big(\mathbf{1}_{x\le X_{i,i-j}} - \mathbb{E}(\mathbf{1}_{x\le X_{i,i-j}}|\mathcal{F}_{i-j-1})\big)\Big| \le \min\{\sqrt{n}H_j(x)^{1/2}, nH_j(x)\}. \tag{68}$$

For any $1 < r \le q$ and $r' = \min\{r, 2\}$, by Lemma 19,

$$H_j(x) \le \frac{\|X_{0,-j}\|_r^r}{|x|^r} \le \frac{K_r^r}{|x|^r}\Big(\sum_{k\ge j}|a_k|^{r'}\mu_r^{r'}\Big)^{r/r'} \le \frac{K_r^r}{|x|^r}\rho^{jr}(1-\rho)^{-r/r'}\mu_r^r. \tag{69}$$

We need to deal with the three cases separately: $q > 2$, $1 < q < 2$ and $q = 2$.
Case $q > 2$: Let $r = 3/2$. By (67) and (68),

$$\mathbb{E}(\Delta_n) \le \sum_{j\ge 0}\sqrt{n}\int_{-\infty}^{\infty}H_j(x)^{1/2}\mathrm{d}x.$$

Since $1-\rho^x \ge 1-\rho$ for $x \ge 1$ and $1-\rho^x \ge 1-\rho^{1/2} \ge (1-\rho)/2$ for $1/2 \le x < 1$, by (69),

$$\begin{aligned}\mathbb{E}(\Delta_n) &\le (K_r \vee K_q)^{q/2}\sum_{j\ge 0}\sqrt{n}\Big(\int_{|x|>\mu_q}\rho^{jq/2}(1-\rho)^{-q/4}|x|^{-q/2}\mu_q^{q/2}\mathrm{d}x \\ &\quad + \int_{|x|\le\mu_q}\rho^{jr/2}(1-\rho)^{-1/2}|x|^{-r/2}\mu_r^{r/2}\mathrm{d}x\Big) \\ &\le (K_r \vee K_q)^{q/2}(2/(q-2)+8)\sqrt{n}(1-\rho)^{-3/2}\mu_q.\end{aligned}$$

Case $1 < q < 2$: By (67) and (68), for $a = n^{1/q}(1-\rho)^{-1/q}K_q\mu_q$,

$$\mathbb{E}(\Delta_n) \le \sum_{j\ge 0}\Big(\int_{|x|>a} nH_j(x)\mathrm{d}x + \int_{|x|\le a} n^{1/2}H_j(x)^{1/2}\mathrm{d}x\Big).$$

By (69), we further obtain

$$\begin{aligned}\mathbb{E}(\Delta_n) &\le K_q^q n\sum_{j\ge 0}\int_{|x|>a}\frac{\rho^{jq}\mu_q^q}{(1-\rho)|x|^q}\mathrm{d}x + K_q^{q/2}n^{1/2}\sum_{j\ge 0}\int_{|x|\le a}\frac{\rho^{qj/2}\mu_q^{q/2}}{(1-\rho)^{1/2}|x|^{q/2}}\mathrm{d}x \\ &\le (1/(q-1)+2/(2-q))K_q n^{1/q}(1-\rho)^{-1/q-1}\mu_q.\end{aligned}$$

Case $q = 2$: Take $a = n^{1/2}(1-\rho)^{-1/2}\mu_2$, $b = \mu_2$, then by (67) and (68),

$$\mathbb{E}(\Delta_n) \le \sum_{j\ge 0}\Big(\int_{|x|>a} nF_j(x)\mathrm{d}x + \int_{b<|x|\le a} n^{1/2}F_j(x)^{1/2}\mathrm{d}x + \int_{|x|\le b} n^{1/2}F_j(x)^{1/2}\mathrm{d}x\Big)$$

By (69), for $r = 3/2$,

$$\begin{aligned}\mathbb{E}(\Delta_n) &\le n\sum_{j\ge 0}\int_{|x|>a}\frac{\rho^{2j}\mu_2^2}{(1-\rho)|x|^2}\mathrm{d}x + n^{1/2}\sum_{j\ge 0}\int_{b<|x|\le a}\frac{\rho^j\mu_2}{(1-\rho)^{1/2}|x|}\mathrm{d}x \\ &\quad + n^{1/2}\sum_{j\ge 0}\int_{|x|\le b}\frac{\rho^{jr/2}\mu_2^{r/2}}{(1-\rho)^{1/2}|x|^{r/2}}\mathrm{d}x \le \frac{10\sqrt{n}\mu_2}{(1-\rho)^{3/2}}\log(n(1-\rho)^{-1}).\end{aligned}$$

■

Remark 26 *Let $K_{3/2}$ and K_q be the constants defined in Lemma 19. With a careful check of the proof of Theorem 7, we can choose constants in Theorem 7 as follows:*

- $C_q = \max\{(K_{3/2}\vee K_q)^{q/2}(2/(q-2)+8), (1/(q-1)+2/(2-q))K_q, 10\}$,
- $C_{q,\gamma} = (4K_q\gamma)^q$,
- $C'_{q,\gamma} = 2^{q+5}e^q(2\vee\gamma)^2q^{-2}$.

6.4 Proofs of Theorem 8 and Proposition 9

Proof [Proof of Theorem 8] The idea of proving Theorem 8 is similar to the proof of Theorem 4. Recall the definitions of $\phi_j(g)$, $\tilde{\phi}_j(g)$, $T_n(g)$ in (32), (35) and definitions of Ω_n, U_n in (36). Then the same argument as in Theorem 4 leads to

$$\begin{aligned}\mathbb{P}(\Delta_n \ge C'_{\beta,q,\gamma}\mu_q n^{3/2-\beta} + z) &\le \mathbb{P}\Big(\max_{g\in A_n}|S_n(g) - \mathbb{E}S_n(g)| \ge C'_{\beta,q,\gamma}\mu_q n^{3/2-\beta} + z/2\Big). \\ &\le \mathbb{P}\Big(U_n \ge C'_{\beta,q,\gamma}\mu_q n^{3/2-\beta} + z/4\Big) + \mathbb{P}(\Omega_n \ge z/4).\end{aligned} \tag{70}$$

Again we shall use $T_n(g)$ to approximate $S_n(g) - \mathbb{E}S_n(g)$, and apply Fuk-Nagaev's inequality to deal with $T_n(g)$ part. By Lemma 27,

$$\mathbb{P}\Big(U_n \ge C'_{\beta,q,\gamma}\mu_q n^{3/2-\beta} + z/4\Big) \le C_{\beta,q,\gamma,1}\mu_q^q\frac{n^{1+(1-\beta)q}}{z^q} + \exp\Big(-\frac{z^2}{C_{\beta,\gamma}n^{3-2\beta}\mu_2^2}\Big), \tag{71}$$

and by Lemma 28,

$$\mathbb{P}(\Omega_n \ge \frac{z}{4}) \le C_{\beta,q,\gamma,2}\mu_q^{2q}\frac{n^{1+(1-\beta)q}[\log|A_n| + \log(n)]^q}{\tilde{c}^q(n,\beta)z^q} + 2|A_n|\exp\Big(-\frac{z^2}{C_{\beta,\gamma}n^{3-2\beta}\mu_2^2}\Big). \tag{72}$$

Combining (70), (71) and (72) with $C_{\beta,q,\gamma} = C_{\beta,q,\gamma,1} + C_{\beta,q,\gamma,2}$, the result follows. ■

Lemma 27 *Recall the definitions of $\phi_j(g)$, $T_n(g)$ in (32), (35) and U_n in (36). Under assumptions of Theorem 8, we have (71).*

Proof The proof is similar to the one of Lemma 23. We shall

(i). Bound the probability $\mathbb{P}(U_n - 2\mathbb{E}U_n \geq z/4)$.

(ii). Bound the expectation $\mathbb{E}U_n$.

Part (i): For $j < -n$, by (39),

$$|\tilde{\phi}_j(g)| \leq \sum_{i=1}^{n} |a_{i-j}|(|\epsilon_j| + \mu) \leq \gamma n(-j)^{-\beta}(|\epsilon_j| + \mu). \tag{73}$$

For $-n \leq j \leq n$, by (39),

$$|\tilde{\phi}_j(g)| \leq \sum_{i=1\vee j}^{n} |a_{i-j}|(|\epsilon_j| + \mu) \leq 2(1-\beta)^{-1}\gamma n^{1-\beta}(|\epsilon_j| + \mu). \tag{74}$$

Denote $V = \max_{g\in A_n} \sum_{j\leq n} \mathbb{E}\tilde{\phi}_j^2(g)$. Hence by (73) and (74),

$$V \leq \max_{g\in A_n} \sum_{j<-n} \mathbb{E}|\tilde{\phi}_j|^2 + \max_{g\in A_n} \sum_{j=-n}^{n} \mathbb{E}|\tilde{\phi}_j|^2 \leq c_1\mu_2^2 n^{3-2\beta}, \tag{75}$$

where $c_1 = 4\gamma^2((2\beta-1)^{-1} + 8(1-\beta)^{-2})$. Also by (73) and (74), we have

$$\sum_{j\leq n} \mathbb{E}\big(\max_{g\in A_n} |\tilde{\phi}_j|^q\big) \leq \sum_{j<-n} \mathbb{E}\big(\max_{g\in A_n} |\tilde{\phi}_j|^q\big) + \sum_{j=-n}^{n} \mathbb{E}\big(\max_{g\in A_n} |\tilde{\phi}_j|^q\big) \leq c_2 n^{1+(1-\beta)q}\mu_q^q, \tag{76}$$

where $c_2 = \gamma^q 2^q(2^{1+q}(1-\beta)^{-q} + (q\beta-1)^{-1})$.

Using the bounds (75) and (76) in the Fuk-Nagaev inequality Lemma 20, we obtain

$$\mathbb{P}\Big(U_n - 2\mathbb{E}U_n \geq z/4\Big) \leq \exp\Big(-\frac{z^2}{C_{\beta,\gamma} n^{3-2\beta}\mu_2^2}\Big) + C_{\beta,q,\gamma,1}\mu_q^q \frac{n^{1+(1-\beta)q}}{z^q}. \tag{77}$$

Part (ii): By Lemma 21 the derivatives $|g'_\infty(x)| \leq 1$, thus

$$\begin{aligned}
\max_{g\in A_n} |T_n(g)| &= \max_{g\in A_n} \Big| \int_{-\infty}^{\infty} \sum_{j\leq n} \sum_{i=1\vee j}^{n} \big(\mathbf{1}_{a_{i-j}\epsilon_j \geq x} - \mathbb{P}(a_{i-j}\epsilon_j \geq x)\big) g'_\infty(x)\mathrm{d}x \Big| \\
&\leq \int_{-\infty}^{\infty} \Big| \sum_{j\leq -n} \sum_{i=1\vee j}^{n} \big(\mathbf{1}_{a_{i-j}\epsilon_j \geq x} - \mathbb{P}(a_{i-j}\epsilon_j \geq x)\big) \Big| \mathrm{d}x \\
&\quad + \int_{-\infty}^{\infty} \Big| \sum_{-n\leq j\leq n} \sum_{i=1\vee j}^{n} \big(\mathbf{1}_{a_{i-j}\epsilon_j \geq x} - \mathbb{P}(a_{i-j}\epsilon_j \geq x)\big) \Big| \mathrm{d}x =: \mathrm{I}_1 + \mathrm{I}_2.
\end{aligned}$$

For I_1 : since ϵ_j are independent,

$$\begin{aligned}
\mathbb{E}(\mathrm{I}_1) &\leq \sum_{i=1}^{n} \int_{-\infty}^{\infty} \Big\| \sum_{j<-n} \big(\mathbf{1}_{a_{i-j}\epsilon_j \geq x} - \mathbb{P}(a_{i-j}\epsilon_j \geq x)\big) \Big\| \mathrm{d}x \\
&= \sum_{i=1}^{n} \int_{-\infty}^{\infty} \Big[\sum_{j<-n} (1 - F_\epsilon(x/a_{i-j}))F_\epsilon(x/a_{i-j}) \Big]^{1/2} \mathrm{d}x.
\end{aligned} \tag{78}$$

Denote $F^*(x) = \mathbb{P}(|\epsilon_0| \geq |x|)$, then

$$\max\big\{F_\epsilon(x) \wedge (1 - F_\epsilon(x)), F_\epsilon(-x) \wedge (1 - F_\epsilon(-x))\big\} \leq F^*(x).$$

Since $F^*(x)$ decreases in $|x|$ and $|a_{i-j}| \leq \gamma(-j)^\beta$, (78) can be further bounded by

$$\begin{aligned}
\mathbb{E}(\mathrm{I}_1) &\leq 2\sum_{i=1}^{n} \int_0^{\infty} \Big[\sum_{j<-n} F^*(x/a_k) \Big]^{1/2} \mathrm{d}x \\
&\leq 2n \int_0^{\infty} \Big[\sum_{j<-n} F^*(x\gamma^{-1}(-j)^\beta) \Big]^{1/2} \mathrm{d}x \\
&\leq 2n \int_0^{\infty} \Big[\int_n^{\infty} F^*(\gamma^{-1}xy^\beta)\mathrm{d}y \Big]^{1/2} \mathrm{d}x \\
&= 2n^{3/2-\beta}\gamma \int_0^{\infty} \Big[\int_1^{\infty} F^*(xy^\beta)\mathrm{d}y \Big]^{1/2} \mathrm{d}x,
\end{aligned} \tag{79}$$

where the last equality is due to a change of variables: $x \mapsto n^\beta x/\gamma$, $y \mapsto y/n$.

Let $r = 1 + 1/(2\beta)$. Then $1/\beta < r < 2$. Since $r < q$, we have $F^*(x) \leq |x|^{-r}\mu_q^r$, $F^*(x) \leq |x|^{-q}\mu_q^q$ and

$$\begin{aligned}
\int_0^{\infty} \Big[\int_1^{\infty} F^*(xy^\beta)\mathrm{d}y \Big]^{1/2} \mathrm{d}x \;&\leq\; \int_0^{\mu_q} \Big[\int_1^{\infty} x^{-r}y^{-r\beta}\mu_q^r \mathrm{d}y \Big]^{1/2} \mathrm{d}x \\
&+ \int_{\mu_q}^{\infty} \Big[\int_1^{\infty} x^{-q}y^{-q\beta}\mu_q^q \mathrm{d}y \Big]^{1/2} \mathrm{d}x \leq c_3\mu_q,
\end{aligned}$$

where $c_3 = 2(2-r)^{-1}(r\beta-1)^{-1} + 2(q-2)^{-1}(q\beta-1)^{-1/2}$.

For I_2: Since ϵ_j are independent, we have

$$\begin{aligned}
\mathbb{E}(\mathrm{I}_2) &= \mathbb{E}\int_{-\infty}^{\infty} \Big| \sum_{k=0}^{2n} \sum_{i=(k-n)\vee 1}^{n} \big(\mathbf{1}_{a_k\epsilon_{i-k} \geq x} - \mathbb{P}(a_k\epsilon_{i-k} \geq x)\big) \Big| \mathrm{d}x \\
&\leq \sum_{k=0}^{2n} \int_{-\infty}^{\infty} \Big(\sum_{i=(k-n)\vee 1}^{n} (1 - F_\epsilon(x/a_k))F_\epsilon(x/a_k) \Big)^{1/2} \mathrm{d}x \\
&\leq \sum_{k=0}^{2n} |a_k| \int_{-\infty}^{\infty} [nF^*(x)]^{1/2} \mathrm{d}x \leq \gamma(1-\beta)^{-1}2^{1-\beta}n^{3/2-1} \int_{-\infty}^{\infty} F^*(x)^{1/2} \mathrm{d}x,
\end{aligned}$$

where the second inequality is by a change of variable and the last inequality is by $|a_k| \le \gamma k^{-\beta}$. Note by definition of $F^*(x)$,

$$\int_{-\infty}^{\infty} F^*(x)^{1/2}\mathrm{d}x = 2\int_0^{\mu_q} 1\mathrm{d}x + 2\int_{\mu_q}^{\infty} F^*(x)^{1/2}\mathrm{d}x \le 2q/(q-2)\mu_q.$$

Combining I_1 and I_2, $\mathbb{E}U_n \le c_4\mu_q n^{3/2-\beta}$, where $c_4 = 2\gamma c_3 + 4\gamma(1-\beta)^{-1}q(q-2)^{-1}$. ■

Lemma 28 *Recall the definitions of $\phi_j(g)$, $\tilde{\phi}_j(g)$, $T_n(g)$ in (32), (35) and definition of Ω_n in (36). Under assumptions of Theorem 8, we have (72).*

Proof The argument is similar to the proof of Lemma 24, that is, we shall apply Lemma 18 to bound the tail probability. To this aim, we need to:

(i). Derive the upper bound for $\mathrm{I}_1 = \sum_{j\le n}\mathbb{P}(\max_{g\in A_n}|\phi_j(g) - \tilde{\phi}_j(g)| \ge u)$.

(ii). Bound the term $\mathrm{I}_2 = \max_{g\in A_n}\sum_{j\le n}\mathbb{E}[(\phi_j(g) - \tilde{\phi}_j(g))^2|\mathcal{F}_{j-1}]$.

Part (i): By (50), we have

$$\max_{g\in A_n}|\phi_j(g) - \tilde{\phi}_j(g)| \le \sum_{i=1\vee j}^{n}|a_{i-j}|(|\epsilon_j| + \mu)(|X_{i,j-1}| + \mathbb{E}|X_{i,j}|).$$

Since ϵ_j are independent of $X_{i,j-1}$, above together with (51) leads to

$$\|\max_{g\in A_n}|\phi_j(g) - \tilde{\phi}_j(g)|\|_q \le c_1'\sum_{i=1\vee j}^{n}(i-j)^{-2\beta+1/2}\mu_q^2 \le \begin{cases} c_1' n(-j)^{-2\beta+1/2}, & \text{if } j < -n,\\ 2c_1' h(n,\beta)\mu_q^2, & \text{if } -n \le j \le n,\end{cases}$$

where $c_1' = 4(2\beta-1)^{-1/2}\gamma^2 K_q$, $h(n,\beta) = \log(n)$ if $\beta = 3/4$; $h(n,\beta) = (4\beta-1)/(4\beta-3)$ if $\beta > 3/4$; $h(n,\beta) = 2(3-4\beta)^{-1}n^{3/2-2\beta}$ if $\beta < 3/4$. Therefore by Markov's inequality

$$\begin{aligned}\mathrm{I}_1 &\le u^{-q}\Big(\sum_{-n\le j\le n}\|\max_{g\in A_n}|\phi_j(g) - \tilde{\phi}_j(g)|\|_q^q + \sum_{j<-n}\|\max_{g\in A_n}|\phi_j(g) - \tilde{\phi}_j(g)|\|_q^q\Big)\\ &\lesssim u^{-q}n^{1+(1-\beta)q}\tilde{c}^{-q}(n,\beta)\mu_q^{2q},\end{aligned}$$

where the constant in $\lesssim$ only depends on β, q, γ.
Part (ii): By (50) we obtain

$$\max_{g\in A_n}|\phi_j(g) - \tilde{\phi}_j(g)| \le \sum_{i=1\vee j}^{n}2|a_{i-j}|(|\epsilon_j| + \mu) \le \begin{cases} 2\gamma n(-j)^{-\beta}(|\epsilon_j| + \mu), & \text{if } j < -n,\\ 2\gamma n^{1-\beta}(|\epsilon_j| + \mu), & \text{if } -n \le j \le n.\end{cases}$$

Since ϵ_j is independent of $\mathcal{F}_{j-1}$, $\mathbb{E}(|\epsilon_j|^2|\mathcal{F}_{j-1}) = \mu_2^2$. Hence

$$\begin{aligned}\mathrm{I}_2 \le& \sum_{-n\le j\le n}\mathbb{E}\big[\max_{g\in A_n}(\phi_j(g) - \tilde{\phi}_j(g))^2|\mathcal{F}_{j-1}\big] + \sum_{j<-n}\mathbb{E}\big[\max_{g\in A_n}(\phi_j(g) - \tilde{\phi}_j(g))^2|\mathcal{F}_{j-1}\big]\\ \le& 16\gamma^2\sum_{-n\le j\le n}n^{2(1-\beta)}\mu_2^2 + 16\gamma^2\sum_{j<-n}n^2(-j)^{-2\beta}\mu_2^2 \le c_2' n^{3-2\beta}\mu_2^2,\end{aligned}$$

where $c_2' = 32\gamma^2\beta(2\beta-1)^{-1}$.

Combining two parts and applying them to Lemma 18, we have

$$\mathbb{P}\Big(\Omega_n \ge \frac{z}{4}\Big) \lesssim \frac{n^{1+(1-\beta)q}\tilde{c}^{-q}(n,\beta)\mu_q^{2q}}{u^q} + 2|A_n|\exp\Big(-\frac{z^2}{C_{\beta,\gamma}n^{3-2\beta}\mu_2^2}\Big) + 2|A_n|\exp\Big(-\frac{z^2}{2zu}\Big),$$

where the constant in $\lesssim$ only depends on β, q, γ. Let $R_n = 2|A_n|\tilde{c}^q(n,\beta)z^q/n^{1+(1-\beta)q}$ and $u = z/(4\log(R_n))$. Notice $\log(R_n) \lesssim \log|A_n| + \log(n)$, where constant in $\lesssim$ only depends on β, q, γ. Then the desired result follows. ■

Remark 29 *With a careful check of the proofs of Theorem 8, Lemmas 27 and 28, we can choose constants in Theorem 8 as follows:*

- $C_{\beta,\gamma} = 64\max\{3\gamma^2((2\beta-1)^{-1} + 8(1-\beta)^{-2}), 16\gamma^2\beta(2\beta-1)^{-1}\}$.
- $C_{\beta,q,\gamma} = \max\{C_{\beta,q,\gamma,1}, C_{\beta,q,\gamma,2}\}$, *where* $C_{\beta,q,\gamma,1} = 4^q K_q' c_2$, *with* $c_2 = \gamma^q 2^q(2^{1+q}(1-\beta)^{-q} + (q\beta-1)^{-1})$ *and* $C_{\beta,q,\gamma,2} = 1 + 8{c_1'}^q(2\beta q - q/2 - 1)^{-1} + 2^{q+3}{c_1'}^q\max^q\{1, (4\beta-1)/(4\beta-3), 2(3-4\beta)^{-1}\}$, *with* $c_1' = 4(2\beta-1)^{-1/2}\gamma^2 K_q$.
- $C'_{\beta,q,\gamma} = 2\gamma c_3 + 4\gamma(1-\beta)^{-1}q(q-2)^{-1}$, *where* $c_3 = 2(2-r)^{-1}(r\beta-1)^{-1} + 2(q-2)^{-1}(q\beta-1)^{-1/2}$ *with* $r = 1 + 1/(2\beta)$.

Here K_q (resp. K_q') is the constant given in Lemma 19 (resp. Lemma 20).

Proof [Proof of Proposition 9] Construct A_n as in the proof of Theorem 4. Recall (31) for the function g_k. Note that $g_1(X_{i,i-1}) = \mathbb{E}[g(X_i)|\mathcal{F}_{i-1}]$. By (36), we have

$$\begin{aligned}\mathbb{P}(|\Delta_n| \ge a + z) \le& \mathbb{P}\Big(\max_{g\in A_n}|S_n(g) - \mathbb{E}S_n(g)| \ge a + z/2\Big)\\ \le& \mathbb{P}\Big(\max_{g\in A_n}\Big|\sum_{i=1}^{n}\big(g_1(X_{i,i-1}) - \mathbb{E}g_1(X_{i,i-1})\big)\Big| \ge a + z/4\Big)\\ &+ \sum_{g\in A_n}\mathbb{P}\Big(\Big|\sum_{i=1}^{n}(g(X_i) - \mathbb{E}[g(X_i)|\mathcal{F}_{i-1}])\Big| \ge z/4\Big) =: \mathrm{I}_1 + \mathrm{I}_2,\end{aligned}$$

where $a = C_q a_* \mu_q c(n,q)$ and $a = C'_{\beta,q,\gamma}\mu_q n^{3/2-\beta}$ for Propositions 5 and 9, respectively.

Since $|g| \le 1$ and $g(X_i) - \mathbb{E}[g(X_i)|\mathcal{F}_{i-1}]$, $1 \le i \le n$, are martingale differences, by Azuma's inequality, $\mathrm{I}_2 \le 2|A_n|\exp\{-z^2/(64n)\}$. For I_1, notice

$$g_1(x) = \int_{-\infty}^{\infty}g(x+y)f_\epsilon(y)\mathrm{d}y = \int_{-\infty}^{\infty}g(y)f_\epsilon(y-x)\mathrm{d}y.$$

By Assumption (A'), $\sup_{g\in\mathcal{A}}|g_1|_\infty$, $\sup_{g\in\mathcal{A}}|g_1'|_\infty$ and $\sup_{g\in\mathcal{A}}|g_1''|_\infty$ are all bounded by 1. Thus in the I_1 part, the function g_1 satisfies Assumption (A) and can be dealt with by Theorem 4 and Theorem 8 for Propositions 5 and 9 respectively. Combining I_1 and I_2, we complete the proof. ■

6.5 Proof of Theorem 11

Proof [Proof of Theorem 11] We shall apply the argument in the proof of Theorem 7. Recall (56) for D_k, I_1 and I_2. Case (a) follows from the following three claims:
(a.i) $\mathrm{I}_1 \le e^{-C_2 z^2/n}$, (a.ii) $\mathrm{I}_2 \le e^{-C_2 z^2/n}$, (a.iii) $\mathbb{E}\Delta_n \le C_1\sqrt{n}$,
while Case (b) follows from the following three:
(b.i) $\mathrm{I}_1 \le e^{-C_4 z^2 n^{-(3-2\beta)}}$, (b.ii) $\mathrm{I}_2 \le e^{-C_4 z^2 n^{-(3-2\beta)}}$, (b.iii) $\mathbb{E}\Delta_n \le C_3 n^{3/2-\beta}$.
Part (a.i) and (b.i): By (58), for $k \le -n$,

$$|D_k| \le \sum_{i=1\vee k}^{n} |a_{i-k}|(|\epsilon_k|+\mu) \le \gamma n(-k)^{-\beta}(|\epsilon_k|+\mu). \tag{80}$$

Let $h^* = c_0/(2\gamma)$. By the same argument in (63) and (64), for $0 < h \le h^*$,

$$\mathbb{E}(e^{D_k h}|\mathcal{F}_{k-1}) \le 1 + \mathbb{E}\Big[\frac{e^{|D_k|h^*} - |D_k|h^* - 1}{h^{*2}}\Big|\mathcal{F}_{k-1}\Big]h^2. \tag{81}$$

Denote $\theta = n(-k)^{-\beta}/2$. Note that for any fixed $x > 0$, $e^{tx} - tx - 1$ is increasing on $t \in (0,\infty)$. Applying the upper bound for $|D_k|$ in (80), we have

$$\begin{aligned}
&\mathbb{E}(e^{|D_k|h^*} - |D_k|h^* - 1|\mathcal{F}_{k-1}) \le \mathbb{E}\big[e^{c_0\theta(|\epsilon_k|+\mu)} - c_0\theta(|\epsilon_k|+\mu) - 1\big] \\
&= \mathbb{E}\Big[\int_0^\infty \frac{\mathrm{d}}{\mathrm{d}x}(e^{\theta x} - \theta x - 1)\cdot \mathbf{1}_{\{c_0(|\epsilon_k|+\mu)\ge x\}}\mathrm{d}x\Big] = \int_0^\infty (\theta e^{\theta x} - \theta)\mathbb{P}(c_0(|\epsilon_k|+\mu) \ge x)\mathrm{d}x,
\end{aligned}$$

where the last equality is by Fubini's theorem. Note that $\mathbb{P}(c_0(|\epsilon_k|+\mu) \ge x) \le c_1 e^{-x}$, where $c_1 = e^{c_0\mu}\mu_e$. Then we further have

$$\mathbb{E}(e^{|D_k|h^*} - |D_k|h^* - 1|\mathcal{F}_{k-1}) \le \int_0^\infty c_1 e^{-x}(\theta e^{\theta x} - \theta)\mathrm{d}x = c_1\theta^2/(1-\theta) \le 2c_1\theta^2. \tag{82}$$

where the last inequality is due to $\theta \le 1/2$. Hence by (81) and (82) we have for any $h \le h^*$,

$$\mathbb{E}(e^{D_k h}|\mathcal{F}_{k-1}) \le 1 + c_2 n^2(-k)^{-2\beta}h^2 \le e^{c_2 n^2(-k)^{-2\beta}h^2}. \tag{83}$$

where $c_2 = 2c_1\gamma^2/c_0^2$ and the last inequality is due to $1 + x \le e^x$. By Markov's inequality

$$\mathrm{I}_1 \le e^{-zh/2}\mathbb{E}\Big(e^{\sum_{k\le -n} D_k h}\Big) \le e^{-zh/2}\mathbb{E}\Big(e^{\sum_{k\le -n-1} D_k h}\mathbb{E}(e^{D_{-n}h}|\mathcal{F}_{-n-1})\Big).$$

Hence recursively applying (83), we obtain

$$\mathrm{I}_1 \le \exp\Big(-zh/2 + c_2 n^2 \sum_{k\le -n}(-k)^{-2\beta}h^2\Big) \le \exp\Big(-zh/2 + c_2 c_3(2\beta-1)^{-1}n^{3-2\beta}h^2\Big),$$

where $c_3 = \max\{(2\beta-1)/(4c_2h^*), 1\}$. Take $h = (2\beta-1)(4c_2c_3)^{-1}z/n$ for (a.i) and $h = (2\beta-1)(4c_2c_3)^{-1}z/n^{3-2\beta}$ for (b.i) respectively, then $h \le h^*$ and we have $\mathrm{I}_1 \le e^{-C_{21}z^2/n}$ for (a.i) and $\mathrm{I}_1 \le e^{-C_{21}z^2/n^{3-2\beta}}$ for (b.i), where $C_{21} = (2\beta-1)/(16c_2c_3)$.

Part (a.ii): By (58), for $-n < k \le n$,

$$|D_k| \le \sum_{i=1\vee k}^{n} |a_{i-k}|(|\epsilon_k|+\mu) \le 2\beta(\beta-1)^{-1}\gamma(|\epsilon_k|+\mu). \tag{84}$$

Let $c_4 = 2\beta(\beta-1)^{-1}\gamma$ and $h^* = c_0 c_4^{-1}$. By the same argument as (83) in Part (a.i), we have for any $h \le h^*$,

$$\mathbb{E}(e^{D_k h}|\mathcal{F}_{k-1}) \le e^{c_5 h^2}, \tag{85}$$

where $c_5 = c_1 c_4^2/(2c_0^2)$. Similarly to Part (a.i), by Markov's inquality and recursively applying (85),

$$\mathrm{I}_2 \le e^{-zh/2}\mathbb{E}\Big(e^{\sum_{-n<k\le n} D_k h}\Big) \le \exp\Big(-zh/2 + 2c_5c_6nh^2\Big),$$

where $c_6 = \max\{c_4/(8c_0c_5), 1\}$. Let $h = (8c_5c_6)^{-1}z/n$. Then $h \le h^*$ and we have $\mathrm{I}_2 \le e^{-C_{22}z^2/n}$, where $C_{22} = (32c_5c_6)^{-1}$.
Part (b.ii): By (58), for $-n < k \le n$,

$$|D_k| \le (1-\beta)^{-1}\gamma n^{1-\beta}(|\epsilon_k|+\mu). \tag{86}$$

Let $c_7 = (1-\beta)^{-1}\gamma$, $h^* = c_0 c_7^{-1} n^{-1+\beta}$ and $c_8 = 2\max\{c_7/(8c_0), c_1c_7^2/(2c_0^2)\}$. By the same argument as in Part (a.ii) with the bound in (84) replaced by (86), we have for any $h \le h^*$,

$$\mathrm{I}_2 \le e^{-zh/2}\mathbb{E}\Big(e^{\sum_{-n<k\le n} D_k h}\Big) \le \exp\Big(-zh/2 + c_8 n^{3-2\beta}h^2\Big).$$

Take $h = (4c_8)^{-1}zn^{-(3-2\beta)}$, then $h \le h^*$ and we have $\mathrm{I}_2 \le e^{-C_{42}z^2 n^{-(3-2\beta)}}$.

Part (a.iii) and (b.iii): Applying Theorem 1 in Wu (2003) with $p = 0$, $k = 1$ and $\gamma = 2$ therein, we have $\mathbb{E}(\Delta_n) = C_1 n^{1/2}$ (resp. $\mathbb{E}(\Delta_n) = C_3 n^{3/2-\beta}$) for SRD (resp. LRD) processes, where the constants C_1 and C_3 only depend on $\beta, \gamma, f_*, \mu_e, c_0$. ■

Remark 30 *Based on the proof of Theorem 11, the constants can take the following values:* $C_2 = \max(C_{21}, C_{22})$, $C_4 = \max(C_{21}, C_{42})$, *where* $C_{21} = (2\beta-1)/(16c_2c_3)$, $C_{22} = (32c_5c_6)^{-1}$ *and* $C_{42} = (16c_8)^{-1}$, *with* $c_1 = e^{c_0\mu}\mu_e$, $c_2 = 2c_1\gamma^2/c_0^2$, $c_3 = \max\{(2\beta-1)\gamma/(2c_0c_2), 1\}$, $c_4 = 2\beta(\beta-1)^{-1}\gamma$, $c_5 = c_1c_4^2/(2c_0^2)$, $c_6 = \max\{c_4/(8c_0c_5), 1\}$, $c_7 = (1-\beta)^{-1}\gamma$ *and* $c_8 = 2\max\{c_7/(8c_0), c_1c_7^2/(2c_0^2)\}$. *Constants* C_1 *and* C_3 *only depend on* $\beta, \gamma, f_*, \mu_e, c_0$.

6.6 Proofs of Theorem 12 and Proposition 13

Proof [Proof of Theorem 12] Since F_ϵ is the c.d.f of ϵ_i and $a_0 = 1$, $\mathbb{E}(\mathbf{1}_{X_i\le t}|\mathcal{F}_{i-1}) = F_\epsilon(t - X_{i,i-1})$. The summation $S_n(t)$ can be decomposed into two parts:

$$S_n(t) = \sum_{i=1}^n [\mathbf{1}_{X_i\le t} - \mathbb{E}(\mathbf{1}_{X_i\le t}|\mathcal{F}_{i-1})] + \sum_{i=1}^n [F_\epsilon(t - X_{i,i-1}) - F(t)] =: Q_n(t) + R_n(t). \tag{87}$$

Note that summands of $Q_n(t)$ are martingale differences. We shall derive bounds for

(i). $\mathbb{P}(\sup_{t\in\mathbb{R}}|Q_n(t)|/f_*\geq z/2)$.

(ii). $\mathbb{P}\big(\sup_{t\in\mathbb{R}}|R_n(t)|/f_*\geq C_0a_*\mu_q c(n,q)+z/2\big)$, for SRD case;
$\mathbb{P}(\sup_{t\in\mathbb{R}}|R_n(t)|/f_*\geq C_0'\mu_q n^{3/2-\beta}+z/2)$, for LRD case.

We shall apply Azuma's inequality on $Q_n(t)$ since it is the sum of martingale differences. For $R_n(t)$, since F_ϵ is smooth, we apply Theorems 4 and 8 for SRD and LRD cases, respectively.
Part (i): Let $M=2\mu_q n^{2\beta}$, $H(t)=\sum_{i=1}^n\mathbf{1}_{X_i\leq t}$ and $\tilde{H}(t)=\sum_{i=1}^n F_\epsilon(t-X_{i,i-1})$. Then $Q_n(t)=H(t)-\tilde{H}(t)$ and

$$\mathbb{P}\Big(\sup_{t\in\mathbb{R}}|Q_n(t)|/f_*\geq z/2\Big)\leq \mathrm{I}_1+\mathrm{I}_2,\ \text{where}$$

$$\mathrm{I}_1=\mathbb{P}\Big(\sup_{t\in\mathbb{R}}|H(t)-\tilde{H}(t)|/f_*\geq z/2,\ \max_{i\leq n}|X_{i,i-1}|\leq M\Big),\ \mathrm{I}_2=\sum_{i=1}^n\mathbb{P}\Big(|X_{i,i-1}|\geq M\Big).$$

For I_1, let $t_k=-2M+\delta k,\ k=0,\ldots,\lceil 4M/\delta\rceil$, where $\delta=z/(4n)$. Since $|F_\epsilon'|\leq f_*$, under this construction, $|\tilde{H}(t_k)-\tilde{H}(t_{k+1})|/f_*\leq z/4$.

Moreover, since $n\mathbb{P}(|\epsilon_0|\geq M)\leq n^{1-2q\beta}\leq z/4$, $\tilde{H}(t_0)$ and $1-\tilde{H}(t_{\lceil 4M/\delta\rceil})$ are less than $z/4$ on the set $\{\max_{i\leq n}|X_{i,i-1}|\leq M\}$.

Since $H(t)$ and $\tilde{H}(t)$ are both non-decreasing, for $s_1\leq s_2$ and $t\in[s_1,s_2]$, we have

$$|H(t)-\tilde{H}(t)|\leq|\tilde{H}(s_1)-\tilde{H}(s_2)|+\max\big\{|H(s_1)-\tilde{H}(s_1)|,|H(s_2)-\tilde{H}(s_2)|\big\}.$$

Consequently,

$$\mathrm{I}_1\leq\sum_{k=0}^{\lceil 4M/\delta\rceil}\mathbb{P}\left(|H(t_k)-\tilde{H}(t_k)|/f_*>z/4\right).\tag{88}$$

For any $t\in\mathbb{R}$, since the martingale differences $\mathbf{1}_{X_i\leq t}-\mathbb{E}(\mathbf{1}_{X_i\leq t}|\mathcal{F}_{i-1}),i=1,\ldots,n$, are bounded in absolute value by 1, by Azuma's inequality,

$$\mathbb{P}\left(|H(t)-\tilde{H}(t)|>z\right)\leq 2\exp(-z^2/2n).\tag{89}$$

With (88) and (89), we obtain

$$\mathrm{I}_1\leq(64\mu_q)z^{-1}n^{2\beta+1}\exp(-z^2/(32n)).$$

For I_2, by (51) and Markov's inequality,

$$\mathrm{I}_2\leq M^{-q}\sum_{i=1}^n\|X_{i,i-1}\|_q^q\leq c_1n^{1-2q\beta},\tag{90}$$

where $c_1=(K_q\gamma/2)^q(\beta q'-1)^{-q/q'}$. Combining I_1 and I_2 we complete the proof for this part.

Part (ii): Let $M=c_2\mu_q n^{2\beta}$, where $c_2=2K_q\gamma q'\beta$, then for any $\tau>0$,

$$\begin{aligned}&\mathbb{P}(\sup_{t\in\mathbb{R}}|R_n(t)|/f_*\geq z/2+\tau)\\ \leq&\mathbb{P}\Big(\sup_{|t|\leq 2M}|R_n(t)|/f_*\geq z/4+\tau\Big)+\sum_{i=1}^n\mathbb{P}\big(|X_{i,i-1}|\geq M\big)\\ +&\mathbb{P}\Big(\sup_{|t|\geq 2M}|R_n(t)|/f_*\geq z/4,\ \max_{i\leq n}|X_{i,i-1}|\leq M\Big)=:\mathrm{I}_1'+\mathrm{I}_2'+\mathrm{I}_3'.\end{aligned}\tag{91}$$

For I_1', let

$$A_n=\{-2M+\delta k|\delta=z/(8n),k=0,1,\ldots,\lceil 4M/\delta\rceil\}.\tag{92}$$

Then $\sup_{|t|\leq 2M}\min_{s\in A_n}(|F_\epsilon(t-\cdot)-F_\epsilon(s-\cdot)|_\infty+|F(t)-F(s)|)/f_*\leq z/(4n)$, and the cardinal number $|A_n|\leq(32c_2\mu_q)n^{2\beta+1}/z$. Hence under short- (resp. long-) range dependence, take $\tau=C_qa_*\mu_qc(n,q)$ (resp. $\tau=C'_{\beta,q\gamma}\mu_qn^{3/2-\beta}$), then I_1' can be bounded by Theorem 4 (resp. Theorem 8), that is, for SRD case,

$$\mathrm{I}_1'\leq 2^{2q\beta}C_{\beta,q,\gamma}\mu_q^q\frac{n}{z^{q\beta}}+3\exp\Big(-\frac{z^2}{16C_{\beta,\gamma}\mu_{q'}^{q'}n}+\log(|A_n|)\Big)+2\exp\Big(-\frac{z^v}{2^{3+2v}\mu_q^{v'}}+\log(|A_n|)\Big),$$

and for LRD case,

$$\mathrm{I}_1'\leq 2^{2q}C_{\beta,q,\gamma}(\mu_q^{2q}\vee\mu_q^q)\frac{n^{1+(1-\beta)q}}{z^q}\Big(1+\frac{[\log(|A_n|)+\log(n)]^q}{\tilde{c}^q(n,\beta)}\Big)+3\exp\Big(-\frac{z^2}{16C_{\beta,\gamma}n^{3-2\beta}\mu_2^2}+\log(|A_n|)\Big),$$

where $C_{\beta,q,\gamma}$ and $C_{\beta,\gamma}$ take the same values as in Theorems 4 and 8, respectively.

For I_2', by (90) we have $\mathrm{I}_2'\leq n^{1-2q\beta}$.

For I_3', if $|X_{i,i-1}|\leq M$ and $t\leq -2M$, then $F_\epsilon(t-X_{i,i-1})\leq F_\epsilon(-M)\leq M^{-q}\mu_q^q$ and $F(t)\leq(2M)^{-q}\mathbb{E}|X_0|^q$. By a similar argument for $t\geq 2M$, we obtain $R_n(x)<z/4$ for $|X_{i,i-1}|\leq M$ and $|t|\geq 2M$, that is $\mathrm{I}_3'=0$. ■

Remark 31 *Recall Lemma 19 for K_q. We can choose constants in Theorem 12 as follows:*

SRD $C_0=C_q$, $C_1=(K_q\gamma/2)^q(\beta q'-1)^{-q/q'}+1+2^{q\beta}C_{\beta,q,\gamma}$, $C_2=(16C_{\beta,\gamma}\vee 32)^{-1}$, $C_3=64K_q\gamma q'\beta(2\beta+1)$, *where $C_{\beta,q,\gamma}$, $C_{\beta,\gamma}$ and C_q take same values as those in Theorem 4.*

LRD $C_0'=C'_{\beta,q,\gamma}$, $C_1'=(K_q\gamma/2)^q(\beta q'-1)^{-q/q'}+1+2^{2q}C_{\beta,q,\gamma}c_0$, $C_2'=(16C_{\beta,\gamma}\vee 32)^{-1}$, $C_3'=64K_q\gamma q'\beta(2\beta+1)$, *where $C'_{\beta,q,\gamma}$, $C_{\beta,q,\gamma}$, $C_{\beta,\gamma}$ take same values as those in Theorem 8,* $c_0=1+\max_{n\geq 1}\log^q(c_0'n^{2\beta+1})\tilde{c}^{-q}(n,\beta)$, *with* $c_0'=64K_q\gamma q'\beta$. *Since* $\tilde{c}(n,\beta)=n^\alpha$ *some* $\alpha>0$ *and* $\log(n)/n^\alpha\to 0$, *c_0' is a finite constant.*

The following lemma is a variant of the Fuk-Nagaev inequality which will be used in the proof of Proposition 13.

Lemma 32 *Let $X_i = (X_{i1}, \ldots, X_{ip})^\mathsf{T}$, $i \in \mathbb{Z}$, be independent mean 0 random vectors in $\mathbb{R}^p$ and $S_{nj} = \sum_{i\le n} X_{ij}$. Assume there exist constants $s, r, c > 0$ such that*

$$\sum_{i\le n} \mathbb{P}\big(\max_{1\le j\le p} |X_{ij}| \ge y\big) \le cn/(y^s \log^r(y)), \qquad \text{for all } y > e.$$

Let $\sigma_n^2 = \max_{1\le j\le p} \sum_{i\le n} \mathbb{E}(X_{ij}^2)$. Then for any $z \ge c' n^{1/2}$, where $c' > 0$,

$$\mathbb{P}\Big(\max_{1\le j\le p} |S_{nj}| \ge 2\mathbb{E}\big[\max_{1\le j\le p} |S_{nj}|\big] + z\Big) \le C_1 e^{-z^2/(3\sigma_n^2)} + C_2 n/(z^s \log^r(z)),$$

where C_1, C_2 are positive constants that only depend on c, c', s and r.

Proof We shall apply the argument in Theorem 3.1 of Einmahl and Li (2008) with $(B, \|\cdot\|) = (\mathbb{R}^p, |\cdot|_\infty)$, $\eta = \delta = 1$ and $\beta(y) = \beta_{sr}(y) = M/(y^s \log^r(y))$. Notice Λ_n^2 in Theorem 3.1 of Einmahl and Li (2008) is bounded by σ^2 in our settings (cf. proof of Lemma A.2 in Chernozhukov et al. (2017)). ■

Proof [Proof of Proposition 13] Recall the proof of Theorem 12 for I_1, I_2, I_1'-I_3' and A_n in (92). For $z \ge c\sqrt{n}\log^\alpha(n)$, where $\alpha > 1/2$, all terms except I_1' are of order $o\big(nz^{-q\beta}\log^{-r_0}(z)\big)$. Hence we only need to show that $\mathrm{I}_1' \lesssim z^{-q\beta}\log^{-r_0}(z) n\mu_q^q$ for $\tau = C_q a_* \mu_q \sqrt{n}$. Let

$$\phi_j(t) = \sum_{i=1}^n P_j F_\epsilon(t - X_{i,i-1}) = \sum_{i=1\vee(j+1)}^n \big(F_{i-j}(t - X_{i,j}) - F_{i-j+1}(t - X_{i,j-1})\big),$$

Then $R_n(t) = \sum_{j\le n-1} \phi_j(t)$. Define

$$\tilde\phi_j(t) = \sum_{i=1\vee(j+1)}^n \big(F(t - a_{i-j}\epsilon_j) - \mathbb{E}F(t - a_{i-j}\epsilon_j)\big) \text{ and } \tilde R_n(t) = \sum_{j\le n-1} \tilde\phi_j(t).$$

By the same argument for I_1' in the proof of Theorem 12, we have

$$\mathrm{I}_1' \le \mathbb{P}\Big(\max_{t\in A_n} |R_n(t)|/f_* \ge z/4 + C_q a_* \mu_q \sqrt{n}\Big).$$

The idea is similar to the proof of Theorem 4, that is, we shall show:

(i). $R_n(t)$ can be approximated by $\tilde R_n(t)$, specifically,

$$\mathbb{P}\Big(\max_{t\in A_n} |R_n(t) - \tilde R_n(t)|/f_* \ge z/8\Big) = o\big(nz^{-q\beta}\log^{-r_0}(z)\big). \tag{93}$$

(ii). The tail probability of $\tilde R_n(t)$,

$$\mathbb{P}\big(\max_{t\in A_n} |\tilde R_n(t)|/f_* \ge C_q a_* \mu_q \sqrt{n} + z/8\big) \lesssim \frac{\mu_q^q n}{z^{q\beta}\log^{r_0}(z)}.$$

Part (i): Note that $\log(|A_n|) \lesssim \log(n)$ (actually $\log(|A_n|) \asymp \log(n)$) and that F_ϵ/f_*, f_ϵ/f_*, f_ϵ'/f_* are all bounded in absolute value by 1. By the same argument as in the proof of Lemma 24, using $u = z/\log^{3/2}(z)$ in inequality (55), we obtain (93).

Part (ii): Denote $V = \max_{t\in A_n} \sum_{j\le n-1} \mathbb{E}[\tilde\phi_j^2(t)]$, then by (39), we have $V \lesssim n$, where the constant in $\lesssim$ only depends on β, q, γ. For $j \le -n$, by (40) and $f' \le f_*$, we have

$$\sum_{j\le -n} \mathbb{P}(\max_{t\in A_n} |\tilde\phi_j(t)|/f_* \ge z) \le \sum_{j\le -n} \mathbb{P}(\gamma n(-j)^{-\beta}(|\epsilon_j| + \mu) \ge z) \lesssim \mu_q^q n/(z^{q\beta}\log^{r_0}(z)),$$

where the constant in $\lesssim$ only depends on β, q, γ, r_0, L. For $-n < j \le n$, by (41),

$$\sum_{-n<j\le n} \mathbb{P}(\max_{t\in A_n} |\tilde\phi_j(t)|/f_* \ge z) \le \sum_{-n<j\le n} \mathbb{P}(c_\beta(|\epsilon_j| + \mu)^{1/\beta} \ge z) \lesssim \mu_q^q n/(z^{q\beta}\log^{r_0}(z)),$$

where the constant in $\lesssim$ only depends on β, q, γ, r_0, L. By Lemma 32 we have

$$\mathbb{P}\Big(\max_{t\in A_n} \big|\tilde R_n(t)\big|/f_* - 2\mathbb{E}\Big[\max_{t\in A_n} |\tilde R_n(t)|\Big]/f_* \ge z\Big) \lesssim e^{-z^2/(3V)} + \frac{\mu_q^q n}{z^{q\beta}\log^{r_0}(z)} \lesssim \frac{\mu_q^q n}{z^{q\beta}\log^{r_0}(z)}.$$

By (49), we have $\mathbb{E}[\max_{t\in A_n} |\tilde R_n(t)|] \le C_q a_* \mu_q \sqrt{n}$, which implies the desired result. ■

6.7 Proof of Theorem 14

Proof [Proof of Theorem 14] Define

$$W_n(t) = \sum_{j=0}^{n-1} \varphi_j(t), \text{ where } \varphi_j(t) = \sum_{k\ge 1} \big[F(t - k^{-\beta}\epsilon_j) - \mathbb{E}F(t - k^{-\beta}\epsilon_j)\big]. \tag{94}$$

We claim that:

(i). $S_n(t)$ can be approximated by $W_n(t)$, specifically for $\theta_0 = (2\alpha - 1)/4$,

$$\mathbb{P}\Big(|S_n(t) - W_n(t)| \ge z/\log^{\theta_0}(z)\Big) = o\big(nz^{-q\beta}/\log^{r_0}(z)\big). \tag{95}$$

(ii). The tail distribution of $\varphi_0(t)$ satisfies

$$\mathbb{P}(\varphi_0(t) > z) \sim \frac{C_1}{z^{q\beta}\log^{r_0}(z)} \text{ and } \mathbb{P}(\varphi_0(t) < -z) \sim \frac{C_2}{z^{q\beta}\log^{r_0}(z)}, \tag{96}$$

where C_1, C_2 only depend on q, β, r_0, t, F.

Proofs of (95) and (96) will be given in Lemmas 33 and 34, respectively. By (95),

$$\begin{aligned}\mathbb{P}(S_n(t) > z) \ge& \mathbb{P}(W_n(t) \ge z + z/\log^{\theta_0}(z)) - \mathbb{P}(|S_n(t) - W_n(t)| \ge z/\log^{\theta_0}(z)),\\ =& \mathbb{P}(W_n(t) \ge z + z/\log^{\theta_0}(z)) + o\big(nz^{-q\beta}/\log^{r_0}(z)\big),\end{aligned} \tag{97}$$

and similarly

$$\mathbb{P}(S_n(t) > z) \le \mathbb{P}(W_n(t) \ge z - z/\log^{\theta_0}(z)) + o\big(nz^{-q\beta}/\log^{r_0}(z)\big). \tag{98}$$

Since φ_j has a regularly varying tail (96), by Theorem 1.9 in Nagaev (1979),

$$\sup_{w \ge \sqrt{n}\log^{\alpha}(n)} \Big|\frac{\mathbb{P}(W_n(t) \ge w)}{n\mathbb{P}(\varphi_0(t) \ge w)} - 1\Big| \to 0, \text{ as } n \to \infty.$$

Hence we have $\mathbb{P}(S_n(t) \ge z) \sim C_1 n z^{-q\beta}\log^{-r_0}(z)$ by (96), (97) and (98) in view of

$$\mathbb{P}(W_n(t) \ge z + z/\log^{\theta_0}(z)) \sim C_1 \frac{n}{z^{q\beta}\log^{r_0}(z)} \sim \mathbb{P}(W_n(t) \ge z - z/\log^{\theta_0}(z)).$$

Similarly we can derive $\mathbb{P}(S_n(t) \le -z) \sim C_2 n z^{-q\beta}\log^{-r_0}(z)$. ■

Lemma 33 *Recall definitions of $S_n(t)$, $W_n(t)$ in (94). Under assumptions of Theorem 14, we have for $\theta_0 = (2\alpha-1)/4$, (95) holds.*

Proof Recall (87) for $Q_n(t)$ and $R_n(t)$. Let

$$\tilde{W}_n(t) = \sum_{j\le n-1}\sum_{i=1\vee(j+1)}^{n} [F(t-(i-j)^{-\beta}\epsilon_j) - \mathbb{E}F(t-(i-j)^{-\beta}\epsilon_j)].$$

Then

$$\begin{aligned}
&\mathbb{P}\big(|S_n(t) - W_n(t)| \ge z\log^{-\theta_0}(z)\big)\\
\le&\mathbb{P}\big(|Q_n(t)| \ge z\log^{-\theta_0}(z)/3\big) + \mathbb{P}\big(|R_n(t) - \tilde{W}_n(t)| \ge z\log^{-\theta_0}(z)/3\big)\\
+&\mathbb{P}\big(|W_n(t) - \tilde{W}_n(t)| \ge z\log^{-\theta_0}(z)/3\big) =: \mathrm{I}_1 + \mathrm{I}_2 + \mathrm{I}_3.
\end{aligned}$$

Part I_1: Since $Q_n(t)$ is the summation of martingale differences bounded in absolute value by 1, Azuma's inequality leads to

$$\mathbb{P}\big(|Q_n(t)| \ge z/\log^{\theta_0}(z)\big) \le 2\exp\Big\{-\frac{z^2}{2n\log^{2\theta_0}(z)}\Big\} = o\big(nz^{-q\beta}/\log^{r_0}(z)\big). \tag{99}$$

Part I_2: Note that

$$R_n(t) = \sum_{j\le n-1}\sum_{i=1\vee(j+1)}^{n} \big(F_{i-j}(t-X_{i,j}) - F_{i-j+1}(t-X_{i,j-1})\big).$$

Take $F_\epsilon(t-\cdot)$ as $g(\cdot)$ in Lemma 24, then $g_\infty(\cdot) = F(t-\cdot)$. By Lemma 24, but in inequality (55), take $u = z/\log^{\theta_0+2}(z)$ instead, we obtain

$$\mathbb{P}\big(|R_n(t) - \tilde{W}_n(t)| \ge z/\log^{\theta_0}(z)\big) = o\big(nz^{-q\beta}\log^{-r_0}(z)\big).$$

Part I_3: Since $\tilde{W}_n(t)$ can be rewritten as

$$\tilde{W}_n(t) = \sum_{j\le n-1}\sum_{k=1\vee(1-j)}^{n-j} [F(t-k^{-\beta}\epsilon_j) - \mathbb{E}F(t-k^{-\beta}\epsilon_j)].$$

Notice that

$$\begin{aligned}
&\tilde{W}_n(t) - W_n(t)\\
=&\sum_{j\le -1}\sum_{k=1-j}^{n-j} [F(t-k^{-\beta}\epsilon_j) - \mathbb{E}F(t-k^{-\beta}\epsilon_j)] - \sum_{j=0}^{n-1}\sum_{k\ge n-j+1} [F(t-k^{-\beta}\epsilon_j) - \mathbb{E}F(t-k^{-\beta}\epsilon_j)].
\end{aligned}$$

For $j<0$, let $\phi_j = \sum_{k=1-j}^{n-j}[F(t-k^{-\beta}\epsilon_j) - \mathbb{E}F(t-k^{-\beta}\epsilon_j)]$, then by Lemma 22,

$$\begin{aligned}
|\phi_j| &\le \sum_{k=1-j}^{n-j} \min\big\{f_* k^{-\beta}(|\epsilon_j| + \mu), 1\big\}\\
&\le f_* \min\Big\{2\beta(\beta-1)^{-1}(|\epsilon_j|+\mu)^{1/\beta}, n(-j)^{-\beta}(|\epsilon_j|+\mu)\Big\}.
\end{aligned} \tag{100}$$

Denote $V = \sum_{j\le -1}\mathbb{E}(|\phi_j|^2)$. By Corollary 1.8 in Nagaev (1979), for $x = \lfloor n/\log^{\Gamma_0}(n)\rfloor$ with $\Gamma_0 = r_0 + \theta_0 q\beta + 1/2$,

$$\begin{aligned}
\mathbb{P}\Big(\Big|\sum_{j\le -1}\phi_j\Big| \ge z/\log^{\theta_0}(z)\Big) \lesssim& \sum_{j=-x}^{-1}\frac{\log^{\theta_0 q}(z)}{z^q}\mathbb{E}(|\phi_j|^q) + \sum_{j<-x}\frac{\log^{\theta_0 q\beta}(z)}{z^{q\beta}}\mathbb{E}(|\phi_j|^{q\beta})\\
&+\exp\Big(-\frac{z^2}{\log^{2\theta_0}(z)V}\Big) = \mathrm{II}_1 + \mathrm{II}_2 + \mathrm{II}_3,
\end{aligned}$$

where the constant in $\lesssim$ only depends on q and β.

For II_1, by (100),

$$\mathrm{II}_1 \lesssim \frac{\log^{\theta_0 q\beta}(z)}{z^{q\beta}} x\mu_q^q = \frac{n}{z^{q\beta}\log^{r_0}(z)}\frac{x}{n}[\log(z)]^{\theta_0 q\beta + r_0}\mu_q^q = o\big(nz^{-q\beta}/\log^{r_0}(z)\big),$$

where the constant in $\lesssim$ only depends on μ_q, f_*, q and β.

For II_2, by (100),

$$\begin{aligned}
\mathrm{II}_2 &\lesssim \frac{\log^{\theta_0 q}(z)}{z^q}\sum_{j<-x} n^q(-j)^{q\beta}\mu_q^q\\
&\lesssim \frac{n}{z^{q\beta}\log^{r_0}(z)}\frac{[\log(z)]^{\theta_0 q + r_0} z^{q(\beta-1)} n^{q-1}}{x^{q\beta-1}}\mu_q^q = o\big(nz^{-q\beta}/\log^{r_0}(z)\big),
\end{aligned}$$

where the constants in $\lesssim$ only depend on μ_q, f_*, q and β.

For II_3, by (100),

$$V \lesssim \sum_{j<-n}(n(-j)^{-\beta})^{q'} n^{2-q'}\mu_q^{q'} + \sum_{-n\le j\le n}\mu_q^q \lesssim n\mu_q^q,$$

where the constants in $\lesssim$ only depends on f_*, q and β.

Combining II_1-II_3, we have $\mathbb{P}(|\sum_{j\le -1}\phi_j| \ge z/\log^{\theta_0}(z)) = o(nz^{-q\beta}/\log^{r_0}(z))$. A similar argument will lead to the same bound for $j \ge 0$ part, thus

$$\mathbb{P}\big(|\tilde{W}_n(t) - W_n(t)| \ge z/\log^{\theta_0}(z)\big) = o\big(nz^{-q\beta}/\log^{r_0}(z)\big).$$

Thus the lemma follows from I_1-I_3. ∎

Lemma 34 *Recall* (94) *for* $\varphi_0(t)$. *Under conditions of Theorem 14, we have*

$$\mathbb{P}\left(\varphi_0(t) > z\right) \sim \frac{C_1}{z^{q\beta}\log^{r_0}(z)} \quad \textit{and} \quad \mathbb{P}\left(\varphi_0(t) < -z\right) \sim \frac{C_2}{z^{q\beta}\log^{r_0}(z)}, \ \textit{as } z \to \infty,$$

where $C_1 = L_1^{q\beta}(t)\beta^{-r_0}$, $C_2 = L_2^{q\beta}(t)\beta^{-r_0}$, *and*

$$L_1(t) = \int_0^\infty \frac{F(t+u) - F(t)}{\beta u^{1+1/\beta}}\mathrm{d}u, \quad L_2(t) = \int_0^\infty \frac{F(t) - F(t-u)}{\beta u^{1+1/\beta}}\mathrm{d}u. \tag{101}$$

Proof Since $f_\epsilon \le 1$, by Lemma 21, f is bounded by 1. Let

$$\tilde{\varphi}_0(t) = \int_0^\infty \big[F(t - s^{-\beta}\epsilon_0) - F(t)\big]\mathrm{d}s. \tag{102}$$

Since $|F(t - s^{-\beta}\epsilon_0) - F(t)| \le \min\{1, s^{-\beta}|\epsilon_0|\}$, we have

$$|\tilde{\varphi}_0(t)| \le 1 + \int_1^\infty s^{-\beta}|\epsilon_0|\mathrm{d}s \le 1 + (\beta-1)^{-1}|\epsilon_0|.$$

Thus $\tilde{\varphi}_0(t)$ is well defined. Note that the lemma follows from the following two claims:

(i). $|\varphi_0(t) - \tilde{\varphi}_0(t)| \le f_*\mu\beta/(\beta-1) + 1$, which is bounded.

(ii). $\mathbb{P}(\tilde{\varphi}_0(t) > z) \sim C_1\log^{-r_0}(z)z^{-q\beta}$ and $\mathbb{P}(\tilde{\varphi}_0(t) < -z) \sim C_2\log^{-r_0}(z)z^{-q\beta}$ as $z \to \infty$.

Part (i): Since F is non-decreasing, for any $s \in [k-1, k]$,

$$|F(t - s^{-\beta}\epsilon_0) - F(t - k^{-\beta}\epsilon_0)| \le \mathrm{sign}(\epsilon_0)\Big\{F(t - k^{-\beta}\epsilon_0) - F(t - (k-1)^{-\beta}\epsilon_0)\Big\}.$$

Since $F(-\infty) = 0$ and $F(\infty) = 1$,

$$\mathrm{I}_1 := \sum_{k=1}^\infty \int_{k-1}^k |F(t - s^{-\beta}\epsilon_0) - F(t - k^{-\beta}\epsilon_0)|\mathrm{d}s \le 1.$$

Since f is bounded, we have

$$\mathrm{I}_2 := \sum_{k=1}^\infty |F(t) - \mathbb{E}F(t - k^{-\beta}\epsilon_0)| \le f_* \sum_{k=1}^\infty k^{-\beta}\mu \le f_*\mu\beta/(\beta-1).$$

Thus $|\varphi_0(t) - \tilde{\varphi}_0(t)| \le \mathrm{I}_1 + \mathrm{I}_2 \le f_*\mu\beta/(\beta-1) + 1$, a finite constant.

Part (ii): Let $u > 0$. Then $0 \le F(t+u) - F(t) \le \min\{f_*u, 1\}$. Hence $L_1(t)$ is bounded by $\int_0^\infty \min\{f_*u, 1\}/(\beta u^{1+1/\beta})\mathrm{d}u \le f_*\beta/(\beta-1)$. Similarly $L_2(t) \le f_*\beta/(\beta-1)$. Note that

$$\int_0^\infty \big[F(t - s^{-\beta}y) - F(t)\big]\mathrm{d}s = \begin{cases} L_1(t)|y|^{1/\beta}, & \text{if } y < 0, \\ -L_2(t)|y|^{1/\beta}, & \text{if } y > 0. \end{cases}$$

Since ϵ_0 is symmetric, by (22) and the definition of $\tilde{\varphi}_0(t)$ in (102), (ii) follows. ∎

Remark 35 *Values of* C_1 *and* C_2 *are given in Lemma 34. A careful check of the proof of Theorem 14 suggests that the constant* Γ *can be chosen as* $\Gamma = [\theta_0 q + r_0 + (q\beta-1)\Gamma']/(q\beta - q)$, *where* $\theta_0 = (2\alpha-1)/4$ *and* $\Gamma' = r_0 + \theta_0 q\beta + 1$.

6.8 Proof of Corollaries 15 and 16

Proof [Proof of Corollary 15] We shall first deal with the SRD case. Recall $\mathcal{F}_j = (\epsilon_j, \epsilon_{j-1}, \ldots)$. Write

$$M_n(x) = \sum_{j=1}^n \Big(K_b(x - X_j) - \mathbb{E}[K_b(x - X_j)|\mathcal{F}_{j-1}]\Big),$$

$$R_n(x) = \sum_{j=1}^n \Big(\mathbb{E}[K_b(x - X_j)|\mathcal{F}_{j-1}] - \mathbb{E}[K_b(x - X_j)]\Big) = n(\hat{f}_n(x) - \mathbb{E}\hat{f}_n(x)) - M_n(x).$$

Note that $M_n(x)$ is a martingale w.r.t. filter $\sigma(\mathcal{F}_n)$. Let $\tau_n = n^\beta$ and $l_* = K_* \vee f_*$. Then

$$\begin{aligned} \mathrm{I} :&= \mathbb{P}\Big(\sup_{x\in\mathbb{R}} n|\hat{f}_n(x) - \mathbb{E}\hat{f}_n(x)| \ge l_*z\Big) \\ &\le \sum_{j=1}^n \mathbb{P}(|X_j| \ge \tau_n) + \mathbb{P}\Big(\sup_{x\in\mathbb{R}} n|\hat{f}_n(x) - \mathbb{E}\hat{f}_n(x)| \ge l_*z, \max_{1\le j\le n}|X_j| < \tau_n\Big) \\ &=: \mathrm{I}_1 + \mathrm{I}_1'. \end{aligned} \tag{103}$$

Since K has support $[-1, 1]$, $K_b(x - X_j) = 0$ when $|X_j| < \tau_n$ and $|x| > \tau_n + b_n$. Hence if $\max_{j\le n}|X_j| < \tau_n$ and $|x| > \tau_n + b_n$, we have $\hat{f}_n(x) = 0$. Note that $\sup_{|x|\le\tau_n+b_n}|M_n(x)| + \sup_{|x|\le\tau_n+b_n}|R_n(x)| \ge \sup_{|x|\le\tau_n+b_n} n|\hat{f}_n(x) - \mathbb{E}\hat{f}_n(x)|$, we have

$$\begin{aligned} \mathrm{I}_1' \le &\mathbb{P}\Big(\sup_{|x|>\tau_n+b_n} n|\mathbb{E}K_b(x - X_1)| \ge l_*z/4\Big) + \mathbb{P}\Big(\sup_{|x|\le\tau_n+b_n}|M_n(x)| \ge l_*z/2\Big) \\ &+ \mathbb{P}\Big(\sup_{|x|\le\tau_n+b_n}|R_n(x)| \ge l_*z/4\Big) =: \mathrm{I}_2 + \mathrm{I}_3 + \mathrm{I}_4. \end{aligned} \tag{104}$$

Hence by (103) and (104), we have $\mathrm{I} \le \mathrm{I}_1 + \mathrm{I}_2 + \mathrm{I}_3 + \mathrm{I}_4$. For I_1-I_3 we shall bound them through some basic inequalities, for I_4, we will apply Corollary 6.

For I_1: By Lemma 19, $\mathbb{E}(|X_0|^q) \le K_q^q(\sum_{j\ge 0}|a_j|^{q'})^{q/q'}\mu_q^q$. Hence by Markov's inequality $\mathrm{I}_1 \le n\tau_n^{-q}\mathbb{E}(|X_0|^q) \lesssim nz^{-q\beta}\mu_q^q$, where the constant in $\lesssim$ only depends on q, β and γ.

For I_2: Since $|K|_\infty$ is bounded by K_* with support $[-1,1]$, we have $|\mathbb{E}K_b(x-X_1)| \le K_* b_n^{-1}\mathbb{P}(|X_1-x|\le b_n)$. When $|x| > \tau_n + b_n$, $\mathbb{P}(|X_1-x|\le b_n) \le \mathbb{P}(|X_1|\ge \tau_n)$. Hence

$$n|\mathbb{E}K_b(x-X_1)| \le nK_*b_n^{-1}\mathbb{P}(|X_1|\ge\tau_n) \le K_*b_n^{-1}n^{1-q\beta}\mathbb{E}|X_0|^q = o(K_*z),$$

in view of $z \ge c(n/b_n)^{1/2}\log^{1/2}(n)$ and $nb_n \to \infty$. Thus $\mathrm{I}_2 = 0$ for all large n.

For I_3: Let $A_n = \{-(\tau_n+b_n)+\delta_n k, k = 0,1,\ldots,\lfloor 2(\tau_n+b_n)/\delta_n + 1\rfloor\}$, where $\delta_n = zb_n^2/(8n)$. Then

$$\sup_{|x|\le\tau_n+b_n}\min_{y\in A_n}|M_n(x)-M_n(y)| \le K_*z/4,$$

and $\mathrm{I}_3 \le \sum_{x\in A_n}\mathbb{P}(|M_n(x)|\ge l_*z/4)$. Since $|K|_\infty \le K_*$ and $|f_\epsilon|_\infty \le f_*$, for $X_{j,j-1} = \sum_{k\ge 1}a_k\epsilon_{j-k}$,

$$\begin{aligned}\mathbb{E}[K_b^2(x-X_j)|\mathcal{F}_{j-1}]| &= \int_{-\infty}^{\infty} b_n^{-2}K^2(\frac{x-X_{j,j-1}-u}{b_n})f_\epsilon(u)\mathrm{d}u\\ &= \int_{-\infty}^{\infty} b_n^{-1}K^2(y)f_\epsilon(x-b_ny-X_{j,j-1})\mathrm{d}y\\ &\le K_*f_*b_n^{-1}\int_{-\infty}^{\infty}K(y)\mathrm{d}y = K_*f_*b_n^{-1}.\end{aligned}$$

Therefore for $\xi_j(x) = K_b(x-X_j) - \mathbb{E}[K_b(x-X_j)|\mathcal{F}_{j-1}]$,

$$V(x) := \sum_{j=1}^n \mathbb{E}(\xi_j(x)^2|\mathcal{F}_{j-1}) \le nK_*f_*b_n^{-1}.$$

Note $|K_b| \le K_*/b_n$, therefore by Freedman's inequality (Lemma 18), we have

$$\begin{aligned}\mathrm{I}_3 &\le \sum_{x\in A_n}\mathbb{P}\big(|M_n(x)|\ge l_*z/4\big) \le 2\sum_{x\in A_n}\exp(-\frac{z^2}{2zb_n^{-1}+2nb_n^{-1}})\\ &\le \frac{32n(\tau_n+b_n)}{zb_n^2}\exp(-\frac{z^2b_n}{4n}).\end{aligned}$$

Since $z \ge c(n/b_n)^{1/2}\log^{1/2}(n)$, for c sufficiently large $\mathrm{I}_3 = o(n/z^{q\beta})$.

For I_4: Since $\mathbb{E}[K_{b_n}(x-X_j)|\mathcal{F}_{j-1}] = \int_{\mathbb{R}}K(u)f_\epsilon(x-b_nu-X_{j,j-1})\mathrm{d}u$, we have $R_n(x) = \sum_{j=1}^n N_n(x, X_{j,j-1})$, where

$$N_n(x,y) = \int_{-\infty}^{\infty}K(u)[f_\epsilon(x-b_nu-y) - f(x-b_nu)]\mathrm{d}u. \tag{105}$$

Let function class $\mathcal{A}_n = \{N_n(x,\cdot), |x| \le \tau_n + b_n\}$, then for any function in $\mathcal{A}_n$, its up to second order derivatives are bounded by f_* and $\mathcal{N}_{\mathcal{A}_n}(f_*z/n) \le 4n(\tau_n+b_n)/z$. Therefore by Corollary 6, we have $\mathrm{I}_4 \lesssim \mu_q^q nz^{-q\beta}$, where the constant in $\lesssim$ only depends on β, q and γ. Thus (25) follows from I_1-I_4.

For the LRD case, define $M_n(x)$ and $R_n(x)$ as in the SRD case and let $\tau_n = z$. Again we have

$$\begin{aligned}&\mathbb{P}\Big(\sup_{x\in\mathbb{R}} n|\hat{f}_n(x) - \mathbb{E}\hat{f}_n(x)| \ge l_*z\Big) \le \sum_{j=1}^n \mathbb{P}(|X_j|\ge z)\\ &+\mathbb{P}\Big(\sup_{|x|>z+b_n} n|\mathbb{E}K_b(x-X_1)| \ge l_*z/4\Big) + \mathbb{P}\Big(\sup_{|x|\le z+b_n}|M_n(x)| \ge l_*z/2\Big)\\ &+\mathbb{P}\Big(\sup_{|x|\le z+b_n}|R_n(x)| \ge l_*z/4\Big) =: \mathrm{I}_1+\mathrm{I}_2+\mathrm{I}_3+\mathrm{I}_4.\end{aligned}$$

Using same argument as for SRD case with τ_n replaced by z, we obtain $\mathrm{I}_1, \mathrm{I}_2, \mathrm{I}_3 \lesssim nz^{-q}\mu_q^q$, where the constants in $\lesssim$ only depend on q, β and γ. For I_4, we still have (105). Let $\mathcal{A}_n = \{N_n(x,\cdot), |x| \le z+b_n\}$. Then $\mathcal{N}_{\mathcal{A}_n}(f_*z/n) \le 4n(z+b_n)/z$. Therefore by Corollary 10, we have $\mathrm{I}_3 \lesssim (\mu_q^{2q}\vee\mu_q^q)n^{3/2-\beta}z^{-q}$, where the constant in $\lesssim$ only depends on β, q and γ. ■

Proof [Proof of Corollary 16] Let $\mathcal{G}_i = (\epsilon_i, \epsilon_{i-1}, \ldots; \eta_i, \eta_{i-1}, \ldots)$ and $X_{i,i-1} = \sum_{j=1}^\infty a_j\epsilon_{i-j}$. Then we have $\mathbb{E}[L(X_i, Y_i, h(X_i))|\mathcal{G}_{i-1}] = Q_h(X_{i,i-1})$, where

$$Q_h(w) = \int_{-\infty}^{\infty}\mathbb{E}[L(u, H_0(u,\eta_i), h(u))]f_\epsilon(u-w)\mathrm{d}u. \tag{106}$$

Let $J_h(x) = Q_h(x) - \mathbb{E}[L(X_i, Y_i, h(X_i))]$. Write

$$n(R_n(h) - R(h)) = \sum_{i=1}^n\Big[L(X_i,Y_i,h(X_i)) - Q_h(X_{i,i-1})\Big] + \sum_{i=1}^n J_h(X_{i,i-1}) =: \mathrm{I}_1(h) + \mathrm{I}_2(h).$$

For $h, g \in \mathcal{H}$, let $D(h,g) = \sup_{x,y\in\mathbb{R}}|L(x,y,h(x)) - L(x,y,g(x))|$. Let $\mathcal{H}_n$ be the subset of $\mathcal{H}$ such that $\sup_{h_1\in\mathcal{H}}\inf_{h_2\in\mathcal{H}_n}D(h_1,h_2) \le z/(4n)$ and $|\mathcal{H}_n| \le \mathcal{N}_{\mathcal{A}}(z/(4n))$. Then for $\tau > 0$,

$$\begin{aligned}\mathbb{P}(n\Psi_n/f_* \ge z+\tau) &\le \mathbb{P}\Big(\max_{h\in\mathcal{H}_n} n|R_n(h)-R(h)|/f_* \ge z/2+\tau\Big)\\ &\le \sum_{h\in\mathcal{H}_n}\mathbb{P}(|\mathrm{I}_1(h)|/f^* \ge z/4) + \mathbb{P}\Big(\max_{h\in\mathcal{H}_n}|\mathrm{I}_2(h)|/f_* \ge z/4+\tau\Big).\end{aligned}$$

Since $0 \le L \le 1$, $f^* \ge 1$ and the summands of $\mathrm{I}_1(h)$ are bounded martingale differences with respect to $\mathcal{G}_i$, by Azuma's inequality, we have $\sum_{h\in\mathcal{H}_n}\mathbb{P}(|\mathrm{I}_1(h)| \ge z) \le 2|\mathcal{H}_n|e^{-z^2/(32n)}$. Since both $\int_{-\infty}^\infty|f_\epsilon'(x)|\mathrm{d}x$ and $\int_{-\infty}^\infty|f_\epsilon''(x)|\mathrm{d}x$ are bounded by f_*, by (106), for $h \in \mathcal{H}$, Q_h, Q_h' and Q_h'' exist and are uniformly bounded by f_* in absolute value. Thus (16) (resp. (20)) follows from applying Corollary 6 to Q_h/f_* with $\tau = C_q a_*\mu_q c(n,q)$ (resp. Corollary 10 with $\tau = C_{\beta,q,\gamma}\mu_q n^{3/2-\beta}$). ■

Acknowledgments

We thank the reviewers and the editor for very helpful suggestions, which substantially improve the paper.

References

Radosław Adamczak. A tail inequality for suprema of unbounded empirical processes with applications to Markov chains. *Electron. J. Probab.*, 13:no. 34, 1000–1034, 2008. ISSN 1083-6489.

Terrence M. Adams and Andrew B. Nobel. Uniform approximation of Vapnik-Chervonenkis classes. *Bernoulli*, 18(4):1310–1319, 2012. ISSN 1350-7265.

Alekh Agarwal and John C Duchi. The generalization ability of online algorithms for dependent data. *IEEE Transactions on Information Theory*, 59(1):573–587, 2013.

Pierre Alquier and Olivier Wintenberger. Fast rates in learning with dependent observations. *JMLR: Workshop and Conference Proceedings 1-15*, 2012.

Donald W. K. Andrews. Nonstrong mixing autoregressive processes. *J. Appl. Probab.*, 21 (4):930–934, 1984. ISSN 0021-9002.

M. A. Arcones and B. Yu. Central limit theorems for empirical and U-processes of stationary mixing sequences. *J. Theoret. Probab.*, 7(1):47–71, 1994. ISSN 0894-9840.

Ery Arias-Castro, David Mason, and Bruno Pelletier. On the estimation of the gradient lines of a density and the consistency of the mean-shift algorithm. *J. Mach. Learn. Res.*, 17:Paper No. 43, 28, 2016. ISSN 1532-4435.

Richard T. Baillie. Long memory processes and fractional integration in econometrics. *J. Econometrics*, 73(1):5–59, 1996. ISSN 0304-4076.

Jan Beran. *Statistics for long-memory processes*, volume 61 of *Monographs on Statistics and Applied Probability*. Chapman and Hall, New York, 1994. ISBN 0-412-04901-5.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. Concentration inequalities using the entropy method. *Ann. Probab.*, 31(3):1583–1614, 2003. ISSN 0091-1798.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities*. Oxford University Press, Oxford, 2013. ISBN 978-0-19-953525-5. A nonasymptotic theory of independence, With a foreword by Michel Ledoux.

Christian Brownlees, Emilien Joly, and Gábor Lugosi. Empirical risk minimization for heavy-tailed losses. *Ann. Statist.*, 43(6):2507–2536, 2015. ISSN 0090-5364.

Donald L. Burkholder. Sharp inequalities for martingales and stochastic integrals. *Astérisque*, (157-158):75–94, 1988. ISSN 0303-1179. Colloque Paul Lévy sur les Processus Stochastiques (Palaiseau, 1987).

Andrea Caponnetto and Alexander Rakhlin. Stability properties of empirical risk minimization over Donsker classes. *J. Mach. Learn. Res.*, 7:2565–2583, 2006. ISSN 1532-4435.

Olivier Catoni. Challenging the empirical mean and empirical variance: a deviation study. *Ann. Inst. Henri Poincaré Probab. Stat.*, 48(4):1148–1185, 2012. ISSN 0246-0203.

Likai Chen and Wei Biao Wu. Stability and asymptotics for autoregressive processes. *Electron. J. Stat.*, 10(2):3723–3751, 2016. ISSN 1935-7524.

Yen-Chi Chen, Christopher R. Genovese, and Larry Wasserman. A comprehensive approach to mode clustering. *Electron. J. Stat.*, 10(1):210–241, 2016. ISSN 1935-7524.

Victor Chernozhukov, Denis Chetverikov, and Kengo Kato. Testing many moment inequalities. *arXiv preprint arXiv:1312.7614, Review of Economic Studies, revise and resubmit*, 2017.

Yuan Shih Chow and Henry Teicher. *Probability theory*. Springer Texts in Statistics. Springer-Verlag, New York, third edition, 1997. ISBN 0-387-98228-0. Independence, interchangeability, martingales.

Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*, volume 31 of *Applications of Mathematics (New York)*. Springer-Verlag, New York, 1996. ISBN 0-387-94618-7.

A. Dvoretzky, J. Kiefer, and J. Wolfowitz. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *Ann. Math. Statist.*, 27: 642–669, 1956. ISSN 0003-4851.

Uwe Einmahl and Deli Li. Characterization of LIL behavior in Banach space. *Trans. Amer. Math. Soc.*, 360(12):6677–6693, 2008. ISSN 0002-9947.

Uwe Einmahl and David M. Mason. Uniform in bandwidth consistency of kernel-type function estimators. *Ann. Statist.*, 33(3):1380–1403, 2005. ISSN 0090-5364.

David A. Freedman. On tail probabilities for martingales. *Ann. Probability*, 3:100–118, 1975.

Evarist Giné and Armelle Guillou. Rates of strong uniform consistency for multivariate kernel density estimators. *Ann. Inst. H. Poincaré Probab. Statist.*, 38(6):907–921, 2002. ISSN 0246-0203. En l'honneur de J. Bretagnolle, D. Dacunha-Castelle, I. Ibragimov.

Evarist Giné and Richard Nickl. Confidence bands in density estimation. *Ann. Statist.*, 38 (2):1122–1170, 2010. ISSN 0090-5364.

Lee-Ad Gottlieb, Aryeh Kontorovich, and Robert Krauthgamer. Efficient regression in metric spaces via approximate lipschitz extension. *IEEE Transactions on Information Theory*, 2017.

Clive WJ Granger and Roselyne Joyeux. An introduction to long-memory time series models and fractional differencing. *Journal of time series analysis*, 1(1):15–29, 1980.

Zheng-Chu Guo and Lei Shi. Classification with non-i.i.d. sampling. *Math. Comput. Modelling*, 54(5-6):1347–1364, 2011. ISSN 0895-7177.

Erich Haeusler. An exact rate of convergence in the functional central limit theorem for special martingale difference arrays. *Z. Wahrsch. Verw. Gebiete*, 65(4):523–534, 1984. ISSN 0044-3719.

Hanyuan Hang and Ingo Steinwart. Fast learning from α-mixing observations. *Journal of Multivariate Analysis*, 127:184–199, 2014.

Hanyuan Hang and Ingo Steinwart. A bernstein-type inequality for some mixing processes and dynamical systems with an application to learning. *Annals of Statistics, To Appear*, 2016.

Hanyuan Hang, Ingo Steinwart, Yunlong Feng, and Johan AK Suykens. Kernel density estimation for dynamical systems. *arXiv preprint arXiv:1607.03792*, 2016.

Hwai-Chung Ho and Tailen Hsing. On the asymptotic expansion of the empirical process of long-memory moving averages. *Ann. Statist.*, 24(3):992–1024, 1996. ISSN 0090-5364.

Jonathan RM Hosking. Fractional differencing. *Biometrika*, 68(1):165–176, 1981.

Wenxin Jiang. On uniform deviations of general empirical risks with unboundedness, dependence, and high dimensionality. *J. Mach. Learn. Res.*, 10:977–996, 2009. ISSN 1532-4435.

T. Klein and E. Rio. Concentration around the mean for maxima of empirical processes. *Ann. Probab.*, 33(3):1060–1077, 2005. ISSN 0091-1798.

A Kontorovich and A Brockwell. A strong law of large numbers for strongly mixing processes. *Communications in Statistics-Theory and Methods*, 43(18):3777–3796, 2014.

Aryeh Kontorovich and Maxim Raginsky. Concentration of measure without independence: a unified approach via the martingale method. In *Convexity and Concentration*, pages 183–210. Springer, 2017.

Aryeh Kontorovich and Roi Weiss. Uniform Chernoff and Dvoretzky-Kiefer-Wolfowitz-type inequalities for Markov chains and related processes. *J. Appl. Probab.*, 51(4):1100–1113, 2014. ISSN 0021-9002. doi: 10.1239/jap/1421763330. URL `http://dx.doi.org/10.1239/jap/1421763330`.

Leonid Kontorovich and Kavita Ramanan. Concentration inequalities for dependent random variables via the martingale method. *Ann. Probab.*, 36(6):2126–2158, 2008. ISSN 0091-1798.

Michael R. Kosorok. *Introduction to empirical processes and semiparametric inference*. Springer Series in Statistics. Springer, New York, 2008. ISBN 978-0-387-74977-8. doi: 10.1007/978-0-387-74978-5. URL `http://dx.doi.org/10.1007/978-0-387-74978-5`.

Vitaly Kuznetsov and Mehryar Mohri. Generalization bounds for time series prediction with non-stationary processes. In *International Conference on Algorithmic Learning Theory*, pages 260–274. Springer, 2014.

Vitaly Kuznetsov and Mehryar Mohri. Learning theory and algorithms for forecasting non-stationary time series. In *Advances in neural information processing systems*, pages 541–549, 2015.

John Lafferty, Han Liu, and Larry Wasserman. Sparse nonparametric graphical models. *Statist. Sci.*, 27(4):519–537, 2012. ISSN 0883-4237.

Johannes Lederer and Sara van de Geer. New concentration inequalities for suprema of empirical processes. *Bernoulli*, 20(4):2020–2038, 2014. ISSN 1350-7265.

Michel Ledoux. On talagrand's deviation inequalities for product measures. *ESAIM: Probability and statistics*, 1:63–87, 1997.

Michel Ledoux. *The concentration of measure phenomenon*, volume 89 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 2001. ISBN 0-8218-2864-9.

Shlomo Levental. Uniform limit theorems for Harris recurrent Markov chains. *Probab. Theory Related Fields*, 80(1):101–118, 1988. ISSN 0178-8051.

Han Liu, Min Xu, Haijie Gu, Anupam Gupta, John Lafferty, and Larry Wasserman. Forest density estimation. *J. Mach. Learn. Res.*, 12:907–951, 2011. ISSN 1532-4435.

Benoit Mandelbrot. The variation of certain speculative prices. *The Journal of Business*, 36(4):394–419, 1963.

K. Marton. A measure concentration inequality for contracting Markov chains. *Geom. Funct. Anal.*, 6(3):556–571, 1996. ISSN 1016-443X.

Katalin Marton. Measure concentration for a class of random processes. *Probab. Theory Related Fields*, 110(3):427–439, 1998. ISSN 0178-8051.

P. Massart. The tight constant in the Dvoretzky-Kiefer-Wolfowitz inequality. *Ann. Probab.*, 18(3):1269–1283, 1990. ISSN 0091-1798.

Pascal Massart. About the constants in Talagrand's concentration inequalities for empirical processes. *Ann. Probab.*, 28(2):863–884, 2000. ISSN 0091-1798.

Pascal Massart. *Concentration inequalities and model selection*, volume 1896 of *Lecture Notes in Mathematics*. Springer, Berlin, 2007. ISBN 978-3-540-48497-4; 3-540-48497-3. Lectures from the 33rd Summer School on Probability Theory held in Saint-Flour, July 6–23, 2003, With a foreword by Jean Picard.

Dharmendra S. Modha and Elias Masry. Minimum complexity regression estimation with weakly dependent observations. *IEEE Trans. Inform. Theory*, 42(6, part 2):2133–2145, 1996. ISSN 0018-9448.

Mehryar Mohri and Afshin Rostamizadeh. Stability bounds for stationary ϕ-mixing and β-mixing processes. *J. Mach. Learn. Res.*, 11:789–814, 2010. ISSN 1532-4435.

S. V. Nagaev. Large deviations of sums of independent random variables. *Ann. Probab.*, 7 (5):745–789, 1979. ISSN 0091-1798.

Tuan D. Pham and Lanh T. Tran. Some mixing properties of time series models. *Stochastic Process. Appl.*, 19(2):297–303, 1985. ISSN 0304-4149. doi: 10.1016/0304-4149(85)90031-6. URL `http://dx.doi.org/10.1016/0304-4149(85)90031-6`.

Svetlozar Rachev and Stefan Mittnik. *Stable Paretian models in finance*. John Willey & Sons, New York, 2000.

Alessandro Rinaldo, Aarti Singh, Rebecca Nugent, and Larry Wasserman. Stability of density-based clustering. *J. Mach. Learn. Res.*, 13:905–948, 2012. ISSN 1532-4435.

Emmanuel Rio. Moment inequalities for sums of dependent random variables under projective conditions. *J. Theoret. Probab.*, 22(1):146–163, 2009. ISSN 0894-9840.

Paul-Marie Samson. Concentration of measure inequalities for Markov chains and Φ-mixing processes. *Ann. Probab.*, 28(1):416–461, 2000. ISSN 0091-1798.

Cosma Shalizi and Aryeh Kontorovich. Predictive pac learning and process decompositions. In *Advances in neural information processing systems*, pages 1619–1627, 2013.

Ingo Steinwart and Andreas Christmann. Fast learning from non-iid observations. In *Advances in Neural Information Processing Systems*, pages 1768–1776, 2009.

M. Talagrand. Sharper bounds for Gaussian and empirical processes. *Ann. Probab.*, 22(1): 28–76, 1994. ISSN 0091-1798.

Michel Talagrand. Concentration of measure and isoperimetric inequalities in product spaces. *Inst. Hautes Études Sci. Publ. Math.*, (81):73–205, 1995. ISSN 0073-8301.

Michel Talagrand. New concentration inequalities in product spaces. *Invent. Math.*, 126 (3):505–563, 1996. ISSN 0020-9910.

A. W. van der Vaart. *Asymptotic statistics*, volume 3 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge, 1998. ISBN 0-521-49603-9; 0-521-78450-6.

Aad W. van der Vaart and Jon A. Wellner. *Weak convergence and empirical processes*. Springer Series in Statistics. Springer-Verlag, New York, 1996. ISBN 0-387-94640-3. With applications to statistics.

Vladimir N. Vapnik. *Statistical learning theory*. Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley & Sons, Inc., New York, 1998. ISBN 0-471-03003-1. A Wiley-Interscience Publication.

Vladimir N. Vapnik. *The nature of statistical learning theory*. Statistics for Engineering and Information Science. Springer-Verlag, New York, second edition, 2000. ISBN 0-387-98780-0.

VN Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264, 1971.

Wei Biao Wu. Empirical processes of long-memory sequences. *Bernoulli*, 9(5):809–831, 2003. ISSN 1350-7265.

Wei Biao Wu and Jan Mielniczuk. Kernel density estimation for linear processes. *Ann. Statist.*, 30(5):1441–1459, 2002. ISSN 0090-5364.

Bin Yu. Rates of convergence for empirical processes of stationary mixing sequences. *The Annals of Probability*, pages 94–116, 1994.

Bin Zou and Luoqing Li. The performance bounds of learning machines based on exponentially strongly mixing sequences. *Comput. Math. Appl.*, 53(7):1050–1058, 2007. ISSN 0898-1221.

Bin Zou, Luoqing Li, and Zongben Xu. The generalization performance of erm algorithm with strongly mixing observations. *Machine learning*, 75(3):275–295, 2009.

Journal of Machine Learning Research 18 (2018) 1-39 Submitted 8/17; Revised 3/18; Published 7/18

A Cluster Elastic Net for Multivariate Regression

Bradley S. Price BRAD.PRICE@MAIL.WVU.EDU
College of Business and Economics
West Virginia University
Morgantown, WV 26505, USA

Ben Sherwood BEN.SHERWOOD@KU.EDU
School of Business
University of Kansas
Lawrence, KS 66045, USA

Editor: Sara van de Geer

Abstract

We propose a method for simultaneously estimating regression coefficients and clustering response variables in a multivariate regression model, to increase prediction accuracy and give insights into the relationship between response variables. The estimates of the regression coefficients and clusters are found by using a penalized likelihood estimator, which includes a cluster fusion penalty, to shrink the difference in fitted values from responses in the same cluster, and an L_1 penalty for simultaneous variable selection and estimation. We propose a two-step algorithm, that iterates between k-means clustering and solving the penalized likelihood function assuming the clusters are known, which has desirable parallel computational properties obtained by using the cluster fusion penalty. If the response variable clusters are known *a priori* then the algorithm reduces to just solving the penalized likelihood problem. Theoretical results are presented for the penalized least squares case, including asymptotic results allowing for $p \gg n$. We extend our method to the setting where the responses are binomial variables. We propose a coordinate descent algorithm for the normal likelihood and a proximal gradient descent algorithm for the binomial likelihood, which can easily be extended to other generalized linear model (GLM) settings. Simulations and data examples from business operations and genomics are presented to show the merits of both the least squares and binomial methods.

Keywords: Multivariate Regression, Clustering, Fusion Penalty

1. Introduction

In this article we consider the pair $(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^n$, with $\mathbf{x}_i^T = (x_{i1}, \ldots, x_{ip}) \in \mathcal{R}^p$ and $\mathbf{y}_i = (y_{i1}, \ldots, y_{ir})^T \in \mathcal{R}^r$. Define $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T \in \mathcal{R}^{n \times p}$ and $Y = (\mathbf{y}_1, \ldots, \mathbf{y}_n)^T \in \mathcal{R}^{n \times r}$. We initially assume the linear model

$$\mathbf{y}_i = B^{*T}\mathbf{x}_i + \boldsymbol{\epsilon}_i, \tag{1}$$

where $\boldsymbol{\epsilon}_i = (\epsilon_{i1}, \ldots, \epsilon_{ir})^T \in \mathcal{R}^r$ are realizations of an i.i.d. random variable with mean zero and covariance matrix Σ, $B^* = (\boldsymbol{\beta}_1^*, \ldots, \boldsymbol{\beta}_r^*) \in \mathcal{R}^{p \times r}$ and $\boldsymbol{\beta}_k^* = (\beta_{1k}^*, \ldots, \beta_{pk}^*)^T \in \mathcal{R}^p$. We will refer to the matrix of error as $E = (\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_n)^T \in \mathcal{R}^{n \times r}$. Under mild assumptions a consistent estimator of $\boldsymbol{\beta}_k^*$ is the ordinary least squares (OLS) estimator of

$$\tilde{\boldsymbol{\beta}}_k = \underset{\boldsymbol{\beta}_k}{\operatorname{argmin}} \sum_{i=1}^n (y_{ik} - \mathbf{x}_i^T \boldsymbol{\beta}_k)^2.$$

If $\boldsymbol{\epsilon}_i$ are i.i.d. and $\boldsymbol{\epsilon}_i \sim N(\mathbf{0}_r, \Sigma)$ the estimator $\tilde{\boldsymbol{\beta}}_k$ is the MLE. This estimator does not use the other responses, ignoring potentially useful information.

Throughout this paper for a vector $\mathbf{a}$ define $||\mathbf{a}||_q$ as the L_q norm and for a matrix A we define $||A||_q$ as the entrywise L_q norm. If there is *a priori* information that the fitted values of response k and m should be close then we could impose a penalty on the difference in the fitted values and consider the estimators

$$(\tilde{\boldsymbol{\beta}}_k, \tilde{\boldsymbol{\beta}}_m) = \underset{\boldsymbol{\beta}_k, \boldsymbol{\beta}_m}{\operatorname{argmin}} \sum_{i=1}^n \left\{ (y_{ik} - \mathbf{x}_i^T \boldsymbol{\beta}_k)^2 + (y_{im} - \mathbf{x}_i^T \boldsymbol{\beta}_m)^2 \right\} + \frac{\gamma}{n} ||X(\boldsymbol{\beta}_k - \boldsymbol{\beta}_m)||_2^2, \tag{2}$$

where γ is a tuning parameter controlling the amount of agreement between the two fitted values vectors. We propose an objective function that generalizes (2) for multiple responses from multiple clusters that may not be known *a priori*. The proposed objective function also includes an L_1 penalty for simultaneous estimation and variable selection, which allows our method to be used to increase prediction accuracy, select relevant variables for each response, and detect groupings of response variables without assuming or estimating a covariance structure. In our theory, simulations, and applied examples we consider cases where $p \gg n$. We extend the proposed method to the generalized linear model framework, specifically focusing on multiple binary responses. This extension allows the method to be used in many different contexts, such as understanding co-morbidities related to patient information recorded in electronic medical records, or product level purchasing habits of customers based on information obtained from a loyalty program. We propose a coordinate descent algorithm for the least squares case and proximal coordinate descent algorithm for the binomial GLM case, which provides a general framework for extending the method to other GLM or M-estimator settings.

Our work has been influenced by previous work in estimating high dimensional models. When $\frac{1}{n}X'X = I_p$ the penalty function is equivalent to a ridge penalty (Hoerl and Kennard, 1970) on the difference of the coefficient vectors for the two responses. We add the L_1 penalty as proposed in Tibshirani (1996) to do simultaneous variable selection and estimation. Similar to the work of Zou and Hastie (2005) we combine the ridge and L_1 penalties. The proposed estimator simultaneously estimates clusters of the response and fuses the fitted values of the clustered responses. Previous work has been done on clustering covariates for high dimensional regression with a univariate response. This work is most similar to the work of Witten et al. (2014) who proposed the cluster elastic net (CEN) that simultaneously estimates clusters of covariates and fuses the effects of covariates within the same cluster. Our proposed method is also similar to Grace estimators proposed in Li and Li (2008) and Li and Li (2010), which use regularization based on external network information to minimize the difference of coefficients for related predictors and use a lasso penalty for sparsity. Huang et al. (2011) proposed the sparse Laplacian shrinkage method, which preforms variable selection and promotes similarities among coefficients of correlated covariates. Zhao and Shojaie (2016) proposed the Grace Test, a testing framework for Grace estimators, that

allows for some uncertainty in the graph and showed that if the external graph is informative it increases the power of the Grace test. Bühlmann et al. (2013) proposed two different penalized methods for clustered covariates in high-dimensional regression: cluster representative lasso (CRL) and cluster group lasso (CGL). In CRL the covariates are clustered, dimension reduction is done by replacing the original covariates with the cluster centers and a lasso model is fit using the cluster centers as covariates. In CGL the group penalty of Yuan and Lin (2005) is applied using the previously found clusters as the groups. Zhou et al. (2017) demonstrated that averaging over models using different cluster centers for both responses and predictors can improve prediction accuracy of DNase I hypersensitivity using gene expression data. Kim et al. (2009) proposed graph-guided fused lasso (GGFL) to the specific problem of association analysis to quantitative trait networks. GGFL presents a fused lasso framework in multivariate regression that leverages correlated traits based on a network structure. Our work is related to the fused lasso literature as well, though we do not achieve exact fusion (Tibshirani et al., 2005; Rinaldo, 2009; Hoefling, 2010; Tibshirani, 2014). The proposed method differs from the works mentioned in this setting because it focuses on using correlation between the response variables to improve estimation, however all of the works mentioned were instrumental in helping us derive our final estimator.

The idea of using information from different responses to improve estimation in multivariate regression is not new and our work builds upon previous works in this area. Breiman and Friedman (1997) introduced the Curds and Whey method whose predictions are an optimal linear combination of least squares predictions. Rothman et al. (2010) proposed multivariate regression with covariance estimation (MRCE), which is a penalized likelihood approach to simultaneously estimate the regression coefficients and the inverse covariance matrix of the errors. MRCE leverages correlation in unexplained variation to improve estimation, while our proposed method leverages correlation in explained variation to improve estimation. Other estimators assume both the response and covariates are multivariate normal and exploit this structure to derive estimators (Lee and Liu, 2012; Molstad and Rothman, 2016). Rai et al. (2012) proposed a penalized likelihood method for multivariate regression that simultaneously estimates regression coefficients, the inverse covariance matrix of the errors, and the covariance matrix of the regression coefficients across responses using lasso type penalties. Peng et al. (2010) introduced regularized multivariate regression for identifying master predictors (remMap), which relies on *a priori* information about valuable predictors and imposes a group L_1 and L_2 norm, across responses, on all covariates not prespecified as being useful predictors. Kim and Xing (2012) proposed the tree guided group lasso, which uses an *a priori* hierarchical clustering of the responses to define overlapping group lasso penalties for the multivariate regression model. They propose a weighting method that ensures all coefficients are penalized equally, while using the hierarchical structure to impose a similar sparsity structure across highly correlated responses.

Another approach to improving efficiency is by doing dimension reduction on Y to find a smaller subspace that retains the material information needed for estimation of the regression coefficients (Cook et al., 2010; Cook and Zhang, 2015; Sun et al., 2015). Cook et al. (2010) introduced the envelope estimator for the multivariate linear model, which projects the maximum likelihood estimator onto the estimated subspace with the material information. Cook and Zhang (2015) provided envelope models for GLMs and weighted least squares. Sun et al. (2015) proposed a sparse regression model (SPReM) for estimating models where r is very large. SPReM projects the response variables into a lower-dimensional space while maintaining the structure needed for a specific hypothesis test. The key difference between our proposed method and these approaches is that we are interested in simultaneously estimating clustering of the response variables and fusing the fitted values from responses within the same cluster.

The proposed method simultaneously estimates clusters of the response and coefficients. Changes in cluster groups are discrete changes and as a result our objective function is discontinuous, similar to k-means clustering, thus making it difficult to derive an efficient algorithm that will find the optimal estimates for coefficients and groups. Witten et al. (2014) dealt with a similar difficulty for the CEN estimator, but noticed that if the groups are fixed then the problem is convex, while if the regression coefficients are fixed the problem becomes a k-means clustering problem. We modify the approach proposed in Witten et al. (2014) to our problem of grouping responses and extend the approach to the case of generalized linear models, specifically the binomial logistic model. In our theoretical results we assume the clustering groups are known, but the problem remains challenging as we are dealing with multiple responses, allow for $p \gg n$ and for p to increase with n.

In Section 2 we present our method for the multivariate linear regression model and provide theoretical results, including consistency of our estimator, to better understand the basic properties of the penalized likelihood solution. In Section 3 we provide details on the two-step iterative algorithm and show estimating the regression coefficients for the different clusters is an embarrassingly parallel problem, which is a property of our cluster fusion penalty that fuses within group fitted values. This avoids issues that would arise in fusing all possible combinations of regression coefficients, or having to specify a fusion set *a priori*. Examples of the issues that can arise can be found in Price et al. (2017), who discussed the importance of choosing the fusion set, and the original fused lasso paper which fused only consecutive coefficients (Tibshirani et al., 2005). In Section 4 we present the model for binomial responses along with an algorithm, demonstrating how the use of the cluster fusion penalty can exploit relationships of response variables beyond the traditional Gaussian problem. Simulations for both conditional Gaussian and binomial responses are presented in Section 5. The least squares version of our method is applied to model baby birth weight, placental weight and cotinine levels given maternal gene expression and demographic information. The binomial case is applied to model concession stand purchases using customer information as covariates. Both applied analysis are presented in Section 6. We conclude with a summary in Section 7.

2. Least Squares Model

2.1 Method

First, we consider estimating (1) when there are Q unknown clusters of the r responses. We further assume that $\sum_{i=1}^{n} y_{ik} = 0$ for all $k = 1, \ldots, r$, $\sum_{i=1}^{n} x_{ij} = 0$ and $\sum_{i=1}^{n} x_{ij}^2 \leq n$ for all $j = 1 \ldots, p$. The model requires rp parameters to be estimated for prediction, which is problematic when r or p are large. Let $D = (D_1, \ldots, D_Q)$ be a partition of the set $\{1, \ldots, r\}$. For a set A define $|A|$ as the cardinality of that set. We propose the multivariate cluster

elastic net (MCEN) estimator as

$$(\hat{B},\hat{D}) = \underset{B\in\mathcal{R}^{p\times r},D_1,\ldots,D_Q}{\arg\min} \frac{1}{2n}\sum_{i=1}^{n}\sum_{c=1}^{r}(y_{ic}-\mathbf{x}_i^T\boldsymbol{\beta}_c)^2+\delta||B||_1 \\ +\frac{\gamma}{2n}\sum_{q=1}^{Q}\frac{1}{|D_q|}\sum_{l,m\in D_q}||X(\boldsymbol{\beta}_l-\boldsymbol{\beta}_m)||_2^2, \tag{3}$$

where Q is the number of clusters and γ and δ are non-negative user specified tuning parameters. In addition Q, the total number of clusters, can be considered a tuning parameter. The cluster fusion penalty, associated with tuning parameter γ, is used to exploit similarities in the fitted values. The lasso penalty, with tuning parameter δ, is used to perform simultaneous estimation and variable selection. When $\gamma = 0$ or $Q = r$, the optimization in (3) reduces to r independent lasso penalized least squares problems with tuning parameter δ. If $\hat{D}$ is known then the optimization in (3) can be split into Q independent optimizations that are similar to the optimizations presented in Li and Li (2008), Li and Li (2010), and Witten et al. (2014) and can be solved in parallel. We exploit this computational feature in our algorithm, which is a result of using the cluster fusion penalty.

The proposed method uses a combination of L_1 and L_2 penalties as proposed by Zou and Hastie (2005). Similar methods have been proposed for grouping the effects of predictors with a univariate response such as CEN (Witten et al., 2014) and Grace estimators (Li and Li, 2008, 2010; Zhao and Shojaie, 2016). Kim and Xing (2012) proposed a method that uses a predetermined hierarchical clustering of the responses that provides an L_1 penalty for all coefficients and a group L_2 penalty for responses that are grouped together. Chen et al. (2016) proposed a method using conjoint clustering to incorporate similarities in preferences between individuals in conjoint analysis. This method does not simultaneously estimate coefficients and groupings. It requires a two-step algorithm to estimate the number of clusters, and then estimates coefficients using regularization based on the estimated cluster. The proposed approach uses non-hierarchical clusters, allows for the clustering structure to be unknown before estimation of the coefficients and focuses more on imposing similar fitted values for grouped responses, compared to directly imposing a similar sparsity structure.

Selecting the triplet, (Q,γ,δ), of tuning parameters can be done by K-fold cross validation minimizing the squared prediction error. Let $\mathcal{F}_k$ be the set of indices in the kth fold, $k\in\{1,\ldots,K\}$, and $\hat{\boldsymbol{\beta}}_c^{(-\mathcal{F}_k)}(Q,\gamma,\delta)$ be the estimated regression coefficient vector using Q, γ and δ for response c produced from the training set with $\mathcal{F}_k$ removed. Then select the triplet, $(\hat{Q},\hat{\gamma},\hat{\delta})$, that minimizes

$$V(Q,\delta,\gamma)=\sum_{k=1}^{K}\sum_{c=1}^{r}\sum_{i\in\mathcal{F}_k}\left\{y_{ic}-x_i^T\hat{\boldsymbol{\beta}}_c^{(-\mathcal{F}_k)}(Q,\gamma,\delta)\right\}^2. \tag{4}$$

2.2 Theoretical Results

For theoretical discussions we assume that D is known for some fixed value of Q. This is because for D unknown the objective function in (3) is discontinuous because of the discrete changes in groups, however if D is known (3) is a convex function. In this section we will look at properties of the MCEN estimator for the special case of fixed n and p with $\delta = 0$. In addition, we present a consistency result that allows for $p \gg n$ when $\delta = o(1)$ and $\gamma = o(1)$.

Thus, the first two theorems refer to the following estimator

$$\bar{B} = \underset{B\in\mathcal{R}^{p\times r}}{\arg\min}\frac{1}{2n}\sum_{i=1}^{n}\sum_{c=1}^{r}(y_{ic}-\mathbf{x}_i^T\boldsymbol{\beta}_c)^2+\delta||B||_1 \\ +\frac{\gamma}{2n}\sum_{q=1}^{Q}\frac{1}{|D_q|}\sum_{l,m\in D_q}||X(\boldsymbol{\beta}_l-\boldsymbol{\beta}_m)||_2^2. \tag{5}$$

The estimator $\bar{B}$ does not simultaneously estimate the groups, it assumes they are known *a priori*, and thus is different than $\hat{B}$. There are instances where the grouping structure is known before data analysis and thus using $\bar{B}$ would be preferable in practice. In addition $\bar{B}$ is a key component to the algorithm discussed in Section 3. We begin by relating the estimator in (5) to ordinary least squares (OLS), for the special case of $\delta = 0$. Removing the L_1 penalty allows us to derive a closed form for the estimator.

Theorem 1 *Assume $n > p$, $\delta = 0$, and Q and γ are fixed values. Define $\dot{B} = (\dot{\boldsymbol{\beta}}_1,\ldots,\dot{\boldsymbol{\beta}}_r)$ to be the OLS estimates for the r response variables and $\bar{B} = (\bar{\boldsymbol{\beta}}_1,\ldots,\bar{\boldsymbol{\beta}}_r)$ be the solution to (5) with tuning parameter γ. Given $l\in D_q$ then $\bar{\boldsymbol{\beta}}_l$ has the closed form solution of*

$$\bar{\boldsymbol{\beta}}_l = \dot{\boldsymbol{\beta}}_l + \frac{2\gamma}{(1+2\gamma)|D_q|}\sum_{c\in D_q c\neq l}(\dot{\boldsymbol{\beta}}_c-\dot{\boldsymbol{\beta}}_l). \tag{6}$$

Theorem 1 provides some intuition about the MCEN estimator. As γ increases the MCEN estimator approaches a weighted average of the OLS coefficients within a cluster. In addition the results from Theorem 1 can be used to calculate the bias and variance of $\bar{B}$, which are needed for proving Theorem 2. The proof of Theorem 1 and the following Theorems can be found in the appendix.

Theorem 2 *Assume $E(\epsilon_{ic}^2)=1$ for all $i\in\{1,\ldots,n\}$ and $c\in\{1,\ldots,r\}$ and $E(\epsilon_{ic}\epsilon_{ik})=\rho$ for $c\neq k$, where $\rho\in(0,1)$. Set $\delta=0$, then for a fixed n and p where $n>p$ there exists a positive γ such that*

$$E\left(\left|\left|\bar{B}-B^*\right|\right|_2^2\right) < E\left(\left|\left|\dot{B}-B^*\right|\right|_2^2\right), \tag{7}$$

where B^* are the true regression coefficients, $\dot{B}$ is as defined in Theorem 1 and $\bar{B}$ is as defined in (5).

Similar to ridge regression Theorem 2 shows that for some positive γ the estimator from (5) has a smaller mean squared error than OLS. Note, we are not assuming that for $l,s\in D_m$ that $\boldsymbol{\beta}_l^* = \boldsymbol{\beta}_m^*$ and unless this condition holds the estimator $\bar{B}$ is biased. Thus, there exists a value of γ for which there is a favorable bias-variance trade off.

Next we examine the asymptotic performance of the estimator with the L_1 penalty. At times it will be easier to refer to a vectorized version of a matrix and for any matrix $A\in\mathcal{R}^{a\times b}$, $\text{vec}(A)\in\mathcal{R}^{ab}$. Where $\text{vec}(A)$ is the vector formed by stacking the columns of A.

Define S as the set of active predictors. That is, S is a subset of $\{1,\ldots,rp\}$ where $m \in S$ if $\text{vec}(B^*)_m \neq 0$. The subspace for the active predictors is

$$\mathcal{M}(S) \equiv \{\boldsymbol{\theta} \in \mathcal{R}^{pr} | \theta_j = 0 \text{ if } j \notin S\}.$$

The parameter space will be separated using projections of vectors into orthogonal complements. We define a projection of a vector $\mathbf{u}$ into space $\mathcal{M}(S)$ as

$$\mathbf{u}_{\mathcal{M}(S)} \equiv \underset{\mathbf{v}\in\mathcal{M}(S)}{\arg\min} ||\mathbf{u}-\mathbf{v}||_2.$$

The orthogonal complement of space $\mathcal{M}(S) \subseteq \mathcal{R}^p$ is

$$\mathcal{M}^{\perp}(S) \equiv \{\mathbf{v} \in \mathcal{R}^{pr} | \langle \mathbf{u}, \mathbf{v}\rangle = 0 \text{ for all } \mathbf{u} \in \mathcal{M}(S)\}.$$

The following set is central to our proof of consistency,

$$\mathcal{C} \equiv \{\boldsymbol{\theta} \in \mathcal{R}^{pr} | \, ||\boldsymbol{\theta}_{\mathcal{M}^{\perp}(S)}||_1 \leq ||\boldsymbol{\theta}_{\mathcal{M}}||_1\}.$$

For our proof of the consistency of $\bar{B}$ we make the following six assumptions:

A1 Define $\mathbf{X}_j$ to be the jth column vector of X, then $\mathbf{X}_j \in \mathcal{R}^p$ has the condition that $\frac{||\mathbf{X}_j||_2^2}{n} \leq 1$.

A2 Define $\boldsymbol{\epsilon}_c = (\epsilon_{1c},\ldots,\epsilon_{nc})^T \in \mathcal{R}^n$ as the error vector for response c. The error vector $\boldsymbol{\epsilon}_c$ has a mean of zero and sub-Gaussian tails for all $c \in \{1,\ldots,r\}$. That is, there exists a constant σ_c such that for any $\mathbf{a} \in \mathcal{R}^n$, with $||\mathbf{a}||_2 = 1$,

$$P(|\langle \boldsymbol{\epsilon}_c, \mathbf{a}\rangle| > t) \leq 2\text{exp}\left(-\frac{t^2}{2\sigma_c^2}\right).$$

Define $\sigma = \max_c \sigma_c$.

A3 Define $\tilde{X} = I_r \otimes X \in \mathcal{R}^{rn\times rp}$, where $\otimes$ is the standard Kronecker product. There exists a positive constant κ such that

$$\kappa||\boldsymbol{\theta}||_2^2 \leq \min_{\boldsymbol{\theta}\in\mathcal{C}} n^{-1}||\tilde{X}\boldsymbol{\theta}||_2^2.$$

A4 There exists a positive constant $\acute{b}$ such that $\max_{q=1,\ldots,Q}\max_{(l,k)\in D_q}||\boldsymbol{\beta}_l^* - \boldsymbol{\beta}_k||_2 \leq \acute{b}$.

A5 Given $l,k \in D_q$, if $\beta_{lj}^* = 0$ then $\beta_{kj}^* = 0$, for all $j \in \{1,\ldots,p\}$ and $q \in \{1,\ldots,Q\}$.

A6 Define $\rho_{\max}(A)$ as the maximum eigenvalue of square matrix A and $X_{S_{D_q}}$ as the matrix of true predictors for cluster q, where the jth predictor is a true predictor if $\boldsymbol{\beta}_{lj}^* \neq 0$ for any $l \in D_q$. There exists a positive constant $\rho_{\max}$ such that

$$\max_{q=1,\ldots,q} \rho_{\max}\left(\frac{1}{n}X_{S_{D_q}}^T X_{S_{D_q}}\right) \leq \rho_{\max}.$$

Assumption A1 is a standard assumption for lasso-type penalties and can be achieved by appropriately scaling the covariates, which is commonly done in penalized regression. Assumption A2 is a generalization of the sub-Gaussian error assumption for penalized regression for a univariate response. Assumption A1 could be relaxed to allow for certain unbounded covariates, but then A2 would be replaced by assuming the errors are normally distributed (Candes and Tao, 2007; Meinshausen and Yu, 2009). Assumption A3 is a generalization of the common restricted eigenvalue assumption. Motivation for assumption A3 is discussed in great detail by Negahban et al. (2012) and a version for $r = 1$ has been used in several works analyzing asymptotic behaviors of the lasso estimator (Bickel et al., 2009; van de Geer and Bühlmann, 2009; Meinshausen and Yu, 2009). Assumptions A4 and A5 provide that the true coefficients are similar for responses in the same group. Assumption A5 provides that they have the same sparsity structure. While, assumption A4 ensures that the difference in the non-zero elements can be bounded by a finite constant, even if the number of predictors increases with n. Assumption A6 assumes the maximum eigenvalues of the sample covariance of the true predictors are bounded, a common assumption in high-dimensional work. Assumptions A4-A6 can be replaced by an assumption similar to assumption A2 from Witten et al. (2014) that if $b, c \in D_m$ then $\boldsymbol{\beta}_b^* = \boldsymbol{\beta}_c^*$, for all $m \in \{1,\ldots,Q\}$, thus the bias of the MCEN estimator only comes from the L_1 penalty.

Using assumptions A3 and A5 we can provide a closed form definition of the asymptotic bias when $\delta = 0$. This relationship will be central to our proof of consistency of $\bar{B}$.

Corollary 3 *Let B^* be an s-sparse matrix, whose column vectors are all sparse and $E[X^TX/n] \in \mathcal{R}^{p\times p}$ to be a positive definite matrix. Assume Q and γ are fixed values. Define,*

$$\acute{B} = \left(\acute{\beta}_1,\ldots\acute{\beta}_r\right) = \underset{\beta_1,\ldots,\beta_r\in\mathcal{R}^p}{\arg\min}\; E\left(\frac{1}{2n}\sum_{i=1}^{n}\sum_{c=1}^{r}(y_{ic} - \mathbf{x}_i^T\boldsymbol{\beta}_c)^2 + \frac{\gamma}{2n}\sum_{q=1}^{Q}\frac{1}{|D_q|}\sum_{l,m\in D_q}||X(\boldsymbol{\beta}_l - \boldsymbol{\beta}_m)||_2^2\right),$$

Assume $l \in D_q$ then $\acute{\beta}_l$ has closed form solution,

$$\acute{\beta}_l = \beta_l^* + \frac{2\gamma}{(1+2\gamma)|D_q|}\sum_{c\in D_q, c\neq l}(\beta_c^* - \beta_l^*).$$

Corollary 3 provides insight into what $\bar{B}$ would converge to for a fixed γ. Knowing this exact relationship is used in our consistency proof because it allows us to understand the exact nature of the bias caused by the L_2 penalty and for γ going to zero at a given rate we can show that the bias is asymptotically negligible.

Theorem 4 *Let B^* be an s-sparse matrix, whose column vectors are all sparse and $E[X^TX/n] \in \mathcal{R}^{p\times p}$ to be a positive definite matrix. Given $\delta = 16\sigma\sqrt{\frac{\log(rp)}{n}}$, $\gamma \leq \frac{5}{4\rho_{\max}\acute{b}}\sigma\sqrt{\frac{\log(rp)}{n}}$ and assumptions A1-A6 hold then there exist constants c_1, c_2, c_3 and c_4 such that*

$$\left|\left|vec\left(\bar{B} - B^*\right)\right|\right|_2 \leq \sigma\sqrt{\frac{s\log(rp)}{n}}\left(\frac{c_3}{\kappa} + \frac{c_4}{\rho_{\max}}\right), \tag{8}$$

with probability at least $1 - c_1\exp(-c_2 n\delta^2)$.

The convergence rate derived is similar to rates found in lasso-type estimators with a univariate response, with $\log(rp)$ replacing $\log(p)$ to accommodate for the multiple responses (Bickel et al., 2009; Candes and Tao, 2007; Meinshausen and Yu, 2009; Negahban et al., 2012). Thus, under the conditions of Theorem 4 if $pr \to \infty$ then $||\text{vec}(\bar{B} - B^*)||_2 = O_p\left\{\sqrt{\frac{s\log(rp)}{n}}\right\}$. Our results prove consistency of our estimator when the group structure is known. Zhao and Shojaie (2016) propose the Grace test for an estimator with a similar penalty for grouping predictors with a univariate response and establish asymptotic results that allow for inference even if there is some uncertainty to the grouping structure.

3. Algorithm

The optimization in (3) is discontinuous because of the estimation of cluster assignments. To simplify the optimization we propose an iterative algorithm that alternates between estimating the groups with the regression coefficients fixed, and estimating the regression coefficients with the groups fixed. If the clusters are known (5) then it is a convex optimization problem that can be solved by a coordinate descent algorithm. Let $R = \frac{1}{n}X^TX$, define $\mathbf{R}_j$ as the jth column of R. The super script $(-h)$ denotes the hth element of the vector has been removed, and r_{jj} is jth diagonal element of R. Define $S(a,b) = \text{sign}(a)\max(0, |a| - b)$. To solve (5), we use a coordinate descent algorithm where each update is preformed by

$$\bar{\beta}_{jk} \leftarrow \frac{S\left[\frac{1}{n}y_k^T\mathbf{X}_j - \left\{1 + \frac{\gamma(|D_q|-1)}{|D_q|}\right\}\mathbf{R}_j^{(-j)T}\overline{\boldsymbol{\beta}}_k^{(-j)} + \frac{\gamma}{|D_q|}\sum_{s\in D_q, s\neq k}\mathbf{R}_j^T\overline{\boldsymbol{\beta}}_s, \delta/2\right]}{r_{jj}\left(1 + \gamma\frac{|D_q|-1}{|D_q|}\right)}. \quad (9)$$

Thus, (5) is solved by iterating through $j \in \{1,\ldots,p\}$ and $k \in \{1,\ldots,r\}$ until the solution converges, similar to other coordinate descent solutions (Witten et al., 2014; Li and Li, 2010, 2008; Friedman et al., 2008). If B is known then the solution to (3) reduces to the well studied k-means clustering problem. Recognizing this, we propose a two-step iterative procedure to obtain a local minimum. To start the algorithm an initial estimate of D or B is needed. We propose initializing the regression coefficients for the different responses separately with the elastic net estimator of response c of

$$\hat{\boldsymbol{\beta}}_c^1 = \underset{\boldsymbol{\beta}_c\in\mathcal{R}^p}{\arg\min}\frac{1}{2n}\sum_{i=1}^n(y_{ic} - \mathbf{x}_i^T\boldsymbol{\beta}_c)^2 + \delta||\boldsymbol{\beta}_c||_1 + \gamma||\boldsymbol{\beta}_c||_2^2, \quad (10)$$

where $\hat{B}^w = \left(\hat{\boldsymbol{\beta}}_1^w,\ldots,\hat{\boldsymbol{\beta}}_r^w\right)$ represents the wth iterative estimate of B^*. Given a fixed (Q,γ,δ) we propose the following algorithm.

1. Begin with initial estimates, $\hat{\boldsymbol{\beta}}_1^1,\ldots,\hat{\boldsymbol{\beta}}_r^1$.
2. For the wth step, where $w > 1$, repeat the steps below until the group estimates do not change:
   (a) Hold $\hat{B}^{w-1}$ fixed and minimize,

$$\left(\hat{D}_1^w,\ldots,\hat{D}_Q^w\right) = \underset{D_1,\ldots,D_Q}{\text{minimize}}\left\{\sum_{q=1}^Q\frac{1}{|D_q|}\sum_{l,m\in D_q}\left|\left|X\left(\hat{\boldsymbol{\beta}}_l^{w-1} - \hat{\boldsymbol{\beta}}_m^{w-1}\right)\right|\right|_2^2\right\}. \quad (11)$$

The above can be solved by performing K-means clustering on the r $n-$dimensional vectors $X\hat{\boldsymbol{\beta}}_1^{w-1},\ldots,X\hat{\boldsymbol{\beta}}_r^{w-1}$.

   (b) Holding $\hat{D}_1^w,\ldots,\hat{D}_Q^w$ fixed the wth estimate of B^* is

$$\begin{aligned}\hat{B}^w = \underset{B\in\mathcal{R}^{p\times r}}{\arg\min}\frac{1}{2n}\sum_{i=1}^n\sum_{c=1}^r(y_{ic} - \mathbf{x}_i^T\boldsymbol{\beta}_c)^2 + \delta||B||_1 \\ + \frac{\gamma}{2n}\sum_{q=1}^Q\frac{1}{|\hat{D}_q^w|}\sum_{l,m\in\hat{D}_q^w}||X(\boldsymbol{\beta}_l - \boldsymbol{\beta}_m)||_2^2.\end{aligned} \quad (12)$$

   Note that for the groups known, instead of estimated, $\hat{B}^w$ is equivalent to $\bar{B}$. Thus (12) can be solved using the coordinate descent solution from (9) using $\hat{B}^{w-1}$ as the initial estimates for the coordinate descent algorithm.

Convergence is reached once the groups at the wth and $(w-1)$th iteration are the same. The optimization in (5) is separable with respect to $\hat{D}$, and results in Q independent optimization problems that can be solved in parallel. The algorithm for (5) can be solved in solution path type form where we iterate across different values of δ in a similar fashion as proposed in the glmnet algorithm (Friedman et al., 2008). If all of the initial elastic net estimators are fully sparse, we set the solution to be a zero matrix and thus following Friedman et al. (2008), initialize the algorithm by beginning the sequence with $\delta_{\max}$ at

$$\delta_{\max} = 2\max_{j,k}\left|\frac{\sum_{i=1}^n y_{ik}x_{ij}}{n}\right|.$$

Our two-step approach is closely related to the CEN algorithm proposed by Witten et al. (2014), who proposed a two-step algorithm where the two steps are solved by coordinate descent and k-means algorithms. The major difference in our proposal is that we cluster the responses rather than the predictors, and have the ability to solve the optimization in parallel due to the nature of our regularization in a multiple response setting.

4. Binomial Model

4.1 Method

Next we extend the multivariate cluster elastic net to generalized linear models. We focus specifically on the binomial response case, but our discussion here will scale to other exponential families. A fusion penalty has been proposed for merging groups from a multinomial response (Price et al., 2017), but our method differs as it aims to leverage association between multiple binomial responses. Kasap et al. (2016) proposed an ensemble method that combines association rule mining and binomial logistic regression via a multiple linear regression model. Our method differs from this by simultaneously estimating the clusters of the response variables and estimating the regression coefficients. An example is n customers, with p covariates, such as demographic and historic purchasing variables, and r indicators of product purchasing statuses for each customer. You could run r independent models, but this would not allow for modeling the relationship between the different products. Extending the multivariate cluster elastic net to multiple binomial responses would

allow us to group products by purchase probabilities to identify and use relationships between products. This could also be used to create a probabilistic model for diseases based on patient demographic and medical information.

For the linear model we ignore the intercept term as it can be removed by appropriately scaling Y and X. This is not possible in logistic regression, therefore the model needs an intercept term. We define $\mathbf{u}_i = (1, \mathbf{x}_i^T)^T \in \mathcal{R}^{p+1}$, $U = (\mathbf{u}_1^T, \ldots, \mathbf{u}_n)^T \in \mathcal{R}^{n\times p+1}$, $\mathbf{U}_k \in \mathcal{R}^n$ as the kth column vector of U and $\tilde{R} = U^T U$. The true coefficients for response k is defined as $\boldsymbol{\theta}_k^* \in \mathcal{R}^{p+1}$, $\Theta^* = (\boldsymbol{\theta}_1^*, \ldots, \boldsymbol{\theta}_r^*) \in \mathcal{R}^{p+1\times r}$, $\Theta_{-1}^* \in \mathcal{R}^{p\times r}$ is the matrix with the first row, the row of intercept coefficients, of Θ^* removed and $\boldsymbol{\theta}_{(-1)k}^* \in \mathcal{R}^p$ is the kth column vector of Θ_{-1}^*. In this model y_{ik} is an independent draw from

$$Bin\left(1, \pi_{ik}^*\right), \tag{13}$$

where

$$\pi_{ik}^* = \frac{\exp(\mathbf{u}_i^T \boldsymbol{\theta}_k^*)}{1 + \exp(\mathbf{u}_i^T \boldsymbol{\theta}_k^*)}. \tag{14}$$

The penalized negative log-likelihood function is

$$\begin{aligned} &\sum_{k=1}^{r}\sum_{i=1}^{n} y_{ik}\mathbf{u}_i^T\boldsymbol{\theta}_k - \log\left\{1 + \exp(\mathbf{u}_i^T\boldsymbol{\theta}_k)\right\} \\ &+ \frac{\gamma}{2n}\sum_{q=1}^{q}\frac{1}{|D_q|}\sum_{l,m\in D_q}||U(\boldsymbol{\theta}_l - \boldsymbol{\theta}_m)||_2^2 + \delta||\Theta_{-1}||_1. \end{aligned} \tag{15}$$

4.2 Algorithm

We propose solving (15) by approximating it with a penalized quadratic function similar to the glmnet algorithm proposed by Friedman et al. (2008). Define,

$$g(\pi_{ik}) = \log\left(\frac{\pi_{ik}}{1-\pi_{ik}}\right) = \mathbf{u}_i^T\boldsymbol{\theta}_k. \tag{16}$$

To implement this approximation we define

$$\begin{aligned} z_{ik} &= g(y_{ik}) = g(\pi_{ik}) + \frac{y_{ik} - \pi_{ik}}{\pi_{ik}(1-\pi_{ik})}, & (17)\\ w_{ik} &= \pi_{ik}(1-\pi_{ik}), & (18)\\ -l_{Ak}(\boldsymbol{\theta}_k) &= \sum_{i=1}^{n} w_{ik}(z_{ik} - \mathbf{u}_i^T\boldsymbol{\theta}_k)^2. & (19) \end{aligned}$$

Note that z_{ik} is just the first order Taylor approximation of $g(y_{ik})$, and that w_{ik} is the conditional variance of z_{ik} given $\mathbf{u}_i$. Define $\mathbf{Z}_k = (z_{1k}, \ldots, z_{nk})^T \in \mathcal{R}^n$ and $\mathbf{W} = (w_{1k}, \ldots, w_{nk})^T \in \mathcal{R}^n$.

The MCEN estimator for the binomial model is

$$\begin{aligned} (\hat{\Theta}, \hat{D}) = \underset{\Theta\in\mathcal{R}^{p+1\times r}, D_1,\ldots,D_Q}{\arg\min} &\sum_{k=1}^{r} -l_{Ak}(\boldsymbol{\theta}_k) + \delta||\Theta_{-1}||_1 \\ &+ \frac{\gamma}{2n}\sum_{q=1}^{q}\frac{1}{|D_q|}\sum_{l,m\in D_q}||U(\boldsymbol{\theta}_r - \boldsymbol{\theta}_s)||_2^2. \end{aligned} \tag{20}$$

If the groups are known *a priori* the solution is

$$\begin{aligned} \bar{\Theta} = \underset{\Theta\in\mathcal{R}^{p+1\times r}}{\arg\min} &\sum_{k=1}^{r} -l_{Ak}(\boldsymbol{\theta}_k) + \delta||\Theta_{(-1)}||_1 \\ &+ \frac{\gamma}{2n}\sum_{q=1}^{q}\frac{1}{|D_q|}\sum_{l,m\in D_q}||U(\boldsymbol{\theta}_r - \boldsymbol{\theta}_s)||_2^2. \end{aligned} \tag{21}$$

For same length vectors $\mathbf{a}$ and $\mathbf{b}$ let $\mathbf{a}\circ\mathbf{b}$ represent the component wise multiplication of the two vectors. To solve (21), we use a proximal coordinate descent algorithm where each update is performed by

$$\bar{\theta}_{jk} \leftarrow \frac{S\left\{(\mathbf{w}_k\circ\mathbf{z}_k)^T\mathbf{U}_j - M_{jk}, I(j\neq 0)\delta/2\right\}}{r_{jj}\gamma\frac{|D_q|-1}{n|D_q|} + \mathbf{U}_j^T(\mathbf{w}_k\circ\mathbf{U}_j)}, \tag{22}$$

where

$$\begin{aligned} M_{jk} &= \sum_{c=1,c\neq h}^{p}\mathbf{U}_j^T(\mathbf{w}_k\circ\mathbf{U}_c)\bar{\Theta}_{cj} + \frac{\gamma(|D_q|-1)}{n|D_q|}\tilde{\mathbf{R}}_j^{(-j)T}\bar{\boldsymbol{\theta}}_k^{(-j)} \\ &- \frac{\gamma}{n|D_q|}\sum_{s\in D_q, s\neq k}\tilde{\mathbf{R}}_j^T\bar{\boldsymbol{\theta}}_s. \end{aligned}$$

The final solution is found by iterating through $j \in \{1, \ldots, p\}$ and $k \in \{1, \ldots, r\}$ until convergence. Again this is a solution similar to the glmnet algorithm proposed by Friedman et al. (2008).

To solve (20), we propose an algorithm that is similar in nature to the penalized least squares solution proposed in Section 3. The main difference is that we solve (20) with $D_1, \ldots, D_Q$ fixed using an iteratively reweighed least squares (IRWLS) solution with a proximal coordinate descent algorithm. The initial estimator for each response is done separately with

$$\hat{\boldsymbol{\theta}}_k^1 = \underset{\boldsymbol{\theta}_k\in\mathcal{R}^{p+1}}{\arg\min} -l_{Ak}(\boldsymbol{\theta}_k) + \delta||\boldsymbol{\theta}_{(-1)k}||_1 + \gamma||\boldsymbol{\theta}_{(-1)k}||_2^2. \tag{23}$$

The following is our proposed algorithm for estimating (20).

1. Begin with initial estimates of $\hat{\Theta}^1 = \left(\hat{\boldsymbol{\theta}}_1^1, \ldots, \hat{\boldsymbol{\theta}}_r^1\right) \in \mathcal{R}^{p+1\times r}$.
2. For the wth step, where $w > 1$, repeat the steps below until the group estimates do not change:
   (a) Hold $\hat{\Theta}^{w-1}$ fixed and minimize

$$\left(\hat{D}_1^w, \ldots, \hat{D}_Q^w\right) = \underset{D_1,\ldots,D_Q}{\text{minimize}}\left\{\sum_{q=1}^{Q}\frac{1}{|D_q|}\sum_{l,m\in D_q}\left|\left|U\left(\boldsymbol{\theta}_l^{w-1} - \boldsymbol{\theta}_m^{w-1}\right)\right|\right|_2^2\right\}. \tag{24}$$

   The above can be solved by performing K-means clustering.

(2b) Holding $\hat{D}_1^w, \ldots, \hat{D}_Q^w$ fixed the wth update for the coefficients is

$$\begin{aligned}\hat{\Theta}^w = \underset{\Theta \in \mathcal{R}^{p+1\times r}}{\arg\min} \sum_{k=1}^{r} -l_{Ak}(\boldsymbol{\theta}_k) + \delta||\Theta_{-1}||_1 \\ + \frac{\gamma}{2n}\sum_{q=1}^{q}\frac{1}{|\hat{D}_q^w|}\sum_{l,m\in\hat{D}_q^w}||U(\boldsymbol{\theta}_r - \boldsymbol{\theta}_s)||_2^2.\end{aligned} \tag{25}$$

Where (25) can be solved using the proximal coordinate descent solution presented in (22), using $\hat{\Theta}^{w-1}$ as the initial estimates for the proximal coordinate descent algorithm.

The triplet (Q, γ, δ) can be selected using K-Fold cross validation maximizing the validation log-likelihood. Let $\mathcal{F}_k$ be the set of indices in the kth fold ($k \in \{1, \ldots, K\}$) and $\hat{\pi}_{ic}^{(-\mathcal{F}_k)}(Q, \gamma, \delta)$ be the estimated probability for observation i and response c produced from the model with $\mathcal{F}_k$ removed using Q, γ and δ. Specifically we select the triplet that maximizes

$$V(Q, \delta, \gamma) = \sum_{v=1}^{K}\sum_{c=1}^{r}\sum_{i\in\mathcal{F}_v}\left[y_{ic}\log\left\{\hat{\pi}_{ic}^{(-\mathcal{F}_k)}\right\} + (1-y_{ik})\log\left\{1-\hat{\pi}_{ic}^{(-\mathcal{F}_k)}\right\}\right]. \tag{26}$$

The quadratic approximation defined by (19), is a standard technique used to estimate parameters in generalized linear models, making this framework and our algorithm scalable to other exponential family settings (Faraway, 2006). Tuning parameter selection would then be done by updating (26) with the appropriate likelihood.

5. Simulations

5.1 Gaussian Simulations

In this section we compare the performances of the MCEN estimator (3), the true MCEN (TMCEN) (5), with clustering structure known *a priori*, the separate elastic net (SEN) estimator (10), the joint elastic net (JEN) estimator

$$\hat{B}_{\text{JEN}} = \underset{B}{\arg\min}\frac{1}{2n}\sum_{k=1}^{r}\sum_{i=1}^{n}(y_{ik} - \mathbf{x}_i^T\boldsymbol{\beta}_k)^2 + \delta\sum_{j=1}^{p}\sqrt{\beta_{j1}^2 + \ldots + \beta_{jr}^2} + \gamma\sum_{k=1}^{r}\sum_{j=1}^{p}\beta_{jk}^2, \tag{27}$$

and the tree-guided group lasso (TGL) (Kim and Xing, 2012). Define $\mathbf{B}_j \in \mathcal{R}^r$ as the jth row vector of matrix B. Given a tree T with vertices V, where each node $v \in V$ is associated with group G_v define $B_j^{G_v}$ as a vector of the jth predictors from responses in group G_v. The TGL estimator is

$$\hat{B}_{\text{TGL}} = \underset{B}{\arg\min}\frac{1}{2}\sum_{k=1}^{r}\sum_{i=1}^{n}(y_{ik} - \mathbf{x}_i^T\boldsymbol{\beta}_k)^2 + \delta\sum_{j=1}^{p}\sum_{v\in V}w_v||B_j^{G_v}||_2, \tag{28}$$

where w_v are weights that can vary with the nodes. See Kim and Xing (2012) for a detailed presentation of TGL, including how the weights, w_v, are derived.

The JEN and SEN models are fit using the `glmnet` package in R (Friedman et al., 2008). Tuning parameters for all methods are selected using 10-folds cross validation. For the MCEN and TMCEN methods cluster sizes of 2, 3 and 4 are considered. We include the TMCEN estimator for two reasons. First, in practice the TMCEN estimator could be used if the practitioner has a predetermined clustering of the responses. Second, the TMCEN is useful as a benchmark to compare with the MCEN estimator because if the grouping of responses is useful TMCEN provides the optimal grouping.In all of the simulations the sample size is 100 and the number of responses is 15. For the number of covariates we considered 12, 100 and 300. Next we define how the covariates are generated and then will present the generating process for the response variables.

Define $\tilde{\Sigma} \in \mathcal{R}^{12\times 12}$ with entries $\tilde{\sigma}_{ii} = 1$ and $\tilde{\sigma}_{ij} = \rho$, for $i \neq j$. Let $0_{a,b} \in \mathcal{R}^{a\times b}$ be a matrix with all entries equal to zero. The covariates are generated by $\mathbf{x}_i \sim N(\mathbf{0}_p, \Sigma_x)$, where $\Sigma_x = \tilde{\Sigma}$ for $p = 12$ and otherwise

$$\Sigma_x = \begin{pmatrix}\tilde{\Sigma} & 0_{p-12,p-12} \\ 0_{p-12,p-12} & I_{p-12}\end{pmatrix},$$

with $\rho = .7$.

For a group of responses we define the grouped coefficients as $\mathbf{b}_q(\eta, \lambda) = (\boldsymbol{\eta}_q - \lambda, \boldsymbol{\eta}_q^*, \boldsymbol{\eta}_q + \lambda, \boldsymbol{\eta}_q + 2\lambda, \boldsymbol{\eta} + 3\lambda) \in \mathcal{R}^{q\times 5}$, where λ is a constant and $\boldsymbol{\eta}_q \in \mathcal{R}^q$ with each element equal to η. In the case of $p = 12$ the matrix of coefficients is

$$B_{\eta,\lambda}^* = \begin{Bmatrix}\mathbf{b}_4(\eta,\lambda) & \mathbf{0}_{4,5} & \mathbf{0}_{4,5} \\ \mathbf{0}_{4,5} & \mathbf{b}_4(\eta,\lambda) & \mathbf{0}_{4,5} \\ \mathbf{0}_{4,5} & \mathbf{0}_{4,5} & \mathbf{b}_4(\eta,\lambda)\end{Bmatrix},$$

otherwise

$$B_{\eta,\lambda}^* = \begin{Bmatrix}\mathbf{b}_{10}(\eta,\lambda) & \mathbf{0}_{10,5} & \mathbf{0}_{10,5} \\ \mathbf{0}_{10,5} & \mathbf{b}_{10}(\eta,\lambda) & \mathbf{0}_{10,5} \\ \mathbf{0}_{10,5} & \mathbf{0}_{10,5} & \mathbf{b}_{10}(\eta,\lambda) \\ \mathbf{0}_{p-30,5} & \mathbf{0}_{p-30,5} & \mathbf{0}_{p-30,5}\end{Bmatrix}.$$

Define $\Sigma_\epsilon \in \mathcal{R}^{15\times 15}$ with $\sigma(\epsilon)_{ij}$ being the entry for the ith row and jth column of Σ_ϵ. The generating process for the response is

$$\mathbf{y}_i = {B_{\eta,\lambda}^*}^T\mathbf{x}_i + \boldsymbol{\epsilon}_i, \tag{29}$$

where $\boldsymbol{\epsilon}_i \sim N(\mathbf{0}_{15}, \Sigma_\epsilon)$, $\sigma(\epsilon)_{ii} = 1$ and $\sigma(\epsilon)_{ij} = 0$, for i not equal to j. In all simulations we set the sample size to 100, perform 50 replications and with p set consider the following 9 different combinations for the true coefficient matrix,

$$(\eta, \lambda) \in \{0.25, 0.5, 0.75, 1\} \times \{0.02, 0.05, 0.10\}.$$

Models are fit using the training data with a sample size of 100. The tree for TGL is defined by performing complete-linkage hierarchical clustering on the responses in the training data. In addition we generate 1000 additional testing samples to assess the prediction accuracies of the models. Let y_{ij}^* represent the ith training sample for the jth response and

$\hat{y}_{ij}$ represent a predicted value of that sample and response. The average squared prediction error (ASPE) is defined as

$$\frac{1}{15000}\sum_{i=1}^{1000}\sum_{j=1}^{15}\left(y_{ij}^{*}-\hat{y}_{ij}\right)^{2}. \tag{30}$$

We also report the mean squared error (MSE) of the estimators where for an estimator B

$$\text{MSE}\left(B\right)=\sum_{j=1}^{15}\left|\left|\boldsymbol{\beta}_{j}-\boldsymbol{\beta}_{j}^{*}\right|\right|_{2}^{2}. \tag{31}$$

In addition we report the number of true variables selected (TV), out of a maximum of 60 for $p = 12$ and 150 otherwise, and the number of false variables selected (FV). Box plots of the statistics for $p = 300$ and the different combinations of η and λ are reported in Figures 1–4. These results show that TMCEN generally outperforms all methods in terms of ASPE and MSE. The one exception being when $\eta = 1$, particularly for larger values of λ, TGL is competitive with or outperforms TMCEN. For larger values of λ we expect more bias in the MCEN and TMCEN solutions and our simulation setting is favorable to TGL because the sparsity structure is the same for responses in the same cluster. With regards to ASPE and MSE, MCEN generally does better than TGL when $\eta = .5$ or .75. This suggests that the MCEN approach is advantageous with several smaller signals, but the signals need to be strong enough to correctly identify the clustering of the responses. The MCEN method also outperforms JEN and SEN in terms of ASPE and MSE, except in the case of $\eta = .25$ where JEN outperforms MCEN. In this case the signal is too small resulting in the MCEN method not finding the true clustering structure, and thus the grouping penalty will not be optimal. The MCEN and TMCEN methods tend to pick a larger model than SEN, but a smaller model than JEN. This results in the MCEN and TMCEN methods correctly choosing more true predictors than SEN and fewer false positive predictors than JEN for weaker signal cases. In terms of variable selection MCEN and TMCEN tend to do better than TGL in terms of both true and false variable selection. For the stronger signal cases the SEN approach does the best in terms of variable selection, tending to have the maximum number of true covariates selected, while a smaller number of false covariates selected. Similar conclusions can be derived for the plots of $p = 12$ and $p = 100$, which are available in the supplementary material.

5.2 Binomial Simulations

In this setting we have a binomial response variable and compare performance of the MCEN estimator (20) to SEN (23), for $r = 15$ and $p = 12, 100$ or 300. The SEN models were fit using the `glmnet` package in R (Friedman et al., 2008). Similar to the previous section, the covariates are generated by $\mathbf{x}_i \sim N(\mathbf{0}_p, \Sigma_x)$, where Σ_x has the same structure provided in the Gaussian simulations with $\rho = .9$.

We use the same structure of B presented in Section 5.1, consider the same values of η and λ and again perform 50 replications with a sample size of 100. Tuning parameters for the models are estimated via 10-folds cross validation as explained in Section 4.2. For Q, the number of groups, we consider values of 2, 3, and 4. For the SEN method each response

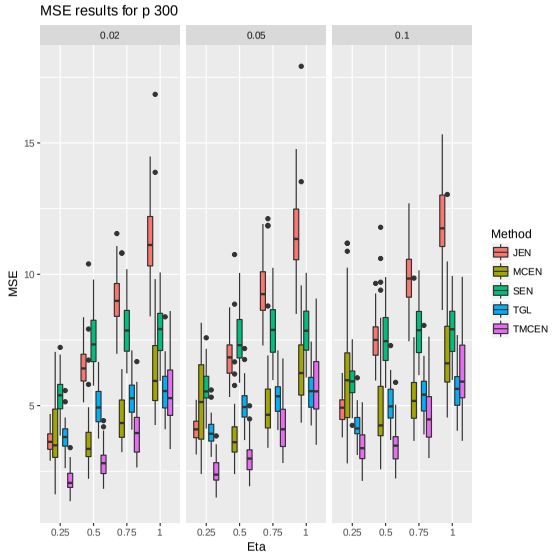


Figure 1: MSE results for the Gaussian simulations with p equal to 300. Different box plots correspond to different values of λ, while x-axis values are for different values of η.

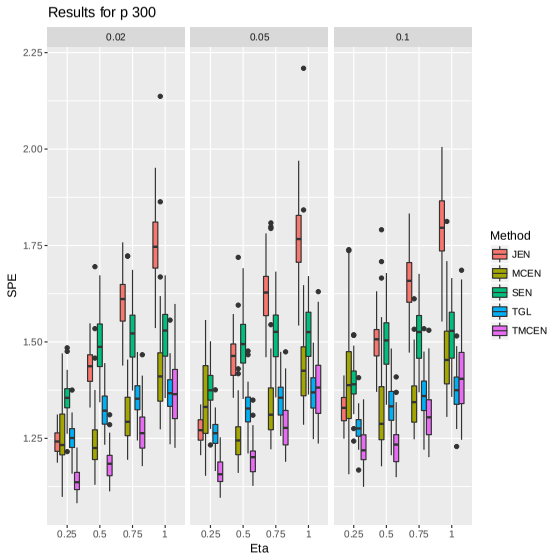


Figure 2: ASPE results for the Gaussian simulations with p equal to 300. Different box plots correspond to different values of λ, while x-axis values are for different values of η.

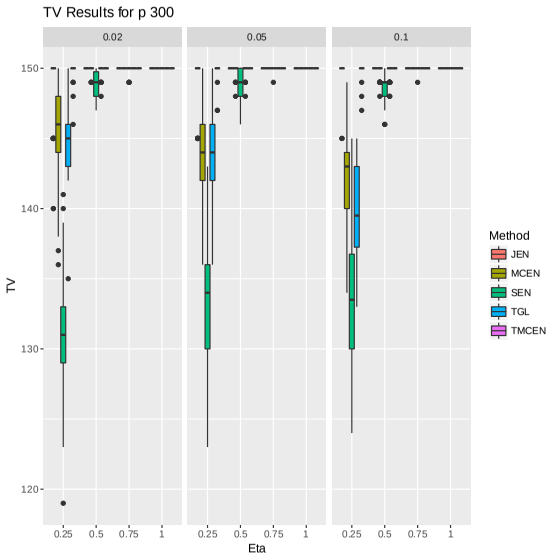


Figure 3: TV results for the Gaussian simulations with p equal to 300. Different box plots correspond to different values of λ, while x-axis values are for different values of η.

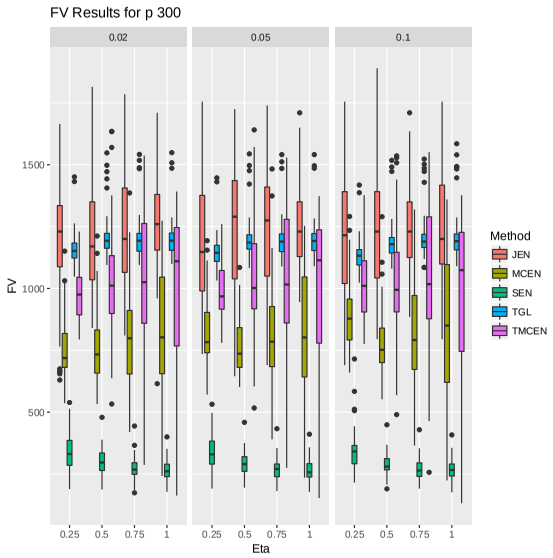


Figure 4: FV results for the Gaussian simulations with p equal to 300. Different box plots correspond to different values of λ, while x-axis values are for different values of η.

$c \in \{1, \ldots, r\}$ will be associated with its own tuning parameters of γ_c and δ_c that will be selected by maximizing the equivalent of (26) for only one response.

Define $\boldsymbol{\beta}^*_k(\eta, \lambda)$ as the kth column vector of $B^*_{\eta,\lambda}$. In all settings the kth response of the ith observation, y_{ik}, is an independent draw from $\text{Bin}(1, \pi^*_{ik})$ where

$$\pi^*_{ik} = \frac{\exp\left\{\mathbf{x}_i'\boldsymbol{\beta}^*_k(\eta,\lambda)\right\}}{1+\exp\left\{\mathbf{x}_i'\boldsymbol{\beta}^*_k(\eta,\lambda)\right\}}.$$

To evaluate the methods, 1000 validation observations are generated from the data generating model and the KL divergence is measured for each of the 50 replications. The KL divergence for a replication is defined as,

$$\frac{1}{1000}\sum_{i=1}^{1000}\sum_{k=1}^{15}\left\{\log\left(\frac{\widehat{\pi}_{ik}}{\pi^*_{ik}}\right)\widehat{\pi}_{ik} + \log\left(\frac{1-\widehat{\pi}_{ik}}{1-\pi^*_{ik}}\right)(1-\widehat{\pi}_{ik})\right\},$$

where π^*_{ik} is the true probability and $\widehat{\pi}_k(x_i, \delta, \gamma)$ is the estimated probability for response k for validation observation i.

Box plots are presented to compare the KL divergence of MCEN and SEN for the different settings in the case of $p = 300$. The results of simulation in cases where $p = 12$ and 100 are available in the supplementary material. Figure 5 presents the KL divergence results from the 50 replications for the different settings of η and λ. In terms of KL divergence MCEN outperforms SEN in all settings.

A comparison of MSE of coefficient estimates between methods is shown using box plots in Figure 6 and shows similar results to the cases of $p = 12$ and 100 available in the supplementary material. The results show that based on MSE binomial MCEN either outperforms or performs as well as binomial SEN. Figures 7 and 8 report the number of true positive and false positive predictors selected by each method for each combination of η and λ when $p = 300$, and MCEN outperforms SEN by generally selecting more true positive predictors, while the number of false predictors selected varies by the signal size. For smaller signals MCEN selects a smaller number of false predictors, but for larger signals MCEN tends to select more false predictors.

6. Data Example

6.1 Genomics Data

Votavova et al. (2011) collected gene expression profiles, demographic and birth information from 72 pregnant mothers. Using these data we modeled four response variables: placental weight, newborn weight, cotinine level from the mothers' peripheral blood sample and cotinine level from the umbilical cord blood sample. Smoking status, mother's age, mother's BMI, parity, gestational age and expression data for 24,526 gene probes from the mother's peripheral blood sample were used as covariates. Our analysis was limited to the 65 mothers with complete data. From a clinical perspective an accurate model for birth weight would be the primary interest as birth weight is associated with both short and long term negative health outcomes (Turan et al., 2012). Including placental weight as an additional response could potentially be helpful in the MCEN model because previous studies found placental and newborn weight are correlated (Molteni et al., 1978; Panti et al., 2012; Thame et al.,

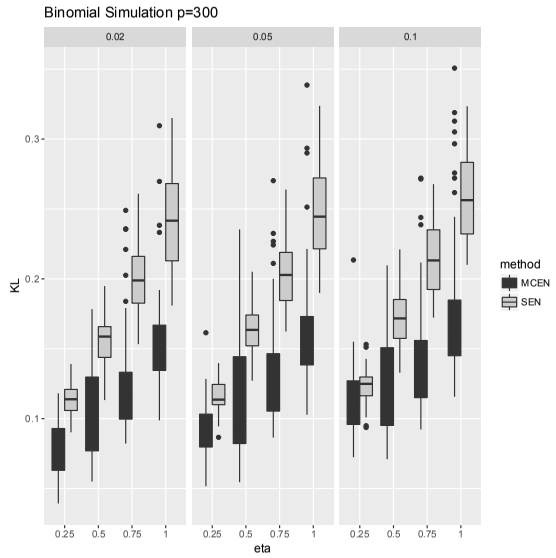


Figure 5: Simulation results comparing binomial SEN and binomial MCEN for p=300 at varying values of λ and η. Each box plot represents results for a different value of η, given at the top of the plot.

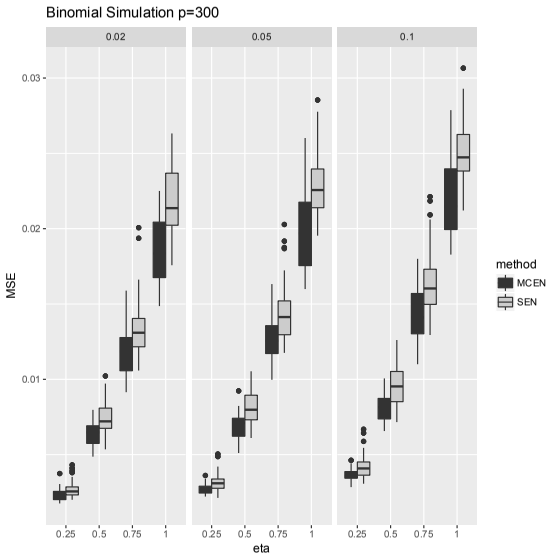


Figure 6: Simulation results comparing MSE of binomial SEN and binomial MCEN when p=300 at varying values of λ and η. Each box plot represents results for a different value of η, given at the top of the plot.

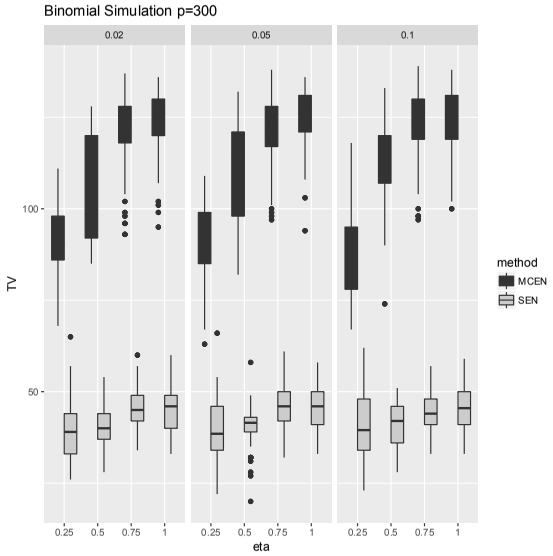


Figure 7: Simulation results comparing TV results by binomial SEN and binomial MCEN when p=300 at varying values of λ and η. Each box plot represents results for a different value of η, given at the top of the plot.

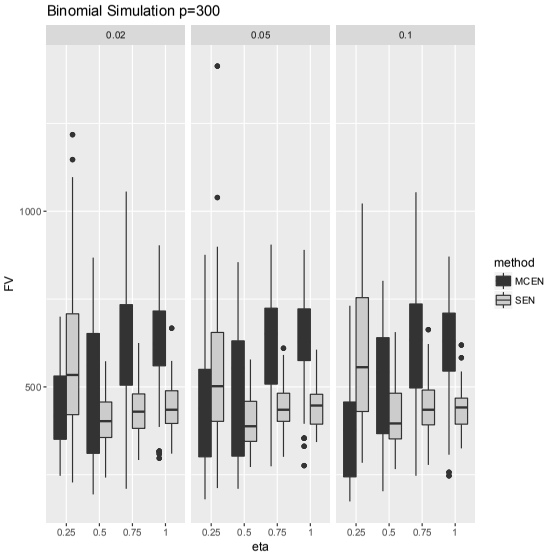


Figure 8: Simulation results comparing FV results by binomial SEN and binomial MCEN when p=300 at varying values of λ and η. Each box plot represents results for a different value of η, given at the top of the plot.

2004), but placental weight is hard to use as a predictor since it is observed at birth. The two measurements of cotinine levels are essentially measuring the same thing and are clearly related to smoking status. Thus we can test if these variables were correctly clustered and smoking status selected in the MCEN models.

The same methods used in Section 5.1 are used to fit the data, except we did not implement the TMCEN method as we did not assume to know the true clustering structure of the response variables. To evaluate the methods we randomly partitioned the data into 50 training and 15 testing samples. All four response variables are modeled on the log scale. In the training data all variables are centered and scaled to have mean zero and a standard deviation of one. We filter the gene expression data for each response by using the top 25 genes in terms of absolute value of correlation with a given response. For the joint modeling methods we use a union of the top 25 genes for each response. Models are fit using the training data, then predictions are evaluated on the testing samples. We compare the methods by looking at the ASPE, as defined in Section 5.1. For MCEN we consider clusters of size 1, 2 and 3. The process is repeated 100 times and the ASPE for all methods and responses are included in Figure 9. The MCEN method performs the best for modeling birth weight, the most clinically interesting variable, and is about the same as the other methods for modeling placenta weight. However, it does worse than the other three methods for modeling cotinine level. In all 100 random partitions the MCEN method correctly grouped the two cotinine responses together and selected smoking status as a predictor for those two responses.

6.2 Concession Data

We analyzed 2000 concession transactions from a major event venue. Each transaction is linked with the customer's information from the venue's loyalty program. These data are proprietary and cannot be made publicly available. Whether a customer purchases a specific item, 0 if they do and 1 if they do not, is the response and customer information from the loyalty program, such as seat identification and amount spent on previous concession sales, are treated as the covariates. The multiple response setting comes from there being multiple items available for sale at the concession stands. In total there are 34 predictor variables, stemming from purchase history from the venue, ticketing, and seating. The same customer may appear in the data more than once, but any correlation structure is ignored. We analyze two different sets of responses with the same covariates. The point-of-sale system records purchases in two different item set groupings; menu group (7 items) and food group (12 items). The different groups provide different insights into customer habits as the items form different groups.

Similar to the simulation section we compared SEN and MCEN, with tuning parameter selected as described in Section 5.2. For Q, the number of groups, we consider values of $\{1, 2, \ldots, 7\}$. We divide 2000 transactions into training and validation sets. There is a time component to our data, which we ignore, but use to evaluate the predictive performance of our models. The first 1000 transactions are used to train our models, with 3-fold cross validation used to select the tuning parameters for both MCEN and SEN. The predictive performance of the models are then compared using the next 1000 transactions.

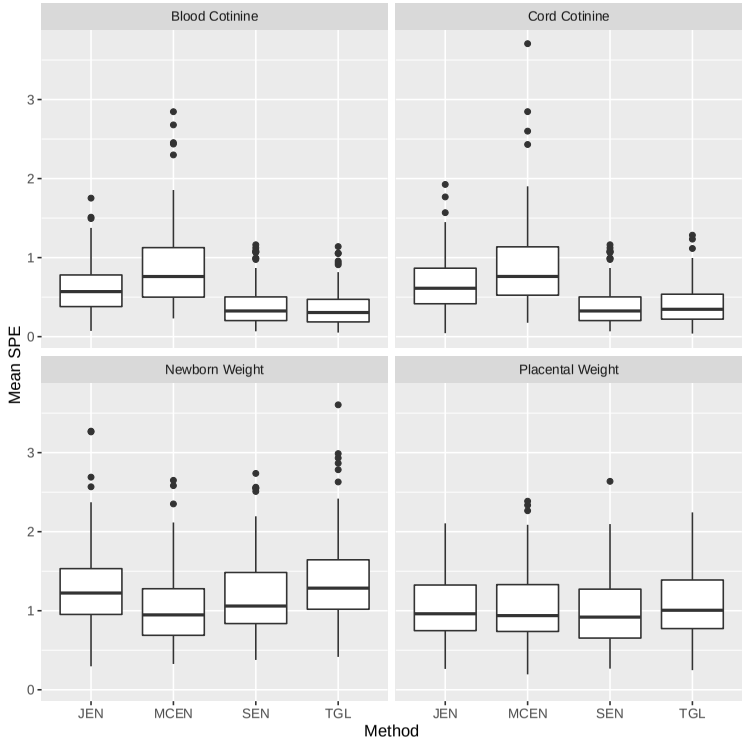


Figure 9: Mean SPE from 100 random partitions

For comparison of the methods we present the ROC curves as a metric for classification performance on the 1000 validation observations. Figure 10 presents the ROC curves and shows that in most situations the binomial logistic MCEN was competitive with SEN. In this analysis MCEN found 3 response clusters where the first cluster contained concession food, the second cluster contained both alcoholic and non-alcoholic beverages, and the third cluster contained all specialty item groups. For comparison we used k-means clustering on the predicted values of the independent elastic net, and selected the number of clusters based on the gap statistic. It selected 2 clusters. The first cluster had concession and both beverage types, while the second cluster contained all specialty items.

The resulting ROC curves for the food group analysis are presented in Figure 11. Five clusters were found by binomial logistic MCEN. The first cluster has popcorn, hamburger, french fires, bottled water, appetizers, and a chicken basket. These correspond to low selling non-alcoholic items. The second cluster consists of hot dogs, craft beer and misc sides, which represents a group of higher selling items. The last three clusters are singleton clusters consisting of non-alcoholic beverages, domestic beer, and liquor. These clusters represent high selling items with different demographics important in each. We also ran k-means clustering on the predicted values from the EN results, and found no distinct clustering using the gap statistic to select the number of clusters. Thus the MCEN method clusters all cold beverages together, while using k-means on fitted values from SEN does not find this clustering. The results of both analyses show that SEN outperforms MCEN using ROC curves. This could be due to the coarseness of MCEN framework, which assumes a similar sparsity structure for all responses. The grouping insights given from the resulting MCEN clusters provide a starting point for investigating each cluster individually with its own MCEN models. This procedure would allow for different levels of sparsity for different clusters. Flexibility such as this should be addressed in extensions of MCEN.

7. Discussion

We present a method for simultaneous estimation of regression coefficients and response clustering for a multivariate response model. The method is introduced for the case of continuous and binary responses. Future work could include extending the model to other GLM settings. Currently, our model imposes the same amount of sparsity on all response models, but this could be relaxed by allowing a sparsity tuning parameter for each individual response or each response group. The `mcen` R package that implements the methods outlined in this article is available on CRAN (Sherwood and Price, 2018).

Define $\ell(B)$ as a likelihood or convex objective function, $P(\boldsymbol{\beta}, D_q)$ as a distance function between all elements where $\sum_{q=1}^{Q} P(B, D_q)$ is an optimization problem to separate the r p-dimensional coefficient vectors into Q clusters and $p_\delta(B)$ as a penalty function with tuning parameter δ. Then the MCEN method could be generalized to a larger class of estimators where

$$(\hat{B}, \hat{D}) = \underset{B, D_1, \ldots, D_Q}{\arg\min} \; \ell(B) + \gamma \sum_{q=1}^{Q} P(B, D_q) + p_\delta(B). \tag{32}$$

One example would be to define $P(B, D_q)$ as an L_1 norm to penalize the difference between fitted values, similar to a fused lasso penalty (Tibshirani et al., 2005; Tibshirani, 2014). An

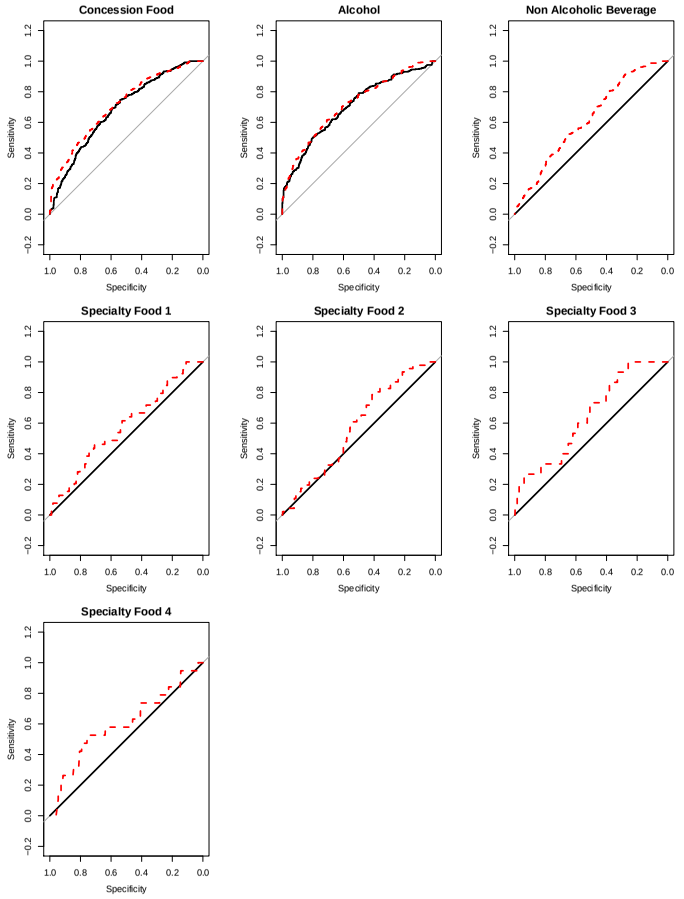


Figure 10: ROC curves for the 1000 validation observations for the menu group item responses. The black lines represent the ROC for MCEN and red for SEN.

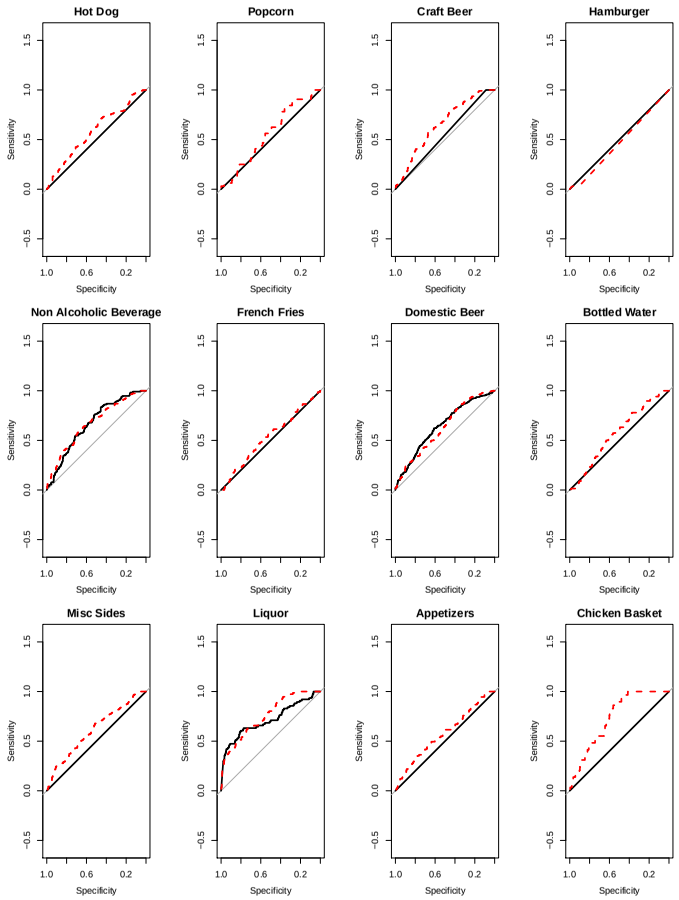


Figure 11: ROC curves for the 1000 validation observations for the food group item responses comparing EN and MCEN. The black lines represent the ROC for MCEN and red for SEN.

advantage of the estimator proposed in this paper is that by defining $P(B, D_q)$ as the L_2 norm squared, when the coefficients are fixed, the minimization problem is equivalent to a k-means problem. However, different definitions of $P(B, D_q)$ may not have well studied clustering algorithms to solve the optimization to define the groupings. One challenge of extending this work would be finding functions $P(B, D_q)$ that become well defined clustering problems when B is known or proposing new algorithms for solving $P(B, D_q)$. Otherwise the two-step algorithm proposed in this paper would not work.

The asymptotics in this paper are limited to consistency of the estimator when groups are known. Zhao and Shojaie (2016) presented an inference framework for a similar estimator that uses a fusion penalty and demonstrated that inference is still possible even if the structure of the graph that determines the fusion penalty is not correctly specified. Extending the results provided here to include inference would be of great use to practitioners and a good topic for future research.

Acknowledgments

We would like to thank Dr. Sara van de Geer and an anonymous reviewer for their insightful comments on early versions of this work. We would also like to thank Suzanna Emelio of the University of Kansas and Dr. Nancy McIntyre for their careful editing of our manuscript. This work was also partially sponsored by Big XII Faculty Fellowships from West Virginia University and the University of Kansas.

Appendix

A.1. Proof of Theorem 1

Proof Define

$$L(B) = \frac{1}{2n}\sum_{i=1}^{n}\sum_{c=1}^{r}(y_{ic} - \mathbf{x}_i^T\boldsymbol{\beta}_c)^2 + \frac{\gamma}{2n}\sum_{q=1}^{Q}\frac{1}{|D_q|}\sum_{l,m\in D_q}||X(\boldsymbol{\beta}_l - \boldsymbol{\beta}_m)||_2^2.$$

For $l \in D_q$

$$\frac{\partial}{\partial\boldsymbol{\beta}_l}L(B) = -\frac{1}{n}\left(X^TY - X^TX\boldsymbol{\beta}_l\right) + X^TX\frac{2\gamma}{n|D_q|}\sum_{c\in D_q, c\neq l}\boldsymbol{\beta}_l - \boldsymbol{\beta}_c.$$

Thus,

$$\bar{\boldsymbol{\beta}}_l\left\{1 + \frac{2\gamma(|D_q|-1)}{|D_q|}\right\} - \dot{\boldsymbol{\beta}}_l - \frac{2\gamma}{|D_q|}\sum_{c\in D_q, c\neq l}\bar{\boldsymbol{\beta}}_c = 0. \tag{33}$$

Therefore for $l, m \in D_q$

$$\begin{aligned}
&\bar{\boldsymbol{\beta}}_l\left\{1 + \frac{2\gamma(|D_q|-1)}{|D_q|}\right\} - \dot{\boldsymbol{\beta}}_l - \frac{2\gamma}{|D_q|}\sum_{c\in D_q, c\neq l}\bar{\boldsymbol{\beta}}_c \\
- \quad &\bar{\boldsymbol{\beta}}_m\left\{1 + \frac{2\gamma(|D_q|-1)}{|D_q|}\right\} - \dot{\boldsymbol{\beta}}_m - \frac{2\gamma}{|D_q|}\sum_{c\in D_q, c\neq m}\bar{\boldsymbol{\beta}}_c \\
= \quad &(\bar{\boldsymbol{\beta}}_l - \bar{\boldsymbol{\beta}}_m)(1+2\gamma) - \dot{\boldsymbol{\beta}}_l + \dot{\boldsymbol{\beta}}_m = 0.
\end{aligned}$$

Therefore for $l, m \in D_q$ and $l \neq m$

$$\bar{\boldsymbol{\beta}}_m = \bar{\boldsymbol{\beta}}_l + \frac{1}{1+2\gamma}\left(\dot{\boldsymbol{\beta}}_m - \dot{\boldsymbol{\beta}}_l\right). \tag{34}$$

Combining (33) and (34) gives

$$\begin{aligned}\bar{\boldsymbol{\beta}}_l\left\{1+\frac{2\gamma(|D_q|-1)}{|D_q|}\right\} &= \dot{\boldsymbol{\beta}}_l + \frac{2\gamma}{|D_q|}\sum_{c\in D_q, c\neq l}\bar{\boldsymbol{\beta}}_l + \frac{1}{1+2\gamma}\left(\dot{\boldsymbol{\beta}}_c - \dot{\boldsymbol{\beta}}_l\right)\\ &= \dot{\boldsymbol{\beta}}_l + \frac{2\gamma(|D_q|-1)}{|D_q|}\bar{\boldsymbol{\beta}}_l + \frac{2\gamma}{(1+2\gamma)|D_q|}\sum_{c\in D_q, c\neq l}\left(\dot{\boldsymbol{\beta}}_c - \dot{\boldsymbol{\beta}}_l\right),\end{aligned}$$

which completes the proof. ■

A.2. Proof of Theorem 2

Proof It is assumed that $E(\epsilon_{ic}^2) = 1$ and for $c \neq k$ that $E(\epsilon_{ic}\epsilon_{ik}) = \rho$. Thus, note that for any $v \in \{1, \ldots, r\}$

$$\begin{aligned}\text{Var}\left(\bar{\boldsymbol{\beta}}_v\right) &= \text{Var}\left\{\frac{|D_q|+2\gamma}{(1+2\gamma)|D_q|}\dot{\boldsymbol{\beta}}_v + \frac{2\gamma}{(1+2\gamma)|D_q|}\sum_{s\in D_q, s\neq v}\dot{\boldsymbol{\beta}}_s\right\}\\ &= (X^TX)^{-1}\left\{\frac{|D_q|\left(|D_q|+4\gamma+4\gamma^2\right)}{(1+2\gamma)^2|D_q|^2} + 4\rho\gamma(|D_q|-1)\frac{|D_q|+2\gamma|D_q|-2\gamma}{(1+2\gamma)^2|D_q|^2}\right\}.\end{aligned}$$

Define $\mathbf{b}_v = \sum_{s\in D_q, s\neq v}(\boldsymbol{\beta}_s^* - \boldsymbol{\beta}_v^*)$. The squared bias term is then

$$\begin{aligned}&E\left[\left\{E\left(\bar{\boldsymbol{\beta}}_v\right) - \boldsymbol{\beta}_v^*\right\}'\left\{E\left(\bar{\boldsymbol{\beta}}_v\right) - \boldsymbol{\beta}_v^*\right\}\right]\\ =\ &E\left[\left\{\boldsymbol{\beta}_v^* + \frac{2\gamma}{(1+2\gamma)|D_q|}\mathbf{b}_v - \boldsymbol{\beta}_v^*\right\}'\left\{\boldsymbol{\beta}_v^* + \frac{2\gamma}{(1+2\gamma)|D_q|}\mathbf{b}_v - \boldsymbol{\beta}_v^*\right\}\right]\\ =\ &\frac{4\gamma^2}{(1+2\gamma)^2|D_q|^2}||\mathbf{b}_v||_2^2.\end{aligned}$$

Let $\omega = \text{Trace}\left\{\left(X^TX\right)^{-1}\right\}$ then MSE of $\bar{\boldsymbol{\beta}}_v$ will be smaller than MSE of $\dot{\boldsymbol{\beta}}_v$ if

$$\begin{aligned}&\omega\left\{\frac{|D_q|\left(|D_q|+4\gamma+4\gamma^2\right)}{(1+2\gamma)^2|D_q|^2} + 4\rho\gamma(|D_q|-1)\frac{|D_q|+2\gamma|D_q|-2\gamma}{(1+2\gamma)^2|D_q|^2}\right\}\\ +\ &\frac{4\gamma^2}{(1+2\gamma)^2|D_q|^2}||\mathbf{b}_v||_2^2\\ <\ &\omega,\end{aligned}$$

which is equivalent to

$$\begin{aligned}\gamma||\mathbf{b}_v||_2^2 &< \omega\Big\{|D_q|(|D_q|-1) + \gamma|D_q|(|D_q|-1)\\ &\quad -\rho\left\{(|D_q|-1)|D_q| + 2\gamma(|D_q|-1)^2\right\}\Big\}.\end{aligned}$$

Note that, $\omega|D_q|(|D_q|-1)(1-\rho) > 0$ and thus if $||\mathbf{b}_v||_2^2 \leq \omega(|D_q|-1)\left\{|D_q| - 2\rho(|D_q|-1)\right\}$ then the MSE of $\bar{\boldsymbol{\beta}}_v$ is smaller than the MSE of $\dot{\boldsymbol{\beta}}_v$ for any $\gamma > 0$. Otherwise, the MSE of $\bar{\boldsymbol{\beta}}_v$ will be smaller for any $\gamma \in \left(0, \frac{\omega|D_q|(|D_q|-1)(1-\rho)}{||\mathbf{b}_v||_2^2 - \omega(|D_q|-1)\{|D_q|-2\rho(|D_q|-1)\}}\right)$. Thus for any $v \in \{1, \ldots, r\}$ then any $\gamma > 0$ or any γ sufficiently small will result in $\bar{\boldsymbol{\beta}}_v$ having a smaller MSE than $\dot{\boldsymbol{\beta}}_v$. The proof is complete because we can then find a γ sufficiently small that will result in $\bar{\boldsymbol{\beta}}_v$ having a smaller MSE than $\dot{\boldsymbol{\beta}}_v$ for all $v \in \{1, \ldots, r\}$. ■

A.3. Proof of Corollary 3

The proof of Corollary 3 is similar to the proof of Theorem 1 and only changes with respect to the expected loss rather than the observed loss.

A.4. Theorem 4

The proof of Theorem 4 will include some new definitions and an alternative formulation of (5). In our proof we use a vectorized version of many of the matrices. Let $\tilde{\mathbf{Y}} = \text{vec}(Y)$, $\tilde{\boldsymbol{\beta}} = \text{vec}(B)$, $\tilde{\boldsymbol{\beta}}' = \text{vec}(\acute{B})$ and $\tilde{\mathbf{E}} = \text{vec}(E)$. Define $\mathbf{A}_{m,s} \in \mathcal{R}^r$, where $(m,s) \in D_q$, with $\sqrt{\frac{1}{|D_q|}}$ in the mth element, $-\sqrt{\frac{1}{|D_q|}}$ in the sth element and 0 in all other elements, $A_{D_q} \in \mathcal{R}^{|D_q|(|D_q|-1)\times r}$ as the matrix with row vectors $\mathbf{A}_{m,s}$ where $(m,s) \in D_q$, and $A_D \equiv \left(A_{D_1}^T, \ldots, A_{D_Q}^T\right)^T \in \mathcal{R}^{\sum_{q=1}^Q |D_q|(|D_q|-1)\times r}$.

Then the objective function from (5) can be restated as

$$\begin{aligned}&\frac{1}{2n}\left[\tilde{\boldsymbol{\beta}}^T\left\{\tilde{X}^T\tilde{X} + \gamma(A_D\otimes X)^T(A_D\otimes X)\right\}\tilde{\boldsymbol{\beta}} - 2\tilde{\mathbf{Y}}^T\tilde{X}\tilde{\boldsymbol{\beta}}\right] + \delta||\tilde{\boldsymbol{\beta}}||_1\\ =\ &\ell(\tilde{\boldsymbol{\beta}}) + \delta g(\tilde{\boldsymbol{\beta}}).\end{aligned}$$

In addition define, $\tilde{\ell}(\boldsymbol{\Delta}, \tilde{\boldsymbol{\beta}}) \equiv \ell(\tilde{\boldsymbol{\beta}} + \boldsymbol{\Delta}) - \ell(\tilde{\boldsymbol{\beta}}) - \langle\nabla\ell(\tilde{\boldsymbol{\beta}}, \Delta)\rangle$.

First, we will present some lemmas that are helpful in proving Theorem 4. A general outline of the proof for Theorem 4 is by using the triangle inequality we have $||\text{vec}(\bar{B} - B*)||_2 \leq ||\text{vec}(\bar{B} - \acute{B})||_2 + ||\tilde{\boldsymbol{\beta}}' - \tilde{\boldsymbol{\beta}}^*||_2$. Completing the proof is done by establishing upper bounds for $||\text{vec}(\bar{B} - \acute{B})||_2$ and $||\tilde{\boldsymbol{\beta}}' - \tilde{\boldsymbol{\beta}}^*||_2$. Much of the proof will require working with $\tilde{\boldsymbol{\beta}}'$ and we introduce the following notation to easily relate $\tilde{\boldsymbol{\beta}}'$ and $\tilde{\boldsymbol{\beta}}^*$. For response l in group q define $\mathbf{H}_l = \frac{1}{\sqrt{|D_q|}}\sum_{c\in D_q, c\neq l}\mathbf{A}_{c,l}$ where $\mathbf{H}_l \in \mathcal{R}^r$ and $H = (\mathbf{H}_1, \ldots, \mathbf{H}_r)^T \in \mathcal{R}^{r\times r}$. Then we have

$$\tilde{\boldsymbol{\beta}}' = \left\{\left(I_r + \frac{2\gamma}{2\gamma+1}H\right)\otimes I_p\right\}\tilde{\boldsymbol{\beta}}^*.$$

For response l is in group q define $\mathbf{U}_l = \frac{1}{|D_q|}\sum_{k\in D_q}(\beta_k^* - \beta_l^*)$ where $\mathbf{U}_l \in \mathcal{R}^p$ and $U = (\mathbf{U}_1, \ldots, \mathbf{U}_r)$ with $U \in \mathcal{R}^{p\times r}$ and $\tilde{\mathbf{U}} = \text{vec}(U) \in \mathcal{R}^{pr}$, then

$$\left|\left|\text{vec}\left(\acute{B} - B^*\right)\right|\right|_2 = \frac{2\gamma}{1+2\gamma}\left|\left|(H\otimes I_p)\tilde{\boldsymbol{\beta}}^*\right|\right|_2 = \frac{2\gamma}{1+2\gamma}\left|\left|\tilde{\mathbf{U}}\right|\right|_2.$$

Lemma 5 *Under assumption A3*

$$\tilde{\ell}(\boldsymbol{\Delta}, \tilde{\boldsymbol{\beta}}') \geq \kappa||\boldsymbol{\Delta}||_2^2 \text{ for all } \boldsymbol{\Delta} \in \mathcal{C}.$$

Proof From the definition of $\tilde{\ell}(\boldsymbol{\Delta}, \tilde{\boldsymbol{\beta}})$, assumption A3 and that $\boldsymbol{\Delta} \in \mathcal{C}$ it follows that

$$\begin{aligned}\tilde{\ell}(\boldsymbol{\Delta}, \tilde{\boldsymbol{\beta}}') &= \frac{1}{2n}\boldsymbol{\Delta}^T\left\{\tilde{X}^T\tilde{X} + \gamma(A_D\otimes X)^T(A_D\otimes X)\right\}\boldsymbol{\Delta} \\ &\geq \frac{1}{2n}\boldsymbol{\Delta}^T\tilde{X}^T\tilde{X}\boldsymbol{\Delta} \\ &\geq \frac{\kappa}{2}||\boldsymbol{\Delta}||_2^2.\end{aligned}$$

■

For any vector $\mathbf{a} = (a_1, \ldots, a_{pr})^T \in \mathcal{R}^{pr}$ we define the $||\mathbf{a}||_\infty$ as the L_∞ norm of $\mathbf{a}$, that is $||\mathbf{a}||_\infty = \max_i |a_i|$.

Lemma 6 *For $\bar{B}$ from (5), under assumptions A1-A4 with $\delta \geq 2\left|\left|\nabla\ell(\tilde{\boldsymbol{\beta}}')\right|\right|_\infty$ then there exists a positive constant c_3 such that*

$$\left|\left|vec(\bar{B} - \acute{B})\right|\right|_2^2 \leq 9\frac{\delta^2}{\kappa^2}s.$$

Proof Define the set $\acute{S} = \{j \in \{1, \ldots, rp\}, \tilde{\boldsymbol{\beta}}_j' \neq 0\}$. By assumption A5 and Corollary 3 $\acute{S} = S$, that is $\tilde{\boldsymbol{\beta}}_j' = 0$ if and only if $\tilde{\boldsymbol{\beta}}_j^* = 0$. Define $\psi(\mathcal{M}) \equiv \sup_{\mathbf{u}\in\mathcal{M}\setminus\{\mathbf{0}\}}\frac{||\mathbf{u}||_1}{||\mathbf{u}||_2}$. Note that $\psi\{\mathcal{M}(S)\} = \sqrt{s}$. Also, note that the dual norm of the L_1 norm is the L_∞ norm. Results follow from Theorem 1 of Negahban et al. (2012) and Lemma 5. ■

Lemma 7 *Under the conditions of Theorem 4 there exists positive c_1, c_2 and c_3 such that*

$$\left|\left|vec\left(\bar{B} - \acute{B}\right)\right|\right|_2 \leq \frac{48\sigma}{\kappa}\sqrt{\frac{s\log(rp)}{n}},$$

with probability at least $1 - c_1 exp(-c_2 n\delta^2)$.

Proof If we can find positive constants c_1 and c_2 such that with probability at least $1 - c_1\exp(-c_2 n\delta^2)$ that $\delta \geq 2\left|\left|\nabla\ell(\tilde{\boldsymbol{\beta}}')\right|\right|_\infty$ then proof will be complete by Lemma 6 and by

the condition that $\delta = 16\sigma\sqrt{\frac{\log(rp)}{n}}$. Note that

$$\begin{aligned}2\left|\left|\nabla\ell(\tilde{\boldsymbol{\beta}}')\right|\right|_\infty &= 2\left|\left|\frac{1}{n}\left[\left\{\tilde{X}^T\tilde{X} + \gamma(A_D\otimes X)^T(A_D\otimes X)\right\}\tilde{\boldsymbol{\beta}}' - \tilde{X}^T\tilde{Y}\right]\right|\right|_\infty \\ &= 2\left|\left|\frac{1}{n}\left[\left\{\tilde{X}^T\tilde{X} + \gamma(A_D\otimes X)^T(A_D\otimes X)\right\}\left\{\left(I_r + \frac{2\gamma}{2\gamma+1}H\right)\otimes I_p\right\}\tilde{\boldsymbol{\beta}}^* - \tilde{X}^T\left(\tilde{X}\tilde{\boldsymbol{\beta}}^* + \tilde{\mathbf{E}}\right)\right]\right|\right| \\ &\leq 2\left|\left|\frac{2\gamma}{n(1+2\gamma)}\tilde{X}^T\tilde{X}\tilde{\mathbf{U}}\right|\right|_\infty + 2\left|\left|\frac{\gamma}{n}(A_D\otimes X)^T(A_D\otimes X)\tilde{\boldsymbol{\beta}}^*\right|\right|_\infty \\ &\quad + 2\left|\left|\frac{2\gamma^2}{n(1+2\gamma)}(A_D\otimes X)^T(A_D\otimes X)\tilde{\mathbf{U}}\right|\right|_\infty + 2\left|\left|\frac{1}{n}\tilde{X}^T\tilde{\mathbf{E}}\right|\right|_\infty.\end{aligned}$$

Next, we will establish upper bounds for the first three terms. Define $I(l \in D_q)$ to be 1 if $l \in D_q$ and zero otherwise. Using the definition of $\tilde{\mathbf{U}}$ and assumptions A4-A6,

$$\begin{aligned}2\left|\left|\frac{2\gamma}{n(1+2\gamma)}\tilde{X}^T\tilde{X}\tilde{\mathbf{U}}\right|\right|_\infty &= \frac{4\gamma}{1+2\gamma}\max_{l\in\{1,\ldots,r\}}\left|\left|\frac{1}{n}X^TX\sum_{q=1}^{Q}I(l\in D_q)\sum_{k\in D_q}\frac{1}{|D_q|}(\boldsymbol{\beta}_k^* - \boldsymbol{\beta}_l^*)\right|\right|_\infty \\ &\leq \frac{4\gamma}{1+2\gamma}\rho_{\max}\max_{l\in\{1,\ldots,r\}}\left|\left|\sum_{q=1}^{Q}I(l\in D_q)\sum_{k\in D_q}\frac{1}{|D_q|}(\boldsymbol{\beta}_k^* - \boldsymbol{\beta}_l^*)\right|\right|_2 \\ &\leq \frac{4\gamma}{1+2\gamma}\rho_{\max}\acute{b}.\end{aligned}$$

Using assumptions A4-A6,

$$\begin{aligned}2\left|\left|\frac{\gamma}{n}(A_D\otimes X)^T(A_D\otimes X)\tilde{\boldsymbol{\beta}}^*\right|\right|_\infty &= 2\gamma\max_{l\in\{1,\ldots,r\}}\left|\left|\frac{1}{n}X^TX\sum_{k,l\in D_q,k\neq l}\frac{1}{|D_q|}(\beta_k^* - \beta_l^*)\right|\right|_\infty \\ &\leq 2\gamma\rho_{\max}\max_{l\in\{1,\ldots,r\}}\left|\left|\sum_{k,l\in D_q}\frac{1}{|D_q|}(\beta_k^* - \beta_l^*)\right|\right|_2 \\ &\leq 2\gamma\rho_{\max}\acute{b}.\end{aligned}$$

Note that for $a \in D_q$ and $b \in D_q$ that

$$\begin{aligned}\mathbf{U}_a - \mathbf{U}_b &= \frac{1}{|D_q|}\left(\sum_{l\in D_q}\boldsymbol{\beta}_l^* - \boldsymbol{\beta}_a^* - \sum_{l\in D_q}\boldsymbol{\beta}_l^* - \boldsymbol{\beta}_b^*\right) \\ &= \frac{1}{|D_q|}\sum_{l\in D_q}\boldsymbol{\beta}_b^* - \boldsymbol{\beta}_a^* = \boldsymbol{\beta}_b^* - \boldsymbol{\beta}_a^*.\end{aligned}$$

Therefore

$$\begin{aligned}
2\left\|\frac{2\gamma^2}{n(1+2\gamma)}(A_D\otimes X)^T(A_D\otimes X)\tilde{\mathbf{U}}\right\|_\infty &= \frac{4\gamma^2}{1+2\gamma}\max_{l\in\{1,\ldots,r\}}\left\|\frac{1}{n}X^TX\sum_{q=1}^{Q}I(l\in D_q)\sum_{k\in D_q}\frac{1}{|D_q|}(\mathbf{U}_k-\mathbf{U}_l)\right\|_\infty \\
&= \frac{4\gamma^2}{1+2\gamma}\max_{l\in\{1,\ldots,r\}}\left\|\frac{1}{n}X^TX\sum_{q=1}^{Q}I(l\in D_q)\sum_{k\in D_q}\frac{1}{|D_q|}\boldsymbol{\beta}_l^*-\boldsymbol{\beta}_k^*\right\|_\infty \\
&\leq \frac{4\gamma^2}{1+2\gamma}\rho_{\max}\max_{l\in\{1,\ldots,r\}}\left\|\sum_{q=1}^{Q}I(l\in D_q)\sum_{k\in D_q}\frac{1}{|D_q|}\boldsymbol{\beta}_l^*-\boldsymbol{\beta}_k^*\right\|_2 \\
&\leq \frac{4\gamma^2}{1+2\gamma}\rho_{\max}\acute{b} \\
&\leq 2\gamma\rho_{\max}\acute{b}.
\end{aligned}$$

Under assumptions A1 and A2 it follows that

$$P\left(\left\|\frac{1}{n}\tilde{X}^T\tilde{\mathbf{E}}\right\|_\infty > t\right)\leq 2\exp\left\{\frac{-nt^2}{2\sigma^2}+\log(rp)\right\}. \tag{35}$$

Thus,

$$\begin{aligned}
P\left\{\delta\geq 2\left\|\nabla\ell(\tilde{\boldsymbol{\beta}}')\right\|_\infty\right\} &\geq P\left\{\delta\geq 2\left\|\frac{1}{n}\tilde{X}^T\tilde{\mathbf{E}}\right\|_\infty+\rho_{\max}\acute{b}\left(\frac{4\gamma}{1+2\gamma}+2\gamma+2\gamma\right)\right\} \\
&\geq P\left(\delta\geq 2\left\|\frac{1}{n}\tilde{X}^T\tilde{\mathbf{E}}\right\|_\infty+8\gamma\rho_{\max}\acute{b}\right) \\
&\geq P\left(\frac{3}{16}\delta\geq\left\|\frac{1}{n}\tilde{X}^T\tilde{\mathbf{E}}\right\|_\infty\right) \\
&\geq 1-2\exp\left\{\frac{-9n\delta^2}{16^2 2\sigma^2}+\log(rp)\right\} \\
&= 1-2\exp\left\{-\frac{7}{2}\log(rp)\right\}.
\end{aligned}$$

Set $c_1=2$ and $c_2=\frac{7}{2}$ and the proof is complete. ■

Proof of Theorem 4

Proof Applying the triangle inequality we have

$$\left\|\text{vec}\left(\hat{\beta}-\beta^*\right)\right\|_2\leq\left\|\text{vec}\left(\hat{\beta}-\acute{\beta}\right)\right\|_2+\left\|\tilde{\boldsymbol{\beta}}'-\tilde{\boldsymbol{\beta}}^*\right\|_2. \tag{36}$$

For the second term using the upper bound for γ stated in the conditions for Theorem 4 and assumptions A4 and A5 it follows that

$$\begin{aligned}
\left\|\tilde{\boldsymbol{\beta}}'-\tilde{\boldsymbol{\beta}}^*\right\|_2 &= \frac{2\gamma}{1+2\gamma}\left\|\tilde{\mathbf{U}}\right\|_2 \\
&\leq 2\gamma\sqrt{s}\acute{b}\leq\frac{5\sigma}{2\rho_{\max}}\sqrt{\frac{s\log(rp)}{n}}.
\end{aligned}$$

Combining the above inequality with (36) and Lemma 7 it follows that there exists positive constants c_1 and c_2 such that

$$\left|\left|\text{vec}\left(\bar{B}-B^*\right)\right|\right|_2\leq\frac{48\sigma}{\kappa}\sqrt{\frac{s\log(rp)}{n}}+\frac{5\sigma}{2\rho_{\max}}\sqrt{\frac{s\log(rp)}{n}},$$

with probability at least $1-c_1\exp(-c_2n\delta^2)$. To complete the proof set $c_3=48$ and $c_4=\frac{5}{2}$. ■

References

Peter J. Bickel, Ya'acov Ritov, and Alexandre B. Tsybakov. Simultaneous analysis of lasso and dantzig selector. *Ann. Statist.*, 37(4):1705–1732, 2009.

Leo Breiman and Jerome H. Friedman. Predicting multivariate responses in multiple linear regression. *J. R. Statist. Soc. B*, 59(1):3–54, 1997.

Peter Bühlmann, Philipp Rütimann, Sara van de Greer, and Cun-Hui Zhang. Correlated variables in regression: Clustering and sparse estimation. *J. Statist. Planng Inf*, 143(11): 1835–1858, 2013.

Emmanuel Candes and Terence Tao. The dantzig selector: statistical estimation when p is much larger than n. *Ann. Statist.*, 35(6):2313–2351, 2007.

Yupeng Chen, Raghuram Iyengar, and Garud Iyengar. Modeling multimodal continuous heterogeneity in conjoint analysis—a sparse learning approach. *Marketing Science*, 36(1): 140–156, 2016.

R. Dennis Cook and Xin Zhang. Foundations for envelope models and methods. *J. Am. Statist. Ass*, 110(510):599–611, 2015.

R. Dennis Cook, Bing Li, and Francesca Chiaromonte. Envelope models for parsimonious and efficient multivariate linear regression (with discussion). *Statistica Sinica*, 20:927–1010, 2010.

Julian J. Faraway. *Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models*. CRC Press, 2006.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularized paths for genearlized linear models via coordinate descent. *Journal of Statistial Softwawre*, 33(1), 2008.

Holger Hoefling. A path algorithm for the fused lasso signal approximator. *Journal of Computational and Graphical Statistics*, 19(4):984–1006, 2010.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Jian Huang, Shuangge, Hongzhe Li, and Cun-Hui Zhang. The sparse laplacian shrinkage estimator for high-dimensional regression. *Annals of Statistics*, 39(4), 2011.

Ozge Yucel Kasap, Nevzat Ekmekci, and Utku Gorkem Ketenci. Combining logisitc regression analysis and association rule mining via mlr algorithm. In *ICSEA 2016 The Eleventh International Conference on Software Engineering Advances*, pages 154–159. IARA, 2016.

Seyoung Kim and Eric P. Xing. Tree-guided group lasso for multi-response regressin with structured sparsity with an application to eqtl mapping. *The Annals fo Applied Statistics*, 6(3):1095–1117, 2012.

Seyoung Kim, Kyung-Ah Sohn, and Eric P. Xing. A multivariate regression approach to association analysis of a quantitative trait network. *Bioinformatics*, 25(12):i204–i212, 2009.

Wonyul Lee and Yufeng Liu. Simultaneous multiple response regression and inverse covariance matrix estimation via penalized gaussian maximum likelihood. *Journal of Multivariate Analysis*, 111:241–255, 2012.

Caiyan Li and Hongzhe Li. Network-constrained regularization and variable selection of genomic data. *Bioinformatics*, 24(9):1175–1182, 2008.

Caiyan Li and Hongzhe Li. Variable selection and regression analysis for graph-structure covariates with an application to geneomics. *The Annals of Applied Statistics*, 4(3): 1498–1516, 2010.

Nicolai Meinshausen and Bin Yu. Lasso-type recovery of sparse representations for high-dimensional data. *Ann. Statist.*, 37(1):246–270, 2009.

Aaron J. Molstad and Adam J. Rothman. Indirect multivariate response linear regression. *Biometrika*, 3(103):595–607, 2016.

R. A. Molteni, S. J. Stys, and F. C. Battaglia. Relationship of fetal and placental weight in human beings: fetal/placental weight ratios at various gestational ages and birth weight distributions. *The Journal of Reproductive Medicine*, 21:327–334, 1978.

Sahand N. Negahban, Pradeep Ravikumar, Martin J. Wainwright, and Bin Yu. A unified framework for high-dimensional analysis fo m-estimators with decomposable regualrizers. *Statistical Science*, 27(4):538–557, 2012.

Abubakar A. Panti, Bissala A. Ekele, Emmanuel I. Nwobodo, and Ahmed Yakubu. The relationship between the weight of the placenta and birth weight of the neonate in a nigerian hospital. *Nigerian Medical Journal*, 53(2):80–84, 2012.

Jie Peng, Ji Zhu, Anna Beramaschi, Wonshik Han, Dong-Young Noh, Jonathan R. Pollac, and Pei Wang. Regularized multivariate regression for identifying master predictors with application to integrative genomics study of breast cancer. *Annals of Applied Statistics*, 4(1):53–77, 2010.

Bradley S. Price, Charles J. Geyer, and Adam J. Rothman. Automatic response category combination in multinomial logistic regression. https://arxiv.org/abs/1705.03594, May 2017.

Piyush Rai, Abhishek Kumar, and Hal Daume. Simultaneously leveraging output and task structures for multiple-output regression. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 3185–3193. Curran Associates, Inc., 2012. URL `http://papers.nips.cc/paper/4501-simultaneously-leveraging-output-and-task-structures-for-multiple-output-regression.pdf`.

Alessandro Rinaldo. Properties and refinements of the fused lasso. *Ann. Statist.*, 37(5B): 2597–3097, 2009.

Adam J. Rothman, Elizaveta Levina, and Ji Zhu. Sparse multivariate regression with covariance estimation. *Journal of Computational and Graphical Statistics*, 19(4):947–962, 2010.

Ben Sherwood and Bradley S. Price. *mcen: Multivariate Cluster Elastic Net*, 2018. R package version 1.0.

Qiang Sun, Hongtu Zhu, Yufeng Liu, and Joseph G. Ibrahim. Sprem: Sparse projection regression model for high-dimensional linear regression. *J. Am. Statist. Ass*, 110(509): 289–302, 2015.

M. Thame, C. Osmond, F. Bennett, R. Wilks, and T. Forrester. Fetal growth is directly related to maternal anthropometry and placental volume. *European Journal of Clinical Nutrition*, 58:894–900, 2004.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Statist. Soc. B*, 58(1):267–288, 1996.

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *J. R. Statist. Soc. B*, pages 91–108, 2005.

Ryan J. Tibshirani. Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics*, 42(1):285–323, 2014.

Nahid Turan, Mohamed F. Ghalwash, Sunita Katari, Christos Coutifaris, Zoran Obradovic, and Carmen Sapienza. Dna methylation differences at growth related genes correlate with birth weight: a molecular signature linked to developmental origins of adult disease? *BMC Medical Genomics*, 5(10):1–21, 2012.

Sara van de Geer and Peter Bühlmann. On the conditions used to prove oracle results for the lasso. *Electronic Journal of Statistics*, 3:1360–1392, 2009.

Hana Votavova, Michaela Dostalova Merkerova, Kamilia Fejglova, A. Vasikova, Zdenek Krejcik, A. Pastorkova, N. Tabashidze, J. Topinka, M. Veleminsky Jr., R.J. Sram, and R. Brdicka. Transcriptome alterations in maternal and fetal cells induced by tobacco smoke. *Placenta*, 32:763–770, 2011.

Daniela M. Witten, Ali Shojaie, and Fan Zhang. The cluster elastic net for high-dimensional regression with unknown variable grouping. *Technometrics*, 56(1):112–122, 2014.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *J. R. Statist. Soc. B*, 68(1):49–67, 2005.

Sen Zhao and Ali Shojaie. A significance test for graph constrained estimation. *Biometrics*, 72(2):484–493, 2016.

Weiqiang Zhou, Ben Sherwood, Zhicheng Ji, Yingchao Xue, Fang Du, Jiawei Bai, Mingyao Ying, and Hongkai Ji. Genome-wide prediction of dnase i hypersensitivity using gene expression. *Nature Communications*, 8:1–17, 2017.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *J. R. Statist. Soc. B*, 67:301–320, 2005.

Journal of Machine Learning Research 18 (2018) 1-29 Submitted 8/17; Revised 6/18; Published 7/18

Characteristic and Universal Tensor Product Kernels

Zoltán Szabó zoltan.szabo@polytechnique.edu
CMAP, École Polytechnique
Route de Saclay, 91128 Palaiseau, France

Bharath K. Sriperumbudur bks18@psu.edu
Department of Statistics
Pennsylvania State University
314 Thomas Building
University Park, PA 16802

Editor: Francis Bach

Abstract

Maximum mean discrepancy (MMD), also called energy distance or N-distance in statistics and Hilbert-Schmidt independence criterion (HSIC), specifically distance covariance in statistics, are among the most popular and successful approaches to quantify the difference and independence of random variables, respectively. Thanks to their kernel-based foundations, MMD and HSIC are applicable on a wide variety of domains. Despite their tremendous success, quite little is known about when HSIC characterizes independence and when MMD with tensor product kernel can discriminate probability distributions. In this paper, we answer these questions by studying various notions of characteristic property of the tensor product kernel.

Keywords: tensor product kernel, kernel mean embedding, characteristic kernel, $\mathcal{I}$-characteristic kernel, universality, maximum mean discrepancy, Hilbert-Schmidt independence criterion

1. Introduction

Kernel methods (Schölkopf and Smola, 2002) are among the most flexible and influential tools in machine learning and statistics, with superior performance demonstrated in a large number of areas and applications. The key idea in these methods is to map the data samples into a possibly infinite-dimensional feature space—precisely, a reproducing kernel Hilbert space (RKHS; Aronszajn, 1950)—and apply linear methods in the feature space, without the explicit need to compute the map. A generalization of this idea to probability measures, i.e., mapping probability measures into an RKHS (Berlinet and Thomas-Agnan, 2004, Chapter 4; Smola et al., 2007) has found novel applications in nonparametric statistics and machine learning. Formally, given a probability measure $\mathbb{P}$ defined on a measurable space $\mathcal{X}$ and an RKHS $\mathcal{H}_k$ with $k : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ as the reproducing kernel (which is symmetric and positive definite), $\mathbb{P}$ is embedded into $\mathcal{H}_k$ as

$$\mathbb{P} \mapsto \int_{\mathcal{X}} k(\cdot, x)\, \mathrm{d}\mathbb{P}(x) =: \mu_k(\mathbb{P}), \tag{1}$$

where $\mu_k(\mathbb{P})$ is called the *mean element* or *kernel mean embedding* of $\mathbb{P}$. The *mean embedding* of $\mathbb{P}$ has lead to a new generation of solutions in two-sample testing (Baringhaus and Franz, 2004; Székely and Rizzo, 2004, 2005; Borgwardt et al., 2006; Harchaoui et al., 2007; Gretton et al., 2012), goodness-of-fit testing (Chwialkowski et al., 2016; Liu et al., 2016; Jitkrittum et al., 2017b; Balasubramanian et al., 2017), domain adaptation (Zhang et al., 2013) and generalization (Blanchard et al., 2017), kernel belief propagation (Song et al., 2011), kernel Bayes' rule (Fukumizu et al., 2013), model criticism (Lloyd et al., 2014; Kim et al., 2016), approximate Bayesian computation (Park et al., 2016), probabilistic programming (Schölkopf et al., 2015), distribution classification (Muandet et al., 2011; Zaheer et al., 2017), distribution regression (Szabó et al., 2016; Law et al., 2018) and topological data analysis (Kusano et al., 2016). A recent survey on the topic is provided by Muandet et al. (2017).

Crucial to the success of the mean embedding based representation is whether it encodes all the information about the distribution, in other words whether the map in (1) is injective in which case the kernel is referred to as *characteristic* (Fukumizu et al., 2008; Sriperumbudur et al., 2010). Various characterizations for the characteristic property of k is known in the literature (Fukumizu et al., 2008, 2009; Sriperumbudur et al., 2010; Gretton et al., 2012) using which the popular kernels on $\mathbb{R}^d$ such as Gaussian, Laplacian, B-spline, inverse multiquadrics, and the Matérn class are shown to be characteristic. The characteristic property is closely related to the notion of *universality* (Steinwart, 2001; Micchelli et al., 2006; Carmeli et al., 2010; Sriperumbudur et al., 2011)—k is said to be universal if the corresponding RKHS $\mathcal{H}_k$ is dense in a certain target function class, for example, the class of continuous functions on compact domains—and the relation between these notions has recently been explored by Sriperumbudur et al. (2011); Simon-Gabriel and Schölkopf (2016).

Based on the mean embedding in (1), Smola et al. (2007) and Gretton et al. (2012) defined a semi-metric, called the maximum mean discrepancy (MMD) on the space of probability measures:

$$\mathrm{MMD}_k(\mathbb{P}, \mathbb{Q}) := \|\mu_k(\mathbb{P}) - \mu_k(\mathbb{Q})\|_{\mathcal{H}_k},$$

which is a metric iff k is characteristic. A fundamental application of MMD is in nonparametric hypothesis testing that includes two-sample (Gretton et al., 2012) and independence tests (Gretton et al., 2008). Particularly in independence testing, as a measure of independence, MMD measures the distance between the joint distribution $\mathbb{P}_{XY}$ and the product of marginals $\mathbb{P}_X \otimes \mathbb{P}_Y$ of two random variables X and Y which are respectively defined on measurable spaces $\mathcal{X}$ and $\mathcal{Y}$, with the kernel k being defined on $\mathcal{X}\times\mathcal{Y}$. As aforementioned, if k is characteristic, then $\mathrm{MMD}_k(\mathbb{P}_{XY}, \mathbb{P}_X \otimes \mathbb{P}_Y) = 0$ implies $\mathbb{P}_{XY} = \mathbb{P}_X \otimes \mathbb{P}_Y$, i.e., X and Y are independent. A simple way to define a kernel on $\mathcal{X}\times\mathcal{Y}$ is through the tensor product of kernels k_X and k_Y defined on $\mathcal{X}$ and $\mathcal{Y}$ respectively: $k = k_X \otimes k_Y$, i.e., $k\left((x, y), (x', y')\right) = k_X(x, x')k_Y(y, y')$, $x, x' \in \mathcal{X}$, $y, y' \in \mathcal{Y}$, with the corresponding RKHS $\mathcal{H}_k = \mathcal{H}_{k_X} \otimes \mathcal{H}_{k_Y}$ being the tensor product space generated by $\mathcal{H}_{k_X}$ and $\mathcal{H}_{k_Y}$. This means, when $k = k_X \otimes k_Y$,

$$\mathrm{MMD}_k(\mathbb{P}_{XY}, \mathbb{P}_X \otimes \mathbb{P}_Y) = \|\mu_{k_X\otimes k_Y}(\mathbb{P}_{XY}) - \mu_{k_X\otimes k_Y}(\mathbb{P}_X \otimes \mathbb{P}_Y)\|_{\mathcal{H}_{k_X}\otimes\mathcal{H}_{k_Y}}. \tag{2}$$

In addition to the simplicity of defining a joint kernel k on $\mathcal{X}\times\mathcal{Y}$, the tensor product kernel offers a principled way of combining inner products (k_X and k_Y) on domains that can

correspond to different modalities (say images, texts, audio). By exploiting the isomorphism between tensor product Hilbert spaces and the space of Hilbert-Schmidt operators[1], it follows from (2) that

$$\mathrm{MMD}_k(\mathbb{P}_{XY}, \mathbb{P}_X \otimes \mathbb{P}_Y) = \|C_{XY}\|_{\mathrm{HS}} =: \mathrm{HSIC}_k(\mathbb{P}_{XY}), \tag{3}$$

which is the Hilbert-Schmidt norm of the cross-covariance operator $C_{XY} := \mu_{k_X \otimes k_Y}(\mathbb{P}_{XY}) - \mu_{k_X}(\mathbb{P}_X) \otimes \mu_{k_Y}(\mathbb{P}_Y)$ and is known as the *Hilbert-Schmidt independence criterion* (HSIC) (Gretton et al., 2005a). HSIC has enjoyed tremendous success in a variety of applications such as independent component analysis (Gretton et al., 2005a), feature selection (Song et al., 2012), independence testing (Gretton et al., 2008; Jitkrittum et al., 2017a), post selection inference (Yamada et al., 2018) and causal detection (Mooij et al., 2016; Pfister et al., 2017; Strobl et al., 2017). Recently, MMD and HSIC (as defined in (3) for two components) have been shown by Sejdinovic et al. (2013b) to be equivalent to other popular statistical measures such as the energy distance (Baringhaus and Franz, 2004; Székely and Rizzo, 2004, 2005)—also known as N-distance (Zinger et al., 1992; Klebanov, 2005)—and distance covariance (Székely et al., 2007; Székely and Rizzo, 2009; Lyons, 2013) respectively. HSIC has been generalized to $M \geq 2$ components (Quadrianto et al., 2009; Sejdinovic et al., 2013a) to measure the joint independence of M random variables

$$\mathrm{HSIC}_k\left(\mathbb{P}\right) = \left\| \mu_{\otimes_{m=1}^M k_m}(\mathbb{P}) - \otimes_{m=1}^M \mu_{k_m}\left(\mathbb{P}_m\right) \right\|_{\otimes_{m=1}^M \mathcal{H}_{k_m}}, \tag{4}$$

where $\mathbb{P}$ is a joint measure on the product space $\mathcal{X} := \times_{m=1}^M \mathcal{X}_m$ and $(\mathbb{P}_m)_{m=1}^M$ are the marginal measures of $\mathbb{P}$ defined on $(\mathcal{X}_m)_{m=1}^M$ respectively. The extended HSIC measure has recently been analyzed in the context of independence testing (Pfister et al., 2017). In addition to testing, the extended HSIC measure is also useful in the problem of independent subspace analysis (ISA; Cardoso, 1998), wherein the latent sources are separated by maximizing the degree of independence among them. In all the applications of HSIC, the key requirement is that $k = \otimes_{m=1}^M k_m$ captures the joint independence of M random variables (with joint distribution $\mathbb{P}$)—we call this property as $\mathcal{I}$-characteristic—, which is guaranteed if k is characteristic. Since k is defined in terms of $(k_m)_{m=1}^M$, it is of fundamental importance to understand the characteristic and $\mathcal{I}$-characteristic properties of k in terms of the characteristic property of $(k_m)_{m=1}^M$, which is one of the main goals of this work.

For $M = 2$, the characterization of independence, i.e., the $\mathcal{I}$-characteristic property of k, is studied by Blanchard et al. (2011) and Gretton (2015) where it has been shown that if k_1 and k_2 are universal, then k is universal[2] and therefore HSIC captures independence. A stronger version of this result can be obtained by combining (Lyons, 2013, Theorem 3.11) and (Sejdinovic et al., 2013b, Proposition 29): if k_1 and k_2 are characteristic, then the HSIC associated with $k = k_1 \otimes k_2$ characterizes independence. Apart from these results, not much is known about the characteristic/$\mathcal{I}$-characteristic/universality properties of k in terms of the individual kernels. Our **goal** is to resolve this question and understand the characteristic, $\mathcal{I}$-characteristic and universal property of the product kernel $(\otimes_{m=1}^M k_m)$ in terms of the kernel components $((k_m)_{m=1}^M)$ for $M \geq 2$. Because of the relatedness of MMD and HSIC to energy distance and distance covariance, our results also contribute to the better understanding of these other measures that are popular in the statistical literature.

Specifically, our results shed light on the following **surprising phenomena** of the $\mathcal{I}$-characteristic property of $\otimes_{m=1}^M k_m$ for $M \geq 3$:

1. characteristic property of $(k_m)_{m=1}^M$ is not sufficient but necessary for $\otimes_{m=1}^M k_m$ to be $\mathcal{I}$-characteristic;
2. universality of $(k_m)_{m=1}^M$ is sufficient for $\otimes_{m=1}^M k_m$ to be $\mathcal{I}$-characteristic, and
3. if at least one of $(k_m)_{m=1}^M$ is only characteristic and not universal, then $\otimes_{m=1}^M k_m$ need not be $\mathcal{I}$-characteristic.

The paper is organized as follows. In Section 3, we conduct a comprehensive analysis about the above mentioned properties of k and $(k_m)_{m=1}^M$ for any positive integer M. To this end, we define various notions of characteristic property on the product space $\mathcal{X}$ (see Definition 1 and Figure 2(a) in Section 3) and explore the relation between them. In order to keep our presentation in this section to be non-technical, we relegate the problem formulation to Section 3, with the main results of the paper being presented in Section 4. A summary of the results is captured in Figure 1 while the proofs are provided in Section 5. Various definitions and notation that are used throughout the paper are collected in Section 2.

2. Definitions and Notation

$\mathbb{N} := \{1, 2, \ldots\}$ and $\mathbb{R}$ denotes the set of natural numbers and real numbers respectively. For $M \in \mathbb{N}$, $[M] := \{1, \ldots, M\}$. $\mathbf{1}_d := (1, 1, \ldots, 1) \in \mathbb{R}^d$ and $\mathbf{0}$ denotes the matrix of zeros. For $a := (a_1, \ldots, a_d) \in \mathbb{R}^d$ and $b := (b_1, \ldots, b_d) \in \mathbb{R}^d$, $\langle a, b\rangle = \sum_{i=1}^d a_i b_i$ is the Euclidean inner product. For sets A and B, $A \backslash B = \{a \in A : a \notin B\}$ is their difference, $|A|$ is the cardinality of A and $\times_{m=1}^M A_m = \{(a_1, \ldots, a_M) : a_m \in A_m\, m \in [M]\}$ is the Descartes product of sets $(A_m)_{m=1}^M$. $\mathcal{P}(\mathcal{X})$ denotes the power set of a set $\mathcal{X}$, i.e., all subsets of $\mathcal{X}$ (including the empty set and $\mathcal{X}$). The Kronecker delta is defined as $\delta_{a,b} = 1$ if $a = b$, and zero otherwise. χ_A is the indicator function of set A: $\chi_A(x) = 1$ if $x \in A$ and $\chi_A(x) = 0$ otherwise. $\mathbb{R}^{d_1 \times \ldots \times d_M}$ is the set of $d_1 \times \ldots \times d_M$-sized tensors.

For a topological space $(\mathcal{X}, \tau_{\mathcal{X}})$, $\mathcal{B}(\mathcal{X}) := \mathcal{B}(\tau_{\mathcal{X}})$ is the Borel sigma-algebra on $\mathcal{X}$ induced by the topology $\tau_{\mathcal{X}}$. Probability and finite signed measures in the paper are meant w.r.t. the measurable space $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$. Given $\{(\mathcal{X}_i, \tau_i)\}_{i \in I}$ topological spaces, their product $\times_{i \in I} \mathcal{X}_i$ is enriched with the product topology; it is the coarsest topology for which the canonical projections $\pi_i : \times_{i \in I} \mathcal{X}_i \to (\mathcal{X}_i, \tau_i)$ are continuous for all $i \in I$. A topological space $(\mathcal{X}, \tau_{\mathcal{X}})$ is called second-countable if $\tau_{\mathcal{X}}$ has a countable basis.[3] $C(\mathcal{X})$ denotes the space of continuous functions on $\mathcal{X}$. $C_0(\mathcal{X})$ denotes the class of real-valued functions vanishing at infinity on a locally compact Hausdorff (LCH) space[4] $\mathcal{X}$, i.e., for any $\epsilon > 0$, the set $\{x \in \mathcal{X} : |f(x)| \geq \epsilon\}$

1. In the equivalence one assumes that $\mathcal{H}_{k_X}$, $\mathcal{H}_{k_Y}$ are separable; this holds under mild conditions, for example if $\mathcal{X}$ and $\mathcal{Y}$ are separable topological domains and k_X, k_Y are continuous (Steinwart and Christmann, 2008, Lemma 4.33).
2. Blanchard et al. (2011) deal with c-universal kernels while Gretton (2015) deals with c_0-universal kernels. A brief description of these notions are given in Section 3. Carmeli et al. (2010); Sriperumbudur et al. (2010) provide further details on these notions of universality.
3. Second-countability implies separability; in metric spaces the two notions coincide (Dudley, 2004, Proposition 2.1.4). By the Urysohn's theorem, a topological space is separable and metrizable if and only if it is regular, Hausdorff and second-countable. Any uncountable discrete space is *not* second-countable.
4. LCH spaces include $\mathbb{R}^d$, discrete spaces, and topological manifolds. Open or closed subsets, finite products of LCH spaces are LCH. Infinite-dimensional Hilbert spaces are *not* LCH.

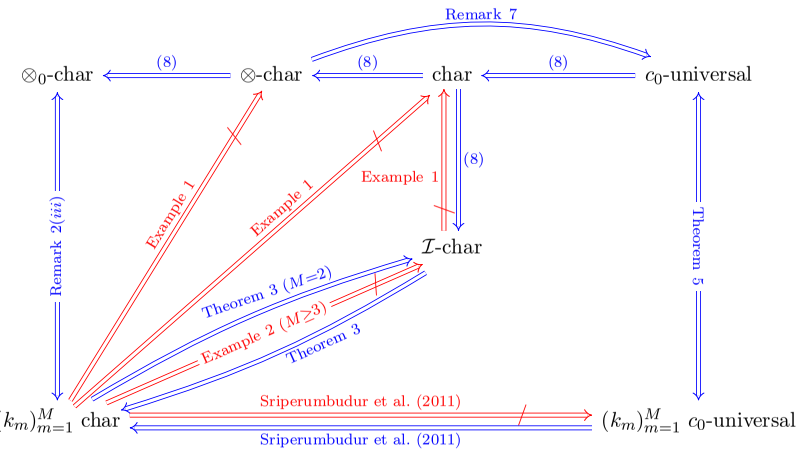


Figure 1: Summary of results: "char" denotes characteristic. In addition to the usual characteristic property, three new notions $\otimes_0$-characteristic, $\otimes$-characteristic and $\mathcal{I}$-characteristic are introduced in Definition 1 which along with c_0-universal (in the top right corner) correspond to the property of the tensor product kernel $\otimes_{m=1}^M k_m$, while the bottom part of the picture corresponds to the individual kernels $(k_m)_{m=1}^M$ being characteristic or c_0-universal. If $(k_m)_{m=1}^M$-s are continuous, bounded and translation invariant kernels on $\mathbb{R}^{d_m}$, $m \in [M]$, all the notions are equivalent (see Theorem 4).

is compact. $C_0(\mathcal{X})$ is endowed with the uniform norm $\|f\|_\infty = \sup_{x\in\mathcal{X}} |f(x)|$. $\mathcal{M}_b(\mathcal{X})$ and $\mathcal{M}_1^+(\mathcal{X})$ are the space of finite signed measures and probability measures on $\mathcal{X}$, respectively. For $\mathbb{P}_m \in \mathcal{M}_1^+(\mathcal{X}_m)$, $\otimes_{m=1}^M \mathbb{P}_m$ denotes the product probability measure on the product space $\times_{m=1}^M \mathcal{X}_m$, i.e., $\otimes_{m=1}^M \mathbb{P}_m \in \mathcal{M}_1^+(\times_{m=1}^M \mathcal{X}_m)$. δ_x is the Dirac measure supported on $x \in \mathcal{X}$. For $\mathbb{F} \in \mathcal{M}_b\left(\times_{m=1}^M \mathcal{X}_m\right)$, the finite signed measure $\mathbb{F}_m$ denotes its marginal on $\mathcal{X}_m$. $\mathcal{H}_{k_m}$ is the reproducing kernel Hilbert space (RKHS) associated with the reproducing kernel $k_m : \mathcal{X}_m \times \mathcal{X}_m \to \mathbb{R}$, which in this paper is assumed to be measurable and bounded. The tensor product of $(k_m)_{m=1}^M$ is a kernel, defined as

$$\otimes_{m=1}^M k_m\left((x_1,\ldots,x_M),(x'_1,\ldots,x'_M)\right) = \prod_{m=1}^M k_m\left(x_m,x'_m\right), \quad x_m,x'_m \in \mathcal{X}_m,$$

whose associated RKHS is denoted as $\mathcal{H}_{\otimes_{m=1}^M k_m} = \otimes_{m=1}^M \mathcal{H}_{k_m}$ (Berlinet and Thomas-Agnan, 2004, Theorem 13), where the r.h.s. is the tensor product of RKHSs $(\mathcal{H}_{k_m})_{m=1}^M$. For $h_m \in \mathcal{H}_m$, $m \in [M]$, the multi-linear operator $\otimes_{m=1}^M h_m \in \otimes_{m=1}^M \mathcal{H}_m$ is defined as

$$\left(\otimes_{m=1}^M h_m\right)(v_1,\ldots,v_M) = \prod_{m=1}^M \langle h_m, v_m\rangle_{\mathcal{H}_m}, \quad v_m \in \mathcal{H}_m.$$

A kernel $k : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ defined on a LCH space $\mathcal{X}$ is called a c_0-kernel if $k(\cdot,x) \in C_0(\mathcal{X})$ for all $x \in \mathcal{X}$. $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is said to be a translation invariant kernel on $\mathbb{R}^d$ if $k(x,y) = \psi(x-y)$, $x,y \in \mathbb{R}^d$ for a positive definite function $\psi : \mathbb{R}^d \to \mathbb{R}$. $\mu_k(\mathbb{F})$ denotes the kernel mean embedding of $\mathbb{F} \in \mathcal{M}_b(\mathcal{X})$ to $\mathcal{H}_k$ which is defined as $\mu_k(\mathbb{F}) = \int_{\mathcal{X}} k(\cdot,x)\,\mathrm{d}\mathbb{F}(x)$, where the integral is meant in the Bochner sense.

3. Problem Formulation

In this section, we formally introduce the goal of the paper. To this end, we start with a definition. For simplicity, throughout the paper, we assume that all kernels are bounded. The definition is based on the observation (Sriperumbudur et al., 2010, Lemma 8) that a bounded kernel k on a topological space $(\mathcal{X},\tau_{\mathcal{X}})$ is characteristic if and only if

$$\int_{\mathcal{X}}\int_{\mathcal{X}} k(x,x')\,\mathrm{d}\mathbb{F}(x)\,\mathrm{d}\mathbb{F}(x') > 0, \ \forall\, \mathbb{F} \in \mathcal{M}_b(\mathcal{X})\backslash\{0\} \text{ such that } \mathbb{F}(\mathcal{X}) = 0.$$

In other words, characteristic kernels are integrally strictly positive definite (ispd; see Sriperumbudur et al., 2010, p. 1523) w.r.t. the class of finite signed measures that assign zero measure to $\mathcal{X}$. The following definition extends this observation to tensor product kernels on product spaces.

Definition 1 **($\mathcal{F}$-ispd tensor product kernel)** *Suppose $k_m : \mathcal{X}_m \times \mathcal{X}_m \to \mathbb{R}$ is a bounded kernel on a topological space $(\mathcal{X}_m,\tau_{\mathcal{X}_m})$, $m \in [M]$. Let $\mathcal{F} \subseteq \mathcal{M}_b(\mathcal{X})$ be such that $0 \in \mathcal{F}$ where $\mathcal{X} := \times_{m=1}^M \mathcal{X}_m$. $k := \otimes_{m=1}^M k_m$ is said to be* $\mathcal{F}$-ispd *if*

$$\mu_k(\mathbb{F}) = 0 \Rightarrow \mathbb{F} = 0 \quad (\mathbb{F} \in \mathcal{F}), \text{ or equivalently}$$

$$\|\mu_k(\mathbb{F})\|_{\mathcal{H}_k}^2 = \int_{\times_{m=1}^M \mathcal{X}_m}\int_{\times_{m=1}^M \mathcal{X}_m} \left(\otimes_{m=1}^M k_m\right)(x,x')\ \mathrm{d}\mathbb{F}(x)\,\mathrm{d}\mathbb{F}(x') > 0, \quad \forall\, \mathbb{F} \in \mathcal{F}\backslash\{0\}. \quad (5)$$

Specifically,

- *if k_m-s are c_0-kernels on locally compact Polish (LCP)* [5] *spaces $\mathcal{X}_m$-s and $\mathcal{F} = \mathcal{M}_b(\mathcal{X})$, then k is called c_0-universal.*
- *if*

$$\begin{aligned}
\mathcal{F} &= [\mathcal{M}_b(\mathcal{X})]^0 && := \{\mathbb{F} \in \mathcal{M}_b(\mathcal{X}) : \mathbb{F}(\mathcal{X}) = 0\},\\
\mathcal{F} &= \left[\otimes_{m=1}^M \mathcal{M}_b(\mathcal{X}_m)\right]^0 && := \left\{\mathbb{F} \in \otimes_{m=1}^M \mathcal{M}_b\left(\mathcal{X}_m\right), \mathbb{F}(\mathcal{X}) = 0\right\},\\
\mathcal{F} &= \mathcal{I} && := \left\{\mathbb{P} - \otimes_{m=1}^M \mathbb{P}_m : \mathbb{P} \in \mathcal{M}_1^+\left(\times_{m=1}^M \mathcal{X}_m\right)\right\}, \quad (M \geq 2)\\
\mathcal{F} &= \otimes_{m=1}^M \mathcal{M}_b^0(\mathcal{X}_m) && := \left\{\mathbb{F} = \otimes_{m=1}^M \mathbb{F}_m \ : \mathbb{F}_m \in \mathcal{M}_b\left(\mathcal{X}_m\right), \mathbb{F}_m(\mathcal{X}_m) = 0, \ \forall\, m \in [M]\right\},
\end{aligned}$$

then k is called characteristic, $\otimes$-characteristic, $\mathcal{I}$-characteristic *and* $\otimes_0$-characteristic, *respectively.*

5. A topological space is called Polish if it is complete, separable and metrizable. For example, $\mathbb{R}^d$ and countable discrete spaces are Polish. Open and closed subsets, products and disjoint unions of countably many Polish spaces are Polish. Every second-countable LCH space is Polish.

In Definition 1, k being characteristic matches the usual notion of characteristic kernels on a product space, i.e., there are no two distinct probability measures on $\mathcal{X} = \times_{m=1}^{M} \mathcal{X}_m$ such that the MMD between them is zero. The other notions such as $\otimes$-characteristic, $\mathcal{I}$-characteristic and $\otimes_0$-characteristic are typically weaker than the usual characteristic property since

$$\otimes_{m=1}^{M} \mathcal{M}_b^0(\mathcal{X}_m) \quad \subseteq \left[\otimes_{m=1}^{M} \mathcal{M}_b(\mathcal{X}_m)\right]^0 \quad \subseteq \quad \left[\mathcal{M}_b\left(\times_{m=1}^{M} \mathcal{X}_m\right)\right]^0 \subseteq \quad \mathcal{M}_b\left(\times_{m=1}^{M} \mathcal{X}_m\right). \qquad (6)$$

$$\cup\!| \quad \mathcal{I}$$

Below we provide further intuition on the $\mathcal{F}$ measure classes enlisted in Definition 1.

Remark 2 *(i)* $\boldsymbol{\mathcal{F} = \mathcal{M}_b(\mathcal{X})}$ **:** *If k_m-s are c_0-kernels on LCH spaces $\mathcal{X}_m$ for all $m \in [M]$, then k is also a c_0-kernel on LCH space $\mathcal{X}$ implying that if k satisfies* (5)*, then k is c_0-universal (Sriperumbudur et al., 2010, Proposition 2). It is well known (Sriperumbudur et al., 2010) that c_0-universality reduces to c-universality (i.e., the notion of universality proposed by Steinwart, 2001) if $\mathcal{X}$ is compact which is guaranteed if and only if each $\mathcal{X}_m$, $m \in [M]$ is compact.*

(ii) $\boldsymbol{\mathcal{F} = \mathcal{I}}$ **:** *This family is useful to describe the joint* independence *of M random variables—hence the name $\mathcal{I}$-characteristic—defined on kernel-endowed domains $(\mathcal{X}_m)_{m=1}^M$: If $\mathbb{P}$ denotes the joint distribution of random variables $(X_m)_{m=1}^M$ and $(\mathbb{P}_m)_{m=1}^M$ are the associated marginals on $(\mathcal{X}_m)_{m=1}^M$, then by definition $k = \otimes_{m=1}^M k_m$ is $\mathcal{I}$-characteristic iff*

$$\mathrm{HSIC}_k(\mathbb{P}) = 0 \Longleftrightarrow \mathbb{P} = \otimes_{m=1}^{M} \mathbb{P}_m.$$

In other words, HSIC captures joint independence exactly with $\mathcal{I}$-characteristic kernels. Similarly, the $\mathcal{I}$-characteristic property ensures that COCO (constrained covariance; Gretton et al., 2005b) is a joint independence measure as COCO is defined by replacing the Hilbert-Schmidt norm of the cross-covariance operator (see (3) *and* (4)*) with its spectral norm.*

(iii) $\boldsymbol{\mathcal{F} = \otimes_{m=1}^{M} \mathcal{M}_b^0(\mathcal{X}_m)}$ **:** *In this case $\mathcal{F}$ is chosen to be the product of finite signed measures on $\mathcal{X}$ such that each marginal measure $\mathbb{F}_m$ assigns zero to the corresponding space $\mathcal{X}_m$. This choice is relevant as the characteristic property of individual kernels $(k_m)_{m=1}^M$ need* not *imply the characteristic property of $\otimes_{m=1}^M k_m$, but is equivalent to the $\otimes_0$-characteristic property of $\otimes_{m=1}^M k_m$. The equivalence holds for bounded kernels $k_m : \mathcal{X}_m \times \mathcal{X}_m \to \mathbb{R}$ on topological spaces $\mathcal{X}_m$ ($m \in [M]$) since for any $\mathbb{F} = \otimes_{m=1}^M \mathbb{F}_m \in \otimes_{m=1}^M \mathcal{M}_b(\mathcal{X}_m)$, $\mathbb{F}_m(\mathcal{X}_m) = 0$ $(\forall\, m \in [M])$*

$$\|\mu_k(\mathbb{F})\|^2_{\mathcal{H}_{\otimes_{m=1}^M k_m}} = \prod_{m=1}^{M} \|\mu_{k_m}(\mathbb{F}_m)\|^2_{\mathcal{H}_{k_m}}, \qquad (7)$$

and the l.h.s. is positive iff each term on the r.h.s. is positive.

(iv) $\boldsymbol{\mathcal{F} = \left[\otimes_{m=1}^{M} \mathcal{M}_b(\mathcal{X}_m)\right]^0}$ **:** *This class is similar to the one discussed in (iii) above—i.e., class of product measures—with the slight difference that the joint measure $\mathbb{F}$ is restricted to assign zero measure to $\mathcal{X}$ without requiring all the marginal measures $\mathbb{F}_m$ to assign zero measure to the corresponding space $\mathcal{X}_m$. While the need for considering such a measure class may not be clear at this juncture, however, based on* (7)*, it turns out that this choice of $\mathcal{F}$ has quite surprising connections to the characteristic property and c_0-universality of the product kernel; for details see Remark 7.*

(v) ***$\mathcal{F}$-ispd relations:*** *Given the relations in* (6)*, it immediately follows that $k = \otimes_{m=1}^M k_m$ satisfies*

$$\otimes_0\text{-characteristic} \Longleftarrow \otimes\text{-characteristic} \Longleftarrow \begin{array}{c} \text{characteristic} \\ \Downarrow \\ \mathcal{I}\text{-characteristic} \end{array} \Longleftarrow c_0\text{-universal} \qquad (8)$$

when $\mathcal{X}_m$ for all $m \in [M]$ are LCP. A visual illustration of (6) *and* (8) *is provided in Figure 2.*

(vi) $\boldsymbol{\left[\otimes_{m=1}^{M} \mathcal{M}_b(\mathcal{X}_m)\right]^0 \cap \mathcal{I} = \{0\}}$ **:** *While it is clear that $\left[\otimes_{m=1}^M \mathcal{M}_b(\mathcal{X}_m)\right]^0$ and $\mathcal{I}$ are subsets of $\left[\mathcal{M}_b(\times_{m=1}^M \mathcal{X}_m)\right]^0$, it is interesting to note that $\left[\otimes_{m=1}^M \mathcal{M}_b(\mathcal{X}_m)\right]^0$ and $\mathcal{I}$ have a trivial intersection with 0 being the measure common to each of them, assuming that $\mathcal{X}_m$-s are second-countable for all $m \in [M]$; see Section 5.1.*

Having defined the $\mathcal{F}$-ispd property, our **goal** is to investigate whether the characteristic or c_0-universal property of k_m-s ($m \in [M]$) imply different $\mathcal{F}$-ispd properties of $\otimes_{m=1}^M k_m$, and vice versa.

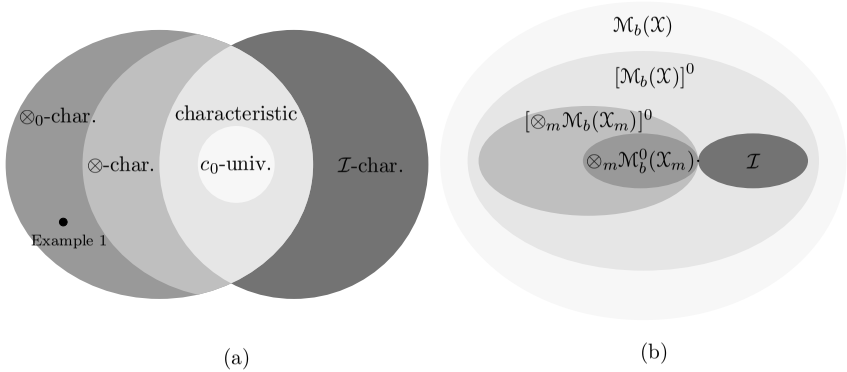


Figure 2: (a) $\mathcal{F}$-ispd $\otimes_{m=1}^M k_m$ kernels (see (8)); (b) $\mathcal{F} \subseteq \mathcal{M}_b(\mathcal{X})$, $\mathcal{X} = \times_{m=1}^M \mathcal{X}_m$. Example 1: $\otimes_{m=1}^M k_m$ is $\otimes_0$-characteristic but not $\otimes$-characteristic and therefore not characteristic.

4. Main Results

In this section, we present our main results related to the $\mathcal{F}$-ispd property of tensor product kernels, which are summarized in Figure 1. The results in this section will deal with various assumptions on $\mathcal{X}_m$, such as second-countability, Hausdorff, locally compact Hausdorff

(LCH) and locally compact Polish (LCP), so that they are presented in more generality. However, for simplicity, all these assumptions can be unified by simply assuming a stronger condition that $\mathcal{X}_m$'s are LCP.

Our first example illustrates that the characteristic property of k_m-s does not imply the characteristic property of the tensor product kernel. In light of Remark 2(iv) of Section 3, it follows that the class of $\otimes_0$-characteristic tensor product kernels form a *strictly* larger class than characteristic tensor product kernels; see also Figure 2.

Example 1 *Let $\mathcal{X}_1 = \mathcal{X}_2 = \{1,2\}$, $\tau_{\mathcal{X}_1} = \tau_{\mathcal{X}_2} = \mathcal{P}(\{1,2\})$, $k_1(x,x') = k_2(x,x') = 2\delta_{x,x'} - 1$. It is easy to verify that k_1 and k_2 are characteristic. However, it can be proved that $k_1 \otimes k_2$ is* not *$\otimes$-characteristic and therefore not characteristic. On the hand, interestingly, $k_1 \otimes k_2$ is $\mathcal{I}$-characteristic. We refer the reader to Section 5.2 for details.*

In the above example, we showed that the tensor product of k_1 and k_2 (which are characteristic kernels) is $\mathcal{I}$-characteristic. The following result generalizes this behavior for any bounded characteristic kernels. In addition, under a mild assumption, it shows the converse to be true for any M.

Theorem 3 *Let $k_m : \mathcal{X}_m \times \mathcal{X}_m \to \mathbb{R}$ be bounded kernels on topological spaces $\mathcal{X}_m$ for all $m \in [M]$, $M \geq 2$. Then the following holds.*

(i) Suppose $\mathcal{X}_m$ is second-countable for all $m \in [M]$ with $M = 2$. If k_1 and k_2 are characteristic, then $k_1 \otimes k_2$ is $\mathcal{I}$-characteristic.

(ii) Suppose $\mathcal{X}_m$ is Hausdorff and $|\mathcal{X}_m| \geq 2$ for all $m \in [M]$. If $\otimes_{m=1}^M k_m$ is $\mathcal{I}$-characteristic, then $k_1, \ldots, k_M$ are characteristic.

Lyons (2013) has showed an analogous result to Theorem 3(i) for distance covariances ($M = 2$) on metric spaces of negative type (Lyons, 2013, Theorem 3.11), which by Sejdinovic et al. (2013b, Proposition 29) holds for HSIC yielding the $\mathcal{I}$-characteristic property of $k_1 \otimes k_2$. Recently, Gretton (2015) presented a direct proof showing that HSIC corresponding to $k_1 \otimes k_2$ captures independence if k_1 and k_2 are translation invariant characteristic kernels on $\mathbb{R}^d$ (which is equivalent to c_0-universality). Blanchard et al. (2011) proved a result similar to Theorem 3(i) assuming that $\mathcal{X}_m$'s are compact and k_1, k_2 being c-universal. In contrast, Theorem 3(i) establishes the result for bounded kernels on general second-countable topological spaces. In fact, the results of Gretton (2015); Blanchard et al. (2011) are special cases of Theorems 4 and 5 below. Theorem 3(i) raises a pertinent question: whether $\otimes_{m=1}^M k_m$ is $\mathcal{I}$-characteristic if k_m-s are characteristic for all $m \in [M]$ where $M > 2$? The following example provides a negative answer to this question. On a positive side, however, we will see in Theorem 5 that the $\mathcal{I}$-characteristic property of $\otimes_{m=1}^M k_m$ can be guaranteed for any $M \geq 2$ if a stronger condition is imposed on k_m-s (and $\mathcal{X}_m$-s). Theorem 3(ii) generalizes Proposition 3.15 of Lyons (2013) for any $M > 2$, which states that every kernel k_m, $m \in [M]$ being characteristic is necessary for the tensor kernel $\otimes_{m=1}^M k_m$ to be $\mathcal{I}$-characteristic.

Example 2 *Let $M = 3$ and $\mathcal{X}_m := \{1,2\}$, $\tau_{\mathcal{X}_m} = \mathcal{P}(\mathcal{X}_m)$, $k_m(x,x') = 2\delta_{x,x'} - 1$ ($m = 1,2,3$). As mentioned in Example 1, $(k_m)_{m=1}^3$ are characteristic. However, it can be shown that $\otimes_{m=1}^3 k_m$ is not $\mathcal{I}$-characteristic. See Section 5.4 for details.*

In Remark 2(iii) and Example 1, we showed that in general, only the $\otimes_0$-characteristic property of $\otimes_{m=1}^M k_m$ is equivalent to the characteristic property of k_m-s. Our next result shows that all the various notions of characteristic property of $\otimes_{m=1}^M k_m$ *coincide* if k_m-s are translation-invariant, continuous bounded kernels on $\mathbb{R}^d$.

Theorem 4 *Suppose $k_m : \mathbb{R}^{d_m} \times \mathbb{R}^{d_m} \to \mathbb{R}$ are continuous, bounded and translation-invariant kernels for all $m \in [M]$. Then the following statements are equivalent:*

(i) k_m-s are characteristic for all $m \in [M]$;
(ii) $\otimes_{m=1}^M k_m$ is $\otimes_0$-characteristic;
(iii) $\otimes_{m=1}^M k_m$ is $\otimes$-characteristic;
(iv) $\otimes_{m=1}^M k_m$ is $\mathcal{I}$-characteristic;
(v) $\otimes_{m=1}^M k_m$ is characteristic.

The following result shows that on LCP spaces, the tensor product of $M \geq 2$ c_0-universal kernels is also c_0-universal, and vice versa.

Theorem 5 *Suppose $k_m : \mathcal{X}_m \times \mathcal{X}_m \to \mathbb{R}$ are c_0-kernels on LCP spaces $\mathcal{X}_m$ ($m \in [M]$). Then $\otimes_{m=1}^M k_m$ is c_0-universal iff k_m-s are c_0-universal for all $m \in [M]$.*

Remark 6 *(i) A special case of Theorem 5 for $M = 2$ is proved by Lyons (2013, Lemma 3.8) in the context of distance covariance which reduces to Theorem 5 through the equivalence established by Sejdinovic et al. (2013b). Another special case of Theorem 5 is proved by Blanchard et al. (2011, Lemma 5.2) for c-universality with $M = 2$ using the Stone-Weierstrass theorem: if k_1 and k_2 are c-universal then $k_1 \otimes k_2$ is c-universal.*

(ii) Since the notions of c_0-universality and characteristic property are equivalent for translation invariant c_0-kernels on $\mathbb{R}^d$ (Carmeli et al., 2010, Prop. 5.16, Sriperumbudur et al., 2010, Theorem 9), Theorem 4 can be considered as a special case of Theorem 5. In other words, requiring $(k_m)_{m=1}^M$ to be also c_0-kernels in Theorem 4(i)-(iv) is equivalent to

(v) k_m-s are c_0-universal for all $m \in [M]$;
(vi) $\otimes_{m=1}^M k_m$ is c_0-universal.

(iii) Since the c_0-universality of $\otimes_{m=1}^M k_m$ implies its $\mathcal{I}$-characteristic property (see (8)), Theorem 5 also provides a generalization of Theorem 3(i) to $M \geq 2$ under additional assumptions on k_m-s, while constraining $\mathcal{X}_m$-s to LCP-s instead of second-countable topological spaces.

In Example 2 and Theorem 5, we showed that for $M \geq 3$ components while the characteristic property of $(k_m)_{m=1}^M$ is not sufficient, their universality is enough to guarantee the $\mathcal{I}$-characteristic property of $\otimes_{m=1}^M k_m$. The next example demonstrates that these results are tight: If at least one k_m is not universal but only characteristic, then $\otimes_{m=1}^M k_m$ might not be $\mathcal{I}$-characteristic.

Example 3 *Let $M = 3$ and $\mathcal{X}_m := \{1,2\}$, $\tau_{\mathcal{X}_m} = \mathcal{P}(\mathcal{X}_m)$, for all $m \in [3]$, $k_1(x,x') = 2\delta_{x,x'} - 1$, and $k_m(x,x') = \delta_{x,x'}$ ($m = 2,3$). k_1 is characteristic (Example 1), k_2 and k_3 are universal since the associated Gram matrix $\mathbf{G} = [k_m(x,x')]_{x,x' \in \mathcal{X}_m}$ is an identity matrix,*

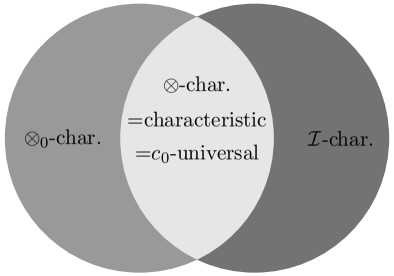


Figure 3: Simplification of the $\mathcal{F}$-ispd property of tensor product kernels; see Remark 7.

which is strictly positive definite ($m = 2, 3$). However, $\otimes_{m=1}^3 k_m$ is not *$\mathcal{I}$-characteristic. See Section 5.7 for details.*

Remark 7 *Note that the l.h.s. in* (7) *is positive if and only if each term on the r.h.s. is positive, i.e., if $k = \otimes_{m=1}^M k_m$ is $\otimes$-characteristic with k_m-s being c_0-kernels on LCP $\mathcal{X}_m$-s, then all k_m-s are c_0-universal. A similar result was also proved by Steinwart and Ziegel (2017, Lemma 3.4). Combining this with Theorem 5 yields that for tensor product c_0-kernels, the notions of $\otimes$-characteristic, characteristic and c_0-universality are equivalent, which is quite surprising as for a joint kernel k (that is not of product type), these notions need not necessarily coincide. In light of this discussion, Figure 2(a) can be simplified to Figure 3.*

5. Proofs

In this section, we provide the proofs of our results presented in Section 4.

5.1 Proof of Remark 2(iv)

By the second-countability of $\mathcal{X}_m$-s, $\mathcal{B}\left(\times_{m=1}^M \mathcal{X}_m\right) = \otimes_{m=1}^M \mathcal{B}(\mathcal{X}_m)$, where the r.h.s. is defined as the σ-field generated by the cylinder sets $A_m \times_{n\neq m} \mathcal{X}_n$ where $m \in [M]$ and $A_m \in \mathcal{B}(\mathcal{X}_m)$. Suppose there exists $\mathbb{F} \in \left[\otimes_{m=1}^M \mathcal{M}_b(\mathcal{X}_m)\right]^0 \cap \mathcal{I}$ such that $\mathbb{F} \neq 0$. This means there exists $\mathbb{P} \in \mathcal{M}_1^+\left(\times_{m=1}^M \mathcal{X}_m\right)$ with $(\mathbb{P}_m)_{m=1}^M$ being the marginals of $\mathbb{P}$ such that $\mathbb{F} = \otimes_{m=1}^M \mathbb{F}_m = \mathbb{P} - \otimes_{m=1}^M \mathbb{P}_m$. Since $\mathbb{F} \neq 0$ there exists $A_m \times_{n\neq m} \mathcal{X}_n$ for some $m \in [M]$ and $A_m \in \mathcal{B}(\mathcal{X}_m)$ such that $0 \neq \mathbb{F}(A_m \times_{n\neq m} \mathcal{X}_n) = \mathbb{F}_m(A_m) \prod_{n\neq m} \mathbb{F}_n(\mathcal{X}_n) = \mathbb{P}\left(A_m \times_{n\neq m} \mathcal{X}_n\right) - \mathbb{P}_m(A_m) \prod_{n\neq m} \mathbb{P}_n(\mathcal{X}_n) = \mathbb{P}_m(A_m) - \mathbb{P}_m(A_m) = 0$, leading to a contradiction.

5.2 Proof of Example 1

The proof is structured as follows.

1. First we show that $k := k_1 = k_2$ is a kernel and it is characteristic.
2. Next it is proved that $k_1 \otimes k_2$ is not $\otimes$-characteristic, which implies $k_1 \otimes k_2$ is not characteristic.
3. Finally, the $\mathcal{I}$-characteristic property of $k_1 \otimes k_2$ is established.

The individual steps are as follows:

k **is a kernel.** Assume w.l.o.g. that $x_1 = \ldots = x_N = 1$, $x_{N+1} = \ldots = x_n = 2$. Then it is easy to verify that the Gram matrix $\mathbf{G} = [k(x_i, x_j)]_{i,j=1}^n = \mathbf{a}\mathbf{a}^\top$ where $\mathbf{a} := \left(\mathbf{1}_N^\top, -\mathbf{1}_{n-N}^\top\right)^\top$ and $\mathbf{a}^\top$ is the transpose of $\mathbf{a}$. Clearly $\mathbf{G}$ is positive semidefinite and so k is a kernel.

k **is characteristic**. We will show that k satisfies (5). On $\mathcal{X} = \{1, 2\}$ a finite signed measure $\mathbb{F}$ takes the form $\mathbb{F} = a_1\delta_1 + a_2\delta_2$ for some $a_1, a_2 \in \mathbb{R}$. Thus,

$$\mathbb{F} \in \mathcal{M}_b(\mathcal{X})\backslash\{0\} \Leftrightarrow (a_1, a_2) \neq \mathbf{0} \qquad \text{and} \qquad \mathbb{F}(\mathcal{X}) = 0 \Leftrightarrow a_1 + a_2 = 0. \tag{9}$$

Consider

$$\begin{aligned}\int_{\mathcal{X}}\int_{\mathcal{X}} k(x, x')\,\mathrm{d}\mathbb{F}(x)\,\mathrm{d}\mathbb{F}(x') &= a_1^2 k(1,1) + a_2^2 k(2,2) + 2a_1a_2k(1,2)\\ &= a_1^2 + a_2^2 - 2a_1a_2 = (a_1 - a_2)^2 = 4a_1^2 > 0,\end{aligned} \tag{10}$$

where we used (9) and the facts that $k(1,1) = k(2,2) = 1$, $k(1,2) = -1$.

$k_1 \otimes k_2$ **is not** $\otimes$**-characteristic**. We construct a witness $\mathbb{F} = \mathbb{F}_1 \otimes \mathbb{F}_2 \in \otimes_{m=1}^2 \mathcal{M}_b(\mathcal{X}_m)\backslash\{0\}$ such that

$$\mathbb{F}(\mathcal{X}_1 \times \mathcal{X}_2) = \mathbb{F}_1(\mathcal{X}_1)\mathbb{F}_2(\mathcal{X}_2) = 0, \tag{11}$$

and

$$\begin{aligned}0 &= \int_{\mathcal{X}_1\times\mathcal{X}_2}\int_{\mathcal{X}_1\times\mathcal{X}_2} \underbrace{(k_1 \otimes k_2)((i_1, i_2), (i_1', i_2'))}_{k_1(i_1,i_1')k_2(i_2,i_2')}\,\mathrm{d}\mathbb{F}(i_1, i_2)\,\mathrm{d}\mathbb{F}(i_1', i_2')\\ &= \prod_{m=1}^2 \int_{\mathcal{X}_m}\int_{\mathcal{X}_m} k_m(i_m, i_m')\,\mathrm{d}\mathbb{F}_m(i_m)\,\mathrm{d}\mathbb{F}_m(i_m').\end{aligned} \tag{12}$$

Finite signed measures on $\{1, 2\}$ take the form $\mathbb{F}_1 = \mathbb{F}_1(\mathbf{a}) = a_1\delta_1 + a_2\delta_2$, $\mathbb{F}_2 = \mathbb{F}_2(\mathbf{b}) = b_1\delta_1 + b_2\delta_2$ form, where $\mathbf{a} = (a_1, a_2) \in \mathbb{R}^2, \mathbf{b} = (b_1, b_2) \in \mathbb{R}^2$. With these notations, (11) and (12) can be rewritten as

$$\begin{aligned}0 &= (a_1 + a_2)(b_1 + b_2),\\ 0 &= \left[\sum_{i,i'=1}^2 k_1(i, i')a_ia_{i'}\right]\left[\sum_{j,j'=1}^2 k_2(j, j')b_jb_{j'}\right] = (a_1 - a_2)^2(b_1 - b_2)^2.\end{aligned}$$

Keeping the solutions where neither $\mathbf{a}$ nor $\mathbf{b}$ is the zero vector, there are 2 (symmetric) possibilities: (i) $a_1 + a_2 = 0$, $b_1 = b_2$ and (ii) $a_1 = a_2$, $b_1 + b_2 = 0$. In other words, for any $a, b \neq 0$, the possibilities are (i) $\mathbf{a} = (a, -a)$, $\mathbf{b} = (b, b)$ and (ii) $\mathbf{a} = (a, a)$, $\mathbf{b} = (b, -b)$. This establishes the non-$\left[\otimes_{m=1}^2 \mathcal{M}_b(\mathcal{X}_m)\right]^0$-ispd property of $k_1 \otimes k_2$.

$k_1 \otimes k_2$ **is $\mathcal{I}$-characteristic**. Our goal is to show that $k_1 \otimes k_2$ is $\mathcal{I}$-characteristic, i.e., for any $\mathbb{P} \in \mathcal{M}_1^+(\mathcal{X}_1 \times \mathcal{X}_2)$, $\mu_{k_1 \otimes k_2}(\mathbb{F}) = 0$ implies $\mathbb{F} = 0$, where $\mathbb{F} = \mathbb{P} - \mathbb{P}_1 \otimes \mathbb{P}_2$. We divide the proof into two parts:

1. First we derive the equations of
$$\mathbb{F}(\mathcal{X}_1 \times \mathcal{X}_2) = 0 \quad \text{and} \quad \int\int_{(\mathcal{X}_1 \times \mathcal{X}_2)^2} (k_1 \otimes k_2)\left((i,j),(r,s)\right) \mathrm{d}\mathbb{F}(i,j)\, \mathrm{d}\mathbb{F}(r,s) = 0 \quad (13)$$
for general finite signed measures $\mathbb{F} = \sum_{i,j=1}^2 a_{ij}\delta_{(i,j)}$ on $\mathcal{X}_1 \times \mathcal{X}_2$.
2. Then, we apply the $\mathbb{F} = \mathbb{P} - \mathbb{P}_1 \otimes \mathbb{P}_2$ parameterization and solve for $\mathbb{P}$ that satisfies (13) to conclude that $\mathbb{P} = \mathbb{P}_1 \otimes \mathbb{P}_2$, i.e., $\mathbb{F} = 0$. Note that in the chosen parametrization for $\mathbb{F}$, $\mathbb{F}(\mathcal{X}_1 \times \mathcal{X}_2) = 0$ holds automatically.

The details are as follows.

Step 1.

$$0 = \mathbb{F}(\mathcal{X}_1 \times \mathcal{X}_2) \Leftrightarrow 0 = a_{11} + a_{12} + a_{21} + a_{22}, \quad (14)$$

$$\begin{aligned}
0 &= \int_{\mathcal{X}_1 \times \mathcal{X}_2} \int_{\mathcal{X}_1 \times \mathcal{X}_2} \underbrace{(k_1 \otimes k_2)\left((i,j),(r,s)\right)}_{k_1(i,r)k_2(j,s)} \mathrm{d}\mathbb{F}(i,j)\, \mathrm{d}\mathbb{F}(r,s) \\
&= \sum_{i,j=1}^2 \sum_{r,s=1}^2 k_1(i,r)k_2(j,s)a_{ij}a_{rs} = \sum_{i,r=1}^2 k_1(i,r) \sum_{j,s=1}^2 k_2(j,s)a_{ij}a_{rs} \\
&= k_1(1,1)\left[k_2(1,1)a_{11}a_{11} + k_2(1,2)a_{11}a_{12} + k_2(2,1)a_{12}a_{11} + k_2(2,2)a_{12}a_{12}\right] \\
&\quad + k_1(1,2)\left[k_2(1,1)a_{11}a_{21} + k_2(1,2)a_{11}a_{22} + k_2(2,1)a_{12}a_{21} + k_2(2,2)a_{12}a_{22}\right] \\
&\quad + k_1(2,1)\left[k_2(1,1)a_{21}a_{11} + k_2(1,2)a_{21}a_{12} + k_2(2,1)a_{22}a_{11} + k_2(2,2)a_{22}a_{12}\right] \\
&\quad + k_1(2,2)\left[k_2(1,1)a_{21}a_{21} + k_2(1,2)a_{21}a_{22} + k_2(2,1)a_{22}a_{21} + k_2(2,2)a_{22}a_{22}\right] \\
&= \underbrace{\left(a_{11}^2 - 2a_{11}a_{12} + a_{12}^2\right)}_{(a_{11}-a_{12})^2} + \underbrace{\left(a_{21}^2 - 2a_{21}a_{22} + a_{22}^2\right)}_{(a_{21}-a_{22})^2} - 2\underbrace{\left(a_{11}a_{21} - a_{11}a_{22} - a_{12}a_{21} + a_{12}a_{22}\right)}_{(a_{11}-a_{12})(a_{21}-a_{22})} \\
&= (a_{11} - a_{12} - a_{21} + a_{22})^2. \quad (15)
\end{aligned}$$

Solving (14) and (15) yields

$$a_{11} + a_{22} = 0 \quad \text{and} \quad a_{12} + a_{21} = 0. \quad (16)$$

Step 2. Any $\mathbb{P} \in \mathcal{M}_1^+(\mathcal{X}_1 \times \mathcal{X}_2)$ can be parametrized as

$$\mathbb{P} = \sum_{i,j=1}^2 p_{ij}\delta_{(i,j)}, \quad p_{ij} \geq 0, \forall\,(i,j) \quad \text{and} \quad \sum_{i,j=1}^2 p_{ij} = 1. \quad (17)$$

Let $\mathbb{F} = \mathbb{P} - \mathbb{P}_1 \otimes \mathbb{P}_2 = \sum_{i,j=1}^2 a_{ij}\delta_{(i,j)}$; for illustration see Table 1. It follows from step 1 that $\mathbb{F}$ satisfying (16) is equivalent to satisfying (13). Therefore, for the choice of $\mathbb{F} := \mathbb{P} - \mathbb{P}_1 \otimes \mathbb{P}_2$, we obtain

$$p_{11} - (p_{11} + p_{12})(p_{11} + p_{21}) + p_{22} - (p_{21} + p_{22})(p_{12} + p_{22}) = 0, \quad (18)$$

$\mathbb{P}$: $y\backslash x$	1	2	$\mathbb{P}_2$
1	p_{11}	p_{21}	$q_1 = p_{11} + p_{21}$
2	p_{12}	p_{22}	$q_2 = p_{12} + p_{22}$
$\mathbb{P}_1$	$p_1 = p_{11} + p_{12}$	$p_2 = p_{21} + p_{22}$	

$\Rightarrow$

$\mathbb{F} := \mathbb{P} - \mathbb{P}_1 \otimes \mathbb{P}_2$	1	2
1	$a_{11} = p_{11} - (p_{11} + p_{12})(p_{11} + p_{21})$	$a_{21} = p_{21} - (p_{21} + p_{22})(p_{11} + p_{21})$
2	$a_{12} = p_{12} - (p_{11} + p_{12})(p_{12} + p_{22})$	$a_{22} = p_{22} - (p_{21} + p_{22})(p_{12} + p_{22})$

Table 1: Joint ($\mathbb{P}$), joint minus product of the marginals ($\mathbb{P} - \mathbb{P}_1 \otimes \mathbb{P}_2$).

$\mathbb{P}$: $y\backslash x$	1	2	$\mathbb{P}_2$
1	$p_{11} = \frac{a[1-(a+b)]}{a+b}$	$p_{21} = a$	$q_1 = \frac{a}{a+b}$
2	$p_{12} = \frac{b[1-(a+b)]}{a+b}$	$p_{22} = b$	$q_2 = \frac{b}{a+b}$
$\mathbb{P}_1$	$p_1 = 1 - (a+b)$	$p_2 = a + b$	

Table 2: Family of probability distributions solving (17)–(19).

$$p_{12} - (p_{11} + p_{12})(p_{12} + p_{22}) + p_{21} - (p_{21} + p_{22})(p_{11} + p_{21}) = 0, \quad (19)$$

where $(p_{ij})_{i,j\in[2]}$ satisfy (17). Solving (17)–(19), we obtain

$$p_{11} = \frac{a[1-(a+b)]}{a+b}, \quad p_{12} = \frac{b[1-(a+b)]}{a+b}, \quad p_{21} = a \quad \text{and} \quad p_{22} = b,$$

with $0 \leq a, b \leq 1$, $a + b \leq 1$ and $(a, b) \neq \mathbf{0}$. The resulting distribution family with its marginals is summarized in Table 2. It can be seen that each member of this family (any a, b in the constraint set) factorizes: $\mathbb{P} = \mathbb{P}_1 \otimes \mathbb{P}_2$. In other words, $\mathbb{F} = \mathbb{P} - \mathbb{P}_1 \otimes \mathbb{P}_2 = 0$; hence $k_1 \otimes k_2$ is $\mathcal{I}$-characteristic.

Remark. We would like to mention that while k_1 and k_2 are characteristic, they are not universal. Since $\mathcal{X}$ is finite, the usual notion of universality (also called c-universality) matches with c_0-universality. Therefore, from (10), we have $\int_{\mathcal{X}}\int_{\mathcal{X}} k(x,x')\,\mathrm{d}\mathbb{F}(x)\,\mathrm{d}\mathbb{F}(x) = (a_1 - a_2)^2$ where $\mathbb{F} = a_1\delta_1 + a_2\delta_2$ for some $a_1, a_2 \in \mathbb{R}\backslash\{0\}$. Clearly, the choice of $a_1 = a_2$ establishes that there exists $\mathbb{F} \in \mathcal{M}_b(\mathcal{X})\backslash\{0\}$ such that $\int_{\mathcal{X}}\int_{\mathcal{X}} k(x,x')\,\mathrm{d}\mathbb{F}(x)\,\mathrm{d}\mathbb{F}(x) = 0$. Hence k is not universal. Note that the constraint in (9), which is needed to verify the characteristic property of k is not needed to verify its universality.

5.3 Proof of Theorem 3

Define $\mathcal{H}_m := \mathcal{H}_{k_m}$.

(i) Suppose k_1 and k_2 are characteristic and that for some $\mathbb{F} = \mathbb{P} - \mathbb{P}_1 \otimes \mathbb{P}_2 \in \mathcal{I}$,

$$\mathcal{H}_1 \otimes \mathcal{H}_2 \ni \int_{\mathcal{X}_1 \times \mathcal{X}_2} (k_1 \otimes k_2)(\cdot, x) \, \mathrm{d}\mathbb{F}(x) = \int_{\mathcal{X}_1 \times \mathcal{X}_2} k_1(\cdot, x_1) \otimes k_2(\cdot, x_2) \, \mathrm{d}\mathbb{F}(x) = 0, \tag{20}$$

where $x = (x_1, x_2)$. We want to show that $\mathbb{F} = 0$. By the second-countability of $\mathcal{X}_m$-s, the product σ-field, i.e., $\otimes_{m=1}^2 \mathcal{B}(\mathcal{X}_m)$ generated by the cylinder sets $B_1 \times \mathcal{X}_2$ and $\mathcal{X}_1 \times B_2$ ($B_m \in \mathcal{B}(\mathcal{X}_m), m = 1, 2$), coincides with the Borel σ-field $\mathcal{B}(\mathcal{X}_1 \times \mathcal{X}_2)$ on the product space (Dudley, 2004, Lemma 4.1.7):

$$\otimes_{m=1}^2 \mathcal{B}(\mathcal{X}_m) = \mathcal{B}(\mathcal{X}_1 \times \mathcal{X}_2).$$

Hence, it is sufficient to prove that $\mathbb{F}(B_1 \times B_2) = 0$, $\forall B_m \in \mathcal{B}(\mathcal{X}_m)$, $m = 1, 2$. To this end, it follows from (20) that for all $h_2 \in \mathcal{H}_2$,

$$\mathcal{H}_1 \ni \int_{\mathcal{X}_1 \times \mathcal{X}_2} k_1(\cdot, x_1) h_2(x_2) \, \mathrm{d}\mathbb{F}(x) = \int_{\mathcal{X}_1} k_1(\cdot, x_1) \, \mathrm{d}\nu(x_1) = 0, \tag{21}$$

where

$$\nu(B_1) := \nu_{h_2}(B_1) = \int_{\mathcal{X}_1 \times \mathcal{X}_2} \chi_{B_1}(x_1) h_2(x_2) \, \mathrm{d}\mathbb{F}(x), \quad B_1 \in \mathcal{B}(\mathcal{X}_1).$$

Since k_1 is characteristic, (21) implies $\nu = 0$, provided that $|\nu|(\mathcal{X}_1) < \infty$ and $\nu(\mathcal{X}_1) = 0$. These two requirements hold:

$$\begin{aligned}
\nu(\mathcal{X}_1) &= \int_{\mathcal{X}_1 \times \mathcal{X}_2} h_2(x_2) \, \mathrm{d}\mathbb{F}(x) = \int_{\mathcal{X}_2} h_2(x_2) \, \mathrm{d}[\mathbb{P}_2 - \mathbb{P}_2](x_2) = 0, \\
|\nu|(\mathcal{X}_1) &\leq \int_{\mathcal{X}_1 \times \mathcal{X}_2} \underbrace{|h_2(x_2)|}_{\left|\langle h_2, k_2(\cdot, x_2) \rangle_{\mathcal{H}_2}\right|} \mathrm{d}[\mathbb{P} + \mathbb{P}_1 \otimes \mathbb{P}_2](x_1, x_2) \\
&\leq \|h_2\|_{\mathcal{H}_2} \int_{\mathcal{X}_1 \times \mathcal{X}_2} \sqrt{k_2(x_2, x_2)} \, \mathrm{d}[\mathbb{P} + \mathbb{P}_1 \otimes \mathbb{P}_2](x_1, x_2) \\
&\leq 2 \|h_2\|_{\mathcal{H}_2} \int_{\mathcal{X}_2} \sqrt{k_2(x_2, x_2)} \, \mathrm{d}\mathbb{P}_2(x_2) < \infty,
\end{aligned}$$

where the last inequality follows from the boundedness of k_2. The established $\nu = 0$ implies that for $\forall B_1 \in \mathcal{B}(\mathcal{X}_1)$ and $\forall h_2 \in \mathcal{H}_2$,

$$0 = \nu(B_1) = \left\langle h_2, \int_{\mathcal{X}_1 \times \mathcal{X}_2} \chi_{B_1}(x_1) k_2(\cdot, x_2) \, \mathrm{d}\mathbb{F}(x) \right\rangle_{\mathcal{H}_2},$$

and hence

$$0 = \int_{\mathcal{X}_1 \times \mathcal{X}_2} \chi_{B_1}(x_1) k_2(\cdot, x_2) \, \mathrm{d}\mathbb{F}(x) = \int_{\mathcal{X}_2} k_2(\cdot, x_2) \, \mathrm{d}\theta_{B_1}(x_2), \tag{22}$$

where

$$\theta_{B_1}(B_2) = \int_{\mathcal{X}_1 \times \mathcal{X}_2} \chi_{B_1}(x_1) \chi_{B_2}(x_2) \, \mathrm{d}\mathbb{F}(x), \quad B_2 \in \mathcal{B}(\mathcal{X}_2).$$

Using the characteristic property of k_2, it follows from (22) that $\theta_{B_1} = 0$ for $\forall B_1 \in \mathcal{B}(\mathcal{X}_1)$, i.e.,

$$0 = \theta_{B_1}(B_2) = \mathbb{F}(B_1 \times B_2), \quad \forall B_1 \in \mathcal{B}(\mathcal{X}_1), \forall B_2 \in \mathcal{B}(\mathcal{X}_2)$$

provided that $\theta_{B_1}(\mathcal{X}_2) = 0$ and $|\theta_{B_1}|(\mathcal{X}_2) < \infty$. Indeed, both these conditions hold:

$$\begin{aligned}
\theta_{B_1}(\mathcal{X}_2) &= \int_{\mathcal{X}_1 \times \mathcal{X}_2} \chi_{B_1}(x_1) \, \mathrm{d}\mathbb{F}(x) = \int_{\mathcal{X}_1} \chi_{B_1}(x_1) \, \mathrm{d}[\mathbb{P}_1 - \mathbb{P}_1](x_1) = 0, \\
|\theta_{B_1}|(\mathcal{X}_2) &\leq \int_{\mathcal{X}_1 \times \mathcal{X}_2} \mathrm{d}[\mathbb{P} + \mathbb{P}_1 \otimes \mathbb{P}_2](x) = 2.
\end{aligned}$$

(ii) Assume w.l.o.g. that k_1 is not characteristic. This means there exists $\mathbb{P}_1 \neq \mathbb{P}_1' \in \mathcal{M}_1^+(\mathcal{X}_1)$ such that $\mu_{k_1}(\mathbb{P}_1) = \mu_{k_1}(\mathbb{P}_1')$. Our goal is to construct an $\mathbb{F} \in \mathcal{M}_1^+\left(\times_{m=1}^M \mathcal{X}_m\right)$ such that

$$\mu_{\otimes_{m=1}^M k_m}\left(\mathbb{F} - \otimes_{m=1}^M \mathbb{F}_m\right) = \int_{\times_{m=1}^M} \otimes_{m=1}^M k_m(\cdot, x_m) \, \mathrm{d}\left[\mathbb{F} - \otimes_{m=1}^M \mathbb{F}_m\right] = 0, \text{ but } \mathbb{F} \neq \otimes_{m=1}^M \mathbb{F}_m.$$

Define $\mathbb{I} := \mathbb{F} - \otimes_{m=1}^M \mathbb{F}_m \in \mathcal{I}$. In other words we want to get a witness $\mathbb{I} \in \mathcal{I}$ proving that $\otimes_{m=1}^M k_m$ is not $\mathcal{I}$-characteristic. Let us take $z \neq z' \in \mathcal{X}_2$, which is possible since $|\mathcal{X}_2| \geq 2$. Let us define $\mathbb{F}$ as[6]

$$\mathbb{F} = \frac{\mathbb{P}_1 \otimes \delta_z \otimes (\otimes_{m=3}^M \mathbb{Q}_m) + \mathbb{P}_1' \otimes \delta_{z'} \otimes (\otimes_{m=3}^M \mathbb{Q}_m)}{2} \in \mathcal{M}_1^+\left(\times_{m=1}^M \mathcal{X}_m\right).$$

It is easy to verify that

$$\mathbb{F}_1 = \frac{\mathbb{P}_1 + \mathbb{P}_1'}{2}, \; \mathbb{F}_2 = \frac{\delta_z + \delta_{z'}}{2} \text{ and } \mathbb{F}_m = \mathbb{Q}_m \quad (m = 3, \ldots, M),$$

where $\mathbb{Q}_3, \ldots, \mathbb{Q}_M$ are arbitrary probability measures on $\mathcal{X}_3, \ldots, \mathcal{X}_M$, respectively. First we check that $\mathbb{I} \neq 0$. Indeed it is the case since

- $z \neq z'$ and $\mathcal{X}_2$ is a Hausdorff space, there exists $B_2 \in \mathcal{B}(\mathcal{X}_2)$ such that $z \in B_2$, $z' \notin B_2$.
- $\mathbb{P}_1 \neq \mathbb{P}_1'$, $\mathbb{P}_1(B_1) \neq \mathbb{P}_1'(B_1)$ for some $B_1 \in \mathcal{B}(\mathcal{X}_1)$.

Let $S = B_1 \times B_2 \times \left(\times_{m=3}^M \mathcal{X}_m\right)$, and compare its measure under $\mathbb{F}$ and $\otimes_{m=1}^M \mathbb{F}_m$:

$$\begin{aligned}
\mathbb{F}(S) &= \frac{\mathbb{P}_1(B_1) \overbrace{\delta_z(B_2)}^{=1\,(z \in B_2)} \prod_{m=3}^M \overbrace{\mathbb{Q}_m(\mathcal{X}_m)}^{=1} + \mathbb{P}_1'(B_1) \overbrace{\delta_{z'}(B_2)}^{=0\,(z' \notin B_2)} \prod_{m=3}^M \overbrace{\mathbb{Q}_m(\mathcal{X}_m)}^{=1}}{2} \\
&= \frac{\mathbb{P}_1(B_1)}{2}, \\
\left(\otimes_{m=1}^M \mathbb{F}_m\right)(S) &= \prod_{m=1}^M \mathbb{F}_m(B_m) = \frac{\mathbb{P}_1(B_1) + \mathbb{P}_1'(B_1)}{2} \frac{\overbrace{\delta_z(B_2)}^{=1} + \overbrace{\delta_{z'}(B_2)}^{=0}}{2} \prod_{m=3}^M \overbrace{\mathbb{Q}_m(\mathcal{X}_m)}^{=1}
\end{aligned}$$

6. The $\mathbb{F}$ construction specializes to that of Lyons (2013, Proposition 3.15) in the $M = 2$ case; Lyons used it for distance covariances, which is known to be equivalent to HSIC (Sejdinovic et al., 2013b).

$$= \frac{\mathbb{P}_1(B_1) + \mathbb{P}'_1(B_1)}{4} \neq \frac{\mathbb{P}_1(B_1)}{2},$$

where the last equality holds since $\mathbb{P}_1(B_1) \neq \mathbb{P}'_1(B_1)$. This shows that $\mathbb{I} = \mathbb{F} - \otimes_{m=1}^M \mathbb{F}_m \neq 0$ since $\mathbb{I}(S) \neq 0$.
Next we prove that $\mu_{\otimes_{m=1}^M k_m}\left(\mathbb{F} - \otimes_{m=1}^M \mathbb{F}_m\right) = 0$. Indeed,

$$\begin{aligned}
\mu_{\otimes_{m=1}^M k_m}(\mathbb{I}) &= \int_{\times_{m=1}^M \mathcal{X}_m} \otimes_{m=1}^M k_m(\cdot, x_m)\, \mathrm{d}\left[\mathbb{F} - \otimes_{m=1}^M \mathbb{F}_m\right](x_1, \ldots, x_M) \\
&= \int_{\times_{m=1}^M \mathcal{X}_m} \otimes_{m=1}^M k_m(\cdot, x_m)\, \mathrm{d}\left(\left[\frac{\mathbb{P}_1 \otimes \delta_z + \mathbb{P}'_1 \otimes \delta_{z'}}{2} - \frac{\mathbb{P}_1 + \mathbb{P}'_1}{2} \otimes \frac{\delta_z + \delta_{z'}}{2}\right]\right. \\
&\qquad \left.\otimes \left(\otimes_{m=3}^M \mathbb{Q}_m\right)\right)(x_1, \ldots, x_M) \\
&= \int_{\times_{m=1}^M \mathcal{X}_m} \otimes_{m=1}^M k_m(\cdot, x_m)\, \mathrm{d}\left(\left[\frac{\mathbb{P}_1(x_1) \otimes \delta_z(x_2) + \mathbb{P}'_1(x_1) \otimes \delta_{z'}(x_2)}{2}\right.\right. \\
&\qquad - \frac{\mathbb{P}_1(x_1) \otimes \delta_z(x_2) + \mathbb{P}_1(x_1) \otimes \delta_{z'}(x_2)}{4} \\
&\qquad \left.\left. - \frac{\mathbb{P}'_1(x_1) \otimes \delta_z(x_2) + \mathbb{P}'_1(x_1) \otimes \delta_{z'}(x_2)}{4}\right] \otimes \left(\otimes_{m=3}^M \mathbb{Q}_m(x_m)\right)\right) \\
&\overset{(*)}{=} \left[\frac{\mu_{k_1}(\mathbb{P}_1) \otimes k_2(\cdot, z) + \mu_{k_1}(\mathbb{P}'_1) \otimes k_2(\cdot, z')}{2}\right. \\
&\qquad - \frac{\mu_{k_1}(\mathbb{P}_1) \otimes k_2(\cdot, z) + \mu_{k_1}(\mathbb{P}_1) \otimes k_2(\cdot, z')}{4} \\
&\qquad \left. - \frac{\mu_{k_1}(\mathbb{P}'_1) \otimes k_2(\cdot, z) + \mu_{k_1}(\mathbb{P}'_1) \otimes k_2(\cdot, z')}{4}\right] \otimes \left[\otimes_{m=3}^M \mu_{k_m}(\mathbb{Q}_m)\right] \\
&= \underbrace{0}_{\in \mathcal{H}_{k_1 \otimes k_2}} \otimes \left[\otimes_{m=3}^M \mu_{k_m}(\mathbb{Q}_m)\right] = 0,
\end{aligned}$$

where we used $\mu_{k_1}(\mathbb{P}_1) = \mu_{k_1}(\mathbb{P}'_1)$ in $(*)$.

5.4 Proof of Example 2

Let $M = 3$, $\times_{m=1}^M \mathcal{X}_m = \{(i_1, i_2, i_3) : i_m \in \{1, 2\},\, m \in [3]\}$, $k_m(x, x') = 2\delta_{x,x'} - 1$. Our goal is to show that $\otimes_{m=1}^3 k_m$ is *not* $\mathcal{I}$-characteristic. The structure of the proof is as follows:

1. First we describe the equations of the non-characteristic property of $\otimes_{m=1}^3 k_m$ with a general finite signed measure $\mathbb{F} = \sum_{i_1,i_2,i_3=1}^2 a_{i_1,i_2,i_3} \delta_{(i_1,i_2,i_3)}$ on $\times_{m=1}^3 \mathcal{X}_m$ where $a_{i_1,i_2,i_3} \in \mathbb{R}$ $(\forall\, i_1, i_2, i_3)$.
2. Next, we apply the $\mathbb{F} = \mathbb{P} - \otimes_{m=1}^3 \mathbb{P}_m$ parameterization and show that there exists $\mathbb{P}$ that satisfies the equations of step 1 to conclude that $\otimes_{m=1}^3 k_m$ is not $\mathcal{I}$-characteristic.

The details are as follows.

Step 1. The equations of non-characteristic property in terms of $\mathbf{A} = [a_{i_1,i_2,i_3}]_{(i_m)_{m=1}^3 \in [2]^3} \in \mathbb{R}^{2\times 2\times 2}$ are

$$\mathbb{F} \in \mathcal{M}_b\left(\times_{m=1}^3 \mathcal{X}_m\right) \setminus \{0\} \Leftrightarrow \mathbf{A} \neq \mathbf{0},$$

$$0 = \mathbb{F}(\times_{m=1}^3 \mathcal{X}_m) \Leftrightarrow 0 = \sum_{i_1,i_2,i_3=1}^2 a_{i_1,i_2,i_3}, \tag{23}$$

$$\begin{aligned}
0 &= \int_{\times_{m=1}^3 \mathcal{X}_m} \int_{\times_{m=1}^3 \mathcal{X}_m} \underbrace{\left(\otimes_{m=1}^3 k_m\right)\left((i_1, i_2, i_3), (i'_1, i'_2, i'_3)\right)}_{\Pi_{m=1}^3 k_m(i_m, i'_m)} \mathrm{d}\mathbb{F}(i_1, i_2, i_3)\, \mathrm{d}\mathbb{F}(i'_1, i'_2, i'_3) \\
&= \sum_{i_1,i_2,i_3=1}^2 \sum_{i'_1,i'_2,i'_3=1}^2 \prod_{m=1}^3 k_m(i_m, i'_m) a_{i_1,i_2,i_3} a_{i'_1,i'_2,i'_3}.
\end{aligned} \tag{24}$$

Solving (23) and (24) yields

$$a_{1,1,1} + a_{1,2,2} + a_{2,1,2} + a_{2,2,1} = 0 \quad \text{and} \quad a_{1,1,2} + a_{1,2,1} + a_{2,1,1} + a_{2,2,2} = 0.$$

Step 2. The equations of non $\mathcal{I}$-characteristic property can be obtained from step 1 by choosing $\mathbb{F} = \mathbb{P} - \otimes_{m=1}^M \mathbb{P}_m$, where

$$\mathbb{P} = \sum_{i_1,i_2,i_3=1}^2 p_{i_1,i_2,i_3} \delta_{(i_1,i_2,i_3)} \quad \text{and} \quad \mathbf{P} = [p_{i_1,i_2,i_3}]_{(i_m)_{m=1}^3 \in [2]^3} \in \mathbb{R}^{2\times 2\times 2}.$$

In other words, it is sufficient to obtain a $\mathbf{P}$ that solves the following system of equations for which $\mathbf{A} = \mathbf{A}(\mathbf{P}) \neq \mathbf{0}$:

$$\sum_{i_1,i_2,i_3=1}^2 p_{i_1,i_2,i_3} = 1, \tag{25}$$

$$p_{i_1,i_2,i_3} \geq 0,\ \forall\, (i_1, i_2, i_3) \in [2]^3, \tag{26}$$

$$a_{1,1,1} + a_{1,2,2} + a_{2,1,2} + a_{2,2,1} = 0, \tag{27}$$

$$a_{1,1,2} + a_{1,2,1} + a_{2,1,1} + a_{2,2,2} = 0, \tag{28}$$

where

$$a_{i_1,i_2,i_3} = p_{i_1,i_2,i_3} - p_{1,i_1} p_{2,i_2} p_{3,i_3}, \tag{29}$$

and

$$p_{1,i_1} = \sum_{i_2,i_3=1}^2 p_{i_1,i_2,i_3}, \quad p_{2,i_2} = \sum_{i_1,i_3=1}^2 p_{i_1,i_2,i_3}, \quad p_{3,i_3} = \sum_{i_1,i_2=1}^2 p_{i_1,i_2,i_3}. \tag{30}$$

One can get an analytical description for the solution of (25)–(30), where the solution $\mathbf{P}(\mathbf{z})$ is parameterized by $\mathbf{z} = (z_0, \ldots, z_5) \in \mathbb{R}^6$. For explicit expressions, we refer the reader to Appendix A. In the following, we present two examples of $\mathbf{P}$ that satisfy (25)–(30) such that $\mathbf{A} \neq \mathbf{0}$, thereby establishing the non $\mathcal{I}$-characteristic property of $\otimes_{m=1}^3 k_m$.

1. $\mathbf{P}$:

$$\begin{aligned}
p_{1,1,1} &= \frac{1}{5}, & p_{1,1,2} &= \frac{1}{10}, & p_{1,2,1} &= \frac{1}{10}, & p_{1,2,2} &= \frac{1}{10}, \\
p_{2,1,1} &= \frac{1}{5}, & p_{2,1,2} &= \frac{1}{10}, & p_{2,2,1} &= \frac{1}{10}, & p_{2,2,2} &= \frac{1}{10},
\end{aligned}$$

and **A**:

$$a_{1,1,1}=\frac{1}{50}, \quad a_{1,1,2}=-\frac{1}{50}, \quad a_{1,2,1}=-\frac{1}{50}, \quad a_{1,2,2}=\frac{1}{50}, \tag{31}$$
$$a_{2,1,1}=\frac{1}{50}, \quad a_{2,1,2}=-\frac{1}{50}, \quad a_{2,2,1}=-\frac{1}{50}, \quad a_{2,2,2}=\frac{1}{50}. \tag{32}$$

2. **P**:

$$p_{1,1,1}=0, \quad p_{1,1,2}=\frac{1}{10}, \quad p_{1,2,1}=\frac{1}{10}, \quad p_{1,2,2}=\frac{1}{10},$$
$$p_{2,1,1}=\frac{1}{10}, \quad p_{2,1,2}=\frac{1}{10}, \quad p_{2,2,1}=\frac{3}{10}, \quad p_{2,2,2}=\frac{1}{5},$$

and **A**:

$$a_{1,1,1}=-\frac{9}{200}, \quad a_{1,1,2}=\frac{11}{200}, \quad a_{1,2,1}=-\frac{1}{200}, \quad a_{1,2,2}=-\frac{1}{200},$$
$$a_{2,1,1}=-\frac{1}{200}, \quad a_{2,1,2}=-\frac{1}{200}, \quad a_{2,2,1}=\frac{11}{200}, \quad a_{2,2,2}=-\frac{9}{200}.$$

In fact these examples are obtained with the choices $\mathbf{z}=(\frac{1}{10},\frac{1}{10},\frac{1}{10},\frac{1}{10},\frac{1}{10},\frac{1}{10})$ and $\mathbf{z}=(\frac{3}{10},\frac{1}{10},\frac{1}{10},\frac{1}{10},\frac{1}{10},\frac{2}{10})$ respectively. See Appendix A for details.

5.5 Proof of Theorem 4

It follows from (8) and Remark 2(iii) that $(v)\Rightarrow(iii)\Rightarrow(ii)\Leftrightarrow(i)$. It also follows from (8) and Theorem 3(ii) that $(v)\Rightarrow(iv)\Rightarrow(i)$. We now show that $(i)\Rightarrow(v)$ which establishes the equivalence of (i)–(v). Suppose (i) holds. Then by Bochner's theorem (Wendland, 2005, Theorem 6.6), we have that for all $m\in[M]$,

$$k_m(x_m,y_m)=\int_{\mathbb{R}^{d_m}} e^{-\sqrt{-1}\langle\omega_m,x_m-y_m\rangle}\,\mathrm{d}\Lambda_m(\omega_m),\ x_m,y_m\in\mathbb{R}^{d_m},$$

where $(\Lambda_m)_{m=1}^M$ are finite non-negative Borel measures on $(\mathbb{R}^{d_m})_{m=1}^M$ respectively. This implies

$$\otimes_{m=1}^M k_m(x_m,y_m)=\otimes_{m=1}^M\int_{\mathbb{R}^{d_m}} e^{-\sqrt{-1}\langle\omega_m,x_m-y_m\rangle}\,\mathrm{d}\Lambda_m(\omega_m)=\int_{\mathbb{R}^d} e^{-\sqrt{-1}\langle\omega,x-y\rangle}\,\mathrm{d}\Lambda(\omega),$$

where $x=(x_1,\ldots,x_M)\in\mathbb{R}^d$, $y=(y_1,\ldots,y_M)\in\mathbb{R}^d$, $\omega=(\omega_1,\ldots,\omega_M)\in\mathbb{R}^d$, $d=\sum_{m=1}^M d_m$ and $\Lambda:=\otimes_{m=1}^M\Lambda_m$. Sriperumbudur et al. (2010, Theorem 9) showed that k_m is characteristic iff $\mathrm{supp}\,(\Lambda_m)=\mathbb{R}^{d_m}$, where $\mathrm{supp}(\cdot)$ denotes the support of its argument. Since $\mathrm{supp}(\Lambda)=\mathrm{supp}\left(\otimes_{m=1}^M\Lambda_m\right)=\times_{m=1}^M\mathrm{supp}\,(\Lambda_m)=\times_{m=1}^M\mathbb{R}^{d_m}=\mathbb{R}^d$, it follows that $\otimes_{m=1}^M k_m$ is characteristic.

5.6 Proof of Theorem 5

The c_0-kernel property of k_m-s $(m=1,\ldots,M)$ implies that of $\otimes_{m=1}^M k_m$. Moreover, $\mathcal{X}_m$-s are LCP spaces, hence $\times_{m=1}^M\mathcal{X}_m$ is also LCP.

($\Leftarrow$) Assume that $\otimes_{m=1}^M k_m$ is c_0-universal. Since $\otimes_{m=1}^M\mathcal{M}_b\left(\mathcal{X}_m\right)\subseteq\mathcal{M}_b\left(\times_{m=1}^M\mathcal{X}_m\right)$, we have that for all $\mathbb{F}=\otimes_{m=1}^M\mathbb{F}_m\in\otimes_{m=1}^M\mathcal{M}_b(\mathcal{X}_m)\backslash\{0\}$,

$$0<\int_{\times_{m=1}^M\mathcal{X}_m}\int_{\times_{m=1}^M\mathcal{X}_m}\underbrace{\left(\otimes_{m=1}^M k_m\right)(x,x')}_{\prod_{m=1}^M k_m(x_m,x'_m)}\mathrm{d}\mathbb{F}(x)\,\mathrm{d}\mathbb{F}(x')$$
$$=\prod_{m=1}^M\int_{\mathcal{X}_m\times\mathcal{X}_m} k_m(x_m,x'_m)\,\mathrm{d}\mathbb{F}_m(x_m)\,\mathrm{d}\mathbb{F}_m\left(x'_m\right),$$

where $x=(x_1,\ldots,x_M)$ and $x'=(x'_1,\ldots,x'_M)$. The above inequality implies

$$\int_{\mathcal{X}_m\times\mathcal{X}_m} k_m(x_m,x'_m)\,\mathrm{d}\mathbb{F}_m(x_m)\,\mathrm{d}\mathbb{F}_m\left(x'_m\right)>0,\ \forall\, m\in[M].$$

Since $\mathbb{F}\in\otimes_{m=1}^M\mathcal{M}_b\left(\mathcal{X}_m\right)\backslash\{0\}$ iff $\mathbb{F}_m\in\mathcal{M}_b(\mathcal{X}_m)\backslash\{0\}$ for all $m\in[M]$, the result follows.

($\Rightarrow$) Assume that k_m-s are c_0-universal. By the note above $\otimes_{m=1}^M k_m$ is c_0-kernel; its c_0-universality is equivalent to the injectivity of $\mu=\mu_{\otimes_{m=1}^M k_m}$ on $\mathcal{M}_b\left(\times_{m=1}^M\mathcal{X}_m\right)$. In other words, we want to prove that $\mu(\mathbb{F})=0$ implies $\mathbb{F}=0$, where $\mathbb{F}\in\mathcal{M}_b\left(\times_{m=1}^M\mathcal{X}_m\right)$. We will use the shorthand $\mathcal{H}_m=\mathcal{H}_{k_m}$ below.

Suppose there exists $\mathbb{F}\in\mathcal{M}_b\left(\times_{m=1}^M\mathcal{X}_m\right)$ such that

$$\mu_{\mathbb{F}}=\int_{\times_{m=1}^M\mathcal{X}_m}\underbrace{\left(\otimes_{m=1}^M k_m\right)(\cdot,x)}_{\otimes_{m=1}^M k_m(\cdot,x_m)}\mathrm{d}\mathbb{F}(x)=0\ (\in\otimes_{m=1}^M\mathcal{H}_m). \tag{33}$$

Since $\mathcal{X}_m$-s are LCP, $\otimes_{m=1}^M\mathcal{B}\left(\mathcal{X}_m\right)=\mathcal{B}\left(\times_{m=1}^M\mathcal{X}_m\right)$ (Steinwart and Christmann, 2008, page 480). Hence, in order to get $\mathbb{F}=0$ it is sufficient to prove that

$$\mathbb{F}\left(\times_{m=1}^M B_m\right)=0,\quad \forall\, B_m\in\mathcal{B}(\mathcal{X}_m), m\in[M].$$

We will prove by induction that for $m=0,\ldots,M$

$$\left(\otimes_{j=m+1}^M\mathcal{H}_j\ni\right)0=\int_{\times_{j=1}^M\mathcal{X}_j}\prod_{j=1}^m\chi_{B_j}(x_j)\otimes_{j=m+1}^M k_j(\cdot,x_j)\,\mathrm{d}\mathbb{F}(x)$$
$$=:o(B_1,\ldots,B_m,k_{m+1},\ldots,k_M),\forall\, B_j\in\mathcal{B}(\mathcal{X}_j),\ j\in[m], \tag{34}$$

which

- $(*)$ reduces to (33) when $m=0$ by defining $\prod_{j=1}^0\chi_{B_j}(x_j):=1$;
- $(\dagger)$ for $m=M$, $\otimes_{m=M+1}^M\mathcal{H}_m$ is defined to be equal to $\mathbb{R}$ and $\otimes_{m=M+1}^M k_m(\cdot,x_m):=1$, in which case $o(B_1,\ldots,B_M)=\mathbb{F}\left(\times_{j=1}^M B_j\right)=0\Rightarrow\mathbb{F}=0$, the result we want to prove.

From the above, it is clear that (34) holds for $m=0$. Assuming (34) holds for some m, we now prove that it holds for $m+1$. To this end, it follows from (34) that $\forall\, h_{m+2}\in\mathcal{H}_{m+2},\ldots,\forall\, h_M\in\mathcal{H}_M$,

$$\left(\mathcal{H}_{m+1}\ni\right)0=o(B_1,\ldots,B_m,k_{m+1},\ldots,k_M)\,(h_{m+2},\ldots,h_M)$$

$$= \left[\int_{\times_{j=1}^M \mathcal{X}_j} \left(\prod_{j=1}^m \chi_{B_j}(x_j)\right) \otimes_{j=m+1}^M k_j(\cdot, x_j)\,\mathrm{d}\mathbb{F}(x)\right](h_{m+2}, \ldots, h_M)$$
$$= \int_{\times_{j=1}^M \mathcal{X}_j} k_{m+1}(\cdot, x_{m+1}) \prod_{j=1}^m \chi_{B_j}(x_j) \prod_{j=m+2}^M h_j(x_j)\,\mathrm{d}\mathbb{F}(x)$$
$$= \int_{\mathcal{X}_{m+1}} k_{m+1}(\cdot, x_{m+1})\,\mathrm{d}\nu(x_{m+1}),$$

where

$$\nu(B) := \nu_{B_1,\ldots,B_m,h_{m+2},\ldots,h_M}(B)$$
$$= \int_{\times_{j=1}^M \mathcal{X}_j} \left[\prod_{j=1}^m \chi_{B_j}(x_j)\right] \chi_B(x_{m+1}) \left[\prod_{j=m+2}^M h_j(x_j)\right] \mathrm{d}\mathbb{F}(x),\ B \in \mathcal{B}(\mathcal{X}_{m+1}).$$

By the c_0-universality of k_{m+1},

$$\nu = 0 \text{ for } \forall\, h_{m+2} \in \mathcal{H}_{m+2}, \ldots, \forall\, h_M \in \mathcal{H}_M \tag{35}$$

provided that $\nu \in \mathcal{M}_b(\mathcal{X}_{m+1})$, in other words if $|\nu|(\mathcal{X}_{m+1}) < \infty$. This condition is met:

$$|\nu|(\mathcal{X}_{m+1}) \le \int_{\times_{j=1}^M \mathcal{X}_j} \prod_{j=m+2}^M \underbrace{\left|\langle h_j, k_j(\cdot, x_j)\rangle_{\mathcal{H}_j}\right|}_{\le \|h_j\|_{\mathcal{H}_j}\sqrt{k_j(x_j,x_j)}} \mathrm{d}|\mathbb{F}|(x)$$
$$\le |\mathbb{F}|\left(\times_{m=1}^M \mathcal{X}_m\right) \prod_{j=m+2}^M \|h_j\|_{\mathcal{H}_j} \sup_{x\in\mathcal{X}_j, x'\in\mathcal{X}_j} \sqrt{k_j(x,x')} < \infty,$$

where we used the boundedness of k_m-s in the last inequality. (35) implies that for $\forall\, B_1 \in \mathcal{B}(\mathcal{X}_1), \ldots, \forall\, B_{m+1} \in \mathcal{B}(\mathcal{X}_{m+1})$ and $\forall\, h_{m+2} \in \mathcal{H}_{m+2}, \ldots, \forall\, h_M \in \mathcal{H}_M$

$$0 = \nu(B_{m+1}) = \int_{\times_{j=1}^M \mathcal{X}_j} \left[\prod_{j=1}^{m+1} \chi_{B_j}(x_j)\right] \left[\prod_{j=m+2}^M h_j(x_j)\right] \mathrm{d}\mathbb{F}(x)$$
$$= \left\langle \otimes_{j=m+2}^M h_j, \int_{\times_{j=1}^M \mathcal{X}_j} \left[\prod_{j=1}^{m+1} \chi_{B_j}(x_j)\right] \otimes_{j=m+2}^M k_j(\cdot, x_j)\,\mathrm{d}\mathbb{F}(x)\right\rangle_{\otimes_{j=m+2}^M \mathcal{H}_j},$$

and therefore

$$o(B_1, \ldots, B_{m+1}, k_{m+2}, \ldots, k_M) = \int_{\times_{j=1}^M \mathcal{X}_j} \left[\prod_{j=1}^{m+1} \chi_{B_j}(x_j)\right] \otimes_{j=m+2}^M k(\cdot, x_j)\,\mathrm{d}\mathbb{F}(x)$$
$$= 0\left(\in \otimes_{j=m+2}^M \mathcal{H}_j\right)$$

for $\forall\, B_1 \in \mathcal{B}(\mathcal{X}_1), \ldots, \forall\, B_{m+1} \in \mathcal{B}(\mathcal{X}_{m+1})$, i.e., (34) holds for $m+1$. Therefore, by induction, (34) holds for $m = M$ and the result follows from (†). To justify the convention in (†), consider the case of $m = M-1$ in which case (34) can be written as

$$\int_{\mathcal{X}_M} k_M(\cdot, x_M)\,\mathrm{d}\nu(x_M) = 0,$$

where

$$\nu(B) = \int_{\times_{j=1}^M \mathcal{X}_j} \left[\prod_{j=1}^{M-1} \chi_{B_j}(x_j)\right] \chi_B(x_M)\,\mathrm{d}\mathbb{F}(x),\ B \in \mathcal{B}(\mathcal{X}_M).$$

Then by the c_0-universal property of k_M, since

$$|\nu|(\mathcal{X}_M) \le \int_{\times_{j=1}^M \mathcal{X}_j} 1\,\mathrm{d}|\mathbb{F}|(x) = |\mathbb{F}|\left(\times_{j=1}^M \mathcal{X}_j\right) < \infty$$

we obtain

$$\int_{\times_{j=1}^M \mathcal{X}_j} \prod_{j=1}^M \chi_{B_j}(x_j)\,\mathrm{d}\mathbb{F}(x) = \mathbb{F}\left(\times_{j=1}^M B_j\right) = 0, \forall\, B_1 \in \mathcal{B}(\mathcal{X}_1), \ldots, \forall\, B_M \in \mathcal{B}(\mathcal{X}_M).$$

5.7 Proof of Example 3

The proof follows by a simple modification of that of Example 2 (Section 5.4). The equations of a witness $\mathbf{A} = [a_{i_1,i_2,i_3}]_{(i_m)_{m=1}^3 \in [2]^3}$ (and corresponding $\mathbf{P} = [p_{i_1,i_2,i_3}]_{(i_m)_{m=1}^3 \in [2]^3}$) for the non-$\mathcal{I}$-characteristic property of $\otimes_{m=1}^3 k_m$ take the form:

$$\mathbf{A} \ne \mathbf{0},$$
$$0 = \sum_{i_1,i_2,i_3=1}^2 a_{i_1,i_2,i_3}, \tag{36}$$
$$0 = \sum_{i_1,i_2,i_3=1}^2 \sum_{i_1',i_2',i_3'=1}^2 \prod_{m=1}^3 k_m(i_m, i_m') a_{i_1,i_2,i_3} a_{i_1',i_2',i_3'}$$
$$= (a_{1,1,1} - a_{2,1,1})^2 + (a_{1,1,2} - a_{2,1,2})^2 + (a_{1,2,1} - a_{2,2,1})^2 + (a_{1,2,2} - a_{2,2,2})^2, \tag{37}$$

where (36) and (37) are equivalent to

$$0 = \sum_{i_1,i_2,i_3=1}^2 a_{i_1,i_2,i_3},\quad a_{1,1,1} = a_{2,1,1},\quad a_{1,1,2} = a_{2,1,2},\quad a_{1,2,1} = a_{2,2,1},\quad a_{1,2,2} = a_{2,2,2}. \tag{38}$$

While (38) is more restrictive than (27) and (28) (hence its solution set might even be empty), one can immediately see that the example of $\mathbf{A} \ne \mathbf{0}$ given in (31) and (32) fulfills (38) proving the non-$\mathcal{I}$-characteristic property of $\otimes_{m=1}^3 k_m$.

Acknowledgments

The authors profusely thank Ingo Steinwart for fascinating discussions on topics related to the paper and for contributing to Remark 7. The authors also thank the anonymous reviewers for their constructive comments that improved the manuscript. A part of the work was carried out while BKS was visiting ZSz at CMAP, École Polytechnique. BKS is supported by NSF-DMS-1713011 and also thanks CMAP and DSI for their generous support. ZSz is highly grateful for the Greek hospitality around the Aegean Sea; it greatly contributed to the development of the induction arguments.

Appendix A. Analytical Solution to (25)–(30) in Example 2

The solution of (25)–(30) takes the form

$$p_{1,1,1} = -\frac{\begin{array}{l} z_2+z_1+z_4+z_5-3z_2z_1-4z_2z_4-4z_1z_4-z_2z_3-2z_2z_0-2z_1z_3-3z_2z_5 \\ -2z_4z_3-z_1z_0-3z_1z_5-2z_4z_0-4z_4z_5-z_3z_0-z_3z_5-z_0z_5+2z_2z_1^2+2z_2^2z_1 \\ +4z_2z_4^2+2z_2^2z_4+4z_1z_4^2+2z_1^2z_4+2z_2^2z_0+2z_1^2z_3+2z_2z_5^2+2z_2^2z_5+2z_4^2z_3 \\ +2z_1z_5^2+2z_1^2z_5+2z_4^2z_0+2z_4z_5^2+4z_4^2z_5-z_2^2-z_1^2-3z_4^2+2z_4^3-z_5^2 \\ +6z_2z_1z_4+2z_2z_1z_3+2z_2z_4z_3+2z_2z_1z_0+4z_2z_1z_5+4z_2z_4z_0+4z_1z_4z_3 \\ +6z_2z_4z_5+2z_1z_4z_0+6z_1z_4z_5+2z_2z_3z_0+2z_2z_3z_5+2z_1z_3z_0+2z_2z_0z_5 \\ +2z_1z_3z_5+2z_4z_3z_0+2z_4z_3z_5+2z_1z_0z_5+2z_4z_0z_5 \end{array}}{\begin{array}{l} 2z_2z_1-z_1-2z_4-z_3-z_0-2z_5-z_2+2z_2z_4+2z_1z_4+2z_2z_0+2z_1z_3+2z_2z_5 \\ +2z_4z_3+2z_1z_5+2z_4z_0+4z_4z_5+2z_3z_0+2z_3z_5+2z_0z_5+2z_4^2+2z_5^2 \end{array}},$$

$$p_{1,1,2} = z_2,$$

$$p_{1,2,1} = z_1,$$

$$p_{1,2,2} = z_4,$$

$$p_{2,1,1} = -\frac{\begin{array}{l} z_4+z_3+z_0+z_5-z_2z_1-z_2z_4-z_1z_4-z_2z_3-2z_2z_0-2z_1z_3-2z_2z_5 \\ -3z_4z_3-z_1z_0-2z_1z_5-3z_4z_0-4z_4z_5-3z_3z_0-4z_3z_5-4z_0z_5+2z_2z_0^2 \\ +2z_1z_3^2+2z_2z_5^2+2z_4z_3^2+2z_4^2z_3+2z_1z_5^2+2z_4z_0^2+2z_4^2z_0+4z_4z_5^2+2z_4^2z_5 \\ +2z_3z_0^2+2z_3^2z_0+4z_3z_5^2+2z_3^2z_5+4z_0z_5^2+2z_0^2z_5-z_4^2-z_3^2-z_0^2-3z_5^2 \\ +2z_5^3+2z_2z_1z_3+2z_2z_4z_3+2z_2z_1z_0+2z_2z_1z_5+2z_2z_4z_0+2z_1z_4z_3 \\ +2z_2z_4z_5+2z_1z_4z_0+2z_1z_4z_5+2z_2z_3z_0+2z_2z_3z_5+2z_1z_3z_0+4z_2z_0z_5 \\ +4z_1z_3z_5+4z_4z_3z_0+6z_4z_3z_5+2z_1z_0z_5+6z_4z_0z_5+6z_3z_0z_5 \end{array}}{\begin{array}{l} 2z_2z_1-z_1-2z_4-z_3-z_0-2z_5-z_2+2z_2z_4+2z_1z_4+2z_2z_0+2z_1z_3+2z_2z_5 \\ +2z_4z_3+2z_1z_5+2z_4z_0+4z_4z_5+2z_3z_0+2z_3z_5+2z_0z_5+2z_4^2+2z_5^2 \end{array}},$$

$$p_{2,1,2} = z_3,$$

$$p_{2,2,1} = z_0,$$

$$p_{2,2,2} = z_5,$$

form, where $\mathbf{z} = (z_0, z_1, \ldots, z_5) \in \mathbb{R}^6$ satisfies

$$\begin{aligned} 0 \leq\ & (2z_0z_2 - z_1 - z_2 - z_3 - 2z_4 - 2z_5 - z_0 + 2z_0z_3 + 2z_1z_2 + 2z_0z_4 + 2z_1z_3 + 2z_0z_5 \\ & +2z_1z_4 + 2z_1z_5 + 2z_2z_4 + 2z_2z_5 + 2z_3z_4 + 2z_3z_5 + 4z_4z_5 + 2z_4^2 + 2z_5^2) \times \\ & (z_0z_3 - z_3 - z_4 - z_5 - z_0z_1 - z_0 - z_1z_2 + z_0z_5 - 2z_1z_4 - z_2z_3 - z_1z_5 - 2z_2z_4 - z_2z_5 \\ & +z_3z_5 + 2z_0z_2^2 + 2z_1z_2^2 + 2z_1^2z_2 + 2z_0z_4^2 + 2z_1^2z_3 + 4z_1z_4^2 + 2z_1^2z_4 + 2z_1z_5^2 + 4z_2z_4^2 \\ & +2z_1^2z_5 + 2z_2^2z_4 + 2z_2z_5^2 + 2z_3z_4^2 + 2z_2^2z_5 + 2z_4z_5^2 + 4z_4^2z_5 - z_1^2 - z_2^2 - z_4^2 + 2z_4^3 + z_5^2 \\ & +2z_0z_1z_2 + 2z_0z_1z_3 + 2z_0z_1z_4 + 2z_0z_2z_3 + 2z_0z_1z_5 + 4z_0z_2z_4 + 2z_1z_2z_3 + 2z_0z_2z_5 \\ & +2z_0z_3z_4 + 6z_1z_2z_4 + 4z_1z_2z_5 + 4z_1z_3z_4 + 2z_0z_4z_5 + 2z_1z_3z_5 + 2z_2z_3z_4 + 6z_1z_4z_5 \\ & +2z_2z_3z_5 + 6z_2z_4z_5 + 2z_3z_4z_5), \end{aligned}$$

$$\begin{aligned} 0 \leq\ & (2z_0z_2 - z_1 - z_2 - z_3 - 2z_4 - 2z_5 - z_0 + 2z_0z_3 + 2z_1z_2 + 2z_0z_4 + 2z_1z_3 + 2z_0z_5 \\ & +2z_1z_4 + 2z_1z_5 + 2z_2z_4 + 2z_2z_5 + 2z_3z_4 + 2z_3z_5 + 4z_4z_5 + 2z_4^2 + 2z_5^2) \times \\ & (z_1z_2 - z_2 - z_4 - z_5 - z_0z_1 - z_0z_3 - z_1 - z_0z_4 - 2z_0z_5 + z_1z_4 - z_2z_3 + z_2z_4 \\ & -z_3z_4 - 2z_3z_5 + 2z_0^2z_2 + 2z_0z_3^2 + 2z_0^2z_3 + 2z_0z_4^2 + 2z_1z_3^2 + 2z_0^2z_4 + 4z_0z_5^2 + 2z_0^2z_5 \\ & +2z_1z_5^2 + 2z_2z_5^2 + 2z_3z_4^2 + 2z_3^2z_4 + 4z_3z_5^2 + 2z_3^2z_5 + 4z_4z_5^2 + 2z_4^2z_5 - z_0^2 - z_3^2 + z_4^2 \\ & -z_5^2 + 2z_5^3 + 2z_0z_1z_2 + 2z_0z_1z_3 + 2z_0z_1z_4 + 2z_0z_2z_3 + 2z_0z_1z_5 + 2z_0z_2z_4 + 2z_1z_2z_3 \\ & +4z_0z_2z_5 + 4z_0z_3z_4 + 6z_0z_3z_5 + 2z_1z_2z_5 + 2z_1z_3z_4 + 6z_0z_4z_5 + 4z_1z_3z_5 + 2z_2z_3z_4 \\ & +2z_1z_4z_5 + 2z_2z_3z_5 + 2z_2z_4z_5 + 6z_3z_4z_5), \end{aligned}$$

$$\begin{aligned} & 2z_0z_2 + 2z_0z_3 + 2z_1z_2 + 2z_0z_4 + 2z_1z_3 + 2z_0z_5 + 2z_1z_4 + 2z_1z_5 + 2z_2z_4 + 2z_2z_5 \\ & \quad + 2z_3z_4 + 2z_3z_5 + 4z_4z_5 + 2z_4^2 + 2z_5^2 \neq z_0 + z_1 + z_2 + z_3 + 2z_4 + 2z_5, \end{aligned}$$

$$\begin{aligned} & (2z_0z_2 - z_1 - z_2 - z_3 - 2z_4 - 2z_5 - z_0 + 2z_0z_3 + 2z_1z_2 + 2z_0z_4 + 2z_1z_3 + 2z_0z_5 \\ & \quad +2z_1z_4 + 2z_1z_5 + 2z_2z_4 + 2z_2z_5 + 2z_3z_4 + 2z_3z_5 + 4z_4z_5 + 2z_4^2 + 2z_5^2) \times \\ & \quad (z_1 + z_2 + z_4 + z_5 - z_0z_1 - 2z_0z_2 - z_0z_3 - 3z_1z_2 - 2z_0z_4 - 2z_1z_3 - z_0z_5 - 4z_1z_4 \\ & \quad -z_2z_3 - 3z_1z_5 - 4z_2z_4 - 3z_2z_5 - 2z_3z_4 - z_3z_5 - 4z_4z_5 + 2z_0z_2^2 + 2z_1z_2^2 + 2z_1^2z_2 \\ & \quad +2z_0z_4^2 + 2z_1^2z_3 + 4z_1z_4^2 + 2z_1^2z_4 + 2z_1z_5^2 + 4z_2z_4^2 + 2z_1^2z_5 + 2z_2^2z_4 + 2z_2z_5^2 \\ & \quad +2z_3z_4^2 + 2z_2^2z_5 + 2z_4z_5^2 + 4z_4^2z_5 - z_1^2 - z_2^2 - 3z_4^2 + 2z_4^3 - z_5^2 + 2z_0z_1z_2 \\ & \quad +2z_0z_1z_3 + 2z_0z_1z_4 + 2z_0z_2z_3 + 2z_0z_1z_5 + 4z_0z_2z_4 + 2z_1z_2z_3 + 2z_0z_2z_5 \\ & \quad +2z_0z_3z_4 + 6z_1z_2z_4 + 4z_1z_2z_5 + 4z_1z_3z_4 + 2z_0z_4z_5 + 2z_1z_3z_5 + 2z_2z_3z_4 + 6z_1z_4z_5 \\ & \quad + 2z_2z_3z_5 + 6z_2z_4z_5 + 2z_3z_4z_5) \leq 0, \end{aligned}$$

$$\begin{aligned} & (2z_0z_2 - z_1 - z_2 - z_3 - 2z_4 - 2z_5 - z_0 + 2z_0z_3 + 2z_1z_2 + 2z_0z_4 + 2z_1z_3 + 2z_0z_5 \\ & \quad +2z_1z_4 + 2z_1z_5 + 2z_2z_4 + 2z_2z_5 + 2z_3z_4 + 2z_3z_5 + 4z_4z_5 + 2z_4^2 + 2z_5^2) \times \\ & \quad (z_0 + z_3 + z_4 + z_5 - z_0z_1 - 2z_0z_2 - 3z_0z_3 - z_1z_2 - 3z_0z_4 - 2z_1z_3 - 4z_0z_5 \\ & \quad -z_1z_4 - z_2z_3 - 2z_1z_5 - z_2z_4 - 2z_2z_5 - 3z_3z_4 - 4z_3z_5 - 4z_4z_5 + 2z_0^2z_2 \end{aligned}$$

$$
\begin{aligned}
&+2z_0z_3^2 + 2z_0^2z_3 + 2z_0z_4^2 + 2z_1z_3^2 + 2z_0^2z_4 + 4z_0z_5^2 + 2z_0^2z_5 + 2z_1z_5^2 + 2z_2z_5^2\\
&+2z_3z_4^2 + 2z_3^2z_4 + 4z_3z_5^2 + 2z_3^2z_5 + 4z_4z_5^2 + 2z_4^2z_5 - z_0^2 - z_3^2 - z_4^2 - 3z_5^2 + 2z_5^3\\
&+2z_0z_1z_2 + 2z_0z_1z_3 + 2z_0z_1z_4 + 2z_0z_2z_3 + 2z_0z_1z_5 + 2z_0z_2z_4 + 2z_1z_2z_3\\
&+4z_0z_2z_5 + 4z_0z_3z_4 + 6z_0z_3z_5 + 2z_1z_2z_5 + 2z_1z_3z_4 + 6z_0z_4z_5 + 4z_1z_3z_5\\
&+2z_2z_3z_4 + 2z_1z_4z_5 + 2z_2z_3z_5 + 2z_2z_4z_5 + 6z_3z_4z_5) \leq 0,
\end{aligned}
$$

and $0 \leq z_0, z_1, z_2, z_3, z_4, z_5 \leq 1$.

The above analytic solution to (25)–(30) is obtained by symbolic math programming in MATLAB.

References

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

K. Balasubramanian, T. Li, and M. Yuan. On the optimality of kernel-embedding based goodness-of-fit tests. Technical report, 2017. (`https://arxiv.org/abs/1709.08148`).

L. Baringhaus and C. Franz. On a new multivariate two-sample test. *Journal of Multivariate Analysis*, 88:190–206, 2004.

A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer, 2004.

G. Blanchard, G. Lee, and C. Scott. Generalizing from several related classification tasks to a new unlabeled sample. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2178–2186, 2011.

G. Blanchard, A. A. Deshmukh, U. Dogan, G. Lee, and C. Scott. Domain generalization by marginal transfer learning. Technical report, 2017. (`https://arxiv.org/abs/1711.07910`).

K. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22:e49–e57, 2006.

J.-F. Cardoso. Multidimensional independent component analysis. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1941–1944, 1998.

C. Carmeli, E. De Vito, A. Toigo, and V. Umanitá. Vector valued reproducing kernel Hilbert spaces and universality. *Analysis and Applications*, 8:19–61, 2010.

K. Chwialkowski, H. Strathmann, and A. Gretton. A kernel test of goodness of fit. In *International Conference on Machine Learning (ICML; PMLR)*, volume 48, pages 2606–2615, 2016.

R. M. Dudley. *Real Analysis and Probability*. Cambridge University Press, 2004.

K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf. Kernel measures of conditional dependence. In *Advances in Neural Information Processing Systems (NIPS)*, pages 498–496, 2008.

K. Fukumizu, F. Bach, and M. Jordan. Kernel dimension reduction in regression. *The Annals of Statistics*, 37(4):1871–1905, 2009.

K. Fukumizu, L. Song, and A. Gretton. Kernel Bayes' rule: Bayesian inference with positive definite kernels. *Journal of Machine Learning Research*, 14:3753–3783, 2013.

A. Gretton. A simpler condition for consistency of a kernel independence test. Technical report, University College London, 2015. (`http://arxiv.org/abs/1501.06103`).

A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In *Algorithmic Learning Theory (ALT)*, pages 63–78, 2005a.

A. Gretton, R. Herbrich, A. Smola, O. Bousquet, and B. Schölkopf. Kernel methods for measuring independence. *Journal of Machine Learning Research*, 6:2075–2129, 2005b.

A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Schölkopf, and A. J. Smola. A kernel statistical test of independence. In *Advances in Neural Information Processing Systems (NIPS)*, pages 585–592, 2008.

A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

Z. Harchaoui, F. Bach, and E. Moulines. Testing for homogeneity with kernel Fisher discriminant analysis. In *Advances in Neural Information Processing Systems (NIPS)*, pages 609–616, 2007.

W. Jitkrittum, Z. Szabó, and A. Gretton. An adaptive test of independence with analytic kernel embeddings. In *International Conference on Machine Learning (ICML; PMLR)*, volume 70, pages 1742–1751, 2017a.

W. Jitkrittum, W. Xu, Z. Szabó, K. Fukumizu, and A. Gretton. A linear-time kernel goodness-of-fit test. In *Advances in Neural Information Processing Systems (NIPS)*, pages 261–270, 2017b.

B. Kim, R. Khanna, and O. O. Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2280–2288, 2016.

L. Klebanov. *N-Distances and Their Applications*. Charles University, Prague, 2005.

G. Kusano, K. Fukumizu, and Y. Hiraoka. Persistence weighted Gaussian kernel for topological data analysis. In *International Conference on Machine Learning (ICML)*, pages 2004–2013, 2016.

H. C. L. Law, D. J. Sutherland, D. Sejdinovic, and S. Flaxman. Bayesian approaches to distribution regression. In *International Conference on Artificial Intelligence and Statistics (AISTATS; PMLR)*, volume 84, pages 1167–1176, 2018.

Q. Liu, J. Lee, and M. Jordan. A kernelized Stein discrepancy for goodness-of-fit tests. In *International Conference on Machine Learning (ICML)*, pages 276–284, 2016.

J. R. Lloyd, D. Duvenaud, R. Grosse, J. B. Tenenbaum, and Z. Ghahramani. Automatic construction and natural-language description of nonparametric regression models. In *AAAI Conference on Artificial Intelligence*, pages 1242–1250, 2014.

R. Lyons. Distance covariance in metric spaces. *The Annals of Probability*, 41:3284–3305, 2013.

C. A. Micchelli, Y. Xu, and H. Zhang. Universal kernels. *Journal of Machine Learning Research*, 7:2651–2667, 2006.

J. M. Mooij, J. Peters, D. Janzing, J. Zscheischler, and B. Schölkopf. Distinguishing cause from effect using observational data: Methods and benchmarks. *Journal of Machine Learning Research*, 17:1–102, 2016.

K. Muandet, K. Fukumizu, F. Dinuzzo, and B. Schölkopf. Learning from distributions via support measure machines. In *Advances in Neural Information Processing Systems (NIPS)*, pages 10–18, 2011.

K. Muandet, K. Fukumizu, B. Sriperumbudur, and B. Schölkopf. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends in Machine Learning*, 10 (1-2):1–141, 2017.

M. Park, W. Jitkrittum, and D. Sejdinovic. K2-ABC: Approximate Bayesian computation with kernel embeddings. In *International Conference on Artificial Intelligence and Statistics (AISTATS; PMLR)*, volume 51, pages 51:398–407, 2016.

N. Pfister, P. Bühlmann, B. Schölkopf, and J. Peters. Kernel-based tests for joint independence. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(1): 5–31, 2017.

N. Quadrianto, L. Song, and A. Smola. Kernelized sorting. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1289–1296, 2009.

B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

B. Schölkopf, K. Muandet, K. Fukumizu, S. Harmeling, and J. Peters. Computing functions of random variables via reproducing kernel Hilbert space representations. *Statistics and Computing*, 25(4):755–766, 2015.

D. Sejdinovic, A. Gretton, and W. Bergsma. A kernel test for three-variable interactions. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1124–1132, 2013a.

D. Sejdinovic, B. K. Sriperumbudur, A. Gretton, and K. Fukumizu. Equivalence of distance-based and RKHS-based statistics in hypothesis testing. *Annals of Statistics*, 41:2263–2291, 2013b.

C.-J. Simon-Gabriel and B. Schölkopf. Kernel distribution embeddings: Universal kernels, characteristic kernels and kernel metrics on distributions. Technical report, Max Planck Institute for Intelligent Systems, 2016. (`https://arxiv.org/abs/1604.05251`).

A. Smola, A. Gretton, L. Song, and B. Schölkopf. A Hilbert space embedding for distributions. In *Algorithmic Learning Theory (ALT)*, pages 13–31, 2007.

L. Song, A. Gretton, D. Bickson, Y. Low, and C. Guestrin. Kernel belief propagation. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 707–715, 2011.

L. Song, A. Smola, A. Gretton, J. Bedo, and K. Borgwardt. Feature selection via dependence maximization. *Journal of Machine Learning Research*, 13:1393–1434, 2012.

B. K. Sriperumbudur, A. Gretton, K. Fukumizu, B. Schölkopf, and G. R. G. Lanckriet. Hilbert space embeddings and metrics on probability measures. *Journal of Machine Learning Research*, 11:1517–1561, 2010.

B. K. Sriperumbudur, K. Fukumizu, and G. R. G. Lanckriet. Universality, characteristic kernels and RKHS embedding of measures. *Journal of Machine Learning Research*, 12: 2389–2410, 2011.

I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 6(3):67–93, 2001.

I. Steinwart and A. Christmann. *Support Vector Machines*. Springer, 2008.

I. Steinwart and J. F. Ziegel. Strictly proper kernel scores and characteristic kernels on compact spaces. Technical report, Faculty for Mathematics and Physics, University of Stuttgart, 2017. (`https://arxiv.org/abs/1712.05279`).

E. V. Strobl, S. Visweswaran, and K. Zhang. Approximate kernel-based conditional independence tests for fast non-parametric causal discovery. Technical report, 2017. (`https://arxiv.org/abs/1702.03877`).

Z. Szabó, B. Sriperumbudur, B. Póczos, and A. Gretton. Learning theory for distribution regression. *Journal of Machine Learning Research*, 17(152):1–40, 2016.

G. J. Székely and M. L. Rizzo. Testing for equal distributions in high dimension. *InterStat*, 5, 2004.

G. J. Székely and M. L. Rizzo. A new test for multivariate normality. *Journal of Multivariate Analysis*, 93:58–80, 2005.

G. J. Székely and M. L. Rizzo. Brownian distance covariance. *The Annals of Applied Statistics*, 3:1236–1265, 2009.

G. J. Székely, M. L. Rizzo, and N. K. Bakirov. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35:2769–2794, 2007.

H. Wendland. *Scattered Data Approximation*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2005.

M. Yamada, Y. Umezu, K. Fukumizu, and I. Takeuchi. Post selection inference with kernels. In *International Conference on Artificial Intelligence and Statistics (AISTATS; PMLR)*, volume 84, pages 152–160, 2018.

M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Póczos, R. R. Salakhutdinov, and A. J. Smola. Deep sets. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3394–3404, 2017.

K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang. Domain adaptation under target and conditional shift. *Journal of Machine Learning Research*, 28(3):819–827, 2013.

A. A. Zinger, A. V. Kakosyan, and L. B. Klebanov. A characterization of distributions by mean values of statistics and certain probabilistic metrics. *Journal of Soviet Mathematics*, 1992.

Journal of Machine Learning Research 18 (2018) 1-78 Submitted 11/17; Published 6/18

Learning Certifiably Optimal Rule Lists for Categorical Data

Elaine Angelino ELAINE@EECS.BERKELEY.EDU
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, Berkeley, CA 94720

Nicholas Larus-Stone NLARUSSTONE@ALUMNI.HARVARD.EDU
Daniel Alabi ALABID@G.HARVARD.EDU
Margo Seltzer MARGO@EECS.HARVARD.EDU
School of Engineering and Applied Sciences
Harvard University, Cambridge, MA 02138

Cynthia Rudin* CYNTHIA@CS.DUKE.EDU
Department of Computer Science and Department of Electrical and Computer Engineering
Duke University, Durham, NC 27708

Editor: Maya Gupta
*To whom correspondence should be addressed.

Abstract

We present the design and implementation of a custom discrete optimization technique for building rule lists over a categorical feature space. Our algorithm produces rule lists with optimal training performance, according to the regularized empirical risk, with a certificate of optimality. By leveraging algorithmic bounds, efficient data structures, and computational reuse, we achieve several orders of magnitude speedup in time and a massive reduction of memory consumption. We demonstrate that our approach produces optimal rule lists on practical problems in seconds. Our results indicate that it is possible to construct optimal sparse rule lists that are approximately as accurate as the COMPAS proprietary risk prediction tool on data from Broward County, Florida, but that are completely interpretable. This framework is a novel alternative to CART and other decision tree methods for interpretable modeling.

Keywords: rule lists, decision trees, optimization, interpretable models, criminal justice applications

1. Introduction

As machine learning continues to gain prominence in socially-important decision-making, the interpretability of predictive models remains a crucial problem. Our goal is to build models that are highly predictive, transparent, and easily understood by humans. We use rule lists, also known as decision lists, to achieve this goal. Rule lists are predictive models composed of if-then statements; these models are interpretable because the rules provide a reason for each prediction (Figure 1).

Constructing rule lists, or more generally, decision trees, has been a challenge for more than 30 years; most approaches use greedy splitting techniques (Rivest, 1987; Breiman et al., 1984; Quinlan, 1993). Recent approaches use Bayesian analysis, either to find a locally optimal solution (Chipman et al., 1998) or to explore the search space (Letham et al., 2015;

if $(age = 18-20)$ **and** $(sex = male)$ **then predict** *yes*
else if $(age = 21-23)$ **and** $(priors = 2-3)$ **then predict** *yes*
else if $(priors > 3)$ **then predict** *yes*
else predict *no*

Figure 1: An example rule list that predicts two-year recidivism for the ProPublica data set, found by CORELS.

Yang et al., 2017). These approaches achieve high accuracy while also managing to run reasonably quickly. However, despite the apparent accuracy of the rule lists generated by these algorithms, there is no way to determine either if the generated rule list is optimal or how close it is to optimal, where optimality is defined with respect to minimization of a regularized loss function.

Optimality is important, because there are societal implications for a lack of optimality. Consider the ProPublica article on the Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) recidivism prediction tool (Larson et al., 2016). It highlights a case where a black box, proprietary predictive model is being used for recidivism prediction. The authors hypothesize that the COMPAS scores are racially biased, but since the model is not transparent, no one (outside of the creators of COMPAS) can determine the reason or extent of the bias (Larson et al., 2016), nor can anyone determine the reason for any particular prediction. By using COMPAS, users implicitly assumed that a transparent model would not be sufficiently accurate for recidivism prediction, *i.e.*, they assumed that a black box model would provide better accuracy. We wondered whether there was indeed no transparent and sufficiently accurate model. Answering this question requires solving a computationally hard problem. Namely, we would like to both find a transparent model that is optimal within a particular pre-determined class of models and produce a certificate of its optimality, with respect to the regularized empirical risk. This would enable one to say, for this problem and model class, with certainty and before resorting to black box methods, whether there exists a transparent model. While there may be differences between training and test performance, finding the simplest model with optimal training performance is prescribed by statistical learning theory.

To that end, we consider the class of rule lists assembled from pre-mined frequent itemsets and search for an optimal rule list that minimizes a regularized risk function, R. This is a hard discrete optimization problem. Brute force solutions that minimize R are computationally prohibitive due to the exponential number of possible rule lists. However, this is a worst case bound that is not realized in practical settings. For realistic cases, it is possible to solve fairly large cases of this problem to optimality, with the careful use of algorithms, data structures, and implementation techniques.

We develop specialized tools from the fields of discrete optimization and artificial intelligence. Specifically, we introduce a special branch-and bound algorithm, called Certifiably Optimal RulE ListS (CORELS), that provides the optimal solution according to the training objective, along with a certificate of optimality. The certificate of optimality means that we can investigate how close other models (*e.g.*, models provided by greedy algorithms) are to optimal.

Within its branch-and-bound procedure, CORELS maintains a lower bound on the minimum value of R that each incomplete rule list can achieve. This allows CORELS to prune an incomplete rule list (and every possible extension) if the bound is larger than the error of the best rule list that it has already evaluated. The use of careful bounding techniques leads to massive pruning of the search space of potential rule lists. The algorithm continues to consider incomplete and complete rule lists until it has either examined or eliminated every rule list from consideration. Thus, CORELS terminates with the optimal rule list and a certificate of optimality.

The efficiency of CORELS depends on how much of the search space our bounds allow us to prune; we seek a tight lower bound on R. The bound we maintain throughout execution is a maximum of several bounds that come in three categories. The first category of bounds are those intrinsic to the rules themselves. This category includes bounds stating that each rule must capture sufficient data; if not, the rule list is provably non-optimal. The second type of bound compares a lower bound on the value of R to that of the current best solution. This allows us to exclude parts of the search space that could never be better than our current solution. Finally, our last type of bound is based on comparing incomplete rule lists that capture the same data and allows us to pursue only the most accurate option. This last class of bounds is especially important—without our use of a novel *symmetry-aware map*, we are unable to solve most problems of reasonable scale. This symmetry-aware map keeps track of the best accuracy over all observed permutations of a given incomplete rule list.

We keep track of these bounds using a modified *prefix tree*, a data structure also known as a trie. Each node in the prefix tree represents an individual rule; thus, each path in the tree represents a rule list such that the final node in the path contains metrics about that rule list. This tree structure, together with a search policy and sometimes a queue, enables a variety of strategies, including breadth-first, best-first, and stochastic search. In particular, we can design different best-first strategies by customizing how we order elements in a priority queue. In addition, we are able to limit the number of nodes in the trie and thereby enable tuning of space-time tradeoffs in a robust manner. This trie structure is a useful way of organizing the generation and evaluation of rule lists.

We evaluated CORELS on a number of publicly available data sets. Our metric of success was 10-fold cross-validated prediction accuracy on a subset of the data. These data sets involve hundreds of rules and thousands of observations. CORELS is generally able to find an optimal rule list in a matter of seconds and certify its optimality within about 10 minutes. We show that we are able to achieve better or similar out-of-sample accuracy on these data sets compared to the popular greedy algorithms, CART and C4.5.

CORELS targets large (not massive) problems, where interpretability and certifiable optimality are important. We illustrate the efficacy of our approach using (1) the ProPublica COMPAS data set (Larson et al., 2016), for the problem of two-year recidivism prediction, and (2) stop-and-frisk data sets from the NYPD (New York Police Department, 2016) and the NYCLU (New York Civil Liberties Union, 2014), to predict whether a weapon will be found on a stopped individual who is frisked or searched. On these data, we produce certifiably optimal, interpretable rule lists that achieve the same accuracy as approaches such as random forests. This calls into question the need for use of a proprietary, black box algorithm for recidivism prediction.

Our work overlaps with the thesis of Larus-Stone (2017). We have also written a preliminary conference version of this article (Angelino et al., 2017), and a report highlighting systems optimizations of our implementation (Larus-Stone et al., 2018); the latter includes additional empirical measurements not presented here.

Our code is at `https://github.com/nlarusstone/corels`, where we provide the C++ implementation we used in our experiments (§6). Kaxiras and Saligrama (2018) have also created an interactive web interface at `https://corels.eecs.harvard.edu`, where a user can upload data and run CORELS from a browser.

2. Related Work

Since every rule list is a decision tree and every decision tree can be expressed as an equivalent rule list, the problem we are solving is a version of the "optimal decision tree" problem, though regularization changes the nature of the problem (as shown through our bounds). The optimal decision tree problem is computationally hard, though since the late 1990's, there has been research on building optimal decision trees using optimization techniques (Bennett and Blue, 1996; Dobkin et al., 1996; Farhangfar et al., 2008). A particularly interesting paper along these lines is that of Nijssen and Fromont (2010), who created a "bottom-up" way to form optimal decision trees. Their method performs an expensive search step, mining all possible leaves (rather than all possible rules), and uses those leaves to form trees. Their method can lead to memory problems, but it is possible that these memory issues can be mitigated using the theorems in this paper.[1] None of these methods used the tight bounds and data structures of CORELS.

Because the optimal decision tree problem is hard, there are a huge number of algorithms such as CART (Breiman et al., 1984) and C4.5 (Quinlan, 1993) that do not perform exploration of the search space beyond greedy splitting. Similarly, there are decision list and associative classification methods that construct rule lists iteratively in a greedy way (Rivest, 1987; Liu et al., 1998; Li et al., 2001; Yin and Han, 2003; Sokolova et al., 2003; Marchand and Sokolova, 2005; Vanhoof and Depaire, 2010; Rudin et al., 2013). Some exploration of the search space is done by Bayesian decision tree methods (Dension et al., 1998; Chipman et al., 2002, 2010) and Bayesian rule-based methods (Letham et al., 2015; Yang et al., 2017). The space of trees of a given depth is much larger than the space of rule lists of that same depth, and the trees within the Bayesian tree algorithms are grown in a top-down greedy way. Because of this, authors of Bayesian tree algorithms have noted that their MCMC chains tend to reach only locally optimal solutions. The RIPPER algorithm (Cohen, 1995) is similar to the Bayesian tree methods in that it grows, prunes, and then locally optimizes. The space of rule lists is smaller than that of trees, and has simpler structure. Consequently, Bayesian rule list algorithms tend to be more successful at escaping local minima and can introduce methods of exploring the search space that exploit this structure—these properties motivate our focus on lists. That said, the tightest bounds for the Bayesian lists (namely, those of Yang et al., 2017, upon whose work we build), are not nearly as tight as those of CORELS.

1. There is no public version of their code for distribution as of this writing.

Tight bounds, on the other hand, have been developed for the (immense) literature on building disjunctive normal form (DNF) models; a good example of this is the work of Rijnbeek and Kors (2010). For models of a given size, since the class of DNF's is a proper subset of decision lists, our framework can be restricted to learn optimal DNF's. The field of DNF learning includes work from the fields of rule learning/induction (*e.g.*, early algorithms by Michalski, 1969; Clark and Niblett, 1989; Frank and Witten, 1998) and associative classification (Vanhoof and Depaire, 2010). Most papers in these fields aim to carefully guide the search through the space of models. If we were to place a restriction on our code to learn DNF's, which would require restricting predictions within the list to the positive class only, we could potentially use methods from rule learning and associative classification to help order CORELS' queue, which would in turn help us eliminate parts of the search space more quickly.

Some of our bounds, including the minimum support bound (§3.7, Theorem 10), come from Rudin and Ertekin (2016), who provide flexible mixed-integer programming (MIP) formulations using the same objective as we use here; MIP solvers in general cannot compete with the speed of CORELS.

CORELS depends on pre-mined rules, which we obtain here via enumeration. The literature on association rule mining is huge, and any method for rule mining could be reasonably substituted.

CORELS' main use is for producing interpretable predictive models. There is a growing interest in interpretable (transparent, comprehensible) models because of their societal importance (see Rüping, 2006; Bratko, 1997; Dawes, 1979; Vellido et al., 2012; Giraud-Carrier, 1998; Holte, 1993; Shmueli, 2010; Huysmans et al., 2011; Freitas, 2014). There are now regulations on algorithmic decision-making in the European Union on the "right to an explanation" (Goodman and Flaxman, 2016) that would legally require interpretability of predictions. There is work in both the DNF literature (Rückert and Raedt, 2008) and decision tree literature (Garofalakis et al., 2000) on building interpretable models. Interpretable models must be so sparse that they need to be heavily optimized; heuristics tend to produce either inaccurate or non-sparse models.

Interpretability has many meanings, and it is possible to extend the ideas in this work to other definitions of interpretability; these rule lists may have exotic constraints that help with ease-of-use. For example, Falling Rule Lists (Wang and Rudin, 2015a) are constrained to have decreasing probabilities down the list, which makes it easier to assess whether an observation is in a high risk subgroup. In parallel to this paper, we have been working on an algorithm for Falling Rule Lists (Chen and Rudin, 2018) with bounds similar to those presented here, but even CORELS' basic support bounds do not hold for the falling case, which is much more complicated. One advantage of the approach taken by Chen and Rudin (2018) is that it can handle class imbalance by weighting the positive and negative classes differently; this extension is possible in CORELS but not addressed here.

The models produced by CORELS are predictive only; they cannot be used for policy-making because they are not causal models, they do not include the costs of true and false positives, nor the cost of gathering information. It is possible to adapt CORELS' framework for causal inference (Wang and Rudin, 2015b), dynamic treatment regimes (Zhang et al., 2015), or cost-sensitive dynamic treatment regimes (Lakkaraju and Rudin, 2017) to

if $(age = 18-20)$ **and** $(sex = male)$ **then predict** *yes*	**if** p_1 **then predict** q_1
else if $(age = 21-23)$ **and** $(priors = 2-3)$ **then predict** *yes*	**else if** p_2 **then predict** q_2
else if $(priors > 3)$ **then predict** *yes*	**else if** p_3 **then predict** q_3
else predict *no*	**else predict** q_0

Figure 2: The rule list $d = (r_1, r_2, r_3, r_0)$. Each rule is of the form $r_k = p_k \rightarrow q_k$, for all $k = 0, \ldots, 3$. We can also express this rule list as $d = (d_p, \delta_p, q_0, K)$, where $d_p = (p_1, p_2, p_3)$, $\delta_p = (1, 1, 1, 1)$, $q_0 = 0$, and $K = 3$. This is the same 3-rule list as in Figure 1, that predicts two-year recidivism for the ProPublica data set.

help with policy design. CORELS could potentially be adapted to handle these kinds of interesting problems.

3. Learning Optimal Rule Lists

In this section, we present our framework for learning certifiably optimal rule lists. First, we define our setting and useful notation (§3.1) and then the objective function we seek to minimize (§3.2). Next, we describe the principal structure of our optimization algorithm (§3.3), which depends on a hierarchically structured objective lower bound (§3.4). We then derive a series of additional bounds that we incorporate into our algorithm, because they enable aggressive pruning of our state space.

3.1 Notation

We restrict our setting to binary classification, where rule lists are Boolean functions; this framework is straightforward to generalize to multi-class classification. Let $\{(x_n, y_n)\}_{n=1}^N$ denote training data, where $x_n \in \{0,1\}^J$ are binary features and $y_n \in \{0,1\}$ are labels. Let $\mathbf{x} = \{x_n\}_{n=1}^N$ and $\mathbf{y} = \{y_n\}_{n=1}^N$, and let $x_{n,j}$ denote the j-th feature of x_n.

A rule list $d = (r_1, r_2, \ldots, r_K, r_0)$ of length $K \geq 0$ is a $(K+1)$-tuple consisting of K distinct association rules, $r_k = p_k \rightarrow q_k$, for $k = 1, \ldots, K$, followed by a default rule r_0. Figure 2 illustrates a rule list, $d = (r_1, r_2, r_3, r_0)$, which for clarity, we sometimes call a K-rule list. An association rule $r = p \rightarrow q$ is an implication corresponding to the conditional statement, "if p, then q." In our setting, an antecedent p is a Boolean assertion that evaluates to either true or false for each datum x_n, and a consequent q is a label prediction. For example, $(x_{n,1} = 0) \wedge (x_{n,3} = 1) \rightarrow (y_n = 1)$ is an association rule. The final default rule r_0 in a rule list can be thought of as a special association rule $p_0 \rightarrow q_0$ whose antecedent p_0 simply asserts true.

Let $d = (r_1, r_2, \ldots, r_K, r_0)$ be a K-rule list, where $r_k = p_k \rightarrow q_k$ for each $k = 0, \ldots, K$. We introduce a useful alternate rule list representation: $d = (d_p, \delta_p, q_0, K)$, where we define $d_p = (p_1, \ldots, p_K)$ to be d's prefix, $\delta_p = (q_1, \ldots, q_K) \in \{0,1\}^K$ gives the label predictions associated with d_p, and $q_0 \in \{0,1\}$ is the default label prediction. For example, for the rule list in Figure 1, we would write $d = (d_p, \delta_p, q_0, K)$, where $d_p = (p_1, p_2, p_3)$, $\delta_p = (1, 1, 1)$, $q_0 = 0$, and $K = 3$. Note that $((), (), q_0, 0)$ is a well-defined rule list with an empty prefix; it is completely defined by a single default rule.

Let $d_p = (p_1, \ldots, p_k, \ldots, p_K)$ be an antecedent list, then for any $k \leq K$, we define $d_p^k = (p_1, \ldots, p_k)$ to be the k-prefix of d_p. For any such k-prefix d_p^k, we say that d_p starts with d_p^k.

For any given space of rule lists, we define $\sigma(d_p)$ to be the set of all rule lists whose prefixes start with d_p:

$$\sigma(d_p) = \{(d'_p, \delta'_p, q'_0, K') : d'_p \text{ starts with } d_p\}. \tag{1}$$

If $d_p = (p_1, \ldots, p_K)$ and $d'_p = (p_1, \ldots, p_K, p_{K+1})$ are two prefixes such that d'_p starts with d_p and extends it by a single antecedent, we say that d_p is the parent of d'_p and that d'_p is a child of d_p.

A rule list d classifies datum x_n by providing the label prediction q_k of the first rule r_k whose antecedent p_k is true for x_n. We say that an antecedent p_k of antecedent list d_p captures x_n in the context of d_p if p_k is the first antecedent in d_p that evaluates to true for x_n. We also say that a prefix captures those data captured by its antecedents; for a rule list $d = (d_p, \delta_p, q_0, K)$, data not captured by the prefix d_p are classified according to the default label prediction q_0.

Let β be a set of antecedents. We define $\text{cap}(x_n, \beta) = 1$ if an antecedent in β captures datum x_n, and 0 otherwise. For example, let d_p and d'_p be prefixes such that d'_p starts with d_p, then d'_p captures all the data that d_p captures:

$$\{x_n : \text{cap}(x_n, d_p)\} \subseteq \{x_n : \text{cap}(x_n, d'_p)\}.$$

Now let d_p be an ordered list of antecedents, and let β be a subset of antecedents in d_p. Let us define $\text{cap}(x_n, \beta \,|\, d_p) = 1$ if β captures datum x_n in the context of d_p, *i.e.*, if the first antecedent in d_p that evaluates to true for x_n is an antecedent in β, and 0 otherwise. Thus, $\text{cap}(x_n, \beta \,|\, d_p) = 1$ only if $\text{cap}(x_n, \beta) = 1$; $\text{cap}(x_n, \beta \,|\, d_p) = 0$ either if $\text{cap}(x_n, \beta) = 0$, or if $\text{cap}(x_n, \beta) = 1$ but there is an antecedent α in d_p, preceding all antecedents in β, such that $\text{cap}(x_n, \alpha) = 1$. For example, if $d_p = (p_1, \ldots, p_k, \ldots, p_K)$ is a prefix, then

$$\text{cap}(x_n, p_k \,|\, d_p) = \left(\bigwedge_{k'=1}^{k-1} \neg \, \text{cap}(x_n, p_{k'}) \right) \wedge \text{cap}(x_n, p_k)$$

indicates whether antecedent p_k captures datum x_n in the context of d_p. Now, define $\text{supp}(\beta, \mathbf{x})$ to be the normalized support of β,

$$\text{supp}(\beta, \mathbf{x}) = \frac{1}{N} \sum_{n=1}^{N} \text{cap}(x_n, \beta), \tag{2}$$

and similarly define $\text{supp}(\beta, \mathbf{x} \,|\, d_p)$ to be the normalized support of β in the context of d_p,

$$\text{supp}(\beta, \mathbf{x} \,|\, d_p) = \frac{1}{N} \sum_{n=1}^{N} \text{cap}(x_n, \beta \,|\, d_p), \tag{3}$$

Next, we address how empirical data constrains rule lists. Given training data $(\mathbf{x}, \mathbf{y})$, an antecedent list $d_p = (p_1, \ldots, p_K)$ implies a rule list $d = (d_p, \delta_p, q_0, K)$ with prefix d_p, where the label predictions $\delta_p = (q_1, \ldots, q_K)$ and q_0 are empirically set to minimize the number of misclassification errors made by the rule list on the training data. Thus for $1 \leq k \leq K$, label prediction q_k corresponds to the majority label of data captured by antecedent p_k in the context of d_p, and the default q_0 corresponds to the majority label of data not captured by d_p. In the remainder of our presentation, whenever we refer to a rule list with a particular prefix, we implicitly assume these empirically determined label predictions.

Our method is technically an associative classification method since it leverages pre-mined rules.

3.2 Objective Function

We define a simple objective function for a rule list $d = (d_p, \delta_p, q_0, K)$:

$$R(d, \mathbf{x}, \mathbf{y}) = \ell(d, \mathbf{x}, \mathbf{y}) + \lambda K. \tag{4}$$

This objective function is a regularized empirical risk; it consists of a loss $\ell(d, \mathbf{x}, \mathbf{y})$, measuring misclassification error, and a regularization term that penalizes longer rule lists. $\ell(d, \mathbf{x}, \mathbf{y})$ is the fraction of training data whose labels are incorrectly predicted by d. In our setting, the regularization parameter $\lambda \geq 0$ is a small constant; *e.g.*, $\lambda = 0.01$ can be thought of as adding a penalty equivalent to misclassifying 1% of data when increasing a rule list's length by one association rule.

3.3 Optimization Framework

Our objective has structure amenable to global optimization via a branch-and-bound framework. In particular, we make a series of important observations, each of which translates into a useful bound, and that together interact to eliminate large parts of the search space. We discuss these in depth in what follows:

- Lower bounds on a prefix also hold for every extension of that prefix. (§3.4, Theorem 1)
- If a rule list is not accurate enough with respect to its length, we can prune all extensions of it. (§3.4, Lemma 2)
- We can calculate *a priori* an upper bound on the maximum length of an optimal rule list. (§3.5, Theorem 6)
- Each rule in an optimal rule list must have support that is sufficiently large. This allows us to construct rule lists from frequent itemsets, while preserving the guarantee that we can find a globally optimal rule list from pre-mined rules. (§3.7, Theorem 10)
- Each rule in an optimal rule list must predict accurately. In particular, the number of observations predicted correctly by each rule in an optimal rule list must be above a threshold. (§3.7, Theorem 11)
- We need only consider the optimal permutation of antecedents in a prefix; we can omit all other permutations. (§3.10, Theorem 15 and Corollary 16)
- If multiple observations have identical features and opposite labels, we know that any model will make mistakes. In particular, the number of mistakes on these observations will be at least the number of observations with the minority label. (§3.14, Theorem 20)

3.4 Hierarchical Objective Lower Bound

We can decompose the misclassification error in (4) into two contributions corresponding to the prefix and the default rule:

$$\ell(d, \mathbf{x}, \mathbf{y}) \equiv \ell_p(d_p, \delta_p, \mathbf{x}, \mathbf{y}) + \ell_0(d_p, q_0, \mathbf{x}, \mathbf{y}),$$

where $d_p = (p_1, \ldots, p_K)$ and $\delta_p = (q_1, \ldots, q_K)$;

$$\ell_p(d_p, \delta_p, \mathbf{x}, \mathbf{y}) = \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} \mathrm{cap}(x_n, p_k \,|\, d_p) \wedge \mathbb{1}[q_k \neq y_n]$$

is the fraction of data captured and misclassified by the prefix, and

$$\ell_0(d_p, q_0, \mathbf{x}, \mathbf{y}) = \frac{1}{N} \sum_{n=1}^{N} \neg\, \mathrm{cap}(x_n, d_p) \wedge \mathbb{1}[q_0 \neq y_n]$$

is the fraction of data not captured by the prefix and misclassified by the default rule. Eliminating the latter error term gives a lower bound $b(d_p, \mathbf{x}, \mathbf{y})$ on the objective,

$$b(d_p, \mathbf{x}, \mathbf{y}) \equiv \ell_p(d_p, \delta_p, \mathbf{x}, \mathbf{y}) + \lambda K \leq R(d, \mathbf{x}, \mathbf{y}), \tag{5}$$

where we have suppressed the lower bound's dependence on label predictions δ_p because they are fully determined, given $(d_p, \mathbf{x}, \mathbf{y})$. Furthermore, as we state next in Theorem 1, $b(d_p, \mathbf{x}, \mathbf{y})$ gives a lower bound on the objective of *any* rule list whose prefix starts with d_p.

Theorem 1 (Hierarchical objective lower bound) *Define $b(d_p, \mathbf{x}, \mathbf{y})$ as in (5). Also, define $\sigma(d_p)$ to be the set of all rule lists whose prefixes starts with d_p, as in (1). Let $d = (d_p, \delta_p, q_0, K)$ be a rule list with prefix d_p, and let $d' = (d'_p, \delta'_p, q'_0, K') \in \sigma(d_p)$ be any rule list such that its prefix d'_p starts with d_p and $K' \geq K$, then $b(d_p, \mathbf{x}, \mathbf{y}) \leq R(d', \mathbf{x}, \mathbf{y})$.*

Proof Let $d_p = (p_1, \ldots, p_K)$ and $\delta_p = (q_1, \ldots, q_K)$; let $d'_p = (p_1, \ldots, p_K, p_{K+1}, \ldots, p_{K'})$ and $\delta'_p = (q_1, \ldots, q_K, q_{K+1}, \ldots, q_{K'})$. Notice that d'_p yields the same mistakes as d_p, and possibly additional mistakes:

$$\begin{aligned}
&\ell_p(d'_p, \delta'_p, \mathbf{x}, \mathbf{y}) = \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K'} \mathrm{cap}(x_n, p_k \,|\, d'_p) \wedge \mathbb{1}[q_k \neq y_n] \\
&= \frac{1}{N} \sum_{n=1}^{N} \left(\sum_{k=1}^{K} \mathrm{cap}(x_n, p_k \,|\, d_p) \wedge \mathbb{1}[q_k \neq y_n] + \sum_{k=K+1}^{K'} \mathrm{cap}(x_n, p_k \,|\, d'_p) \wedge \mathbb{1}[q_k \neq y_n] \right) \\
&= \ell_p(d_p, \delta_p, \mathbf{x}, \mathbf{y}) + \frac{1}{N} \sum_{n=1}^{N} \sum_{k=K+1}^{K'} \mathrm{cap}(x_n, p_k \,|\, d'_p) \wedge \mathbb{1}[q_k \neq y_n] \geq \ell_p(d_p, \delta_p, \mathbf{x}, \mathbf{y}),
\end{aligned} \tag{6}$$

where in the second equality we have used the fact that $\mathrm{cap}(x_n, p_k \,|\, d'_p) = \mathrm{cap}(x_n, p_k \,|\, d_p)$ for $1 \leq k \leq K$. It follows that

$$\begin{aligned}
b(d_p, \mathbf{x}, \mathbf{y}) &= \ell_p(d_p, \delta_p, \mathbf{x}, \mathbf{y}) + \lambda K \\
&\leq \ell_p(d'_p, \delta'_p, \mathbf{x}, \mathbf{y}) + \lambda K' = b(d'_p, \mathbf{x}, \mathbf{y}) \leq R(d', \mathbf{x}, \mathbf{y}).
\end{aligned} \tag{7}$$

Algorithm 1 Branch-and-bound for learning rule lists.

Input: Objective function $R(d, \mathbf{x}, \mathbf{y})$, objective lower bound $b(d_p, \mathbf{x}, \mathbf{y})$, set of antecedents $S = \{s_m\}_{m=1}^{M}$, training data $(\mathbf{x}, \mathbf{y}) = \{(x_n, y_n)\}_{n=1}^{N}$, initial best known rule list d^0 with objective $R^0 = R(d^0, \mathbf{x}, \mathbf{y})$; d^0 could be obtained as output from another (approximate) algorithm, otherwise, $(d^0, R^0) = (\text{null}, 1)$ provide reasonable default values
Output: Provably optimal rule list d^* with minimum objective R^*

```
(d^c, R^c) ← (d^0, R^0)                          ▷ Initialize best rule list and objective
Q ← queue( [ ( ) ] )                             ▷ Initialize queue with empty prefix
while Q not empty do                             ▷ Stop when queue is empty
    d_p ← Q.pop( )                               ▷ Remove prefix d_p from the queue
    d ← (d_p, δ_p, q_0, K)   ▷ Set label predictions δ_p and q_0 to minimize training error
    if b(d_p, x, y) < R^c then                   ▷ Bound: Apply Theorem 1
        R ← R(d, x, y)                           ▷ Compute objective of d_p's rule list d
        if R < R^c then                          ▷ Update best rule list and objective
            (d^c, R^c) ← (d, R)
        end if
        for s in S do                            ▷ Branch: Enqueue d_p's children
            if s not in d_p then
                Q.push( (d_p, s) )
            end if
        end for
    end if
end while
(d*, R*) ← (d^c, R^c)                            ▷ Identify provably optimal solution
```

■

To generalize, consider a sequence of prefixes such that each prefix starts with all previous prefixes in the sequence. It follows that the corresponding sequence of objective lower bounds increases monotonically. This is precisely the structure required and exploited by branch-and-bound, illustrated in Algorithm 1.

Specifically, the objective lower bound in Theorem 1 enables us to prune the state space hierarchically. While executing branch-and-bound, we keep track of the current best (smallest) objective R^c, thus it is a dynamic, monotonically decreasing quantity. If we encounter a prefix d_p with lower bound $b(d_p, \mathbf{x}, \mathbf{y}) \geq R^c$, then by Theorem 1, we do not need to consider *any* rule list $d' \in \sigma(d_p)$ whose prefix d'_p starts with d_p. For the objective of such a rule list, the current best objective provides a lower bound, *i.e.*, $R(d', \mathbf{x}, \mathbf{y}) \geq b(d'_p, \mathbf{x}, \mathbf{y}) \geq b(d_p, \mathbf{x}, \mathbf{y}) \geq R^c$, and thus d' cannot be optimal.

Next, we state an immediate consequence of Theorem 1.

Lemma 2 (Objective lower bound with one-step lookahead) *Let d_p be a K-prefix and let R^c be the current best objective. If $b(d_p, \mathbf{x}, \mathbf{y}) + \lambda \geq R^c$, then for any K'-rule list $d' \in \sigma(d_p)$ whose prefix d'_p starts with d_p and $K' > K$, it follows that $R(d', \mathbf{x}, \mathbf{y}) \geq R^c$.*

Proof By the definition of the lower bound (5), which includes the penalty for longer prefixes,

$$\begin{aligned} R(d'_p, \mathbf{x}, y) \geq b(d'_p, \mathbf{x}, \mathbf{y}) &= \ell_p(d'_p, \delta'_p, \mathbf{x}, \mathbf{y}) + \lambda K' \\ &= \ell_p(d'_p, \delta'_p, \mathbf{x}, \mathbf{y}) + \lambda K + \lambda(K' - K) \\ &= b(d_p, \mathbf{x}, \mathbf{y}) + \lambda(K' - K) \geq b(d_p, \mathbf{x}, \mathbf{y}) + \lambda \geq R^c. \end{aligned} \tag{8}$$

■

Therefore, even if we encounter a prefix d_p with lower bound $b(d_p, \mathbf{x}, \mathbf{y}) \leq R^c$, as long as $b(d_p, \mathbf{x}, \mathbf{y}) + \lambda \geq R^c$, then we can prune all prefixes d'_p that start with and are longer than d_p.

3.5 Upper Bounds on Prefix Length

In this section, we derive several upper bounds on prefix length:

- The simplest upper bound on prefix length is given by the total number of available antecedents. (Proposition 3)
- The current best objective R^c implies an upper bound on prefix length. (Theorem 4)
- For intuition, we state a version of the above bound that is valid at the start of execution. (Corollary 5)
- By considering specific families of prefixes, we can obtain tighter bounds on prefix length. (Theorem 6)

In the next section (§3.6), we use these results to derive corresponding upper bounds on the number of prefix evaluations made by Algorithm 1.

Proposition 3 (Trivial upper bound on prefix length) *Consider a state space of all rule lists formed from a set of M antecedents, and let $L(d)$ be the length of rule list d. M provides an upper bound on the length of any optimal rule list $d^* \in \operatorname{argmin}_d R(d, \mathbf{x}, \mathbf{y})$, i.e., $L(d) \leq M$.*

Proof Rule lists consist of distinct rules by definition. ■

At any point during branch-and-bound execution, the current best objective R^c implies an upper bound on the maximum prefix length we might still have to consider.

Theorem 4 (Upper bound on prefix length) *Consider a state space of all rule lists formed from a set of M antecedents. Let $L(d)$ be the length of rule list d and let R^c be the current best objective. For all optimal rule lists $d^* \in \operatorname{argmin}_d R(d, \mathbf{x}, \mathbf{y})$*

$$L(d^*) \leq \min\left(\left\lfloor \frac{R^c}{\lambda} \right\rfloor, M\right), \tag{9}$$

where λ is the regularization parameter. Furthermore, if d^c is a rule list with objective $R(d^c, \mathbf{x}, \mathbf{y}) = R^c$, length K, and zero misclassification error, then for every optimal rule list $d^* \in \operatorname{argmin}_d R(d, \mathbf{x}, \mathbf{y})$, if $d^c \in \operatorname{argmin}_d R(d, \mathbf{x}, \mathbf{y})$, then $L(d^*) \leq K$, or otherwise if $d^c \notin \operatorname{argmin}_d R(d, \mathbf{x}, \mathbf{y})$, then $L(d^*) \leq K - 1$.

Proof For an optimal rule list d^* with objective R^*,

$$\lambda L(d^*) \leq R^* = R(d^*, \mathbf{x}, \mathbf{y}) = \ell(d^*, \mathbf{x}, \mathbf{y}) + \lambda L(d^*) \leq R^c.$$

The maximum possible length for d^* occurs when $\ell(d^*, \mathbf{x}, \mathbf{y})$ is minimized; combining with Proposition 3 gives bound (9).

For the rest of the proof, let $K^* = L(d^*)$ be the length of d^*. If the current best rule list d^c has zero misclassification error, then

$$\lambda K^* \leq \ell(d^*, \mathbf{x}, \mathbf{y}) + \lambda K^* = R(d^*, \mathbf{x}, \mathbf{y}) \leq R^c = R(d^c, \mathbf{x}, \mathbf{y}) = \lambda K,$$

and thus $K^* \leq K$. If the current best rule list is suboptimal, *i.e.*, $d^c \notin \operatorname{argmin}_d R(d, \mathbf{x}, \mathbf{y})$, then

$$\lambda K^* \leq \ell(d^*, \mathbf{x}, \mathbf{y}) + \lambda K^* = R(d^*, \mathbf{x}, \mathbf{y}) < R^c = R(d^c, \mathbf{x}, \mathbf{y}) = \lambda K,$$

in which case $K^* < K$, *i.e.*, $K^* \leq K - 1$, since K is an integer. ■

The latter part of Theorem 4 tells us that if we only need to identify a single instance of an optimal rule list $d^* \in \operatorname{argmin}_d R(d, \mathbf{x}, \mathbf{y})$, and we encounter a perfect K-rule list with zero misclassification error, then we can prune all prefixes of length K or greater.

Corollary 5 (Simple upper bound on prefix length) *Let $L(d)$ be the length of rule list d. For all optimal rule lists $d^* \in \operatorname{argmin}_d R(d, \mathbf{x}, \mathbf{y})$,*

$$L(d^*) \leq \min\left(\left\lfloor \frac{1}{2\lambda} \right\rfloor, M\right). \tag{10}$$

Proof Let $d = ((), (), q_0, 0)$ be the empty rule list; it has objective $R(d, \mathbf{x}, \mathbf{y}) = \ell(d, \mathbf{x}, \mathbf{y}) \leq 1/2$, which gives an upper bound on R^c. Combining with (9) and Proposition 3 gives (10). ■

For any particular prefix d_p, we can obtain potentially tighter upper bounds on prefix length for the family of all prefixes that start with d_p.

Theorem 6 (Prefix-specific upper bound on prefix length) *Let $d = (d_p, \delta_p, q_0, K)$ be a rule list, let $d' = (d'_p, \delta'_p, q'_0, K') \in \sigma(d_p)$ be any rule list such that d'_p starts with d_p, and let R^c be the current best objective. If d'_p has lower bound $b(d'_p, \mathbf{x}, \mathbf{y}) < R^c$, then*

$$K' < \min\left(K + \left\lfloor \frac{R^c - b(d_p, \mathbf{x}, \mathbf{y})}{\lambda} \right\rfloor, M\right). \tag{11}$$

Proof First, note that $K' \geq K$, since d'_p starts with d_p. Now recall from (7) that

$$b(d_p, \mathbf{x}, \mathbf{y}) = \ell_p(d_p, \delta_p, \mathbf{x}, \mathbf{y}) + \lambda K \leq \ell_p(d'_p, \delta'_p, \mathbf{x}, \mathbf{y}) + \lambda K' = b(d'_p, \mathbf{x}, \mathbf{y}),$$

and from (6) that $\ell_p(d_p, \delta_p, \mathbf{x}, \mathbf{y}) \leq \ell_p(d'_p, \delta'_p, \mathbf{x}, \mathbf{y})$. Combining these bounds and rearranging gives

$$\begin{aligned} b(d'_p, \mathbf{x}, \mathbf{y}) &= \ell_p(d'_p, \delta'_p, \mathbf{x}, \mathbf{y}) + \lambda K + \lambda(K' - K) \\ &\geq \ell_p(d_p, \delta_p, \mathbf{x}, \mathbf{y}) + \lambda K + \lambda(K' - K) = b(d_p, \mathbf{x}, \mathbf{y}) + \lambda(K' - K). \end{aligned} \tag{12}$$

Combining (12) with $b(d'_p, \mathbf{x}, \mathbf{y}) < R^c$ and Proposition 3 gives (11). ■

We can view Theorem 6 as a generalization of our one-step lookahead bound (Lemma 2), as (11) is equivalently a bound on $K' - K$, an upper bound on the number of remaining 'steps' corresponding to an iterative sequence of single-rule extensions of a prefix d_p. Notice that when $d = ((), (), q_0, 0)$ is the empty rule list, this bound replicates (9), since $b(d_p, \mathbf{x}, \mathbf{y}) = 0$.

3.6 Upper Bounds on the Number of Prefix Evaluations

In this section, we use our upper bounds on prefix length from §3.5 to derive corresponding upper bounds on the number of prefix evaluations made by Algorithm 1. First, we present Theorem 7, in which we use information about the state of Algorithm 1's execution to calculate, for any given execution state, upper bounds on the number of additional prefix evaluations that might be required for the execution to complete. The relevant execution state depends on the current best objective R^c and information about prefixes we are planning to evaluate, *i.e.*, prefixes in the queue Q of Algorithm 1. We define the number of *remaining prefix evaluations* as the number of prefixes that are currently in or will be inserted into the queue.

We use Theorem 7 in some of our empirical results (§6, Figure 18) to help illustrate the dramatic impact of certain algorithm optimizations. The execution trace of this upper bound on remaining prefix evaluations complements the execution traces of other quantities, *e.g.*, that of the current best objective R^c. After presenting Theorem 7, we also give two weaker propositions that provide useful intuition. In particular, Proposition 9 is a practical approximation to Theorem 7 that is significantly easier to compute; we use it in our implementation as a metric of execution progress that we display to the user.

Theorem 7 (Fine-grained upper bound on remaining prefix evaluations) *Consider the state space of all rule lists formed from a set of M antecedents, and consider Algorithm 1 at a particular instant during execution. Let R^c be the current best objective, let Q be the queue, and let $L(d_p)$ be the length of prefix d_p. Define $\Gamma(R^c, Q)$ to be the number of remaining prefix evaluations, then*

$$\Gamma(R^c, Q) \leq \sum_{d_p \in Q} \sum_{k=0}^{f(d_p)} \frac{(M - L(d_p))!}{(M - L(d_p) - k)!}, \tag{13}$$

where

$$f(d_p) = \min\left(\left\lfloor \frac{R^c - b(d_p, \mathbf{x}, \mathbf{y})}{\lambda} \right\rfloor, M - L(d_p)\right).$$

Proof The number of remaining prefix evaluations is equal to the number of prefixes that are currently in or will be inserted into queue Q. For any such prefix d_p, Theorem 6 gives an upper bound on the length of any prefix d'_p that starts with d_p:

$$L(d'_p) \leq \min\left(L(d_p) + \left\lfloor \frac{R^c - b(d_p, \mathbf{x}, \mathbf{y})}{\lambda} \right\rfloor, M\right) \equiv U(d_p).$$

This gives an upper bound on the number of remaining prefix evaluations:

$$\Gamma(R^c, Q) \leq \sum_{d_p \in Q} \sum_{k=0}^{U(d_p) - L(d_p)} P(M - L(d_p), k) = \sum_{d_p \in Q} \sum_{k=0}^{f(d_p)} \frac{(M - L(d_p))!}{(M - L(d_p) - k)!},$$

where $P(m, k)$ denotes the number of k-permutations of m. ■

Proposition 8 is strictly weaker than Theorem 7 and is the starting point for its derivation. It is a naïve upper bound on the total number of prefix evaluations over the course of Algorithm 1's execution. It only depends on the number of rules and the regularization parameter λ; *i.e.*, unlike Theorem 7, it does not use algorithm execution state to bound the size of the search space.

Proposition 8 (Upper bound on the total number of prefix evaluations) *Define $\Gamma_{\text{tot}}(S)$ to be the total number of prefixes evaluated by Algorithm 1, given the state space of all rule lists formed from a set S of M rules. For any set S of M rules,*

$$\Gamma_{\text{tot}}(S) \leq \sum_{k=0}^{K} \frac{M!}{(M - k)!},$$

where $K = \min(\lfloor 1/2\lambda \rfloor, M)$.

Proof By Corollary 5, $K \equiv \min(\lfloor 1/2\lambda \rfloor, M)$ gives an upper bound on the length of any optimal rule list. Since we can think of our problem as finding the optimal selection and permutation of k out of M rules, over all $k \leq K$,

$$\Gamma_{\text{tot}}(S) \leq 1 + \sum_{k=1}^{K} P(M, k) = \sum_{k=0}^{K} \frac{M!}{(M - k)!}.$$

■

Our next upper bound is strictly tighter than the bound in Proposition 8. Like Theorem 7, it uses the current best objective and information about the lengths of prefixes in the

queue to constrain the lengths of prefixes in the remaining search space. However, Proposition 9 is weaker than Theorem 7 because it leverages only coarse-grained information from the queue. Specifically, Theorem 7 is strictly tighter because it additionally incorporates prefix-specific objective lower bound information from prefixes in the queue, which further constrains the lengths of prefixes in the remaining search space.

Proposition 9 (Coarse-grained upper bound on remaining prefix evaluations) *Consider a state space of all rule lists formed from a set of M antecedents, and consider Algorithm 1 at a particular instant during execution. Let R^c be the current best objective, let Q be the queue, and let $L(d_p)$ be the length of prefix d_p. Let Q_j be the number of prefixes of length j in Q,*

$$Q_j = \left|\{d_p : L(d_p) = j, d_p \in Q\}\right|$$

and let $J = \operatorname{argmax}_{d_p \in Q} L(d_p)$ be the length of the longest prefix in Q. Define $\Gamma(R^c, Q)$ to be the number of remaining prefix evaluations, then

$$\Gamma(R^c, Q) \leq \sum_{j=1}^{J} Q_j \left(\sum_{k=0}^{K-j} \frac{(M-j)!}{(M-j-k)!} \right),$$

where $K = \min(\lfloor R^c/\lambda \rfloor, M)$.

Proof The number of remaining prefix evaluations is equal to the number of prefixes that are currently in or will be inserted into queue Q. For any such remaining prefix d_p, Theorem 4 gives an upper bound on its length; define K to be this bound: $L(d_p) \leq \min(\lfloor R^c/\lambda \rfloor, M) \equiv K$. For any prefix d_p in queue Q with length $L(d_p) = j$, the maximum number of prefixes that start with d_p and remain to be evaluated is:

$$\sum_{k=0}^{K-j} P(M-j, k) = \sum_{k=0}^{K-j} \frac{(M-j)!}{(M-j-k)!},$$

where $P(T, k)$ denotes the number of k-permutations of T. This gives an upper bound on the number of remaining prefix evaluations:

$$\Gamma(R^c, Q) \leq \sum_{j=0}^{J} Q_j \left(\sum_{k=0}^{K-j} P(M-j, k) \right) = \sum_{j=0}^{J} Q_j \left(\sum_{k=0}^{K-j} \frac{(M-j)!}{(M-j-k)!} \right).$$

■

3.7 Lower Bounds on Antecedent Support

In this section, we give two lower bounds on the normalized support of each antecedent in any optimal rule list; both are related to the regularization parameter λ.

Theorem 10 (Lower bound on antecedent support) *Let $d^* = (d_p, \delta_p, q_0, K)$ be any optimal rule list with objective R^*, i.e., $d^* \in \operatorname{argmin}_d R(d, \mathbf{x}, \mathbf{y})$. For each antecedent p_k in prefix $d_p = (p_1, \ldots, p_K)$, the regularization parameter λ provides a lower bound on the normalized support of p_k,*

$$\lambda \leq \operatorname{supp}(p_k, \mathbf{x} \,|\, d_p). \tag{14}$$

Proof Let $d^* = (d_p, \delta_p, q_0, K)$ be an optimal rule list with prefix $d_p = (p_1, \ldots, p_K)$ and labels $\delta_p = (q_1, \ldots, q_K)$. Consider the rule list $d = (d'_p, \delta'_p, q'_0, K-1)$ derived from d^* by deleting a rule $p_i \to q_i$, therefore $d'_p = (p_1, \ldots, p_{i-1}, p_{i+1}, \ldots, p_K)$ and $\delta'_p = (q_1, \ldots, q_{i-1}, q'_{i+1}, \ldots, q'_K)$, where q'_k need not be the same as q_k, for $k > i$ and $k = 0$.

The largest possible discrepancy between d^* and d would occur if d^* correctly classified all the data captured by p_i, while d misclassified these data. This gives an upper bound:

$$\begin{aligned} R(d, \mathbf{x}, \mathbf{y}) = \ell(d, \mathbf{x}, \mathbf{y}) + \lambda(K-1) &\leq \ell(d^*, \mathbf{x}, \mathbf{y}) + \operatorname{supp}(p_i, \mathbf{x} \,|\, d_p) + \lambda(K-1) \\ &= R(d^*, \mathbf{x}, \mathbf{y}) + \operatorname{supp}(p_i, \mathbf{x} \,|\, d_p) - \lambda \\ &= R^* + \operatorname{supp}(p_i, \mathbf{x} \,|\, d_p) - \lambda \end{aligned} \tag{15}$$

where $\operatorname{supp}(p_i, \mathbf{x} \,|\, d_p)$ is the normalized support of p_i in the context of d_p, defined in (3), and the regularization 'bonus' comes from the fact that d is one rule shorter than d^*.

At the same time, we must have $R^* \leq R(d, \mathbf{x}, \mathbf{y})$ for d^* to be optimal. Combining this with (15) and rearranging gives (14), therefore the regularization parameter λ provides a lower bound on the support of an antecedent p_i in an optimal rule list d^*. ■

Thus, we can prune a prefix d_p if any of its antecedents captures less than a fraction λ of data, even if $b(d_p, \mathbf{x}, \mathbf{y}) < R^*$. Notice that the bound in Theorem 10 depends on the antecedents, but not the label predictions, and thus does not account for misclassification error. Theorem 11 gives a tighter bound by leveraging this additional information, which specifically tightens the upper bound on $R(d, \mathbf{x}, \mathbf{y})$ in (15).

Theorem 11 (Lower bound on accurate antecedent support) *Let d^* be any optimal rule list with objective R^*, i.e., $d^* = (d_p, \delta_p, q_0, K) \in \operatorname{argmin}_d R(d, \mathbf{x}, \mathbf{y})$. Let d^* have prefix $d_p = (p_1, \ldots, p_K)$ and labels $\delta_p = (q_1, \ldots, q_K)$. For each rule $p_k \to q_k$ in d^*, define a_k to be the fraction of data that are captured by p_k and correctly classified:*

$$a_k \equiv \frac{1}{N} \sum_{n=1}^{N} \operatorname{cap}(x_n, p_k \,|\, d_p) \wedge \mathbb{1}[q_k = y_n]. \tag{16}$$

The regularization parameter λ provides a lower bound on a_k:

$$\lambda \leq a_k. \tag{17}$$

Proof As in Theorem 10, let $d = (d'_p, \delta'_p, q'_0, K-1)$ be the rule list derived from d^* by deleting a rule $p_i \to q_i$. Now, let us define ℓ_i to be the portion of R^* due to this rule's misclassification error,

$$\ell_i \equiv \frac{1}{N} \sum_{n=1}^{N} \operatorname{cap}(x_n, p_i \,|\, d_p) \wedge \mathbb{1}[q_i \neq y_n].$$

The largest discrepancy between d^* and d would occur if d misclassified all the data captured by p_i. This gives an upper bound on the difference between the misclassification error of d and d^*:

$$\begin{aligned}\ell(d,\mathbf{x},\mathbf{y}) - \ell(d^*,\mathbf{x},\mathbf{y}) &\leq \text{supp}(p_i,\mathbf{x}\,|\,d_p) - \ell_i \\ &= \frac{1}{N}\sum_{n=1}^{N}\text{cap}(x_n,p_i\,|\,d_p) - \frac{1}{N}\sum_{n=1}^{N}\text{cap}(x_n,p_i\,|\,d_p)\wedge\mathbb{1}[q_i\neq y_n] \\ &= \frac{1}{N}\sum_{n=1}^{N}\text{cap}(x_n,p_i\,|\,d_p)\wedge\mathbb{1}[q_i= y_n] = a_i,\end{aligned}$$

where we defined a_i in (16). Relating this bound to the objectives of d and d^* gives

$$\begin{aligned}R(d,\mathbf{x},\mathbf{y}) = \ell(d,\mathbf{x},\mathbf{y}) + \lambda(K-1) &\leq \ell(d^*,\mathbf{x},\mathbf{y}) + a_i + \lambda(K-1) \\ &= R(d^*,\mathbf{x},\mathbf{y}) + a_i - \lambda \\ &= R^* + a_i - \lambda. \qquad (18)\end{aligned}$$

Combining (18) with the requirement $R^* \leq R(d,\mathbf{x},\mathbf{y})$ gives the bound $\lambda \leq a_i$. ■

Thus, we can prune a prefix if any of its rules correctly classifies less than a fraction λ of data. While the lower bound in Theorem 10 is a sub-condition of the lower bound in Theorem 11, we can still leverage both—since the sub-condition is easier to check, checking it first can accelerate pruning. In addition to applying Theorem 10 in the context of constructing rule lists, we can furthermore apply it in the context of rule mining (§3.1). Specifically, it implies that we should only mine rules with normalized support of at least λ; we need not mine rules with a smaller fraction of observations.[2] In contrast, we can only apply Theorem 11 in the context of constructing rule lists; it depends on the misclassification error associated with each rule in a rule list, thus it provides a lower bound on the number of observations that each such rule must correctly classify.

3.8 Upper Bound on Antecedent Support

In the previous section (§3.7), we proved lower bounds on antecedent support; in Appendix A, we give an upper bound on antecedent support. Specifically, Theorem 21 shows that an antecedent's support in a rule list cannot be too similar to the set of data not captured by preceding antecedents in the rule list. In particular, Theorem 21 implies that we should only mine rules with normalized support less than or equal to $1-\lambda$; we need not mine rules with a larger fraction of observations. Note that we do not otherwise use this bound in our implementation, because we did not observe a meaningful benefit in preliminary experiments.

3.9 Antecedent Rejection and its Propagation

In this section, we demonstrate further consequences of our lower (§3.7) and upper bounds (§3.8) on antecedent support, under a unified framework we refer to as antecedent rejection. Let $d_p = (p_1,\ldots,p_K)$ be a prefix, and let p_k be an antecedent in d_p. Define p_k to have

2. We describe our application of this idea in Appendix E, where we provide details on data processing.

insufficient support in d_p if it does not obey the bound in (14) of Theorem 10. Define p_k to have insufficient accurate support in d_p if it does not obey the bound in (17) of Theorem 11. Define p_k to have excessive support in d_p if it does not obey the bound in (37) of Theorem 21 (Appendix A). If p_k in the context of d_p has insufficient support, insufficient accurate support, or excessive support, let us say that prefix d_p rejects antecedent p_K. Next, in Theorem 12, we describe large classes of related rule lists whose prefixes all reject the same antecedent.

Theorem 12 (Antecedent rejection propagates) *For any prefix $d_p = (p_1,\ldots,p_K)$, let $\phi(d_p)$ denote the set of all prefixes d'_p such that the set of all antecedents in d_p is a subset of the set of all antecedents in d'_p, i.e.,*

$$\phi(d_p) = \{d'_p = (p'_1,\ldots,p'_{K'}) \text{ s.t. } \{p_k : p_k \in d_p\} \subseteq \{p'_\kappa : p'_\kappa \in d'_p\}, K' \geq K\}. \qquad (19)$$

Let $d = (d_p, \delta_p, q_0, K)$ be a rule list with prefix $d_p = (p_1,\ldots,p_{K-1},p_K)$, such that d_p rejects its last antecedent p_K, either because p_K in the context of d_p has insufficient support, insufficient accurate support, or excessive support. Let $d_p^{K-1} = (p_1,\ldots,p_{K-1})$ be the first $K-1$ antecedents of d_p. Let $D = (D_p, \Delta_p, Q_0, \kappa)$ be any rule list with prefix $D_p = (P_1,\ldots,P_{K'-1}, P_{K'},\ldots,P_\kappa)$ such that D_p starts with $D_p^{K'-1} = (P_1,\ldots,P_{K'-1}) \in \phi(d_p^{K-1})$ and antecedent $P_{K'} = p_K$. It follows that prefix D_p rejects $P_{K'}$ for the same reason that d_p rejects p_K, and furthermore, D cannot be optimal, i.e., $D \notin \text{argmin}_{d^\dagger} R(d^\dagger,\mathbf{x},\mathbf{y})$.

Proof Combine Proposition 13, Proposition 14, and Proposition 22. The first two are found below, and the last in Appendix A. ■

Theorem 12 implies potentially significant computational savings. We know from Theorems 10, 11, and 21 that during branch-and-bound execution, if we ever encounter a prefix $d_p = (p_1,\ldots,p_{K-1},p_K)$ that rejects its last antecedent p_K, then we can prune d_p. By Theorem 12, we can also prune *any* prefix d'_p whose antecedents contains the set of antecedents in d_p, in almost any order, with the constraint that all antecedents in $\{p_1,\ldots,p_{K-1}\}$ precede p_K. These latter antecedents are also rejected directly by the bounds in Theorems 10, 11, and 21; this is how our implementation works in practice. In a preliminary implementation (not shown), we maintained additional data structures to support the direct use of Theorem 12. We leave the design of efficient data structures for this task as future work.

Proposition 13 (Insufficient antecedent support propagates) *First define $\phi(d_p)$ as in (19), and let $d_p = (p_1,\ldots,p_{K-1},p_K)$ be a prefix, such that its last antecedent p_K has insufficient support, i.e., the opposite of the bound in (14): $\text{supp}(p_K,\mathbf{x}\,|\,d_p) < \lambda$. Let $d_p^{K-1} = (p_1,\ldots,p_{K-1})$, and let $D = (D_p,\Delta_p,Q_0,\kappa)$ be any rule list with prefix $D_p = (P_1,\ldots,P_{K'-1}, P_{K'},\ldots,P_\kappa)$, such that D_p starts with $D_p^{K'-1} = (P_1,\ldots,P_{K'-1}) \in \phi(d_p^{K-1})$ and $P_{K'} = p_K$. It follows that $P_{K'}$ has insufficient support in prefix D_p, and furthermore, D cannot be optimal, i.e., $D \notin \text{argmin}_d R(d,\mathbf{x},\mathbf{y})$.*

Proof The support of p_K in d_p depends only on the set of antecedents in $d_p^K = (p_1, \ldots, p_K)$:

$$\begin{aligned}\text{supp}(p_K, \mathbf{x} \,|\, d_p) = \frac{1}{N}\sum_{n=1}^{N} \text{cap}(x_n, p_K \,|\, d_p) &= \frac{1}{N}\sum_{n=1}^{N} \left(\neg \text{cap}(x_n, d_p^{K-1})\right) \wedge \text{cap}(x_n, p_K) \\ &= \frac{1}{N}\sum_{n=1}^{N} \left(\bigwedge_{k=1}^{K-1} \neg \text{cap}(x_n, p_k)\right) \wedge \text{cap}(x_n, p_K) < \lambda,\end{aligned}$$

and the support of $P_{K'}$ in D_p depends only on the set of antecedents in $D_p^{K'} = (P_1, \ldots, P_{K'})$:

$$\begin{aligned}\text{supp}(P_{K'}, \mathbf{x} \,|\, D_p) = \frac{1}{N}\sum_{n=1}^{N} \text{cap}(x_n, P_{K'} \,|\, D_p) &= \frac{1}{N}\sum_{n=1}^{N} \left(\bigwedge_{k=1}^{K'-1} \neg \text{cap}(x_n, P_k)\right) \wedge \text{cap}(x_n, P_{K'}) \\ &\leq \frac{1}{N}\sum_{n=1}^{N} \left(\bigwedge_{k=1}^{K-1} \neg \text{cap}(x_n, p_k)\right) \wedge \text{cap}(x_n, P_{K'}) \\ &= \frac{1}{N}\sum_{n=1}^{N} \left(\bigwedge_{k=1}^{K-1} \neg \text{cap}(x_n, p_k)\right) \wedge \text{cap}(x_n, p_K) \\ &= \text{supp}(p_K, \mathbf{x} \,|\, d_p) < \lambda. \qquad (20)\end{aligned}$$

The first inequality reflects the condition that $D_p^{K'-1} \in \phi(d_p^{K-1})$, which implies that the set of antecedents in $D_p^{K'-1}$ contains the set of antecedents in d_p^{K-1}, and the next equality reflects the fact that $P_{K'} = p_K$. Thus, P'_K has insufficient support in prefix D_p, therefore by Theorem 10, D cannot be optimal, *i.e.*, $D \notin \text{argmin}_d R(d, \mathbf{x}, \mathbf{y})$. ■

Proposition 14 (Insufficient accurate antecedent support propagates) *Let $\phi(d_p)$ denote the set of all prefixes d'_p such that the set of all antecedents in d_p is a subset of the set of all antecedents in d'_p, as in* (19)*. Let $d = (d_p, \delta_p, q_0, K)$ be a rule list with prefix $d_p = (p_1, \ldots, p_K)$ and labels $\delta_p = (q_1, \ldots, q_K)$, such that the last antecedent p_K has insufficient accurate support, i.e., the opposite of the bound in* (17)*:*

$$\frac{1}{N}\sum_{n=1}^{N} \text{cap}(x_n, p_K \,|\, d_p) \wedge \mathbb{1}[q_K = y_n] < \lambda.$$

Let $d_p^{K-1} = (p_1, \ldots, p_{K-1})$ and let $D = (D_p, \Delta_p, Q_0, \kappa)$ be any rule list with prefix $D_p = (P_1, \ldots, P_\kappa)$ and labels $\Delta_p = (Q_1, \ldots, Q_\kappa)$, such that D_p starts with $D_p^{K'-1} = (P_1, \ldots, P_{K'-1}) \in \phi(d_p^{K-1})$ and $P_{K'} = p_K$. It follows that $P_{K'}$ has insufficient accurate support in prefix D_p, and furthermore, $D \notin \text{argmin}_{d^\dagger} R(d^\dagger, \mathbf{x}, \mathbf{y})$.

Proof The accurate support of $P_{K'}$ in D_p is insufficient:

$$\begin{aligned}&\frac{1}{N}\sum_{n=1}^{N} \text{cap}(x_n, P_{K'} \,|\, D_p) \wedge \mathbb{1}[Q_{K'} = y_n] \\ &\qquad = \frac{1}{N}\sum_{n=1}^{N} \left(\bigwedge_{k=1}^{K'-1} \neg \text{cap}(x_n, P_k)\right) \wedge \text{cap}(x_n, P_{K'}) \wedge \mathbb{1}[Q_{K'} = y_n] \\ &\qquad \leq \frac{1}{N}\sum_{n=1}^{N} \left(\bigwedge_{k=1}^{K-1} \neg \text{cap}(x_n, p_k)\right) \wedge \text{cap}(x_n, P_{K'}) \wedge \mathbb{1}[Q_{K'} = y_n] \\ &\qquad = \frac{1}{N}\sum_{n=1}^{N} \left(\bigwedge_{k=1}^{K-1} \neg \text{cap}(x_n, p_k)\right) \wedge \text{cap}(x_n, p_K) \wedge \mathbb{1}[Q_{K'} = y_n] \\ &\qquad = \frac{1}{N}\sum_{n=1}^{N} \text{cap}(x_n, p_K \,|\, d_p) \wedge \mathbb{1}[Q_{K'} = y_n] \\ &\qquad \leq \frac{1}{N}\sum_{n=1}^{N} \text{cap}(x_n, p_K \,|\, d_p) \wedge \mathbb{1}[q_K = y_n] < \lambda.\end{aligned}$$

The first inequality reflects the condition that $D_p^{K'-1} \in \phi(d_p^{K-1})$, the next equality reflects the fact that $P_{K'} = p_K$. For the following equality, notice that $Q_{K'}$ is the majority class label of data captured by $P_{K'}$ in D_p, and q_K is the majority class label of data captured by P_K in d_p, and recall from (20) that $\text{supp}(P_{K'}, \mathbf{x} \,|\, D_p) \leq \text{supp}(p_K, \mathbf{x} \,|\, d_p)$. By Theorem 11, $D \notin \text{argmin}_{d^\dagger} R(d^\dagger, \mathbf{x}, \mathbf{y})$. ■

Propositions 13 and 14, combined with Proposition 22 (Appendix A), constitute the proof of Theorem 12.

3.10 Equivalent Support Bound

If two prefixes capture the same data, and one is more accurate than the other, then there is no benefit to considering prefixes that start with the less accurate one. Let d_p be a prefix, and consider the best possible rule list whose prefix starts with d_p. If we take its antecedents in d_p and replace them with another prefix with the same support (that could include different antecedents), then its objective can only become worse or remain the same.

Formally, let D_p be a prefix, and let $\xi(D_p)$ be the set of all prefixes that capture exactly the same data as D_p. Now, let d be a rule list with prefix d_p in $\xi(D_p)$, such that d has the minimum objective over all rule lists with prefixes in $\xi(D_p)$. Finally, let d' be a rule list whose prefix d'_p starts with d_p, such that d' has the minimum objective over all rule lists whose prefixes start with d_p. Theorem 15 below implies that d' also has the minimum objective over all rule lists whose prefixes start with *any* prefix in $\xi(D_p)$.

Theorem 15 (Equivalent support bound) *Define $\sigma(d_p)$ to be the set of all rule lists whose prefixes start with d_p, as in* (1)*. Let $d = (d_p, \delta_p, q_0, K)$ be a rule list with prefix $d_p = (p_1, \ldots, p_K)$, and let $D = (D_p, \Delta_p, Q_0, \kappa)$ be a rule list with prefix $D_p = (P_1, \ldots, P_\kappa)$,*

such that d_p and D_p capture the same data, i.e.,

$$\{x_n : \text{cap}(x_n, d_p)\} = \{x_n : \text{cap}(x_n, D_p)\}.$$

If the objective lower bounds of d and D obey $b(d_p, \mathbf{x}, \mathbf{y}) \leq b(D_p, \mathbf{x}, \mathbf{y})$, then the objective of the optimal rule list in $\sigma(d_p)$ gives a lower bound on the objective of the optimal rule list in $\sigma(D_p)$:

$$\min_{d' \in \sigma(d_p)} R(d', \mathbf{x}, \mathbf{y}) \leq \min_{D' \in \sigma(D_p)} R(D', \mathbf{x}, \mathbf{y}). \tag{21}$$

Proof See Appendix B for the proof of Theorem 15. ■

Thus, if prefixes d_p and D_p capture the same data, and their objective lower bounds obey $b(d_p, \mathbf{x}, \mathbf{y}) \leq b(D_p, \mathbf{x}, \mathbf{y})$, Theorem 15 implies that we can prune D_p. Next, in Sections 3.11 and 3.12, we highlight and analyze the special case of prefixes that capture the same data because they contain the same antecedents.

3.11 Permutation Bound

If two prefixes are composed of the same antecedents, *i.e.*, they contain the same antecedents up to a permutation, then they capture the same data, and thus Theorem 15 applies. Therefore, if one is more accurate than the other, then there is no benefit to considering prefixes that start with the less accurate one. Let d_p be a prefix, and consider the best possible rule list whose prefix starts with d_p. If we permute its antecedents in d_p, then its objective can only become worse or remain the same.

Formally, let $P = \{p_k\}_{k=1}^K$ be a set of K antecedents, and let Π be the set of all K-prefixes corresponding to permutations of antecedents in P. Let prefix d_p in Π have the minimum prefix misclassification error over all prefixes in Π. Also, let d' be a rule list whose prefix d'_p starts with d_p, such that d' has the minimum objective over all rule lists whose prefixes start with d_p. Corollary 16 below, which can be viewed as special case of Theorem 15, implies that d' also has the minimum objective over all rule lists whose prefixes start with *any* prefix in Π.

Corollary 16 (Permutation bound) *Let π be any permutation of $\{1, \ldots, K\}$, and define $\sigma(d_p) = \{(d'_p, \delta'_p, q'_0, K') : d'_p \text{ starts with } d_p\}$ to be the set of all rule lists whose prefixes start with d_p. Let $d = (d_p, \delta_p, q_0, K)$ and $D = (D_p, \Delta_p, Q_0, K)$ denote rule lists with prefixes $d_p = (p_1, \ldots, p_K)$ and $D_p = (p_{\pi(1)}, \ldots, p_{\pi(K)})$, respectively, i.e., the antecedents in D_p correspond to a permutation of the antecedents in d_p. If the objective lower bounds of d and D obey $b(d_p, \mathbf{x}, \mathbf{y}) \leq b(D_p, \mathbf{x}, \mathbf{y})$, then the objective of the optimal rule list in $\sigma(d_p)$ gives a lower bound on the objective of the optimal rule list in $\sigma(D_p)$:*

$$\min_{d' \in \sigma(d_p)} R(d', \mathbf{x}, \mathbf{y}) \leq \min_{D' \in \sigma(D_p)} R(D', \mathbf{x}, \mathbf{y}).$$

Proof Since prefixes d_p and D_p contain the same antecedents, they both capture the same data. Thus, we can apply Theorem 15. ■

Thus if prefixes d_p and D_p have the same antecedents, up to a permutation, and their objective lower bounds obey $b(d_p, \mathbf{x}, \mathbf{y}) \leq b(D_p, \mathbf{x}, \mathbf{y})$, Corollary 16 implies that we can prune D_p. We call this symmetry-aware pruning, and we illustrate the subsequent computational savings next in §3.12.

3.12 Upper Bound on Prefix Evaluations with Symmetry-aware Pruning

Here, we present an upper bound on the total number of prefix evaluations that accounts for the effect of symmetry-aware pruning (§3.11). Since every subset of K antecedents generates an equivalence class of $K!$ prefixes equivalent up to permutation, symmetry-aware pruning dramatically reduces the search space.

First, notice that Algorithm 1 describes a breadth-first exploration of the state space of rule lists. Now suppose we integrate symmetry-aware pruning into our execution of branch-and-bound, so that after evaluating prefixes of length K, we only keep a single best prefix from each set of prefixes equivalent up to a permutation.

Theorem 17 (Upper bound on prefix evaluations with symmetry-aware pruning) *Consider a state space of all rule lists formed from a set S of M antecedents, and consider the branch-and-bound algorithm with symmetry-aware pruning. Define $\Gamma_{\text{tot}}(S)$ to be the total number of prefixes evaluated. For any set S of M rules,*

$$\Gamma_{\text{tot}}(S) \leq 1 + \sum_{k=1}^{K} \frac{1}{(k-1)!} \cdot \frac{M!}{(M-k)!},$$

where $K = \min(\lfloor 1/2\lambda \rfloor, M)$.

Proof By Corollary 5, $K \equiv \min(\lfloor 1/2\lambda \rfloor, M)$ gives an upper bound on the length of any optimal rule list. The algorithm begins by evaluating the empty prefix, followed by M prefixes of length $k = 1$, then $P(M, 2)$ prefixes of length $k = 2$, where $P(M, 2)$ is the number of size-2 subsets of $\{1, \ldots, M\}$. Before proceeding to length $k = 3$, we keep only $C(M, 2)$ prefixes of length $k = 2$, where $C(M, k)$ denotes the number of k-combinations of M. Now, the number of length $k = 3$ prefixes we evaluate is $C(M, 2)(M - 2)$. Propagating this forward gives

$$\Gamma_{\text{tot}}(S) \leq 1 + \sum_{k=1}^{K} C(M, k-1)(M-k+1) = 1 + \sum_{k=1}^{K} \frac{1}{(k-1)!} \cdot \frac{M!}{(M-k)!}.$$

■

Pruning based on permutation symmetries thus yields significant computational savings. Let us compare, for example, to the naïve number of prefix evaluations given by the upper bound in Proposition 8. If $M = 100$ and $K = 5$, then the naïve number is about 9.1×10^9, while the reduced number due to symmetry-aware pruning is about 3.9×10^8, which is smaller by a factor of about 23. If $M = 1000$ and $K = 10$, the number of evaluations falls from about 9.6×10^{29} to about 2.7×10^{24}, which is smaller by a factor of about 360,000.

While 10^{24} seems infeasibly enormous, it does not represent the number of rule lists we evaluate. As we show in our experiments (§6), our permutation bound in Corollary 16 and

our other bounds together conspire to reduce the search space to a size manageable on a single computer. The choice of $M = 1000$ and $K = 10$ in our example above corresponds to the state space size our efforts target. $K = 10$ rules represents a (heuristic) upper limit on the size of an interpretable rule list, and $M = 1000$ represents the approximate number of rules with sufficiently high support (Theorem 10) we expect to obtain via rule mining (§3.1).

3.13 Similar Support Bound

We now present a relaxation of Theorem 15, our equivalent support bound. Theorem 18 implies that if we know that no extensions of a prefix d_p are better than the current best objective, then we can prune all prefixes with support similar to d_p's support. Understanding how to exploit this result in practice represents an exciting direction for future work; our implementation (§5) does not currently leverage the bound in Theorem 18.

Theorem 18 (Similar support bound) *Define $\sigma(d_p)$ to be the set of all rule lists whose prefixes start with d_p, as in (1). Let $d_p = (p_1, \ldots, p_K)$ and $D_p = (P_1, \ldots, P_\kappa)$ be prefixes that capture nearly the same data. Specifically, define ω to be the normalized support of data captured by d_p and not captured by D_p, i.e.,*

$$\omega \equiv \frac{1}{N}\sum_{n=1}^{N} \neg \operatorname{cap}(x_n, D_p) \wedge \operatorname{cap}(x_n, d_p). \tag{22}$$

Similarly, define Ω to be the normalized support of data captured by D_p and not captured by d_p, i.e.,

$$\Omega \equiv \frac{1}{N}\sum_{n=1}^{N} \neg \operatorname{cap}(x_n, d_p) \wedge \operatorname{cap}(x_n, D_p). \tag{23}$$

We can bound the difference between the objectives of the optimal rule lists in $\sigma(d_p)$ and $\sigma(D_p)$ as follows:

$$\min_{D^\dagger \in \sigma(D_p)} R(D^\dagger, \mathbf{x}, \mathbf{y}) - \min_{d^\dagger \in \sigma(d_p)} R(d^\dagger, \mathbf{x}, \mathbf{y}) \geq b(D_p, \mathbf{x}, \mathbf{y}) - b(d_p, \mathbf{x}, \mathbf{y}) - \omega - \Omega, \tag{24}$$

where $b(d_p, \mathbf{x}, \mathbf{y})$ and $b(D_p, \mathbf{x}, \mathbf{y})$ are the objective lower bounds of d and D, respectively.

Proof See Appendix C for the proof of Theorem 18. ■

Theorem 18 implies that if prefixes d_p and D_p are similar, and we know the optimal objective of rule lists starting with d_p, then

$$\begin{aligned}\min_{D' \in \sigma(D_p)} R(D', \mathbf{x}, \mathbf{y}) &\geq \min_{d' \in \sigma(d_p)} R(d', \mathbf{x}, \mathbf{y}) + b(D_p, \mathbf{x}, \mathbf{y}) - b(d_p, \mathbf{x}, \mathbf{y}) - \chi \\ &\geq R^c + b(D_p, \mathbf{x}, \mathbf{y}) - b(d_p, \mathbf{x}, \mathbf{y}) - \chi,\end{aligned}$$

where R^c is the current best objective, and χ is the normalized support of the set of data captured either exclusively by d_p or exclusively by D_p. It follows that

$$\min_{D' \in \sigma(D_p)} R(D', \mathbf{x}, \mathbf{y}) \geq R^c + b(D_p, \mathbf{x}, \mathbf{y}) - b(d_p, \mathbf{x}, \mathbf{y}) - \chi \geq R^c$$

if $b(D_p, \mathbf{x}, \mathbf{y}) - b(d_p, \mathbf{x}, \mathbf{y}) \geq \chi$. To conclude, we summarize this result and combine it with our notion of lookahead from Lemma 2. During branch-and-bound execution, if we demonstrate that $\min_{d' \in \sigma(d_p)} R(d', \mathbf{x}, \mathbf{y}) \geq R^c$, then we can prune all prefixes that start with any prefix D'_p in the following set:

$$\left\{ D'_p : b(D'_p, \mathbf{x}, \mathbf{y}) + \lambda - b(d_p, \mathbf{x}, \mathbf{y}) \geq \frac{1}{N}\sum_{n=1}^{N} \operatorname{cap}(x_n, d_p) \oplus \operatorname{cap}(x_n, D'_p) \right\},$$

where the symbol $\oplus$ denotes the logical operation, exclusive or (XOR).

3.14 Equivalent Points Bound

The bounds in this section quantify the following: If multiple observations that are not captured by a prefix d_p have identical features and opposite labels, then no rule list that starts with d_p can correctly classify all these observations. For each set of such observations, the number of mistakes is at least the number of observations with the minority label within the set.

Consider a data set $\{(x_n, y_n)\}_{n=1}^N$ and also a set of antecedents $\{s_m\}_{m=1}^M$. Define distinct observations to be equivalent if they are captured by exactly the same antecedents, *i.e.*, $x_i \neq x_j$ are equivalent if

$$\frac{1}{M}\sum_{m=1}^{M} \mathbb{1}[\operatorname{cap}(x_i, s_m) = \operatorname{cap}(x_j, s_m)] = 1.$$

Notice that we can partition a data set into sets of equivalent points; let $\{e_u\}_{u=1}^U$ enumerate these sets. Let e_u be the equivalent points set that contains observation x_i. Now define $\theta(e_u)$ to be the normalized support of the minority class label with respect to set e_u, *e.g.*, let

$$e_u = \{x_n : \forall m \in [M],\, \mathbb{1}[\operatorname{cap}(x_n, s_m) = \operatorname{cap}(x_i, s_m)]\},$$

and let q_u be the minority class label among points in e_u, then

$$\theta(e_u) = \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}[x_n \in e_u]\, \mathbb{1}[y_n = q_u]. \tag{25}$$

The existence of equivalent points sets with non-singleton support yields a tighter objective lower bound that we can combine with our other bounds; as our experiments demonstrate (§6), the practical consequences can be dramatic. First, for intuition, we present a general bound in Proposition 19; next, we explicitly integrate this bound into our framework in Theorem 20.

Proposition 19 (General equivalent points bound) *Let $d = (d_p, \delta_p, q_0, K)$ be a rule list, then*

$$R(d, \mathbf{x}, \mathbf{y}) \geq \sum_{u=1}^{U} \theta(e_u) + \lambda K.$$

Proof Recall that the objective is $R(d, \mathbf{x}, \mathbf{y}) = \ell(d, \mathbf{x}, \mathbf{y}) + \lambda K$, where the misclassification error $\ell(d, \mathbf{x}, \mathbf{y})$ is given by

$$\begin{aligned}\ell(d, \mathbf{x}, \mathbf{y}) &= \ell_0(d_p, q_0, \mathbf{x}, \mathbf{y}) + \ell_p(d_p, \delta_p, \mathbf{x}, \mathbf{y}) \\ &= \frac{1}{N} \sum_{n=1}^{N} \left(\neg \operatorname{cap}(x_n, d_p) \wedge \mathbb{1}[q_0 \neq y_n] + \sum_{k=1}^{K} \operatorname{cap}(x_n, p_k \,|\, d_p) \wedge \mathbb{1}[q_k \neq y_n] \right).\end{aligned}$$

Any particular rule list uses a specific rule, and therefore a single class label, to classify all points within a set of equivalent points. Thus, for a set of equivalent points u, the rule list d correctly classifies either points that have the majority class label, or points that have the minority class label. It follows that d misclassifies a number of points in u at least as great as the number of points with the minority class label. To translate this into a lower bound on $\ell(d, \mathbf{x}, \mathbf{y})$, we first sum over all sets of equivalent points, and then for each such set, count differences between class labels and the minority class label of the set, instead of counting mistakes:

$$\begin{aligned}&\ell(d, \mathbf{x}, \mathbf{y}) \\ &= \frac{1}{N} \sum_{u=1}^{U} \sum_{n=1}^{N} \left(\neg \operatorname{cap}(x_n, d_p) \wedge \mathbb{1}[q_0 \neq y_n] + \sum_{k=1}^{K} \operatorname{cap}(x_n, p_k \,|\, d_p) \wedge \mathbb{1}[q_k \neq y_n] \right) \mathbb{1}[x_n \in e_u] \\ &\geq \frac{1}{N} \sum_{u=1}^{U} \sum_{n=1}^{N} \left(\neg \operatorname{cap}(x_n, d_p) \wedge \mathbb{1}[y_n = q_u] + \sum_{k=1}^{K} \operatorname{cap}(x_n, p_k \,|\, d_p) \wedge \mathbb{1}[y_n = q_u] \right) \mathbb{1}[x_n \in e_u].\end{aligned} \tag{26}$$

Next, we factor out the indicator for equivalent point set membership, which yields a term that sums to one, because every datum is either captured or not captured by prefix d_p.

$$\begin{aligned}\ell(d, \mathbf{x}, \mathbf{y}) &= \frac{1}{N} \sum_{u=1}^{U} \sum_{n=1}^{N} \left(\neg \operatorname{cap}(x_n, d_p) + \sum_{k=1}^{K} \operatorname{cap}(x_n, p_k \,|\, d_p) \right) \wedge \mathbb{1}[x_n \in e_u] \, \mathbb{1}[y_n = q_u] \\ &= \frac{1}{N} \sum_{u=1}^{U} \sum_{n=1}^{N} (\neg \operatorname{cap}(x_n, d_p) + \operatorname{cap}(x_n, d_p)) \wedge \mathbb{1}[x_n \in e_u] \, \mathbb{1}[y_n = q_u] \\ &= \frac{1}{N} \sum_{u=1}^{U} \sum_{n=1}^{N} \mathbb{1}[x_n \in e_u] \, \mathbb{1}[y_n = q_u] = \sum_{u=1}^{U} \theta(e_u),\end{aligned}$$

where the final equality applies the definition of $\theta(e_u)$ in (25). Therefore, $R(d, \mathbf{x}, \mathbf{y}) = \ell(d, \mathbf{x}, \mathbf{y}) + \lambda K \geq \sum_{u=1}^{U} \theta(e_u) + \lambda K$. ■

Now, recall that to obtain our lower bound $b(d_p, \mathbf{x}, \mathbf{y})$ in (5), we simply deleted the default rule misclassification error $\ell_0(d_p, q_0, \mathbf{x}, \mathbf{y})$ from the objective $R(d, \mathbf{x}, \mathbf{y})$. Theorem 20 obtains a tighter objective lower bound via a tighter lower bound on the default rule misclassification error, $0 \leq b_0(d_p, \mathbf{x}, \mathbf{y}) \leq \ell_0(d_p, q_0, \mathbf{x}, \mathbf{y})$.

Theorem 20 (Equivalent points bound) *Let d be a rule list with prefix d_p and lower bound $b(d_p, \mathbf{x}, \mathbf{y})$, then for any rule list $d' \in \sigma(d)$ whose prefix d'_p starts with d_p,*

$$R(d', \mathbf{x}, \mathbf{y}) \geq b(d_p, \mathbf{x}, \mathbf{y}) + b_0(d_p, \mathbf{x}, \mathbf{y}), \tag{27}$$

where

$$b_0(d_p, \mathbf{x}, \mathbf{y}) = \frac{1}{N} \sum_{u=1}^{U} \sum_{n=1}^{N} \neg \operatorname{cap}(x_n, d_p) \wedge \mathbb{1}[x_n \in e_u] \, \mathbb{1}[y_n = q_u]. \tag{28}$$

Proof See Appendix D for the proof of Theorem 20. ■

4. Incremental Computation

For every prefix d_p evaluated during Algorithm 1's execution, we compute the objective lower bound $b(d_p, \mathbf{x}, \mathbf{y})$ and sometimes the objective $R(d, \mathbf{x}, \mathbf{y})$ of the corresponding rule list d. These calculations are the dominant computations with respect to execution time. This motivates our use of a highly optimized library, designed by Yang et al. (2017), for representing rule lists and performing operations encountered in evaluating functions of rule lists. Furthermore, we exploit the hierarchical nature of the objective function and its lower bound to compute these quantities incrementally throughout branch-and-bound execution. In this section, we provide explicit expressions for the incremental computations that are central to our approach. Later, in §5, we describe a cache data structure for supporting our incremental framework in practice.

For completeness, before presenting our incremental expressions, let us begin by writing down the objective lower bound and objective of the empty rule list, $d = ((), (), q_0, 0)$, the first rule list evaluated in Algorithm 1. Since its prefix contains zero rules, it has zero prefix misclassification error and also has length zero. Thus, the empty rule list's objective lower bound is zero, *i.e.*, $b((), \mathbf{x}, \mathbf{y}) = 0$. Since none of the data are captured by the empty prefix, the default rule corresponds to the majority class, and the objective corresponds to the default rule misclassification error, *i.e.*, $R(d, \mathbf{x}, \mathbf{y}) = \ell_0((), q_0, \mathbf{x}, \mathbf{y})$.

Now, we derive our incremental expressions for the objective function and its lower bound. Let $d = (d_p, \delta_p, q_0, K)$ and $d' = (d'_p, \delta'_p, q'_0, K+1)$ be rule lists such that prefix $d_p = (p_1, \ldots, p_K)$ is the parent of $d'_p = (p_1, \ldots, p_K, p_{K+1})$. Let $\delta_p = (q_1, \ldots, q_K)$ and $\delta'_p = (q_1, \ldots, q_K, q_{K+1})$ be the corresponding labels. The hierarchical structure of Algorithm 1 enforces that if we ever evaluate d', then we will have already evaluated both the objective and objective lower bound of its parent, d. We would like to reuse as much of these computations as possible in our evaluation of d'. We can write the objective lower bound of d' incrementally,

with respect to the objective lower bound of d:

$$\begin{aligned} b(d'_p, \mathbf{x}, \mathbf{y}) &= \ell_p(d'_p, \delta'_p, \mathbf{x}, \mathbf{y}) + \lambda(K+1) \\ &= \frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K+1} \mathrm{cap}(x_n, p_k \,|\, d'_p) \wedge \mathbb{1}[q_k \neq y_n] + \lambda(K+1) \qquad (29) \\ &= \ell_p(d_p, \delta_p, \mathbf{x}, \mathbf{y}) + \lambda K + \lambda + \frac{1}{N}\sum_{n=1}^{N} \mathrm{cap}(x_n, p_{K+1} \,|\, d'_p) \wedge \mathbb{1}[q_{K+1} \neq y_n] \\ &= b(d_p, \mathbf{x}, \mathbf{y}) + \lambda + \frac{1}{N}\sum_{n=1}^{N} \mathrm{cap}(x_n, p_{K+1} \,|\, d'_p) \wedge \mathbb{1}[q_{K+1} \neq y_n] \\ &= b(d_p, \mathbf{x}, \mathbf{y}) + \lambda + \frac{1}{N}\sum_{n=1}^{N} \neg\,\mathrm{cap}(x_n, d_p) \wedge \mathrm{cap}(x_n, p_{K+1}) \wedge \mathbb{1}[q_{K+1} \neq y_n]. \qquad (30) \end{aligned}$$

Thus, if we store $b(d_p, \mathbf{x}, \mathbf{y})$, then we can reuse this quantity when computing $b(d'_p, \mathbf{x}, \mathbf{y})$. Transforming (29) into (30) yields a significantly simpler expression that is a function of the stored quantity $b(d_p, \mathbf{x}, \mathbf{y})$. For the objective of d', first let us write a naïve expression:

$$\begin{aligned} &R(d', \mathbf{x}, \mathbf{y}) = \ell(d', \mathbf{x}, \mathbf{y}) + \lambda(K+1) = \ell_p(d'_p, \delta'_p, \mathbf{x}, \mathbf{y}) + \ell_0(d'_p, q'_0, \mathbf{x}, \mathbf{y}) + \lambda(K+1) \\ &= \frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K+1} \mathrm{cap}(x_n, p_k \,|\, d'_p) \wedge \mathbb{1}[q_k \neq y_n] + \frac{1}{N}\sum_{n=1}^{N} \neg\,\mathrm{cap}(x_n, d'_p) \wedge \mathbb{1}[q'_0 \neq y_n] + \lambda(K+1). \end{aligned} \qquad (31)$$

Instead, we can compute the objective of d' incrementally with respect to its objective lower bound:

$$\begin{aligned} R(d', \mathbf{x}, \mathbf{y}) &= \ell_p(d'_p, \delta'_p, \mathbf{x}, \mathbf{y}) + \ell_0(d'_p, q'_0, \mathbf{x}, \mathbf{y}) + \lambda(K+1) \\ &= b(d'_p, \mathbf{x}, \mathbf{y}) + \ell_0(d'_p, q'_0, \mathbf{x}, \mathbf{y}) \\ &= b(d'_p, \mathbf{x}, \mathbf{y}) + \frac{1}{N}\sum_{n=1}^{N} \neg\,\mathrm{cap}(x_n, d'_p) \wedge \mathbb{1}[q'_0 \neq y_n] \\ &= b(d'_p, \mathbf{x}, \mathbf{y}) + \frac{1}{N}\sum_{n=1}^{N} \neg\,\mathrm{cap}(x_n, d_p) \wedge (\neg\,\mathrm{cap}(x_n, p_{K+1})) \wedge \mathbb{1}[q'_0 \neq y_n]. \qquad (32) \end{aligned}$$

The expression in (32) is simpler to compute than that in (31), because the former reuses $b(d'_p, \mathbf{x}, \mathbf{y})$, which we already computed in (30). Note that instead of computing $R(d', \mathbf{x}, \mathbf{y})$ incrementally from $b(d'_p, \mathbf{x}, \mathbf{y})$ as in (32), we could have computed it incrementally from $R(d, \mathbf{x}, \mathbf{y})$. However, doing so would in practice require that we store $R(d, \mathbf{x}, \mathbf{y})$ in addition to $b(d_p, \mathbf{x}, \mathbf{y})$, which we already must store to support (30). We prefer the incremental approach suggested by (32) since it avoids this additional storage overhead.

We present an incremental branch-and-bound procedure in Algorithm 2, and show the incremental computations of the objective lower bound (30) and objective (32) as two separate functions in Algorithms 3 and 4, respectively. In Algorithm 2, we use a cache to

Algorithm 2 Incremental branch-and-bound for learning rule lists, for simplicity, from a cold start. We explicitly show the incremental objective lower bound and objective functions in Algorithms 3 and 4, respectively.

Input: Objective function $R(d, \mathbf{x}, \mathbf{y})$, objective lower bound $b(d_p, \mathbf{x}, \mathbf{y})$, set of antecedents $S = \{s_m\}_{m=1}^{M}$, training data $(\mathbf{x}, \mathbf{y}) = \{(x_n, y_n)\}_{n=1}^{N}$, regularization parameter λ
Output: Provably optimal rule list d^* with minimum objective R^*

```
d^c ← ((), (), q_0, 0)                          ▷ Initialize current best rule list with empty rule list
R^c ← R(d^c, x, y)                              ▷ Initialize current best objective
Q ← queue( [ ( ) ] )                            ▷ Initialize queue with empty prefix
C ← cache( [ ( ( ) , 0 ) ] )                    ▷ Initialize cache with empty prefix and its objective lower bound
while Q not empty do                            ▷ Optimization complete when the queue is empty
    d_p ← Q.pop( )                              ▷ Remove a length-K prefix d_p from the queue
    b(d_p, x, y) ← C.find(d_p)                  ▷ Look up d_p's lower bound in the cache
    u ← ¬cap(x, d_p)                            ▷ Bit vector indicating data not captured by d_p
    for s in S do                               ▷ Evaluate all of d_p's children
        if s not in d_p then
            D_p ← (d_p, s)                      ▷ Branch: Generate child D_p
            v ← u ∧ cap(x, s)                   ▷ Bit vector indicating data captured by s in D_p
            b(D_p, x, y) ← b(d_p, x, y) + λ + IncrementalLowerBound(v, y, N)
            if b(D_p, x, y) < R^c then          ▷ Bound: Apply bound from Theorem 1
                R(D, x, y) ← b(D_p, x, y) + IncrementalObjective(u, v, y, N)
                D ← (D_p, Δ_p, Q_0, K + 1)      ▷ Δ_p, Q_0 are set in the incremental functions
                if R(D, x, y) < R^c then
                    (d^c, R^c) ← (D, R(D, x, y)) ▷ Update current best rule list and objective
                end if
                Q.push(D_p)                     ▷ Add D_p to the queue
                C.insert(D_p, b(D_p, x, y))     ▷ Add D_p and its lower bound to the cache
            end if
        end if
    end for
end while
(d*, R*) ← (d^c, R^c)                           ▷ Identify provably optimal rule list and objective
```

store prefixes and their objective lower bounds. Algorithm 2 additionally reorganizes the structure of Algorithm 1 to group together the computations associated with all children of a particular prefix. This has two advantages. The first is to consolidate cache queries: all children of the same parent prefix compute their objective lower bounds with respect to the parent's stored value, and we only require one cache 'find' operation for the entire group of children, instead of a separate query for each child. The second is to shrink the queue's size: instead of adding all of a prefix's children as separate queue elements, we represent the entire group of children in the queue by a single element. Since the number of children associated with each prefix is close to the total number of possible antecedents, both of these effects can yield significant savings. For example, if we are trying to optimize over rule lists

Algorithm 3 Incremental objective lower bound (30) used in Algorithm 2.

Input: Bit vector $\mathbf{v} \in \{0,1\}^N$ indicating data captured by s, the last antecedent in D_p, bit vector of class labels $\mathbf{y} \in \{0,1\}^N$, number of observations N
Output: Component of D's misclassification error due to data captured by s

function IncrementalLowerBound($\mathbf{v}, \mathbf{y}, N$)
  $n_v = \text{sum}(\mathbf{v})$ ▷ Number of data captured by s, the last antecedent in D_p
  $\mathbf{w} \leftarrow \mathbf{v} \wedge \mathbf{y}$ ▷ Bit vector indicating data captured by s with label 1
  $n_w = \text{sum}(\mathbf{w})$ ▷ Number of data captured by s with label 1
  if $n_w/n_v > 0.5$ **then**
    return $(n_v - n_w)/N$ ▷ Misclassification error of the rule $s \to 1$
  else
    return n_w/N ▷ Misclassification error of the rule $s \to 0$
  end if
end function

Algorithm 4 Incremental objective function (32) used in Algorithm 2.

Input: Bit vector $\mathbf{u} \in \{0,1\}^N$ indicating data not captured by D_p's parent prefix, bit vector $\mathbf{v} \in \{0,1\}^N$ indicating data not captured by s, the last antecedent in D_p, bit vector of class labels $\mathbf{y} \in \{0,1\}^N$, number of observations N
Output: Component of D's misclassification error due to its default rule

function IncrementalObjective($\mathbf{u}, \mathbf{v}, \mathbf{y}, N$)
  $\mathbf{f} \leftarrow \mathbf{u} \wedge \neg\, \mathbf{v}$ ▷ Bit vector indicating data not captured by D_p
  $n_f = \text{sum}(\mathbf{f})$ ▷ Number of data not captured by D_p
  $\mathbf{g} \leftarrow \mathbf{f} \wedge \mathbf{y}$ ▷ Bit vector indicating data not captured by D_p with label 1
  $n_g = \text{sum}(\mathbf{g})$ ▷ Number of data not captued by D_p with label 1
  if $n_g/n_f > 0.5$ **then**
    return $(n_f - n_g)/N$ ▷ Misclassification error of the default label prediction 1
  else
    return n_g/N ▷ Misclassification error of the default label prediction 0
  end if
end function

formed from a set of 1000 antecedents, then the maximum queue size in Algorithm 2 will be smaller than that in Algorithm 1 by a factor of nearly 1000.

5. Implementation

We implement our algorithm using a collection of optimized data structures that we describe in this section. First, we explain how we use a prefix tree (§5.1) to support the incremental computations that we motivated in §4. Second, we describe several queue designs that implement different search policies (§5.2). Third, we introduce a symmetry-aware map (§5.3) to support symmetry-aware pruning (Corollary 16, §3.11). Next, we summarize how these data structures interact throughout our model of incremental execution (§5.4). In particular, Algorithms 5 and 6 illustrate many of the computational details from CORELS' inner loop, highlighting each of the bounds from §3 that we use to prune the search space. We additionally describe how we garbage collect our data structures (§5.5). Finally, we explore how our queue can be used to support custom scheduling policies designed to improve performance (§5.6).

5.1 Prefix Tree

Our incremental computations (§4) require a cache to keep track of prefixes that we have already evaluated and that are also still under consideration by the algorithm. We implement this cache as a prefix tree, a data structure also known as a trie, which allows us to efficiently represent structure shared between related prefixes. Each node in the prefix tree encodes an individual rule $r_k = p_k \to q_k$. Each path starting from the root represents a prefix, such that the final node in the path also contains metadata associated with that prefix. For a prefix $d_p = (p_1, \ldots, p_K)$, let $\varphi(d_p)$ denote the corresponding node in the trie. The metadata at node $\varphi(d_p)$ supports the incremental computation and includes:

- An index encoding p_K, the last antecedent.
- The objective lower bound $b(d_p, \mathbf{x}, \mathbf{y})$, defined in (5), the central bound in our framework (Theorem 1).
- The lower bound on the default rule misclassification error $b_0(d_p, \mathbf{x}, \mathbf{y})$, defined in (28), to support our equivalent points bound (Theorem 20).
- An indicator denoting whether this node should be deleted (see §5.5).
- A representation of viable extensions of d_p, *i.e.*, length $K+1$ prefix that start with d_p and have not been pruned.

For evaluation purposes and convenience, we store additional information in the prefix tree; for a prefix d_p with corresponding rule list $d = (d_p, \delta_p, q_0, K)$, the node $\varphi(d_p)$ also stores:

- The length K; equivalently, node $\varphi(d_p)$'s depth in the trie.
- The label prediction q_K corresponding to antecedent p_K.
- The default rule label prediction q_0.
- N_{cap}, the number of samples captured by prefix d_p, as in (34).
- The objective value $R(d, \mathbf{x}, \mathbf{y})$, defined in (4).

Finally, we note that we implement the prefix tree as a custom C++ class.

5.2 Queue

The queue is a worklist that orders exploration over the search space of possible rule lists; every queue element corresponds to a leaf in the prefix tree, and vice versa. In our implementation, each queue element points to a leaf; when we pop an element off the queue, we use the leaf's metadata to incrementally evaluate the corresponding prefix's children.

We order entries in the queue to implement several different search policies. For example, a first-in-first-out (FIFO) queue implements breadth-first search (BFS), and a priority queue implements best-first search. In our experiments (§6), we use the C++ Standard Template Library (STL) queue and priority queue to implement BFS and best-first search, respectively. For CORELS, priority queue policies of interest include ordering by the lower bound, the objective, or more generally, any function that maps prefixes to real values; stably ordering by prefix length and inverse prefix length implement BFS and depth-first search (DFS), respectively. In our released code, we present a unified implementation, where we use the STL priority queue to support BFS, DFS, and several best-first search policies. As we demonstrate in our experiments (§6.8), we find that using a custom search strategy, such as ordering by the lower bound, usually leads to a faster runtime than BFS.

We motivate the design of additional custom search strategies in §5.6. In preliminary work (not shown), we also experimented with stochastic exploration processes that bypass the need for a queue by instead following random paths from the root to leaves; developing such methods could be an interesting direction for future work. We note that these search policies are referred to as node selection strategies in the MIP literature. Strategies such as best-first (best-bound) search and DFS are known as static methods, and the framework we present in §5.6 has the spirit of estimate-based methods (Linderoth and Savelsbergh, 1999).

5.3 Symmetry-aware Map

The symmetry-aware map supports the symmetry-aware pruning justified in §3.10. In our implementation, we specifically leverage our permutation bound (Corollary 16), though it is also possible to directly exploit the more general equivalent support bound (Theorem 15). We use the C++ STL unordered map to keep track of the best known ordering of each evaluated set of antecedents. The keys of our symmetry-aware map encode antecedents in canonical order, *i.e.*, antecedent indices in numerically sorted order, and we associate all permutations of a set of antecedents with a single key. Each key maps to a value that encodes the best known prefix in the permutation group of the key's antecedents, as well as the objective lower bound of that prefix.

Before we consider adding a prefix d_p to the trie and queue, we check whether the map already contains a permutation $\pi(d_p)$ of that prefix. If no such permutation exists, then we insert d_p into the map, trie, and queue. Otherwise, if a permutation $\pi(d_p)$ exists and the lower bound of d_p is better than that of $\pi(d_p)$, *i.e.*, $b(d_p, \mathbf{x}, \mathbf{y}) < b(\pi(d_p), \mathbf{x}, \mathbf{y})$, then we update the map and remove $\pi(d_p)$ and its entire subtree from the trie; we also insert d_p into the trie and queue. Otherwise, if there exists a permutation $\pi(d_p)$ such that $b(\pi(d_p), \mathbf{x}, \mathbf{y}) \leq b(d_p, \mathbf{x}, \mathbf{y})$, then we do nothing, *i.e.*, we do not insert d_p into any data structures.

5.4 Incremental Execution

Mapping our algorithm to our data structures produces the following execution strategy, which we also illustrate in Algorithms 5 and 6. We initialize the current best objective R^c and rule list d^c. While the trie contains unexplored leaves, a scheduling policy selects the next prefix d_p to extend; in our implementation, we pop elements from a (priority) queue, until the queue is empty. Then, for every antecedent s that is not in d_p, we construct a new prefix D_p by appending s to d_p; we incrementally calculate the lower bound $b(D_p, \mathbf{x}, \mathbf{y})$, the objective $R(D, \mathbf{x}, \mathbf{y})$, of the associated rule list D, and other quantities used by our algorithm, summarized by the metadata fields of the (potential) prefix tree node $\varphi(D_p)$.

If the objective $R(D, \mathbf{x}, \mathbf{y})$ is less than the current best objective R^c, then we update R^c and d^c. If the lower bound of the new prefix D_p is less than the current best objective, then as described in §5.3, we query the symmetry-aware map for D_p; if we insert d'_p into the symmetry-aware map, then we also insert it into the trie and queue. Otherwise, then by our hierarchical lower bound (Theorem 1), no extension of D_p could possibly lead to a rule list with objective better than R^c, thus we do not insert D_p into the tree or queue. We also leverage our other bounds from §3 to aggressively prune the search space; we highlight each of these bounds in Algorithms 5 and 6, which summarize the computations and data structure operations performed in CORELS' inner loop. When there are no more leaves to explore, *i.e.*, the queue is empty, we output the optimal rule list. We can optionally terminate early according to some alternate condition, *e.g.*, when the size of the prefix tree exceeds some threshold.

5.5 Garbage Collection

During execution, we garbage collect the trie. Each time we update the minimum objective, we traverse the trie in a depth-first manner, deleting all subtrees of any node with lower bound larger than the current minimum objective. At other times, when we encounter a node with no children, we prune upwards, deleting that node and recursively traversing the tree towards the root, deleting any childless nodes. This garbage collection allows us to constrain the trie's memory consumption, though in our experiments we observe the minimum objective to decrease only a small number of times.

In our implementation, we cannot immediately delete prefix tree leaves because each corresponds to a queue element that points to it. The C++ STL priority queue is a wrapper container that prevents access to the underlying data structure, and thus we cannot access elements in the middle of the queue, even if we know the relevant identifying information. We therefore have no way to update the queue without iterating over every element. We address this by marking prefix tree leaves that we wish to delete (see §5.1), deleting the physical nodes lazily, after they are popped from the queue. Later, in our section on experiments (§6), we refer to two different queues that we define here: the physical queue corresponds to the C++ queue, and thus all prefix tree leaves, and the logical queue corresponds only to those prefix tree leaves that have not been marked for deletion.

Algorithm 5 The inner loop of CORELS, which evaluates all children of a prefix d_p.

Define $\mathbf{z} \in \{0,1\}^N$, s.t. $z_n = \sum_{u=1}^{U} \mathbb{1}[x_n \in e_u][y_n = q_u]$
▷ e_u is the equivalent points set containing x_n and q_u is the minority class label of e_u (§3.14)
Define $b(d_p, \mathbf{x}, \mathbf{y})$ and $\mathbf{u} = \neg \operatorname{cap}(\mathbf{x}, d_p)$ ▷ $\mathbf{u}$ is a bit vector indicating data not captured by d_p
for s in S **if** s not in d_p **then do** ▷ Evaluate all of d_p's children
  $D_p \leftarrow (d_p, s)$ ▷ **Branch**: Generate child D_p
  $\mathbf{v} \leftarrow \mathbf{u} \wedge \operatorname{cap}(\mathbf{x}, s)$ ▷ Bit vector indicating data captured by s in D_p
  $n_v = \operatorname{sum}(\mathbf{v})$ ▷ Number of data captured by s, the last antecedent in D_p
  if $n_v/N < \lambda$ **then**
    continue ▷ **Lower bound on antecedent support (Theorem10)**
  end if
  $\mathbf{w} \leftarrow \mathbf{v} \wedge \mathbf{y}$ ▷ Bit vector indicating data captured by s with label 1
  $n_w = \operatorname{sum}(\mathbf{w})$ ▷ Number of data captured by s with label 1
  if $n_w/n_v \geq 0.5$ **then**
    $n_c \leftarrow n_w$ ▷ Number of correct predictions by the new rule $s \rightarrow 1$
  else
    $n_c \leftarrow n_v - n_w$ ▷ Number of correct predictions by the new rule $s \rightarrow 0$
  end if
  if $n_c/N < \lambda$ **then**
    continue ▷ **Lower bound on accurate antecedent support (Theorem 11)**
  end if
  $\delta_b \leftarrow (n_v - n_c)/N$ ▷ Misclassification error of the new rule
  $b(D_p, \mathbf{x}, \mathbf{y}) \leftarrow b(d_p, \mathbf{x}, \mathbf{y}) + \lambda + \delta_b$ ▷ Incremental lower bound (30)
  if $b(D_p, \mathbf{x}, \mathbf{y}) \geq R^c$ **then** ▷ **Hierarchical objective lower bound (Theorem 1)**
    continue
  end if
  $\mathbf{f} \leftarrow \mathbf{u} \wedge \neg \mathbf{v}$ ▷ Bit vector indicating data not captured by D_p
  $n_f = \operatorname{sum}(\mathbf{f})$ ▷ Number of data not captured by D_p
  $\mathbf{g} \leftarrow \mathbf{f} \wedge \mathbf{y}$ ▷ Bit vector indicating data not captured by D_p with label 1
  $n_g = \operatorname{sum}(\mathbf{g})$ ▷ Number of data not captued by D_p with label 1
  if $n_g/n_f \geq 0.5$ **then**
    $\delta_R \leftarrow (n_f - n_g)/N$ ▷ Misclassification error of the default label prediction 1
  else
    $\delta_R \leftarrow n_g/N$ ▷ Misclassification error of the default label prediction 0
  end if
  $R(D, \mathbf{x}, \mathbf{y}) \leftarrow b(D_p, \mathbf{x}, \mathbf{y}) + \delta_R$ ▷ Incremental objective (32)
  $D \leftarrow (D_p, \Delta_p, Q_0, K+1)$ ▷ Δ_p, Q_0 are set in the incremental functions
  if $R(D, \mathbf{x}, \mathbf{y}) < R^c$ **then**
    $(d^c, R^c) \leftarrow (D, R(D, \mathbf{x}, \mathbf{y}))$ ▷ Update current best rule list and objective
    GarbageCollectPrefixTree(R^c) ▷ Delete nodes with lower bound $\geq R^c - \lambda$ (§5.5),
  end if using the **Lookahead bound (Lemma 2)**
  $b_0(D_p, \mathbf{x}, \mathbf{y}) \leftarrow \operatorname{sum}(\mathbf{f} \wedge \mathbf{z})/N$ ▷ Lower bound on the default rule misclassification
  $b \leftarrow b(D_p, \mathbf{x}, \mathbf{y}) + b_0(D_p, \mathbf{x}, \mathbf{y})$ error defined in (28)
  if $b + \lambda \geq R^c$ **then** ▷ **Equivalent points bound (Theorem 20)**
    continue combined with the **Lookahead bound (Lemma 2)**
  end if
  CheckMapAndInsert(D_p, b) ▷ Check the **Permutation bound (Corollary 16)** and
end for possibly insert D_p into data structures (Algorithm 6)

Algorithm 6 Possibly insert a prefix into CORELS' data structures, after first checking the symmetry-aware map, which supports search space pruning triggered by the permutation bound (Corollary 16). For further context, see Algorithm 5.

T is the prefix tree (§5.1)
Q is the queue, for concreteness, a priority queue ordered by the lower bound (§5.2)
P is the symmetry-aware map (§5.3)

function CheckMapAndInsert(D_p, b)
  $\pi_0 \leftarrow \operatorname{sort}(D_p)$ ▷ D_p's antecedents in canonical order
  $(D_\pi, b_\pi) \leftarrow P.\operatorname{find}(\pi_0)$ ▷ Look for a permutation of D_p
  if D_π exists **then**
    if $b < b_\pi$ **then** ▷ D_p is better than D_π
      $P.\operatorname{update}(\pi_0, (D_p, b))$ ▷ Replace D_π with D_p in the map
      $T.\operatorname{delete\_subtree}(D_\pi)$ ▷ Delete D_π and its subtree from the prefix tree
      $T.\operatorname{insert}(\varphi(D_p))$ ▷ Add node for D_p to the prefix tree
      $Q.\operatorname{push}(D_p, b)$ ▷ Add D_p to the queue
    else
      pass ▷ D_p is inferior to D_π, thus do not insert it into any data structures
    end if
  else
    $P.\operatorname{insert}(\pi_0, (D_p, b))$ ▷ Add D_p to the map
    $T.\operatorname{insert}(\varphi(D_p))$ ▷ Add node for D_p to the prefix tree
    $Q.\operatorname{push}(D_p, b)$ ▷ Add D_p to the queue
  end if
end function

5.6 Custom Scheduling Policies

In our setting, an ideal scheduling policy would immediately identify an optimal rule list, and then certify its optimality by systematically eliminating the remaining search space. This motivates trying to design scheduling policies that tend to quickly find optimal rule lists. When we use a priority queue to order the set of prefixes to evaluate next, we are free to implement different scheduling policies via the ordering of elements in the queue. This motivates designing functions that assign higher priorities to 'better' prefixes that we believe are more likely to lead to optimal rule lists. We follow the convention that priority queue elements are ordered by keys, such that keys with smaller values have higher priorities.

We introduce a custom class of functions that we call *curiosity* functions. Broadly, we think of the curiosity of a rule list d as the expected objective value of another rule list d' that is related to d; different models of the relationship between d and d' lead to different curiosity functions. In general, the curiosity of d is, by definition, equal to the sum of the expected misclassification error and the expected regularization penalty of d':

$$\mathcal{C}(d_p, \mathbf{x}, \mathbf{y}) \equiv \mathbb{E}[R(d', \mathbf{x}, \mathbf{y})] = \mathbb{E}[\ell(d'_p, \delta'_p, \mathbf{x}, \mathbf{y})] + \lambda \mathbb{E}[K']. \tag{33}$$

Next, we describe a simple curiosity function for a rule list d with prefix d_p. First, let N_{cap} denote the number of observations captured by d_p, *i.e.*,

$$N_{\text{cap}} \equiv \sum_{n=1}^{N} \text{cap}(x_n, d_p). \tag{34}$$

We now describe a model that generates another rule list $d' = (d'_p, \delta'_p, q'_0, K')$ from d_p. Assume that prefix d'_p starts with d_p and captures all the data, such that each additional antecedent in d'_p captures as many 'new' observations as each antecedent in d_p, on average; then, the expected length of d'_p is

$$\mathbb{E}[K'] = \frac{N}{N_{\text{cap}}/K}. \tag{35}$$

Furthermore, assume that each additional antecedent in d'_p makes as many mistakes as each antecedent in d_p, on average, thus the expected misclassification error of d'_p is

$$\begin{aligned}\mathbb{E}[\ell(d'_p, \delta'_p, \mathbf{x}, \mathbf{y})] &= \mathbb{E}[\ell_p(d'_p, \delta'_p, \mathbf{x}, \mathbf{y})] + \mathbb{E}[\ell_0(d'_p, q'_0, \mathbf{x}, \mathbf{y})] \\ &= \mathbb{E}[\ell_p(d'_p, \delta'_p, \mathbf{x}, \mathbf{y})] = \mathbb{E}[K'] \left(\frac{\ell_p(d_p, \delta_p, \mathbf{x}, \mathbf{y})}{K} \right). \end{aligned} \tag{36}$$

Note that the default rule misclassification error $\ell_0(d'_p, q'_0, \mathbf{x}, \mathbf{y})$ is zero because we assume that d'_p captures all the data. Inserting (35) and (36) into (33) thus gives curiosity for this model:

$$\begin{aligned}\mathcal{C}(d_p, \mathbf{x}, \mathbf{y}) &= \left(\frac{N}{N_{\text{cap}}} \right) \left(\ell_p(d_p, \delta_p, \mathbf{x}, \mathbf{y}) + \lambda K \right) \\ &= \left(\frac{1}{N} \sum_{n=1}^{N} \text{cap}(x_n, d_p) \right)^{-1} b(d_p, \mathbf{x}, \mathbf{y}) = \frac{b(d_p, \mathbf{x}, \mathbf{y})}{\text{supp}(d_p, \mathbf{x})}, \end{aligned}$$

where for the second equality, we used the definitions in (34) of N_{cap} and in (5) of d_p's lower bound, and for the last equality, we used the definition in (2) of d_p's normalized support.

The curiosity for a prefix d_p is thus also equal to its objective lower bound, scaled by the inverse of its normalized support. For two prefixes with the same lower bound, curiosity gives higher priority to the one that captures more data. This is a well-motivated scheduling strategy if we model prefixes that extend the prefix with smaller support as having more 'potential' to make mistakes. We note that using curiosity in practice does not introduce new bit vector or other expensive computations; during execution, we can calculate curiosity as a simple function of already derived quantities.

In preliminary experiments, we observe that using a priority queue ordered by curiosity sometimes yields a dramatic reduction in execution time, compared to using a priority queue ordered by the objective lower bound. Thus far, we have observed significant benefits on specific small problems, where the structure of the solutions happen to render curiosity particularly effective (not shown). Designing and studying other 'curious' functions, that are effective in more general settings, is an exciting direction for future work.

Data set	Prediction problem	N	Positive fraction	Resample training set	Training set size	Test set size
ProPublica	Two-year recidivism	6,907	0.46	No	6,217	692
NYPD	Weapon possession	325,800	0.03	Yes	566,839	32,580
NYCLU	Weapon possession	29,595	0.05	Yes	50,743	2,959

Table 1: Summary of data sets and prediction problems. The last five columns report the total number of observations, the fraction of observations with the positive class label, whether we resampled the training set due to class imbalance, and the sizes of each training and test set in our 10-fold cross-validation studies.

6. Experiments

Our experimental analysis addresses five questions: How does CORELS' predictive performance compare to that of COMPAS scores and other algorithms? (§6.4, §6.5, and §6.6) How does CORELS' model size compare to that of other algorithms? (§6.6) How rapidly do the objective value and its lower bound converge, for different values of the regularization parameter λ? (§6.7) How much does each of the implementation optimizations contribute to CORELS' performance? (§6.8) How rapidly does CORELS prune the search space? (§6.7 and §6.8) Before proceeding, we first describe our computational environment (§6.1), as well as the data sets and prediction problems we use (§6.2), and then in Section 6.3 show example optimal rule lists found by CORELS.

6.1 Computational Environment

All timed results ran on a server with an Intel Xeon E5-2699 v4 (55 MB cache, 2.20 GHz) processor and 264 GB RAM, and we ran each timing measurement separately, on a single hardware thread, with nothing else running on the server. Except where we mention a memory constraint, all experiments can run comfortably on smaller machines, *e.g.*, a laptop with 16 GB RAM.

6.2 Data Sets and Prediction Problems

Our evaluation focuses on two socially-important prediction problems associated with recent, publicly-available data sets. Table 1 summarizes the data sets and prediction problems, and Table 2 summarizes feature sets extracted from each data set, as well as antecedent sets we mine from these feature sets. We provide some details next. For further details about data sets, preprocessing steps, and antecedent mining, see Appendix E.

6.2.1 Recidivism Prediction

For our first problem, we predict which individuals in the ProPublica COMPAS data set (Larson et al., 2016) recidivate within two years. This data set contains records for all offenders in Broward County, Florida in 2013 and 2014 who were given a COMPAS score pre-trial. Recidivism is defined as being charged with a new crime within two years

Data set	Feature set	Categorical attributes	Binary features	Mined antecedents	Max number of clauses	Negations
ProPublica	A	6	13	122	2	No
ProPublica	B	7	17	189	2	No
NYPD	C	5	28	28	1	No
NYPD	D	3	20	20	1	No
NYCLU	E	5	28	46	1	Yes

Table 2: Summary of feature sets and mined antecedents. The last five columns report the number of categorical attributes, the number of binary features, the average number of mined antecedents, the maximum number of clauses in each antecedent, and whether antecedents include negated clauses.

after receiving a COMPAS assessment; the article by Larson et al. (2016), and their code,[3] provide more details about this definition. From the original data set of records for 7,214 individuals, we identify a subset of 6,907 records without missing data. For the majority of our analysis, we extract a set of 13 binary features (Feature Set A), which our antecedent mining framework combines into $M = 122$ antecedents, on average (folds ranged from containing 121 to 123 antecedents). We also consider a second, similar antecedent set in §6.3, derived from a superset of Feature Set A that includes 4 additional binary features (Feature Set B).

6.2.2 Weapon Prediction

For our second problem, we use New York City stop-and-frisk data to predict whether a weapon will be found on a stopped individual who is frisked or searched. For experiments in Sections 6.3 and 6.5 and Appendix G, we compile data from a database maintained by the New York Police Department (NYPD) (New York Police Department, 2016), from years 2008-2012, following Goel et al. (2016). Starting from 2,941,390 records, each describing an incident involving a stopped person, we first extract 376,488 records where the suspected crime was criminal possession of a weapon (CPW).[4] From these, we next identify a subset of 325,800 records for which the individual was frisked and/or searched; of these, criminal possession of a weapon was identified in only 10,885 instances (about 3.3%). Resampling due to class imbalance, for 10-fold cross-validation, yields training sets that each contain 566,839 datapoints. (We form corresponding test sets without resampling.) From a set of 5 categorical features, we form a set of 28 single-clause antecedents corresponding to 28 binary features (Feature Set C). We also consider another, similar antecedent set, derived from a subset of Feature Set C that excludes 8 location-specific binary features (Feature Set D).

In Sections 6.3, 6.6, 6.7, and 6.8, we also use a smaller stop-and-frisk data set, derived by the NYCLU from the NYPD's 2014 data (New York Civil Liberties Union, 2014). From the

3. Data and code used in the analysis by Larson et al. (2016) can be found at `https://github.com/propublica/compas-analysis`.

4. We filter for records that explicitly match the string 'CPW'; we note that additional records, after converting to lowercase, contain strings such as 'cpw' or 'c.p.w.'

if $(age = 21 - 22)$ **and** $(priors = 2 - 3)$ **then predict** *yes*
else if $(age = 18 - 20)$ **and** $(sex = male)$ **then predict** *yes*
else if $(priors > 3)$ **then predict** *yes*
else predict *no*

if $(age = 23 - 25)$ **and** $(priors = 2 - 3)$ **then predict** *yes*
else if $(age = 18 - 20)$ **and** $(sex = male)$ **then predict** *yes*
else if $(age = 21 - 22)$ **and** $(priors = 2 - 3)$ **then predict** *yes*
else if $(priors > 3)$ **then predict** *yes*
else predict *no*

Figure 3: Example optimal rule lists that predict two-year recidivism for the ProPublica data set (Feature Set B, $M = 189$), found by CORELS ($\lambda = 0.005$), across 10 cross-validation folds. While some input antecedents contain features for race, no optimal rule list includes such an antecedent. Every optimal rule list is the same or similar to one of these examples, with prefixes containing the same rules, up to a permutation, and same default rule.

original data set of 45,787 records, each describing an incident involving a stopped person, we identify a subset of 29,595 records for which the individual was frisked and/or searched. Of these, criminal possession of a weapon was identified in about 5% of instances. As with the larger NYPD data set, we resample the data to form training sets (but not to form test sets). From the same set of 5 categorical features as in Feature Set C, we form a set of $M = 46$ single-clause antecedents, including negations (Feature Set E).

6.3 Example Optimal Rule Lists

To motivate Feature Set A, described in Appendix E, which we used in most of our analysis of the ProPublica data set, we first consider Feature Set B, a larger superset of features.

Figure 3 shows optimal rule lists learned by CORELS, using Feature Set B, which additionally includes race categories from the ProPublica data set (African American, Caucasian, Hispanic, Other[5]). For Feature Set B, our antecedent mining procedure generated an average of 189 antecedents, across folds. None of the optimal rule lists contain antecedents that directly depend on race; this motivated our choice to exclude race, by using Feature Set A, in our subsequent analysis. For both feature sets, we replaced the original ProPublica age categories (<25, 25-45, >45) with a set that is more fine-grained for younger individuals (18-20, 21-22, 23-25, 26-45, >45). Figure 4 shows example optimal rule lists that CORELS learns for the ProPublica data set (Feature Set A, $\lambda = 0.005$), using 10-fold cross validation.

Figures 5 and 6 show example optimal rule lists that CORELS learns for the NYCLU ($\lambda = 0.01$) and NYPD data sets. Figure 6 shows optimal rule lists that CORELS learns for the larger NYPD data set.

While our goal is to provide illustrative examples, and not to provide a detailed analysis nor to advocate for the use of these specific models, we note that these rule lists are short and easy to understand. For the examples and regularization parameter choices in this section,

5. We grouped the original Native American (<0.003), Asian (<0.005), and Other (<0.06) categories.

if $(age = 18-20)$ **and** $(sex = male)$ **then predict** *yes*
else if $(age = 21-22)$ **and** $(priors = 2-3)$ **then predict** *yes*
else if $(priors > 3)$ **then predict** *yes*
else predict *no*

if $(age = 18-20)$ **and** $(sex = male)$ **then predict** *yes*
else if $(age = 21-22)$ **and** $(priors = 2-3)$ **then predict** *yes*
else if $(age = 23-25)$ **and** $(priors = 2-3)$ **then predict** *yes*
else if $(priors > 3)$ **then predict** *yes*
else predict *no*

Figure 4: Example optimal rule lists that predict two-year recidivism for the ProPublica data set (Feature Set A, $M = 122$), found by CORELS ($\lambda = 0.005$), across 10 cross-validation folds. Feature Set A is a subset of Feature Set B (Figure 3) that excludes race features. Optimal rule lists found using the two feature sets are very similar. The upper and lower rule lists are representative of 7 and 3 folds, respectively. Each of the remaining 8 solutions is the same or similar to one of these, with prefixes containing the same rules, up to a permutation, and the same default rule. See Figure 20 in Appendix F for a complete listing.

if $(location = transit\ authority)$ **then predict** *yes*
else if $(stop\ reason = suspicious\ bulge)$ **then predict** *yes*
else if $(stop\ reason = suspicious\ object)$ **then predict** *yes*
else predict *no*

Figure 5: An example rule list that predicts whether a weapon will be found on a stopped individual who is frisked or searched, for the NYCLU stop-and-frisk data set. Across 10 cross-validation folds, the other optimal rule lists found by CORELS ($\lambda = 0.01$) contain the same or equivalent rules, up to a permutation. See also Figure 23 in Appendix F.

the optimal rule lists are relatively robust across cross-validation folds: the rules are nearly the same, up to permutations of the prefix rules. For smaller values of the regularization parameter, we observe less robustness, as rule lists are allowed to grow in length. For the sets of optimal rule lists represented in Figures 3, 4, and 5, each set could be equivalently expressed as a DNF rule; *e.g.*, this is easy to see when the prefix rules all predict the positive class label and the default rule predicts the negative class label. Our objective is not designed to enforce any of these properties, though some may be seen as desirable.

As we demonstrate in §6.6, optimal rule lists learned by CORELS achieve accuracies that are competitive with a suite of other models, including black box COMPAS scores. See Appendix F for additional listings of optimal rule lists found by CORELS, for each of our prediction problems, across cross-validation folds, for different regularization parameters λ.

Weapon prediction $(\lambda = 0.01$, **Feature Set C**)

if $(stop\ reason = suspicious\ object)$ **then predict** *yes*
else if $(location = transit\ authority)$ **then predict** *yes*
else predict *no*

Weapon prediction $(\lambda = 0.01$, **Feature Set D**)

if $(stop\ reason = suspicious\ object)$ **then predict** *yes*
else if $(inside\ or\ outside = outside)$ **then predict** *no*
else predict *yes*

Weapon prediction $(\lambda = 0.005$, **Feature Set C**)

if $(stop\ reason = suspicious\ object)$ **then predict** *yes*
else if $(location = transit\ authority)$ **then predict** *yes*
else if $(location = housing\ authority)$ **then predict** *no*
else if $(city = Manhattan)$ **then predict** *yes*
else predict *no*

Weapon prediction $(\lambda = 0.005$, **Feature Set D**)

if $(stop\ reason = suspicious\ object)$ **then predict** *yes*
else if $(stop\ reason = acting\ as\ lookout)$ **then predict** *no*
else if $(stop\ reason = fits\ description)$ **then predict** *no*
else if $(stop\ reason = furtive\ movements)$ **then predict** *no*
else predict *yes*

Figure 6: Example optimal rule lists for the NYPD stop-and-frisk data set, found by CORELS. Feature Set C contains attributes for 'location' and 'city', while Feature Set D does not. For each choice of regularization parameter and feature set, the rule lists learned by CORELS, across all 10 cross-validation folds, contain the same or equivalent rules, up to a permutation, with the exception of a single fold (Feature Set C, $\lambda = 0.005$). For a complete listing, see Figures 21 and 22 in Appendix F.

6.4 Comparison of CORELS to the Black Box COMPAS Algorithm

The accuracies of rule lists learned by CORELS are competitive with scores generated by the black box COMPAS algorithm at predicting two-year recidivism for the ProPublica data set (Figure 9). Across 10 cross-validation folds, optimal rule lists learned by CORELS (Figure 4, $\lambda = 0.005$) have a mean test accuracy of 0.665, with standard deviation 0.018. The COMPAS algorithm outputs scores between 1 and 10, representing low (1-4), medium (5-7), and high (8-10) risk for recidivism. As in the analysis by Larson et al. (2016), we interpret a medium or high score as a positive prediction for two-year recidivism, and a low score as a negative prediction. Across the 10 test sets, the COMPAS algorithm scores obtain a mean accuracy of 0.660, with standard deviation 0.019.

Figure 7 shows that CORELS and COMPAS perform similarly across both black and white individuals. Both algorithms have much higher true positive rates (TPR's) and false positive rates (FPR's) for blacks than whites (left), and higher true negative rates (TNR's)

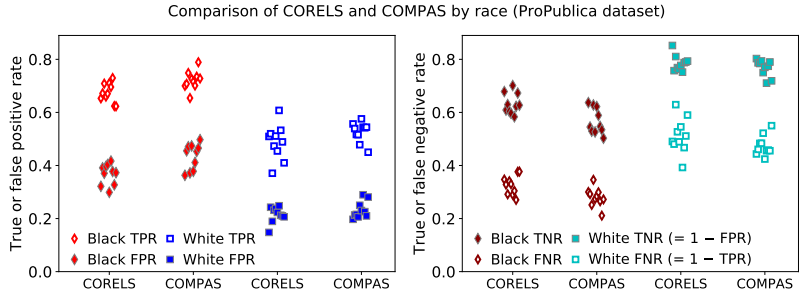


Figure 7: Comparison of TPR and FPR (left), as well as TNR and FNR (right), for different races in the ProPublica data set, for CORELS and COMPAS, across 10 cross-validation folds.

and false negative rates (FNR's) for whites than blacks (right). The fact that COMPAS has higher FPR's for blacks and higher FNR's for whites was a central observation motivating ProPublica's claim that COMPAS is racially biased (Larson et al., 2016). The fact that CORELS' models are so simple, with almost the same results as COMPAS, and contain only counts of past crimes, age, and gender, indicates possible explanations for the uneven predictions of both COMPAS and CORELS among blacks and whites. In particular, blacks evaluated within Broward County tend to be younger and have longer criminal histories within the data set, (on average, 4.4 crimes for blacks versus 2.6 crimes for whites) leading to higher FPR's for blacks and higher FNR's for whites. This aspect of the data could help to explain why ProPublica concluded that COMPAS was racially biased.

Similar observations have been reported for other datasets, namely that complex machine learning models do not have an advantage over simpler transparent models (Tollenaar and van der Heijden, 2013; Bushway, 2013; Zeng et al., 2017). There are many definitions of fairness, and it is not clear whether CORELS' models are fair either, but it is much easier to debate about the fairness of a model when it is transparent. Additional fairness constraints or transparency constraints can be placed on CORELS' models if desired, though one would need to edit our bounds (§3) and implementation (§5) to impose more constraints.

Regardless of whether COMPAS is racially biased (which our analysis does not indicate is necessarily true as long as criminal history and age are allowed to be considered as features), COMPAS may have many other fairness defects that might be considered serious. Many of COMPAS's survey questions are direct inquiries about socioeconomic status. For instance, a sample COMPAS survey[6] asks: "Is it easy to get drugs in your neighborhood?," "How often do you have barely enough money to get by?," "Do you frequently get jobs that don't pay more than minimum wage?," "How often have you moved in the last 12 months?" COMPAS's survey questions also ask about events that were not caused by the person who

6. A sample COMPAS survey contributed by Julia Angwin, ProPublica, can be found at `https://www.documentcloud.org/documents/2702103-Sample-Risk-Assessment-COMPAS-CORE.html`.

is being evaluated, such as: "If you lived with both parents and they later separated, how old were you at the time?," "Was one of your parents ever sent to jail or prison?," "Was your mother ever arrested, that you know of?"

The fact that COMPAS requires over 130 questions to be answered, many of whose answers may not be verifiable, means that the computation of the COMPAS score is prone to errors. Even the Arnold Foundation's "public-safety assessment" (PSA) score—which is completely transparent, and has only 9 factors—has been miscalculated in serious criminal trials, leading to a recent lawsuit (Westervelt, 2017). It is substantially more difficult to obtain the information required to calculate COMPAS scores than PSA scores (with over 14 times the number of survey questions). This significant discrepancy suggests that COMPAS scores are more fallible than PSA scores, as well as even simpler models, like those produced by CORELS. Some of these problems could be alleviated by using only data within electronic records that can be automatically calculated, instead of using information entered by hand and/or collected via subjective surveys.

The United States government pays Northpointe (now called Equivant) to use COMPAS. In light of our observations that CORELS is as accurate as COMPAS on a real-world data set where COMPAS is used in practice, CORELS predicts similarly to COMPAS for both blacks and whites, and CORELS' models are completely transparent, it is not clear what value COMPAS scores possess. Our experiments also indicate that the proprietary survey data required to compute COMPAS scores has not boosted its prediction accuracy above that of transparent models in practice.

Risk predictions are important for the integrity of the judicial system; judges cannot be expected to keep entire databases in their heads to calculate risks, whereas models (when used correctly) can help to ensure equity. Risk prediction models also have the potential to heavily impact how efficient the judicial system is, in terms of bail and parole decisions; efficiency in this case means that dangerous individuals are not released, whereas non-dangerous individuals are granted bail or parole. High stakes decisions, such as these, are ideal applications for machine learning algorithms that produce transparent models from high dimensional data.

Currently, justice system data does not support highly accurate risk predictions, but current risk models are useful in practice, and these risk predictions will become more accurate as more and higher quality data are made available.

6.5 Comparison of CORELS to a Heuristic Model for Weapon Prediction

CORELS generates simple, accurate models for the task of weapon prediction, using the NYPD stop-and-frisk data set. Our approach offers a principled alternative to heuristic models proposed by Goel et al. (2016), who develop a series of regression models to analyze racial disparities in New York City's stop-and-frisk policy for a related, larger data set. In particular, the authors arrive at a heuristic that they suggest could potentially help police officers more effectively decide when to frisk and/or search stopped individuals, *i.e.*, when such interventions are likely to discover criminal possession of a weapon (CPW). Starting from a full regression model with 7,705 variables, the authors reduce this to a smaller model with 98 variables; from this, they keep three variables with the largest coefficients. This gives

a heuristic model of the form $ax + by + cz \geq T$, where

$$x = \mathbb{1}[\textit{stop reason} = \textit{suspicious object}]$$
$$y = \mathbb{1}[\textit{stop reason} = \textit{suspicious bulge}]$$
$$z = \mathbb{1}[\textit{additional circumstances} = \textit{sights and sounds of criminal activity}],$$

and T is a threshold, such that the model predicts CPW when the threshold is met or exceeded. We focus on their approach that uses a single threshold, rather than precinct-specific thresholds. To increase ease-of-use, the authors round the coefficients to the nearest integers, which gives $(a, b, c) = (3, 1, 1)$; this constrains the threshold to take one of six values, $T \in \{0, 1, 2, 3, 4, 5\}$. To employ this heuristic model in the field, "...officers simply need to add at most three small, positive integers ... and check whether the sum exceeds a fixed threshold..." (Goel et al., 2016).

Figure 8 directly compares various models learned by CORELS to the heuristic models, using the same data set as Goel et al. (2016) and 10-fold cross-validation. Recall that we train on resampled data to correct for class imbalance; we evaluate with respect to test sets that have been formed without resampling. For CORELS, the models correspond to the rule lists illustrated in Figure 6 from Section 6.3, and Figures 21 and 22 in Appendix F, we consider both Feature Sets C and D and both regularization parameters $\lambda = 0.005$ and 0.01. The top panel plots the fraction of weapons recovered as a function of the fraction of stops where the individual was frisked and/or searched. Goel et al. (2016) target models that efficiently recover a majority of weapons (while also minimizing racial disparities, which we do not address here). Interestingly, the models learned by CORELS span a significant region that is not available to the heuristic model, which would require larger or non-integer parameters to access the region. The region is possibly desirable, since it includes models ($\lambda = 0.005$, bright red) that recover a majority ($\geq 50\%$) of weapons (that are known in the data set). More generally, CORELS' models all recover at least 40% of weapons on average, *i.e.*, more weapons than any of the heuristic models with $T \geq 2$, which recover less than 25% of weapons on average. At the same time, CORELS' models all require well under 25% of stops—significantly less than the heuristic model with $T = 1$, which requires over 30% of stops to recover a fraction of weapons comparable to the CORELS model that recovers the most weapons.

The bottom panel in Figure 8 plots both TPR and FPR and labels model size, for each of the models in the top panel. For the heuristic, we define model size as the number of model parameters; for CORELS, we use the number of rules in the rule list, which is equal to the number of leaves when we view a rule list as a decision tree. The heuristic models all have 4 parameters, while the different CORELS models have either 3 or approximately 5 rules. CORELS' models are thus approximately as small, interpretable, and transparent as the heuristic models; furthermore, their predictions are straightforward to compute, without even requiring arithmetic.

6.6 Predictive Performance and Model Size for CORELS and Other Algorithms

We ran a 10-fold cross validation experiment using CORELS and eight other algorithms: logistic regression, support vector machines (SVM), AdaBoost, CART, C4.5, random for-

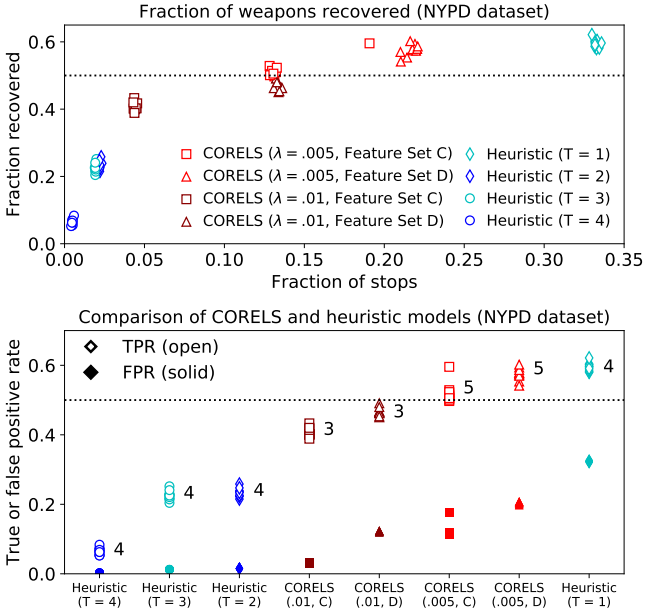


Figure 8: Weapon prediction with the NYPD stop-and-frisk data set, for various models learned by CORELS and the heuristic model by Goel et al. (2016), across 10 cross-validation folds. Note that the fraction of weapons recovered (top) is equal to the TPR (bottom, open markers). Markers above the dotted horizontal lines at the value 0.5 correspond to models that recover a majority of weapons (that are known in the data set). Top: Fraction of weapons recovered as a function of the fraction of stops where the individual was frisked and/or searched. In the legend, entries for CORELS (red markers) indicate the regularization parameter (λ) and whether or not extra location features were used ("location"); entries for the heuristic model (blue markers) indicate the threshold value (T). The results we report for the heuristic model are our reproduction of the results reported in Figure 9 by Goel et al. (2016) (first four open circles in that figure, from left to right; we exclude the trivial open circle showing 100% of weapons recovered at 100% of stops, obtained by setting the threshold at 0). Bottom: Comparison of TPR (open markers) and FPR (solid markers) for various CORELS and heuristic models. Models are sorted left-to-right by TPR. Markers and abbreviated horizontal tick labels correspond to the legend in the top figure. Numbers in the plot label model size; there was no variation in model size across folds, except for a single fold for CORELS ($\lambda = 0.005$, Feature Set C), which found a model of size 6.

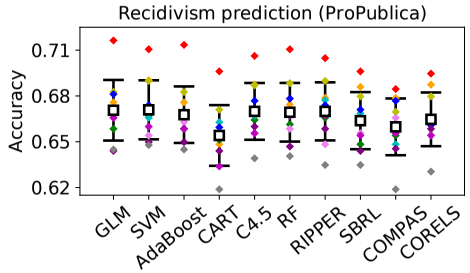


Figure 9: Two-year recidivism prediction for the ProPublica COMPAS data set. Comparison of CORELS and a panel of nine other algorithms: logistic regression (GLM), support vector machines (SVM), AdaBoost, CART, C4.5, random forests (RF), RIPPER, scalable Bayesian rule lists (SBRL), and COMPAS. For CORELS, we use regularization parameter $\lambda = 0.005$.

est (RF), RIPPER, and scalable Bayesian rule lists (SBRL).[7] We use standard R packages, with default parameter settings, for the first seven algorithms.[8] We use the same antecedent sets as input to the two rule list learning algorithms, CORELS and SBRL; for the other algorithms, the inputs are binary feature sets corresponding to the single clause antecedents in the aforementioned antecedent sets (see Appendix E).

Figure 9 shows that for the ProPublica data set, there were no statistically significant differences in test accuracies across algorithms, the difference between folds was far larger than the difference between algorithms. These algorithms also all perform similarly to the black box COMPAS algorithm. Figure 10 shows that for the NYCLU data set, logistic regression, SVM, and AdaBoost have the highest TPR's and also the highest FPR's; we show TPR and FPR due to class imbalance. For this problem, CORELS obtains an intermediate TPR, compared to other algorithms, while achieving a relatively low FPR. We conclude that CORELS produces models whose predictive performance is comparable to or better than those found via other algorithms.

Figures 11 and 12 summarize differences in predictive performance and model size for CORELS and other tree (CART, C4.5) and rule list (RIPPER, SBRL) learning algorithms. Here, we vary different algorithm parameters, and increase the number of iterations for SBRL to 10,000. For two-year recidivism prediction with the ProPublica data set (Figure 11), we plot both training and test accuracy, as a function of the number of leaves in the learned model. Due to class imbalance for the weapon prediction problem with the NYCLU stop-and-frisk data set (Figure 12), we plot both true positive rate (TPR) and false positive rate (FPR), again as a function of the number of leaves. For both problems, CORELS can learn short rule lists without sacrificing predictive performance. For listings of example optimal rule lists that correspond to the results for CORELS summarized here,

7. For SBRL, we use the C implementation at `https://github.com/Hongyuy/sbrlmod`. By default, SBRL sets $\eta = 3$, $\lambda = 9$, the number of chains to 11 and iterations to 1,000.
8. For CART, C4.5 (J48), and RIPPER, we use the R packages rpart, RWeka, and caret, respectively. By default, CART uses complexity parameter $cp = 0.01$ and C4.5 uses complexity parameter $C = 0.25$.

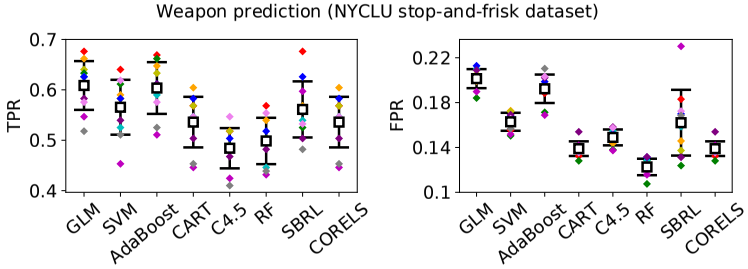


Figure 10: TPR (left) and FPR (right) for the test set, for CORELS and a panel of seven other algorithms, for the weapon prediction problem with the NYCLU stop-and-frisk data set. Means (white squares), standard deviations (error bars), and values (colors correspond to folds), for 10-fold cross-validation experiments. For CORELS, we use $\lambda = 0.01$. Note that we were unable to execute RIPPER for the NYCLU problem.

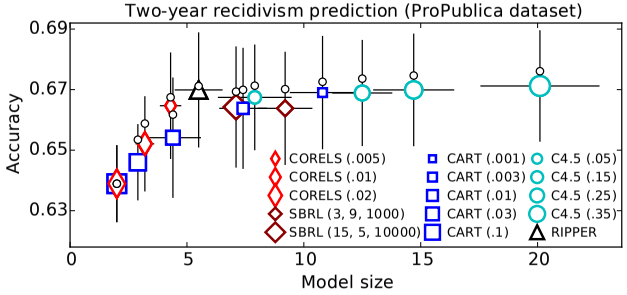


Figure 11: Training and test accuracy as a function of model size, across different methods, for two-year recidivism prediction with the ProPublica COMPAS data set. In the legend, numbers in parentheses are algorithm parameters that we vary for CORELS (λ), CART (cp), C4.5 (C), and SBRL (η, λ, i), where i is the number of iterations. Legend markers and error bars indicate means and standard deviations, respectively, across cross-validation folds. Small circles mark training accuracy means. None of the models exhibit significant overfitting; mean training accuracy never exceeds mean test accuracy by more than about 0.01.

see Appendix F. Also see Figure 25 in Appendix G; it uses the larger NYPD data set and is similar to Figure 12.

In Figure 4, we used CORELS to identify short rule lists, depending on only three features—age, prior convictions, and sex—that achieve test accuracy comparable to COMPAS (Figure 9, also see Angelino et al., 2017). If we restrict CORELS to search the space of rule lists formed from only age and prior convictions ($\lambda = 0.005$), the optimal rule lists it

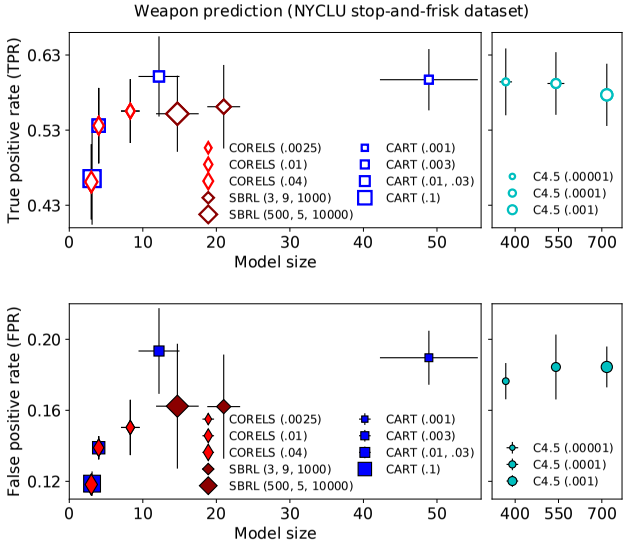


Figure 12: TPR (top) and FPR (bottom) for the test set, as a function of model size, across different methods, for weapon prediction with the NYCLU stop-and-frisk data set. In the legend, numbers in parentheses are algorithm parameters, as in Figure 11. Legend markers and error bars indicate means and standard deviations, respectively, across cross-validation folds. C4.5 finds large models for all tested parameters.

if $(priors > 3)$ **then predict** *yes*
else if $(age < 25)$ **and** $(priors = 2-3)$ **then predict** *yes*
else predict *no*

Figure 13: When restricted to two features (age, priors), CORELS ($\lambda = 0.005$) finds the same rule list across 10 cross-validation folds.

finds achieve test accuracy that is again comparable to COMPAS. CORELS identifies the same rule list across all 10 folds of 10-fold cross-validation experiments (Figure 13). In work subsequent to ours (Angelino et al., 2017), Dressel and Farid (2018) confirmed this result, in the sense that they used logistic regression to construct a linear classifier with age and prior convictions, and also achieved similar accuracy to COMPAS. However, computing a logistic regression model requires multiplication and addition, and their model cannot easily be computed, in the sense that it requires a calculator (and thus is potentially error-prone). Our rule lists require no such computation.

CORELS with different regularization parameters (NYCLU stop-and-frisk data set)

λ	Total time (s)	Time to optimum (s)	Max evaluated prefix length	Optimal prefix length
.04	.61 (.03)	.002 (.001)	6	2
.01	70 (6)	.008 (.002)	11	3
.0025	1600 (100)	56 (74)	16-17	6-10

λ	Lower bound evaluations ($\times 10^6$)	Total queue insertions ($\times 10^3$)	Max queue size ($\times 10^3$)
.04	.070 (.004)	2.2 (.1)	.9 (.1)
.01	7.5 (.6)	210 (20)	130 (10)
.0025	150 (10)	4400 (300)	2500 (170)

Table 3: Summary of CORELS executions, for the NYCLU stop-and-frisk data set ($M = 46$), for same three regularization parameter (λ) values as in Figure 12. The columns report the total execution time, time to optimum, maximum evaluated prefix length, optimal prefix length, number of times we completely evaluate a prefix d_p's lower bound $b(d_p, \mathbf{x}, \mathbf{y})$, total number of queue insertions (this number is equal to the number of cache insertions), and the maximum queue size. For prefix lengths, we report single values or ranges corresponding to the minimum and maximum observed values; in the other columns, we report means (and standard deviations) over 10 cross-validation folds. See also Figures 14 and 15.

6.7 CORELS Execution Traces, for Different Regularization Parameters

In this section, we illustrate several views of CORELS execution traces, for the NYCLU stop-and-frisk data set with $M = 46$ antecedents, for the same three regularization parameters ($\lambda = .04, .01, .025$) as in Figure 12.

Table 3 summarizes execution traces across all 10 cross-validation folds. For each value of λ, CORELS achieves the optimum in a small fraction of the total execution time. As λ decreases, these times increase because the search problems become more difficult, as is summarized by the observation that CORELS must evaluate longer prefixes; consequently, our data structures grow in size. We report the total number of elements inserted into the queue and the maximum queue size; recall from §5 that the queue elements correspond to the trie's leaves, and that the symmetry-aware map elements correspond to the trie's nodes.

The upper panels in Figure 14 plot example execution traces, from a single cross-validation fold, of both the current best objective value R^c and the lower bound $b(d_p, \mathbf{x}, \mathbf{y})$ of the prefix d_p being evaluated. These plots illustrate that CORELS certifies optimality when the lower bound matches the objective value. The lower panels in Figure 14 plot corresponding traces of an upper bound on the size of the remaining search space (Theorem 7), and illustrate that as λ decreases, it becomes more difficult to eliminate regions of the search space. For Figure 14, we dynamically and incrementally calculate $\lfloor \log_{10} \Gamma(R^c, Q) \rfloor$, which adds some computational overhead; we do not calculate this elsewhere unless noted.

Figure 15 visualizes the elements in CORELS' logical queue, for each of the executions in Figure 14. Recall from §5.5 that the logical queue corresponds to elements in the (physical)

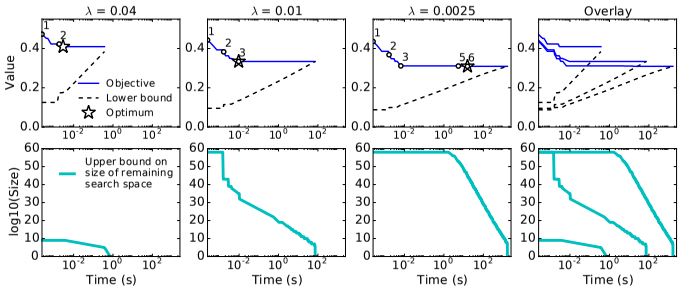


Figure 14: Example executions of CORELS, for the NYCLU stop-and-frisk data set ($M = 46$). See also Table 3 and Figure 15. Top: Objective value (solid line) and lower bound (dashed line) for CORELS, as a function of wall clock time (log scale). Numbered points along the trace of the objective value indicate when the length of the best known rule list changes and are labeled by the new length. For each value of λ, a star marks the optimum objective value and time at which it was achieved. Bottom: $\lfloor \log_{10} \Gamma(R^c, Q) \rfloor$, as a function of wall clock time (log scale), where $\Gamma(R^c, Q)$ is the upper bound on remaining search space size (Theorem 7). Rightmost panels: For visual comparison, we overlay the execution traces from the panels to the left, for the three different values of λ.

queue that have not been garbage collected from the trie; these are prefixes that CORELS has already evaluated and whose children the algorithm plans to evaluate next. As an execution progresses, longer prefixes are placed in the queue; as λ decreases, the algorithm must spend more time evaluating longer and longer prefixes.

6.8 Efficacy of CORELS Algorithm Optimizations

This section examines the efficacy of each of our bounds and data structure optimizations. We remove a single bound or data structure optimization from our final implementation and measure how the performance of our algorithm changes. We examine these performance traces on both the NYCLU and the ProPublica data sets, and highlight the result that on different problems, the relative performance improvements of our optimizations can vary.

Table 4 provides summary statistics for experiments using the full CORELS implementation (first row) and five variants (subsequent rows) that each remove a specific optimization: (1) Instead of a priority queue (§5.2) ordered by the objective lower bound, we use a queue that implements breadth-first search (BFS). (2) We remove checks that would trigger pruning via our lower bounds on antecedent support (Theorem 10) and accurate antecedent support (Theorem 11). (3) We remove the effect of our lookahead bound (Lemma 2), which otherwise tightens the objective lower bound by an amount equal to the regularization parameter λ. (4) We disable the symmetry-aware map (§5.3), our data structure that enables pruning triggered by the permutation bound (Corollary 16). (5) We do not identify sets of equivalent points, which we otherwise use to tighten the objective lower bound via the equivalent points bound (Theorem 20).

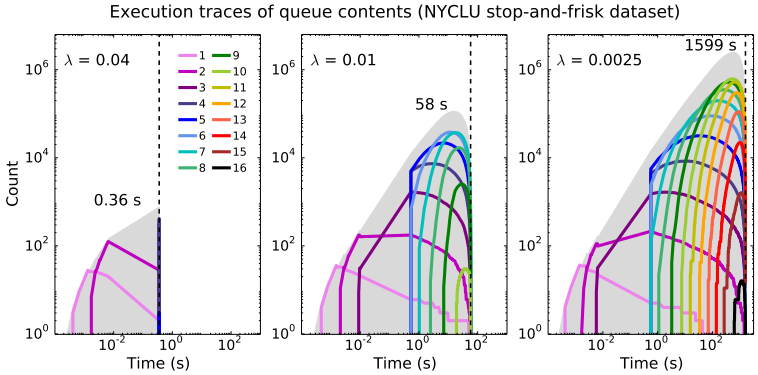


Figure 15: Summary of CORELS' logical queue, for the NYCLU stop-and-frisk data set ($M = 46$), for same three regularization parameters as in Figure 12 and Table 3. Solid lines plot the numbers of prefixes in the logical queue (log scale), colored by length (legend), as a function of wall clock time (log scale). All plots are generated using a single, representative cross-validation training set. For each execution, the gray shading fills in the area beneath the total number of queue elements, *i.e.*, the sum over all lengths; we also annotate the total time in seconds, marked with a dashed vertical line.

Removing any single optimization increases total execution time, by varying amounts across these optimizations. Similar to our experiments in §6.7, we always encounter the optimal rule list in far less time than it takes to certify optimality. As in Table 3, we report metrics that are all proxies for how much computational work our algorithm must perform; these metrics thus scale with the overall slowdown with respect to CORELS execution time (Table 4, first column).

Figure 16 visualizes execution traces of the elements in CORELS' logical queue, similar to Figure 15, for a single, representative cross-validation fold. Panels correspond to different removed optimizations, as in Table 4. These plots demonstrate that our optimizations reduce the number of evaluated prefixes and are especially effective at limiting the number of longer evaluated prefixes. For the ProPublica data set, the most important optimization is the equivalent points bound—without it, we place prefixes of at least length 10 in our queue, and must terminate these executions before they are complete. In contrast, CORELS and most other variants evaluate only prefixes up to at most length 5, except for the variant without the lookahead bound, which evaluates prefixes up to length 6.

Table 5 and Figure 17 summarize an analogous set of experiments for the NYCLU data set. Note that while the equivalent points bound proved to be the most important optimization for the ProPublica data set, the symmetry-aware map is the crucial optimization for the NYCLU data set.

Finally, Figure 18 highlights the most significant algorithm optimizations for our prediction problems: the equivalent points bound for the ProPublica data set (left) and the

Per-component performance improvement (ProPublica data set)

Algorithm variant	Total time (min)	Slow-down	Time to optimum (s)	Max evaluated prefix length
CORELS	.98 (.6)	—	1 (1)	5
No priority queue (BFS)	1.03 (.6)	1.1×	2 (4)	5
No support bounds	1.5 (.9)	1.5×	1 (2)	5
No lookahead bound	12.3 (6.2)	13.3×	1 (1)	6
No symmetry-aware map	9.1 (6.4)	8.4×	2 (3)	5
No equivalent points bound*	>130 (2.6)	>180×	>1400 (2000)	≥11

Algorithm variant	Lower bound evaluations ($\times 10^6$)	Total queue insertions ($\times 10^6$)	Max queue size ($\times 10^6$)
CORELS	26 (15)	.29 (.2)	.24 (.1)
No priority queue (BFS)	27 (16)	.33 (.2)	.20 (.1)
No support bounds	42 (25)	.40 (.2)	.33 (.2)
No lookahead bound	320 (160)	3.6 (1.8)	3.0 (1.5)
No symmetry-aware map	250 (180)	2.5 (1.7)	2.4 (1.7)
No equivalent points bound*	>940 (5)	>510 (1.1)	>500 (1.2)

Table 4: Per-component performance improvement, for the ProPublica data set ($\lambda = 0.005$, $M = 122$). The columns report the total execution time, time to optimum, maximum evaluated prefix length, number of times we completely evaluate a prefix d_p's lower bound $b(d_p, \mathbf{x}, \mathbf{y})$, total number of queue insertions (which is equal to the number of cache insertions), and maximum logical queue size. The first row shows CORELS; subsequent rows show variants that each remove a specific implementation optimization or bound. (We are not measuring the cumulative effects of removing a sequence of components.) All rows represent complete executions that certify optimality, except those labeled 'No equivalent points bound,' for which each execution was terminated due to memory constraints, once the size of the cache reached 5×10^8 elements, after consuming ∼250GB RAM. In all but the final row and column, we report means (and standard deviations) over 10 cross-validation folds. We also report the mean slowdown in total execution time, with respect to CORELS. In the final row, we report the mean (and standard deviation) of the incomplete execution time and corresponding slowdown, and a lower bound on the mean time to optimum; in the remaining fields, we report minimum values across folds. See also Figure 16.
* Only 7 out of 10 folds achieve the optimum before being terminated.

symmetry-aware map for the NYCLU data set (right). For CORELS (thin lines) with the ProPublica recidivism data set (left), the objective drops quickly, achieving the optimal value within a second. CORELS certifies optimality in about a minute—the objective lower bound steadily converges to the optimal objective (top) as the search space shrinks (bottom). As in Figure 14, we dynamically and incrementally calculate $\lfloor \log_{10} \Gamma(R^c, Q) \rfloor$, where $\Gamma(R^c, Q)$ is the upper bound (13) on remaining search space size (Theorem 7); this adds some computational overhead. In the same plots (left), we additionally highlight a

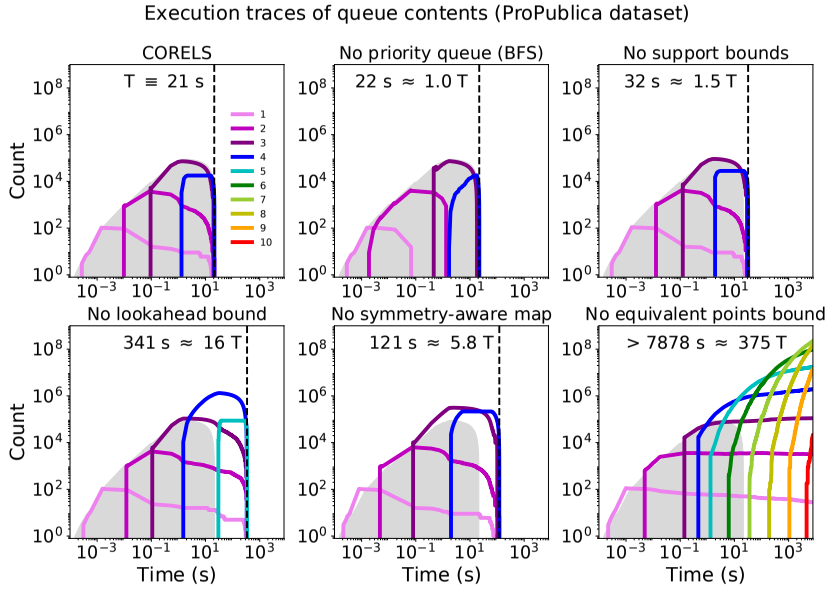


Figure 16: Summary of the logical queue's contents, for full CORELS (top left) and five variants that each remove a specific implementation optimization or bound, for the ProPublica data set ($\lambda = 0.005$, $M = 122$). See also Table 4. Solid lines plot the numbers of prefixes in the logical queue (log scale), colored by length (legend), as a function of wall clock time (log scale). All plots are generated using a single, representative cross-validation training set. The gray shading fills in the area beneath the total number of queue elements for CORELS, *i.e.*, the sum over all lengths in the top left figure. For comparison, we replicate the same gray region in the other five subfigures. For each execution, we indicate the total time in seconds, relative to the full CORELS implementation (T = 21 s), and with a dashed vertical line. The execution without the equivalent points bound (bottom right) is incomplete.

separate execution of CORELS without the equivalent points bound (Theorem 20) (thick lines). After more than 2 hours, the execution is still far from complete; in particular, the lower bound is far from the optimum objective value (top) and much of the search space remains unexplored (bottom). For the NYCLU stop-and-frisk data set (right), CORELS achieves the optimum objective in well under a second, and certifies optimality in a little over a minute. CORELS without the permutation bound (Corollary 16), and thus the symmetry-aware map, requires more than an hour, *i.e.*, orders of magnitude more time, to complete (thick lines).

Per-component performance improvement (NYCLU stop-and-frisk data set)

Algorithm variant	Total time (min)	Slow-down	Time to optimum (μs)	Max evaluated prefix length
CORELS	1.1 (.1)	—	8.9 (.1)	11
No priority queue (BFS)	2.2 (.2)	2.0×	110 (10)	11
No support bounds	1.2 (.1)	1.1×	8.8 (.8)	11
No lookahead bound	1.7 (.2)	1.6×	7.3 (1.8)	11-12
No symmetry-aware map	$>$ 73 (5)	$>$ 68×	$>$ 7.6 (.4)	$>$ 10
No equivalent points bound	4 (.3)	3.8×	6.4 (.9)	14

Algorithm variant	Lower bound evaluations ($\times 10^6$)	Total queue insertions ($\times 10^5$)	Max queue size ($\times 10^5$)
CORELS	7 (1)	2.0 (.2)	1.3 (.1)
No priority queue (BFS)	14 (1)	4.1 (.4)	1.4 (.1)
No support bounds	8 (1)	2.1 (.2)	1.3 (.1)
No lookahead bound	11 (1)	3.2 (.3)	2.1 (.2)
No symmetry-aware map	$>$ 390 (40)	$>$ 1000 (0)	$>$ 900 (10)
No equivalent points bound	33 (2)	9.4 (.7)	6.0 (.4)

Table 5: Per-component performance improvement, as in Table 4, for the NYCLU stop-and-frisk data set ($\lambda = 0.01$, $M = 46$). All rows except those labeled 'No symmetry-aware map' represent complete executions. A single fold running without a symmetry-aware map required over 2 days to complete, so in order to run all 10 folds above, we terminated execution after the prefix tree (§5.1) reached 10^8 nodes. See Table 4 for a detailed caption, and also Figure 17.

Algorithmic approach	Max evaluated prefix length	Lower bound evaluations	Predicted runtime	
CORELS	5	2.8×10^7	36 seconds	
Brute force	5	2.5×10^{10}	9.0 hours	$\approx 3.3 \times 10^4$ s
Brute force	10	5.0×10^{20}	21×10^6 years	$\approx 6.5 \times 10^{14}$ s
CORELS (1984)	5	2.8×10^7	13.5 days	$\approx 1.2 \times 10^6$ s

Table 6: Algorithmic speedup for the ProPublica data set ($\lambda = 0.005$, $M = 122$). Solving this problem using brute force is impractical due to the inability to explore rule lists of reasonable lengths. Removing only the equivalent points bound requires exploring prefixes of up to length 10 (see Table 4), a clearly intractable problem. Even with all of our improvements, however, it is only recently that processors have been fast enough for this type of discrete optimization algorithm to succeed.

6.9 Algorithmic Speedup

Table 6 shows the overall speedup of CORELS compared to a naïve implementation and demonstrates the infeasibility of running our algorithm 30 years ago. Consider an execution of CORELS for the ProPublica data set, with $M = 122$ antecedents, that evaluates prefixes

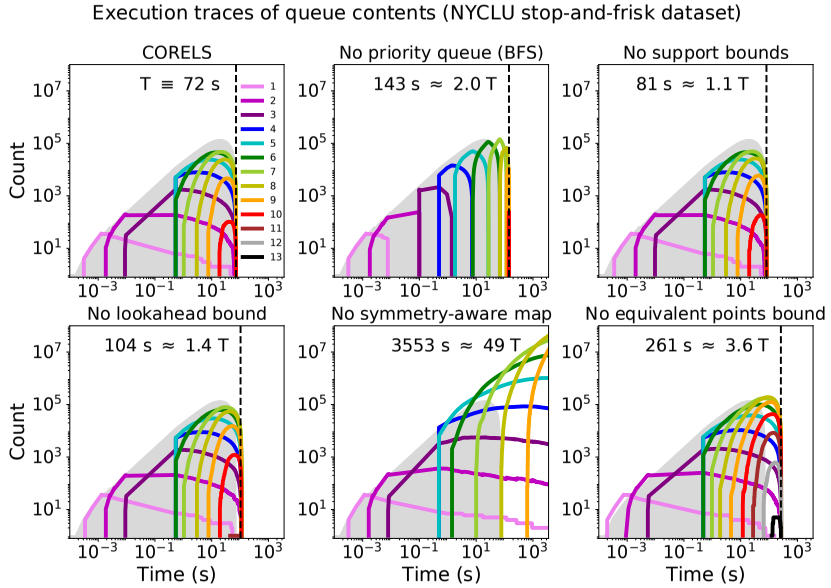


Figure 17: Summary of the logical queue's contents, for full CORELS (top left) and five variants that each remove a specific implementation optimization or bound, for the NYCLU stop-and-frisk data set ($\lambda = 0.01$, $M = 46$), as in Table 5. The execution without the symmetry-aware map (bottom center) is incomplete. See Figure 16 for a detailed caption.

up to length 5 in order to certify optimality (Table 4). A brute force implementation that naïvely considers all prefixes of up to length 5 would evaluate 2.5×10^{10} prefixes. As shown in Figure 4, the optimal rule list has prefix length 3, thus the brute force algorithm would identify the optimal rule list. However, for this approach to certify optimality, it would have to consider far longer prefixes. Without our equivalent points bound, but with all of our other optimizations, we evaluate prefixes up to at least length 10 (see Table 4 and Figure 16)—thus a brute force algorithm would have to evaluate prefixes of length 10 or longer. Naïvely evaluating all prefixes up to length 10 would require looking at 5.0×10^{20} different prefixes.

However, CORELS examines only 28 million prefixes in total—a reduction of 893× compared to examining all prefixes up to length 5 and a reduction of 1.8×10^{13} for the case of length 10. On a laptop, we require about 1.3 μs to evaluate a single prefix (given by dividing the number of lower bound evaluations by the total time in Table 4). Our runtime is only about 36 seconds, but the naïve solutions of examining all prefixes up to lengths 5

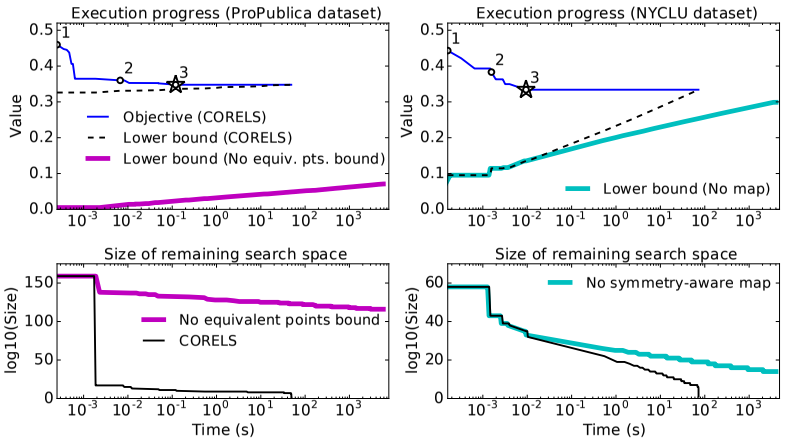


Figure 18: Execution progress of CORELS and selected variants, for the ProPublica ($\lambda = 0.005$, $M = 122$) (left) and NYCLU ($\lambda = 0.01$, $M = 46$) (right) data sets. Top: Objective value (thin solid lines) and lower bound (dashed lines) for CORELS, as a function of wall clock time (log scale). Numbered points along the trace of the objective value indicate when the length of the best known rule list changes, and are labeled by the new length. CORELS quickly achieves the optimal value (star markers), and certifies optimality when the lower bound matches the objective value. On the left, a separate and significantly longer execution of CORELS without the equivalent points (Theorem 20) bound remains far from complete, and its lower bound (thick solid line) far from the optimum. On the right, a separate execution of CORELS without the permutation bound (Corollary 16), and thus the symmetry-aware map, requires orders of magnitude more time to complete. Bottom: $\lfloor \log_{10} \Gamma(R^c, Q) \rfloor$, as a function of wall clock time (log scale), where $\Gamma(R^c, Q)$ is the upper bound (13) on remaining search space size (Theorem 7). For these problems, the equivalent points (left) and permutation (right) bounds are responsible for the ability of CORELS to quickly eliminate most of the search space (thin solid lines); the remaining search space decays much more slowly without these bounds (thick solid lines).

and 10 would take 9 hours and 21 million years, respectively. It is clear that brute force would not scale to larger problems.

We compare our current computing circumstances to those of 1984, the year when CART was published. Moore's law holds that computing power doubled every 18 months from 1984 to 2006. This is a period of 264 months, which means computing power has gone up by at least a factor of 32,000 since 1984. Thus, even with our algorithmic and data structural improvements, CORELS would have required about 13.5 days in 1984—an unreasonable amount of time. Our advances are meaningful only because we can run them on a modern

system. Combining our algorithmic improvements with the increase in modern processor speeds, our algorithm runs more than 10^{13} times faster than a naïve implementation would have in 1984. This helps explain why neither our algorithm, nor other branch-and-bound variants, had been developed before now.

7. Summary and Future Work on Bounds

Here, we highlight our most significant bounds, as well as directions for future work based on bounds that we have yet to leverage in practice.

In empirical studies, we found our equivalent support (§3.10, Theorem 15) and equivalent points (§3.14, Theorem 20) bounds to yield the most significant improvements in algorithm performance. In fact, they sometimes proved critical for finding solutions and proving optimality, even on small problems.

Accordingly, we would hope that our similar support bound (§3.13, Theorem 18) could be useful; understanding how to efficiently exploit this result in practice represents an important direction for future work. In particular, this type of bound may lead to principled approximate variants of our approach.

We presented several sets of bounds in which at least one bound was strictly tighter than the other(s). For example, the lower bound on accurate antecedent support (Theorem 11) is strictly tighter than the lower bound on accurate support (Theorem 10). It might seem that we should only use this tighter bound, but in practice, we can use both—the looser bound can be checked before completing the calculation required to check the tighter bound. Similarly, the equivalent support bound (Theorem 15) is more general than the special case of the permutation bound (Corollary 16). We have implemented data structures, which we call symmetry-aware maps, to support both of these bounds, but have not yet identified an efficient approach for supporting the more general equivalent points bound. A good solution may be related to the challenge of designing an efficient data structure to support the similar support bound.

We also presented results on antecedent rejection that unify our understanding of our lower (§3.7) and upper bounds (§3.8) on antecedent support. In a preliminary implementation (not described here), we experimented with special data structures to support the direct use of our observation that antecedent rejection propagates (§3.9, Theorem 12). We leave the design of efficient data structures for this task as future work.

During execution, we find it useful to calculate an upper bound on the size of the remaining search space—*e.g.*, via Theorem 7, or the looser Proposition 9, which incurs less computational overhead—since these provide meaningful information about algorithm progress and allow us to estimate the remaining execution time. As we illustrated in Section 6.8, these calculations also help us qualify the impact of different algorithmic bounds, *e.g.*, by comparing executions that keep or remove bounds.

When our algorithm terminates, it outputs an optimal solution of the training optimization problem, with a certificate of optimality. On a practical note, our approach can also provide useful results even for incomplete executions. As shown earlier, we have empirically observed that our algorithm often identifies the optimal rule list very quickly, compared to the total time required to prove optimality, *e.g.*, in seconds, versus minutes, respectively. Furthermore, our objective's lower bounds allow us to place an upper bound on the size of

the remaining search space, and provides guarantees on the quality of a solution output by an incomplete execution.

The order in which we evaluate prefixes can impact the rate at which we prune the search space, and thus the total runtime. We think that it is possible to design search policies that significantly improve performance.

8. Conclusion and More Possible Directions for Future Work

Finally, we would like to clarify some limitations of CORELS. As far as we can tell, CORELS is the current best algorithm for solving a specialized optimal decision tree problem. While our approach scales well to large numbers of observations, it could have difficulty proving optimality for problems with many possibly relevant features that are highly correlated, when large regions of the search space might be challenging to exclude.

CORELS is not designed for raw image processing or other problems where the features themselves are not interpretable. It could instead be used as a final classifier for image processing problems where the features were created beforehand; for instance, one could create classifiers for each part of an image, and use CORELS to create a final combined classifier. The notions of interpretability used in image classification tend to be completely different from those for structured data where each feature is separately meaningful (e.g., see Li et al., 2018). For structured data, decision trees, along with scoring systems, tend to be popular forms of transparent models. Scoring systems are sparse linear models with integer coefficients, and they can also be created from data (Ustun and Rudin, 2017, 2016).

In some of our experiments, CORELS produces a DNF formula by coincidence, but it might be possible to create a much simpler version of CORELS that only produces DNF formulae. This could build off previous algorithms for creating an optimal DNF formula (Rijnbeek and Kors, 2010; Wang et al., 2016, 2017).

CORELS does not automatically rank the subgroups in order of the likelihood of a positive outcome; doing so would require an algorithm such as Falling Rule Lists (Wang and Rudin, 2015a; Chen and Rudin, 2018), which forces the estimated probabilities to decrease along the list. Furthermore, while CORELS does not technically produce estimates of $\mathbb{P}(Y = 1 \,|\, x)$, one could form such an estimate by computing the empirical proportion $\hat{\mathbb{P}}(Y = 1 \,|\, x \text{ obeys } p_k)$ for each antecedent p_k. CORELS is also not designed to assist with causal inference applications, since it does not estimate the effect of a treatment via the conditional difference $\mathbb{P}(Y = 1 \,|\, \text{treatment} = \text{True}, x) - \mathbb{P}(Y = 1 \,|\, \text{treatment} = \text{False}, x)$. Alternative algorithms that estimate conditional differences with interpretable rule lists include Causal Falling Rule Lists (Wang and Rudin, 2015b), Cost-Effective Interpretable Treatment Regimes (CITR) (Lakkaraju and Rudin, 2017), and an approach by Zhang et al. (2015) for constructing interpretable and parsimonious treatment regimes. Alternatively, one could use a complex machine learning model to predict outcomes for the treatment group and a separate complex model for the control group that would allow counterfactuals to be estimated for each observation; from there, CORELS could be applied to produce a transparent model for personalized treatment effects. A similar approach to this was taken by Goh and Rudin (2018), who use CORELS to understand a black box causal model.

CORELS could be adapted to handle cost-sensitive learning or weighted regularization. This would require creating more general versions of our theorems, which would be an extension of this work.

While CORELS does not directly handle continuous variables, we have found that it is not difficult in practice to construct a rule set that is sufficient for creating a useful model. It may be possible to use techniques such as Fast Boxes (Goh and Rudin, 2014) to discover useful and interpretable rules for continuous data that can be used within CORELS.

An interesting direction for future research would be to create a hybrid interpretable/ black box model in the style of Wang (2018), where the rule list would eliminate large parts of the space away from the decision boundary, and the observations that remain are evaluated by a black box model rather than a default rule.

Lastly, CORELS does not create generic single-variable-split decision trees. CORELS optimizes over rule lists, which are one-sided decision trees; in our setting, the leaves of these 'trees' are conjunctions. It may be possible to generalize ideas from our approach to handle generic decision trees, which could be an interesting project for future work. There are more symmetries to handle in that case, since there would be many equivalent decision trees, leading to challenges in developing symmetry-aware data structures.

Acknowledgments

E.A. conducted most of this work while supported by the Miller Institute for Basic Research in Science, University of California, Berkeley, and hosted by Prof. M.I. Jordan at RISELab. C.D.R. is supported in part by MIT-Lincoln Labs and the National Science Foundation under IIS-1053407. E.A. would like to thank E. Jonas, E. Kohler, and S. Tu for early implementation guidance, A. D'Amour for pointing out the work by Goel et al. (2016), V. Kanade, S. McCurdy, J. Schleier-Smith and E. Thewalt for helpful conversations, and members of RISELab, SAIL, and the UC Berkeley Database Group for their support and feedback. We thank H. Yang and B. Letham for sharing advice and code for processing data and mining rules, B. Coker for his critical advice on using the ProPublica COMPAS data set, as well as V. Kaxiras and A. Saligrama for their recent contributions to our implementation and for creating the CORELS website. We are very grateful to our editor and anonymous reviewers.

Appendix A. Excessive Antecedent Support

Theorem 21 (Upper bound on antecedent support) *Let $d^* = (d_p, \delta_p, q_0, K)$ be any optimal rule list with objective R^*, i.e., $d^* \in \operatorname{argmin}_d R(d, \mathbf{x}, \mathbf{y})$, and let $d_p = (p_1, \ldots, p_{k-1}, p_k, \ldots, p_K)$ be its prefix. For each $k \leq K$, antecedent p_k in d_p has support less than or equal to the fraction of all data not captured by preceding antecedents, by an amount greater than the regularization parameter λ:*

$$\operatorname{supp}(p_k, \mathbf{x} \,|\, d_p) \leq 1 - \operatorname{supp}(d_p^{k-1}, \mathbf{x}) - \lambda, \tag{37}$$

where $d_p^{k-1} = (p_1, \ldots, p_{k-1})$. For the last antecedent, i.e., when $p_k = p_K$, equality implies that there also exists a shorter optimal rule list $d' = (d_p^{K-1}, \delta_p', q_0', K-1) \in \operatorname{argmin}_d R(d, \mathbf{x}, \mathbf{y})$.

Proof First, we focus on the last antecedent p_{K+1} in a rule list d'. Let $d = (d_p, \delta_p, q_0, K)$ be a rule list with prefix $d_p = (p_1, \ldots, p_K)$ and objective $R(d, \mathbf{x}, \mathbf{y}) \geq R^*$, where $R^* \equiv \min_D R(D, \mathbf{x}, \mathbf{y})$ is the optimal objective. Let $d' = (d'_p, \delta'_p, q'_0, K+1)$ be a rule list whose prefix $d'_p = (p_1, \ldots, p_K, p_{K+1})$ starts with d_p and ends with a new antecedent p_{K+1}. Suppose p_{K+1} in the context of d'_p captures nearly all data not captured by d_p, except for a fraction ϵ upper bounded by the regularization parameter λ:

$$1 - \text{supp}(d_p, \mathbf{x}) - \text{supp}(p_{K+1}, \mathbf{x} \,|\, d'_p) \equiv \epsilon \leq \lambda.$$

Since d'_p starts with d_p, its prefix misclassification error is at least as great; the only discrepancy between the misclassification errors of d and d' can come from the difference between the support of the set of data not captured by d_p and the support of p_{K+1}:

$$|\ell(d', \mathbf{x}, \mathbf{y}) - \ell(d, \mathbf{x}, \mathbf{y})| \leq 1 - \text{supp}(d_p, \mathbf{x}) - \text{supp}(p_{K+1}, \mathbf{x} \,|\, d'_p) = \epsilon.$$

The best outcome for d' would occur if its misclassification error were smaller than that of d by ϵ, therefore

$$\begin{aligned} R(d', \mathbf{x}, \mathbf{y}) &= \ell(d', \mathbf{x}, \mathbf{y}) + \lambda(K+1) \\ &\geq \ell(d, \mathbf{x}, \mathbf{y}) - \epsilon + \lambda(K+1) = R(d, \mathbf{x}, \mathbf{y}) - \epsilon + \lambda \geq R(d, \mathbf{x}, \mathbf{y}) \geq R^*. \end{aligned}$$

d' is an optimal rule list, *i.e.*, $d' \in \text{argmin}_D R(D, \mathbf{x}, y)$, if and only if $R(d', \mathbf{x}, \mathbf{y}) = R(d, \mathbf{x}, \mathbf{y}) = R^*$, which requires $\epsilon = \lambda$. Otherwise, $\epsilon < \lambda$, in which case

$$R(d', \mathbf{x}, \mathbf{y}) \geq R(d, \mathbf{x}, \mathbf{y}) - \epsilon + \lambda > R(d, \mathbf{x}, \mathbf{y}) \geq R^*,$$

therefore d' is not optimal, *i.e.*, $d' \notin \text{argmin}_D R(D, \mathbf{x}, \mathbf{y})$. This demonstrates the desired result for $k = K$.

In the remainder, we prove the bound in (37) by contradiction, in the context of a rule list d''. Let d and d' retain their definitions from above, thus as before, that the data not captured by d'_p has normalized support $\epsilon \leq \lambda$, *i.e.*,

$$1 - \text{supp}(d'_p, \mathbf{x}) = 1 - \text{supp}(d_p, \mathbf{x}) - \text{supp}(p_{K+1}, \mathbf{x} \,|\, d'_p) = \epsilon \leq \lambda.$$

Thus for any rule list d'' whose prefix $d''_p = (p_1, \ldots, p_{K+1}, \ldots, p_{K'})$ starts with d'_p and ends with one or more additional rules, each additional rule p_k has support $\text{supp}(p_k, \mathbf{x} \,|\, d''_p) \leq \epsilon \leq \lambda$, for all $k > K+1$. By Theorem 10, all of the additional rules have insufficient support, therefore d''_p cannot be optimal, *i.e.*, $d'' \notin \text{argmin}_D R(D, \mathbf{x}, \mathbf{y})$. ■

Similar to Theorem 10, our lower bound on antecedent support, we can apply Theorem 21 in the contexts of both constructing rule lists and rule mining (§3.1). Theorem 21 implies that if we only seek a single optimal rule list, then during branch-and-bound execution, we can prune a prefix if we ever add an antecedent with support too similar to the support of the set of data not captured by the preceding antecedents. One way to view this result is that if $d = (d_p, \delta_p, q_0, K)$ and $d' = (d'_p, \delta'_p, q'_0, K+1)$ are rule lists such that d'_p starts with d_p and ends with an antecedent that captures all or nearly all data not captured by d_p, then the new rule in d' behaves similar to the default rule of d. As a result, the misclassification error of d' must be similar to that of d, and any reduction may not be sufficient to offset the penalty for longer prefixes.

Proposition 22 (Excessive antecedent support propagates) *Define $\phi(d_p)$ as in (19), and let $d_p = (p_1, \ldots, p_K)$ be a prefix, such that its last antecedent p_K has excessive support, i.e., the opposite of the bound in (37):*

$$\text{supp}(p_K, \mathbf{x} \,|\, d_p) > 1 - \text{supp}(d_p^{K-1}, \mathbf{x}) - \lambda,$$

where $d_p^{K-1} = (p_1, \ldots, p_{K-1})$. Let $D = (D_p, \Delta_p, Q_0, \kappa)$ be any rule list with prefix $D_p = (P_1, \ldots, P_\kappa)$ such that D_p starts with $D_p^{K'-1} = (P_1, \ldots, P_{K'-1}) \in \phi(d_p^{K-1})$ and $P_{K'} = p_K$. It follows that $P_{K'}$ has excessive support in prefix D_p, and furthermore, $D \notin \text{argmin}_d R(d, \mathbf{x}, \mathbf{y})$.

Proof Since $D_p^{K'} = (P_1, \ldots, P_{K'})$ contains all the antecedents in d_p, we have that

$$\text{supp}(D_p^{K'}, \mathbf{x}) \geq \text{supp}(d_p, \mathbf{x}).$$

Expanding these two terms gives

$$\begin{aligned} \text{supp}(D_p^{K'}, \mathbf{x}) &= \text{supp}(D_p^{K'-1}, \mathbf{x}) + \text{supp}(P_{K'}, \mathbf{x} \,|\, D_p) \\ &\geq \text{supp}(d_p, \mathbf{x}) = \text{supp}(d_p^{K-1}, \mathbf{x}) + \text{supp}(p_K, \mathbf{x} \,|\, d_p) > 1 - \lambda. \end{aligned}$$

Rearranging gives

$$\text{supp}(P_{K'}, \mathbf{x} \,|\, D_p) > 1 - \text{supp}(D_p^{K'-1}, \mathbf{x}) - \lambda,$$

thus $P_{K'}$ has excessive support in D_p. By Theorem 21, $D \notin \text{argmin}_d R(d, \mathbf{x}, \mathbf{y})$. ■

Appendix B. Proof of Theorem 15 (Equivalent Support Bound)

We begin by defining four related rule lists. First, let $d = (d_p, \delta_p, q_0, K)$ be a rule list with prefix $d_p = (p_1, \ldots, p_K)$ and labels $\delta_p = (q_1, \ldots, q_K)$. Second, let $D = (D_p, \Delta_p, Q_0, \kappa)$ be a rule list with prefix $D_p = (P_1, \ldots, P_\kappa)$ that captures the same data as d_p, and labels $\Delta_p = (Q_1, \ldots, Q_\kappa)$. Third, let $d' = (d'_p, \delta'_p, q'_0, K') \in \sigma(d_p)$ be any rule list whose prefix starts with d_p, such that $K' \geq K$. Denote the prefix and labels of d' by $d'_p = (p_1, \ldots, p_K, p_{K+1}, \ldots, p_{K'})$ and $\delta_p = (q_1, \ldots, q_{K'})$, respectively. Finally, define $D' = (D'_p, \Delta'_p, Q'_0, \kappa') \in \sigma(D_p)$ to be the 'analogous' rule list, *i.e.*, whose prefix $D'_p = (P_1, \ldots, P_\kappa, P_{\kappa+1}, \ldots, P_{\kappa'}) = (P_1, \ldots, P_\kappa, p_{K+1}, \ldots, p_{K'})$ starts with D_p and ends with the same $K' - K$ antecedents as d'_p. Let $\Delta'_p = (Q_1, \ldots, Q_{\kappa'})$ denote the labels of D'.

Next, we claim that the difference in the objectives of rule lists d' and d is the same as the difference in the objectives of rule lists D' and D. Let us expand the first difference as

$$\begin{aligned} &R(d', \mathbf{x}, \mathbf{y}) - R(d, \mathbf{x}, \mathbf{y}) = \ell(d', \mathbf{x}, \mathbf{y}) + \lambda K' - \ell(d, \mathbf{x}, \mathbf{y}) - \lambda K \\ &= \ell_p(d'_p, \delta'_p, \mathbf{x}, \mathbf{y}) + \ell_0(d'_p, q'_0, \mathbf{x}, \mathbf{y}) - \ell_p(d_p, \delta_p, \mathbf{x}, \mathbf{y}) - \ell_0(d_p, q_0, \mathbf{x}, \mathbf{y}) + \lambda(K' - K). \end{aligned}$$

Similarly, let us expand the second difference as

$$\begin{aligned} &R(D', \mathbf{x}, \mathbf{y}) - R(D, \mathbf{x}, \mathbf{y}) = \ell(D', \mathbf{x}, \mathbf{y}) + \lambda\kappa' - \ell(D, \mathbf{x}, \mathbf{y}) - \lambda\kappa \\ &= \ell_p(D'_p, \Delta'_p, \mathbf{x}, \mathbf{y}) + \ell_0(D'_p, Q'_0, \mathbf{x}, \mathbf{y}) - \ell_p(D_p, \Delta_p, \mathbf{x}, \mathbf{y}) - \ell_0(D_p, Q_0, \mathbf{x}, \mathbf{y}) + \lambda(K' - K), \end{aligned}$$

where we have used the fact that $\kappa' - \kappa = K' - K$.

The prefixes d_p and D_p capture the same data. Equivalently, the set of data that is not captured by d_p is the same as the set of data that is not captured by D_p, *i.e.*,

$$\{x_n : \neg \operatorname{cap}(x_n, d_p)\} = \{x_n : \neg \operatorname{cap}(x_n, D_p)\}.$$

Thus, the corresponding rule lists d and D share the same default rule, *i.e.*, $q_0 = Q_0$, yielding the same default rule misclassification error:

$$\ell_0(d_p, q_0, \mathbf{x}, \mathbf{y}) = \ell_0(D_p, Q_0, \mathbf{x}, \mathbf{y}).$$

Similarly, prefixes d'_p and D'_p capture the same data, and thus rule lists d' and D' have the same default rule misclassification error:

$$\ell_0(d_p, q_0, \mathbf{x}, \mathbf{y}) = \ell_0(D_p, Q_0, \mathbf{x}, \mathbf{y}).$$

At this point, to demonstrate our claim relating the objectives of d, d', D, and D', what remains is to show that the difference in the misclassification errors of prefixes d'_p and d_p is the same as that between D'_p and D_p. We can expand the first difference as

$$\ell_p(d'_p, \delta'_p, \mathbf{x}, \mathbf{y}) - \ell_p(d_p, \delta_p, \mathbf{x}, \mathbf{y}) = \frac{1}{N} \sum_{n=1}^{N} \sum_{k=K+1}^{K'} \operatorname{cap}(x_n, p_k \,|\, d'_p) \wedge \mathbb{1}[q_k \neq y_n],$$

where we have used the fact that since d'_p starts with d_p, the first K rules in d'_p make the same mistakes as those in d_p. Similarly, we can expand the second difference as

$$\begin{aligned}
\ell_p(D'_p, \Delta'_p, \mathbf{x}, \mathbf{y}) - \ell_p(D_p, \Delta_p, \mathbf{x}, \mathbf{y}) &= \frac{1}{N} \sum_{n=1}^{N} \sum_{k=\kappa+1}^{\kappa'} \operatorname{cap}(x_n, P_k \,|\, D'_p) \wedge \mathbb{1}[Q_k \neq y_n] \\
&= \frac{1}{N} \sum_{n=1}^{N} \sum_{k=K+1}^{K'} \operatorname{cap}(x_n, p_k \,|\, D'_p) \wedge \mathbb{1}[Q_k \neq y_n] \\
&= \frac{1}{N} \sum_{n=1}^{N} \sum_{k=K+1}^{K'} \operatorname{cap}(x_n, p_k \,|\, d'_p) \wedge \mathbb{1}[q_k \neq y_n] \qquad (38) \\
&= \ell_p(d'_p, \delta'_p, \mathbf{x}, \mathbf{y}) - \ell_p(d_p, \delta_p, \mathbf{x}, \mathbf{y}).
\end{aligned}$$

To justify the equality in (38), we observe first that prefixes D'_p and d'_p start with κ and K antecedents, respectively, that capture the same data. Second, prefixes D'_p and d'_p end with exactly the same ordered list of $K' - K$ antecedents, therefore for any $k = 1, \ldots, K' - K$, antecedent $P_{\kappa+k} = p_{K+k}$ in D'_p captures the same data as p_{K+k} captures in d'_p. It follows that the corresponding labels are all equivalent, *i.e.*, $Q_{\kappa+k} = q_{K+k}$, for all $k = 1, \ldots, K' - K$, and consequently, the prefix misclassification error associated with the last $K' - K$ antecedents of d'_p is the same as that of D'_p. We have therefore shown that the difference between the objectives of d' and d is the same as that between D' and D, *i.e.*,

$$R(d', \mathbf{x}, \mathbf{y}) - R(d, \mathbf{x}, \mathbf{y}) = R(D', \mathbf{x}, \mathbf{y}) - R(D, \mathbf{x}, \mathbf{y}). \qquad (39)$$

Next, suppose that the objective lower bounds of d and D obey $b(d_p, \mathbf{x}, \mathbf{y}) \leq b(D_p, \mathbf{x}, \mathbf{y})$, therefore

$$\begin{aligned}
R(d, \mathbf{x}, \mathbf{y}) &= \ell_p(d_p, \delta_p, \mathbf{x}, \mathbf{y}) + \ell_0(d_p, q_0, \mathbf{x}, \mathbf{y}) + \lambda K \\
&= b(d_p, \mathbf{x}, \mathbf{y}) + \ell_0(d_p, q_0, \mathbf{x}, \mathbf{y}) \\
&\leq b(D_p, \mathbf{x}, \mathbf{y}) + \ell_0(d_p, q_0, \mathbf{x}, \mathbf{y}) = b(D_p, \mathbf{x}, \mathbf{y}) + \ell_0(D_p, Q_0, \mathbf{x}, \mathbf{y}) = R(D, \mathbf{x}, \mathbf{y}).
\end{aligned} \qquad (40)$$

Now let d^* be an optimal rule list with prefix constrained to start with d_p,

$$d^* \in \operatorname*{argmin}_{d^\dagger \in \sigma(d_p)} R(d^\dagger, \mathbf{x}, \mathbf{y}),$$

and let K^* be the length of d^*. Let D^* be the analogous κ^*-rule list whose prefix starts with D_p and ends with the same $K^* - K$ antecedents as d^*, where $\kappa^* = \kappa + K^* - K$. By (39),

$$R(d^*, \mathbf{x}, \mathbf{y}) - R(d, \mathbf{x}, \mathbf{y}) = R(D^*, \mathbf{x}, \mathbf{y}) - R(D, \mathbf{x}, \mathbf{y}). \qquad (41)$$

Furthermore, we claim that D^* is an optimal rule list with prefix constrained to start with D_p,

$$D^* \in \operatorname*{argmin}_{D^\dagger \in \sigma(D_p)} R(D^\dagger, \mathbf{x}, \mathbf{y}). \qquad (42)$$

To demonstrate (42), we consider two separate scenarios. In the first scenario, prefixes d_p and D_p are composed of the same antecedents, *i.e.*, the two prefixes are equivalent up to a permutation of their antecedents, and as a consequence, $\kappa = K$ and $\kappa^* = K^*$. Here, every rule list $d'' \in \sigma(d_p)$ that starts with d_p has an analogue $D'' \in \sigma(D_p)$ that starts with D_p, such that d'' and D'' obey (39), and vice versa, and thus (42) is a direct consequence of (41).

In the second scenario, prefixes d_p and D_p are not composed of the same antecedents. Define $\phi = \{p_k : (p_k \in d_p) \wedge (p_k \notin D_p)\}$ to be the set of antecedents in d_p that are not in D_p, and define $\Phi = \{P_k : (P_k \in D_p) \wedge (P_k \notin d_p)\}$ to be the set of antecedents in D_p that are not in d_p; either $\phi \neq \emptyset$, or $\Phi \neq \emptyset$, or both.

Suppose $\phi \neq \emptyset$, and let $p \in \phi$ be an antecedent in ϕ. It follows that there exists a subset of rule lists in $\sigma(D_p)$ that do not have analogues in $\sigma(d_p)$. Let $D'' \in \sigma(D_p)$ be such a rule list, such that its prefix $D''_p = (P_1, \ldots, P_\kappa, \ldots, p, \ldots)$ starts with D_p and contains p among its remaining antecedents. Since p captures a subset of the data that d_p captures, and D_p captures the same data as d_p, it follows that p does not capture any data in D''_p, *i.e.*,

$$\frac{1}{N} \sum_{n=1}^{N} \operatorname{cap}(x_n, p \,|\, D''_p) = 0 \leq \lambda.$$

By Theorem 10, antecedent p has insufficient support in D'', and thus D'' cannot be optimal, *i.e.*, $D'' \notin \operatorname{argmin}_{D^\dagger \in \sigma(D_p)} R(D^\dagger, \mathbf{x}, \mathbf{y})$. By a similar argument, if $\Phi \neq \emptyset$ and $P \in \Phi$, and $d'' \in \sigma(d_p)$ is any rule list whose prefix starts with d_p and contains antecedent P, then d'' cannot be optimal, *i.e.*, $d'' \notin \operatorname{argmin}_{d^\dagger \in \sigma(d_p)} R(d^\dagger, \mathbf{x}, \mathbf{y})$.

To finish justifying claim (42) for the second scenario, first define

$$\tau(d_p, \Phi) \equiv \{d'' = (d''_p, \delta''_p, q''_0, K'') : d'' \in \sigma(d_p) \text{ and } p_k \notin \Phi, \forall p_k \in d''_p\} \subset \sigma(d_p)$$

to be the set of all rule lists whose prefixes start with d_p and do not contain any antecedents in Φ. Now, recognize that the optimal prefixes in $\tau(d_p, \Phi)$ and $\sigma(d_p)$ are the same, *i.e.*,

$$\operatorname*{argmin}_{d^\dagger \in \tau(d_p, \Phi)} R(d^\dagger, \mathbf{x}, \mathbf{y}) = \operatorname*{argmin}_{d^\dagger \in \sigma(d_p)} R(d^\dagger, \mathbf{x}, \mathbf{y}),$$

and similarly, the optimal prefixes in $\tau(D_p, \phi)$ and $\sigma(D_p)$ are the same, *i.e.*,

$$\operatorname*{argmin}_{D^\dagger \in \tau(D_p, \phi)} R(D^\dagger, \mathbf{x}, \mathbf{y}) = \operatorname*{argmin}_{D^\dagger \in \sigma(D_p)} R(D^\dagger, \mathbf{x}, \mathbf{y}).$$

Since we have shown that every $d'' \in \tau(d_p, \Phi)$ has a direct analogue $D'' \in \tau(D_p, \phi)$, such that d'' and D'' obey (39), and vice versa, we again have (42) as a consequence of (41).

We can now finally combine (40) and (42) to obtain the desired inequality in (21):

$$\min_{d' \in \sigma(d_p)} R(d', \mathbf{x}, \mathbf{y}) = R(d^*, \mathbf{x}, \mathbf{y}) \leq R(D^*, \mathbf{x}, \mathbf{y}) = \min_{D' \in \sigma(D_p)} R(D', \mathbf{x}, \mathbf{y}).$$

Appendix C. Proof of Theorem 18 (Similar Support Bound)

We begin by defining four related rule lists. First, let $d = (d_p, \delta_p, q_0, K)$ be a rule list with prefix $d_p = (p_1, \ldots, p_K)$ and labels $\delta_p = (q_1, \ldots, q_K)$. Second, let $D = (D_p, \Delta_p, Q_0, \kappa)$ be a rule list with prefix $D_p = (P_1, \ldots, P_\kappa)$ and labels $\Delta_p = (Q_1, \ldots, Q_\kappa)$. Define ω as in (22) and Ω as in (23), and require that $\omega, \Omega \leq \lambda$. Third, let $d' = (d'_p, \delta'_p, q'_0, K') \in \sigma(d_p)$ be any rule list whose prefix starts with d_p, such that $K' \geq K$. Denote the prefix and labels of d' by $d'_p = (p_1, \ldots, p_K, p_{K+1}, \ldots, p_{K'})$ and $\delta_p = (q_1, \ldots, q_{K'})$, respectively. Finally, define $D' = (D'_p, \Delta'_p, Q'_0, \kappa') \in \sigma(D_p)$ to be the 'analogous' rule list, *i.e.*, whose prefix $D'_p = (P_1, \ldots, P_\kappa, P_{\kappa+1}, \ldots, P_{\kappa'}) = (P_1, \ldots, P_\kappa, p_{K+1}, \ldots, p_{K'})$ starts with D_p and ends with the same $K' - K$ antecedents as d'_p. Let $\Delta'_p = (Q_1, \ldots, Q_{\kappa'})$ denote the labels of D'.

The smallest possible objective for D', in relation to the objective of d', reflects both the difference between the objective lower bounds of D and d and the largest possible discrepancy between the objectives of d' and D'. The latter would occur if d' misclassified all the data corresponding to both ω and Ω while D' correctly classified this same data, thus

$$R(D', \mathbf{x}, \mathbf{y}) \geq R(d', \mathbf{x}, \mathbf{y}) + b(D_p, \mathbf{x}, \mathbf{y}) - b(d_p, \mathbf{x}, \mathbf{y}) - \omega - \Omega. \tag{43}$$

Now let D^* be an optimal rule list with prefix constrained to start with D_p,

$$D^* \in \operatorname*{argmin}_{D^\dagger \in \sigma(D_p)} R(D^\dagger, \mathbf{x}, \mathbf{y}),$$

and let κ^* be the length of D^*. Also let d^* be the analogous K^*-rule list whose prefix starts with d_p and ends with the same $\kappa^* - \kappa$ antecedents as D^*, where $K^* = K + \kappa^* - \kappa$. By (43),

we obtain the desired inequality in (24):

$$\begin{aligned}\min_{D^\dagger \in \sigma(D_p)} R(D^\dagger, \mathbf{x}, \mathbf{y}) &= R(D^*, \mathbf{x}, \mathbf{y}) \\ &\geq R(d^*, \mathbf{x}, \mathbf{y}) + b(D_p, \mathbf{x}, \mathbf{y}) - b(d_p, \mathbf{x}, \mathbf{y}) - \omega - \Omega \\ &\geq \min_{d^\dagger \in \sigma(d_p)} R(d^\dagger, \mathbf{x}, \mathbf{y}) + b(D_p, \mathbf{x}, \mathbf{y}) - b(d_p, \mathbf{x}, \mathbf{y}) - \omega - \Omega.\end{aligned}$$

Appendix D. Proof of Theorem 20 (Equivalent Points Bound)

We derive a lower bound on the default rule misclassification error $\ell_0(d_p, q_0, \mathbf{x}, \mathbf{y})$, analogous to the lower bound (26) on the misclassification error $\ell(d, \mathbf{x}, \mathbf{y})$ in the proof of Proposition 19. As before, we sum over all sets of equivalent points, and then for each such set, we count differences between class labels and the minority class label of the set, instead of counting mistakes made by the default rule:

$$\begin{aligned}\ell_0(d_p, q_0, \mathbf{x}, \mathbf{y}) &= \frac{1}{N} \sum_{n=1}^{N} \neg \operatorname{cap}(x_n, d_p) \wedge \mathbb{1}[q_0 \neq y_n] \\ &= \frac{1}{N} \sum_{u=1}^{U} \sum_{n=1}^{N} \neg \operatorname{cap}(x_n, d_p) \wedge \mathbb{1}[q_0 \neq y_n] \, \mathbb{1}[x_n \in e_u] \\ &\geq \frac{1}{N} \sum_{u=1}^{U} \sum_{n=1}^{N} \neg \operatorname{cap}(x_n, d_p) \wedge \mathbb{1}[y_n = q_u] \, \mathbb{1}[x_n \in e_u] = b_0(d_p, \mathbf{x}, \mathbf{y}),\end{aligned} \tag{44}$$

where the final equality comes from the definition of $b_0(d_p, \mathbf{x}, \mathbf{y})$ in (28). Since we can write the objective $R(d, \mathbf{x}, \mathbf{y})$ as the sum of the objective lower bound $b(d_p, \mathbf{x}, \mathbf{y})$ and default rule misclassification error $\ell_0(d_p, q_0, \mathbf{x}, \mathbf{y})$, applying (44) gives a lower bound on $R(d, \mathbf{x}, \mathbf{y})$:

$$\begin{aligned}R(d, \mathbf{x}, \mathbf{y}) = \ell_p(d_p, \delta_p, \mathbf{x}, \mathbf{y}) + \ell_0(d_p, q_0, \mathbf{x}, \mathbf{y}) + \lambda K &= b(d_p, \mathbf{x}, \mathbf{y}) + \ell_0(d_p, q_0, \mathbf{x}, \mathbf{y}) \\ &\geq b(d_p, \mathbf{x}, \mathbf{y}) + b_0(d_p, \mathbf{x}, \mathbf{y}).\end{aligned} \tag{45}$$

It follows that for any rule list $d' \in \sigma(d)$ whose prefix d'_p starts with d_p, we have

$$R(d', \mathbf{x}, \mathbf{y}) \geq b(d'_p, \mathbf{x}, \mathbf{y}) + b_0(d'_p, \mathbf{x}, \mathbf{y}). \tag{46}$$

Finally, we show that the lower bound on $R(d, \mathbf{x}, \mathbf{y})$ in (45) is not greater than the lower bound on $R(d', \mathbf{x}, \mathbf{y})$ in (46). First, let us define

$$\Upsilon(d'_p, K, \mathbf{x}, \mathbf{y}) \equiv \frac{1}{N} \sum_{u=1}^{U} \sum_{n=1}^{N} \sum_{k=K+1}^{K'} \operatorname{cap}(x_n, p_k \mid d'_p) \wedge \mathbb{1}[x_n \in e_u] \, \mathbb{1}[y_n = q_u]. \tag{47}$$

Now, we write a lower bound on $b(d'_p, \mathbf{x}, \mathbf{y})$ with respect to $b(d_p, \mathbf{x}, \mathbf{y})$:

$$
\begin{aligned}
b(d'_p, \mathbf{x}, \mathbf{y}) &= \ell_p(d'_p, \delta_p, \mathbf{x}, \mathbf{y}) + \lambda K' = \frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K'} \mathrm{cap}(x_n, p_k \,|\, d'_p) \wedge \mathbb{1}[q_k \neq y_n] + \lambda K' \\
&= \ell_p(d_p, \delta_p, \mathbf{x}, \mathbf{y}) + \lambda K + \frac{1}{N}\sum_{n=1}^{N}\sum_{k=K}^{K'} \mathrm{cap}(x_n, p_k \,|\, d'_p) \wedge \mathbb{1}[q_k \neq y_n] + \lambda(K' - K) \\
&= b(d_p, \mathbf{x}, \mathbf{y}) + \frac{1}{N}\sum_{n=1}^{N}\sum_{k=K+1}^{K'} \mathrm{cap}(x_n, p_k \,|\, d'_p) \wedge \mathbb{1}[q_k \neq y_n] + \lambda(K' - K) \\
&= b(d_p, \mathbf{x}, \mathbf{y}) + \frac{1}{N}\sum_{u=1}^{U}\sum_{n=1}^{N}\sum_{k=K+1}^{K'} \mathrm{cap}(x_n, p_k \,|\, d'_p) \wedge \mathbb{1}[q_k \neq y_n]\, \mathbb{1}[x_n \in e_u] + \lambda(K' - K) \\
&\geq b(d_p, \mathbf{x}, \mathbf{y}) + \frac{1}{N}\sum_{u=1}^{U}\sum_{n=1}^{N}\sum_{k=K+1}^{K'} \mathrm{cap}(x_n, p_k \,|\, d'_p) \wedge \mathbb{1}[y_n = q_u]\, \mathbb{1}[x_n \in e_u] + \lambda(K' - K) \\
&= b(d_p, \mathbf{x}, \mathbf{y}) + \Upsilon(d'_p, K, \mathbf{x}, \mathbf{y}) + \lambda(K' - K), \qquad (48)
\end{aligned}
$$

where the last equality uses (47). Next, we write $b_0(d_p, \mathbf{x}, \mathbf{y})$ with respect to $b_0(d'_p, \mathbf{x}, \mathbf{y})$,

$$
\begin{aligned}
b_0(d_p, \mathbf{x}, \mathbf{y}) &= \frac{1}{N}\sum_{u=1}^{U}\sum_{n=1}^{N} \neg\,\mathrm{cap}(x_n, d_p) \wedge \mathbb{1}[x_n \in e_u]\, \mathbb{1}[y_n = q_u] \\
&= \frac{1}{N}\sum_{u=1}^{U}\sum_{n=1}^{N} \left(\neg\,\mathrm{cap}(x_n, d'_p) + \sum_{k=K+1}^{K'} \mathrm{cap}(x_n, p_k \,|\, d'_p) \right) \wedge \mathbb{1}[x_n \in e_u]\, \mathbb{1}[y_n = q_u] \\
&= b_0(d'_p, \mathbf{x}, \mathbf{y}) + \frac{1}{N}\sum_{u=1}^{U}\sum_{n=1}^{N}\sum_{k=K+1}^{K'} \mathrm{cap}(x_n, p_k \,|\, d'_p) \wedge \mathbb{1}[x_n \in e_u]\, \mathbb{1}[y_n = q_u]. \qquad (49)
\end{aligned}
$$

Rearranging (49) gives

$$
b_0(d'_p, \mathbf{x}, \mathbf{y}) = b_0(d_p, \mathbf{x}, y) - \Upsilon(d'_p, K, \mathbf{x}, \mathbf{y}). \qquad (50)
$$

Combining (46) with first (50) and then (48) gives the desired inequality in (27):

$$
\begin{aligned}
R(d', \mathbf{x}, \mathbf{y}) &\geq b(d'_p, \mathbf{x}, \mathbf{y}) + b_0(d'_p, \mathbf{x}, \mathbf{y}) \\
&= b(d'_p, \mathbf{x}, \mathbf{y}) + b_0(d_p, \mathbf{x}, y) - \Upsilon(d'_p, K, \mathbf{x}, \mathbf{y}) \\
&\geq b(d_p, \mathbf{x}, \mathbf{y}) + \Upsilon(d'_p, K, \mathbf{x}, \mathbf{y}) + \lambda(K' - K) + b_0(d_p, \mathbf{x}, y) - \Upsilon(d'_p, K, \mathbf{x}, \mathbf{y}) \\
&= b(d_p, \mathbf{x}, \mathbf{y}) + b_0(d_p, \mathbf{x}, y) + \lambda(K' - K) \geq b(d_p, \mathbf{x}, \mathbf{y}) + b_0(d_p, \mathbf{x}, \mathbf{y}).
\end{aligned}
$$

Appendix E. Data Processing Details and Antecedent Mining

In this appendix, we provide details regarding datasets used in our experiments (Section 6).

E.1 ProPublica Recidivism Data Set

Table 7 shows the 6 attributes and corresponding 17 categorical values that we use for the ProPublica data set. From these, we construct 17 single-clause antecedents, for example, $(age = 23 - 25)$. We then combine pairs of these antecedents as conjunctions to form two-clause antecedents, *e.g.*, $(age = 23 - 25) \wedge (priors = 2 - 3)$. By virtue of our lower bound on antecedent support, (Theorem 10, §3.7), we eliminate antecedents with support less than 0.005 or greater than 0.995, since $\lambda = 0.005$ is the smallest regularization parameter value we study for this problem. With this filtering step, we generate between 121 and 123 antecedents for each fold; without it, we would instead generate about 130 antecedents as input to our algorithm.

Note that we exclude the 'current charge' attribute (which has two categorical values, 'misdemeanor' and 'felony'); for individuals in the data set booked on multiple charges, this attribute does not appear to consistently reflect the most serious charge.

Feature	Value range	Categorical values	Count
sex	—	male, female	2
age	18-96	18-20, 21-22, 23-25, 26-45, >45	5
juvenile felonies	0-20	0, >0	2
juvenile misdemeanors	0-13	0, >0	2
juvenile crimes	0-21	0, >0	2
priors	0-38	0, 1, 2-3, >3	4

Table 7: Categorical features (6 attributes, 17 values) from the ProPublica data set. We construct the feature *juvenile crimes* from the sum of *juvenile felonies*, *juvenile misdemeanors*, and the number of juvenile crimes that were neither felonies nor misdemeanors (not shown).

E.2 NYPD Stop-and-frisk Data Set

This data set is larger than, but similar to the NYCLU stop-and-frisk data set, described next.

E.3 NYCLU Stop-and-frisk Data Set

The original data set contains 45,787 records, each describing an incident involving a stopped person; the individual was frisked in 30,345 (66.3%) of records and and searched in 7,283 (15.9%). In 30,961 records, the individual was frisked and/or searched (67.6%); of those, a criminal possession of a weapon was identified 1,445 times (4.7% of these records). We remove 1,929 records with missing data, as well as a small number with extreme values for the individual's age—we eliminate those with age < 12 or > 89. This yields a set of 29,595 records in which the individual was frisked and/or searched. To address the class imbalance for this problem, we sample records from the smaller class with replacement. We generate cross-validation folds first, and then resample within each fold. In our 10-fold cross-validation experiments, each training set contains 50,743 observations. Table 8 shows the 5

categorical attributes that we use, corresponding to a total of 28 values. Our experiments use these antecedents, as well as negations of the 18 antecedents corresponding to the two features *stop reason* and *additional circumstances*, which gives a total of 46 antecedents.

Feature	Values	Count
stop reason	suspicious object, fits description, casing, acting as lookout, suspicious clothing, drug transaction, furtive movements, actions of violent crime, suspicious bulge	9
additional circumstances	proximity to crime scene, evasive response, associating with criminals, changed direction, high crime area, time of day, sights and sounds of criminal activity, witness report, ongoing investigation	9
city	Queens, Manhattan, Brooklyn, Staten Island, Bronx	5
location	housing authority, transit authority, neither housing nor transit authority	3
inside or outside	inside, outside	2

Table 8: Categorical features (5 attributes, 28 values) from the NYCLU data set.

Appendix F. Example Optimal Rule Lists, for Different Values of λ

For each of our prediction problems, we provide listings of optimal rule lists found by CORELS, across 10 cross-validation folds, for different values of the regularization parameter λ. These rule lists correspond to the results for CORELS summarized in Figures 11 and 12 (§6.6). Recall that as λ decreases, optimal rule lists tend to grow in length.

F.1 ProPublica Recidivism Data Set

We show example optimal rule lists that predict two-year recidivism. Figure 19 shows examples for regularization parameters $\lambda = 0.02$ and 0.01. Figure 20 shows examples for $\lambda = 0.005$; Figure 4 (§6.3) showed two representative examples.

For the largest regularization parameter $\lambda = 0.02$ (Figure 19), we observe that all folds identify the same length-1 rule list. For the intermediate value $\lambda = 0.01$ (Figure 19), the folds identify optimal 2-rule or 3-rule lists that contain the nearly same prefix rules, up to permutations. For the smallest value $\lambda = 0.005$ (Figure 20), the folds identify optimal 3-rule or 4-rule lists that contain the nearly same prefix rules, up to permutations. Across all three regularization parameter values and all folds, the prefix rules always predict the positive class label, and the default rule always predicts the negative class label. We note that our objective is not designed to enforce any of these properties.

Two-year recidivism prediction ($\lambda = 0.02$)

if ($priors > 3$) **then predict** *yes* ▷ **Found by all 10 folds**
else predict *no*

Two-year recidivism prediction ($\lambda = 0.01$)

if ($priors > 3$) **then predict** *yes* ▷ **Found by 3 folds**
else if ($sex = male$) **and** ($juvenile\ crimes > 0$) **then predict** *yes*
else predict *no*

if ($sex = male$) **and** ($juvenile\ crimes > 0$) **then predict** *yes* ▷ **Found by 2 folds**
else if ($priors > 3$) **then predict** *yes*
else predict *no*

if ($age = 21 - 22$) **and** ($priors = 2 - 3$) **then predict** *yes* ▷ **Found by 2 folds**
else if ($priors > 3$) **then predict** *yes*
else if ($age = 18 - 20$) **and** ($sex = male$) **then predict** *yes*
else predict *no*

if ($age = 18 - 20$) **and** ($sex = male$) **then predict** *yes* ▷ **Found by 2 folds**
else if ($priors > 3$) **then predict** *yes*
else predict *no*

if ($priors > 3$) **then predict** *yes* ▷ **Found by 1 fold**
else if ($age = 18 - 20$) **and** ($sex = male$) **then predict** *yes*
else predict *no*

Figure 19: Example optimal rule lists for the ProPublica data set, found by CORELS with regularization parameters $\lambda = 0.02$ (top), and 0.01 (bottom) across 10 cross-validation folds.

Two-year recidivism prediction ($\lambda = 0.005$)

if $(age = 18-20)$ **and** $(sex = male)$ **then predict** *yes* ▷ **Found by 4 folds**
else if $(age = 21-22)$ **and** $(priors = 2-3)$ **then predict** *yes*
else if $(priors > 3)$ **then predict** *yes*
else predict *no*

if $(age = 21-22)$ **and** $(priors = 2-3)$ **then predict** *yes* ▷ **Found by 2 folds**
else if $(priors > 3)$ **then predict** *yes*
else if $(age = 18-20)$ **and** $(sex = male)$ **then predict** *yes*
else predict *no*

if $(age = 18-20)$ **and** $(sex = male)$ **then predict** *yes* ▷ **Found by 1 fold**
else if $(priors > 3)$ **then predict** *yes*
else if $(age = 21-22)$ **and** $(priors = 2-3)$ **then predict** *yes*
else predict *no*

if $(age = 18-20)$ **and** $(sex = male)$ **then predict** *yes* ▷ **Found by 1 fold**
else if $(age = 21-22)$ **and** $(priors = 2-3)$ **then predict** *yes*
else if $(age = 23-25)$ **and** $(priors = 2-3)$ **then predict** *yes*
else if $(priors > 3)$ **then predict** *yes*
else predict *no*

if $(age = 18-20)$ **and** $(sex = male)$ **then predict** *yes* ▷ **Found by 1 fold**
else if $(age = 21-22)$ **and** $(priors = 2-3)$ **then predict** *yes*
else if $(priors > 3)$ **then predict** *yes*
else if $(age = 23-25)$ **and** $(priors = 2-3)$ **then predict** *yes*
else predict *no*

if $(age = 21-22)$ **and** $(priors = 2-3)$ **then predict** *yes* ▷ **Found by 1 fold**
else if $(age = 23-25)$ **and** $(priors = 2-3)$ **then predict** *yes*
else if $(priors > 3)$ **then predict** *yes*
else if $(age = 18-20)$ **and** $(sex = male)$ **then predict** *yes*
else predict *no*

Figure 20: Example optimal rule lists for the ProPublica data set, found by CORELS with regularization parameters $\lambda = 0.005$, across 10 cross-validation folds.

F.2 NYPD Stop-and-frisk Data Set

We show example optimal rule lists that predict whether a weapon will be found on a stopped individual who is frisked or searched, learned from the NYPD data set.

Weapon prediction ($\lambda = 0.01$, **Feature Set C**)

if $(stop\ reason = suspicious\ object)$ **then predict** *yes* ▷ **Found by 8 folds**
else if $(location = transit\ authority)$ **then predict** *yes*
else predict *no*

if $(location = transit\ authority)$ **then predict** *yes* ▷ **Found by 2 folds**
else if $(stop\ reason = suspicious\ object)$ **then predict** *yes*
else predict *no*

Weapon prediction ($\lambda = 0.005$, **Feature Set C**)

if $(stop\ reason = suspicious\ object)$ **then predict** *yes* ▷ **Found by 7 folds**
else if $(location = transit\ authority)$ **then predict** *yes*
else if $(location = housing\ authority)$ **then predict** *no*
else if $(city = Manhattan)$ **then predict** *yes*
else predict *no*

if $(stop\ reason = suspicious\ object)$ **then predict** *yes* ▷ **Found by 1 fold**
else if $(location = housing\ authority)$ **then predict** *no*
else if $(location = transit\ authority)$ **then predict** *yes*
else if $(city = Manhattan)$ **then predict** *yes*
else predict *no*

if $(stop\ reason = suspicious\ object)$ **then predict** *yes* ▷ **Found by 1 fold**
else if $(location = housing\ authority)$ **then predict** *no*
else if $(city = Manhattan)$ **then predict** *yes*
else if $(location = transit\ authority)$ **then predict** *yes*
else predict *no*

if $(stop\ reason = suspicious\ object)$ **then predict** *yes* ▷ **Found by 1 fold**
else if $(location = transit\ authority)$ **then predict** *yes*
else if $(city = Bronx)$ **then predict** *no*
else if $(location = housing\ authority)$ **then predict** *no*
else if $(stop\ reason = furtive\ movements)$ **then predict** *no*
else predict *yes*

Figure 21: Example optimal rule lists for the NYPD stop-and-frisk data set, found by CORELS with regularization parameters $\lambda = 0.01$ (top) and 0.005 (bottom), across 10 cross-validation folds.

Weapon prediction ($\lambda = 0.01$, **Feature Set D**)

if (*stop reason* = *suspicious object*) **then predict** *yes* ▷ **Found by 7 folds**
else if (*inside or outside* = *outside*) **then predict** *no*
else predict *yes*

if (*stop reason* = *suspicious object*) **then predict** *yes* ▷ **Found by 3 folds**
else if (*inside or outside* = *inside*) **then predict** *yes*
else predict *no*

Weapon prediction ($\lambda = 0.005$, **Feature Set D**)

if (*stop reason* = *suspicious object*) **then predict** *yes* ▷ **Found by 2 folds**
else if (*stop reason* = *acting as lookout*) **then predict** *no*
else if (*stop reason* = *fits description*) **then predict** *no*
else if (*stop reason* = *furtive movements*) **then predict** *no*
else predict *yes*

if (*stop reason* = *suspicious object*) **then predict** *yes* ▷ **Found by 2 folds**
else if (*stop reason* = *furtive movements*) **then predict** *no*
else if (*stop reason* = *acting as lookout*) **then predict** *no*
else if (*stop reason* = *fits description*) **then predict** *no*
else predict *yes*

if (*stop reason* = *suspicious object*) **then predict** *yes* ▷ **Found by 1 fold**
else if (*stop reason* = *acting as lookout*) **then predict** *no*
else if (*stop reason* = *furtive movements*) **then predict** *no*
else if (*stop reason* = *fits description*) **then predict** *no*
else predict *yes*

if (*stop reason* = *suspicious object*) **then predict** *yes* ▷ **Found by 1 fold**
else if (*stop reason* = *fits description*) **then predict** *no*
else if (*stop reason* = *acting as lookout*) **then predict** *no*
else if (*stop reason* = *furtive movements*) **then predict** *no*
else predict *yes*

if (*stop reason* = *suspicious object*) **then predict** *yes* ▷ **Found by 1 fold**
else if (*stop reason* = *furtive movements*) **then predict** *no*
else if (*stop reason* = *fits description*) **then predict** *no*
else if (*stop reason* = *acting as lookout*) **then predict** *no*
else predict *yes*

Figure 22: Example optimal rule lists for the NYPD stop-and-frisk data set (Feature Set D) found by CORELS with regularization parameters $\lambda = 0.01$ (top) and 0.005 (bottom), across 10 cross-validation folds. For $\lambda = 0.005$, we show results from 7 folds; the remaining 3 folds were equivalent, up to a permutation of the prefix rules, and started with the same first prefix rule.

F.3 NYCLU Stop-and-frisk Data Set

We show example optimal rule lists that predict whether a weapon will be found on a stopped individual who is frisked or searched, learned from the NYCLU data set. Figure 23 shows regularization parameters $\lambda = 0.04$ and 0.01, and Figure 24 shows $\lambda = 0.0025$. We showed a representative solution for $\lambda = 0.01$ in Figure 5 (§6.3).

For each of the two larger regularization parameters in Figure 23, $\lambda = 0.04$ (top) and 0.01 (bottom), we observe that across the folds, all the optimal rule lists contain the same or equivalent rules, up to a permutation. With the smaller regularization parameter $\lambda = 0.0025$ (Figure 24), we observe a greater diversity of longer optimal rule lists, though they share similar structure.

Weapon prediction ($\lambda = 0.04$)

if (*stop reason* = *suspicious object*) **then predict** *yes* ▷ **Found by 7 folds**
else if (*stop reason* ≠ *suspicious bulge*) **then predict** *no*
else predict *yes*

if (*stop reason* = *suspicious bulge*) **then predict** *yes* ▷ **Found by 3 folds**
else if (*stop reason* ≠ *suspicious object*) **then predict** *no*
else predict *yes*

Weapon prediction ($\lambda = 0.01$)

if (*stop reason* = *suspicious object*) **then predict** *yes* ▷ **Found by 4 folds**
else if (*location* = *transit authority*) **then predict** *yes*
else if (*stop reason* ≠ *suspicious bulge*) **then predict** *no*
else predict *yes*

if (*location* = *transit authority*) **then predict** *yes* ▷ **Found by 3 folds**
else if (*stop reason* = *suspicious bulge*) **then predict** *yes*
else if (*stop reason* = *suspicious object*) **then predict** *yes*
else predict *no*

if (*location* = *transit authority*) **then predict** *yes* ▷ **Found by 2 folds**
else if (*stop reason* = *suspicious object*) **then predict** *yes*
else if (*stop reason* = *suspicious bulge*) **then predict** *yes*
else predict *no*

if (*location* = *transit authority*) **then predict** *yes* ▷ **Found by 1 fold**
else if (*stop reason* = *suspicious object*) **then predict** *yes*
else if (*stop reason* ≠ *suspicious bulge*) **then predict** *no*
else predict *yes*

Figure 23: Example optimal rule lists for the NYCLU stop-and-frisk data set, found by CORELS with regularization parameters $\lambda = 0.04$ (top) and 0.01 (bottom), across 10 cross-validation folds.

Weapon prediction ($\lambda = 0.0025$)

if (*stop reason* = *suspicious object*) **then predict** *yes* ▷ **Found by 4 folds** ($K = 7$)
else if (*stop reason* = *casing*) **then predict** *no*
else if (*stop reason* = *suspicious bulge*) **then predict** *yes*
else if (*stop reason* = *fits description*) **then predict** *no*
else if (*location* = *transit authority*) **then predict** *yes*
else if (*inside or outside* = *inside*) **then predict** *no*
else if (*city* = *Manhattan*) **then predict** *yes*
else predict *no*

if (*stop reason* = *suspicious object*) **then predict** *yes* ▷ **Found by 1 fold** ($K = 6$)
else if (*stop reason* = *casing*) **then predict** *no*
else if (*stop reason* = *suspicious bulge*) **then predict** *yes*
else if (*stop reason* = *fits description*) **then predict** *no*
else if (*location* = *housing authority*) **then predict** *no*
else if (*city* = *Manhattan*) **then predict** *yes*
else predict *no*

if (*stop reason* = *suspicious object*) **then predict** *yes* ▷ **Found by 1 fold** ($K = 6$)
else if (*stop reason* = *suspicious bulge*) **then predict** *yes*
else if (*location* = *housing authority*) **then predict** *no*
else if (*stop reason* = *casing*) **then predict** *no*
else if (*stop reason* = *fits description*) **then predict** *no*
else if (*city* = *Manhattan*) **then predict** *yes*
else predict *no*

if (*stop reason* = *suspicious object*) **then predict** *yes* ▷ **Found by 1 fold** ($K = 6$)
else if (*stop reason* = *casing*) **then predict** *no*
else if (*stop reason* = *suspicious bulge*) **then predict** *yes*
else if (*stop reason* = *fits description*) **then predict** *no*
else if (*location* = *housing authority*) **then predict** *no*
else if (*city* = *Manhattan*) **then predict** *yes*
else predict *no*

if (*stop reason* = *drug transaction*) **then predict** *no* ▷ **Found by 1 fold** ($K = 8$)
else if (*stop reason* = *suspicious object*) **then predict** *yes*
else if (*stop reason* = *suspicious bulge*) **then predict** *yes*
else if (*location* = *housing authority*) **then predict** *no*
else if (*stop reason* = *fits description*) **then predict** *no*
else if (*stop reason* = *casing*) **then predict** *no*
else if (*city* = *Manhattan*) **then predict** *yes*
else if (*city* = *Bronx*) **then predict** *yes*
else predict *no*

if (*stop reason* = *suspicious object*) **then predict** *yes* ▷ **Found by 1 fold** ($K = 9$)
else if (*stop reason* = *casing*) **then predict** *no*
else if (*stop reason* = *suspicious bulge*) **then predict** *yes*
else if (*stop reason* = *fits description*) **then predict** *no*
else if (*location* = *transit authority*) **then predict** *yes*
else if (*inside or outside* = *inside*) **then predict** *no*
else if (*city* = *Manhattan*) **then predict** *yes*
else if (*additional circumstances* = *changed direction*) **then predict** *no*
else if (*city* = *Bronx*) **then predict** *yes*
else predict *no*

if (*stop reason* = *suspicious object*) **then predict** *yes* ▷ **Found by 1 fold** ($K = 10$)
else if (*stop reason* = *casing*) **then predict** *no*
else if (*stop reason* = *suspicious bulge*) **then predict** *yes*
else if (*stop reason* = *actions of violent crime*) **then predict** *no*
else if (*stop reason* = *fits description*) **then predict** *no*
else if (*location* = *transit authority*) **then predict** *yes*
else if (*inside or outside* = *inside*) **then predict** *no*
else if (*city* = *Manhattan*) **then predict** *yes*
else if (*additional circumstances* = *evasive response*) **then predict** *no*
else if (*city* = *Bronx*) **then predict** *yes*
else predict *no*

Figure 24: Example optimal rule lists for the NYCLU stop-and-frisk data set $\lambda = 0.0025$.

Appendix G. Additional Results on Predictive Performance and Model Size for CORELS and Other Algorithms

In this appendix, we plot TPR, FPR, and model size for CORELS and three other algorithms, using the NYPD data set (Feature Set D).

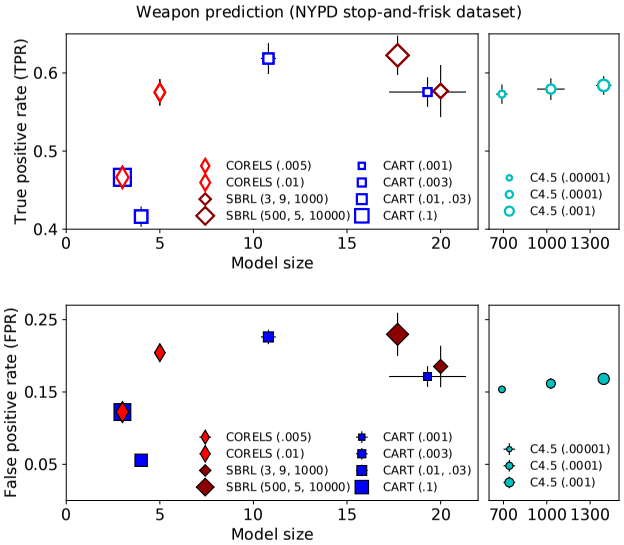


Figure 25: TPR (top) and FPR (bottom) for the test set, as a function of model size, across different methods, for weapon prediction with the NYPD stop-and-frisk data set (Feature Set D). In the legend, numbers in parentheses are algorithm parameters, as in Figure 12. Legend markers and error bars indicate means and standard deviations, respectively, across cross-validation folds. C4.5 finds large models for all tested parameters.

References

E. Angelino, N. Larus-Stone, D. Alabi, M. Seltzer, and C. Rudin. Learning certifiably optimal rule lists for categorical data. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2017.

K. P. Bennett and J. A. Blue. Optimal decision trees. Technical report, R.P.I. Math Report No. 214, Rensselaer Polytechnic Institute, 1996.

I. Bratko. Machine learning: Between accuracy and interpretability. In *Learning, Networks and Statistics*, volume 382 of *International Centre for Mechanical Sciences*, pages 163–

177. Springer Vienna, 1997.

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

S. Bushway. Is there any logic to using logit. *Criminology & Public Policy*, 12(3):563–567, 2013.

C. Chen and C. Rudin. An optimization approach to learning falling rule lists. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

H. A. Chipman, E. I. George, and R. E. McCulloch. Bayesian CART model search. *Journal of the American Statistical Association*, 93(443):935–948, 1998.

H. A. Chipman, E. I. George, and R. E. McCulloch. Bayesian treed models. *Machine Learning*, 48(1):299–320, 2002.

H. A. Chipman, E. I. George, and R. E. McCulloch. BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010.

P. Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3:261–283, 1989.

W. W. Cohen. Fast effective rule induction. In *International Conference on Machine Learning (ICML)*, pages 115–123, 1995.

R. M. Dawes. The robust beauty of improper linear models in decision making. *American Psychologist*, 34(7):571–582, 1979.

D. Dension, B. Mallick, and A.F.M. Smith. A Bayesian CART algorithm. *Biometrika*, 85 (2):363–377, 1998.

D. Dobkin, T. Fulton, D. Gunopulos, S. Kasif, and S. Salzberg. Induction of shallow decision trees, 1996.

J. Dressel and H. Farid. The accuracy, fairness, and limits of predicting recidivism. *Science Advances*, 4(1), 2018.

A. Farhangfar, R. Greiner, and M. Zinkevich. A fast way to produce optimal fixed-depth decision trees. In *International Symposium on Artificial Intelligence and Mathematics (ISAIM)*, 2008.

E. Frank and I. H. Witten. Generating accurate rule sets without global optimization. In *International Conference on Machine Learning (ICML)*, pages 144–151, 1998.

A. A. Freitas. Comprehensible classification models: A position paper. *ACM SIGKDD Explorations Newsletter*, 15(1):1–10, 2014.

M. Garofalakis, D. Hyun, R. Rastogi, and K. Shim. Efficient algorithms for constructing decision trees with constraints. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 335–339, 2000.

C. Giraud-Carrier. Beyond predictive accuracy: What? In *ECML-98 Workshop on Upgrading Learning to Meta-Level: Model Selection and Data Transformation*, pages 78–85, 1998.

S. Goel, J. M. Rao, and R. Shroff. Precinct or prejudice? Understanding racial disparities in New York City's stop-and-frisk policy. *The Annals of Applied Statistics*, 10(1):365–394, 03 2016.

S. T. Goh and C. Rudin. Box drawings for learning with imbalanced data. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2014.

S. T. Goh and C. Rudin. A minimax surrogate loss approach to conditional difference estimation. *CoRR*, abs/1803.03769, 2018. URL `https://arxiv.org/abs/1803.03769`.

B. Goodman and S. Flaxman. European Union regulations on algorithmic decision-making and a "right to explanation". In *ICML Workshop on Human Interpretability in Machine Learning (WHI)*, 2016.

R. C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11(1):63–91, 1993.

J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, and B. Baesens. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems*, 51(1):141–154, 2011.

V. Kaxiras and A. Saligrama. Building predictive models with rule lists, 2018. URL `https://corels.eecs.harvard.edu`.

H. Lakkaraju and C. Rudin. Cost-sensitive and interpretable dynamic treatment regimes based on rule lists. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

J. Larson, S. Mattu, L. Kirchner, and J. Angwin. How we analyzed the COMPAS recidivism algorithm. *ProPublica*, 2016.

N. Larus-Stone, E. Angelino, D. Alabi, M. Seltzer, V. Kaxiras, A. Saligrama, and C. Rudin. Systems optimizations for learning certifiably optimal rule lists. In *SysML Conference*, 2018.

N. L. Larus-Stone. *Learning Certifiably Optimal Rule Lists: A Case For Discrete Optimization in the 21st Century*. 2017. Undergraduate thesis, Harvard College.

B. Letham, C. Rudin, T. H. McCormick, and D. Madigan. Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3):1350–1371, 2015.

O. Li, H. Liu, C. Chen, and C. Rudin. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*, 2018.

W. Li, J. Han, and J. Pei. CMAR: Accurate and efficient classification based on multiple class-association rules. *IEEE International Conference on Data Mining (ICDM)*, pages 369–376, 2001.

J. T. Linderoth and M. W. P. Savelsbergh. A computational study of search strategies for mixed integer programming. *INFORMS Journal on Computing*, 11(2):173–187, 1999.

B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 80–96, 1998.

M. Marchand and M. Sokolova. Learning with decision lists of data-dependent features. *Journal of Machine Learning Research*, 6:427–451, 2005.

R. S. Michalski. On the quasi-minimal solution of the general covering problem. In *International Symposium on Information Processing*, pages 125–128, 1969.

New York Civil Liberties Union. Stop-and-frisk data, 2014. URL `http://www.nyclu.org/content/stop-and-frisk-data`.

New York Police Department. Stop, question and frisk data, 2016. URL `http://www1.nyc.gov/site/nypd/stats/reports-analysis/stopfrisk.page`.

S. Nijssen and E. Fromont. Optimal constraint-based decision tree induction from itemset lattices. *Data Mining and Knowledge Discovery*, 21(1):9–51, 2010.

J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

P. R. Rijnbeek and J. A. Kors. Finding a short and accurate decision rule in disjunctive normal form by exhaustive search. *Machine Learning*, 80(1):33–62, July 2010.

R. L. Rivest. Learning decision lists. *Machine Learning*, 2(3):229–246, November 1987.

U. Rückert and L. De Raedt. An experimental evaluation of simplicity in rule learning. *Artificial Intelligence*, 172:19–28, 2008.

C. Rudin and Ş. Ertekin. Learning customized and optimized lists of rules with mathematical programming. Submitted, 2016.

C. Rudin, B. Letham, and D. Madigan. Learning theory analysis for association rules and sequential event prediction. *Journal of Machine Learning Research*, 14:3384–3436, 2013.

S. Rüping. *Learning interpretable models*. PhD thesis, Universität Dortmund, 2006.

G. Shmueli. To explain or to predict? *Statistical Science*, 25(3):289–310, August 2010.

M. Sokolova, M. Marchand, N. Japkowicz, and J. Shawe-Taylor. The decision list machine. In *Advances in Neural Information Processing Systems (NIPS)*, volume 15, pages 921–928, 2003.

N. Tollenaar and P. van der Heijden. Which method predicts recidivism best?: A comparison of statistical, machine learning and data mining predictive models. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 176(2):565–584, 2013.

B. Ustun and C. Rudin. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 102(3):349–391, 2016.

B. Ustun and C. Rudin. Optimized risk scores. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2017.

K. Vanhoof and B. Depaire. Structure of association rule classifiers: A review. In *International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, pages 9–12, 2010.

A. Vellido, J. D. Martín-Guerrero, and P. J.G. Lisboa. Making machine learning models interpretable. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 2012.

F. Wang and C. Rudin. Falling rule lists. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015a.

F. Wang and C. Rudin. Causal falling rule lists. *CoRR*, abs/1510.05189, 2015b. URL `https://arxiv.org/abs/1510.05189`.

T. Wang. Hybrid decision making: When interpretable models collaborate with black-box models. *CoRR*, abs/1802.04346, 2018. URL `http://arxiv.org/abs/1802.04346`.

T. Wang, C. Rudin, F. Doshi-Velez, Y. Liu, E. Klampfl, and P. MacNeille. Bayesian or's of and's for interpretable classification with application to context aware recommender systems. In *International Conference on Data Mining (ICDM)*, 2016.

T. Wang, C. Rudin, F. Doshi-Velez, Y. Liu, E. Klampfl, and P. MacNeille. A Bayesian framework for learning rule sets for interpretable classification. *Journal of Machine Learning Research*, 18(70):1–37, 2017.

E. Westervelt. Did a bail reform algorithm contribute to this San Francisco man's murder?, 2017. URL `https://www.npr.org/2017/08/18/543976003/did-a-bail-reform-algorithm-contribute-to-this-san-francisco-man-s-murder`.

H. Yang, C. Rudin, and M. Seltzer. Scalable Bayesian rule lists. In *International Conference on Machine Learning (ICML)*, 2017.

X. Yin and J. Han. CPAR: Classification based on predictive association rules. In *SIAM International Conference on Data Mining (SDM)*, pages 331–335, 2003.

J. Zeng, B. Ustun, and C. Rudin. Interpretable classification models for recidivism prediction. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 180(3):689–722, 2017.

Y. Zhang, E. B. Laber, A. Tsiatis, and M. Davidian. Using decision lists to construct interpretable and parsimonious treatment regimes. *Biometrics*, 71(4):895–904, 2015.